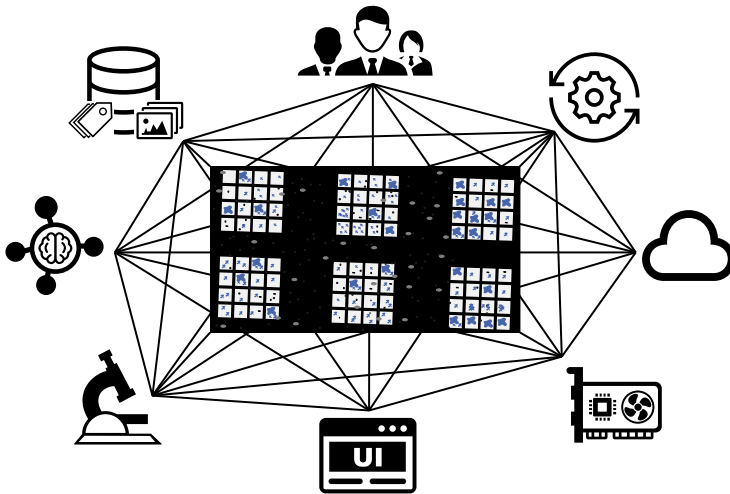


Datenzentriertes Deep Learning für die Hochdurchsatzbildverarbeitung in interdisziplinären Projekten

Marcel Peter Schilling



*Für meine lieben Eltern Gabi und Peter sowie
meine lieben Schwestern Vanessa und Tatjana.*

Datenzentriertes Deep Learning für die Hochdurchsatzbildverarbeitung in interdisziplinären Projekten

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

angenommene
DISSERTATION

von

M.Sc.

Marcel Peter Schilling

aus Freudenstadt

Tag der mündlichen Prüfung:
Erster Gutachter:
Zweite Gutachterin:

15.12.2023
Prof. Dr.-Ing. Markus Reischl
Prof. Dr. rer. nat. Britta Nestler

Gender-Hinweis

Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten – sofern nicht explizit kenntlich gemacht – gleichermaßen für alle Geschlechter.

Abstract

High-throughput experiments are often used in biology, chemistry, medicine, or pharmacy to run large series of experiments quickly and reproducibly. For example, high-throughput experiments are suitable for research in the field of personalized oncology to perform drug sensitivity screening. Experiments executed via platforms, such as droplet microarray, are often divided into hundreds of grid-like sub-experiments to investigate various configurations. Imaging techniques like microscopy are mainly used to interpret the results. However, manual image analysis done by researchers is impractical due to the large amount of data and reproducibility issues.

Machine learning, especially deep learning, is suitable for interpreting complex patterns in data. State-of-the-art machine learning algorithms require annotated datasets for supervised learning. However, annotating data is time-consuming, expensive, and demotivating for domain experts. Consequently, the scarcity of publicly available annotated datasets and the lack of strategies to reduce the annotation effort in the context of high-throughput experiments pose problems. In addition, researchers in interdisciplinary projects often encounter challenges in using elaborated machine learning algorithms. For example, software packages including graphical user interfaces, powerful hardware devices, or concepts for data management are missing.

This thesis presents a comprehensive concept for solving the above-mentioned open problems. In addition, this work provides a corresponding implementation of the concept. In particular, this thesis focuses on the data perspective, i.e., data annotation, annotation quality, or leveraging of all available data (data-centric deep learning), compared to previous investigations regarding optimizing neural network architectures.

New methods for assistance in the annotation process are presented to increase users' efficiency while improving the annotations' quality. Moreover, this work contributes a novel image processing pipeline for evaluating grid-like high-throughput experiments, which can be used to solve heterogeneous problems accurately. Furthermore, methods for solving typical tasks in high-throughput image processing are presented, e.g., image segmentation, object detection, or colorimetric analysis. The annotation-efficient learning methods transfer learning, self-supervised learning, deep active learning, and semi-supervised learning are systematically investigated and compared. This analysis provides researchers with application recommendations for alternatives w.r.t. the parameterization of algorithms compared to the established supervised learning approach.

Furthermore, the practical implementation of the overall concept regarding hardware, software design, data management, and user access is discussed. Besides, the configured image processing server prototype, including the developed open-source software packages KaIDA, Grid Screener, and AI²Seg, is shown. This prototype has been developed as part of the "Screening Platform for Personalized Oncology" project and is used extensively by project participants.

Finally, the transferability of the developed concept is demonstrated using various biomedical image processing problems. Data from the personalized oncology project and different public datasets from challenges are utilized to solve tasks, such as segmenting cell nuclei in histology and microscopy images, calculating cell viability, interpreting pH values based on colorimetric information, or detecting insects.

Kurzfassung

Hochdurchsatzexperimente werden häufig in der Biologie, Chemie, Medizin oder Pharmazie eingesetzt, um große Versuchsreihen schnell und reproduzierbar durchzuführen. Z. B. eignen sich die Experimente für Forschung im Bereich der personalisierten Onkologie, um die Sensitivität des jeweiligen Patientenzellmaterials gegenüber verschiedenen Medikamenten zu untersuchen. Häufig sind die Versuche, die mittels Plattformen wie Droplet Microarray durchgeführt werden, gitterförmig angeordnet und entsprechend in hunderte Teil-Experimente aufgeteilt, um verschiedenste Konfigurationen zu betrachten. Bildgebende Verfahren, wie die Mikroskopie, werden meist zur Interpretation der Ergebnisse genutzt. Allerdings ist eine manuelle Bildanalyse durch Forscher aufgrund der umfangreichen Daten und Probleme im Hinblick auf Reproduzierbarkeit nicht praktikabel.

Maschinelles Lernen, insbesondere Deep Learning, eignet sich für die Interpretation komplexer Muster in Daten. Algorithmen im Bereich des maschinellen Lernens benötigen für überwachte Lernverfahren annotierte Datensätze. Das Annotieren von Daten ist jedoch eine zeitaufwändige, teure und demotivierende Aufgabe für Experten. Der Mangel an öffentlich zugänglichen annotierten Datensätzen und das Fehlen von Strategien zur Verringerung des Annotationsaufwands im Kontext von Hochdurchsatzexperimenten stellen daher Probleme dar. Darüber hinaus ergeben sich für Forscher in interdisziplinären Projekten häufig Herausforderungen bei der Verwendung von Algorithmen. Es fehlen Softwarepakete, einschließlich grafischer Benutzeroberflächen, leistungsfähige Hardwaregeräte oder Konzepte für das Datenmanagement.

Diese Arbeit stellt ein umfassendes Konzept zur Lösung der genannten offenen Probleme vor und liefert zudem eine entsprechende Implementierung

von dessen. Insbesondere die Datenperspektive, d. h. Themen wie die Annotation von Daten, die Annotationsqualität oder die Nutzung aller verfügbaren Daten (datenzentriertes Deep Learning), stehen im Mittelpunkt der Arbeit, im Vergleich zu bisherigen Forschungsarbeiten, die ihren Fokus oftmals auf die Optimierung von Netzarchitekturen gelegt haben.

Es werden neue Methoden zur Unterstützung des Annotationsprozesses vorgestellt, um die Effizienz für Nutzer zu erhöhen und gleichzeitig die Qualität der Annotationen zu verbessern. Die Arbeit stellt eine neue Verarbeitungspipeline zur Auswertung gitterförmiger Hochdurchsatzexperimente bereit, mit der heterogene Probleme in hoher Güte gelöst werden können. Es werden Methoden zur Lösung gängiger Aufgaben in der Hochdurchsatz-Bildverarbeitung vorgestellt, z. B. die Segmentierung von Bildern, die Erkennung von Objekten in Bildern oder die Analyse von Farbverläufen. Zudem werden die annotations-effizienten Lernverfahren Transfer Learning, Self-Supervised Learning, Deep Active Learning und Semi-Supervised Learning systematisch untersucht und verglichen. Diese Analyse liefert Forschern Anwendungsempfehlungen für Alternativen bei der Parametrisierung von lernbasierten Algorithmen im Vergleich zum etablierten überwachten Lernen.

Zudem wird die praktische Umsetzung des Gesamtkonzepts abstrakt diskutiert und dabei auf die Gestaltung von Hardware, Software, Datenerhaltung sowie Nutzerzugriff eingegangen. Ferner wird die prototypische Bildverarbeitungs-Serverumgebung inklusive der entwickelten öffentlich zugänglichen Softwarepakete KaIDA, Grid Screener und AI²Seg gezeigt. Dieser Prototyp ist im Rahmen des Projekts „Screening Platform for Personalized Oncology“ aufgebaut und umfangreich von Projektteilnehmern genutzt worden.

Abschließend wird die Anwendbarkeit des Konzepts anhand verschiedener biomedizinischer Bildverarbeitungsprobleme demonstriert. Dazu werden Daten aus dem Projekt zu personalisierter Onkologie und verschiedene öffentliche Datensätze aus Wettbewerben verwendet. Zu lösende Aufgaben sind z. B. das Segmentieren von Zellen in Histologie- und Mikroskopie-Bildern, das Berechnen der Überlebensrate von Zellen, das Interpretieren von pH-Werten basierend auf Farbinformationen oder das Erkennen von Insekten.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automation und angewandte Informatik des Karlsruher Instituts für Technologie. Ohne die vielfältige Unterstützung von Familie, Kollegen und Freunden wäre dieser lange Weg nicht möglich gewesen.

Mein herzlichster Dank gilt meinem Doktorvater, Herrn Prof. Dr. Markus Reichl, für die ausgezeichnete Betreuung dieser Dissertation. Ich danke ihm für die vielfältigen bereichernden fachlichen Diskussionen, seinen Pragmatismus, die motivierenden Worte, das Schaffen von hervorragenden Rahmenbedingungen zur Erstellung der Arbeit und nicht zuletzt für seine Menschlichkeit während der Zeit als Doktorand in seiner Forschungsgruppe. Stets stand er mit Unterstützung und offenem Ohr bei Herausforderungen zur Seite. Darüber hinaus danke ich Frau Prof. Dr. Britta Nestler für die Übernahme des Korreferats und das hiermit verbundene Interesse an der Thematik, das zum Gelingen der Arbeit beigetragen hat.

Zudem danke ich meinen Kollegen für die sehr angenehme und unterhaltsame Arbeitsatmosphäre, die spannenden Diskussionen in Kaffeepausen, gemeinsamen Erfahrungen auf Dienstreisen, Erlebnisse bei Rennradtouren und anderen Sportevents sowie sonstige gemütliche Runden. Im Speziellen danke dem besten Zimmerkollegen Oliver Neumann und den restlichen Kollegen bzw. Freunden des Diversity-Scanner-Meetings (Moritz Böhland, Roman Bruch, Luca Rettenberger, André Orth, Lorenz Wühl, Tim Scherr, Friedrich Münke, Jan Schützke, Lisa Petani, Mark Schutera) für die Zeit im Büro, bei der das Lachen nie zu kurz gekommen ist. Des Weiteren möchte ich den Studierenden (Lukas Klinger, Svenja Schmelzer, Luca Rettenberger, Fabio Rodrigues, Gustav Keppler, Ulrike Schumacher, Jona Stuck, Niket Ahuja, Himanshu

Maurya) und verschiedenen Projektpartnern (Anna Popova, Joaquín Urrutia, Razan El Khaled El Faraj, Janne Wiedmann, Maryam Salarian, Steffen Sonntag, Haijun Cui, Charlotte Luchena, Pavel Levkin) für die vielen spannenden gemeinsamen Projekte und deren reichliche Unterstützung danken. Darüber hinaus danke ich Miguel Bordallo López für den tollen internationalen Austausch in Finnland. Ferner geht mein Dank an meinen spanischen Kollegen Miguel Molina-Moreno, mit dem neben der Forschungs Kooperation eine länderübergreifende Freundschaft gewachsen ist. Außerdem danke ich Claudia Greceanu für die Hilfe beim Korrekturlesen der Arbeit und Bernadette Lehmann für Ihre vielfältigen Unterstützungen während der Arbeit am Institut.

Ferner danke ich den Autoren in [Gri19] für die Bereitstellung der zum Erstellen der Arbeit genutzten \LaTeX -Formatvorlage.

Zudem möchte ich Marco Möhrle, meinem besten Freund aus der Heimat, für die Unterstützung auf meinem Lebensweg seit Kindheitstagen danken. Ferner danke ich meinen Mitbewohnern (Jeanine Jutzi, Max Hartenbach, Lukas Schröder, Alexander Hornberger) für die Zeit während bzw. abseits des Studiums und der Promotion.

Abschließend geht ein außerordentlicher Dank an meine lieben Eltern Gabi und Peter sowie meine Schwestern Vanessa und Tatjana. Durch ihre große und fortwährende Unterstützung auf meinem gesamten Lebensweg konnte das Fundament gelegt werden, um diese Arbeit überhaupt möglich zu machen. Vielen lieben Dank für Eure Ratschläge, die motivierenden Worte, aufbauenden Telefonate, Aufmerksamkeiten und vieles mehr. Ihr seid als Familie ein unendlich großes Geschenk!

Karlsruhe, im Dezember 2023

Marcel Schilling

Inhaltsverzeichnis

Abstract	i
Kurzfassung	iii
Danksagung	v
1 Einleitung	1
1.1 Bedeutung der Arbeit	1
1.2 Problemkategorisierung	3
1.3 Zielsetzung der Arbeit	6
1.4 Aufbau der Arbeit	7
2 Stand der Technik	9
2.1 Künstliche Intelligenz	9
2.1.1 Definitionen	9
2.1.2 Supervised Learning	10
2.1.3 Unsicherheit	11
2.2 Annotation von Bilddatensätzen	14
2.2.1 Methoden zur Effizienzsteigerung	15
2.2.2 Variabilität von Experten	17
2.3 Biomedizinische Hochdurchsatzbildverarbeitung	21
2.3.1 Überblick	21
2.3.2 Fragestellungen und Methoden	22
2.3.3 Anwendbarkeit der Bildverarbeitungsalgorithmen	27
2.4 Effizientes Lernen für Bildverarbeitung	29
2.4.1 Deep Active Learning	30

- 2.4.2 Self-Supervised Learning 32
- 2.4.3 Semi-Supervised Learning 33
- 2.4.4 Transfer Learning und Ähnlichkeiten von
Datensätzen 35
- 2.5 Zusammenfassung offener Probleme 39

- 3 Konzept und Methoden 41**
- 3.1 Konzept 41
- 3.2 Vorbereitungen 44
 - 3.2.1 Notation 44
 - 3.2.2 Bewertungsmetriken 47
- 3.3 Assistierte Annotieren 52
 - 3.3.1 Überblick 52
 - 3.3.2 Projekterstellung 54
 - 3.3.3 Selektion 54
 - 3.3.4 Pre-Processing 55
 - 3.3.5 Pre-Annotation 56
 - 3.3.6 Bildannotation 57
 - 3.3.7 Post-Annotation-Processing 58
 - 3.3.8 Annotation-Inspektion 58
 - 3.3.9 Versionskontrolle 59
 - 3.3.10 Plugins 60
- 3.4 Variabilität der Experten 60
 - 3.4.1 Überblick 60
 - 3.4.2 Simulation der Variabilität von Experten 63
 - 3.4.3 Automatisierte Inspektion der Variabilität von
Experten 65
- 3.5 Verarbeitung von Bilddaten im Hochdurchsatz 68
 - 3.5.1 Überblick 68
 - 3.5.2 Prozessierung Makroebene 68
 - 3.5.3 Prozessierung Mikroebene 75
- 3.6 Effizientes Lernen 80
 - 3.6.1 Überblick 80
 - 3.6.2 Transfer Learning 81

3.6.3	Deep Active Learning	83
3.6.4	Self-Supervised Learning	84
3.6.5	Semi-Supervised Learning	85
3.7	Architekturkonzeption für interdisziplinäre Datenanalyseprojekte	85
4	Implementierung	89
4.1	Überblick	89
4.2	Umsetzung des Architekturkonzepts	90
4.3	Deep Learning und Zugehöriges	91
4.4	KaIDA	92
4.5	AI ² Seg	95
4.6	Grid Screener	97
5	Anwendungen	99
5.1	Datensätze	99
5.2	Assistiertes Annotieren	103
5.2.1	Selektion	103
5.2.2	Pre-Processing	107
5.2.3	Pre-Annotation	108
5.2.4	Post-Annotation-Processing	112
5.2.5	Annotation-Inspektion	114
5.2.6	Software-Tool KaIDA	115
5.2.7	Variabilität der Experten	117
5.3	Verarbeitung von Bilddaten im Hochdurchsatz	123
5.3.1	Prozessierung Makroebene	123
5.3.2	Prozessierung Mikroebene	131
5.4	Effizientes Lernen	142
5.4.1	Transfer Learning	143
5.4.2	Deep Active Learning	148
5.4.3	Semi-Supervised Learning	150
5.4.4	Self-Supervised Learning	152
5.4.5	Resümee	154
5.5	Anwendung des entwickelten Konzepts	155

6 Zusammenfassung und Ausblick	159
Literatur	163
Eigene Publikationen	201
Patente	207
Betreute studentische Arbeiten	209
Abbildungsverzeichnis	211
Tabellenverzeichnis	215
Liste der Algorithmen	217
Abkürzungsverzeichnis	219
Notation	225
 Anhang	
A Algorithmen zur automatisierten Gittererkennung	235
B Algorithmen zur automatisierten Inspektion der Expertenvariabilität	237
B.1 Intra-Experten-Variabilität	237
B.2 Inter-Experten-Variabilität	239
C Methoden zur Schätzung der Ähnlichkeit von Datensätzen	241
C.1 Unsicherheit	241
C.2 Bildmerkmale	242
C.3 Kombination aus Unsicherheit und Bildmerkmalen	245
C.4 Vergleichsdatensätze	246
D Metriken	248

D.1	Korrelationen	248
D.2	Klassifikation	249
D.3	Segmentierung	250
D.4	Unsicherheit	251
E Deep Neural Networks		254
E.1	Architekturen	254
E.2	Standard-Konfigurationen	255
E.2.1	Optimierungsverfahren	255
E.2.2	Training	256
E.2.3	Vorverarbeitungen	256
E.2.4	Augmentierungen	256
E.2.5	Nachverarbeitungen	257
E.2.6	Zielfunktionen	257
F Schnellstart Bedienungsanleitungen		258
F.1	KaIDA	258
F.2	AI ² Seg	264
F.3	Grid Sceener	265
G Weitere Ergebnisse der Gitterschätzung		268
H Softwarepakete		270
I Prototypischer Bildverarbeitungsserver		271

1 Einleitung

1.1 Bedeutung der Arbeit

Hochdurchsatzexperimente (HDEs), also die schnelle Untersuchung von Tausenden hin zu Millionen verschiedenen Proben hinsichtlich deren biologischen Aktivität [Att14, Ing08], gewinnen in Biologie, Chemie, Pharmazie und Medizin an Bedeutung [Als16, Ben20, Cha22, Cui23, Gar22, Jaa22, Rei17]. Grund hierfür ist, dass HDEs Wissenschaftlern umfangreiche Untersuchungen in kurzer Zeit ermöglichen und zudem die Reproduzierbarkeit steigern [An10, Szy12]. Insbesondere die Miniaturisierung von Experimenten bringt Vorteile in Form von reduzierten Betriebskosten einhergehend mit verbesserter räumlich-zeitlicher Kontrolle mit sich [Vya10].

Als Beispiel hierfür können gitterförmige Plattformen wie Droplet Microarrays (DMAs) genannt werden [Pop16], die sich z. B. zur empirischen Ermittlung des wirksamsten Medikaments für die Therapie des individuellen Krebspatienten eignen. Die HDE-Technologie hat somit eine große Relevanz für die Gesellschaft, da mittels DMAs ein Beitrag zur Optimierung der Behandlung von Tumorerkrankungen geleistet werden kann.

Bildgebende Verfahren, z. B. mittels Mikroskop oder Kamera, sind in Experimenten gängige Praxis zur Auswertung und Interpretation der Ergebnisse. Exemplarisch können Medikamententests [Ban19], Tumoruntersuchungen [Böh22, Gra19, Pop19], Analyse von Bakterien mittels Farbumschlag [Lei20] oder die Untersuchung von embryonalen Körpern [Tro18] genannt werden, in welchen Bildverarbeitung zur Auswertung der Rohbild-daten verwendet wird. Aber auch in Produktionsprozessen spielt die digitale

Bildverarbeitung eine wichtige Rolle, wie beispielsweise zur Qualitätskontrolle von Oberflächen [Luo22, Wan18], der Erkennung von Bauteilpositionen [Bäu21] oder zur Erhöhung der Sicherheit in Montagehallen [Ahm21]. Zudem ist die Interpretation von Bilddaten auch für das automatisierte Fahren [Ise20, Ise21a] oder die Erkennung von Veränderungen in Straßenszenen [Mün21] relevant.

Ziel für die Auswertung der umfangreichen Datenmengen aus HDEs in Form von Rohbilddaten ist die **automatisierte Verarbeitung** der Bilddaten in Kombination mit **hoher Genauigkeit**. Deep Neural Networks (DNNs) zeigen in vielen Bereichen großes Potenzial im Rahmen der Datenanalyse von komplexen Zusammenhängen wie Bildern [Doe19, Lit17] durch akkurate Funktionsapproximation im Vergleich zu klassischen Bildverarbeitungsalgorithmen [Mah19].

Allerdings existieren diverse Hürden, welche die Anwendung von Deep Learning (DL) für Forscher in interdisziplinären Datenanalyseprojekten erschweren [Góm21a, von21]. Das Wissen ist im Falle eines interdisziplinären Projekts zwischen den verschiedenen Disziplinen verteilt, muss jedoch kombiniert werden, um die Daten erfolgreich auszuwerten. Daher bleibt das Potenzial von DNNs oftmals in der Praxis ungenutzt. Hohe Hardwareanforderungen oder notwendige Programmierkenntnisse hindern die Domänenexperten bei der Anwendung von DL. Jedoch können Datenanalysten ebenfalls die Auswertung aufgrund von fehlendem Wissen bezüglich der Daten nicht ohne die Domänenexperten auswerten.

Die vorliegende Arbeit kategorisiert deshalb systematisch aktuelle Probleme bezüglich des Einsatzes von DL in Hochdurchsatz (HD)-Verfahren. Es wird ein umfassendes Konzept zur Lösung der kategorisierten Probleme entwickelt und dessen Implementierung vorgestellt. Die Betrachtung von DL aus der Datenperspektive anstelle der Fokussierung auf DNN-Architekturen ist ein zentraler Baustein der Arbeit (datenzentriertes DL nach Ng [Bro22, Ng23]).

1.2 Problemkategorisierung

Die Forschungsfrage

„Wie können Rohbilddaten aus HDEs im Rahmen von interdisziplinären Datenanalyseprojekten unter Berücksichtigung von Expertenwissen automatisiert mit hoher Genauigkeit interpretiert werden?“ (vgl. Abbildung 1.1)

führt bei Betrachtung des Entwicklungsstands zu verschiedenen Teilproblemen (vgl. Abbildung 1.2):¹

Mangel an Algorithmen und Methoden zur automatisierten Bildverarbeitung Die Komplexität und Heterogenität der Aufgabenstellungen im Rahmen von HDEs erschweren die Verarbeitung von Bilddaten. Häufig ist dadurch die Etablierung von allgemein anwendbaren vollautomatisierten Verarbeitungspipelines nicht möglich, da diese meist individuell auf die jeweilige Aufgabenstellung ausgelegt und damit nicht auf andere Probleme übertragbar sind. Eine manuelle Auswertung erfordert jedoch teure personelle Ressourcen und bringt Schwierigkeiten in puncto Reproduzierbarkeit mit sich.

Unklar ist, wie das Potenzial von DNNs für eine universelle Auswertung von HDEs genutzt werden kann. Häufig sind HDEs eine Komposition vieler Einzelexperimente, die in gitterförmigen Strukturen angeordnet sind. Daher müssen Lösungen in verschiedenen Betrachtungsmaßstäben (Gitter-

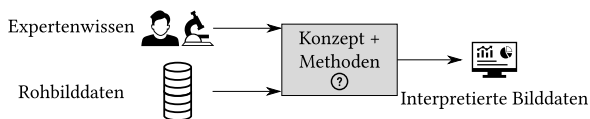


Abbildung 1.1: Visualisierung der Forschungsfrage. Ein Konzept inklusive Methoden wird gesucht, um Rohbilddaten automatisiert mit hoher Genauigkeit basierend auf Expertenwissen zu interpretieren.

¹ Eine detaillierte Beschreibung des Stands der Technik findet sich in Kapitel 2.

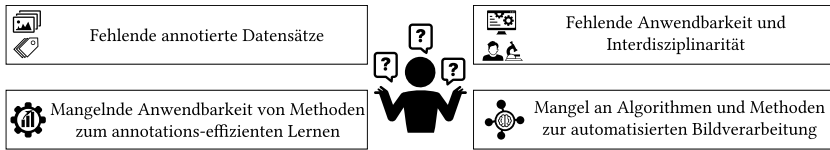


Abbildung 1.2: Übersicht der offenen Teilprobleme. Die verschiedenen offenen Teilprobleme im Rahmen der Forschungsfrage sind gezeigt.

und Einzelexperimentebene) gefunden werden, um eine vollumfängliche automatisierte Auswertung zu ermöglichen.

Fehlende annotierte Datensätze Der Fokus von bisherigen Untersuchungen im Bereich DL liegt auf der Entwicklung von elaborierteren Architekturen, die mit steigender Modellkomplexität einhergehen. Umfangreich annotierte Datensätze sind dabei für die Parametrierung im Rahmen des überwachten Lernens und Evaluierung der Modelle erforderlich.

Korrekt und konsistent annotierte Datensätze werden hierfür als gegeben vorausgesetzt, liegen aber oftmals für stark problemspezifische HDEs nicht vor. Die Problematik des Annotierens in realen Anwendungen wird nicht beleuchtet. Es fehlt an Methoden, die den Fokus auf Datensätze und deren Annotation legen (datenzentriertes DL). Fragen, wie der Aufwand für den Anwender methodisch in Praxisprojekten reduziert oder die Qualität von Annotationen automatisiert überprüft werden kann, bleiben offen.

Mangelnde Anwendbarkeit von Methoden zum annotations-effizienten Lernen Klassische überwachte Lernverfahren sind so konzipiert, dass

- lediglich Daten des jeweiligen Problems zum Lernen verwendet werden (da die Quantifizierung der Ähnlichkeit von Bilddatensätzen schwierig ist),
- diese ohne Priorisierung, also nicht gezielt nach dem jeweiligen Informationsgehalt des Bildes, annotiert werden und

- nur Informationen aus Bildern extrahiert werden, wenn eine zugehörige Annotation vorliegt.

Es existieren aktuelle Forschungsarbeiten zu Methoden, deren Ziel ist, mit möglichst wenig Annotationen hohe Genauigkeiten zu erreichen¹. Allerdings bleiben hier folgende Fragen offen: Sind diese Methoden auf Daten im Bereich HDEs sinnvoll anwendbar? Wie können die unterschiedlichen Ansätze basierend auf Kriterien verglichen werden? Eine Empfehlung für Anwender fehlt, wodurch aktuell meist überwachte Lernverfahren präferiert werden und ein immenser Annotationsaufwand für Experten verbleibt.

Interdisziplinarität und fehlende Anwendbarkeit Wissen bezüglich Algorithmen, Automatisierung oder High-Performance Computing (HPC) ist häufig bei Spezialisten aus dem Bereich Datenanalyse verortet. Jedoch ist das Fachwissen von Domänenexperten bezüglich des zugrundeliegenden Problems, dessen Interpretation und der notwendigen Auswertung unerlässlich. Ergo erfordern HDEs die Expertise aus verschiedenen Fachbereichen. Oftmals sind die entwickelten Methoden zur Datenanalyse für Domänenexperten nicht anwendbar, da zum Beispiel Rechenkapazitäten, Lösungen zum Datenmanagement, Programme oder graphische Oberflächen fehlen. Es mangelt an einem Konzept, das all diese Aspekte berücksichtigt, um eine interdisziplinäre Zusammenarbeit effektiv und effizient zu gestalten. Experten der interdisziplinären Projekte fehlt ein System, das die unterschiedlichen Disziplinen miteinander sinnvoll verknüpft.

¹ Unter dem Begriff „annotations-effizientes Lernen“ sollen im Folgenden alle Verfahren zusammengefasst werden, welche die Genauigkeit von DNNs bei möglichst geringem Einsatz von Nutzer-Annotationen optimieren.

1.3 Zielsetzung der Arbeit

Das Gesamtziel der vorliegenden Arbeit besteht darin, ein neues Konzept für die HD-Verarbeitung von Bilddaten mittels datenzentrierten DL zu entwickeln, das in interdisziplinären Projekten genutzt werden kann. Daraus lassen sich folgende **Teilziele** ableiten, die als Entwicklungsschritte für die Zielerreichung zu nennen sind und von der vorliegenden Arbeit gelöst werden:

1. Ein generisches Konzept ist zu entwickeln, welches für die HD-Verarbeitung in unterschiedlichen Themenfeldern bzw. bei verschiedenen Fragestellungen der Bildanalyse anwendbar ist und damit für Forscher als Nachschlagewerk bei neuen Projekten dienen kann.
2. Eine Implementierung des Konzepts ist zur Sicherstellung der Anwendbarkeit notwendig und muss für Forscher öffentlich zugänglich sein.
3. Quantitative und qualitative Kriterien zur Bewertung des Einsatzes der entwickelten Methoden sind auszuarbeiten bzw. auszuwählen.
4. Das Konzept soll auf verschiedenen realen Anwendungsproblemen anhand der entwickelten bzw. ausgewählten Kriterien evaluiert werden.
5. Lösungen für die benutzerfreundliche Bedienbarkeit der entwickelten Algorithmen über Graphical User Interfaces (GUIs) unter Einbezug von Hardware, Software und Datenmanagement sind zu erarbeiten, um die interdisziplinäre Zusammenarbeit in Projekten zu ermöglichen.
6. Es soll ein Algorithmus entworfen werden, mit dem die Parameter von Gitterstrukturen automatisiert bestimmt werden können. Der Algorithmus soll dabei agnostisch bezüglich der Form der Gitterelemente sein bzw. keine Annahmen zur Orientierung des Gitters machen.
7. Verschiedene Methoden für die automatisierte und genaue Auswertung von Einzelexperimenten innerhalb des Gitters sind zu entwickeln, die komplexe und heterogene Bilddaten verarbeiten können. Beispielsweise soll die Erkennung von Objekten, die Segmentierung von Bildern oder die Analyse von Farbverläufen möglich gemacht werden.

8. Für die Nutzung von datengetriebenen Verfahren sollen für Forscher konkret Lösungen zum Vereinfachen der Annotation von Datensätzen erarbeitet werden.
9. Es gilt Methoden zu entwickeln, die konsistente und fehlerfreie Annotationen gewährleisten sowie den Einfluss von falschen Annotationen untersuchen.
10. Eine kriterienbasierte Anwendungsempfehlung hinsichtlich effizienter Lernverfahren zur Reduktion des Annotationsumfangs für HDEs soll entwickelt werden, um Forschern die Methodenauswahl für das jeweilige Projekt unter den individuellen Randbedingungen zu vereinfachen.

1.4 Aufbau der Arbeit

Der Aufbau der Arbeit ist in [Abbildung 1.3](#) dargestellt. In [Kapitel 2](#) erfolgt eine Vorstellung des Stands der Technik, welcher einen Überblick hinsichtlich der Themen darstellt, die im weiteren Verlauf der Arbeit aufgegriffen werden. Darauf aufbauend wird in [Kapitel 3](#) das entwickelte Konzept inklusive verschiedener Methoden zur Lösung der Forschungsfrage vorgestellt. Details zu Implementierungen finden sich in [Kapitel 4](#). Final erfolgt die Analyse des entwickelten Konzepts in der Anwendung bezüglich verschiedener Datensätze (siehe [Kapitel 5](#)). Abschließend fasst [Kapitel 6](#) die grundlegenden Ergebnisse sowie Grenzen der vorliegenden Arbeit zusammen und gibt darüber hinaus einen Ausblick auf zukünftige Arbeiten. Ferner steht in den [Anhängen A bis I](#) Zusatzmaterial zur Erweiterung des Hauptteils der Arbeit zur Verfügung.

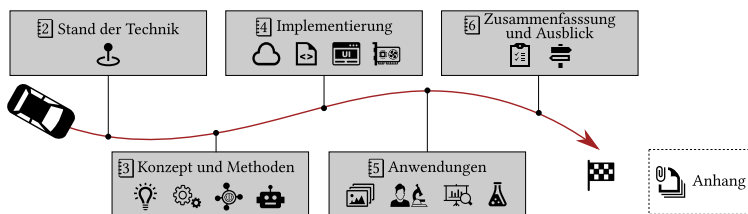


Abbildung 1.3: Aufbau der Arbeit. Der chronologische Aufbau der Arbeit ist dargestellt.

2 Stand der Technik

Das folgende Kapitel gibt einen Überblick über die Grundlagen sowie den aktuellen Entwicklungsstand¹ bezüglich der formulierten Forschungsfrage und den charakterisierten offenen Problemen (vgl. Abschnitt 1.2). In Abschnitt 2.1 erfolgt eine Darstellung der Grundlagen in den Bereichen Künstliche Intelligenz (KI), Maschinelles Lernen (ML) und Deep Learning (DL), die für das Verständnis der Folgeabschnitte notwendig sind. Der Abschnitt 2.2 widmet sich dem Themengebiet „Annotation von Bilddatensätzen“. Ferner wird in Abschnitt 2.3 detailliert auf die verschiedenen Facetten der biomedizinischen Hochdurchsatz (HD)-Bildverarbeitung eingegangen. Abschließend wird in Abschnitt 2.4 der Stand der Technik bezüglich „effizientes Lernen“ für Bildverarbeitungsprobleme präsentiert.

2.1 Künstliche Intelligenz

2.1.1 Definitionen

Nach Lakshmanan et al. [Lak21] beschäftigt sich KI mit Methoden, deren Ziel die Imitation von menschlichem Verhalten und Fähigkeiten durch Algorithmen ist. ML ist dabei ein Teilgebiet der KI. Unter dem Begriff ML werden Algorithmen zusammengefasst, die im Rahmen eines Lernprozesses eine Aufgabe lösen, indem ein Gütekriterium durch Erfahrung aus Daten verbessert wird [Alp10, Jor15, Lak21, Mit97, Moh12]. Daher wird auch oft von datengetriebener Entwicklung gesprochen. Ziel in der Anwendung ist es, genaue

¹ Die Darstellung des Entwicklungsstandes stellt die Literaturrecherche nach bestem Wissen des Autors dar.

Ergebnisse bezüglich unbekannter Daten zu erzielen. Es wird das Ermitteln eines allgemeingültigen Zusammenhangs in den Daten angestrebt, was unter dem Begriff Generalisierung zusammengefasst wird [Mit97, Moh12]. Allerdings ist in der Vergangenheit bei der Nutzung von klassischen ML-Verfahren häufig eine aufwändige Vorverarbeitung der Rohdaten notwendig gewesen, um sinnvolle Merkmale für das ML-System zu extrahieren [LeC15].

Bei DL handelt es sich um ein Teilgebiet von ML [Goo16, Lak21, Moe19]. Hierbei werden Merkmale und deren Interpretation in einem geschlossenen Ansatz gelernt (engl. end-to-end), d. h. es werden alle Schritte zur Übertragung der Rohdaten auf die Ausgabe zusammenhängend gelernt. Den Ausgangspunkt bilden im Falle von DL neuronale Netze (engl. Neural Networks (NNs)), wobei es sich um an die Biologie angelehnte Algorithmen handelt, welche die Verschaltungen der Neuronen im Gehirn nachahmen [Hub62, Sch15]. Die Neuronen können Nichtlinearitäten abbilden, indem diese als Basiselemente mit Aktivierungsfunktionen kombiniert werden. Signale werden damit, je nach Eingangswerten, an benachbarte Schichten unterschiedlich weitergegeben. Bei vielen vorliegenden Neuronenschichten, die miteinander zu tiefen Netzen verschaltet sind, um komplexe Zusammenhänge zu approximieren [Lak21, LeC15], spricht man von DL. Folglich finden Deep Neural Networks (DNNs) beispielsweise bei komplexen Problemen wie der Bild- oder Sprachverarbeitung Anwendung [LeC15].

Das veröffentlichte AlexNet [Kri12] stellt einen Meilenstein in der Historie von DL dar, da erstmalig ein DNN bei der Klassifikation des ImageNet-Datensatzes [Den09] etablierte ML-Algorithmen hinsichtlich der Klassifikationsgüte übertroffen hat.

2.1.2 Supervised Learning

Gängige Vorgehensweise zur Einstellung von ML-Modellen ist das Supervised Learning (SL), dessen Bestandteile und Ablaufprozess im Weiteren anhand der Abbildung 2.1 gemäß den Erläuterungen in [Goo16, LeC15] illustriert werden.

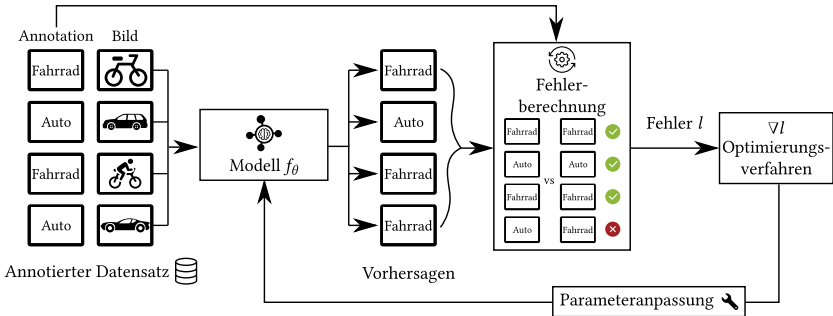


Abbildung 2.1: Supervised Learning. Zu Illustrationszwecken wird ein Beispiel zur Klassifikation von Fahrzeugen der Klassen „Fahrrad“ und „Auto“ verwendet. Mittels eines annotierten Datensatzes kann das Modell f_θ mit Parametern θ Vorhersagen für gegebene Bilder erzeugen. In der Fehlerberechnung wird über einen Vergleich zwischen Vorhersage und Annotation der Fehler l berechnet. Im Anschluss erfolgt die Bestimmung der angepassten Modellparameter mittels eines Optimierungsverfahrens unter Verwendung des Gradientens ∇l .

Ausgangspunkt für SL bildet ein annotierter Datensatz, der beispielsweise aus Bilddaten und korrespondierenden Annotationen (z. B. im Falle einer Klassifikation von Fahrzeugen: „Fahrrad“ oder „Auto“) bestehen kann. Über ein beliebiges Modell f_θ , das über Parameter θ einstellbar ist, können Vorhersagen bezüglich der Eingabebilder generiert werden. Ein Vergleich zwischen Annotation und Vorhersage ermöglicht die Berechnung des Fehlers l . Unter Verwendung eines Optimierungsverfahrens können damit im Anschluss die angepassten Parameter θ zur Reduktion des Fehlers l errechnet werden. Die Optimierungsalgorithmen sind im Bereich DL elaborierte Gradientenabstiegsverfahren wie Adaptive Moment Estimation (Adam) [Kin15] oder Stochastic Gradient Descent (SGD) [Rob51]. Zusammenfassend kann festgehalten werden, dass Annotationen im Rahmen des SL essenziell sind, um die Parameter des Modells während des Lernprozesses geeignet einzustellen bzw. um die Güte eines Modells f_θ zu bewerten.

2.1.3 Unsicherheit

Eine Mehrheit der Bestrebungen im Kontext DL widmet sich Optimierungen hinsichtlich der Leistungsfähigkeit von Modellen, also der Minimierung des

Fehlers l (siehe Abschnitt 2.1.2) bei Vorhersagen. Allerdings ist die Quantifizierung der Unsicherheit in vielen Anwendungen ebenfalls von großer Relevanz [Abd21, Jia18]. Der Begriff Unsicherheit wird verwendet, um das Fehlschlagen von Modellvorhersagen zu quantifizieren [Cor19, Ken17]. Folglich soll damit das Versagen des Modells und ein Mangel an Generalisierung charakterisiert werden. Nach Viering und Loog kann im Idealfall von einem monotonen Verlauf zwischen Generalisierung und Datensatzumfang ausgegangen werden [Vie22].

„Adversarial Attacks“ (zu Deutsch: feindliche Angriffe) machen auf die Schwachstellen von DNNs aufmerksam, indem z. B. leichte Veränderungen von Eingabebildern zu falschen Vorhersagen im Falle der Bildklassifikation [Ey18] führen können. Das Vertrauen des DNN, ausgedrückt durch eine eindeutige Klassenzuweisung im Ausgang des Netzes, ist in diesem Fall allerdings wider Erwarten hoch. DNNs sind häufig zu sicher im Falle von Fehlern [Cor19, Gal16]. Es kann gezeigt werden, dass DNNs nicht kalibriert¹ sind [Cor19, Guo17]. Ergo ist die Ermittlung der Modellunsicherheit von Interesse.

Nach den Ausführungen in [Hül21, Ken17] ist die Unsicherheit von DNNs eine Komposition aus zwei Bestandteilen. Einerseits wird mittels *aleatorischer* Unsicherheit die Zufälligkeit der Eingabe berücksichtigt, wie z. B. das Messrauschen von Sensoren. Dementsprechend kann diese Art der Unsicherheit nicht durch einen größeren Datensatz reduziert werden. Andererseits setzt sich nach Huang et al. [Hua21] die *epistemische* Unsicherheit zusammen aus:

- der Unsicherheit durch die Auswahl einer Hypothese als Modell zur Approximation des Zusammenhangs,
- der Variabilität bei der Optimierung einer nicht-konvexen Zielfunktion und

¹ Nach Guo et al. [Guo17] beschreibt der Begriff „fehlende Kalibrierung“ die Eigenschaft, dass bei DNNs das Vertrauen im Ausgang nicht unbedingt mit der Korrektheit oder einer Wahrscheinlichkeitsverteilung einhergeht. Z. B. sollen bei einem Vertrauen von 0.9 im Falle von 1000 Vorhersagen zirka 900 korrekt sein, wenn ein kalibriertes DNN vorliegt.

- der Variabilität in den Daten, da nur eine Teilmenge aller Daten während des Trainings verfügbar ist.

Somit kann das Sammeln von mehr Daten helfen, die epistemische Unsicherheit eines DNN zu reduzieren.

Im Weiteren sollen konkrete Methoden zur Schätzung der Unsicherheit von DNNs vorgestellt werden.

Bayes-DNNs Nach Wilson und Izmailov [Wil20] unterscheiden sich Bayes-DNNs zu Standard-DNNs, da die Parameter hier als Verteilungen betrachtet werden, die über Marginalisierung statt direkter Optimierung bestimmt werden. Ein Ansatz zur Schätzung der Verteilung am Ausgang des DNN und damit zur Bestimmung der Unsicherheit ist Monte Carlo Dropout (MCD) [Ken17]. Dropout [Sri14], also das zufällige Deaktivieren von Neuronen, was eigentlich als Regularisierungstechnik¹ eingesetzt wird, kann zur Approximation eines Bayes-DNN genutzt werden. Die Dropoutschichten bleiben während der Inferenz aktiv und somit ergeben sich leicht veränderte Netze, um eine Stichprobenmenge bezüglich des Ausgangs zu generieren, mit der die Ausgabeverteilung geschätzt werden kann. Ferner ist es über eine spezielle Modellarchitektur mit der direkten Annahme von Parameterverteilungen möglich, die Unsicherheit eines DNN zu schätzen [Jen19]. Beide genannten Methoden legen den Fokus vorwiegend auf das Abbilden der epistemischen Unsicherheit.

Deep Ensembles Eine weitere Methode zur Vorhersage von Unsicherheiten stellen Deep Ensembles (DEs) dar, welche keinen Schwerpunkt auf eine der beiden Unsicherheitstypen legen. Kern der Methode ist zum einen mehrere Modelle für eine Stichprobe an Vorhersagen zu verwenden und zum anderen feindliches (engl. adversarial) Training nach Goodfellow et al. [Goo15] für höhere Robustheit zu nutzen [Lak17]. Über die Inferenz mittels der verschiedenen Modelle erhält man eine Ausgabeverteilung, um auf die Unsicherheit schließen zu können.

¹ Nach Goodfellow [Goo16] kann „Regularisierung“ als Überbegriff für Methoden im Kontext ML verstanden werden, welche die Generalisierung von datengetriebenen Verfahren verbessern sollen. Unter Umständen kann dabei jedoch die Fehlerfunktion l während des Trainings negativ beeinflusst werden.

Test Time Augmentation Ein weiterer Ansatz, genannt Test Time Augmentation (TTA), adressiert aus der theoretischen Betrachtung die aleatorische Unsicherheit [Wan19]. Durch Augmentierung¹ der Eingangsdaten werden verschiedene Vorhersagen generiert. Über die Rücktransformation der Ausgaben² kann ein Konsistenzvergleich durchgeführt werden, um die Unsicherheit vorherzusagen.

Bezüglich der Metriken zur Unsicherheitsquantifizierung lässt sich im Falle der Segmentierung von Bildern zwischen pixelweiser und bildweiser Unsicherheit differenzieren, die je nach Untersuchungsziel verwendet werden. Monte Carlo Sample Variance, Predictive Entropy und Mutual Information sind geeignete Metriken für eine Quantifizierung auf Pixelebene [Nai20, Smi18], wobei diese insbesondere zur Visualisierung von unsicheren Regionen der Vorhersagen im Bild dienen können. Hingegen sind inverted Dice Agreement in Sample (iDAiS) [Roy19] und inverted Aggregated Jaccard Index Agreement in Sample (iAAiS) Metriken zur Berechnung der Unsicherheit bezogen auf das Gesamtbild. Beispielsweise können damit die besonders unsicher vorhergesagten Bilder eines Datensatzes bestimmt werden. Details zu den Definitionen der Metriken finden sich in Anhang D.

2.2 Annotation von Bilddatensätzen

Generell ist das Ziel von Methoden im Bereich der Annotation, einen Prozess bereitzustellen, der eine maximale Annotationsqualität gewährleistet und gleichzeitig den Aufwand für den Nutzer auf ein Minimum reduziert. Der Entwicklungsstand hinsichtlich der Annotation von Bilddatensätzen wird daher

¹ Nach Goodfellow et al. [Goo16] bezeichnet die Augmentierung eine Technik, um über Veränderungen der Daten (z. B. das Spiegeln oder Drehen der Bilder) den Umfang der Trainingsdaten künstlich zu erhöhen. Das DNN soll damit invariant gegenüber den Veränderungen werden und damit besser generalisieren.

² Unter Berücksichtigung der Augmentierungen, wie z. B. Drehung oder Spiegelung, können über deren Invertierung die Vorhersagen bezüglich des identischen Referenzsystems, also der Darstellung im Ursprungsbild, berechnet werden, um diese korrekt vergleichen zu können.

zweiteilig dargestellt. Zum einen soll in Abschnitt 2.2.1 detailliert auf verschiedene Methoden zur Effizienzsteigerung, also der Reduktion des Aufwandes bei gleichbleibender Güte, um zu einem annotierten Datensatz zu kommen, eingegangen werden. Zum anderen beschäftigt sich Abschnitt 2.2.2 mit der Variabilität von Experten während des Annotationsprozesses. Da Werkzeuge zur Annotation eng mit entsprechenden Methoden verwoben sind, werden auch die verfügbaren Softwarelösungen diskutiert.

2.2.1 Methoden zur Effizienzsteigerung

Als häufiges Argument gegen den Einsatz von DL wird die Notwendigkeit einer Annotation von Datensätzen genannt, die für SL (vgl. Abschnitt 2.1.2) erforderlich ist [Doe19, Mah19]. Die Annotation ist in der Regel ein sehr zeitintensiver Prozess, der insbesondere im Falle biomedizinischer Daten Expertise von Fachleuten benötigt [Chi20].

Die Art der Annotation hängt von der jeweiligen Bildverarbeitungsaufgabe ab. Nachfolgend sind exemplarische Bildverarbeitungsaufgaben und die notwendige Annotation aufgeführt, die vom jeweiligen Experten gemacht werden müssen [Kir19, Lit17]:

- **Klassifikation** - Festlegung einer Klasse pro Gesamtbild
- **Instansegmentierung** - pixelweise Markierung einzelner Segmente
- **Seed Detektion** - Markierung Objektmittelpunkte
- **Semantische Segmentierung** - pixelweise Markierung der Flächen bezüglich deren Klassen
- **Objekterkennung** - Umrandung der Objekte inklusive Klassenzuweisung pro Objekt
- **Panoptische Segmentierung** - pixelweise Markierung einzelner Segmente und Festlegung deren Klassenzugehörigkeit.

Als Basistools zur Annotation von Bildern können beispielsweise das Image-LabelingTool [Bar19], das Pixel Annotation Tool [Bré17], LabelMe [Wad16]

oder die Basisversion von Fiji/ImageJ [Sch12] genannt werden, in denen der Standardablauf des Annotierens ohne Unterstützung umgesetzt wird.

Die Idee von Deep Active Learning (DAL) [Ren21] ist, iterativ die Bilder mit dem höchsten Informationsgehalt zu bestimmen, um diese priorisiert zu annotieren. Details dazu finden sich in Abschnitt 2.4.1. Allerdings wird diese Methodik von gängigen nicht-kommerziellen Annotationswerkzeugen bisher nicht unterstützt [Bar19, Bré17, Sch12, Sek20, Str21b, Wad16]. Nur die kommerziellen Lösungen `hasty.ai` [Rou22] und `Labelbox` [Sha21] bieten eine Möglichkeit zur Nutzung von DAL. Im Falle von `hasty.ai` wird die Funktion jedoch nur in der beta-Version angeboten. Zudem sind bei `hasty.ai` lediglich Heuristiken auf Basis der quantifizierten Unsicherheit verfügbar.

Eine weitere Form zur Assistenz beim Annotieren kann ein Pre-Processing des Bildes sein. Ziel ist es dabei, Nutzern das Annotieren durch eine bessere Darstellung des Bildes zu erleichtern. Z. B. bietet die Software `BeadNet` [Sch20b, Sch23a] eine Möglichkeit zur Veränderung der Bildauflösung, um insbesondere kleinere Objekte besser annotieren zu können. Ferner hält Fiji/ImageJ [Sch12] Optionen zur Kontrastanpassung oder zum Entfernen von Rauschen bereit. Zudem unterstützt die Software `CVAT` [Sek20] die Nutzung von Bildverarbeitungsoperationen der Bibliothek `OpenCV` [Bra00].

Zur Steigerung der Effizienz im Rahmen der Bildannotation kann ferner Pre-Annotation genannt werden. Die Idee dahinter ist, dass häufig bereits eine Heuristik zur Annotation existiert. Nutzer müssen in diesem Fall nicht von null beginnen, sondern sind lediglich für die Korrektur von Annotationen verantwortlich. Diese Idee ist weit verbreitet und es findet sich beispielsweise in `Cellpose` [Str21b], `hasty.ai` [Rou22], `CVAT` [Sek20], im Fiji/ImageJ Plugin `AnnotatorJ` [Hol20] die Möglichkeit zur Integration von DNNs als Pre-Annotation. Außerdem bietet `ilastik` [Ber19] die Möglichkeit zur halbautomatischen Bildsegmentierung, also lediglich eine notwendige Korrektur von Segmentvorhersagen, über eine Verknüpfung von Kantenerkennung mit der Anwendung des Watershed-Algorithmus [Beu79]. Zudem kombiniert die Arbeit in [Eng21] das Otsu-Schwelwertverfahren [Ots79] mit dem Watershed-Algorithmus, um Annotationen automatisiert zu erzeugen.

In der Regel wird die Annotation mit einer Computer-Maus als Eingabegerät durchgeführt. Forlines et al. [For07] vergleichen Eingaben unter der Verwendung einer Computer-Maus zu der Eingabe über Finger in Kombination mit einem berührungsempfindlichen Bildschirm. Die Experimente der Autoren zeigen, dass bei bimanuellen Aufgaben, wie dem Aufsetzen einer Maske auf ein Objekt, Touchscreens von Vorteil sind.

Es fehlt jedoch ein Konzept, das die einzelnen Methoden zur Effizienzsteigerung im Rahmen der Annotation von Bilddatensätzen kombiniert, damit Anwender von den jeweiligen Vorteilen gleichzeitig profitieren können.

2.2.2 Variabilität von Experten

Da das Annotieren ein Prozess ist, in dem Menschen mit unterschiedlichen Auffassungen interagieren und die Zeit von Experten meist knapp ist, stellen inkonsistente Annotationen durch die Variabilität von Experten eine Herausforderung dar [Kar20, Zha18]. Der Trend zu „Crowdsourcing“, also das verteilte Annotieren von Datensätzen mittels unterschiedlicher Annotatoren, über Plattformen wie CloudFactory, Prolific Academic, Amazon Mechanical Turk oder CrowdFlower¹ macht die Qualität der Annotationen schwer kontrollierbar [Zha18]. Abweichend zum klassischen Verständnis in Form von stochastischen Schwankungen aus dem Feld der Mess- und Regelungstechnik soll der Terminus „Rauschen“ im Weiteren als Sammelbegriff für Annotationsfehler, Verzerrungen oder Ähnliches im Kontext von Annotationen genutzt werden.

Nach Karimi et al. [Kar20] kann zwischen

- Variabilität innerhalb der Annotation eines einzelnen Experten (**Intra-Experten-Variabilität**) und
- Variabilität zwischen verschiedenen Experten (**Inter-Experten-Variabilität**) unterschieden werden.

¹ Erreichbar unter <https://info.cloudfactory.com>, <https://www.prolific.co/>, <https://www.mturk.com/> und <https://www.crowdfunder.com/>.

Im Falle eines Aufteilens der Annotation eines Datensatzes besteht die Gefahr, dass verschiedene Annotationsstile in den jeweiligen Teildatensätzen vorzufinden sind. Entscheidungsgrenzen können zwischen Experten variieren. Beispielsweise annotiert ein Experte konsequent lediglich die Kontur eines Objekts, aber ein anderer Experte markiert konträr dazu das Gesamtobjekt. Unterschiedliche Hardware, z. B. variierende Monitoreinstellungen, verschiedene Qualifikationen von Experten, Ermüdung oder Zeitdruck können weitere Gründe für Variabilität bei Annotationen sein. Zudem kann es zu verrauschten Annotationen kommen, wenn Nutzer Pre-Annotation verwenden. Falls der Nutzer in allen Fällen die Vorschläge der Heuristik übernimmt, ohne diese zu kontrollieren, kann die Qualität der Annotationen leiden [Kov22]. Rädtsch et al. [Räd23] nennen zudem, dass unzureichende Anweisungen für Experten zu Variabilität innerhalb der Annotationen führen können.

Verschiedene Arbeiten stellen heraus, dass sich verrauschte Annotationen negativ auf die Güte von DNNs auswirken können, wie z. B. bei der Klassifikation [Nor21a], bei der Segmentierung [Yu20] oder dem Tracking von Objekten [Löf21, Löf23]. Die Interpretation von ähnlichen Mustern, wie z. B. ein helles Objekt vor einem dunklen Hintergrund, gepaart mit inkonsistenten Annotationen, führt zu unterschiedlichen Optimierungsschritten im Rahmen des DNN-Trainings. Im Allgemeinen sind Metriken im Falle verrauschter Annotationen nicht korrekt, da eine falsche Referenz angenommen wird. Damit verknüpft können verrauschte Annotationen eine Auswahl suboptimaler DL-Modelle hervorrufen, da die Ergebnisse von Metriken bei der Validierung mit inkonsistent annotierten Datensätzen nicht zuverlässig sind [Nor21a]. Im Speziellen argumentieren die Autoren in [Kar20, Rol17], dass bei Datensätzen mit geringem Umfang, wie es in der Biomedizin oftmals vorkommt, Rauschen einen großen Einfluss hat.

Allgemein existieren zwei Wege, wie mit verrauschten Annotationen umgegangen werden kann¹:

¹ In [Kar20, Son20, Taj20] finden sich ausführliche Zusammenfassungen zu verrauschten Annotationen im Kontext von DL.

- Es können die **Auswirkungen verrauschter Annotationen** während des Trainings eines DNN **reduziert werden**, ohne diese aus dem Datensatz explizit zu entfernen. Als Methoden ist eine Anpassung von Netzarchitekturen [Suk15], Optimierungsfunktionen [Gho17, Liu20], der Gewichtung von Trainingsbeispielen [Pat17], der Datenrepräsentation [Sch20a] oder des Trainingsverfahrens [Guo18, Ise21b, Mir19] zu nennen.
- **Verrauschte Annotationen** können direkt **erkannt und korrigiert** werden.

Da aufgrund der Methodenvielfalt eine Eingrenzung notwendig ist, fokussiert sich diese Arbeit im Weiteren auf die direkte Detektion von problematischen Annotationen, um diese im Anschluss korrigieren zu können.

Für Klassifikationsaufgaben schlagen die Autoren von CleanNet [Lee18] vor, pro Bild einen Merkmalsvektor zu extrahieren und mit einem Merkmalsvektor zu vergleichen, der für eine bestimmte Klasse repräsentativ ist. Verrauschte Annotationen werden erkannt, wenn die Merkmalsvektoren große Abweichungen von dem Merkmalsvektor einer repräsentativen Klasse aufweisen. Einen ähnlichen Ansatz stellt die Arbeit in [Zhu22] vor, allerdings können beide Methoden nicht auf Segmentierungsaufgaben übertragen werden, da im Gegensatz zu den Klassen ein direkter Vergleich der Segmentierungsmasken nicht möglich ist.

Im Falle von Rank Pruning [Nor17] wird die DNN-Ausgabe für die Erkennung von verrauschten Annotationen berücksichtigt, wobei sich die Arbeit auf Klassifikationen fokussiert. Dabei wird angenommen, dass die Netzwerkausgaben nach Anwendung von Softmax- oder Sigmoid-Funktion als Wahrscheinlichkeitsverteilung interpretiert werden können. Falls die Vorhersage eine eindeutige Ausgabe ausweist und es eine Diskrepanz zwischen Annotation bzw. Vorhersage gibt, liegt eine vermeintlich falsche Annotation vor.

Das Confident Learning Framework stellt eine Generalisierung von Rank Pruning für verschiedene Arten von Datensätzen/Problemstellungen dar und beinhaltet ein zugehöriges Softwarepaket [Nor21b]. Wie in Abschnitt 2.1.3 erwähnt, zeigen Untersuchungen, dass sich Standard-DNNs zu sicher sind und

daher die Ausgabe im Allgemeinen nicht mit einer Wahrscheinlichkeitsverteilung gleichgesetzt werden kann (siehe auch „fehlende Kalibrierung“ in Abschnitt 2.1.3).

Die Arbeit von [Köh19] greift das Problem des Gleichsetzens von Wahrscheinlichkeiten mit DNN-Ausgaben im Falle von Klassifikationsproblemen auf und verwendet daher unsicherheitsbewusste DNNs (vgl. Abschnitt 2.1.3). Die dort genannten Methoden sind jedoch nicht direkt bezüglich Segmentierungsaufgaben anwendbar, da die Ausgabe in diesem Fall, konträr zur Klassifikation, pixelweise vorliegt.

Eine Wiederholung der Annotation durch mehrere Experten kann helfen, um Inter-Experten-Variabilität zu erkennen. Eine Methode, die verschiedene Annotatoren berücksichtigt, ist in [Tan19] dargestellt, wobei auch hier lediglich Klassifikationsprobleme beleuchtet werden. Die Idee ist, Genauigkeiten der Annotatoren untereinander in Form von Konfusionsmatrizen zu schätzen.

Für Segmentierung schlagen die Autoren in [Koh18] als Modellarchitektur ein Probabilistic U-Net vor, um mit verrauschten Annotationen unter Verwendung eines Variational Autoencoder (AE) umzugehen. Die vorgeschlagene Architektur lernt Verteilungen von Segmentierungsmasken zu erzeugen, wie es beispielsweise bei mehreren Experten der Fall ist, anstelle deterministisch eine Segmentierungsmaske vorherzusagen.

Das typische Szenario im Kontext der Annotation ist die Arbeitsteilung zwischen Annotatoren. Der Nachteil bei Methoden, die mehrere Annotationen pro Datenelement verwenden, ist ein erhöhter Annotationsaufwand, wodurch dieses Vorgehen bei vielen Anwendungen aus Zeit- oder Kostengründen nicht möglich ist.

In [Hol20, Ulm17] werden Pipelines zum Erzeugen von annotierten Datensätzen verwendet, um Annotationen automatisiert und ohne Aufwand für Experten zu generieren. Es erfolgt dabei eine manuelle Inspektion für Fälle, in denen korrekt annotierte Daten für Benchmarks oder Evaluierungen benötigt werden.

Hinsichtlich der Umsetzung von Annotations-Inspektionen können CVAT [Sek20]/Labelling [Lin15] für manuelles Markieren von problematischen Bildern genannt werden. Ferner ermöglicht das kommerzielle Softwarepaket *hasty.ai* [Rou22] das Anzeigen von Bildern mit vermeintlich falschen Vorhersagen verglichen zu den annotierten Referenzdaten, um problematische Annotationen aufzuzeigen. Diese Funktion wird sowohl für die Klassifikation als auch für Segmentierungen bereitgestellt. Dabei handelt es sich jedoch nur um eine Beta-Funktion, die sich noch in der Entwicklung befindet. Zudem fehlen ein Open-Source-Algorithmus und eine detaillierte Beschreibung der Methode.

Es kann zusammengefasst werden, dass sich der Entwicklungsstand zur Detektion von verrauschten Annotationen auf Klassifikationsprobleme fokussiert und es an ausführlichen Betrachtungen zur Bildsegmentierung mangelt. Ferner fehlt eine Integration der Methoden in frei zugänglichen Softwaretools, um diese für Nutzer anwendbar zu machen.

2.3 Biomedizinische Hochdurchsatzbildverarbeitung

2.3.1 Überblick

Nach Jähne [Jäh12] ist die Bildverarbeitung mehrstufig und besteht generell aus der Bildaufnahme, Digitalisierung, Vorverarbeitung sowie verschiedenen Analyseschritten. Ziel ist es, aus Rohdaten zu einer interpretierten Szenenbeschreibung zu kommen [Des11]. Die Arbeit in [Mik08] stellt heraus, dass die automatisierte Bildverarbeitung von biomedizinischen Daten in den vergangenen Jahren immens zugenommen hat. Insbesondere steigt die Bedeutung von Hochdurchsatzexperimenten (HDEs) in Bereichen wie Chemie, Pharmazie, Biologie, Biotechnologie oder Medizin [Als16, Ben20, Gal15, Gar22, Jaa22,

Rei17]. Ein Grund hierfür ist das Ermöglichen von umfangreichen Untersuchungen in kurzer Zeit kombiniert mit erhöhter Reproduzierbarkeit von Experimenten [An10, Szy12]. Ferner reduziert die Miniaturisierung von Experimenten die Betriebskosten und verbessert gleichzeitig die räumlich-zeitliche Kontrolle [Vya10].

Gitterstrukturen sind dabei ein gängiger Aufbau bei biomedizinischen HDEs. Als Beispiele können Mikrotiterplatten [Aul20], Microwell-mesh [Mos18], Agarose Microwells [Zha19], Microwell Array Chips [Ban19] oder Droplet Microarrays (DMAs) [Pop16] genannt werden (vgl. Abbildung 2.2).

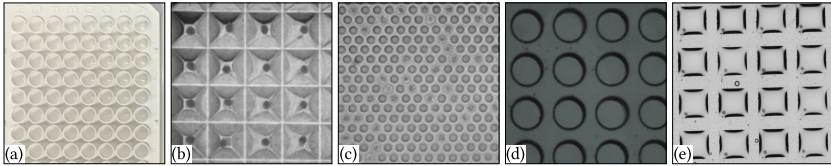


Abbildung 2.2: Gittertypen. Es erfolgt eine Darstellung [Sch23b] von Ausschnitten der Gittertypen (a) Mikrotiterplatte [Aul20], (b) Microwell-mesh [Mos18], (c) Agarose Microwells [Zha19], (d) Microwell Array Chips [Ban19] und (e) DMA [Pop16].

Verschiedene bildgebende Verfahren, wie z. B. Mikroskopie oder die Verwendung einer Kamera [Dös16], werden häufig zur Ermittlung der Resultate von Experimenten oder für Diagnosezwecke verwendet. Allerdings ist eine manuelle Analyse der gitterförmigen HDEs zeitaufwändig, mühsam, dementsprechend kostenintensiv und kritisch in Bezug auf die Reproduzierbarkeit aufgrund der subjektiven Beurteilung [Kli20].

2.3.2 Fragestellungen und Methoden

Deserno [Des11] stuft die biomedizinische Bildverarbeitung als komplexe Aufgabe ein, da i) die Bilder oftmals sehr heterogen sind, ii) Objekte sich in Bildern schwierig abgrenzen lassen und iii) bei Algorithmen hohe Anforderung in puncto Robustheit vorliegen. Zudem ist die Generierung eines annotierten Datensatzes durch Experten herausfordernd, da hier spezifisches Fachwissen notwendig ist [Doe19, Mah19].

Die automatisierte Erkennung von Gitterstrukturen in HDEs stellt ein offenes Problem dar. Folgende Hürden, die eine automatisierte Gittererkennung erschweren, können genannt werden: i) individuelle plattformspezifische Charakteristiken (unterschiedliche Formen, Größen und Abstände der Elemente), ii) Variationen in der Bildaufnahme (z. B. unterschiedliche Beleuchtungsbedingungen bzw. Rotationen und Translationen des Gitters), iii) gegebene Artefakte in den Bildern und iv) vorliegende Sub-Gitterstrukturen, die zu unregelmäßigen Abständen zwischen Elementen führen.

Mittels der frei zugänglichen Software ImageJ/Fiji [Sch12] können Elemente lediglich manuell ausgeschnitten werden. Softwarepakete wie PlantCV [Geh17] oder CellProfiler [Jon08, Sti21] ermöglichen die Definition eines Gitters über die Angabe von Zeilen- und Spaltenanzahl. Jedoch ist weder die automatische Detektion der Gitterstruktur noch die Auswertung rotierter Gitter möglich. In [Lim16] erfolgt die Gitterdefinition über einen Ausrichtungsprozess von auszuwertenden Bildern in Kombination mit einem manuell festgelegten Referenzgitter. Die Arbeit von Seifermann et al. [Sei23] stellt eine Software zur halbautomatischen Gittererkennung vor. Die Methodik dort setzt jedoch ebenfalls ein nicht gedrehtes Gitter voraus bzw. sieht keine Erweiterung der Auswertung pro Element für andere Anwendungsfälle vor. Klimaj et al. [Kli20] schlagen eine mehrstufige Methode zur Schätzung der Gitterelemente vor. Zu Beginn muss allerdings die ungefähre Position des jeweiligen Gitterelements manuell ausgewählt werden. Anschließend wird ein Algorithmus zur Objekterkennung verwendet, um die genaue Segmentform eines Elements zu bestimmen. Der Ansatz wird Nutzern im Rahmen der EVOS-Plattform öffentlich zur Verfügung gestellt.

Ein Auszug der heterogenen Daten, die es zu analysieren gilt, findet sich in Abbildung 2.3. Einerseits untermauern die dargestellten Beispiele die These von Deserno [Des11] bezüglich Komplexität und Heterogenität im Bereich der biomedizinischen HD-Bildverarbeitung. Andererseits lässt sich feststellen, dass Aufgaben zur i) semantischen Segmentierung von Bildern (a,b), ii) Detektion von Instanzen (d,e,f,g) oder iii) Vorhersage einer Kennzahl pro Element (c,h) in Praxisprojekten vermehrt auftreten. Nach Reischl [Rei17] ist zudem oftmals

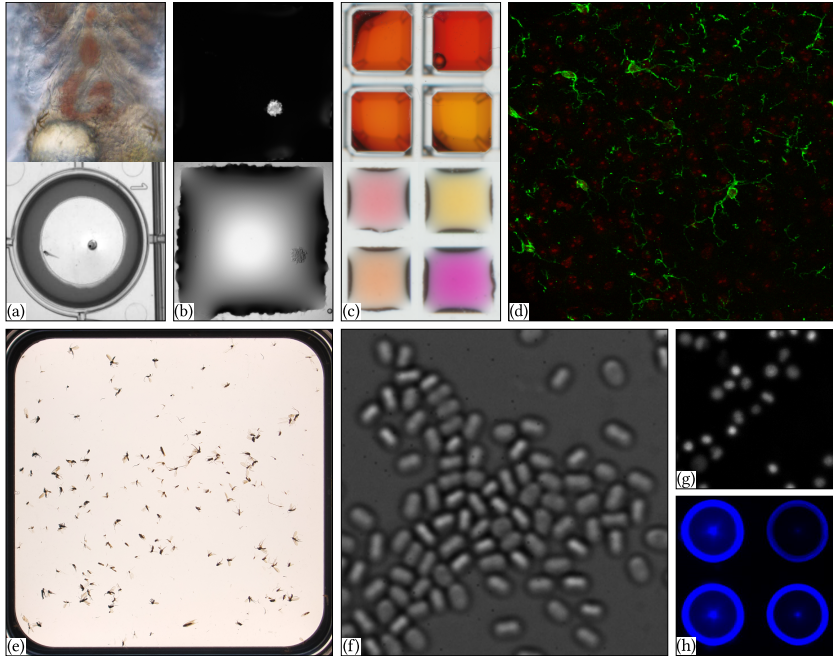


Abbildung 2.3: Fragestellungen im Bereich der Analyse von Elementen. Beispiele sind (a) die Segmentierung von Zebrafärblingen (unten: Segmentierung des Fisches, oben: Erkennung von Organen) [Sch19, Sch22g, Wan21b], (b) die Analyse von Sphäroiden [Pop19, Sch22c] in Fluoreszenz (oben)-/Hellfeldaufnahmen (unten), (c) Farbanalysen, (d) die Erkennung von Immuneffektorzellen Mikroglia, (e) das Zählen von Insekten [Wüh22], (f) eine Analyse des Keimungsprozesses von Bakteriensporen [Sch23d], (g) die Detektion von Zellkernen oder (h) Intensitätsanalysen von Fluoreszenzaufnahmen [Wie22].

im Falle der semantischen Segmentierung oder der Instanzsegmentierung eine weitere Beschreibung der Merkmale einzelner Segmente oder Instanzen (z. B. deren Form oder Flächeninhalt) für Anwender von Interesse.

Traditionelle Segmentierungsansätze berücksichtigen im einfachsten Fall lediglich die Pixelwerte. Hierbei können als Vertreter Schwellwertverfahren genannt werden [Jäh12], wobei Schwellwerte entweder manuell durch Nutzer oder automatisch, z. B. über eine Analyse des Bildhistogramms, eingestellt

werden. Beispiele sind eine statistische perzentilbasierte Schwellwertermittlung oder das Otsu-Verfahren mit dem Ziel einer Maximierung von der Varianz zwischen Vordergrund- und Hintergrund-Klasse [Ots79]. Weitere Vertreter sind das k-means Clusterverfahren [Mac67] oder der parameterlose Clusteralgorithmus Mean Shift [Che95], die versuchen, das Cluster „Segment“ zum Cluster „Hintergrund“ zu unterscheiden.

Wenn „Farbe“ das unterscheidende Merkmal ist, kann gegebenenfalls eine Farbraumtransformation in den Hue-Saturation-Value (zu Deutsch: Farbwert, Farbsättigung, Helligkeit) (HSV)-Farbraum hilfreich sein. Ferner lassen sich Segmentierungen auch basierend auf Kanten [Jäh12, Rei17] oder regionenbasiert mittels berechneter Seeds bestimmen. Gängige Ansätze zur Seed-Bestimmung sind Laplacian of Gaussian (LoG) Filter, deren Approximation Difference of Gaussian (DoG) für eine schnellere Berechnung oder die Methode Determinant of Hessian (DoH) [Lin98]. Aus den Seeds können beispielsweise über den Connected Component Algorithmus [Sam88], Region Growing Algorithmus [Ada94] oder das Watershed-Verfahren [Beu79] Segmente bestimmt werden. Der Ansatz TWANG ermittelt Seeds, die nachgelagert mit Otsu-Thresholding auf einer optimierten Bildrepräsentation zu Segmenten prozessiert werden [Ste14]. Ferner können über ein mathematisches Modell Objekte wie Kreise oder gerade Kanten vom Hintergrund mittels Hough-Transformation segmentiert werden [Hou60]. Bei der Verwendung von Template Matching [Bri01] hingegen wird versucht, ein vorgegebenes Objekt („Template“) in den jeweiligen Bildern wiederzufinden („Matching“). Wie im Ansatz von Laws [Law79] gezeigt, kann die Analyse von Texturen über die Berechnung von Co-Occurrence Matrizen [Che79] ebenfalls für die Segmentierung genutzt werden. Morphologische Operatoren lassen sich verwenden, um Segmente nachzubearbeiten [Jäh12].

Eine Weiterentwicklung im Bereich ML für Bildverarbeitungsaufgaben ist der Ansatz von Viola und Jones [Vio01]. Dort werden manuell Merkmale wie Haar-Features extrahiert, die mit lernbaren Klassifikatoren interpretiert werden sollen. Ferner stellen Convolutional Neural Networks (CNNs) [Kri12, LeC98] ein zentrales Element in gängigen DL-Ansätzen dar, da sich diese als überlegen gegenüber herkömmlicher Bildverarbeitung erwiesen haben. Die

CNN-Architektur besteht aus den zentralen Elementen: i) Faltungsschichten zur lokalen, ortsunabhängigen Mustererkennung in Bildern, ii) Pooling-Schichten für das Kondensieren von Informationen und iii) einer vollständig verbundenen Neuronenschicht (siehe Abschnitt 2.1.1) am Ende zur finalen Interpretation [Goo16, LeC15]. CNNs werden aktuell oftmals zur Analyse von Bildern herangezogen. LeCun et al. [LeC15] stellen in diesem Zusammenhang heraus, dass die Nutzung vieler Schichten in CNNs, Fortschritte in Software und Hardware (z. B. Weiterentwicklungen von Graphics Processing Units (GPUs)) und die Verfügbarkeit vieler Daten die wesentlichen Bestandteile des Erfolgs von CNNs sind.

Doerr et al. [Doe19] betonen in ihrer Arbeit, dass DL einen großen Einfluss auf die Forschungsgemeinschaft in Bereichen wie Mikroskopie oder Medizin hat. Ein CNN-Vertreter ist die Encoder-Decoder-Architektur U-Net [Ron15], welche den Stand der Technik für biomedizinische Segmentierungsprobleme darstellt (vgl. Abbildung E.1 im Anhang). Daraus sind entsprechende Weiterentwicklungen wie 3D U-Net [Çiç16] für 3D-Daten, nnU-Net [Ise21b] zur Selbstkonfiguration, U-Net++ [Zho18] mit angepassten Verbindungen zwischen Encoder und Decoder bzw. tiefer Überwachung zur Auswertung des Fehlers bei unterschiedlichen Modellgrößen entstanden. Die Residual Neural Network (ResNet) [He16] Architektur (siehe Abbildung E.2 im Anhang) nutzt Überbrückungsverbindungen, um verschwindende Gradienten beim Training zu vermeiden und wird oftmals für Klassifikationsprobleme genutzt. Wie in [Cha20b] gezeigt, gibt es Ansätze, die ein U-Net mit ResNet als Encoder statt des im Ursprungsartikel verwendeten Standard-Encoders nutzen [Ron15].

Scherr et al. [Sch20b] stellen BeadNet vor, wobei es sich um eine angepasste Version des U-Nets handelt, die zur Detektion von Seeds in Bildern genutzt

werden kann. In [Sch20a, Sch23a] wird zudem eine Version des U-Nets beschrieben, welche euklidische Distanzkarten von Instanzen bzw. deren Abstand zu Nachbarobjekten vorhersagt, um Instanzsegmentierungen durchzuführen. Dabei wird im Post-Processing das Watershed-Verfahren [Beu79] genutzt. Das Modell in Cellpose [Str21b] nutzt ein U-Net, das zum einen vorhersagt, ob ein Pixel Teil einer Instanz ist. Zum anderen werden die horizontalen und vertikalen Gradienten der Instanzen präzidiert. Die Gradienten lassen sich über Ableitung eines konstruierten Vektorfeldes bestimmen, wobei das Vektorfeld die Wärmediffusion einer Wärmequelle im Zentroid bis zum Rand der Instanz simuliert. Ferner kann die Architektur Mask-R-CNN [He17] für Instanzsegmentierung genutzt werden. Mask-R-CNN besteht aus einem Netz zur Merkmalsextraktion (z. B. ResNet), einem Region Proposal Network zum Generieren von Objektvorschlägen und einem Head-Netz zum Interpretieren der Objekte bzw. zum Anpassen von deren Größen. Eine Weiterentwicklung ist z. B. die Detectron2-Architektur [Wu19], welche durch unterschiedliche Head-Netze, also NNs zur weiteren Interpretation der Objektvorschläge, zudem eine semantische Segmentierung oder die Vorhersage von Keypoints ermöglicht. Neuentwicklungen im Bereich DL sind Vision Transformers, die den Attention Mechanismus [Vas17] auf Bildausschnitte anwenden [Dos21]. Ursprünglich ist der Attention Mechanismus für die Sprachverarbeitung entwickelt worden, um über flexible Gewichtungen Abhängigkeiten in Sequenzen zu modellieren. Allerdings argumentieren Khan et al. [Kha22], dass Transformer ihre Limitierung insbesondere bei wenig verfügbaren annotierten Daten zeigen.

Zusammenfassend lässt sich sagen, dass aktuell vielfältige Entwicklungen hinsichtlich DNN-Architekturen stattfinden. Trotzdem bleibt die Schwierigkeit, dass die Bilddaten für das überwachte Training der DNNs annotiert werden müssen.

2.3.3 Anwendbarkeit der Bildverarbeitungsalgorithmen

Die Anwendbarkeit von Algorithmen zur automatisierten Bildverarbeitung kann aus unterschiedlichen Blickwinkeln betrachtet werden.

Zum einen steigt der Umfang an Daten an, die gleichzeitig dynamisch und unstrukturiert sind. Es sind Konzepte notwendig, um Daten nach dem Findable Accessible Interoperable Reusable (FAIR)-Prinzip auffindbar, zugreifbar, interoperabel und wiederverwendbar zu machen [Wil16].

Zum anderen argumentieren die Autoren in [Góm21b, Mik08, Ouy19, Rei06], dass viele Methoden zur biomedizinischen Bildverarbeitung für Nutzer wie Biologen, Ärzte oder Chemiker nicht anwendbar sind, da beispielsweise nur programmierte Skripte statt Graphical User Interfaces (GUIs) vorliegen, Anforderungen seitens Software erfüllt werden müssen oder spezielle Hardware wie GPUs notwendig ist.

Das Programmpaket ImageJ/Fiji [Sch12] kann für klassische Bildverarbeitungsoperationen wie das Bestimmen von Intensitätsmittelwerten oder Kanten in Bildern genutzt werden. Ferner existiert der Plugin U-Net ImageJ [Fal19b], welcher die U-Net-Architektur als Erweiterung bereitstellt. Jedoch ist die Migration von anderen Standard-DL-Pipelines schwierig, weil diese meist nur in python verfügbar sind und das Plugin hingegen in Java entwickelt ist. Die Software DeepImageJ [Góm21b] ermöglicht ebenfalls die Integration von DNNs, ist allerdings kompatibel mit der Modellbibliothek Bioimage Model Zoo [Ouy22], wodurch zusätzliche Modelle unterstützt werden. Ein benutzerdefiniertes Training ist nicht mittels DeepImageJ möglich. Zum anderen stellen die Tools PlantCV [Geh17] und CellProfiler [Car06] Standardbildverarbeitungsoperatoren bereit. Im Falle von PlantCV [Geh17] fehlt eine GUI. Die Autoren in [Kli20] stellen ihre Methode lediglich für die Software von EVOS-Bildgebungssystemen bereit. ImJoy [Ouy19] ist speziell für die Bereitstellung von Webanwendungen im Kontext DL-basierter Bildverarbeitung entwickelt. Die Software HTS-Corrector [Mak06] stellt Methoden zur statistischen Analyse von HDEs bereit, kann jedoch nicht direkt mit Bildern umgehen, sondern setzt bereits extrahierte Merkmale voraus. In [Cha06] stellen die Autoren einen Ansatz für das Datenmanagement und die Visualisierung von Daten aus HDEs vor, in welchem die Software FreeFlowDB [Sin06] verwendet wird.

Des Weiteren können die Plattformen Large Scale Data Facility (LSDF) [Jej12] des Karlsruher Instituts für Technologie (KIT) und Dataverse [Kin07] der

Universität Harvard¹ als Datenaustauschsysteme für Forschung im Terabyte-Bereich genannt werden. Diese Plattformen sind insbesondere für Forscher in interdisziplinären Projekten notwendig, in welchen große Datenmengen wie Bilddaten aus HDEs organisiert, geteilt oder archiviert werden müssen.

Zusammenfassend lässt sich festhalten, dass es an einem umfassenden Gesamtsoftwarekonzept fehlt, das speziell auf HDEs zugeschnitten ist, generisch ist und sich beliebig erweitern lässt, Datenmanagement berücksichtigt sowie von Nutzern mittels GUI ohne spezielle Programmierkenntnisse bedienbar ist.

2.4 Effizientes Lernen für Bildverarbeitung

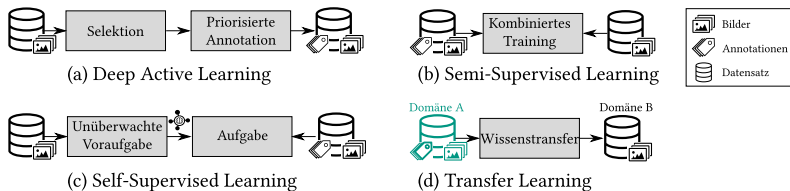


Abbildung 2.4: Übersicht effizientes Lernen. Die Grundprinzipien von (a) DAL, (b) Semi-Supervised Learning (SemSL), (c) Self-Supervised Learning (SSL) und (d) Transfer Learning (TL) sind schematisch illustriert.

Da das Annotieren mit großem Aufwand verbunden ist [Chi20, Mah19], gibt es verschiedene Ansätze, um die Parameter von DNNs effizienter als das klassische überwachte Lernen zu bestimmen. Generell kann zwischen den folgenden Obergruppen (siehe schematische Darstellung in Abbildung 2.4) unterschieden werden [Pen21]:

- (a) **DAL** - der Selektion von Bildern mit hohem Informationsgehalt zur priorisierten Annotation (vgl. Abschnitt 2.4.1),

¹ <https://support.dataverse.harvard.edu/>

- (b) **SemSL** - einer Kombination aus überwachtem/unüberwachtem Lernen (siehe Abschnitt 2.4.3),
- (c) **SSL** - das vorgeschaltete Lernen einer Voraufgabe für ein verbessertes Durchführen der eigentlichen Aufgabe (vgl. Abschnitt 2.4.2) und
- (d) **TL** - ein Wissenstransfer einer gelösten Aufgabe auf das zu lösende Problem (vgl. Abschnitt 2.4.4).

Zudem existiert Weakly Supervised Learning, das versucht, aus Annotationen mit geringem Informationsgehalt Probleme höherer Komplexität zu lösen, wie beispielsweise aus Objektboxen, Punkten oder skizzierten Linien Segmentierungen vorherzusagen. Dies ist jedoch außerhalb des Umfangs der Arbeit.

Bezogen auf alle Methodenübergruppen stellt die Arbeit von Peng und Wang [Pen21] einen Überblick diverser annotations-effizienter Lernverfahren im Kontext der Segmentierung medizinischer Daten bereit. Allerdings fehlt es an einem Schema inklusive verschiedener Metriken zur objektiven Bewertung der Methoden. Die Arbeit in [Cha20a] vergleicht DAL, SSL und SemSL für einen Echtwelt-Klassifikationsdatensatz und kommt zu dem Resultat, dass für den gezeigten Anwendungsfall SSL und SemSL überlegen zu Ansätzen im Bereich DAL sind. Jedoch bleibt die Frage offen, wie sich die Methoden im direkten Vergleich bezüglich biomedizinischen Segmentierungsaufgaben verhalten.

2.4.1 Deep Active Learning

Die Autoren in [Ren21, Set10] führen Active Learning (AL) als Herangehensweise mit dem Ziel ein, die Leistungsfähigkeit von ML-Algorithmen zu steigern, indem während der Lernphase Daten mittels einer entwickelten Strategie (z. B. die Suche nach Datenpunkten von hohem Informationsgehalt) gezielt ausgesucht werden. Der Nutzer wird im Trainingsprozess befragt und soll Daten mit hohem Informationsgehalt priorisiert annotieren.

Es lässt sich zwischen den Kategorien i) Membership Query Synthesis, ii) Stream-Based Selective Sampling und iii) Pool-Based Sampling differenzieren [Set10]. Im Falle von Membership Query Synthesis werden aus nicht

annotierten Daten synthetisch neue Bilder erzeugt, welche im Nachgang annotiert werden sollen. Beispielsweise wird über Interpolation ein Datenpunkt erzeugt, der sich in der Nähe von bisherigen Entscheidungsgrenzen befindet [Wan15]. In der Kategorie Stream-Based Selective Sampling erfolgt die Verarbeitung eines Stroms aus neuen, nicht annotierten Daten. Aufgabe ist die direkte Entscheidung, ob ein Beispiel vom Nutzer zu annotieren oder nicht für die Erweiterung des Datensatzes relevant ist. Im Gegensatz dazu ist die Grundidee im Falle des Pool-Based Samplings, nicht annotierte Daten in einem Datensatz zusammenzufassen und mittels AL die vielversprechendsten Dateninstanzen für die Annotation durch den Nutzer auszuwählen.

Der Fokus liegt im Weiteren auf der zuletzt genannten Kategorie. Grund gegen eine weitere Betrachtung von Membership Query Synthesis ist die Komplexität der Bilddaten im Bereich Biomedizin, die durch Menschen annotiert werden müssen. Synthetische Methoden stellen sich hier als anspruchsvoll heraus. Zudem zielt die vorliegende Arbeit nicht auf Anwendungen von direkt zu bearbeitenden Datenströmen ab, weshalb das Stream-Based Selective Sampling nicht betrachtet werden soll.

AL im Kontext klassischer ML-Verfahren unterscheidet sich zu DAL, da i) hohe Konfidenzwerte am Netzausgang nicht unbedingt mit geringen Modellunsicherheiten gleichzusetzen sind [Abd21, Cor19, Gal16, Ken17], ii) mehr Rechenzeit für das Training beansprucht wird [Mah19] und iii) das Hinzufügen eines einzelnen Trainingsbeispiels den Informationsgehalt der Daten nur wenig erhöht bzw. sogar zu Überanpassung führen kann [Ren21]. Folglich werden dem Datensatz im Falle von DAL pro Iteration mehrere Instanzen hinzugefügt, bevor ein neuer Trainingsdurchlauf gestartet wird.

Im Allgemeinen bilden Verfahren, die basierend auf der Modellunsicherheit nicht annotierte Daten auswählen, wie in [Gal17, Sai21] gezeigt, eine Gruppierung innerhalb des DAL. Eine weitere Kategorie formen Verfahren, die auf Heterogenität im Datensatz bedacht sind. Bekannter Vertreter ist der Core Set Ansatz [Sen18], der für einen Datenpunkt mittels eines Modells Merkmale generiert, um dessen Ähnlichkeit zu anderen Elementen im Datensatz zu quantifizieren. Iterativ wird dann der Datenpunkt gesucht, der maximal unterschiedlich zu dem ähnlichsten bekannten Element (minimaler Abstand) des

Datensatzes ist. Zudem existieren Ansätze, die Unsicherheit und Heterogenität kombinieren [Ash20] oder ein getrenntes Modell zur Merkmalsextraktion im Sinne von Core Sets nutzen [Col20], das jedoch nur einmalig trainiert werden muss. Des Weiteren können über die Approximation der Zielfunktion Elemente mit hohem Informationsgehalt identifiziert werden [Yoo19].

2.4.2 Self-Supervised Learning

Im Gegensatz zu SL ist beim Unsupervised Learning (UL) das Ziel, nicht annotierte Daten zu analysieren. Bekannte Aufgabentypen in diesem Bereich sind Clusterverfahren oder die Dimensionalitätsreduktion [Jin21, Moh12]. Eine Untergruppe des UL bildet SSL, welches ebenfalls auf nicht annotierte Daten im Lernverfahren zurückgreift [Jin21]. Hierbei besteht der Unterschied darin, dass Annotationen automatisiert über spezielle Ansätze für ein Training generiert werden, das Endziel des Lernprozesses jedoch das Lösen einer Aufgabe ist, die klassisch im überwachten Lernen angesiedelt ist. SSL zielt darauf ab, über das Lernen von sinnvollen Repräsentationen¹ in einem ersten Schritt (=Pre-Text-Task) bessere Ergebnisse für die eigentliche Aufgabe (=Downstream-Task) zu erreichen. Der Downstream-Task wird in der Regel klassisch überwacht im Sinne von SL trainiert (vgl. Abschnitt 2.1.2). Damit ersetzen die Ergebnisse aus SSL eine zufällige Initialisierung [He15]. Allgemein gibt es generative [Ala14] Methoden, die auf Bildrekonstruktionen in der Zielfunktion zurückgreifen, oder unterscheidende Methoden mit Zielfunktionen, die direkt Repräsentationen betrachten, bzw. Mischformen [Don19]. Die Arbeit legt im Weiteren den Fokus auf unterscheidende Methoden, da diese Kategorie den Entwicklungsstand dominiert.

Gängiger Ansatz im Bereich der unterscheidenden Methoden ist die Verwendung von Contrastive Learning (CL), bei dem Pseudo-Annotationen über Daten-Augmentierung generiert werden, indem zunächst zwei ähnliche Varianten eines Bildes erzeugt werden. Die Idee der Zielfunktion im Lernprozess

¹ Der Terminus „sinnvolle Repräsentationen“ charakterisiert nach Ganguli et al. [Gan22] das Lernen von Bildmerkmalen, mit denen ein Bild hinsichtlich verschiedener Aufgaben interpretiert werden kann.

ist, dass die ähnlichen Varianten eines Bildes sich in einem ähnlichen Bereich des Merkmalsraums befinden, wohingegen andere Bilder des Datensatzes entsprechend Abstand zu diesen Bildern haben sollen [Che20, Zbo21]. Der Merkmalsraum wird dabei durch einen Encoder erzeugt. Bekannte Vertreter der Zielfunktionen sind NT-Xent [Che20] oder Triplet [Sch03].

Es existieren verschiedene Varianten, die CL nutzen. Im Falle von Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [Che20] wird der Encoder um einen Projektionskopf aus Multi-Layer Perceptrons (MLPs) ergänzt, um einen nieder-dimensionalen Merkmalsraum zu erzeugen. Bei BarlowTwins [Zbo21] werden ebenfalls zwei verschiedene Ansichten einer Menge von Bildern erzeugt, wobei die Kreuzkorrelationsmatrix der unterschiedlichen Bilder im latenten Merkmalsraum berechnet wird. Das Ziel ist das Erzeugen einer Einheitsmatrix, also dass die Merkmale der gleichen Ausgangsbilder korrelieren, jedoch nicht mit unterschiedlichen Ausgangsbildern. Der Ansatz Momentum Contrast (MoCo) verwendet einen Abfrageencoder (regulär trainiert) und einen Momentumencoder (lineare Interpolation zwischen den alten Parametern und Parametern des Abfrageencoders) [He20]. Mit einer integrierten Warteschlange ist es möglich, mehrere unterschiedliche Bilder aus anderen vorherigen Schritten des Trainings abzufragen. In der Methode Bootstrap Your Own Latent (BYOL) [Gri20] gibt es ebenfalls zwei Encoder, wobei ein Encoder über einen exponentiell gleitenden Mittelwert der Parameter des anderen Encoders aktualisiert wird. DenseCL [Wan21a] basiert auf MoCo, allerdings wird ein Projektionskopf, wie in SimCLR, sowie ein weiterer Projektionskopf für Vorhersagen auf Pixelebene (engl. dense predictions) hinzugefügt.

2.4.3 Semi-Supervised Learning

Nach Jing und Tian [Jin21] fasst SemSL Methoden zusammen, die annotierte und nicht annotierte Daten im Lernprozess kombinieren. Quali et al. [Oua20]

präsentiert die verschiedenen Grundideen von SemSL. In Bezug auf den spezifischen Einsatzbereich der biomedizinischen Bildverarbeitung bietet die Arbeit von Peng et al. [Pen21] einen umfassenden Überblick über die vielfältigen Ansätze.

Zum einen versucht Konsistenzregularisierung/-training auszunutzen, dass bei Veränderungen der Eingabebilder auch konsistente Änderungen der Vorhersagen resultieren bzw. bei geringen Abwandlungen Vorhersagen ähnlich bleiben. Zum anderen existiert der Selbsttraining-Ansatz. Hier wird im ersten Schritt versucht, mit den annotierten Daten ein Modell zu trainieren, um dieses im zweiten Schritt zum Generieren von Pseudo-Annotationen für alle nicht annotierten Bilder zu verwenden. Damit steht am Ende ein größerer Umfang an annotierten Bildern für das Training des finalen Modells zur Verfügung. Ko-Training versucht zwei Ansichten auf die Daten zu nutzen (z. B. für eine Klassifikation von Inhalten im Internet, deren Textinhalt und Bilder), die komplementäre Informationen beinhalten, um weitere Annotationen zu erzeugen, wenn Vorhersagen für beide Ansichten die ähnliche Vorhersage hervorrufen. Ferner gibt es generative und graphbasierte Methoden. Da der Entwicklungsstand durch Methoden basierend auf Konsistenzregularisierung oder Selbsttraining dominiert wird, sollen die jeweiligen Ausprägungen der Ansätze im Weiteren detailliert beschrieben werden.

Bekannte Vertreter für Konsistenzregularisierung sind das II-Modell [Lai17] (konsistente Vorhersage bei verschiedenen Augmentierungen), Temporal Ensembling [Lai17] (konsistente Vorhersage im Laufe der Trainingszeit) oder MeanTeacher [Tar17] (Konsistenz zwischen Vorhersagen von Schüler-Modell, trainiert auf annotierten/nicht annotierten Daten, und Lehrer-Modell, mit Parametern, die aus gleitendem Mittelwert der Schüler-Modell-Parameter berechnet werden).

Einen anderen Weg geht der Cross Teacher Training (CCT) [Xia22] Ansatz, der zwar auch auf Konsistenzregularisierung beruht. Allerdings werden die Augmentierungen der Daten im latenten Merkmalsraum statt direkt auf Bildern durchgeführt.

Die Ansätze Pseudo-Label [Lee13] bzw. deren Weiterentwicklung Meta-Pseudo-Label [Pha21] oder ST [Yan22] sind Vertreter aus dem Bereich Selbsttraining.

In der Methode FixMatch [Soh20] erfolgt eine Kombination von Konsistenzregularisierung und Selbsttraining, wobei zudem Curriculum Learning, also die Idee von einfachen hin zu schweren Beispielen zu lernen, integriert ist. Es werden nur Vorhersagen bezüglich nicht annotierter Daten berücksichtigt, welche eine hohe Konfidenz aufweisen.

Darüber hinaus erfolgt eine Kombination verschiedener Methodenbausteine in ST++ [Yan22]. Einerseits wird dort durch verschiedene Augmentierungen der Bilder eine Konsistenzregularisierung angestrebt. Andererseits ist Selbsttraining integriert. Zudem wird Curriculum Learning verwendet, indem auf Basis der Entwicklung von Vorhersagen zu unterschiedlichen Trainingszeitpunkten Bilder als einfach eingestuft werden, falls sich Prädiktionen im zeitlichen Verlauf wenig unterscheiden. Ziel ist es, dadurch zu vermeiden, dass durch das Selbsttraining die Güte des DNN verschlechtert wird, wenn das Netz falsche Vorhersagen bezüglich der unbekanntenen Daten macht.

2.4.4 Transfer Learning und Ähnlichkeiten von Datensätzen

Nach Pan und Yang [Pan10] kann TL als Methode mit Ziel definiert werden, bei gelöster Aufgabenstellung in einer bekannten Domäne (Ursprungsdomäne) eine neue Aufgabe in einer anderen Domäne (Zieldomäne) zu lösen. Es gilt dabei im Allgemeinen die Fragen zu klären: „Welches Wissen ist gemeinsam und gilt es zu übertragen?“, „Wie kann das Wissen übertragen werden?“ und „Ist es überhaupt sinnvoll, Wissen zu übertragen?“. Generell lässt sich TL in die Kategorien induktiv (Annotationen in Zieldomäne vorhanden), transduktiv (Annotationen nur in Ursprungsdomäne existent) und unüberwacht (Annotationen weder in Ziel- noch Ursprungsdomäne verfügbar) einteilen [Sch23e]. Nach Kouw und Loog [Kou18] handelt es sich bei der Domänenanpassung um eine Untergruppe des transduktiven TL, wie z. B.

in [Sch21e] genutzt, um Nachtbilder in Tagbilder zu verwandeln. Die Methoden setzen sich hier mit der Fragestellung auseinander, wie bei gleicher Aufgabenstellung die Merkmale der beiden unterschiedlichen Domänen aneinander angepasst werden können, um vom Wissen der gelösten Aufgabe bezüglich der nicht gelösten Aufgabe zu profitieren. Die vorliegende Arbeit fokussiert sich auf die Domänenanpassung, da untersucht werden soll, wie sich die Annotationen aus verfügbaren Datensätzen für neue Probleme eines gleichen Aufgabentyps nutzen lassen.

Die Anzahl der verfügbaren annotierten Datensätze für SL ist in den vergangenen Jahren stark angestiegen [Noy20], auch im Kontext der Biomedizin [Lit17, Moe19]. Es gilt jedoch systematisch für Nutzer eine ähnliche gelöste Domäne für eine vorliegende ungelöste Problem-domäne zu finden. Die Software DataMed [Che18] ermöglicht die Suche nach ähnlichen biomedizinischen Datensätzen basierend auf deren Metadaten. Interaktive Sprachmodelle wie ChatGPT [Sch22f], LaMDA [Che22] oder dessen Nachfolger Bard [Pic23] sind veröffentlicht worden, um für verschiedene Fragestellungen zum Generieren von Antworten zu dienen. Entsprechend kann das Sprachmodell ebenfalls genutzt werden, um textbasiert einen ähnlichen Datensatz zu erfragen. Beispiele hierfür sind in Abbildung 2.5 gezeigt, wobei deutlich wird, dass nicht jede Antwort des Sprachmodells sinnvoll erscheint, da die genannten Datensätze teilweise nicht existieren.

Die Autoren in [Gat16] stellen eine Methodik auf Basis von Gram-Matrizen vor, mit welcher Inhalt (z. B. ein bestimmtes Objekt) und Stil (beispielsweise eine spezifische Textur oder ein markantes Farbspektrum) getrennt aus Bildern extrahiert werden können, um gezielter Bilder zu analysieren.

Eine Berechnung von Distanzen bezüglich der Pixelwerte von Bildern ist aufgrund des „Fluchs der Dimensionen“ [Bel57] nicht zielführend. In der Arbeit von Cheplygina et al. [Che17] wird für Klassifikationsprobleme ein latenter Meta-Merkmalraum¹ erzeugt, um basierend auf euklidischen Distanzen in

¹ Der Begriff „Meta“ kommt zum Tragen, da der Merkmalsraum nicht aus den eigentlichen Bildmerkmalen aufgespannt wird, sondern in einer übergeordneten Meta-Ebene mittels der Klassifikatorergebnisse gebildet wird.

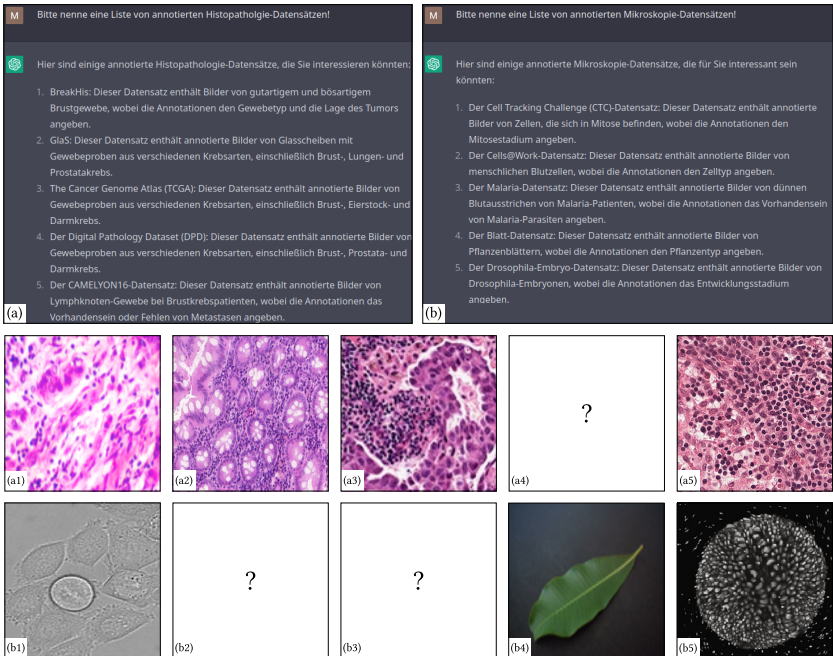


Abbildung 2.5: Antworten von ChatGPT. Die Antworten des Sprachmodells ChatGPT [Sch22f] zu den Anfragen nach annotierten Datensätzen bezüglich Histopathologie-Daten (a) und Mikroskopie-Daten (b) sind gezeigt [Sch22f]. Beispielbilder für Anfrage (a) und (b) finden sich in (a1-a5) [Cha13, Eht17, Sir17, Spa16] bzw. (b1-b5) [Cho19, Jug14, Ulm17]. Falls der Datensatz nicht auffindbar gewesen ist, wird ein Fragezeichen dargestellt.

diesem Raum die Ähnlichkeit der Datensätze zu bewerten. Der latente Merkmalsraum wird aus den Ergebnissen von mehreren einfacheren Klassifikatoren gebildet, die auf dem eigentlichen Datensatz überwacht trainiert werden. Der Nachteil dieses Ansatzes ist, dass im Falle neuer Datensätze alle Metriken erneut berechnet werden müssen. Godau und Maier-Hein [God21] bauen in ihrer Arbeit darauf auf und erzeugen einen Fingerabdruck für die verschiedenen Datensätze in Form eines Merkmalsvektors statischer Länge,

welcher sowohl Bild- als auch teilweise Annotationsinformationen miteinbezieht. Es wird in der vorgestellten Methode entweder i) pro Datensatz ein Klassifikator überwacht trainiert oder ii) ein von den jeweiligen Daten unabhängiger, auf ImageNet [Den09] trainierter Klassifikator verwendet, um Merkmale der Bilder zu extrahieren. Die Merkmale werden mithilfe unterschiedlicher Ansätze, wie z. B. der Fisher-Informationsmatrix (siehe [Ach19]) weiterverarbeitet, um den Fingerabdruck zu generieren.

Allerdings werden in beiden genannten Methoden weder der Fall von semantischer Segmentierung noch Instanzsegmentierung betrachtet. Ferner fokussieren sich die genannten Methoden primär auf latente Bildmerkmale und betrachten keine Unsicherheit von DNNs im vorgeschlagenen Ansatz. Zudem sind häufig Klassifikationsannotationen in den Ansätzen erforderlich oder es erfolgt eine Merkmalsextraktion auf Basis des Realbilderdatensatzes ImageNet, die nicht auf den individuellen Stil der biomedizinischen Daten zugeschnitten ist.

Nach Auswahl eines ähnlichen Datensatzes gibt es verschiedene Möglichkeiten im Rahmen des DL die Domänenanpassung durchzuführen. Falls keine Annotationen der Zieldomäne vorhanden sind, bietet es sich an, das Modell der Ausgangsdomäne direkt für die neue Aufgabe unverändert zu übernehmen. Liegen jedoch Annotationen im geringeren Umfang aus der Zieldomäne vor, wird das Modell häufig einem Fine-Tuning unterzogen, was beispielsweise in [Sch21e] untersucht worden ist. Dabei soll jedoch das Phänomen „Catastrophic Forgetting“, also das Wissen aus der Ursprungsaufgabe zu verlieren, vermieden werden [Kir17, Sch23e]. Eine Strategie ist, das Training auf der Zieldomäne durchzuführen, jedoch mit verminderter Lernrate [Cho20]. Dies wird z. B. im Falle eines großen Unterschieds der Domänen oder bei einem umfangreich annotierten Datensatz der Zieldomäne genutzt. Ferner kann lediglich eine Teilmenge der Schichten des DNN nachtrainiert werden. In der Regel werden dann häufig die Schichten am Beginn des Netzes fixiert, da Basismerkmale wie Kanten, Texturen oder Ecken unabhängig von der Domäne sind [Cho20].

Zudem gibt es elaborierte Ansätze wie SpotTune, bei denen über ein Abstimmungsnetzwerk adaptiv entschieden wird, ob eine Schicht fixiert oder anpassbar ist [Guo19]. Hingegen schlagen Liu et al. [Liu22] ein kombiniertes Training vor, in dem sowohl die Daten der Zieldomäne als auch dazu die am ähnlichsten vorliegenden Daten der Ausgangsdomäne genutzt werden. Die Autoren stellen heraus, dass dies im Speziellen sinnvoll ist, wenn der Datensatz der Zieldomäne besonders klein ist.

Allerdings bleibt aus den genannten Beiträgen offen, welche der Methoden für eine Anwendung auf biomedizinische Segmentierungsprobleme sinnvoll ist. Die bisherigen Untersuchungen fokussieren sich auf Klassifikationsproblemen. Ferner sind die Analysen im Stand der Technik häufig mit Echtweltdatensätzen wie ImageNet [Den09] und nicht bezüglich biomedizinischer Daten durchgeführt worden. Ferner fehlt es an einer umfassenden Methodik, welche die Schätzung von Ähnlichkeiten zwischen Datensätzen mit der anschließenden Domänenanpassung direkt für biomedizinische Segmentierungsprobleme verknüpft.

2.5 Zusammenfassung offener Probleme

Zur Komplettierung der Darstellung des Entwicklungsstands sollen nachfolgend die offenen Probleme zusammengefasst werden:

- Es fehlen Methoden im Hinblick auf die Annotation von Bilddatensätzen, die Nutzern zeiteffizientes Annotieren unter Sicherstellung hoher Qualität ermöglichen und dabei auch insbesondere für Segmentierungsprobleme nutzbar sind.
- Der Stand der Technik bietet keine ausführlichen Analysen zum Einfluss inkonsistenter Annotationen, insbesondere im Hinblick auf kleine biomedizinische Datensätze bei Segmentierungen.
- Es mangelt an leistungsfähigen Algorithmen zur vollautomatisierten Auswertung heterogener und komplexer Bilder aus HDEs, die strukturiert angeordnet sind.

- Nutzern fehlen Anwendungsempfehlungen, welche Methoden aus dem Bereich „annotations-effizientes Lernen“ (vgl. Abschnitt 2.4) hilfreich im Falle von HDEs sind. Insbesondere bleibt diese Frage für Segmentierungen offen. Es mangelt an einem Konzept mit entsprechenden Metriken, um einen fairen Vergleich im identischen Softwaregerüst unter gleichen Randbedingungen zu ermöglichen.
- Es bedarf an Metriken, insbesondere im Falle der Segmentierung, die Ähnlichkeiten zwischen Datensätzen quantifizieren, um bekannte Datensätze im Sinne von TL zur Lösung von neuen Problemen effektiv zu nutzen. Lösungen fehlen, die keine Annotationen voraussetzen, aber dennoch im Rahmen des Vergleichs der Bilder auf die jeweilige Domäne zugeschnitten sind.
- Es gibt derzeit keine zusammenhängende und frei öffentlich zugängliche Lösung, die alle notwendigen Software- und Hardwarehürden für Forscher löst, um diesen im Rahmen der Anwendung von Bildverarbeitungspipelines eigenständiges Handeln zu ermöglichen.
- Zudem fehlt ein Gesamtkonzept, welches alle oben genannten Punkte berücksichtigt. Damit gibt es für Forscher keine direkt einsetzbare sinnvolle Lösung für die HD-Bildverarbeitung, welche den gegebenen Anforderungen in einem interdisziplinären Projekt gerecht wird.

3 Konzept und Methoden

3.1 Konzept

Nach den Darlegungen der offenen Probleme in Abschnitt 2.5, ist ein Konzept notwendig, um die Probleme zu lösen. Das neu entwickelte Konzept zur Lösung der in Abschnitt 1.2 vorgestellten Forschungsfrage bzw. der offenen Probleme ist schematisch in Abbildung 3.1 illustriert. Das Konzept orientiert sich an der Publikation [Sch23b], die im Rahmen der vorliegenden Arbeit entstanden ist. Die Neuheitsbeiträge des entwickelten Konzepts sind explizit rot markiert.

Es soll eine neue Methodik zur automatisierten und genauen Interpretation von Bilddaten aus Hochdurchsatzexperimenten (HDEs) im Rahmen von interdisziplinären Datenanalyseprojekten entwickelt werden.

Den Ausgangspunkt des vorliegenden Bildverarbeitungsproblems bilden die Rohbilddaten und das Expertenwissen aus der Domäne zur Dateninterpretation. Es erfolgt eine Aufteilung der Bilder in einen Anwendungsdatensatz und einen nicht annotierten Lerndatensatz.

Den ersten neuen Methodenbaustein bildet das *assistierte Annotieren* im Vergleich zum bisherigen Workflow ohne Unterstützung von Experten. Erstmals werden detailliert und ganzheitlich Methoden mit Fokus auf die Variabilität von Experten sowie die Effizienzsteigerung präsentiert, um möglichst schnell aus Expertenwissen einen partiell annotierten Datensatz zu erzeugen, der korrekt und konsistent ist. Eine ausführliche Vorstellung der Methodik ist in Abschnitt 3.3 zu finden.

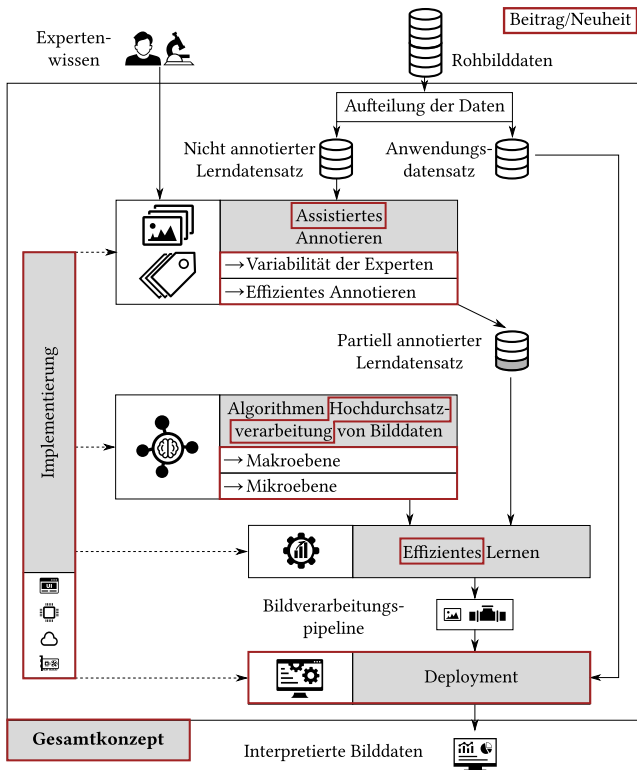


Abbildung 3.1: Konzeptüberblick. Die Neuheitsbeiträge der Arbeit sind mittels (–) markiert. Rohbilddaten werden in einen Anwendungsdatensatz und einen nicht annotierten Lerndatensatz aufgeteilt. Mittels des assistierten Annotierens unter dem Fokus von Effizienz und Variabilität der Experten entsteht ein partiell annotierter Lerndatensatz durch Einbezug von Expertenwissen. Der Datensatz dient für die Parametrierung der entwickelten Algorithmen zur HD-Bildverarbeitung, welche sowohl die Makro- als auch Mikroebene berücksichtigen. Mittels Methoden aus dem Bereich „effizientes Lernen“ resultiert die finale Bildverarbeitungs-pipeline. Diese Pipeline wird im Deployment für die Anwendung ausgerollt, um interpretierte Bilddaten für den Anwendungsdatensatz oder neu generierte Daten zu erzeugen. Generell sind unterschiedliche Formen der Implementierung in den aufgeführten Prozesselementen notwendig [Sch23b].

Bei der Hochdurchsatz (HD)-Bildverarbeitung ist eine Prozessierung von Einzelelementen (Mikroebene) unterschiedlicher Art und Weise innerhalb von Gitterstrukturen (Makroebene) erforderlich (vgl. Abschnitt 3.2). Neu entwickelte, maßgeschneiderte **Algorithmen zur Hochdurchsatzverarbeitung**

von Bilddaten bilden einen weiteren Beitrag der Arbeit, der in Abschnitt 3.5 erläutert wird. Ein entworfenener Algorithmus zur Gitterverarbeitung ermöglicht erstmals eine vollautomatisierte Auswertung. Zudem ist der Transfer von Algorithmen im Bereich Deep Learning (DL) auf den Anwendungsfall der HD-Bildverarbeitung ein weiterer Neuheitsaspekt, da die Übertragbarkeit zwischen verschiedenen Domänen Änderungen in Methoden erfordern. Ziel ist es, das Potenzial von Deep Neural Networks (DNNs) bei komplexen Problemen der HD-Bildverarbeitung zu nutzen.

Tests hinsichtlich **effizienten Lernens** existieren auf gängigen Realwelt-Benchmarkdatensätzen für einen beschränkten Umfang an Problemklassen. Zur Erweiterung des Wissensstandes ist ein systematischer Vergleich von elaborierten annotations-effizienten Lernverfahren zu Standard-Lernverfahren notwendig, um damit Forschern mit Fokus auf die Anwendung erstmalig eine Empfehlung für die Praxis bereitzustellen. Problemklassen, wie die Segmentierung, sollen dabei erstmals erschlossen werden, da der Fokus bisheriger Arbeiten auf Klassifikationsproblemen liegt. Zudem gilt es neben der Verfahrensgüte weitere, für den Anwender relevante Metriken, wie z. B. den Rechenaufwand, zusätzlich miteinzubeziehen. Damit können die Verfahren von Anwendern zielorientierter eingesetzt werden, um den Annotationsaufwand unter den vorliegenden Rahmenbedingungen zu minimieren. Details hierzu finden sich in Abschnitt 3.6.

Neue umfangreiche Methoden werden ausgearbeitet, um entwickelte Algorithmen im **Deployment** auszurollen. Damit können Anwender die Algorithmen selbstständig nutzen. Insbesondere die Einbindung von High-Performance Computing (HPC) Clustern und Graphical User Interfaces (GUIs) in das Gesamtkonzept stellt einen wichtigen Teil des Neuheitswerts dieses Methodenbausteins dar. Hürden, wie fehlende lokale Recheninfrastruktur, eingeschränkte Nutzbarkeit für fachfremde Anwender oder komplizierte Installationen, werden von diesem entwickelten Konzeptelement überwunden. So kann die Entwicklung mit der Nutzung von Auswertungsverfahren kombiniert werden. Erläuterungen diesbezüglich finden sich in Abschnitt 3.7.

Der Konzeptbaustein **Implementierung** (vgl. Kapitel 4) setzt sich mit den konkreten Neuentwicklungen (z. B. entwickelte Softwarepakete oder prototypische Serverarchitekturen) für die praktische Umsetzung der präsentierten Methoden auseinander.

Das umfassende **Gesamtkonzept** wird durch die Komposition der Einzelelemente abschließend vervollständigt, wodurch der Neuheitswert der Arbeit verstärkt wird. Das Konzept dient damit Forschern als Nachschlagewerk bzw. Grundgerüst, um die Bildauswertung von HDEs in interdisziplinären Projekten erfolgreich durchzuführen.

3.2 Vorbereitungen

3.2.1 Notation

Allgemein Vorbereitend soll auf die verwendeten Notationen eingegangen werden (siehe Abbildung 3.2). Ein zu prozessierendes Eingangsbild $\mathbf{x} \in \mathbb{N}^{H \times W \times C}$ mit Bildhöhe $H \in \mathbb{N}$, Bildbreite $W \in \mathbb{N}$ und Bildtiefe $C \in \mathbb{N}$ wird über Bildkoordinaten¹ (u, v) beschrieben, wobei u die horizontale bzw. v die vertikale Koordinatenrichtung charakterisiert. Gängige HDEs sind Gitterstrukturen, bestehend aus Elementen $\mathbf{x}_{i,j}$ (Zeile $i = 1, \dots, I$ bzw. Spalte $j = 1, \dots, J$), die es auszuwerten gilt, wobei die Zeilenanzahl über $I \in \mathbb{N}$ und Spaltenanzahl mittels $J \in \mathbb{N}$ beschrieben werden.

Die Gesamtbildverarbeitungspipeline mit generischen Parametern θ zur Erzeugung der Ausgabe \mathbf{y} sei über $f_{\theta, \text{total}}: \mathbf{x} \mapsto \mathbf{y}$ notiert. Dabei kann die Analyse der HDEs auf verschiedenen Ebenen betrachtet werden [Sch21b]. In der Makroebene ist die Erkennung und Analyse von gitterförmigen Strukturen erforderlich. Die Gitterparameter sind für die nachgelagerte automatisierte Auswertung pro Element (Mikroebene) erforderlich. Um jedes Element $\mathbf{x}_{i,j}$ auswerten zu können, müssen der Drehwinkel $\alpha \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$, der Mittelpunkt

¹ Um die Präsentation der Notation übersichtlicher zu gestalten, wird der zweidimensionale Sonderfall betrachtet, wodurch keine Koordinate für die Tiefe des Bildes eingeführt wird.

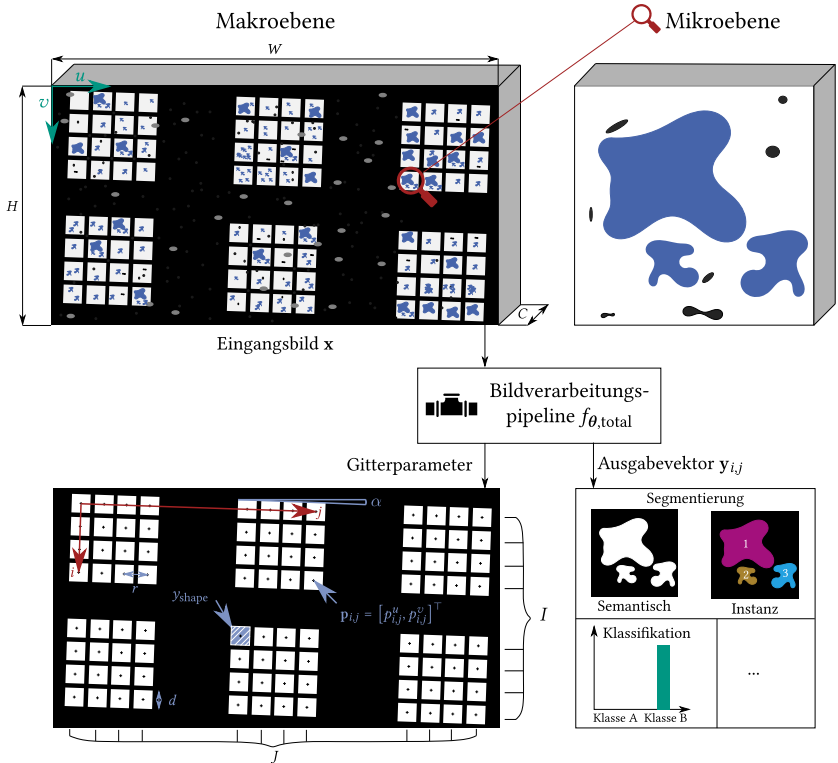


Abbildung 3.2: Notation. Ziel der Bildverarbeitungspipeline $f_{\theta, total}$ ist, ein Eingangsbild x gegeben im (u, v) -Koordinatensystem zu prozessieren, um auf der Makroebene die Gitterparameter für das automatisierte Prozessieren in HD zu ermitteln. Ferner sollen auf der Mikroebene aufgabenspezifische Ergebnisse $y_{i,j}$ für jedes Element $x_{i,j}$ erzeugt werden. Gitterparameter sind die Verdrehung α , der Mittelpunkt des Elements i, j notiert über $p_{i,j} = [p_{i,j}^u, p_{i,j}^v]^T$, die Elementform y_{shape} , der zugehörige Formparameter d im (i, j) -Koordinatensystem und die Zeilen- (I) bzw. Spaltenanzahl (J). Der Abstand zweier Elemente eines Subgitters sei r .

$p_{i,j} = [p_{i,j}^u, p_{i,j}^v]^T$ des Elements eines Gitters, dessen Form y_{shape} und zugehöriger Formparameter $d > 0$ ermittelt werden. Im Folgenden soll sich die Form $y_{shape} \in \{\text{circular, square}\}$ auf Kreise bzw. Quadrate beschränken. Der Formparameter d charakterisiert die Kantenlänge im Falle des Quadrats bzw. den

Durchmesser für Kreise. Wie in Abbildung 3.2 angedeutet, können Elementgruppen auftreten, wodurch die Abstände zwischen Elementen nicht identisch sein müssen. Dieser Fall soll im Weiteren als Sub-Gitter bezeichnet werden. Der Abstand $r > 0$ beschreibt die Distanz zwischen zwei benachbarten Elementen eines Sub-Gitters. Die Ausgabe $\mathbf{y}_{i,j}$ bezüglich Element $\mathbf{x}_{i,j}$ der Mikroebene ist von der jeweiligen Problemklasse abhängig und kann beispielsweise mehrdimensional bei Segmentierungen von Bildelementen ($\dim(\mathbf{y}_{i,j}) = 2$) oder eindimensional im Falle einer Klassifikation (allgemeines Mehrklassenproblem) des Gesamtelements ($\dim(\mathbf{y}_{i,j}) = 1$) sein. Die prädierte Ausgabe wird über $\hat{\mathbf{y}}_{i,j}$ notiert, da Vorhersagen aufgrund von Fehlern im Allgemeinen von Zielgrößen abweichen können. Ferner wird im Kontext der Expertenvariabilität eine verrauschte Annotation/Ausgabegröße mittels $\tilde{\mathbf{y}}_{i,j}$ beschrieben.

Datensätze Für Maschinelles Lernen (ML) wird im Rahmen der datengetriebenen Entwicklung ein Datensatz \mathcal{D} zur Parametrierung von $f_{\theta, \text{total}}$ herangezogen. Der Datensatz $\mathcal{D} = \{(\mathcal{X}, \mathcal{Y})\}$ setzt sich im Allgemeinen aus der Menge $\mathcal{X} = \{\mathbf{x}_i \mid i = 1, \dots, N\}$ aller $N \in \mathbb{N}$ Bilder \mathbf{x}_i und zugehörigen Annotationen $\mathcal{Y} = \{\mathbf{y}_j \mid j \in \mathcal{A} \subseteq \{1, \dots, N\}\}$ zusammen¹. In diesem Zusammenhang bezeichnet \mathcal{A} die Menge aller Indizes, bei denen eine Annotation zum Bild vorliegt. Ergo ergibt sich daraus ein nicht annotierter Datensatz \mathcal{D}^{na} im Falle von $\mathcal{A} = \emptyset$, ein vollständig annotierter Datensatz \mathcal{D}^{a} bei $\text{card}(\mathcal{A}) = N$ bzw. ein partiell annotierter Datensatz \mathcal{D}^{pa} für den Fall von $1 \leq \text{card}(\mathcal{A}) < N$ (vgl. Abbildung 3.3). Die Annotationsrate sei definiert gemäß

$$\rho = \frac{\text{card}(\mathcal{Y})}{\text{card}(\mathcal{X})} = \frac{\text{card}(\mathcal{A})}{N} \in [0, 1], \quad (3.1)$$

wobei $\rho = 1$ für \mathcal{D}^{a} , $\rho \in (0, 1)$ für \mathcal{D}^{pa} und $\rho = 0$ im Falle \mathcal{D}^{na} gelte.

¹ Es soll im Rahmen der Notation angemerkt werden, dass sich $\mathbf{x}_{i,j}$ im Speziellen auf die Bildelemente des Gitters bezieht, wohingegen \mathbf{x}_i ein generisches Bild eines Datensatzes beschreibt, das nicht zwangsläufig Teil der Gitterstruktur sein muss.

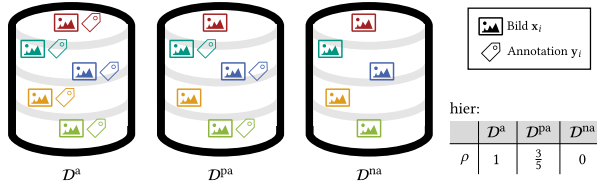


Abbildung 3.3: Überblick Datensätze. Die unterschiedlichen Datensatzarten „vollständig annotiert“ D^a , „partiell annotiert“ D^{pa} und „nicht annotiert“ D^{na} sind im Überblick dargestellt. Einzelne Datensatzelemente werden über Farbkodierung unterschieden. Zusätzlich sind beispielhaft die jeweiligen Annotationsraten ρ dargestellt [Ret21].

3.2.2 Bewertungsmetriken

Verschiedene Bewertungsmetriken Q zur Evaluierung des Konzepts werden nachfolgend vorgestellt. Ausführliche Beschreibungen finden sich als Ergänzung in Anhang D. Im Allgemeinen sind im Ergebnisteil (vgl. Kapitel 5) die Zielrichtungen der Metriken Q über \uparrow für überlegene Ergebnisse mit steigender Metrik bzw. \downarrow für den umgekehrten Fall gezeigt.

Algorithmen Hochdurchsatzbildverarbeitung Für die Bewertung der Gitterschätzung soll die Metrik

$$Q_{\Delta p} = \frac{1}{I \cdot J \cdot r} \sum_{i=1}^I \sum_{j=1}^J \operatorname{argmin}_{\hat{p} \in \hat{\mathcal{P}}} \left\| \mathbf{p}_{i,j} - \hat{\mathbf{p}} \right\|_2 \in [0, \infty) \quad (3.2)$$

genutzt werden, wobei $\hat{\mathcal{P}}$ die Menge der geschätzten Gitterpunkte darstellt. Es wird eine Normierung über den Sub-Gitterabstand r und die Gitterelementanzahl $I \cdot J$ für eine vereinfachte Interpretation durchgeführt. Mittels

$$Q_{\Delta d} = \frac{|d - \hat{d}|}{d} \in [0, \infty) \quad (3.3)$$

wird die final resultierende Schätzung des Formparameters \hat{d} eines Gitters bewertet. Ein Fehlerfall der Gitterschätzung ist empirisch definiert für $Q_{\Delta p} \geq 0.1$ oder $Q_{\Delta d} \geq 0.1$, da für diesen Wertebereich die Gittervorhersagen für Anwender unbrauchbar sind.

Ferner wird die Gesamtzeit zur Bestimmung des Gitters über

$$Q_{t_{\text{grid}}} = Q_{t_{\text{grid, process}}} + Q_{t_{\text{grid, user}}} \quad (3.4)$$

charakterisiert, die in Rechenzeit $Q_{t_{\text{grid, process}}}$ und Aufwand für den Nutzer $Q_{t_{\text{grid, user}}}$ zur Beurteilung des Grads der Automatisierung unterteilt werden kann. Für die Mikroebene ergeben sich analog die Kennzahlen $Q_{t_{\text{element, process}}}$ bzw. $Q_{t_{\text{element, user}}}$ im Hinblick auf die Auswertung von Einzelementen, die in Summe die Gesamtzeit

$$Q_{t_{\text{element}}} = Q_{t_{\text{element, process}}} + Q_{t_{\text{element, user}}} \quad (3.5)$$

bilden [Sch21b].

Gängige Metriken zur Bewertung der Qualität von semantischer Segmentierung sind der Dice-Sørensen Coefficient (DSC) $Q_{\text{DSC}} \in [0,1]$ (D.6), oftmals auch als F1-Score bezeichnet, oder der Jaccard Index (JI) $Q_{\text{JI}} \in [0,1]$ (D.5), auch Intersection over Union (IoU) genannt [Tah15]. Im Falle der Instanzsegmentierung wird häufig auf den Aggregated Jaccard Index (AJI) $Q_{\text{AJI}} \in [0,1]$ bzw. in der verbesserten Form Q_{AJI^+} (vgl. Algorithmus 9) ohne Überbestrafung durch direktes Instanzmapping zurückgegriffen [Gra19, Kum20]. Alternativ kann auch instanzweise Q_{DSC} berechnet werden, was über $Q_{\text{DSC}^2} \in [0,1]$ notiert wird [Vu19]. Für die Betrachtungen von Veränderungen bzw. Verbesserungen der Metriken wird die Notation $\Delta Q_{(\cdot)}$ genutzt.

Zur Beurteilung von Software-Tools für die HD-Bildverarbeitung wird zum einen das qualitative Kriterium „Unterstützung von Bildverarbeitung“ vorgeschlagen, also dass die Software eine Prozessierung von Bilddaten zulässt. Zum anderen dient das Kriterium „Möglichkeiten zur Gittererkennung“ zur Beurteilung, ob Algorithmen für die Detektion von Gitterstrukturen verfügbar sind. Zudem beschreibt das Kriterium „Integration der elementweisen Prozessierung“ das Software-Merkmal, Funktionen zur einzelnen Auswertung eines Gitterelements zur Verfügung zu stellen.

Assistiertes Annotieren Workflows für das assistierte Annotieren können einerseits qualitativ evaluiert werden. Qualitative Aspekte diesbezüglich sind die Möglichkeit der „Selektion“ von Bildern während der Annotation, ein vorgelagertes „Pre-Processing“ der Bilddaten oder die Option zur „Pre-Annotation“, um den Annotationsaufwand zu reduzieren. Weitere qualitative Kriterien können die Verfügbarkeit eines automatisierten „Post-Annotation-Processings“ oder eine „Inspektion von Annotationen“ sein. Ferner sind das Unterstützen „verschiedener Bildverarbeitungsaufgaben“ bzw. die Möglichkeit zum „direkten Training oder der nachgelagerten Anwendung“ qualitative Bewertungsmerkmale. Darüber hinaus ist das Bereitstellen einer Option zum „Versionieren der Daten“ im Rahmen des Annotationsprozesses ein zusätzliches Kriterium [Sch22e].

Andererseits kann quantitativ zur Evaluierung des Annotationsprozesses die Zeitdauer $Q_{t_{\text{annotation}}}$ betrachtet werden, die Nutzer bei unterschiedlichen Methoden benötigen.

Für Betrachtungen im Falle der Experten-Variabilität und Pre-Annotation ist die Übereinstimmung bzw. der Fehler zur absolut korrekten Annotation von Interesse. Dazu können die bereits erwähnten aufgabenspezifischen Gütekriterien wie Q_{DSC} (D.6) oder $Q_{\text{AJI}+}$ (vgl. Algorithmus 9) genutzt werden. Diese Metriken können außerdem verwendet werden, um den Einfluss von veräuschten Annotationen auf die Güte von DNNs zu ermitteln [Sch21a, Sch22b, Sch22d, Sch22e].

Im Rahmen einer automatisierten Inspektion kann die Area Under the Curve (AUC) Q_{AUC} (D.4) der Receiver Operating Characteristic (ROC) genutzt werden, um auszuwerten, wie akkurat das Ermitteln von fehlerhaften Annotationen eines Experten erfolgt, ohne einen fixen Schwellwert für das Warnen des Experten wählen zu müssen. Ein Inspektionsalgorithmus gilt als überlegen verglichen zu zufälligem Raten von Fehlern, wenn $Q_{\text{AUC}} > 0.5$ vorliegt [Sch22d]. Es wird jedoch im Weiteren empirisch eine fehlgeschlagene Inspektion bei $Q_{\text{AUC}} \leq 0.55$ definiert, da sich bis zu diesem Wert kaum ein Benefit für Nutzer ergibt [Sch23c].

Bezeichne $\Gamma_{\text{sim}}({}^A\mathcal{D}, {}^B\mathcal{D})$ ein Ähnlichkeitsmaß zwischen Annotationsstilen A und B bezüglich der Datensätze ${}^A\mathcal{D}$ bzw. ${}^B\mathcal{D}$ [Sch22d], so soll die betragsmäßige Differenz zur vorhergesagten Ähnlichkeit $\hat{\Gamma}_{\text{sim}}({}^A\mathcal{D}, {}^B\mathcal{D})$, also $Q_{\Gamma_{\text{sim}}} = |\Gamma_{\text{sim}}({}^A\mathcal{D}, {}^B\mathcal{D}) - \hat{\Gamma}_{\text{sim}}({}^A\mathcal{D}, {}^B\mathcal{D})|$, zur Bewertung genutzt werden. Im Idealfall sollen sich die Annotationsstile zwischen Datensätzen, die von verschiedenen Experten annotiert worden sind, nicht unterscheiden. Weitere Details finden sich in Abschnitt 3.4.

Effizientes Lernen Sei $\mathcal{M} = \{m_i | i = 1, \dots, M\}$ die Menge der betrachteten effizienten Lernverfahren, so sollen verschiedene Methoden $m_i \in \mathcal{M}$ anhand von den nachfolgend eingeführten Metriken verglichen werden. Die Referenzmethode des überwachten Lernens wird als m_{M+1} notiert. Zunächst wird das Gütekriterium $Q_{(\cdot)}$ linear zwischen den Messwerten $\{Q_{(\cdot)}(\rho_{\min}), \dots, Q_{(\cdot)}(\rho_{\max})\}$ bezüglich der Annotationsraten $\{\rho_{\min}, \dots, \rho_{\max}\}$ interpoliert, wodurch die Funktion $f_{Q_{(\cdot),\text{lin}}}(\rho)$ folgt, die häufig als Lernkurve bezeichnet wird. Zur Beschreibung der Gesamtgüte hinsichtlich verschiedener Annotationsraten ρ wird das integrale Gütekriterium

$$Q_{\text{int},(\cdot)} = \frac{1}{Q_{(\cdot)}(\rho = 1) \cdot (\rho_{\max} - \rho_{\min})} \int_{\rho_{\min}}^{\rho_{\max}} f_{Q_{(\cdot),\text{lin}}}(\rho) d\rho \in [0,1] \quad (3.6)$$

eingeführt. Als Normalisierung dient das Produkt aus erreichter Güte $Q_{(\cdot)}(\rho = 1) > 0$ im Falle des komplett annotierten Datensatzes und Integrationsintervalllänge $(\rho_{\max} - \rho_{\min}) > 0$. Ferner ist anzumerken, dass $Q_{\text{int},(\cdot)} \in [0,1]$ unter der Annahme einer monotonen Lernkurve gilt. In Abbildung 3.4 findet sich eine visuelle Darstellung der Metrik. Darüber hinaus wird für den Vergleich von Rechenzeiten die Metrik

$$Q_{t,\text{norm}}(m_i) = \frac{Q_t(m_i)}{Q_t(m_{M+1})} \in (0, \infty) \quad (3.7)$$

genutzt, wobei der Quotient aus Rechenzeit $Q_t(m_i) > 0$ der jeweiligen Methode m_i und Referenzmethode m_{M+1} berechnet wird.

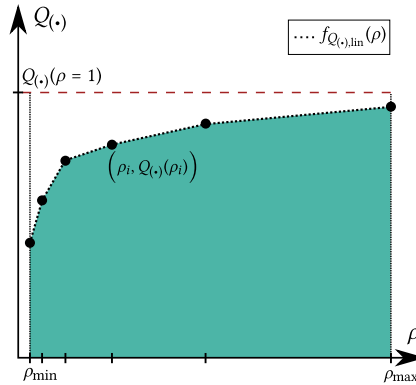


Abbildung 3.4: Illustration integrales Gütekriterium $Q_{\text{int},(\cdot)}$. Es wird über die linear approximierte Gütefunktion $f_{Q(\cdot),\text{lin}}(\rho)$ bezüglich der Grenzen ρ_{min} und ρ_{max} integriert. Das Integral ist über ■ markiert. Als Normalisierungsfaktor dient das Produkt aus Intervalllänge $(\rho_{\text{max}} - \rho_{\text{min}})$ und erreichter Güte $Q(\cdot)(\rho = 1)$ im Falle eines vollständig annotierten Datensatzes [Kep22, Ret23].

Ferner sollen die Methoden hinsichtlich „einzustellender Hyperparameter“ (Anzahl) und qualitativ bezüglich „Implementierungsaufwand“ bzw. „Hardwareanforderung“ bewertet werden. Dabei beschreiben (+/++) positive/sehr positive, (o) neutrale und (-/-) negative/sehr negative Beschaffenheiten der Methoden [Kep22, Ret21, Ret23, Rod23].

In Kontext „Ähnlichkeiten von Datensätzen“ bezeichnen $Q_{\text{Top-1}} \in [0,1]$ (D.2) und $Q_{\text{Top-3}} \in [0,1]$ (D.3) die Auswahlgenauigkeit, inwieweit der als am ähnlichsten vorgeschlagene Datensatz das beste Ergebnis¹ ist bzw. sich unter den drei besten Ergebnissen befindet [Mol23b, Rod23].

¹ Der Terminus „bestes Ergebnis“ wird im Kontext Transfer Learning (TL) verwendet und beschreibt, inwieweit das genutzte Vorwissen aus dem als ähnlichsten eingestuftem bekannten Datensatz die beste aufgabenspezifische Metrik (z. B. Q_{DSC}) auf dem zu lösenden Datensatz liefert. Somit wird beschrieben, ob die berechnete Ähnlichkeit mit dem Erfolg bei der Lösung des Bildverarbeitungsproblems einhergeht, wenn das Wissen des bekannten Datensatzes verwendet wird.

Allgemein Generell werden zur Untersuchung der Signifikanz von Resultaten p -Werte¹ im Rahmen von einseitigen Zweistichproben- t -Tests (unter der Annahme gleicher Varianzen) genutzt, die beispielsweise ausführlich in [Rei17] diskutiert werden. Die Nullhypothese lautet in diesem Fall jeweils, dass (a) die neu zu untersuchende Methode entweder genauso gut oder schlechter abschneidet als die Referenzmethode oder (b) das DNN bei einem verrauschten Datensatz gleichwertige oder bessere Ergebnisse erzielt. Signifikanz liege dabei im Falle von $p < 0.05$ vor. Untersuchte lineare Korrelationen nach Pearson sollen im Weiteren über $\xi \in [0,1]$ (D.1) notiert werden.

Softwaretools und Deployment Ein Merkmal zur Bewertung von Softwaretools ist deren „freie, nicht kommerzielle Zugänglichkeit“. Das „Bereitstellen von Benutzerhandbüchern und GUIs“ sind weitere Kriterien, welche die Bedienbarkeit charakterisieren. Zudem spielt ein einfacher „Installationsprozess inklusive niedriger Software- bzw. Hardwareanforderungen“ eine Rolle. Darüber hinaus stellt „Erweiterbarkeit“ ein zusätzliches Kriterium dar, um Software zu bewerten [Sch21b, Sch22e].

3.3 Assistiertes Annotieren

3.3.1 Überblick

Aufgrund der dargestellten Herausforderungen bei der Annotation von Datensätzen, gilt es eine Methodik zum Lösen des offenen Problems zu entwickeln. In Abbildung 3.5 ist ein Überblick hinsichtlich der vorgeschlagenen Methodik zum assistierten Annotieren dargestellt, die im Rahmen dieser Arbeit entwickelt wurde [Sch21a, Sch21c, Sch22d, Sch22e, Sch23c]. Über das **Erstellen eines Projekts** werden dem nicht annotierten Lerndatensatz

¹ Notationen: $p \geq 0.05 \rightarrow o$, $p < 0.05 \rightarrow *$, $p < 0.005 \rightarrow **$, $p < 0.001 \rightarrow ***$

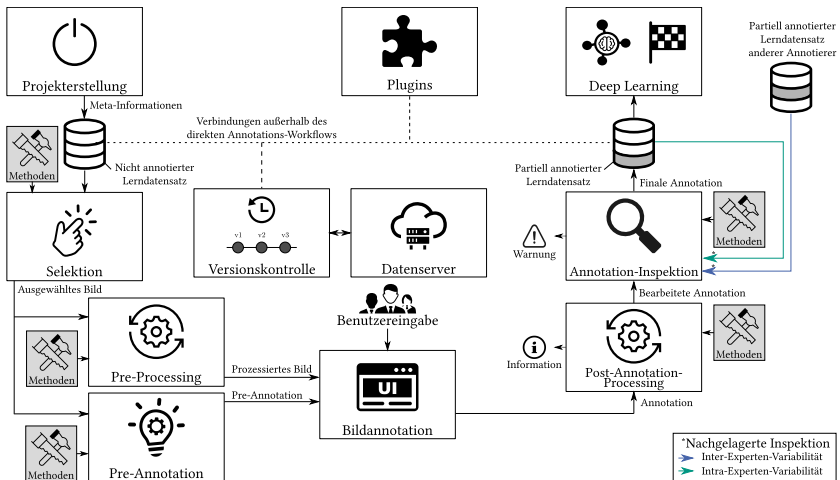


Abbildung 3.5: Überblick assistiertes Annotieren. Das Konzept zum assistierten Annotieren wird in einer Übersicht präsentiert (siehe Abschnitt 3.3.1) [Sch21a, Sch22e].

Meta-Informationen hinzugefügt (siehe Abschnitt 3.3.2). Das Modul **Selektion** beeinflusst die Reihenfolge der Bilder, die dem Nutzer präsentiert werden (vgl. Abschnitt 3.3.3). **Pre-Processing** (siehe Abschnitt 3.3.4) und **Pre-Annotation** (siehe Abschnitt 3.3.5) dienen zur Vereinfachung der Annotation, indem das angezeigte Bild vorverarbeitet wird bzw. dem Nutzer erste Annotationsvorschläge gemacht werden. Je nach Bildverarbeitungsaufgabe werden für die eigentliche **Bildannotation** unterschiedliche GUIs zur Benutzereingabe angezeigt (vgl. Abschnitt 3.3.6). **Post-Annotation-Processing** (siehe Abschnitt 3.3.7) inklusive Informationen zur Nachvollziehbarkeit von Änderungen und Warnungen im Modul **Annotation-Inspektion** (siehe Abschnitt 3.3.8) erhöhen die Annotationsqualität. Dabei ist eine nachgelagerte Inspektion zur Analyse der Variabilität von Experten ebenfalls möglich. Generell sind in den jeweiligen Modulen je nach Aufgabe und Daten unterschiedliche Methoden sinnvoll. Die Anzahl der Annotationen im partiell annotierten Datensatz erhöht sich während des assistierten Annotierens, wodurch anschließend überwachte/halb-überwachte Lernverfahren im Kontext von **DL** möglich sind. Über eine **Versionskontrolle** kann die Historie der Datensätze nachverfolgt werden (siehe Abschnitt 3.3.9). In Kombination mit der Nutzung

eines *Datenservers* wird die Datensicherung und die Verteilung von Daten ermöglicht. *Plugins* runden als Integrationen von hilfreichen Erweiterungen die vorgeschlagene Methodik zum assistierten Annotieren ab (vgl. Abschnitt 3.3.10). Im Nachfolgenden wird eine umfassende Erklärung von den einzelnen Bestandteilen der neu entwickelten Methodik präsentiert.

3.3.2 Projekterstellung

Zunächst gilt es zu ermöglichen, dass dem Datensatz Meta-Informationen hinzugefügt werden können. Diese Informationen sind z. B. Spezifikationen zu vorliegenden Klassen in den gegebenen Bilddaten oder die Charakterisierung der zugrundeliegenden Problemart. Damit ist die Basis für eine weitere wissenschaftliche Datenverarbeitung gelegt [Wil16], da auf die Meta-Informationen in den weiteren Prozessschritten zurückgegriffen werden kann.

3.3.3 Selektion

Allgemein Das Selektionsmodul zielt darauf ab, Deep Active Learning (DAL) in der Praxis anwendbar zu machen und mit möglichst wenig Annotationen leistungsfähige DNNs zu parametrieren. Dabei soll dem Nutzer die Möglichkeit gegeben werden, die Reihenfolge zu beeinflussen, in welcher Bilder vorgeschlagen werden. Dies steht im Kontrast zur bisherigen Vorgehensweise, Datensätze sequentiell vom ersten aufgenommenen Bild bis zum letzten Bild des Datensatzes ohne Logik zu annotieren.

Methoden Mögliche Methoden des Vorschlags sind hier zum einen das zufällige Ziehen von Bildern. Zum anderen wird dem Nutzer mittels „Cherry Picking“ die Möglichkeit bereitgestellt, Bilder über ein Auswahlfenster zu priorisieren. Ferner können mittels eines DNN und optionaler weiterer Dimensionsreduktion wie der Principal Component Analysis (PCA) Merkmale aus

Bildern extrahiert werden. Diese Merkmale können danach mittels eines gewählten Optimierungsverfahrens genutzt werden, möglichst heterogene Bilder zu ziehen [Sen18]. Dabei ist ein Training des DNN auf dem eigentlichen Datensatz nicht zwingend erforderlich, da beispielsweise auch im Sinne von TL ein auf ImageNet [Den09] vortrainiertes ResNet [He16] nutzbar ist. Zusätzlich kann im Falle eines verfügbaren unsicherheitsbewussten DNN (siehe Abschnitt 2.1.3) auch die Unsicherheit als Metrik zur Selektion der nicht annotierten Daten dienen, um den Datensatz gezielt mit schwierigen Bildern zu erweitern. Ferner stellt die im Rahmen der Arbeit entwickelte Methode „Smart Picking“ eine Mischung aus „Cherry Picking“ und DAL dar. Hier wird der Datensatz mittels ResNet [He16] und Dimensionsreduktion auf zwei Merkmale (z. B. über t-Stochastic Neighbor Embedding (SNE) [Van08], Uniform Manifold Approximation and Projection (UMAP) [Mc18] oder PCA) Nutzern in einer schnell interpretierbaren Repräsentation visualisiert. Dadurch ist es Nutzern möglich, die Eigenschaften des Datensatzes vereinfacht in zwei Dimensionen zu betrachten und Bilder zu selektieren.

3.3.4 Pre-Processing

Allgemein Es gibt eine Vielzahl an Beispielen, in welchen die Annotation von Rohbildern problematisch ist. Beispielsweise können

- schlechte Aufnahmebedingungen,
- Verzeichnungen (optische Abbildungsfehler, die ursprünglich gerade Linien einer Szene im resultierenden Bild krümmen),
- zu große Bilder, die irrelevante Bereiche beinhalten oder
- der Fall von einer vorliegenden zu geringen Bildauflösung,

Nutzern das Annotieren von Bildern erschweren. Daher ist es von Vorteil, die Bilddaten vorzuverarbeiten, was jedoch meist unter der Verwendung von separaten Softwarepaketen erfolgt und damit getrennt vom Annotationsprozess ist. Deshalb sieht die vorgeschlagene Methode vor, „Pre-Processing“ in den Ablauf zu integrieren, um Bilder direkt vorverarbeiten zu können und damit einen geschlossenen Prozess für Nutzer bereitzustellen. Es ist anzumerken,

dass die Vorverarbeitung meist lediglich dem Zweck der verbesserten Anzeige für Nutzer und damit der vereinfachten Annotation dient. In der Regel kommt es dadurch zu keinen Veränderungen der Rohdaten. Allerdings kann bei Bedarf optional auch das prozessierte Bild gespeichert werden.

Methoden Das Normalisieren von Bildern kann insbesondere im biomedizinischen Kontext bei hochauflösenden Monokanalbildern mit feiner Grauwertdiskretisierung sinnvoll sein, um relative Unterschiede sichtbar zu machen. Eine Normalisierung über die Berechnung von Perzentilen kann darüber hinaus im Falle von einzelnen Ausreißern die Darstellung verbessern. Filter zur Unterdrückung von Rauschen sind ebenfalls eine Möglichkeit zur Vorverarbeitung. Ferner kann das Bild mittels Ausschneidefunktion auf den Bereich von Interesse reduziert werden. Außerdem ist eine Möglichkeit zur Korrektur von Verzerrungen in Bildern vorgesehen, wie z. B. über die Kamerakalibrierung in [Bra00].

3.3.5 Pre-Annotation

Allgemein Häufig existieren bei der Annotation bereits Vorwissen oder Heuristiken zur Lösung des Problems (z. B. Algorithmen, welche für ein benachbartes Problem im Vorfeld entwickelt worden sind), die jedoch meist ungenutzt bleiben. Infolgedessen startet die Annotation oft von Grund auf, und Nutzer müssen alle Entscheidungen, wie das Markieren aller Segmente, neu und ohne Hilfe treffen. „Pre-Annotation“ nutzt verfügbare Algorithmen als Hilfsmittel zum Vorschlagen von Annotationen. Damit ist lediglich eine Korrektur der Annotation für den Nutzer notwendig.

Methoden Methoden zur „Pre-Annotation“ sind stark an das zugrundeliegende Bildverarbeitungsproblem gekoppelt. Als generisches Verfahren zur „Pre-Annotation“ können vortrainierte DNNs genannt werden [Str21b]. Denkbare Anwendungsfälle sind die Nutzung von Modellen aus benachbarten Problemen oder das Fine-Tuning von Modellen durch eine Erweiterung

des annotierten Datensatzes, wenn bereits ein Datensatz mit geringem Umfang vorliegt. Weitere Details zu TL/Fine-Tuning werden in Abschnitt 3.6 ausführlich aufgegriffen. Zudem eignen sich im Falle von Segmentierungsaufgaben etablierte Bildverarbeitungsansätze wie Schwellwertverfahren, die statisch oder variabel eingestellt werden können [Ret22].

Nutzer-Studie Um den Benefit von „Pre-Annotation“ für Nutzer im Hinblick auf Zeitersparnis zu eruieren, sind Nutzer-Studien anzustreben. In der konzipierten Nutzer-Studie dieser Arbeit sollen Anwender jeweils zwei Datensätze (Instanzsegmentierung und semantische Segmentierung) annotieren, wobei jeweils drei Konfigurationen (keine Vorhersagen, ein konventioneller Bildverarbeitungsalgorithmus und eine vortrainierte DL-Pipeline) untersucht werden. Für die statistische Interpretation der Ergebnisse werden mehrere Nutzer berücksichtigt, die jede Konfiguration bzw. jeden Datensatz wiederholt annotieren. Als Metriken werden einerseits die mittleren Annotationszeiten $Q_{t_{\text{annotation}}}$ betrachtet. Andererseits soll die Qualität der Annotationen über $Q_{\text{DSC}}/Q_{\text{AJI}+}$ beleuchtet werden. Als Referenz dienen die Annotationen aus den Originaldatensätzen.

3.3.6 Bildannotation

Allgemein Im Allgemeinen hängt die Art der Benutzereingabe vom zugrundeliegenden Bildverarbeitungsproblem ab (vgl. Abschnitt 2.2.1). So erfordern Segmentierungsaufgaben eine pixelweise Annotation, wohingegen im Falle der Klassifikation eine Einzelentscheidung pro Gesamtbild getroffen werden muss. Werkzeuge unterstützen oft nicht mehrere Problemklassen. Folglich sieht das vorgeschlagene Konzept für die Bildannotation vor, die GUI basierend auf der in den Meta-Informationen gegebenen Problemklasse anzupassen, um agnostisch bezüglich des Bildverarbeitungsproblems zu sein.

Methoden Nativ sind in dem vorgeschlagenen Konzept die gängigen Problemklassen i) Klassifikation, ii) semantische Segmentierung, iii) Instanzsegmentierung, iv) panoptische Segmentierung (Kombination von Instanzsegmentierung und semantischer Segmentierung), v) Objekterkennung und vi) Seed Detektion (Markierung des Objektmittelpunkts) vorgesehen.

3.3.7 Post-Annotation-Processing

Allgemein Praktische Projekte zeigen, dass bestimmte Annotationsfehler regelmäßig auftreten, wie beispielsweise Löcher in Segmentierungsmasken, die oftmals nicht beabsichtigt sind. Daher kann die automatisierte Nachbearbeitung von Annotationen sinnvoll sein, jedoch fehlt oft eine Integration dieses Gedankens in Annotationswerkzeugen. Einerseits ist die Verwendung von separaten Tools für das „Post-Annotation-Processing“ umständlich. Andererseits kann eine automatisierte Nachbearbeitung ohne eine Kontrolle bzw. die Überwachung durch den Nutzer ein Risiko darstellen, da eventuell Änderungen erzeugt werden, die nicht gewünscht sind. Zur Gewährleistung der Nachvollziehbarkeit von Änderungen ist eine Überlagerung von ursprünglicher und bearbeiteter Annotation in der entwickelten Methode möglich.

Methoden Der Fokus im vorgeschlagenen Modul bezieht sich primär auf Bildverarbeitungsprobleme aus dem Bereich Segmentierung. Hierbei können morphologische Operatoren als Methode zum „Post-Annotation-Processing“ dienen. Das Löschen von kleineren Segmenten ist ebenfalls Teil des Methodenbaukastens. Zudem ist das automatisierte Füllen von Löchern als Verarbeitungsmöglichkeit zu nennen.

3.3.8 Annotation-Inspektion

Allgemein Die Problematik von verrauschten Annotationen ist bisher nicht im Ablauf des Annotationsprozesses berücksichtigt. Deshalb integriert die vorgeschlagene Methodik das Modul „Annotation-Inspektion“, in welchem

die Konsistenz von Annotationen nach geeigneten Qualitätskriterien bewertet wird. Nutzer können Schwellwerte wählen, um das Auftreten von Warnungen zu regulieren. Im Falle einer ausgelösten Warnung wird der Experte zur Entscheidung aufgefordert, ob die Annotation überarbeitet oder beibehalten werden soll.

Methoden Ein Heranziehen von DNNs zur Überwachung ist denkbar, jedoch müssen die DL-Modelle für den Einsatz im jeweiligen Projekt verfügbar sein. Von der kombinierten Nutzung eines identischen DNN zur Pre-Annotation und Inspektion ist abzusehen, da ein „Kurzschluss“¹ im Prozess entstehen würde. Die Berücksichtigung von Elementeigenschaften wie Anzahl, Größe oder Form von Segmenten kann als Entscheidungskriterium dienen, um auf die Qualität der Annotation zu schließen. Nutzer können hierbei Expertenwissen in den Annotationsprozess integrieren, indem die Bereichsgrenzen bezüglich einer Eigenschaft angegeben werden. Das Themenfeld verrauschte Annotationen und deren automatisierten Erkennung wird in Abschnitt 3.4 ausführlicher betrachtet.

3.3.9 Versionskontrolle

In der Regel handelt es sich beim Annotieren von Datensätzen um einen iterativen Prozess. Beispielsweise ist der erforderliche Umfang der annotierten Daten im Vorfeld nicht bekannt. Veränderungen in der Bilddomäne, wie beispielsweise durch geänderte Aufnahmebedingungen verursacht, können zu einer Verschlechterung von DNNs führen. Neue Daten sind dann erforderlich. Zudem besteht die Möglichkeit, dass sich im Laufe des Projekts die festgelegten Regeln für die Annotation verändern, beispielsweise indem zusätzlich die Ränder von Objekten gekennzeichnet werden sollen. Ferner kann es in der Praxis von Interesse sein, Verbesserung bezüglich der Bildgebung oder der Annotationsqualität zwischen verschiedenen Zeitpunkten auszuwerten.

¹ Der Begriff „Kurzschluss“ bedeutet in diesem Zusammenhang, dass sich die Annotationen schwer überprüfen lassen, da der Prüfmechanismus und die Heuristik zum Generieren der Annotation identisch sind.

Zusammenfassend können sich Datensätze während der Entwicklung einer DL-Verarbeitungspipeline ändern. Dieser Aspekt wird jedoch in bisherigen Annotationswerkzeugen nicht berücksichtigt. Daher wird die Einbindung einer Versionskontrolle vorgeschlagen, die auf den Ideen von Git [Tor05] und der Nutzung von Hash-Funktionen zur dateneffizienten Umsetzung basiert [Pet21]. Somit wird ermöglicht, die Historie von Datensätzen zu analysieren bzw. zwischen Datensatzversionen zu wechseln. Darüber hinaus unterstützt die Versionskontrolle eine Verwendung von Datenservern, was in mehrfacher Hinsicht von Vorteil ist: Anwendungsfälle können die Übertragung von Daten an Rechencluster für Trainingsprozesse, die Datensicherung oder eine verbesserte Interaktion zwischen Domänenexperten und Algorithmentwicklern sein.

3.3.10 Plugins

Um Nutzern die Möglichkeit zu geben, zusätzliche Funktionalitäten, die im Kontext der Annotation von Bilddaten relevant sind, einfach verwenden zu können, ist über Plugins ein Platzhalter für Erweiterungen vorgesehen. Beispiele für diesen Methodenbaustein sind das Training bzw. die Anwendung von DNNs (nähere Informationen unter Abschnitt 3.7) oder das Generieren von Bildausschnitten im Falle von großen Bildern/vielen Objekten pro Bild, um die Annotation zu erleichtern. Zudem kann eine nachgelagerte Inspektion des Datensatzes erfolgen, um die Qualität von Annotationen zu erhöhen (Details siehe Abschnitt 3.4).

3.4 Variabilität der Experten

3.4.1 Überblick

Der Terminus „Variabilität der Experten“ soll als Sammelbegriff für verschiedene Arten von Annotationsfehlern, Ungenauigkeiten und systematische Abweichungen genutzt werden. Die Arbeit fokussiert sich im Weiteren auf die

Aufgabenklassen „semantische Segmentierung“ (vgl. Abbildung 3.6) und „Instanzsegmentierung“ (vgl. Abbildung 3.7), da diese i) besonders aufwändig zu annotieren sind sowie ii) detaillierte Analysen und Inspektionsalgorithmen im Kontext der Variabilität von Experten fehlen. Zur Veranschaulichung und Kategorisierung der Variabilität von Experten werden jeweils für beide Aufgabenklassen synthetische Datensätze (siehe Abbildung 3.6 bzw. Abbildung 3.7) eingeführt [Sch22b, Sch22d, Sch23c].

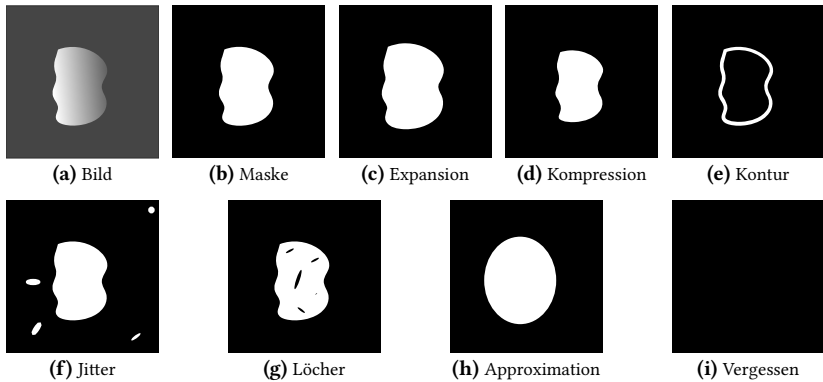


Abbildung 3.6: Synthetischer Datensatz zur semantischen Segmentierung inklusive Verunreinigungen. Ein Beispielbild (a) mit der zugehörigen korrekten Maske (b) wird im Vergleich zu diversen Verunreinigungen (c)-(i) gezeigt [Sch22d].

Zu Beginn sollen die Verunreinigungen vorgestellt werden, welche in beiden Aufgabenklassen vorhanden sind. In diesem Zusammenhang beschreibt der Parameter $\beta \in [0,1]$ den relativen Anteil der betroffenen Segmente. Eine *Expansion/Kompression* der Maske bzw. von Segmenten kann zu Variabilität innerhalb von Annotationen führen. Ferner tritt der Fall *Kontur* bei kontur-basierten Annotationstools mit automatischer Füllfunktion von Masken auf, wenn die Ränder nicht vollständig geschlossen werden. Zudem können kleinere *Löcher* oder zusätzliche Elemente in Form von *Jitter* (zu Deutsch Flackern) eine Verunreinigung darstellen. Die Vernachlässigung der genauen Form einer Instanz bzw. eines Segments und Annäherung durch Ellipsen



Abbildung 3.7: Synthetischer Datensatz der Instanzsegmentierung inklusive Verunreinigungen. Ein exemplarisches Bild (a) mit der korrespondierenden fehlerfreien Instanzmaske (b) wird verschiedenen Verunreinigungen (c)-(k) gegenübergestellt, wobei betroffene Instanzen mit weißen Linienmustern/Kreuzen gekennzeichnet sind. In diesem Beispiel beträgt der Parameter $\beta = 0.2$ (siehe Details in Abschnitt 3.4.2), da zwei von zehn Instanzen verändert worden sind [Sch22b, Sch23c].

soll als *Approximation* bezeichnet werden. Ferner ist das *Vergessen* von Annotationen ein häufiger Fehlerfall. Exklusiv für den Fall der Instanzsegmentierung sind das *Verschmelzen* von benachbarten Instanzen und das irrtümliche *Aufteilen* einer Instanz in zwei Instanzen [Löf21, Löf23]. Generell können alle Klassen von Verunreinigungen zufällig im Sinne von Intra-Experten-Variabilität auftreten. Im Gegensatz dazu beschränkt sich systematisches Annotationsrauschen im Falle von Inter-Experten-Variabilität auf die Teilmenge $\{Expansion, Kompression, Kontur, Approximation\}$.

3.4.2 Simulation der Variabilität von Experten

Um die Variabilität von Experten detailliert untersuchen zu können, wird die Methodik für deren Simulation vorgestellt. Eine Simulation der Variabilität von Experten ist notwendig, da im Allgemeinen keine Datensätze mit gleichzeitig korrekten und verrauschten Annotationen verfügbar sind. Diese Informationen sind jedoch notwendig, um beispielsweise den Einfluss von verrauschten Annotationen hinsichtlich der Genauigkeit von DNNs oder Ansätze zur Erkennung von verrauschten Annotationen evaluieren zu können. Den Startpunkt bildet dabei ein als korrekt angenommener annotierter Ausgangsdatensatz \mathcal{D} . Um die Anzahl der Experimente zu beschränken, soll auf eine Kombination von verschiedenen Fehler-Typen aus Abschnitt 3.4.1 verzichtet werden.

Intra-Experten-Variabilität

Abbildung 3.8 (oben) stellt den entwickelten Ablauf zur Simulation der Intra-Experten-Variabilität dar. Zunächst erfolgt die Aufteilung der Elemente des Datensatzes \mathcal{D} in zwei Subdatensätze ${}^A\mathcal{D}$ und ${}^B\mathcal{D}$, wobei das Aufteilungsverhältnis mittels Aufteilungsparameter $\gamma \in [0,1]$ bestimmt wird. Annahme hierbei ist, dass der Datensatz homogen ist, wodurch die Aufteilung zufällig erfolgen kann. Es gilt somit $\text{card}({}^A\mathcal{D}) = \gamma N$ bzw. $\text{card}({}^B\mathcal{D}) = (1 - \gamma) N$. Im Anschluss werden die Annotationen der Elemente von Subdatensatz ${}^A\mathcal{D}$ durch die vorgestellten Verunreinigungen in Abschnitt 3.4.1 verändert, wodurch der verunreinigte Subdatensatz ${}^A\tilde{\mathcal{D}}$ folgt. Der final resultierende verunreinigte Datensatz $\tilde{\mathcal{D}}$ bildet sich aus der Vereinigung von ${}^A\tilde{\mathcal{D}}$ und ${}^B\mathcal{D}$. Zur Vereinfachung der Untersuchung wird im Falle der semantischen Segmentierung der Parameter γ variiert (alle Segmente von Verunreinigung betroffen, $\beta = 1$). Hingegen wird bei Instanzsegmentierung jedes Bild verunreinigt ($\gamma = 1$), aber der Anteil von betroffenen Instanzen, beschrieben über den Parameter β , ist variabel. Dieses Vorgehen entspricht der Intra-Experten-Variabilität, da ein zufälliger Anteil der Bilder bzw. Instanzen verunreinigt wird.

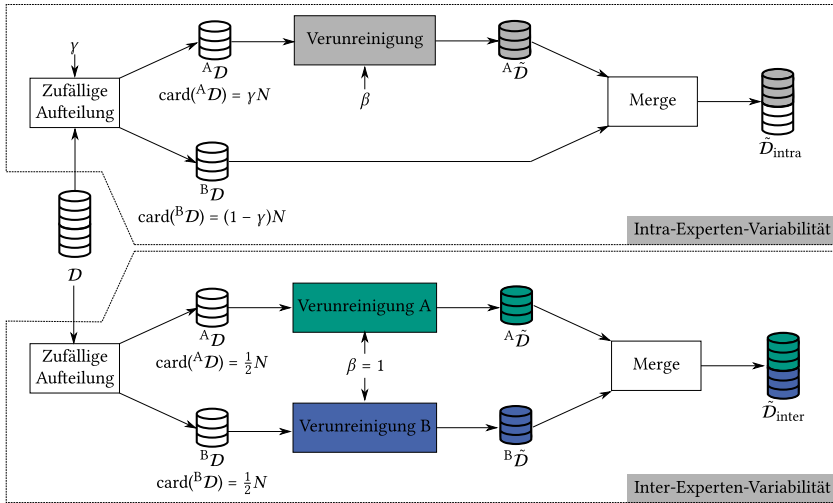


Abbildung 3.8: Simulation der Experten-Variabilität. Im Falle von Intra-Experten-Variabilität (oben) wird der Ausgangsdatensatz \mathcal{D} zufällig gemäß Parameter γ in ${}^A\mathcal{D}$ und ${}^B\mathcal{D}$ aufgeteilt, wobei in ${}^A\mathcal{D}$ Verunreinigungen hinzugefügt werden. Es folgt der verunreinigte Subdatensatz ${}^A\tilde{\mathcal{D}}$. Über β wird der Anteil von betroffenen Segmenten eingestellt. Als Resultat ergibt sich der verunreinigte Datensatz $\tilde{\mathcal{D}}_{\text{intra}}$ (vgl. „Merge“). Für die Simulation der Inter-Experten-Variabilität (unten) wird der Datensatz \mathcal{D} zufällig in Subdatensätze ${}^A\mathcal{D}$ und ${}^B\mathcal{D}$ mit halbiertem Umfang aufgeteilt. Jeweils unterschiedliche Verunreinigungen A/B werden eingefügt. Es resultiert ${}^A\tilde{\mathcal{D}}$ bzw. ${}^B\tilde{\mathcal{D}}$, die zum finalen Resultat $\tilde{\mathcal{D}}_{\text{inter}}$ zusammengefasst werden (siehe „Merge“). Dabei gilt $\beta = 1$.

Inter-Experten-Variabilität

Im Falle der Inter-Experten-Variabilität wird der Datensatz \mathcal{D} für die nachfolgenden Untersuchungen halbiert, wodurch die Subdatensätze ${}^A\mathcal{D}$ und ${}^B\mathcal{D}$ resultieren. Im Nachgang werden die Subdatensätze jeweils systematisch durch Verunreinigungen A und B beaufschlagt, wodurch die Annotationen verändert werden. Es gilt dabei $\beta = 1$, wodurch es zu einer Beeinflussung aller Segmente kommt. Final folgt $\tilde{\mathcal{D}}$ als Komposition der beiden verunreinigten Subdatensätze ${}^A\tilde{\mathcal{D}}$ bzw. ${}^B\tilde{\mathcal{D}}$. Ein Überblick dazu findet sich ebenfalls in Abbildung 3.8 (unten). Der beschriebene Ansatz bildet die Variabilität zwischen Experten ab, da es zu einer systematischen Verunreinigung aller Daten pro Experte kommt. Zur Reduktion des Umfangs der Experimente sollen nicht mehr als zwei unterschiedliche Experten berücksichtigt werden. Der erwähnte Fall

von mehr als zwei Experten wäre jedoch mit der vorgeschlagenen Simulationsmethodik möglich, in dem der Datensatz in mehr als zwei Teile aufgeteilt wird. Umfangreiche Beschreibungen finden sich in dem Zusatzmaterial der im Rahmen dieser Arbeit entstandenen Veröffentlichung [Sch22d].

3.4.3 Automatisierte Inspektion der Variabilität von Experten

Eine Zielstellung dieser Arbeit ist die Quantifizierung der Variabilität von Experten mittels Algorithmen, die eine automatisierte Inspektion ermöglichen sollen. Die entwickelten Methoden werden im Nachfolgenden detailliert erläutert und sind im Rahmen der vorliegenden Arbeit unter [Sch22d, Sch23c] publiziert worden.

Intra-Experten-Variabilität

Abbildung 3.9 illustriert die konzipierte Methodik zur automatisierten Inspektion im Rahmen der Intra-Experten-Variabilität.

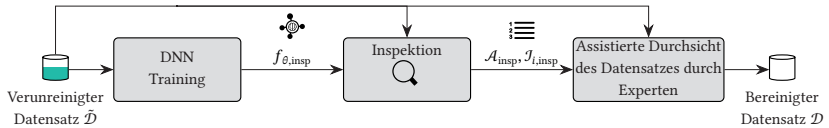


Abbildung 3.9: Automatisierte Inspektion der Intra-Experten-Variabilität. Ein unsicherheitsbewusstes DNN $f_{\theta, \text{insp}}$ wird mittels verunreinigtem Datensatz $\tilde{\mathcal{D}}$ trainiert, um eine automatisierte Inspektion zu ermöglichen. Dabei wird über $\mathcal{A}_{\text{insp}}$ eine geordnete Liste bzw. im Falle der Instanzsegmentierung zusätzlich $\mathcal{J}_{i, \text{insp}}$ zur Bewertung aller Instanzen pro Bild \mathbf{x}_i ausgegeben. Damit können potenziell verunreinigte Annotationen priorisiert durch Experten betrachten werden, um einen bereinigten Datensatz \mathcal{D} zu erzeugen [Sch22d, Sch23c].

Der verunreinigte Datensatz $\tilde{\mathcal{D}}$ wird herangezogen, um die Parameter eines DNN $f_{\theta, \text{insp}}$ im Rahmen eines Trainings zu bestimmen. Es wird dabei angenommen, dass aufgrund des zufälligen Charakters der Intra-Experten-Variabilität verrauschte Annotationen in einem Datensatz nicht dominieren ($\beta < 0.5$ oder $\gamma < 0.5$). Ergo kann sich das DNN auf die Mehrheit der

korrekten Annotationen konzentrieren, um hilfreiche Repräsentationen zu lernen. Damit ist es möglich, die Prädiktion des DNN $\hat{\mathbf{y}}_i$ mit der potenziell fehlerhaften Annotation $\tilde{\mathbf{y}}_i$ zu vergleichen. Allerdings kann ein Vergleich teilweise irreführend sein, falls ein Bild vom DNN fehlerhaft vorhergesagt wird. In diesem Fall entstehen Abweichungen bedingt durch den Fehler des DNN statt durch eine verrauschte Annotation. Infolgedessen muss eine große Diskrepanz zwischen Vorhersage und Annotation nicht unbedingt ein Indikator für Intra-Experten-Variabilität sein. Daher wird der beschriebene Ansatz (Vergleich zwischen Annotation und Vorhersage) erweitert, indem die Unsicherheit des DNN miteinbezogen wird (siehe vorgestellte Ansätze in Abschnitt 2.1.3). Die Inspektion soll sich auf Bilder konzentrieren, die sowohl einen großen Übereinstimmungsfehler von Vorhersage $\hat{\mathbf{y}}_i$ und Annotation $\tilde{\mathbf{y}}_i$ als auch eine geringere Unsicherheit aufweisen, was auf die Verlässlichkeit der Vorhersagen hindeutet.

Details einschließlich aller Berechnungen innerhalb der Algorithmen im Falle von Instanzsegmentierung und semantischer Segmentierung finden sich in Anhang B. Ausgangsgröße der Inspektion ist eine geordnete Liste $\mathcal{A}_{\text{insp}}$, die Experten eine priorisierte Inspektion von Bildern ermöglichen soll, ohne den Datensatz in der Gesamtheit erneut betrachten zu müssen. Im Falle der Instanzsegmentierung wird zudem ein Vorschlag für alle Instanzen pro Bild \mathbf{x}_i generiert, indem die Liste von Instanzen $\mathcal{J}_{i,\text{insp}}$ bestimmt wird, welche die Annotationsqualität pro Instanz bewertet. Diese Vorhersage soll helfen, potenziell falsch-positive oder falsch-negative Instanzen aufzufinden. Damit wird der Experte bei der Generierung eines bereinigten Datensatzes \mathcal{D} unterstützt.

Inter-Experten-Variabilität

Die Inter-Experten-Variabilität kann im naiven Ansatz erkannt werden, wenn mehrere Experten eine Annotation zu einem Bild durchführen. Hierdurch ist ein Vergleich zur Detektion von systematischen Abweichen zwischen Experten möglich, um konsistente Annotationen zu erreichen. Allerdings führt dieses Vorgehen zu zusätzlichem Aufwand und ist in der Praxis oft aus Zeit- und Kostengründen nicht praktikabel.

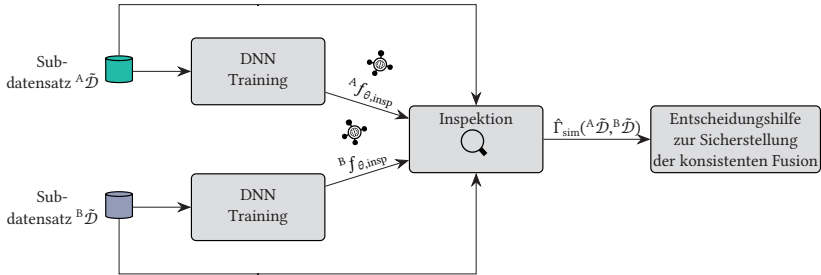


Abbildung 3.10: Automatisierte Inspektion der Inter-Experten-Variabilität. DNNs werden auf den individuellen Datensätzen von Experte A $A\tilde{\mathcal{D}}$ bzw. Experte B $B\tilde{\mathcal{D}}$ trainiert, um im Rahmen der Inspektion einen kreuzweisen Vergleich ohne mehrfache Annotation eines Bildes zu ermöglichen. Es resultiert $\hat{\Gamma}_{sim}(A\tilde{\mathcal{D}}, B\tilde{\mathcal{D}})$ als Metrik für die Konsistenzprüfung zwischen den Experten A und B. Diese Quantifizierung dient zum Erzeugen von Warnungen für Anwender vor der Datensatzfusion, falls potenziell unterschiedliche Annotationsstile gegeben sind [Sch22d].

Deshalb wird die neu entwickelte Pipeline in Abbildung 3.10 vorgestellt, um eine Überprüfung der Inter-Experten-Variabilität zu ermöglichen, wenn Annotationen in Arbeitsteilung¹ generiert worden sind². Es wird im Weiteren angenommen, dass die aufgeteilten Datensätze ähnliche Eigenschaften haben, also zufällig verteilt worden sind. Das Ergebnis der Pipeline ist eine Konsistenzbewertung $\hat{\Gamma}_{sim}(A\tilde{\mathcal{D}}, B\tilde{\mathcal{D}})$, welche die Experten A und B hinsichtlich der Ähnlichkeit des Annotationsstils vergleicht. Damit kann bei der Fusion der Datensätze automatisiert gewarnt werden, wenn deren Zusammenführung vermeintlich Konsistenzprobleme hervorruft. Zu Beginn sollen DNNs $A f_{\theta, insp}/B f_{\theta, insp}$ bezüglich der Datensätze $A\tilde{\mathcal{D}}$ und $B\tilde{\mathcal{D}}$ trainiert werden. Das Ziel ist, den individuellen Annotationsstil zu lernen und damit simulieren zu können. Es wird aus den oben erläuterten Gründen ebenfalls auf unsicherheitsbewusste Netze zurückgegriffen, um mit Fehlern der DNNs umgehen zu

¹ Der Begriff „Arbeitsteilung“ charakterisiert die Eigenschaft im Hinblick auf die annotierten Datensätze, dass keiner der Datensätze von verschiedenen Experten gleichzeitig annotiert worden ist, sondern jeder Annotierer ein Teil des Gesamtdatensatzes annotiert hat. Folglich existiert pro Bild x_i nur eine Annotation \tilde{y}_i .

² Es ist anzumerken, dass mit der vorgeschlagenen Methode zudem eine Analyse bezüglich der Variabilität von Experten möglich ist, wenn pro Bild mehrere Annotationen von verschiedenen Experten vorliegen.

können. Unter Verwendung von ${}^A f_{\theta, \text{insp}} / {}^B f_{\theta, \text{insp}}$ ist ein kreuzweiser Vergleich zwischen den Experten möglich, um deren Ähnlichkeit im Sinne $\hat{\Gamma}_{\text{sim}}({}^A \mathcal{D}, {}^B \mathcal{D})$ vorherzusagen, ohne mehrfache Annotationen pro Bild zu benötigen. Eine ausführliche Beschreibung der Methode inklusive aller Berechnungen innerhalb des Algorithmus ist in Anhang B dargestellt.

3.5 Verarbeitung von Bilddaten im Hochdurchsatz

3.5.1 Überblick

Abbildung 3.11 zeigt die entwickelte Methodik zur automatisierten Bildverarbeitung von HDEs, die im Rahmen dieser Arbeit in [Sch21b] veröffentlicht worden ist. Die Methodik stellt eine Komposition aus den Hauptbestandteilen

- automatisierter Schätzung der Gitterparameter auf Makroebene und
- nachgelagerter elementweisen Einzelbildverarbeitung auf Mikroebene ($f_{\theta, \text{micro}}$) dar,

die im Weiteren detailliert erläutert werden.

3.5.2 Prozessierung Makroebene

3.5.2.1 Pre-Processing

Im Rahmen der Bildvorverarbeitung wird aus dem vorliegenden Rohbild \mathbf{x} das Bild $\check{\mathbf{x}}$ erzeugt. Wenn die Farbinformation zur Detektion der Gitterelemente nicht zwingend erforderlich ist, wird im Falle von RGB-Bildern ($C = 3$) eine Farbraumtransformation zu Mono-Graustufenbildern durchgeführt. Ohne Beschränkung der Allgemeinheit gelte daher im Weiteren $C = 1$. Als Vorteil lassen sich somit flexibel sowohl Farbbilder als auch Graustufenbilder prozessieren. Zudem erfolgt eine Normalisierung des Bildes, um einen geeigneten

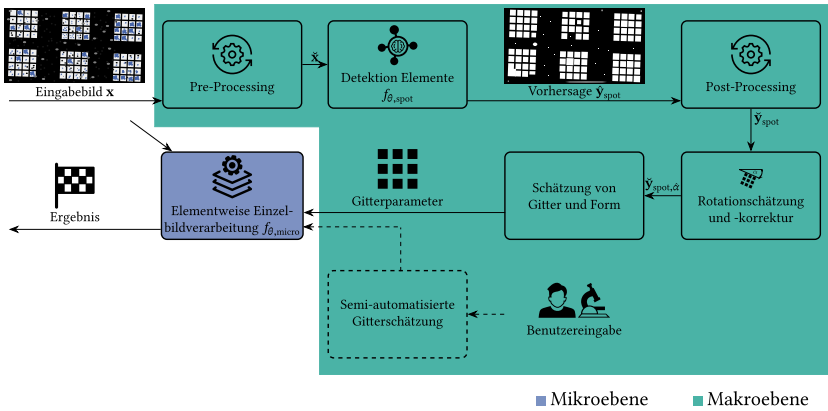


Abbildung 3.11: Gesamtmethodik zur automatisierten Bildverarbeitung von Gitterstrukturen in HD. Zunächst erfolgt die Bestimmung der Gitterparameter auf der Makroebene (■) mittels mehrerer Verarbeitungsschritte. Ein Eingabebild x wird zu \tilde{x} vorverarbeitet („Pre-Processing“), um eine „Detektion der Elemente“ \hat{y}_{spot} durchzuführen, welche zu \check{y}_{spot} nachverarbeitet wird („Post-Processing“). Im Modul „Rotationschätzung und -korrektur“ wird der Drehwinkel α geschätzt und damit die Vorhersage zu $\check{y}_{spot,\alpha}$ korrigiert, wodurch eine „Schätzung von Gitter und Form“ möglich ist. Optional kann eine „semi-automatisierte Gitterschätzung“ ausgewählt werden, die eine Benutzereingabe erfordert (---). Resultate sind in beiden Fällen die eingeführten Gitterparameter (vgl. Abschnitt 3.2). Das Eingabebild x kann analog gedreht werden, um pro Element die „Einzelverarbeitung“ $f_{\theta,micro}$ auf der Mikroebene (■) im identischen Koordinatensystem zu ermöglichen [Sch21b].

Wertebereich für nachfolgende Verarbeitungsschritte zu gewährleisten. Oftmals handelt es sich bei der Bildverarbeitung von gitterförmigen HDEs um hochauflösende Bilder, daher kann eine Reduktion der Bildauflösung Berechnungen beschleunigen.

3.5.2.2 Detektion Elemente

Ein DNN $f_{\theta,spot}$ dient zur Detektion der einzelnen Gitterelemente, indem über $\hat{y}_{spot} = f_{\theta,spot}(\tilde{x}) \in \mathbb{N}^{H \times W}$ pixelweise binär vorhergesagt wird, ob es sich um einen Teil der Gitterelemente handelt. Verglichen mit etablierten Ansätzen, wie Kreis- oder Linienvorhersagen nach Hough, ist der vorgeschlagene Ansatz agnostisch bezüglich der Form der vorliegenden Gitterelemente. Jedoch

kann nicht davon ausgegangen werden, dass die Vorhersage $\hat{\mathbf{y}}_{\text{spot}}$ in der Gesamtheit korrekt ist, da beispielsweise beschädigte oder schlecht belichtete Gitterelemente die Vorhersage erschweren. Eine Nachbearbeitung der Vorhersage $\hat{\mathbf{y}}_{\text{spot}}$ ist hilfreich.

3.5.2.3 Post-Processing

Im Nachbearbeitungsschritt wird eine bereinigte Vorhersage $\check{\mathbf{y}}_{\text{spot}}$ generiert. Morphologische Operatoren unterdrücken dabei Rauschanteile und glätten Formen. Außerdem erfolgt eine flächenbasierte Filterung der einzelnen Elemente zur Entfernung von Ausreißerelementen, indem alle detektierten Elemente in der Vorhersage $\hat{\mathbf{y}}_{\text{spot}}$ gelöscht werden, die sich außerhalb der ermittelten Filtergrenzen befinden. Zur Bestimmung von robusten Filtergrenzen wird die Gesamtverteilung hinsichtlich der Flächeninhalte aller Elemente in $\hat{\mathbf{y}}_{\text{spot}}$ herangezogen, um dann anteilig die obere bzw. untere Schranke der gültigen Flächeninhalte zu bestimmen.

3.5.2.4 Rotationschätzung und -korrektur

Eine direkte Ermittlung des Drehwinkels α aus der bereinigten Vorhersage $\check{\mathbf{y}}_{\text{spot}}$ ist schwierig, da beispielsweise fehlerhafte oder fehlende Detektionen sowie Sub-Gitterstrukturen die Berechnung behindern. Daher wird ein neuer Algorithmus entwickelt, um den Drehwinkel zu schätzen. Parallel erfolgt die Ermittlung des Abstands r zwischen zwei benachbarten Elementen eines Sub-Gitters. Eine ausführliche Beschreibung hierzu findet sich in Anhang A unter Algorithmus 1. Hough-basierte Ansätze zur Schätzung des Drehwinkels werden nicht herangezogen, da einerseits aufgrund der erforderlichen genauen Winkeldiskretisierung kombiniert mit hochauflösenden Bildern hohe Rechenzeiten resultieren. Andererseits erschweren die Fehleranfälligkeit des Verfahrens sowie die eingeschränkte Nutzbarkeit bei unbekanntenen Elementformen eine Anwendung. Diese Problematik tritt z. B. auch bei alternativen Verfahren wie dem Template-Matching auf.

Es soll im Weiteren angenommen werden, dass die Anzahl der benachbarten Elemente eines Sub-Gitters mit gleichen Abständen größer als die Anzahl der Sub-Gitter ist. Daraus folgt, dass mehrheitlich Detektionen ungefähr gleiche Abstände zu benachbarten Detektionen aufweisen und diese ferner senkrecht zueinanderstehen. Diese Annahme ist für gängige biochemische oder biologische Testplattformen, wie bereits in Abbildung 2.2 visualisiert, valide. Abbildung 3.12 (a) illustriert die getroffenen Annahmen.

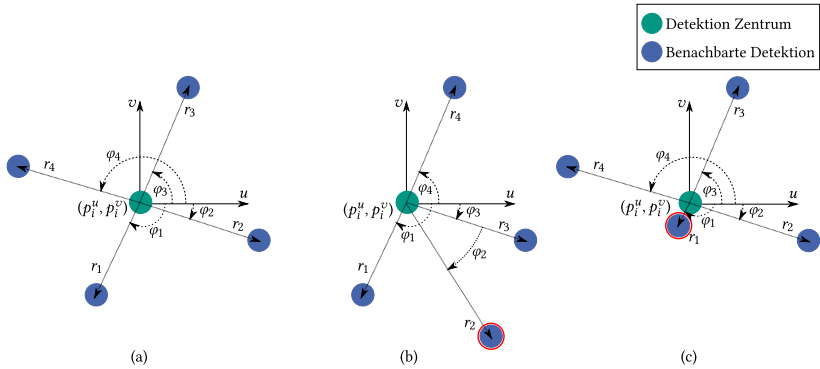


Abbildung 3.12: Rotationschätzung. Die Detektion im Zentrum (p_i^u, p_i^v) ist in einem relativen Polarkoordinatensystem dargestellt, wobei r_j den Betrag und φ_j den Winkel der benachbarten Detektionen j beschreiben. Der Fall (a) zeigt eine gültige Nachbarschaftsbeziehung. Im Gegensatz dazu sind Verletzungen der Nachbarschaften (markiert in rot) bezüglich Winkel (b) oder Betrag (c) gezeigt [Sch21b].

Die pixelweise Information \check{y}_{spot} wird interpretiert und die Mittelpunkte (p_i^u, p_i^v) aller detektierten Elemente bestimmt. Im Folgenden bezeichne der Index i die Nummer einer Zentrumsdetektion, wohingegen der Index j sich auf die Nummern der zugehörigen Nachbarn der Nummer i bezieht. Mittels eines k-Nearest-Neighbor (KNN)-Algorithmus werden die vier nächsten Nachbarn des vermeintlichen Elements ermittelt. Diese Nachbarn werden im Anschluss in ein relatives Polarkoordinatensystem übertragen, dessen Ursprung die Detektion i bildet. Der Betrag r_j und der Winkel φ_j charakterisieren die Nachbarschaftsbeziehung.

Mithilfe der Repräsentation in Polarkoordinaten wird eine anschauliche Zulässigkeitskontrolle ermöglicht, indem die Kriterien (i) eine Winkeldifferenz von ungefähr $\frac{\pi}{2}$ (senkrechte Anordnung) sowie (ii) Ähnlichkeit bezüglich der Beträge aller Nachbarn erfüllt sein müssen. Die zulässigen Toleranzen von Abweichungen in Betrag und Winkel sind dabei einstellbare Parameter des Algorithmus. Eine Toleranz von 10 % bezogen auf mittleren Winkel bzw. Abstand stellt hierbei einen sinnvollen empirisch bestimmten Startwert dar, um binär zwischen zulässiger und unzulässiger Nachbarschaft zu unterscheiden. Abbildung 3.12 (b) stellt eine unzulässige Nachbarschaft dar, bei welcher das Winkelkriterium (i) verletzt ist. Im Gegensatz dazu stellt Abbildung 3.12 (c) eine Missachtung des Kriteriums von ähnlichen Distanzen (ii) dar.

Lediglich zulässige Nachbarschaften werden für die weitere Berechnung in Betracht gezogen. Der Drehwinkel α und der Elementabstand r werden nach den vorgegebenen Zwischenschritten (vgl. Algorithmus 1 in Anhang A) über robuste Parameterschätzverfahren ermittelt, indem der Median aus allen Messungen von α bzw. r bestimmt wird.

Um die Gitterschätzung zu vereinfachen, werden alle Detektionen um den geschätzten Winkel $\hat{\alpha}$ korrigiert, wodurch $\check{\mathbf{y}}_{\text{spot},\hat{\alpha}}$ resultiert. Kompensation bedeutet in diesem Zusammenhang eine Drehung um den negierten Drehwinkel $\hat{\alpha}$, wodurch ein Gitter folgt, das parallel zum (u,v) -Bildkoordinatensystem orientiert ist.

3.5.2.5 Schätzung von Gitter und Form

Das Ziel der Gitterschätzung besteht darin, die Zeilenanzahl I , die Spaltenanzahl J und alle Orte von Elementen (p_i^u, p_i^v) inklusive des Formparameters d zu bestimmen. Eine umfassende Darstellung des Vorgehens in Form von Pseudocode findet sich in Anhang A (vgl. Algorithmus 2). Zunächst wird die um den Verdrehungswinkel korrigierte pixelweise Elementinformation $\check{\mathbf{y}}_{\text{spot},\hat{\alpha}}$ in Listen \mathcal{U} und \mathcal{V} überführt, welche lediglich u - bzw. v -Koordinaten der Detektionsmittelpunkte beinhalten, um somit Zeilen und Spalten unabhängig voneinander bestimmen zu können.

Die Detektionen werden hinsichtlich \mathcal{U} bzw. \mathcal{V} über ein Clusterverfahren in Gruppen (Zugehörigkeiten zu horizontalen bzw. vertikalen Gitterlinien) getrennt, wodurch das Verfahren auch eine Schätzung bei Sub-Gittern ermöglicht. Dabei muss das Clusterverfahren in der Lage sein, die Anzahl der Cluster, welche der Zeilen- (I) bzw. Spaltenanzahl (J) entspricht, automatisiert vorherzusagen. Jedoch können in $\check{\mathbf{y}}_{\text{spot},\hat{\alpha}}$ immer noch verrauschte oder falsche Elemente enthalten sein, wodurch die Anforderung von Robustheit gegenüber Rauschen und Ausreißern für das Verfahren wichtig ist. Der im vorherigen Abschnitt geschätzte mittlere Gitterabstand \hat{r} kann verwendet werden, um zu entscheiden, ob ein neues Cluster gewählt werden muss oder nicht. Es resultieren die Listen \mathcal{C}_u und \mathcal{C}_v , die entsprechende Cluster-Nummern beinhalten, womit jeder Detektion (p_i^u, p_i^v) ein Zeilen- bzw. Spaltencluster zugewiesen respektive als Ausreißer markiert wird.

Zur genauen Bestimmung der Elementpositionen werden Linien für jedes horizontale und vertikale Cluster getrennt mithilfe eines Parameterschätzungsansatzes bestimmt. Alle als Ausreißer markierten Detektionen (beschrieben über wenige Nachbarn in definiertem Abstand) werden nicht berücksichtigt. Mittels geschätzten Linienparametern (Steigung, Verschiebung) können vertikale und horizontale Linien in ein lineares Gleichungssystem überführt werden, um deren Schnittpunkte zu ermitteln. Damit folgen die finalen Mittelpunkte der Gitterelemente. Das vorgeschlagene Vorgehen ermöglicht die direkte Erkennung von Sub-Gittern, da in der Methode keine Annahme über gleiche Punktabstände getroffen worden ist.

Der Formparameter d wird robust geschätzt, indem der Median von den nach der Filterung verbleibenden Detektionen in $\check{\mathbf{y}}_{\text{spot},\hat{\alpha}}$ ermittelt wird.

Außerdem wird die Fläche als Unterscheidungsmerkmal für die Formschätzung herangezogen. Der Flächeninhalt jedes Elements A_i wird mit einer entsprechenden quadratischen oder kreisförmigen Fläche ($A_{\text{square},i}/A_{\text{circle},i}$) verglichen. Es folgt als Klassifikationsregel

$$\hat{y}_{\text{shape},i} = \begin{cases} \text{circle,} & \left\| A_i - A_{\text{circle},i} \right\| < \left\| A_i - A_{\text{square},i} \right\|, \\ \text{square,} & \text{sonst} \end{cases}, \quad (3.8)$$

um die Form des Elements i vorherzusagen, wobei die endgültige Form \hat{y}_{shape} der am häufigsten vorkommenden Vorhersagen aller Elemente entspricht. Um komplexere Formen zu schätzen, kann die Klassifikationsregel durch einen elaborierteren Ansatz (z. B. Pattern-Matching oder DNN zur Klassifikation) ersetzt werden.

3.5.2.6 Semi-automatisierte Gitterschätzung

Bei Problemen im Rahmen der automatisierten Bestimmung der Gitterparameter, z. B. bei fehlenden Trainingsdaten oder schwierigen Aufnahmebedingungen eines Bildes, sieht die vorgeschlagene Methode eine Möglichkeit der semi-automatisierten Gitterschätzung vor. Zunächst ist es erforderlich, dass der Nutzer die Zeilen-/Spaltenanzahl sowie die Form der Elemente angibt. Im Anschluss können durch die Auswahl der Ecken des Gitters der Drehwinkel und die Lage der Mittelpunkte über trigonometrische Zusammenhänge automatisiert bestimmt werden (siehe [Sch21b]).

3.5.2.7 Robustheit und Anwendungsgrenzen

Die Qualität der Vorhersage des DNN ist der Haupteinflussfaktor für eine erfolgreiche Schätzung von Gittern. Um die Robustheit und Grenzen der automatisierten Gittererkennung zu untersuchen, werden die Vorhersagen \hat{y} im Rahmen eines Testverfahrens bewusst simulativ verändert. Wie in Abbildung 3.13 angedeutet, soll zwischen den synthetischen Veränderungen/Verfälschungen

- Rauschen (Hinzufügen von zufälligem Salt- and Pepper-Rauschen in die Vorhersage),
- fehlende Segmente (Elemente werden zufällig gelöscht),
- zusätzliche Segmente (Segmente werden nach dem Zufallsprinzip hinzugefügt),
- Rotation (Segmentierungsmaske wird gedreht) und
- Sub-Gitterstrukturen (Zeilen oder Spalten werden gelöscht)

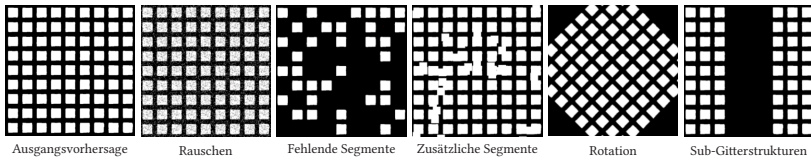


Abbildung 3.13: Robustheit und Anwendungsgrenzen. Verschiedene Veränderungen der Ausgangsvorhersage sind im Überblick illustriert [Sch21b].

differenziert werden. Die Qualität der automatisierten Gitterparameterschätzung wird anschließend über die Gütekriterien $Q_{\Delta p}$ bzw. $Q_{\Delta d}$ (vgl. Abschnitt 3.2) analysiert.

3.5.3 Prozessierung Mikroebene

Zu Beginn soll auf den prinzipiellen Aufbau der vorgeschlagenen Prozessierung in der Mikroebene eingegangen werden (vgl. Abbildung 3.14).

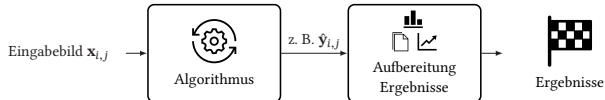


Abbildung 3.14: Prozessierung Mikroebene. Ein Eingabebild $x_{i,j}$ wird durch einen Algorithmus prozessiert. Dessen Ausgaben (z. B. eine Vorhersage $\hat{y}_{i,j}$) werden im Nachgang zu Ergebnissen aufbereitet, die schnell von Anwendern interpretiert oder weiterverarbeitet werden können [Sch23b].

Ein Eingabebild $x_{i,j}$ (z. B. aus der Gitterextraktion) wird durch einen auf die jeweilige Problemstellung angepassten Algorithmus verarbeitet. Es folgen die Ausgaben des Algorithmus, wie beispielsweise eine Vorhersage $\hat{y}_{i,j}$, die anschließend zu Ergebnissen aufbereitet werden. Hier sind beispielhaft die Erstellung von Visualisierungen oder der Export in Textdateien zu nennen, welche Anwendern die Interpretation oder eine Weiterverarbeitung der Ergebnisse erleichtert.

Nachfolgend wird exemplarisch auf verschiedene Algorithmen im Bereich der Mikroebene eingegangen, welche für die Auswertung der Elemente $\mathbf{x}_{i,j}$ genutzt werden können.

Seed Detektion In vielen Fällen ist es für die Auswertung von HDEs ausreichend, Objekte wie Zellen in Bildern zu zählen (=Seed Detektion), anstelle eine vollständige Segmentierung oder Objektdetektion mit Ausdehnungsbestimmung durchzuführen. Insbesondere in Bezug auf den Annotationsaufwand ergibt sich für Nutzer ein Vorteil, da lediglich Mittelpunkte statt Konturen zu markieren sind. Für trivialere Probleme werden Blob [Lin98] oder das Otsu-Schwelwertverfahren als Algorithmen zur Detektion genutzt. Im Falle komplexerer Probleme kann auf ein lernendes, DL-basiertes Verfahren wie BeadNet [Sch20b] zurückgegriffen werden. Hier wird ein DNN verwendet, um die Zentroide von Objekten vorherzusagen.

Segmentierung Bildsegmentierungen, sowohl semantisch als auch von Instanzen, sind in vielen HDEs für deren Auswertung notwendig. Allerdings unterscheidet sich die Art der Bilder zwischen einzelnen Problemstellungen stark, wodurch klassische Bildverarbeitungsalgorithmen aufwändig und problembezogen zu entwickeln sind. Zudem erbringen etablierte Verfahren trotz aufwändiger Entwicklung oftmals nicht die erforderliche Güte. Der hier entwickelte Ansatz zur Lösung von Segmentierungsproblemen findet sich in Abbildung 3.15.

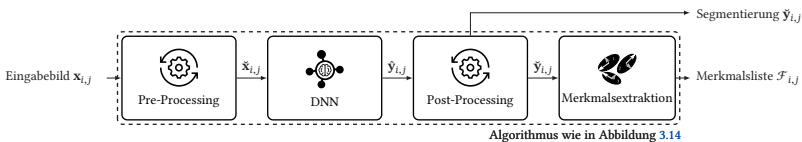


Abbildung 3.15: Analysepipeline zur Segmentierung in HDEs. Das Eingabebild $\mathbf{x}_{i,j}$ des Elements i,j wird zu $\tilde{\mathbf{x}}_{i,j}$ vorverarbeitet (vgl. „Pre-Processing“), um eine Verarbeitung mittels DNN zu erleichtern. Es folgt die Rohvorhersage $\hat{\mathbf{y}}_{i,j}$ bzw. nach „Post-Processing“ die finale Segmentierungsvorhersage $\tilde{\mathbf{y}}_{i,j}$, aus der durch Merkmalsextraktion eine Merkmalsliste $\mathcal{F}_{i,j}$ generiert wird [Sch23b].

Ausgehend vom Eingabebild $\mathbf{x}_{i,j}$ kann im Rahmen des „Pre-Processings“ durch eine Normalisierung die Eingaberepräsentation des Bilds angepasst werden. Somit wird ein einheitlicher Wertebereich am Eingang des DNN sichergestellt, was die Robustheit der Vorhersagen unterstützt. Je nach Anforderung kann die Anpassung der Bildgröße zu einer Verbesserung hinsichtlich der Rechenzeit führen. Das Eingabebild $\mathbf{x}_{i,j}$ soll nach dem „Pre-Processing“ über $\check{\mathbf{x}}_{i,j}$ notiert werden. Mittels DNN kann dann die Rohvorhersage $\hat{\mathbf{y}}_{i,j}$ erzeugt werden, die nach dem „Post-Processing“ in die finale Vorhersage $\check{\mathbf{y}}_{i,j}$ mündet. Beispielsweise kann dies bei semantischer Segmentierung das pixelweise Zusammenfassen des Ausgabesensors zur Klasse mit stärkstem Voting oder im Falle einer Instanzsegmentierung die Bestimmung von Objekten aus euklidischen Distanzkarten sein. Jedoch stellen die Resultate der Segmentierung oftmals lediglich einen Zwischenschritt für die Auswertung von HDEs dar. Nutzer sind an Merkmalen von Segmenten, wie beispielsweise dem Flächeninhalt, der Länge der Hauptachsen oder einem Maß für die Exzentrizität interessiert. Infolgedessen wird allgemein eine Merkmalsliste $\mathcal{F}_{i,j}$ mit den genannten Einträgen als Zusatz generiert.

Mehrdimensionale Instanzanalyse Eine mehrdimensionale Instanzanalyse, also die Auswertung mehrerer überlagerter Bildkanäle hinsichtlich der gegebenen Instanzen, stellt oft eine wichtige Untersuchung dar. Instanzen können beispielsweise Zellen sein. Die Bestimmung der Zellviabilität ist als ein Anwendungsfall der mehrdimensionalen Instanzanalyse zu nennen. Dort gilt es, den Anteil von lebenden Zellen im Verhältnis zu allen vorliegenden Zellen über eine Auswertung verschiedener Farbbildkanäle zu quantifizieren. Die entwickelte Methode zur automatisierten mehrdimensionalen Instanzanalyse findet sich in Abbildung 3.16.

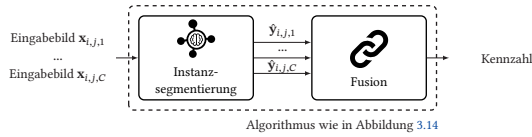


Abbildung 3.16: Analysepipeline für die mehrdimensionale Instanzanalyse. Ausgangspunkt bildet die Pipeline zur Instanzsegmentierung (vgl. Abbildung 3.15) von den Eingabebildern $\mathbf{x}_{i,j,1}, \dots, \mathbf{x}_{i,j,C}$ mit unterschiedlichen Einfärbungen, wodurch die Prädiktionen $\hat{\mathbf{y}}_{i,j,1}, \dots, \hat{\mathbf{y}}_{i,j,C}$ folgen. Mittels eines Fusionsschritts werden die Vorhersagen geeignet kombiniert, um auf die jeweilige Kennzahl schließen zu können [Sch23b].

Eingangsgrößen stellen die Bilder $\mathbf{x}_{i,j,1}, \dots, \mathbf{x}_{i,j,C}$ mit unterschiedlichen Einfärbungen (engl. stainings) dar¹. Die bereits vorgestellte Segmentierungspipeline (vgl. Abbildung 3.15) wird hierzu verwendet, um Instanzen separat für alle C Einfärbungen zu detektieren. Es folgen damit die Vorhersagen $\hat{\mathbf{y}}_{i,j,1}, \dots, \hat{\mathbf{y}}_{i,j,C}$ hinsichtlich der sichtbaren Instanzen pro Färbung. Im Fusionsschritt werden diese dann kombiniert. Um einen Vergleich der Instanzen zwischen den Färbungen zu ermöglichen, soll der KNN-Algorithmus genutzt werden, wodurch ein Assoziieren der Instanzen über die verschiedenen Kanäle erfolgt. Im letzten Schritt kann mittels der Auswertung von logischen Ausdrücken bezüglich der bestimmten Detektionen eine Kennzahl wie die Zellviabilität errechnet werden.

Im Falle der Kombination der Einfärbesubstanzen Hoechst (führender Kanal), Calcein (Ergänzung zum führenden Kanal) und Propidium Iodide (PI) (Negativ-Kanal) sieht das Vorgehen wie folgt aus: Die Gesamtanzahl der Instanzen ergibt sich aus den Objekten in der Hoechst-Färbung. Kriterien für lebendige Zellen sind die zusätzliche Sichtbarkeit der Zelle in Calcein und eine gleichzeitig ausbleibende Detektion in PI. Weitere erläuternde Details zu Zellviabilität und den zugehörigen Einfärbungen finden sich z. B. in der Arbeit von Komarnicka et al. [Kom21].

¹ Alle Bilder $\mathbf{x}_{i,j,1}, \dots, \mathbf{x}_{i,j,C}$ sind zueinander ausgerichtet und befinden sich damit im identischen (u,v) -Koordinatensystem.

Farbanalyse In vielen HDEs kann mittels einer Farbanalyse von Hellfeldaufnahmen auf die Ergebnisse des Experiments geschlossen werden. Als Beispielanwendung ist die Ermittlung des potentia Hydrogenii (pH)-Werts zu nennen. Ziel im Rahmen der Bildverarbeitung ist die automatisierte Berechnung einer quantitativen Kennzahl $y_{i,j}$ pro Bildelement $\mathbf{x}_{i,j}$, die möglichst monoton mit der zu interessierenden Ausgangsgröße, wie dem pH-Wert, korreliert. Über eine Kalibrierungsmessung können die Größen damit ineinander umgerechnet werden. Die Analysepipeline sieht im ersten Schritt vor, den Median-RGB-Wert des Elements zu bestimmen, um die Robustheit des Verfahrens zu steigern. Anschließend werden über eine abstrakte Transformationsfunktion $f_{\text{trafo}} : \mathbb{R}^3 \mapsto \mathbb{R}$ die RGB-Werte in einen Skalar $y_{i,j}$ umgerechnet, der zur Farbquantifizierung des Elements $\mathbf{x}_{i,j}$ genutzt wird. Die Transformationsfunktionen sind stark vom jeweiligen Experiment und dem dort vorliegenden Farbspektrum abhängig. Als Beispiele für f_{trafo} sind Farbraumtransformationen von RGB zu Hue-Saturation-Value (zu Deutsch: Farbwert, Farbsättigung, Helligkeit) (HSV) oder individuelle Linearkombinationen der RGB-Werte zu nennen.

Intensitätsanalyse Eine Intensitätsanalyse ist ein gängiges Verfahren zur Auswertung von HDEs. Ähnlich zur Vorgehensweise bei der Farbanalyse wird im Rahmen dieser Auswertemethode eine skalare quantitative Kennzahl $y_{i,j} \in [0,1]$ pro Element $\mathbf{x}_{i,j}$ berechnet, wobei die Kennzahl in diesem Fall die Stärke der Bildintensität charakterisiert. Der Anwender kann je nach Zielstellung wählen, ob der Mittelwert oder Median des Signals zu berechnen ist. Falls die Intensität beispielsweise stark über das Gesamtelement schwankt und sich dieser Verlauf im Endresultat widerspiegeln soll, wird dies durch den Mittelwert des Elements sinnvoller quantifiziert. Bei homogenen Intensitätsverläufen innerhalb der Elemente können allerdings durch die Nutzung des Medians einzelne Ausreißer erfolgreich unterdrückt werden.

Zusammenfassend stellt die Arbeit damit als Neuentwicklung ein vielfältiges Portfolio zur Auswertung der HDEs in der Mikroebene bereit, um heterogene

Bildverarbeitungsprobleme automatisiert zu lösen. Dabei basieren die konzipierten Algorithmen meist auf DL-Ansätzen, um komplexen und heterogenen Bildverarbeitungsaufgaben gerecht zu werden.

3.6 Effizientes Lernen

3.6.1 Überblick

Die DL-Verfahren zur Auswertung der Bilddaten im HD benötigen im Allgemeinen annotierte Trainingsdatensätze für überwachte Lernverfahren (vgl. Abschnitt 2.1.2). Wie in Kapitel 2 aufgezeigt, gibt es eine Vielzahl an Methoden mit dem Ziel, den Bedarf an Annotationen zu reduzieren bzw. diese möglichst effizient zu nutzen.

Allerdings ist für Anwender a priori nicht klar, welche Methoden am erfolgversprechendsten sind oder ob diese überhaupt für biomedizinische Daten anwendbar sind. Ein Vergleich auf Basis einer identischen Softwareimplementierung unter gleichen Rahmenbedingungen ist nicht verfügbar. Zudem ist die Anwendung von effizientem Lernen primär nur für Klassifikationen untersucht worden, wodurch Analysen im Hinblick auf semantischer Segmentierung und Instanzsegmentierung fehlen. Ferner mangelt es an der Berücksichtigung von erweiterten Metriken, die für den Anwender relevant sind, da meist lediglich die Steigerung der Genauigkeit eines DNN betrachtet wird.

Abbildung 3.17 zeigt das entwickelte Konzept zur Analyse der Methoden aus dem Bereich „effizientes Lernen“. Dieses stützt sich auf die im Rahmen der vorliegenden Arbeit entstandenen Veröffentlichungen [Kep22, Mol23a, Mol23b, Ret21, Ret23, Rod23]. Es sollen die Übergruppen TL, DAL, Self-Supervised Learning (SSL) und Semi-Supervised Learning (SemSL) mit verschiedenen ausgewählten Methoden des aktuellen Stands der Technik analysiert werden. Die ausgiebigen vergleichenden Untersuchungen im Rahmen dieser Arbeit beziehen sich erstmalig auf Datensätze der Problemklassen „semantische Segmentierung“ und „Instanzsegmentierung“, da hierbei Annotationen besonders aufwändig zu erzeugen sind.

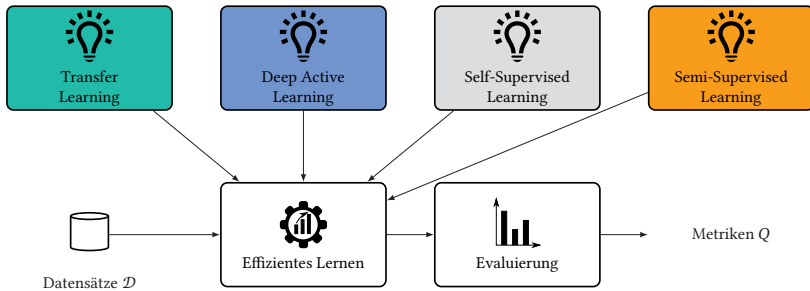


Abbildung 3.17: Methodenvergleich bezüglich Verfahren zum effizienten Lernen. „Effizientes Lernen“ bzw. dessen Anwendbarkeit soll anhand diverser Methoden aus den Übergruppen TL, DAL, SSL und SemSL mittels verschiedener Datensätze \mathcal{D} untersucht werden. Als Resultat der Evaluierung wird für Nutzer eine Zusammenstellung aus umfassenden quantitativen und qualitativen Metriken Q generiert.

Im Rahmen der Evaluierung werden die in Abschnitt 3.2 eingeführten quantitativen und qualitativen Metriken Q genutzt, um für Nutzer eine umfassende Anwendungsempfehlung zu generieren. Es werden verschiedene Annotationsraten $\rho \in \left\{ \frac{2^s}{100} \mid s = 1, \dots, 6 \right\}$ betrachtet. Zudem wird dabei stets ein U-Net [Ron15] als DNN verwendet, um die Rahmenbedingungen jeweils für alle Methoden identisch zu gestalten. Ferner werden alle Ansätze im gleichen Programmgerüst unter konformen Einstellungen getestet, um optimale Vergleichbarkeit zu schaffen.

3.6.2 Transfer Learning

Wie in Abschnitt 2.4.4 erwähnt, zielt TL darauf ab, Wissen aus gelösten¹ benachbarten Problemen auf neue Probleme zu übertragen. Allerdings stellt die automatisierte Auswahl eines passenden, benachbarten, bereits gelösten Problems eine Herausforderung dar. Abbildung 3.18 zeigt die entwickelte Methodik zur automatisierten Auswahl ähnlicher Datensätze für einen bisher nicht

¹ Die Eigenschaft „gelöst“ beschreibt hierbei das Vorliegen von Datensätzen mit Annotationen in ausreichendem Umfang, um ein entsprechendes DNN zur Lösung der Aufgabe trainieren zu können.

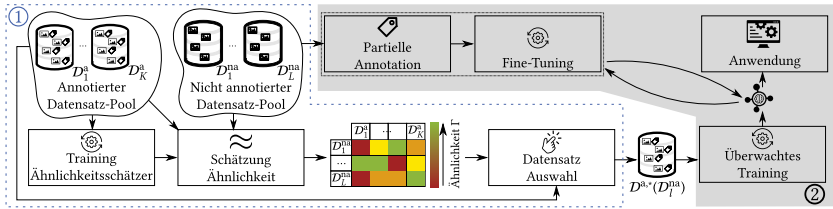


Abbildung 3.18: Pipeline zur automatisierten Auswahl ähnlicher Datensätze für TL. Im ersten Schritt ① wird mittels der annotierten Datensätze ein Modell zur Schätzung der Ähnlichkeit Γ zwischen nicht annotierten Datensätzen und bekannten annotierten Datensätzen bestimmt. Damit kann nach Auswahl des optimalen Datensatzes $\mathcal{D}^{a,*}$ im zweiten Schritt ② ein überwachtes Training beginnen, um ein DNN zu ermitteln, das direkt in der Anwendung zur Lösung der nicht annotierten Datensätze nutzbar ist. Optional kann eine partielle Annotation des unbekannt Datensatzes durchgeführt und diese für ein Fine-Tuning des Modells genutzt werden [Mol23a, Mol23b, Rod23]. © 2023 IEEE.

gelösten Datensatz. Die nachfolgenden Erläuterungen basieren auf im Rahmen der Arbeit entstandenen Veröffentlichungen [Mol23a, Mol23b, Rod23].

Formal gilt es den optimalen Datensatz $\mathcal{D}^{a,*}(\mathcal{D}_l^{na})$ aus dem Pool der $K \in \mathbb{N}$ annotierten Datensätze $\mathcal{D}_1^a, \dots, \mathcal{D}_K^a$ für einen nicht annotierten Datensatz \mathcal{D}_l^{na} aus dem Pool der $L \in \mathbb{N}$ nicht annotierten Datensätze $\mathcal{D}_1^{na}, \dots, \mathcal{D}_L^{na}$ zu finden. Dazu wird mittels eines Trainings ein Modell zur Bestimmung der Ähnlichkeit von Datensätzen $\Gamma \in [0,1]$ ($\Gamma \propto$ Ähnlichkeit zweier Datensätze) ermittelt.

Als Methoden für den Vergleich der Datensätze sind *unsicherheitsbewusste DNNs*, *DL-Ansätze zur Beurteilung der Bildinhalte* oder *Kombinationen* von diesen vorgesehen. Beim Ansatz über *unsicherheitsbewusste DNNs* ist die Ähnlichkeit von Datensätzen umgekehrt proportional zur Unsicherheit von Prädiktionen eines Modells. Im Rahmen der *DL-Ansätze zur Beurteilung der Bildinhalte* wird ein latenter Merkmalsraum erzeugt. Damit kann die Ähnlichkeit über die Distanzen von Elementen eines Datensatzes zu den Elementen anderer Datensätze berechnet werden. Da das Training unüberwacht oder selbstüberwacht erfolgt (siehe Anhang C), ist der Vergleich der Bilder auf die jeweilige Domäne zugeschnitten, ohne die Notwendigkeit von Annotationen, und muss nicht auf einen unveränderlichen Merkmalsextraktor zurückgreifen. Ferner muss im Rahmen des Ansatzes über latente Bildmerkmale keine

Annahme im Hinblick auf die Art der zu lösenden Aufgabe gemacht werden. Zudem können beide Methoden miteinander kombiniert werden.

Alle Details hinsichtlich der entwickelten Algorithmen zur Umsetzung der genannten methodischen Ideen sind in Anhang C ausführlich erläutert. Ein Vorteil des entwickelten Verfahrens ist, beliebige ähnliche, aber bereits annotierte, Datensätze aus dem Internet als Trainingsdaten zu nutzen, was insbesondere durch die steigende Anzahl an online verfügbaren Datensätzen [Noy20] interessant ist.

Unter Verwendung des als am ähnlichsten bestimmten Datensatzes $\mathcal{D}^{a,*}(\mathcal{D}_l^{na})$ sieht die entwickelte Methode zum einen vor, ein Modell mittels überwachtem Training zu bestimmen, das direkt für die Lösung der nicht annotierten Datensätze \mathcal{D}_l^{na} in der Anwendung nutzbar ist. Zum anderen kann eine partielle Annotation des unbekanntes Datensatzes durchgeführt werden, um unter der Verwendung der bestimmten Annotationen ein Fine-Tuning des trainierten Modells durchzuführen. Es werden dabei folgende Strategien zum Fine-Tuning untersucht, angelehnt an die Veröffentlichungen in [Cho20, Liu22]:

- **ALR** - Verwendung der ursprünglich alten Lernrate,
- **GVLRL** - global verringerte Lernrate hinsichtlich aller DNN-Schichten, d. h. Parameter sind gesamtheitlich weniger anpassbar,
- **LVLRL** - linear verringerte Lernrate hinsichtlich der DNN-Schichten, d. h. tiefere Schichten sind stärker anpassbar oder
- **Mix** - ein kombiniertes Training als Mischung aus ähnlichen Bildern der Ursprungsdomäne im Vergleich zur Zieldomäne, die über den Ähnlichkeitsschätzer ausgewählt werden, sowie Bildern der Zieldomäne.

3.6.3 Deep Active Learning

Die weiteren Betrachtungen im Kontext DAL sollen sich auf pool-basierte Sampling-Strategien beziehen. Das Ziel ist es zu untersuchen, ob elaborierte Ansätze zur Auswahl von Bildern während der Annotation einer zufälligen Auswahl überlegen sind.

Einerseits soll die Auswahl von Bildern basierend auf der Unsicherheit von DNNs beleuchtet werden (siehe Ansätze in [Gal17, Sai21]). Andererseits werden Verfahren betrachtet, die auf eine möglichst große Heterogenität der annotierten Bilder Wert legen (vgl. Core Sets [Sen18]). Zur Merkmalsextraktion wird hierbei ein Autoencoder (AE), ein auf ImageNet vortrainiertes ResNet sowie die latenten Merkmale des zur Lösung der eigentlichen Aufgabe trainierten U-Nets verwendet. Eine PCA wird zum Erzeugen eines nieder-dimensionalen Merkmalsraums mit sinnvollen Distanzen verwendet. Ferner wird im Rahmen der Experimente untersucht, inwieweit ein fortgesetztes Training auf Basis des bisher verfügbaren Modells oder ein kompletter neuer Trainingsdurchlauf mit beginnender Zufallsinitialisierung bei steigender Annotationsrate erfolgsversprechender ist.

3.6.4 Self-Supervised Learning

Für die Untersuchungen von SSL sollen die Methoden Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [Che20], BarlowTwins [Zbo21], Bootstrap Your Own Latent (BYOL) [Gri20], DenseCL [Wan21a] und MoCo [He20] untersucht werden. Alle Methoden haben gemein, dass als Resultat des SSL-Verfahrens ein Encoder erstellt wird, der vielversprechende Bildmerkmale generieren soll. Ferner wird ein auf ImageNet vortrainierter Encoder in den Vergleich mitaufgenommen.

Da als Zielaufgaben (=Downstream-Tasks) die semantische Segmentierung bzw. Instanzsegmentierung untersucht werden sollen, stellt der Encoder allerdings nur einen Teil des Zielnetzes dar. Im Folgeschritt sehen die Methoden vor, dass der Encoder in das eigentliche DNN zur Segmentierung integriert wird, welches dann mit unterschiedlichem Annotationsumfang trainiert wird. Es soll untersucht werden, ob eine Integration des über SSL vortrainierten Encoders einem zufällig initialisierten Encoder überlegen ist. Dabei gilt es des Weiteren zu klären, ob ein Einfrieren der Encoder-Parameter oder das Erlauben einer weiteren Anpassung während des eigentlichen Trainings der Segmentierungsaufgabe zielführender ist.

3.6.5 Semi-Supervised Learning

Im Rahmen der Analysen zu SemSL soll der Benefit untersucht werden, im Rahmen des Trainings annotierte Bilder in Kombination mit nicht annotierten Bildern zu verwenden. Benchmark ist ein klassischer überwachter Trainingsdurchlauf, der lediglich annotierte Bilder für die Optimierung der Parameter des DNN nutzt.

Die Untersuchungen beschränken sich dabei auf Ansätze aus den Kategorien „Konsistenzregularisierung“ und „Selbsttraining“, welche mehrheitlich dem Stand der Technik entsprechen. Zum einen werden über MeanTeacher [Tar17] ein klassischer Konsistenzregularisierungs- bzw. über ST [Yan22] ein Selbsttraining-Ansatz untersucht. Andererseits werden die Ansätze FixMatch [Soh20] und ST++ [Yan22] beleuchtet, die beide Kategorien kombinieren und zusätzlich weitere Optimierungen wie Curriculum Learning beinhalten. Des Weiteren soll der Cross Teacher Training (CCT)-Ansatz [Xia22] in den Vergleich aufgenommen werden, der Konsistenzregularisierung nicht wie gewöhnlich auf Bildern, sondern bezüglich latenter Merkmale durchführt.

3.7 Architekturkonzeption für interdisziplinäre Datenanalyseprojekte

Die in Abschnitt 3.1 vorgestellten Methoden sind per se für Domänenexperten nicht nutzbar. Es fehlt häufig an leistungsfähiger Hardware wie Graphics Processing Units (GPUs), die essenziell für DL im Bereich Bildverarbeitung ist. Die Nutzung von HPC-Cluster-Systemen ist für Anwender erschwert, da ein hohes Maß an Komplexität zur Bedienung vorherrscht und große Hürden zum Erlangen eines Zugangs bestehen. Ferner existiert häufig keine Software inklusive GUIs bzw. deren Installation stellt eine Barriere für Nutzer dar, da beispielsweise entsprechende Systemvoraussetzungen notwendig sind. Zudem existieren keine Lösungen für den einfachen Datenaustausch bzw. die Aktualisierung von Software, um Weiterentwicklungen von Programmpaketen

erfolgreich an Nutzer verteilen zu können. Ferner fehlt Anwendern mitunter die Motivation und Zeit, sich in die Details von fachfremden Themen wie DL oder HPC zu vertiefen.

Abbildung 3.19 zeigt das entwickelte Architekturkonzept für interdisziplinäre Datenanalyseprojekte und basiert auf den im Rahmen der Arbeit entstandenen Veröffentlichungen [Neu22, Sch21b, Sch22c, Sch22e].

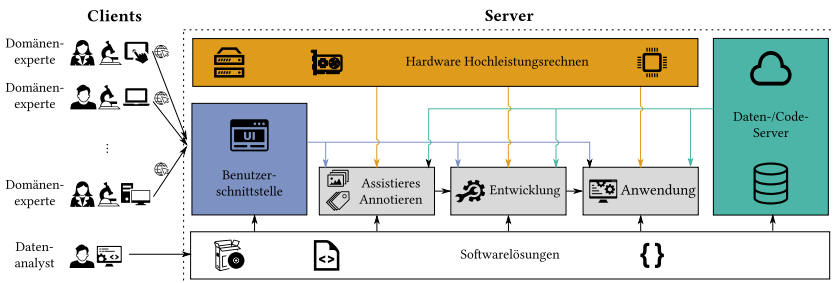


Abbildung 3.19: Überblick der Architekturkonzeption für interdisziplinäre Datenanalyseprojekte. Domänenexperten können über beliebige Endgeräte (Clients) mittels einer Netzwerkschnittstelle auf das gekapselte System (---, Server) zugreifen. Dort werden Benutzerschnittstellen für das assistierte Annotieren, die Entwicklung von Analysepipelines sowie deren Anwendung bereitgestellt. Hierfür sind Verbindungen zu Daten- und Code-Servern sowie zu Hardware für Hochleistungsrechnen vorgesehen. Die Aufgabe des Datenanalysten besteht in der Bereitstellung von den entsprechenden Softwarelösungen [Sch22c, Sch22e].

Das entwickelte Konzept orientiert sich an dem klassischen Client-Server-Modell, also dass Clients Dienste in Anspruch nehmen, die von einem Server zur Verfügung gestellt werden [Ben15]. Endgeräte von Domänenexperten müssen lediglich über eine Software zum Aufbau einer Remote-Desktop-Verbindung verfügen, die betriebssystemübergreifend verfügbar ist (Client). Damit können Nutzer mit beliebigen Endgeräten wie Tablet, Laptop oder Workstation über eine Netzwerkverbindung auf das gekapselte System zur Datenanalyse zugreifen (Server).

Hierbei werden Lösungen inklusive Benutzerschnittstellen für das assistierte Annotieren, die Entwicklung von Analysepipelines sowie deren Anwendung bereitgestellt. Für das Hochleistungsrechnen sind verschiedene Optionen vorgesehen. Einerseits können Projekte mit kleinerem Rechenumfang auf der lokalen Systemhardware durchgeführt werden. Andererseits kann eine Verbindung zu Rechen-Servern oder HPC-Clustern hergestellt werden, um mittlere bis größere Rechenaufträge durchzuführen. Mittels Anbindung an Daten- und Code-Server können Daten einfach ausgetauscht und versioniert sowie Softwareaktualisierungen komfortabel ausgerollt werden. Die Aufgabe des Datenanalysten besteht in der Bereitstellung von entsprechenden Softwarelösungen, jedoch nicht in der direkten Auswertung von Daten aus individuellen Projekten. Damit können Domänenexperten selbstständig die konzipierten Methoden zur automatisierten HD-Verarbeitung (vgl. Abschnitt 3.1) ohne Interaktion mit Datenanalysten nutzen. Das entwickelte Architekturkonzept wird im Weiteren für die Analyse von Bilddaten genutzt, ist jedoch als Konzept auch für andere interdisziplinäre Datenanalyse-Fragestellungen geeignet.

4 Implementierung

4.1 Überblick

Die interdisziplinäre Zusammenarbeit ist häufig für die erfolgreiche Durchführung und Auswertung von Hochdurchsatzexperimenten (HDEs) notwendig, da das dafür erforderliche Wissen oftmals verteilt vorliegt. Domänenexperten besitzen die Expertise hinsichtlich des zugrundeliegenden Problems und dessen Interpretation. Allerdings ist eine Zusammenarbeit mit Spezialisten aus dem Bereich Datenanalyse erforderlich, da die Datenanalysten Kenntnisse im Hinblick auf Algorithmen, Software und Hardware innehaben.

In Kapitel 3 sind bereits die entwickelten Methoden für die Hochdurchsatz (HD)-Bildverarbeitung vorgestellt worden. Es bedarf jedoch einer zugehörigen Implementierung, um das Konzept in der Praxis nutzen zu können.

Das Implementierungskapitel stellt die vielfältigen Aspekte der neu ausgearbeiteten praktischen Realisierung des Konzepts dar, das eine erfolgreiche Zusammenarbeit bei der automatisierten Analyse von HDEs ermöglicht. Dabei wird zunächst detailliert auf die Umsetzung des Architekturkonzepts (vgl. Abschnitt 3.7) in Abschnitt 4.2 eingegangen. Beschreibungen zur praktischen Umsetzung der Deep Learning (DL)-Methoden finden sich in Abschnitt 4.3.

Ferner geht dieses Kapitel explizit auf die entwickelten Softwarelösungen

- KaIDA [Sch22e] (vgl. Abschnitt 4.4),
- AI²Seg [Sch22d, Sch23c] (vgl. Abschnitt 4.5) und
- Grid Screener [Sch21b] (vgl. Abschnitt 4.6)

ein, die im Internet frei zugänglich heruntergeladen werden können.

4.2 Umsetzung des Architekturkonzepts

Die Motivation und vorgeschlagenen Bausteine des Architekturkonzepts sind bereits in Abschnitt 3.7 diskutiert worden. Zur praktischen Umsetzung des Architekturkonzepts ist ein prototypischer Bildverarbeitungsserver eingerichtet, der im Detail unter Anhang I erläutert wird. Nachfolgende Beschreibungen basieren auf den im Rahmen dieser Arbeit entstandenen Veröffentlichungen in [Sch21a, Sch22c, Sch22e].

Mittels einer Remote-Desktop-Software können sich Nutzer mit dem Bildverarbeitungsserver verbinden, ohne spezielle lokale Hardwareanforderungen erfüllen zu müssen. Für Nutzer, die eine Touch-Bedienoberfläche bevorzugen (z. B. im Rahmen der assistierten Annotation), ist ein Lenovo x12 Detachable Tablet eingerichtet worden.

Das assistierte Annotieren und der Entwicklungsprozess von Bildverarbeitungsplines mittels Trainings von Deep Neural Networks (DNNs) ist über das „Karlsruhe Image Data Annotation Tool (KaIDA)“ implementiert. Das Werkzeug beinhaltet eine Umsetzung der beschriebenen Methode aus Abschnitt 3.3 sowie die Bereitstellung zugehöriger Benutzerschnittstellen in einem gebündelten Softwarepaket. Das Plugin „AI-based Annotation Inspection for Segmentation (AI²Seg)“, welches in KaIDA integriert ist, dient für die Anwendung der automatisierten Inspektion von Datensatzannotationen.

Die Anwendung der Bildverarbeitungs pipelines für HDEs in Gitterform ist über das Werkzeug „Grid Screener“ implementiert, wobei auch dieses Softwarepaket zur Erhöhung der Benutzerfreundlichkeit ein Graphical User Interface (GUI) bereitstellt.

Es sind unterschiedliche Möglichkeiten zum Ansteuern von Rechenressourcen berücksichtigt, um den jeweiligen aufgabenspezifischen Anforderungen des jeweiligen HDE gerecht zu werden. Das Hochleistungsrechnen kann für Experimente kleineren Umfangs lokal auf dem eingerichteten *Bildverarbeitungsserver* (siehe Anhang I) durchgeführt werden, da dieser sowohl über eine leistungsfähige Central Processing Unit (CPU) und eingebaute Graphics Processing Unit (GPU) verfügt (Intel Xeon 4210R, NVIDIA Quadro RTX 4000).

Für Auswertungen mittleren Umfangs können Nutzer über ein Representational State Transfer (REST)-Application Programming Interface (API) Anfragen zu einem lokalen *GPU-Server* stellen, welcher nach der Prozessierung die Ergebnisse zurücksendet. Dieser verfügt über eine NVIDIA Tesla V100 GPU bzw. Intel Xeon Gold 5118 CPU. Umfangreiche Rechnungen können auf den *Rechenclustern* HoreKa¹ und HAICORE² durchgeführt werden. Diese verfügen über das Jobsystem Simple Linux Utility for Resource Management (SLURM) [Yoo03]. Mittels ssh können vom Bildverarbeitungsserver Aufträge an das Rechencluster gesendet werden, wobei für Nutzer dieses mittels GUI und ohne direkte Anmeldung auf dem jeweiligen Cluster erledigt werden kann. Die Clusterknoten sind dabei mit Intel Xeon Platinum 8368 CPU und vier bis acht NVIDIA A100 Tensor Core GPUs ausgestattet.

Als geräteübergreifender Datenserver dient die Large Scale Data Facility (LSDF)³ [Jej12] des Karlsruher Instituts für Technologie (KIT). Zum Zwecke der Versionierung von Daten wird auf „Data Version Control (DVC)“ [Pet21] bzw. im Falle von Programmcode auf „Git“ [Tor05] zurückgegriffen.

4.3 Deep Learning und Zugehöriges

Alle DNNs sind in „PyTorch“ [Pas19] mittels des Wrappers „PyTorch Lightning“ [Fal19a] implementiert. „Weights&Biases“ [Bie20] dient als Programmpaket zum Aufzeichnen der Experimente. Für die Augmentierung von Daten wird die Bibliothek „Albumentations“ [Bus20] verwendet. Die Bibliotheken „OpenCV“ [Bra00] und „scikit-image“ [Van14] dienen zur Bereitstellung von speziellen Algorithmen aus dem Bereich Bildverarbeitung. Basierend auf der im Rahmen der Arbeit entstandenen Veröffentlichung [Kli21], wird für die Optimierung der Hyperparameter von DNNs der Algorithmus Random

¹ Ausführliche Beschreibungen der Cluster-Architektur sind unter <https://www.nhr.kit.edu/userdocs/horeka/projects/> zu finden.

² Erläuterungen zum Cluster finden sich analog unter <https://www.nhr.kit.edu/userdocs/haicore/>.

³ Eine Dienstbeschreibung ist unter <http://www.scc.kit.edu/downloads/sdm/Nutzerhandbuch-LSDF-Online-Storage.pdf> abrufbar.

Search [Ber12] genutzt, der einen Kompromiss aus Rechenzeit, Genauigkeit und Komplexität darstellt. Weitere Details finden sich in Anhang E bzw. Anhang H.

Falls nicht gesondert beschrieben, erfolgt die Initialisierung von DNNs nach dem Verfahren von Kaiming [He15]. Um robuste Ergebnisse zu erzielen, werden alle Trainingsdurchläufe der DNNs mindestens fünfmal mit unterschiedlichen Zufallszahlengeneratoren wiederholt und gemittelt. So wird sichergestellt, dass die Ergebnisse nicht von einzelnen Initialisierungen der Anfangsparameter, der Reihenfolge einzelner Bilder im Training oder ähnlichem abhängig sind. Falls nicht abweichend aufgeführt, erfolgt die Modellauswahl mittels Validierungsdatensatz, der aus zufälligen 20 % der Daten generiert wird. Testdatensätze sind in allen Experimenten unveränderlich, werden weder für das Training noch für die Modellauswahl verwendet und lediglich zur Evaluation der Resultate herangezogen.

4.4 KaIDA

Die Umsetzung der konzipierten Methoden im Bereich des assistierten Annotierens (vgl. Abschnitt 3.3) erfolgt mittels KaIDA. Die Softwarelösung ist im Rahmen der Arbeit in [Sch22e] vorgestellt worden. Es umfasst eine GUI zur vereinfachten Anwendung der Methoden für Nutzer, die auf Qt [Riv22] basiert. Abbildung 4.1 zeigt exemplarisch die Software im Einsatz auf einem mobilen Tablet. Mit der entwickelten Software können Nutzer Bilder im Falle verschiedener Bildverarbeitungsaufgaben annotieren, dabei von Assistenzfunktionen wie „Pre-Annotation“ profitieren oder im Nachgang direkt ein DNN trainieren. Die Software ermöglicht Nutzern von Rohdaten hin zum finalen DNN zur Lösung der Aufgabe zu kommen und beschleunigt diesen Prozess durch Assistenzfunktionen bzw. einem Gesamtworkflow, der nur ein Softwarepaket benötigt.

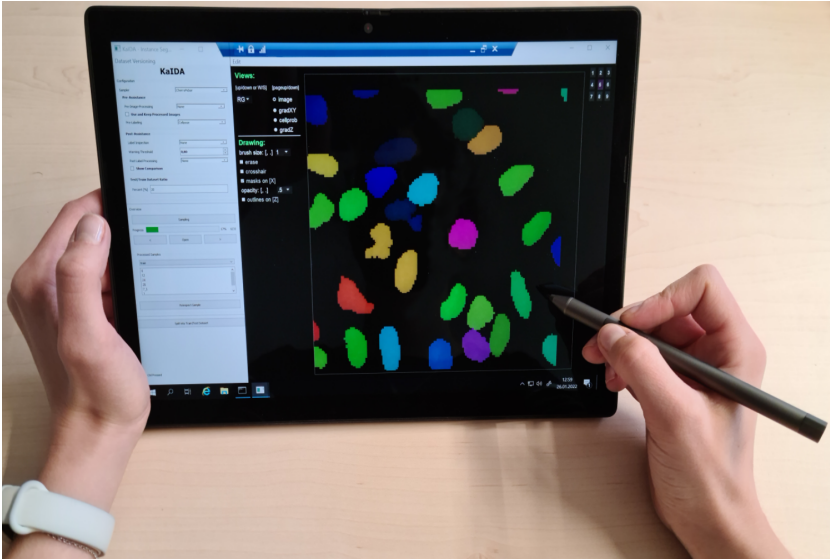


Abbildung 4.1: KaIDA. Durch die GUI kann KaIDA von Anwendern, wie hier auf einem mobilen Tablet (Lenovo x12 Detachable), komfortabel und ergonomisch zum assistierten Annotieren genutzt werden [Sch22e].

Mittels eines Benutzerhandbuchs in Form einer README-Datei kann die open-source Software über <https://gitlab.kit.edu/kit/iai/ml4home/kaida> heruntergeladen und installiert werden. KaIDA ist dafür als pip-Paket inklusive Definition aller Abhängigkeiten zur vereinfachten Installation auf Geräten ausgerollt worden. Das Testen des plattformübergreifenden Softwarepakets ist auf verschiedenen Betriebssystemen (Windows 10, Ubuntu 20.04) mit python 3.8.5 erfolgt. Eine Schnellstart-Bedienungsanleitung findet sich unter Anhang F.1.

Nativ werden die Bildverarbeitungsaufgaben Instanzsegmentierung, Klassifikation, Seed Detektion, Objekterkennung, panoptische Segmentierung und semantische Segmentierung unterstützt. Eine Stärke der KaIDA-Implementierung ist deren modularer Aufbau. Hinsichtlich der Bildverarbeitungsaufgaben können durch austauschbare GUIs und Vererbung im Sinne der Softwareentwicklung zusätzliche Aufgaben hinzugefügt werden.

Eine Basisarchitektur (Elternklasse) definiert die Grundstruktur und kann bezüglich der Aufgabe (Kind) individualisiert werden (siehe Abbildung 4.2).

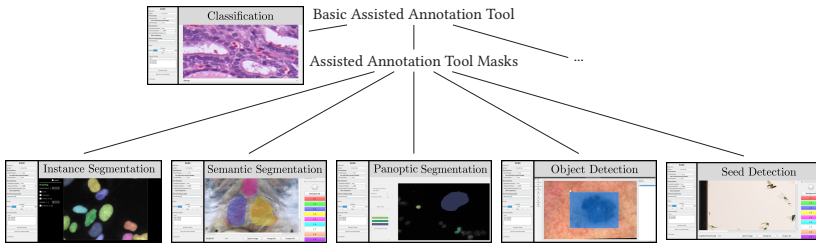


Abbildung 4.2: Aufbau der Implementierung von Bildverarbeitungsaufgaben in KaIDA. Die Implementierung für die spezifische Bildverarbeitungsaufgabe ergibt sich als Ausprägung einer Elternklasse („Basic Assisted Annotation Tool“). Standardmäßig werden Instanzsegmentierung, Klassifikation, Seed Detektion, Objekterkennung, panoptische Segmentierung und semantische Segmentierung unterstützt [Sch22e].

Ferner kann dieser hierarchische Aufbau, analog zu dem erläuterten Vererbungskonzept für unterschiedliche Aufgabentypen, auf die in Abschnitt 3.3 vorgestellten Methoden übertragen werden. Mittels der Definition eines abstrakten Moduls (Eltern), z. B. über den grundsätzlichen Aufbau der „Pre-Annotation“, können individualisierte Methoden umgesetzt werden, wobei zwischen allgemeinen und aufgaben-spezifischen Methoden differenziert wird. Beispielsweise können im Falle des Selektions-Moduls Methoden übergreifend zu den Aufgaben verwendet werden. Konträr dazu variieren jedoch exemplarisch für das Modul der „Pre-Annotation“ die Methoden zur Vorhersage im Falle der verschiedenen Bildverarbeitungsaufgaben.

Der Aufbau ist in Abbildung 4.3 dargestellt. Damit ist es ohne Hürden möglich, Methoden für spezifische Bildverarbeitungsaufgaben oder aufgabenübergreifend in den einzelnen Bausteinen des assistierten Annotierens zu ergänzen, um individuellen Bedürfnissen in Projekten gerecht zu werden.

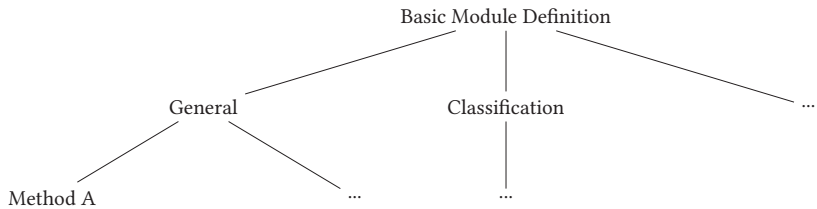


Abbildung 4.3: Aufbau der Implementierung von Methoden in KaIDA. Die Implementierung einer Methode folgt als eine Ausprägung der Elternklasse („Basic Module Definition“). Es wird zwischen allgemeinen (vgl. „General“) und aufgaben-spezifischen (z. B. „Classification“) Methoden (vgl. „Method A“) unterschieden, wodurch die Bausteine des assistierten Annotierens flexibel erweiterbar sind [Sch22e].

4.5 AI²Seg

Die Implementierung von AI²Seg ist in das python-basierte Annotations-Werkzeug KaIDA (vgl. Abschnitt 4.4) als Plugin eingebettet und basiert dementsprechend ebenfalls auf Qt [Riv22]. Eine Veröffentlichung der Softwarelösung ist in [Sch23c] als Teil der vorliegenden Arbeit erfolgt. Die Software setzt die in Abschnitt 3.4 beschriebene Methodik zur Inspektion der Annotationen in Segmentierungs-Datensätzen praktisch um, indem Nutzer potenzielle Fehler im Rahmen der Annotation angezeigt bekommen. Über eine GUI können damit Nutzer assistiert die Qualität des annotierten Datensatzes überprüfen.

Das Plugin ist unter <https://gitlab.kit.edu/kit/iai/ml4home/kaida> als Erweiterung öffentlich frei verfügbar. Eine Bedienungsanleitung ist unter Anhang F.2 dargestellt. Darüber hinaus ist ein Video zur Veranschaulichung des Arbeitsablaufs unter <https://osf.io/t8zx7/> abrufbar, um die Nutzung von AI²Seg zu illustrieren.

Abbildung 4.4 zeigt die entwickelte Benutzeroberfläche zur Inspektion verrauschter Segmentierungsdatensätze. In einem Vergleich werden die verrauschten Benutzerannotationen (links), die DNN-Vorhersagen (rechts) und Vorschläge hinsichtlich der Inspektion visualisiert über Farbkodierung (mittig) gezeigt. Der Nutzer kann entscheiden, ob die bisherige Annotation akzeptiert, die DNN-Vorhersage übernommen oder das Bild erneut

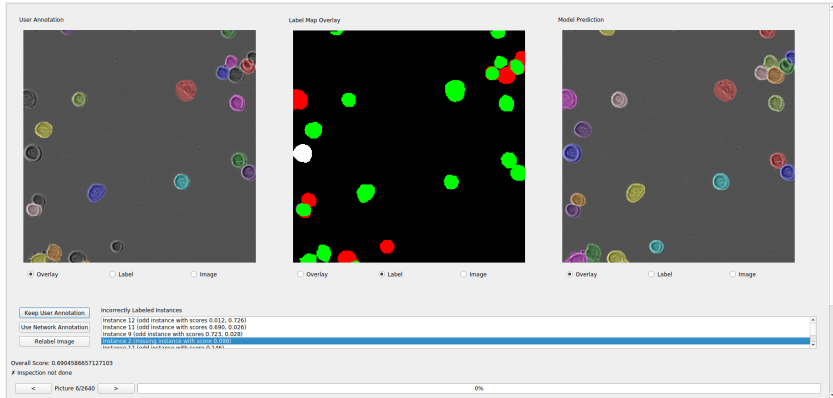


Abbildung 4.4: AI²Seg. Ein Bild aus dem LIVECell [Edl21] Datensatz ist exemplarisch für den Rauschtyp *Vergessen* dargestellt, wobei ein Viertel der Instanzen fehlen ($\beta = 0.25$) (siehe Details in Abschnitt 3.4). Dem Nutzer werden drei verschiedene Ansichten präsentiert: i) die Annotationen des Nutzers (links), ii) die Vorhersage des Modells (rechts) und iii) ein kombiniertes Bild (mittig), das korrekte Segmente/Instanzen grün, Verrauschte rot und Ausgewählte weiß markiert. Dabei können Nutzer zwischen den Ansichten Overlay-Bild, Annotationen oder dem Rohbild wählen. Instanzen sind über unterschiedliche Farben codiert (siehe links und rechts). Es werden alle verrauschten Instanzen in einer Liste angezeigt. Nutzer können wählen, ob die Annotation beibehalten, die DNN-Vorhersage verwendet oder das Bild neu annotiert werden soll. Ferner ergibt sich die Reihenfolge der präsentierten Bilder basierend auf der Gesamtbewertung des vorgeschlagenen Inspektion-Algorithmus, um Bilder mit vielen Fehlern zu priorisieren [Sch23c]. ©2023 IEEE.

annotiert werden soll. Auf Basis der Gesamtbewertung können Nutzer priorisiert mit den Bildern beginnen, die potenziell am häufigsten verrauschte Annotationen enthalten.

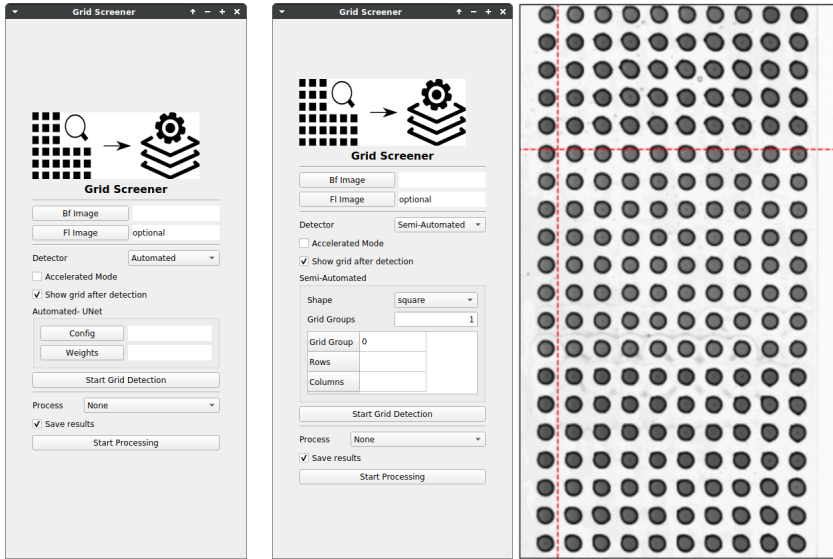
4.6 Grid Screener

Die Methodik zur automatisierten HD-Bildverarbeitung in der Makro- und Mikroebene aus Abschnitt 3.5 ist über das Tool Grid Screener mit GUI auf Basis von Qt [Riv22] umgesetzt worden. Das Werkzeug ist in [Sch21b] veröffentlicht worden. Mit dem Grid Screener können somit Gitter im ersten Schritt automatisiert erkannt werden und im Anschluss die jeweiligen Elemente einzeln abhängig der Aufgabenstellung ausgewertet werden.

Analog zu KaIDA ist dieses zur plattformübergreifenden Nutzung als pip-Paket in python programmiert worden. Das Benutzerhandbuch für die Erklärung zur Installation und Nutzung des open-source Programms findet sich unter <https://gitlab.kit.edu/kit/iai/ml4home/grid-screener>. Die Software ist für Windows 10 und Ubuntu 20.04 in Kombination mit python 3.8.5 getestet. In Abbildung 4.5 ist die Benutzeroberfläche gezeigt und in Anhang F.3 eine Anleitung zum schnellen Programmstart.

Ähnlich zu der erklärten modularen Struktur aus Abbildung 4.3 kann das Programm bezüglich Funktionen zur elementweisen Auswertung erweitert werden. Es wird die Elternklasse „Basic Processing Definition“ definiert, welche für die entsprechende Aufgabenstellung angepasst werden muss (Kindklassen „Method A“, ...). Abbildung 4.6 stellt die hierarchische Struktur der Vererbung in einer Übersicht dar.

Ferner lassen sich die Standarddetektoren, die unter <https://osf.io/b3g5t/> verfügbar sind, austauschen, um andere Gitterstrukturen automatisiert bestimmen zu können. Für das Training anderer DNNs als Detektoren kann KaIDA aus Abschnitt 4.4 verwendet werden.



(a) Automatisiert

(b) Semi-Automatisiert

Abbildung 4.5: Grid Screener. Die GUI des Grid Screeners wird für die automatisierte (a) und semi-automatisierte (b) Gitterdetektion gezeigt, wobei diese sich über die Auswahl eines DNN zur Detektion im Vergleich zum Eingabefenster der Gitterparameter unterscheiden. Im Falle des semi-automatisierten Modus ist ein exemplarisches Eingabefenster für das Markieren der Gittereckpunkte gezeigt. Unter „Process“ können die verfügbaren Methoden im Bereich der Mikroebenen-Prozessierung ausgewählt werden [Sch21b].

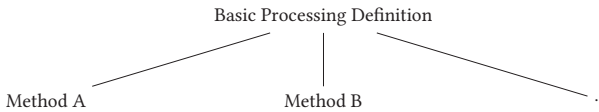


Abbildung 4.6: Aufbau der elementweisen Prozessierung im Grid Screener. Die Implementierung einer spezifischen Methode (vgl. Kindklassen „Method A“, ...) für die Prozessierung pro Gitterelement folgt als eine Ausprägung der Elternklasse („Basic Processing Definition“).

5 Anwendungen

5.1 Datensätze

Die Datensätze zur Evaluierung der entwickelten Konzepte aus Kapitel 3 finden sich in Tabelle 5.1. Es wird zwischen den Kriterien „Aufgabentyp“, „öffentliche Zugänglichkeit“, „Umfang“ und „Bildgröße“ differenziert. Beispielbilder sind in Abbildung 5.1 dargestellt. Das Generieren und Aufbereiten der nicht-öffentlich zugänglichen Datensätze ist Teil des Neuheitswerts der Arbeit.

Es wird eine Vielzahl an unterschiedlichen Datensätzen berücksichtigt, um die entwickelten Methoden bezüglich heterogener Daten zu evaluieren. Die Heterogenität drückt sich aus, indem z. B. verschiedene Aufgaben der Bildverarbeitung oder unterschiedliche Aufnahme modalitäten berücksichtigt werden. Ferner soll über den Einbezug von öffentlich zugänglichen Datensätzen eine Evaluierung der Übertragbarkeit von den entwickelten Methoden auf Probleme außerhalb des Projekts zur personalisierten Onkologie ermöglicht werden. Durch eine gesteigerte Anzahl an Bilddatensätzen wird vermieden, dass sich die Gültigkeit der Analysen auf einzelne Datensätze beschränkt.

Tabelle 5.1: Überblick Datensätze. Die verwendeten Datensätze werden hinsichtlich der Kriterien „Aufgabentyp“, „öffentliche Zugänglichkeit“ (✓/x), „Umfang“ und „Bildgröße“ gegenübergestellt.

Datensatzname	Aufgabentyp	Öffentlich	Training		Test	
			Umfang	Bildgröße in px	Umfang	Bildgröße in px
MoNuSeg [Kum20]	Instanzsegmentierung	✓	592	256x256	14	1000x1000
ISIC-2017-Melanoma [Cod18]	Semantische Segmentierung	✓	2000	~600-6000x~600-4500	600	~600-4500x~600-6000
BBBC038 [Cai19]	Instanzsegmentierung	✓	537	256x256-1024x1024	134	256x256-1024x1024
LIVECell [Edl21]	Instanzsegmentierung	✓	2640	256x256	176	704x502
Insects2022 [Wüh22]	Instanzsegmentierung	x	125	512x512	2	1990 x 1990
DMA-Spheroïd-Hellfeld	Semantische Segmentierung	x	201	128x128	38	128x128
DMA-Spheroïd-Fluoreszenz	Semantische Segmentierung	x	148	2048x2048	31	2048x2048
DMA-Spot	Semantische Segmentierung	x	411	256x256	103	256x256
Wellplate-Spot	Semantische Segmentierung	x	51	256x256	13	256x256
DMA-Zellen-Hellfeld	Instanzsegmentierung	x	78	1920x1440	19	1920x1440
Microglia2023	Seed Detektion	x	60	512x512	8	512x512

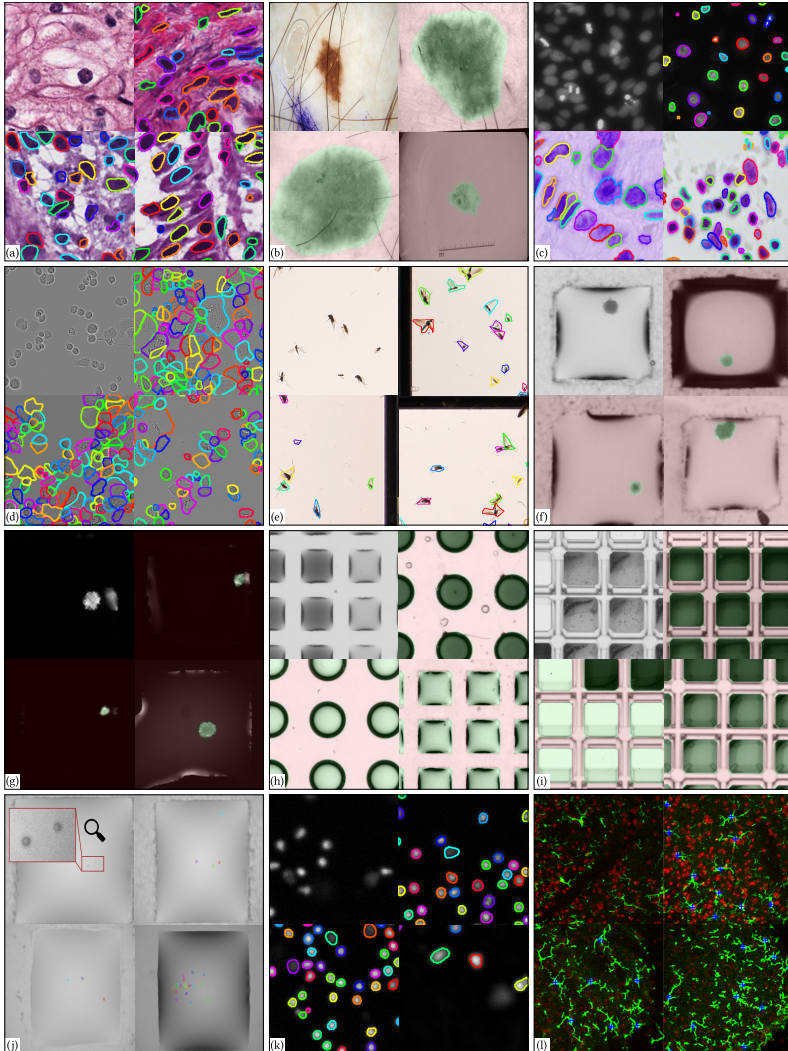


Abbildung 5.1: Auszüge aus den Datensätzen. Beispiele aus den Datensätzen (a) MoNuSeg, (b) ISIC-2017-Melanoma, (c) BBBC038, (d) LIVECell, (e) Insects2022, (f) DMA-Spheroid-Hellfeld, (g) DMA-Spheroid-Fluoreszenz, (h) DMA-Spot, (i) Wellplate-Spot, (j) DMA-Zellen-Hellfeld, (k) DMA-Zellen-Fluoreszenz und (l) Microglia2023 sind gezeigt. Es ist jeweils ein Rohbild ohne Annotation zu sehen (linke obere Ecke). Zur Illustration der Vielfalt sind zudem jeweils drei Bilder mit Annotationen gezeigt. Für semantische Segmentierung ist ein Overlay (grün: Segment, rot: Hintergrund) visualisiert. Instanzen sind farbkodiert, Seeds über blaue Kreuze markiert.

MoNuSeg Der Multi-Organ Nucleus Segmentation (MoNuSeg)-Datensatz¹ [Kum20] (vgl. Abbildung 5.1 a) ist im Rahmen einer Medical Image Computing and Computer Assisted Intervention (MICCAI)-Challenge veröffentlicht worden. Ziel ist es, Zellkerne in Gewebebildern aus sieben unterschiedlichen Organen zu erkennen. Zur Färbung der Histologiebilder wurde dabei Hämatoxylin-Eosin genutzt.

ISIC-2017-Melanoma Der International Skin Imaging Collaboration (ISIC)-2017-Melanoma-Datensatz² wurde ebenfalls im Rahmen einer Challenge publiziert [Cod18]. Ein Ziel des Wettbewerbs ist die automatische Segmentierung von Läsionen in Bildern, um Mediziner bei der Diagnose von Hautkrebs zu unterstützen. Beispiele finden sich in Abbildung 5.1 (b).

BBBC038 Der BBBC038-Datensatz³ (siehe Abbildung 5.1 c) wurde im Rahmen des Kaggle 2018 Data Science Bowl veröffentlicht [Cai19]. Es gilt zwischen Zellkerninstanzen zu unterscheiden. Die Aufnahmen der Organismen (Menschen, Mäuse oder Fliegen) stammen aus verschiedenen Laboren.

LIVECell Der LIVECell-Datensatz⁴ (siehe Abbildung 5.1 d) wurde von Sartorius veröffentlicht [Edl21], wobei dieser acht verschiedene Zellkulturen enthält. Ziel ist es, Instanzen von Zellkernen zu segmentieren. Die Experimente im Rahmen dieser Arbeit verwenden eine reduzierte Form des Datensatzes, worin lediglich Brustkrebszellen SkBr3 betrachtet werden, damit der Umfang des Datensatzes in einer ähnlichen Größenordnung wie die anderen Datensätze ist (vgl. Tabelle 5.1).

Insects2022 Der Insects2022-Datensatz beinhaltet Bilder von Insekten und wurde mithilfe des DiversityScanners [Wüh22] aufgenommen. Ziel ist die

¹ <https://monuseg.grand-challenge.org/Data/>

² <https://challenge.isic-archive.com/data/>

³ <https://bbbc.broadinstitute.org/BBBC038>

⁴ <https://sartorius-research.github.io/LIVECell/>

Segmentierung einzelner Individuen für die Forschung im Bereich der Biodiversität (siehe Abbildung 5.1 e).

DMA-Spheroid-Hellfeld Der Datensatz Droplet Microarray (DMA)-Spheroid-Hellfeld beinhaltet Hellfeldaufnahmen von Sphäroiden¹. Die Zielstellung ist die Segmentierung und Analyse von deren Eigenschaften wie Fläche oder Form, um z. B. die Wirksamkeit von Medikamenten gegen Hirntumore zu untersuchen [Cui23]. Beispiele hierzu finden sich in Abbildung 5.1 (f).

DMA-Spheroid-Fluoreszenz Analog zum DMA-Spheroid-Hellfeld-Datensatz beinhaltet der DMA-Spheroid-Fluoreszenz-Datensatz Aufnahmen von Sphäroiden, die jedoch mittels Fluoreszenzmikroskopie generiert worden sind (vgl. Abbildung 5.1 g). Die Segmentierungsergebnisse werden beispielsweise für die Auswertung von Experimenten im Bereich der personalisierten Onkologie genutzt [Cui23].

DMA-Spot Der DMA-Spot-Datensatz zeigt quadratische und kreisförmige hydrophile Elemente. Diese sollen vom hydrophoben Hintergrund unterschieden werden. Auszüge des Datensatzes finden sich in Abbildung 5.1 (h). Als Anwendung kann die automatisierte Detektion von gitterförmigen HDEs aufgeführt werden.

Wellplate-Spot Analog dazu sollen im Falle des Wellplate-Spot-Datensatzes einzelne Elemente der Gitterstruktur von der Hintergrundfläche differenziert werden (vgl. Abbildung 5.1 i). Der Anwendungsfall ist ebenfalls im Bereich der automatisierten Gittererkennung verortet.

¹ Sphäroide sind Zusammenschlüsse von Zellen, die organähnliche Strukturen in Hochdurchsatzexperimente (HDEs) imitieren sollen.

DMA-Zellen-Hellfeld Abbildung 5.1 (j) zeigt Beispielbilder des DMA-Zellen-Hellfeld-Datensatzes. Hierbei gilt es einzelne Zellen im Hellfeldbild zu erkennen, um beispielsweise Analysen im Bereich der personalisierten Onkologie durchzuführen.

DMA-Zellen-Fluoreszenz Der DMA-Zellen-Fluoreszenz-Datensatz zeigt Fluoreszenzaufnahmen, wobei die Einfärbung mittels der Farbstoffe Hoechst, Calcein oder Propidium Iodide (PI) erfolgt ist, jedoch keine gesonderten Datensätze hinsichtlich der Farbstoffe generiert worden sind. Das Ziel ist es, einzelne Instanzen der Zellkerne zu erkennen. Ein Anwendungsfall ist die Ermittlung der Zellviabilität im Rahmen von Medikamententests. Beispiele hierzu finden sich in Abbildung 5.1 (k).

Microglia2023 Der Microglia2023-Datensatz besteht aus kombinierten Fluoreszenzmarkern zur Detektion von Mikroglia-Immuneffektorzellen, die Teil des zentralen Nervensystems sind. Konträr zu den anderen Datensätzen besteht das Ziel in diesem Fall lediglich in der Quantifizierung der Zellen hinsichtlich deren Anzahl. Eine Bestimmung der Segmente ist nicht notwendig. Beispielbilder sind in Abbildung 5.1 (l) gezeigt.

5.2 Assistierte Annotieren

5.2.1 Selektion

Nachfolgend wird auf die Ergebnisse verschiedener Methoden des Selektionsmoduls (vgl. Abschnitt 3.3.3) eingegangen, das gezielt informative Bilder für die Annotation auswählen soll. Es wird somit die praktische Anwendung von Deep Active Learning (DAL) im Annotationsprozess bewertet. Ausführliche theoretische und quantitative Betrachtungen folgen im Methoden-Vergleich in Abschnitt 5.4.2.

Resultate der Selektion basierend auf der *Heterogenität* im Datensatz finden sich in Abbildung 5.2, wobei diese im Rahmen dieser Arbeit in [Sch23b]

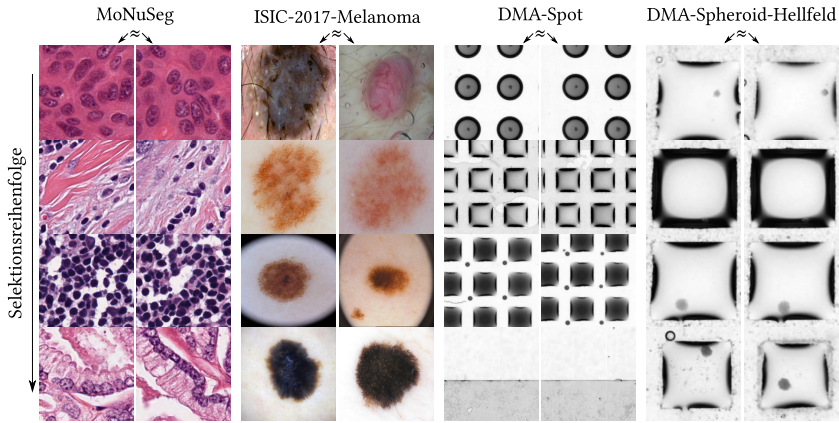


Abbildung 5.2: Automatisierte Selektion basierend auf Heterogenität. Mittels ResNet18 in Kombination mit einer PCA auf zehn Merkmale wird über euklidische Distanzen die Selektionsreihenfolge (linke Reihe des jeweiligen Datensatzes) für verschiedene Beispieldatensätze (MoNuSeg, ISIC-2017-Melanoma, DMA-Spot, DMA-Spheroid-Hellfeld) im Sinne von Core Sets [Sen18] bestimmt und visualisiert. Zudem werden die jeweils zugehörigen Bilder mit größter Ähnlichkeit (" \approx ") zeilenweise gegenübergestellt [Sch23b].

veröffentlicht worden sind. Dabei werden die Datensätze MoNuSeg, ISIC-2017-Melanoma, DMA-Spot und DMA-Spheroid-Hellfeld analysiert. Zur automatisierten Bestimmung heterogener Datenelemente eines Datensatzes, der annotiert werden soll, wurde ein auf ImageNet [Den09] vortrainiertes ResNet18 [He16] zur Generierung von Bildmerkmalen verwendet. In Kombination mit einer Principal Component Analysis (PCA) zur Dimensionsreduktion wird ein zehndimensionaler Merkmalsraum aller Bilder erzeugt, in dem ein Vergleich der Bilder möglich ist. Für den Vergleich der Bilder mittels der generierten Merkmale wurden euklidische Distanzen genutzt. Bei einer qualitativen visuellen Betrachtung der Selektionsreihenfolge wird deutlich, dass die selektierten Bilder (vgl. Abbildung 5.2, vertikale Richtung) jeweils sehr unterschiedlich sind. Ferner zeigt sich, dass auch die nächsten Nachbarn im Merkmalsraum (vgl. Abbildung 5.2, horizontale Richtung) starke visuelle Ähnlichkeiten aufweisen. Dies trifft auf alle getesteten Datensätze zu. Damit eignet sich der Ansatz zur Ermittlung heterogener Dateninstanzen.

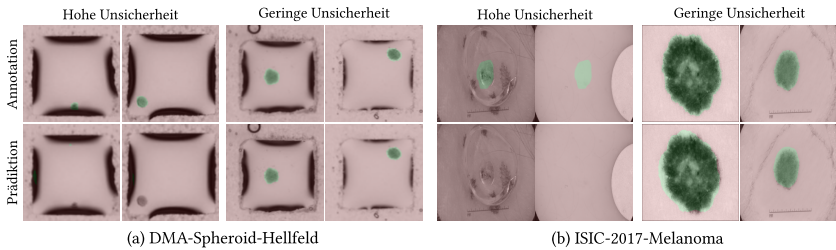


Abbildung 5.3: Automatisierte Selektion basierend auf Unsicherheit. Das DNN ist mit zufälligen $\rho = 8\%$ annotierten Bildern des Gesamtdatensatzes für den DMA-Spheroid-Hellfeld- und ISIC-2017-Melanoma-Datensatz trainiert worden. Mittels MCD werden die Prädiktionen (zweite Zeile) mit hoher und geringer Unsicherheit eines U-Nets bezüglich des nicht annotierten Datensatzes bestimmt. Es erfolgt eine Gegenüberstellung zur vorhandenen Annotation im Ursprungsdatensatz (erste Zeile).

Resultate des Selektionsansatzes basierend auf *Unsicherheit* finden sich in [Abbildung 5.3](#) und [Abbildung 5.4](#).

Ein unsicherheitsbewusstes U-Net, erzeugt über die Verwendung von Monte Carlo Dropout (MCD) (siehe [Abschnitt 2.1.3](#)), ist initial auf $\rho = 8\%$ der verfügbaren Annotationen trainiert worden. Im Anschluss sind damit Bilder des noch nicht annotierten Datensatzes ermittelt worden, die hohe Unsicherheiten bezüglich der Vorhersagen durch das U-Net aufweisen. In [Abbildung 5.3](#) sind jeweils die zwei Beispiele gezeigt, welche die höchste bzw. niedrigste Unsicherheit (gemäß Q_{IDaIS} , vgl. [Gleichung \(D.7\)](#)) aufweisen. Vergleicht man die Prädiktion des Deep Neural Network (DNN) mit der wahren Annotation, zeigt sich bei beiden dargestellten Datensätzen, dass Bilder mit hoher Unsicherheit nur mit großen Fehlern im aktuellen Trainingszustand des Netzes vorhergesagt werden können. Damit ist es mittels des unsicherheitsbewussten DNN möglich, noch fehlerhaft vorhergesagte Bilder zu identifizieren, um diese anschließend zu annotieren und damit in den Trainingsdatensatz zu integrieren. Zur weiteren quantitativen Analyse findet sich in [Abbildung 5.4](#) eine Gegenüberstellung von Segmentierungsfehler ($1 - Q_{\text{DSC}}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$) und präziserer Unsicherheit (Q_{IDaIS}). Korrelationen von $\xi = 0.912$ im Falle DMA-Spheroid-Hellfeld bzw. $\xi = 0.747$ bei ISIC-2017-Melanoma untermauern die These, dass über den genannten Ansatz Bilder prognostiziert werden, bei denen ein DNN noch große Fehler macht.

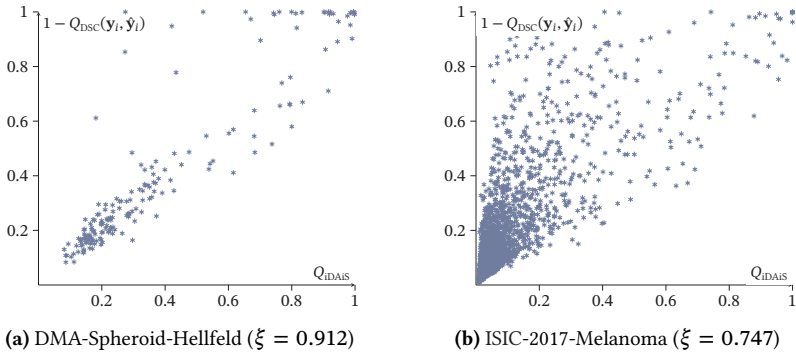


Abbildung 5.4: Vergleich Segmentierungsfehler und Unsicherheit. Für die Datensätze DMA-Spheroid-Hellfeld (a) und ISIC-2017-Melanoma (b) wird der Zusammenhang zwischen Segmentierungsfehler ($1 - Q_{DSC}(y_i, \hat{y}_i)$) und prädizierter Unsicherheit (Q_{IDAIS}) gezeigt. Ferner ist die lineare Korrelation der Größen über ξ gezeigt.

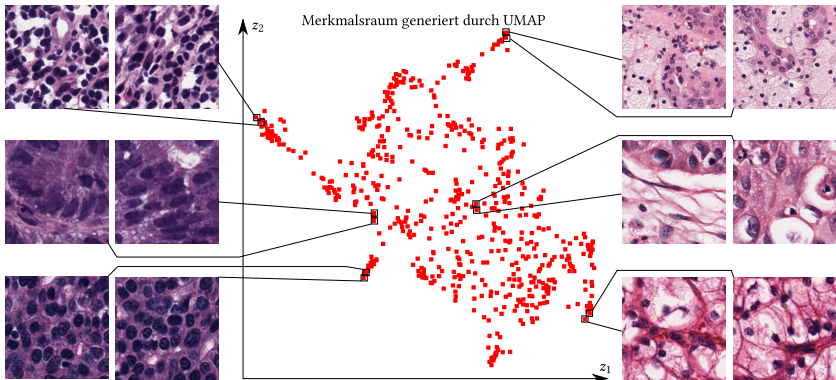


Abbildung 5.5: „Smart Picking“. Es wird der über ein ResNet18 und die Uniform Manifold Approximation and Projection (UMAP)-Dimensionsreduktion erzeugte zweidimensionale Merkmalsraum (z_1, z_2) im Falle von MoNuSeg gezeigt. Exemplarisch sind verschiedene ausgewählte Bilder aus Clustern dargestellt.

In [Abbildung 5.5](#) findet sich die beispielhafte zweidimensionale Repräsentation des MoNuSeg-Datensatzes für die „Smart Picking“-Methode (vgl. [Abschnitt 3.3](#)), welche über die Kombination aus ResNet18 und UMAP-Dimensionsreduktion generiert worden ist. Es zeigt sich bei der Betrachtung

der exemplarisch ausgewählten Bilder aus den jeweiligen Clustern, dass Nutzer durch die entwickelte Methode in vielen Fällen direkt ähnliche Bilder sehen können. Dies ermöglicht eine Zeitersparnis im Vergleich zur „Cherry Picking“-Methode, da Cluster unmittelbar in der 2D-Repräsentation ersicht-lich sind und im Gegensatz dazu nicht der gesamte Datensatz bildweise betrachtet werden muss.

5.2.2 Pre-Processing

Mit der Möglichkeit einer Bildvorverarbeitung soll die Darstellung der zu an-notierenden Bilder optimiert werden. Diese Neuerung im Vergleich zu Ar-beitsabläufen aus dem Stand der Technik soll Experten das Annotieren er-leichtern. Verschiedene Methoden des „Pre-Processings“ finden sich in Ab-bildung 5.6, die in Abschnitt 3.3.4 vorgestellt worden sind. Hier stehen das ursprüngliche und das vorverarbeitete Bild einander gegenüber, um einen di-rekten Vergleich zu ermöglichen. Die Analyse der Bilder aus verschiedenen Datensätzen (DMA-Zellen-Fluoreszenz, BBBC039 [Ljo12], Insects2022, DMA-Zellen-Fluoreszenz) erfolgt im Weiteren qualitativ, da die Intention des „Pre-Processings“ das Generieren einer verbesserten Darstellung des Bildes ist, welche Nutzern das Annotieren vereinfachen soll. Die Ergebnisse gründen sich auf der Veröffentlichung [Sch22e], die im Verlauf dieser Arbeit entstan-den ist.

Der Vergleich zeigt eindeutig, dass durch Normalisierung die Sphäroidstruk-tur sowie die Zellkerne besser erkennbar sind und somit für Experten leichter zu annotieren sind. Zudem können eine Entzeichnung oder die Reduktion von Rauschen in der Anwendung von Vorteil sein, um Artefakte zu reduzieren, die durch die Kameraaufnahme entstehen. Damit wird für Nutzer eine Dar-stellung generiert, die schneller während des Annotierens interpretiert und annotiert werden kann. Dies zeigt sich ferner im Falle der Ausschneidefunk-tion, wodurch lediglich der Bereich von Interesse dargestellt wird. Dies hilft Experten, sich auf relevante Aspekte im Bild zu konzentrieren und erhöht die Übersichtlichkeit.

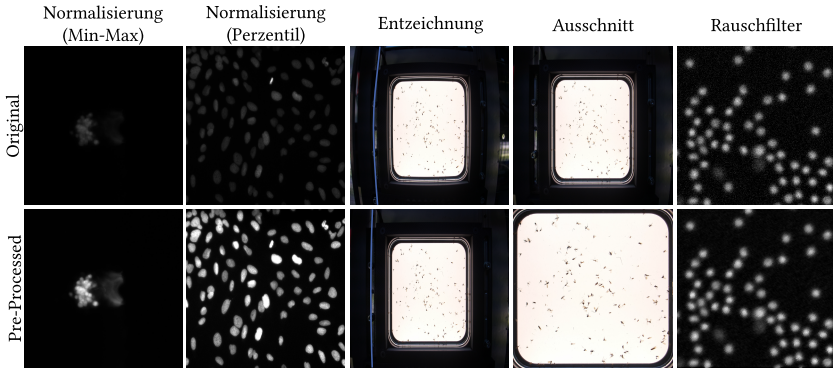


Abbildung 5.6: Pre-Processing. Originale (erste Zeile) und vorverarbeitete Bilder (zweite Zeile) werden für die Methoden Normalisierung (Min-Max und Perzentil), Entzeichnung (Markierung der Randlinien zur Illustration in blau), Ausschnitt und Rauschfilter exemplarisch auf unterschiedlichen Datensätzen dargestellt, um die Vorteile einer Vorverarbeitung qualitativ bewerten zu können. Die Beispiele stammen aus den DMA-Zellen-Fluoreszenz-, BBBC039- [Ljo12], Insects2022- und DMA-Zellen-Fluoreszenz-Datensätzen (von links nach rechts).

5.2.3 Pre-Annotation

Im Weiteren soll auf die Ergebnisse von verschiedenen „Pre-Annotation“-Methoden aus Abschnitt 3.3.5 (Otsu, Schwellwert über Perzentile, Cellpose [Str21b], vortrainiertes U-Net [Ron15]) im Falle der Segmentierung eingegangen werden. Die Methoden werden anhand diverser Datensätze (MoNuSeg, ISIC-2017-Melanoma, DMA-Spheroïd-Hellfeld, DMA-Zellen-Fluoreszenz) ausgewertet. Die Ergebnisse basieren auf der im Rahmen der Arbeit entstandenen Veröffentlichungen in [Sch22e, Sch23b].

Die Resultate hierzu finden sich in Abbildung 5.7, wobei pro Datensatz zwei Beispiele inklusive originaler Annotation im Vergleich zu den Vorhersagen mittels verschiedener „Pre-Annotation“-Methoden gezeigt werden. Die Güte der Methoden wird pro Bild qualitativ mittels Overlays/Konturen und quantitativ über Metriken (Q_{DSC} bzw. Q_{AJI+}) analysiert.

Die Haupteckentnis aus der Bewertung lautet, dass die Eignung der individuellen Methode zur „Pre-Annotation“ von den spezifischen Anforderungen des jeweiligen Problems oder Datensatzes beeinflusst wird. Daher ist es

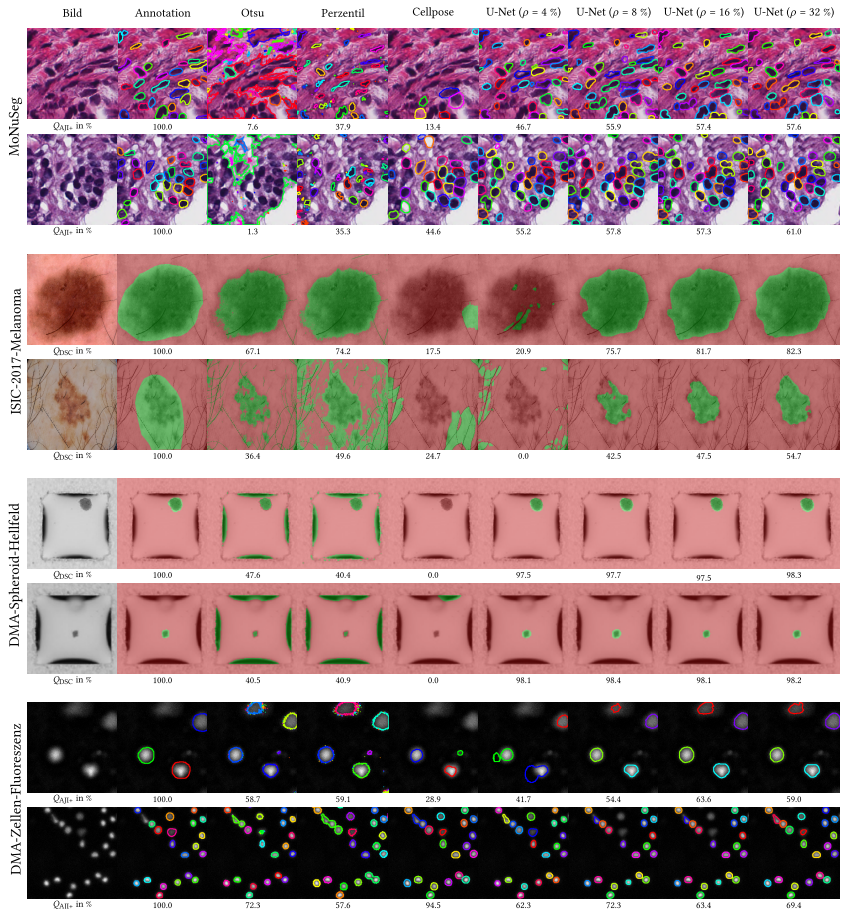


Abbildung 5.7: Pre-Annotation. Die Resultate unterschiedlicher Pre-Annotation-Methoden (Otsu, Schwellwert über Perzentile, Cellpose [Str21b], U-Net [Ron15]) sind für Beispielbilder (erste Spalte) der eingeführten Datensätze im Vergleich zur wahren Annotation (zweite Spalte) illustriert. Lediglich das U-Net ist auf einem bereits annotierten Subdatensatz trainiert worden. Dabei wurden $\rho = \{4, 8, 16, 32\} \%$ der Bilder aus dem Datensatz im Vorfeld ohne Unterstützung annotiert. Die anderen Verfahren erfordern keinen annotierten Subdatensatz. Neben qualitativen Ergebnissen in Form von Konturen bzw. Overlays sind quantitative Metriken (Q_{DSC} , $Q_{\text{AJI}+}$) bildweise dargestellt [Sch23b].

sinnvoll, wie im Konzept vorgeschlagen (vgl. Abschnitt 3.3.5), verschiedene Methoden problembezogen auswählen zu können. Eine Möglichkeit hierzu ist der in Abschnitt 3.6.2 vorgestellte automatisierte Ansatz zur Bestimmung von ähnlichen Datensätzen, um beispielsweise ein bereits vorhandenes DNN zu bestimmen, das sich gut zur „Pre-Annotation“ eignet. Ferner können Nutzer das Ergebnis der verschiedenen Methoden anhand der Leistungsfähigkeit auf einem Beispielbild analysieren, um eine geeignete Methode empirisch auszuwählen.

Das auf einem kleinen Datensatz trainierte DNN (hier: U-Net) ist im Allgemeinen eine universelle Methode, die jedoch als Nachteil einen bereits partiell annotierten Datensatz für das Training erfordert. Zudem zeigt sich der Trend, dass die Vorhersagegenauigkeit steigt, wenn ein größerer Umfang an Annotationen für das initiale Training des DNN zur Verfügung steht. Allerdings ist der monotone Verlauf der Lernkurve, also der Güte in Abhängigkeit der Annotationsrate ρ , bei Betrachtung der Ergebnisse nicht bei allen Datensätzen vorzufinden.

Die Ergebnisse der in Abschnitt 3.3.5 vorgestellten Nutzer-Studie, welche zur Analyse der Vorteile für Nutzer bei der Verwendung von „Pre-Annotation“ genutzt werden soll, finden sich in Abbildung 5.8. Die Studie schließt zwei Nutzer (A1, A2) ein, die pro Datensatz jeweils 20 Bilder annotiert haben. Dazu wird für die semantische Segmentierung der DMA-Spheroid-Hellfeld-Datensatz genutzt und analog für die Instanzsegmentierung der BBBC038-Datensatz verwendet. Jede Konfiguration ist pro Nutzer fünffach wiederholt worden. Die Konfigurationen „keine vorliegenden Vorhersagen“, „vortrainierte Deep Learning (DL)-Pipeline“ und „Otsu-Schwelwertverfahren“ werden gegenübergestellt.

Es ist festzustellen, dass für beide Datensätze Q_{DSC} und $Q_{\text{AJI}+}$ keine beachtlichen Unterschiede zwischen den verschiedenen Methoden aufweisen (vgl. Abbildung 5.8b). Hingegen unterscheiden sich die Annotationszeiten $Q_{t_{\text{annotation}}}$ stark zwischen den Datensätzen sowie bezüglich der Methoden (vgl. Abbildung 5.8a).

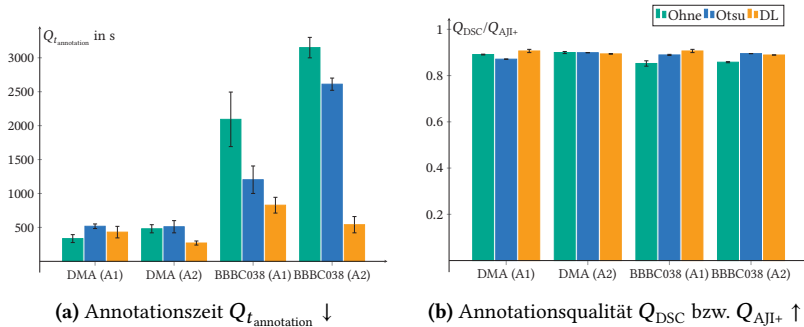


Abbildung 5.8: Vergleich Annotationszeit und Annotationsqualität Pre-Annotation. Die Annotationszeit $Q_{t_{\text{annotation}}}$ (a) und zugehörige Annotationsqualität in Form von Q_{DSC} bzw. Q_{AJI+} (b) wird für verschiedene „Pre-Annotation“-Methoden (Ohne, Otsu, DL) verglichen. Dazu werden die Datensätze DMA-Spheroid-Hellfeld und BBBC038 verwendet. Es werden die Ergebnisse für Nutzer A1 und A2 gezeigt.

Für den BBBC038-Datensatz kann im Falle von Nutzer A1 die Annotationszeit im Vergleich zu keiner „Pre-Annotation“ durch Otsu um zirka 42 % (2093 s \rightarrow 1203 s) reduziert werden, mittels DL sogar um mehr als 60 % (2093 s \rightarrow 828 s). Die Zeitersparnisse zeigen sich auch bei Nutzer A2, wobei die Reduktion von Otsu mit zirka 18 % (3150 s \rightarrow 2610 s) geringer bzw. im Falle von DL mit über 80 % (3150 s \rightarrow 540 s) höher im Vergleich zu Nutzer A1 ausfällt. Der große Benefit ist wahrscheinlich auch auf die vielen Instanzen pro Bild im BBBC038-Datensatz zurückzuführen.

Konträr dazu ergeben sich im Falle des DMA-Datensatzes sogar Nachteile bezüglich $Q_{t_{\text{annotation}}}$ durch „Pre-Annotation“. Hierbei führt „Pre-Annotation“ nach Otsu bei beiden Nutzern zu steigenden Annotationszeiten im Vergleich zu keiner Unterstützung. Dies liegt vermutlich daran, dass die Güte der Verfahren nicht ausreichend ist, wodurch die Korrektur von Fehlern aufwändiger ist, als von Grund auf zu annotieren. Die Nutzung einer DL-Pipeline reduziert im Falle von Nutzer A2 die Annotationszeit um mehr als 40 % (480 s \rightarrow 270 s), bleibt jedoch im Falle von Nutzer A1 auf einem ähnlichen Niveau wie mit ausbleibender Unterstützung. Gründe hierfür liegen in der Beschaffenheit des DMA-Spheroid-Hellfeld-Datensatz. Es liegt meist nur ein Segment vor, das sich in der Regel zügig von Nutzern annotieren lässt. Damit hält sich der

Vorteil von „Pre-Annotation“ bei allgemein geringem Annotationsaufwand in Grenzen, wie z. B. bei wenigen Objekten pro Bild.

5.2.4 Post-Annotation-Processing

Über eine automatisierte Nachverarbeitung von Annotationen, dem „Post-Annotation-Processing“, ergibt sich erstmalig die Möglichkeit, Annotationen auf Basis von Vorwissen gegebenenfalls direkt zu verbessern. Das „Post-Annotation-Processing“ (vgl. Abschnitt 3.3.7) ist beispielhaft in Abbildung 5.9 veranschaulicht, wobei die Bilder aus unterschiedlichen Datensätzen (ISIC-2017-Melanoma, DMA-Spheroid-Hellfeld, DMA-Zellen-Fluoreszenz) stammen. Dabei erfolgt eine qualitative Gegenüberstellung von Annotationen vor und nach der automatisierten Verarbeitung. Zur Verbesserung der Veranschaulichung sind zusätzlich Detailbilder dargestellt. Ferner sollen zur quantitativen Analyse die Metriken ΔQ_{DSC} bzw. ΔQ_{AJI+} (vgl. Abschnitt 3.2) verwendet werden. Die Metriken beschreiben den Unterschied zwischen der ursprünglich fehlerhaften Annotation und korrekten Annotation nach Nachverarbeitung. Je größer die Metriken sind, desto einflussreicher ist die jeweilige Nachverarbeitung (Wertebereich zwischen 0 % und 100 %). Die Ergebnisse bauen auf der im Rahmen der Arbeit entstandenen Veröffentlichung [Sch22e] auf.

Zu Beginn wird die Glättungsmethode über morphologische Operatoren dargestellt, womit Segmentgrenzen begradigt bzw. kontinuierlicher gemacht werden. Allerdings ist die Auswirkung im Sinne ΔQ_{DSC} mit weniger als 1 % sehr gering, da nur ein kleiner Teil des Gesamtsegments betroffen ist. Falls die Annahme von in der Gesamtheit geschlossenen Strukturen getroffen werden kann, garantiert das Füllen von Segmenten die Beseitigung von Löchern in Strukturen. Hierbei folgt ein großer Einfluss durch die Nachverarbeitung ($\Delta Q_{DSC} = 65.1\%$). Ferner lassen sich kleine Segmente herausfiltern und damit das Rauschen in Annotationen reduzieren. Die Auswirkung auf die Metrik ΔQ_{AJI+} ist allerdings vernachlässigbar niedrig (0.5 %), da der Flächenanteil der kleinen Segmente im Vergleich zu den restlichen Segmenten gering ist.

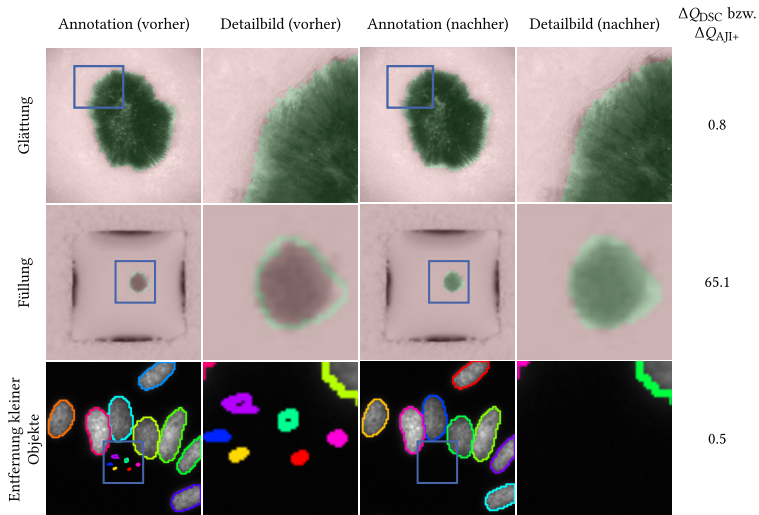


Abbildung 5.9: Post-Annotation-Processing. Annotationen werden vor und nach automatisierter Nachbearbeitung gezeigt. Detailbilder dienen zur verbesserten Veranschaulichung. Die Methoden Glättung, Füllung von Segmenten und Entfernen kleinerer Objekte sind zu sehen. Über die Metriken ΔQ_{DSC} bzw. ΔQ_{AJI+} wird die Verbesserung der Annotationsgenauigkeit ausgedrückt. Die Bilder stammen aus den Datensätzen ISIC-2017-Melanoma, DMA-Spheroid-Hellfeld und DMA-Zellen-Fluoreszenz (von oben nach unten). Im Falle der Glättung bzw. des Entfernens kleinerer Objekte sind kleine Werte im Sinne von ΔQ_{DSC} bzw. ΔQ_{AJI+} festzustellen, da die betroffenen Segmente im Vergleich zur Gesamtsegmentfläche gering sind.

Als Erkenntnis der Bewertung kann festgehalten werden, dass die Nachverarbeitung von Annotationen je nach Fehlerumfang ein beträchtliches Potenzial zur quantitativen Verbesserung der Annotationen aufweist (vgl. ΔQ_{DSC} bzw. ΔQ_{AJI+}).

Eine automatische Nachbearbeitung der Annotationen kann jedoch zu ungewollten Änderungen führen, wenn das Vorwissen nicht der Realität entspricht. Z. B. löscht die Annahme „maximal ein Objekt pro Bild“ alle Objekte mit Ausnahme des flächenmäßig größten Objekts. Bei einer Verletzung der Annahme werden Objekte irrtümlich gelöscht. Daher sieht das Modul die Möglichkeit vor, Nutzern die Änderungen gegenüberzustellen, um eine Kontrolle der Prozessierung zu ermöglichen.

5.2.5 Annotation-Inspektion

Final werden die Resultate der Inspektion von Annotationen präsentiert, welches methodisch in Abschnitt 3.3.8 eingeführt worden ist. Die Neuerung durch diesen Methodenbaustein ist, Nutzer auf vermeintlich falsche Annotationen aufmerksam machen zu können. In Abbildung 5.10 sind Beispiele zu sehen, die eine Warnung an den Benutzer senden und Fälle ohne Warnung, wobei die Datensätze analog zu Abbildung 5.9 sind. Zudem wird äquivalent zu Abschnitt 5.2.4 im Falle einer Warnung das Verbesserungspotenzial (ΔQ_{DSC} bzw. ΔQ_{AJI+}) gezeigt, wenn der Fehler korrigiert wird. Die Resultate stützen sich auf die im Rahmen der Arbeit entstandene Publikation [Sch22e].

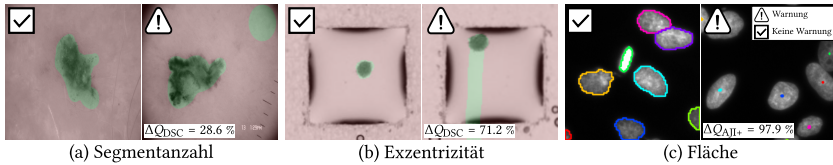


Abbildung 5.10: Annotation-Inspektion. Es werden drei verschiedene Inspektionskriterien gezeigt. Die Annahmen sind: (a) maximal ein Segment pro Bild, (b) keine Exzentrizität und (c) eine Mindestfläche pro Segment. Fälle einer erzeugten Warnung zur Inspektion sowie keiner Warnung sind über Symbole markiert. Die dargestellten Irrtümer sind ein Fehlklick (Segmentanzahl), ein Abrutschen des Malwerkzeugs beim Annotieren (Exzentrizität) und die Annotation von Seeds statt Instanzen (Fläche). Im Falle einer Warnung wird das quantitative Verbesserungspotenzial mittels der Metriken ΔQ_{DSC} bzw. ΔQ_{AJI+} gezeigt. Die Datensätze sind analog zu Abbildung 5.9.

Beispielsweise kann eine Warnung resultieren, wenn die Annahme von maximal einem Segment pro Bild verletzt ist. Dies kann durch einen Fehlklick auftreten. In diesem Fall liegt ein Verbesserungspotenzial von $\Delta Q_{DSC} = 28.6\%$ vor. Ferner ist ein weiterer Anwendungsfall, die Form von Segmenten zu restringieren. Benutzer erhalten Feedback, wenn ein stark exzentrisches Segment vorliegt. Ein mögliches Szenario für diese Fehlerart kann das Abrutschen des Malwerkzeugs beim Annotieren sein. Mit einem Verbesserungspotenzial von $\Delta Q_{DSC} > 70\%$ zeigt sich der Nutzen einer automatisierten Inspektion deutlich. Darüber hinaus kann Vorwissen bezüglich der Segmentgröße integriert werden, wodurch es zur Warnung kommt, wenn z. B. die Mindestfläche

unterschritten wird. Das Annotieren von Seeds statt Instanzen kann als Praxisbeispiel aufgeführt werden, wobei in diesem Fall der Einfluss immens ist (vgl. $\Delta Q_{AJI+} = 97.9\%$). Ausführliche Ergebnisse zur Inspektion von Annotationen finden sich in Abschnitt 5.2.7.

5.2.6 Software-Tool KaIDA

Ein Vergleich von Software-Tools ist in Tabelle 5.2 gezeigt, der auf die in Abschnitt 3.2 eingeführten Kriterien zur Bewertung zurückgreift. Die Gegenüberstellung basiert auf der im Rahmen der Arbeit entstandenen Veröffentlichung in [Sch22e]. Inhalt des Vergleichs sind die Tools ILT [Bar19], labelMe [Wad18], LabelImg [Lin15], Cellpose [Str21b], CVAT [Sek20] und hasty.ai [Rou22] sowie die neu entwickelte Software KaIDA [Sch22e].

Tabelle 5.2: Vergleich der Software-Tools zur Annotation. Verschiedene Tools (ILT, labelMe, LabelImg, Cellpose, CVAT, hasty.ai und KaIDA) werden anhand der in Abschnitt 3.2 eingeführten Bewertungskriterien gegenübergestellt. Dabei wird zwischen vollständiger (✓), partieller (x/✓) und fehlender (x) Zielerreichung differenziert [Sch22e].

Kriterium	Tools						
	ILT [Bar19]	labelMe [Wad18]	LabelImg [Lin15]	Cellpose [Str21b]	CVAT [Sek20]	hasty.ai [Rou22]	KaIDA [Sch22e]
Selektion	x	x	x	x	x	✓	✓
Pre-Annotation	x	x	x	✓	✓	✓	✓
Pre-Processing	x	x	x	x	✓	✓	✓
Post-Annotation-Processing	x	x	x	x	x	x	✓
Annotation-Inspektion	x	x	x/✓	x	x/✓	✓	✓
Verschiedene Bildverarbeitungsarbeiten	x	✓	x	x/✓	✓	✓	✓
Training/Anwendung	x	x	x	x/✓	✓	✓	✓
Versionierung Daten	x	x	x	x	x	x	✓
Zugänglichkeit	✓	✓	✓	✓	✓	x	✓
GUI/Handbuch	✓	✓	✓	✓	✓	✓	✓
Erweiterbarkeit	x	x	x	x	x/✓	x/✓	✓
Software- und Hardwareanforderungen	✓	✓	x/✓	x/✓	✓	✓	x/✓

KaIDA und hasty.ai (in der beta-Version) sind die einzigen Programmpakete, bei denen die Annotationsreihenfolge verändert werden kann (vgl. „Selektion“).

Ferner unterstützen die kompakten Softwarelösungen Image Labeling Tool (ILT), labelMe und LabelImg keine „Pre-Annotation“-Funktion.

Im Hinblick auf vorgelagerte Bildbearbeitungsoperationen erlauben einzig CVAT mittels OpenCV-Integration [Bra00] und KaIDA diese Funktionalität (vgl. „Pre-Processing“).

Die Idee der direkten Nachbearbeitung von Annotationen (vgl. „Post-Annotation-Processing“) ist ein Alleinstellungsmerkmal von KaIDA.

CVAT unterstützt eine manuelle Überprüfung von Annotationen. LabelImg integriert eine Funktion, um geprüfte Annotationen mit einer Flagge zu markieren. Eine automatisierte Überprüfung von Annotationen gibt es jedoch nur in Falle von hasty.ai und KaIDA (vgl. „Annotation-Inspektion“).

ILT (semantische Segmentierung) und LabelImg (Objekterkennung) unterstützen nur eine einzige „Bildverarbeitungsaufgabe“. Cellpose erfüllt das Kriterium partiell, da es nur die Annotation im Falle der Instanzsegmentierung ermöglicht, wobei dies in 2D und 3D unterstützt wird. Alle anderen Tools ermöglichen die Annotation von verschiedenen Problemtypen.

Die Kompakttools wie ILT, labelMe und LabelImg sind nicht darauf ausgelegt, direkt ein Modell nach erfolgreicher Annotation zu trainieren. Hingegen ermöglicht Cellpose das Training und die Anwendung von DNNs mittels Application Programming Interface (API), jedoch nicht über ein Graphical User Interface (GUI). Mit den Tools CVAT, hasty.ai und KaIDA ist die direkte „Anwendung“ von DL über eine benutzerfreundliche Oberfläche möglich.

Zudem ist die „Versionierung von Datensätzen“ lediglich in KaIDA möglich.

Mit Ausnahme von hasty.ai sind alle anderen Tools nicht kommerziell und damit ohne Einschränkung zugänglich (vgl. „Zugänglichkeit“).

Des Weiteren sind in allen Tools „GUIs“ inklusive „Benutzerhandbücher“ vorzufinden.

Die Software von hasty.ai bietet eine API, um Teile des Annotationsworkflows zu individualisieren, wie beispielsweise die Verwendung lokaler DNNs für Pre-Annotation. CVAT unterstützt ebenfalls benutzerdefinierte Modelle für die Pre-Annotation. Allerdings ist KaIDA das einzige Tool, das sich auf

die Anpassung und „Erweiterbarkeit“ von Methoden in jedem Modul konzentriert und damit sehr flexibel ist.

Die „Hardware- und Softwareanforderungen“ stellen im Falle ILT und labelMe keine Probleme dar, da direkt ausführbare Programme für verschiedene Betriebssysteme zur Verfügung gestellt werden. Hingegen kann nur eine teilweise Zielerfüllung für LabelImg (ausführbar nur unter Windows, manuelle Installation für andere Betriebssysteme) und Cellpose/KaIDA (manuelle Installation, Graphics Processing Unit (GPU) von Vorteil) festgestellt werden. Mittels eines Webinterfaces im Falle von CVAT und hasty.ai können die Programme direkt ohne zu erfüllende Anforderungen an Hardware bzw. Software genutzt werden.

Zusammenfassend kann festgehalten werden, dass die im Rahmen der Arbeit entwickelte Software KaIDA mit Ausnahme des Kriteriums „Hardware- und Softwareanforderungen“ den genannten Anforderungen gerecht wird. Somit können die alternativen Tools von der in dieser Arbeit entwickelten Software übertroffen werden, sofern alle Kriterien berücksichtigt werden.

5.2.7 Variabilität der Experten

In Abschnitt 2.2.2 ist dargestellt worden, dass Variabilität bei der Annotation von Datensätzen ein Problem in der Praxis darstellt. Die Aussagen decken sich mit den praktischen Erfahrungen, die zu Beginn des interdisziplinären Projekts für personalisierte Onkologie gemacht worden sind.

Die Ergebnisse im Hinblick auf Variabilität der Experten beim Annotieren werden im Nachfolgenden aus zweierlei Gesichtspunkten analysiert. Einerseits wird die **Auswirkung von verrauschten Annotationen auf die DNN-Güte** eruiert. Andererseits soll auf die **Genauigkeit der vorgestellten automatisierten Inspektionspipelines** eingegangen werden. Dabei werden für semantische Segmentierung die Datensätze ISIC-2017-Melanoma und DMA-Spheroid-Hellfeld betrachtet, wohingegen im Falle der Instanzsegmentierung MoNuSeg und LIVECell zur Evaluierung genutzt werden. Die

Basis bilden dazu Ergebnisse aus den im Rahmen der Arbeit entstandenen Publikationen in [Sch22b, Sch22d, Sch23c].

5.2.7.1 Intra-Experten-Variabilität

Die Auswirkungen verschiedener Verunreinigungen auf die DNN-Güte wird über Q_{DSC} bzw. Q_{AJI+} ausgewertet. Über p -Werte¹ eines einseitigen Zweistichproben- t -Tests erfolgt die statistische Untersuchung, inwieweit die Nullhypothese gilt, dass die Ergebnisse mit Verunreinigung gleich oder besser zum Fall ohne Verunreinigung sind. Ferner wird die Detektionsperformance im Rahmen der automatisierten Inspektion von Verunreinigungen über die Metrik Q_{AUC} für verschiedene Parametrierungen der Algorithmen bewertet.

Semantische Segmentierung Die Ergebnisse im Falle von Intra-Experten-Variabilität bei der semantischen Segmentierung finden sich in Tabelle 5.3.

Fast in allen Fällen von *vergessenen* Segmenten, bei einer großen Anzahl von betroffenen Annotationen (steigende Verunreinigungsrate γ) mit *Kompression/Expansion* von Segmenten sowie ausgeprägter *Kontur-Verunreinigung* kommt es zu signifikant ($p < 0.05$) verschlechterten DNNs. Es ist jedoch festzustellen, dass nicht jede Verunreinigung problematisch im Sinne der Leistungsfähigkeit von Netzen ist. *Jitter*, *Löcher* oder *Approximation* führen nicht zu signifikanten Verschlechterungen ($p \gg 0.05$). Teilweise ergeben sich wider Erwarten Verbesserungen. Beispielsweise resultiert im Falle des ISIC-2017-Melanoma-Datensatzes ein Anstieg von $Q_{DSC} = 0.799$ zum Vergleichswert ohne Verunreinigung $Q_{DSC} = 0.793$ bei *Jitter* in Kombination von $\gamma = 0.5$. Die genannten Ausnahmefälle lassen sich unter Umständen als Augmentierung verstehen und dadurch eine Optimierung von Q_{DSC} erklären. Jedoch sind weitere Studien erforderlich, um diese Aussage zu belegen.

¹ $p \geq 0.05$: o, $p < 0.05$: *, $p < 0.005$: **, $p < 0.001$: ***

Tabelle 5.3: Resultate Intra-Experten-Variabilität semantische Segmentierung. Es werden die Auswirkungen verschiedener Verunreinigungen mit variabler Verunreinigungsrate γ bezüglich der Resultate auf Testdaten über die Metrik Q_{DSC} ausgewertet. Die Benchmarkergebnisse im Falle eines nicht verunreinigten Trainingsdatensatzes betragen für ISIC-2017-Melanoma $Q_{DSC} = 0.793$ bzw. für DMA-Spheroid-Hellfeld $Q_{DSC} = 0.897$. Über die p -Werte eines einseitigen Zweistichproben- t -Tests (unter der Annahme gleicher Varianzen) wird bestimmt, inwieweit die Hypothese gilt, dass sich die Güte des DNN durch die Verunreinigung verschlechtert. Signifikant schlechtere DNNs sind **rot** eingefärbt. Ferner wird ein Vergleich der zwischen Detektionsperformance Q_{AUC} im Falle des naiven Ansatzes und optimaler Performance bei Einbezug der Unsicherheit Q_{AUC}^* gezeigt. Überlegene Resultate sind in Fettdruck markiert. Ein Fehlschlagen der Inspektion ($Q_{AUC} \leq 0.55$) ist ebenfalls **rot** markiert [Sch22d].

Datensatz	Metrik	Kompression			Expansion			Jitter			Löcher			Vergessen			Kontur			Approximation				
		γ	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	
ISIC-2017-Melanoma	Q_{DSC} \uparrow	0.796	0.785	0.753	0.808	0.810	0.797	0.794	0.795	0.799	0.797	0.794	0.795	0.786	0.774	0.742	0.792	0.789	0.779	0.802	0.799	0.798		
	$p \uparrow$	o	*	***	o	o	o	o	o	o	o	o	o	*	**	***	o	o	*	o	o	o	o	o
	Q_{AUC} \uparrow	0.872	0.848	0.729	0.798	0.692	0.281	0.671	0.704	0.697	0.700	0.722	0.722	1.000	1.000	1.000	0.983	0.975	0.973	0.629	0.572	0.528		
	Q_{AUC}^* \uparrow	0.890	0.874	0.767	0.810	0.716	0.234	0.703	0.733	0.733	0.764	0.729	0.741	0.999	0.999	0.999	0.986	0.978	0.975	0.604	0.549	0.529		
DMA-Spheroid-Hellfeld	Q_{DSC} \uparrow	0.898	0.889	0.833	0.908	0.892	0.790	0.903	0.907	0.898	0.907	0.905	0.907	0.906	0.864	0.870	0.904	0.905	0.890	0.897	0.897	0.910	0.910	0.910
	$p \uparrow$	o	o	**	o	o	***	o	o	o	o	o	o	o	**	**	o	o	o	o	o	o	o	o
	Q_{AUC} \uparrow	0.933	0.914	0.870	0.931	0.919	0.381	0.911	0.906	0.861	0.909	0.894	0.879	1.000	0.996	0.998	0.924	0.924	0.890	0.751	0.754	0.747		
	Q_{AUC}^* \uparrow	0.933	0.917	0.875	0.930	0.921	0.385	0.919	0.912	0.873	0.913	0.903	0.886	1.000	0.997	0.998	0.923	0.926	0.887	0.824	0.797	0.760		

Tabelle 5.4: Resultate Intra-Experten-Variabilität Instanzsegmentierung. Analog zu Tabelle 5.3 werden hierbei die Resultate für die Instanzsegmentierung gezeigt. Die Auswirkungen verschiedener Verunreinigungen mit variablem Anteil der betroffenen Instanzen pro Bild (β) werden mittels des Kriteriums Q_{AII+} ausgewertet. Vergleichsergebnisse bei einem Training mit nicht verunreinigten Annotationen betragen für MoNuSeg $Q_{AII+} = 0.631$ bzw. für LIVECell $Q_{AII+} = 0.779$. Es wird die Detektionsgenauigkeit des naiven Ansatzes (Q_{AUC}) mit der optimalen Parametrierung des elaborierteren Ansatzes (Q_{AUC}^*) verglichen [Sch22b, Sch23c]. ©2023 IEEE.

Datensatz	Metrik	Kompression			Expansion			Jitter			Löcher			Vergessen			Kontur			Approximation			Verschmelzung			Teilung		
		β	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50	0.10	0.25	0.50		
MoNuSeg	Q_{AII+} \uparrow	0.633	0.628	0.583	0.637	0.599	0.524	0.634	0.637	0.636	0.635	0.641	0.636	0.623	0.620	0.516	0.628	0.593	0.444	0.637	0.641	0.634	0.640	0.628	0.627	0.628	0.631	0.588
	$p \uparrow$	o	o	***	o	**	***	o	o	o	o	o	o	*	o	***	o	**	***	o	o	o	o	o	o	o	o	***
	Q_{AUC} \uparrow	0.861	0.875	0.852	0.732	0.646	0.413	0.668	0.781	0.861	0.649	0.650	0.648	0.696	0.656	0.429	0.873	0.802	0.614	0.505	0.515	0.506	0.834	0.840	0.830	0.854	0.844	0.807
	Q_{AUC}^* \uparrow	0.873	0.885	0.855	0.733	0.647	0.413	0.668	0.781	0.861	0.656	0.656	0.653	0.767	0.712	0.449	0.881	0.805	0.628	0.505	0.515	0.507	0.834	0.841	0.831	0.871	0.851	0.816
LIVECell	Q_{AII+} \uparrow	0.781	0.777	0.737	0.766	0.715	0.564	0.782	0.783	0.780	0.782	0.782	0.782	0.779	0.772	0.692	0.768	0.710	0.392	0.783	0.779	0.771	0.781	0.777	0.732	0.770	0.754	0.703
	$p \uparrow$	o	o	***	*	***	***	o	o	o	o	o	o	o	o	***	*	***	***	o	o	*	o	o	***	*	***	***
	Q_{AUC} \uparrow	0.931	0.935	0.934	0.776	0.762	0.607	0.713	0.807	0.879	0.695	0.703	0.727	0.736	0.673	0.489	0.846	0.813	0.596	0.623	0.643	0.680	0.751	0.753	0.696	0.910	0.909	0.887
	Q_{AUC}^* \uparrow	0.945	0.948	0.942	0.778	0.767	0.617	0.718	0.807	0.879	0.695	0.703	0.727	0.784	0.725	0.558	0.862	0.822	0.636	0.624	0.644	0.681	0.759	0.763	0.700	0.929	0.922	0.902

Betrachtet man die Detektionsperformance Q_{AUC} bzw. Q_{AUC}^* des vorgeschlagenen Algorithmus, so wird klar, dass bis auf zwei Ausnahmefälle für beide Datensätze die entwickelte Methodik nicht fehlschlägt (Q_{AUC} bzw. $Q_{\text{AUC}}^* > 0.55$). Somit ist die vorgeschlagene Methode in der Regel einem Zufallsdetektor ($Q_{\text{AUC}} = 0.5$) deutlich überlegen.

Lediglich im Falle der *Expansion* in Kombination von einer Verunreinigungsrate $\gamma = 0.5$ ist der Ansatz nicht in der Lage, problematische Annotationen zuverlässig zu erkennen. Allerdings wird hierbei die getroffene Annahme von mehrheitlich korrekten Annotationen im verrauschten Datensatz ($\gamma < 0.5$) verletzt. Ferner zeigt sich der Nutzen einer Betrachtung der Unsicherheit in fast allen Fällen von Verunreinigungen durch gesteigerte Genauigkeiten bei der Inspektion ($Q_{\text{AUC}} < Q_{\text{AUC}}^*$).

Instanzsegmentierung Resultate zur Intra-Experten-Variabilität sind für die Instanzsegmentierung in Tabelle 5.4 dargestellt. Ähnlich zu den Ergebnissen bei der semantischen Segmentierung zeigt sich bei Betrachtung der Metrik Q_{AJI^+} unter Einbezug des zugehörigen p -Werts, dass die Verunreinigungen *Kompression*, *Expansion*, *Vergessen*, *Kontur* oder *Teilung* zu signifikant schlechteren DNNs führen ($p < 0.05$).

Dieses Resultat tritt vorwiegend auf, wenn die Verunreinigungen in großem Maße vorherrschen, also im Falle von einer hohen Rate β der betroffenen Instanzen. Als Beispiel kann ein Abfall der Performance von $Q_{\text{AJI}^+} = 0.631$ (keine Verunreinigung) zu $Q_{\text{AJI}^+} = 0.516$ bei der Verunreinigung *Vergessen* mit $\beta = 0.5$ im Falle des MoNuSeg-Datensatzes genannt werden. Ferner stellt die *Verschmelzung* im Falle von LIVECell ebenfalls bei vielen betroffenen Instanzen ein Problem dar ($Q_{\text{AJI}^+} = 0.779 \rightarrow Q_{\text{AJI}^+} = 0.732$). Im Gegensatz können bei *Jitter* oder *Löchern* auch im Falle der Instanzsegmentierung für beide untersuchten Datensätze keine Verschlechterungen festgestellt werden.

Die konzipierte Methodik zum automatisierten Erkennen von verunreinigten Instanzen liegt mehrheitlich über der definierten Schwelle des Fehlschlagens ($Q_{\text{AUC}} \leq 0.55$). Darüber hinaus ist festzustellen, dass der angepasste Ansatz

dem naiven Vorgehen (alleiniger Vergleich zwischen Annotation und Vorhersage des DNN) überlegen ist.

Im Fall von $\beta = 0.5$ kann es zu Fehlern kommen, was gegen die getroffene Annahme von mehrheitlich korrekten Instanzen verstößt. Ferner ist die Inspektion teilweise nicht erfolgreich im Falle der *Approximation*-Verunreinigung ($Q_{AUC} \approx 0.5$ bei MoNuSeg). Bei dieser Fehlerart handelt es sich um eine geringfügige Abweichung im Vergleich zur ursprünglichen korrekten Annotation, wodurch eine Limitierung der entwickelten Methode deutlich wird.

Zudem gibt es keine klare Beziehung zwischen Q_{AUC}/Q_{AUC}^* und β . Einerseits lässt sich dies dadurch erklären, dass das für die Inspektion verwendete DNN oft degeneriert, wenn die Anzahl der verrauschten Annotationen steigt. Andererseits haben korrekte Instanzen, die falsch positiv sind, aufgrund einer größeren Anzahl an verrauschten Instanzen weniger Einfluss auf das Gütekriterium.

Zu beachten ist ferner, dass der ursprüngliche Datensatz als korrekt angenommen wird und das Rauschen lediglich durch die Simulation erzeugt wird. Ergo können kleinere Abweichungen in den Ergebnissen aufgrund von möglicherweise verrauschten Annotationen auftreten, die entgegen der getroffenen Annahme bereits anfänglich im Datensatz enthalten sind.

5.2.7.2 Inter-Experten-Variabilität

Semantische Segmentierung Die Resultate hinsichtlich Inter-Experten-Variabilität im Falle der semantischen Segmentierung finden sich in Tabelle 5.5.

Zunächst ist festzuhalten, dass die Mehrheit von Verunreinigungen zu signifikant schlechteren DNNs führt. Z. B. degeneriert das DNN von ursprünglich $Q_{DSC} = 0.897$ zu $Q_{DSC} = 0.696$ im Falle der Kombination aus *Kompression/Expansion* für den DMA-Spheroid-Hellfeld-Datensatz. Ausnahmen bilden primär die Paarung von *Approximation* und einem halben Datensatz *ohne Verunreinigung* bzw. zum Teil Kombinationen der genannten Typen mit anderen Annotationsstilen ($p > 0.05$).

Tabelle 5.5: Resultate Inter-Experten-Variabilität semantische Segmentierung. Die Auswirkungen von verschiedenen Annotationsstilen sind für die Datensätze ISIC-2017-Melanoma und DMA-Spheroid-Hellfeld gezeigt (vgl. Q_{DSC}). Mittels der p -Werte eines einseitigen Zweistichproben- t -Tests (unter der Annahme gleicher Varianzen) kann die Hypothese überprüft werden, inwieweit sich die Güte des DNN durch die Verunreinigung verschlechtert. Signifikant schlechtere DNNs sind rot hinterlegt. Zudem erfolgt ein Vergleich der Ähnlichkeitsgüte ($Q_{\Gamma_{sim}}$) ohne Berücksichtigung der Unsicherheit bzw. mit Einbezug der Unsicherheit ($Q_{\Gamma_{sim,unc}}$). Überlegene Resultate sind in Fettdruck dargestellt.

Variabilität		ISIC-2017-Melanoma				DMA-Spheroid-Hellfeld			
Annotator A	Annotator B	$Q_{DSC} \uparrow$	$p \uparrow$	$Q_{\Gamma_{sim}} \downarrow$	$Q_{\Gamma_{sim,unc}} \downarrow$	$Q_{DSC} \uparrow$	$p \uparrow$	$Q_{\Gamma_{sim}} \downarrow$	$Q_{\Gamma_{sim,unc}} \downarrow$
Keine	Keine (Baseline)	0.793	-	-	-	0.897	-	-	-
Expansion	Kompression	0.785	*	0.012	0.064	0.696	***	0.063	0.021
Expansion	Kontur	0.749	***	0.059	0.048	0.745	***	0.098	0.151
Expansion	Approximation	0.791	o	0.022	0.021	0.774	***	0.099	0.049
Expansion	Keine	0.799	o	0.028	0.013	0.765	***	0.063	0.008
Kompression	Kontur	0.733	***	0.028	0.032	0.856	*	0.055	0.088
Kompression	Approximation	0.749	***	0.016	0.044	0.802	***	0.071	0.009
Kompression	Keine	0.735	**	0.015	0.055	0.770	***	0.072	0.015
Kontur	Approximation	0.766	**	0.016	0.009	0.887	o	0.046	0.005
Kontur	Keine	0.775	***	0.004	0.036	0.893	o	0.037	0.013
Approximation	Keine	0.799	o	0.053	0.028	0.890	o	0.123	0.052

Ferner ist festzustellen, dass der vorgeschlagene Ansatz zur Inspektion, welcher die unterschiedlichen Annotationsstile von Experten detektiert, sehr genau ist, was sich durch einen maximalen Fehler bezogen auf die vorhergesagte Ähnlichkeit (vgl. $Q_{\Gamma_{sim}}$ bzw. $Q_{\Gamma_{sim,unc}}$) von weniger als 0.1 für die optimalen Varianten zeigt. Zudem ist bei Betrachtung des DMA-Spheroid-Hellfeld-Datensatzes zu sehen, dass der Einbezug der Unsicherheit des DNN, global über alle Experimente betrachtet, zu überwiegend geringeren Fehlern führt. Im Falle des ISIC-2017-Melanoma-Datensatzes ist jedoch $Q_{\Gamma_{sim,unc}}$ im Vergleich zu $Q_{\Gamma_{sim}}$, über alle Varianten betrachtet, ausgeglichen. Es entsteht also für diesen Datensatz weder Vorteil noch Nachteil durch den Einbezug der Unsicherheit.

Instanzsegmentierung In Tabelle 5.6 sind die Ergebnisse zur Inter-Experten-Variabilität für die Instanzsegmentierungs-Datensätze MoNuSeg und LIVECell gezeigt.

Einerseits lässt sich festhalten, dass, mit Ausnahme einer Kombination von *keiner Verunreinigung/Approximation* ($Q_{AJI+} = 0.633$ verglichen zur Benchmark $Q_{AJI+} = 0.631$), grundsätzlich signifikant schlechtere DNNs bei verschiedenen Annotationsstilen folgen ($p < 0.05$). In diesem Ausnahmefall ist

Tabelle 5.6: Resultate Inter-Experten-Variabilität Instanzsegmentierung. Identisch zu Tabelle 5.5 werden hierbei die Ergebnisse für die Instanzsegmentierung anhand der Datensätze MoNuSeg und LIVECell dargestellt. Allerdings wird hierbei Q_{AII+} als Metrik zur Beurteilung des Einflusses unterschiedlicher Annotationsstile verwendet.

Variabilität		MoNuSeg				LIVECell			
Annotator A	Annotator B	$Q_{AII+} \uparrow$	$p \uparrow$	$Q_{\Gamma_{sim}} \downarrow$	$Q_{\Gamma_{sim,unc}} \downarrow$	$Q_{AII+} \uparrow$	$p \uparrow$	$Q_{\Gamma_{sim}} \downarrow$	$Q_{\Gamma_{sim,unc}} \downarrow$
Keine	Keine (Baseline)	0.631	-	-	-	0.784	-	-	-
Expansion	Kompression	0.448	***	0.002	0.030	0.482	***	0.001	0.036
Expansion	Kontur	0.351	***	0.016	0.028	0.310	***	0.020	0.075
Expansion	Approximation	0.490	***	0.051	0.042	0.568	***	0.019	0.004
Expansion	Keine	0.485	***	0.051	0.041	0.581	***	0.024	0.007
Kompression	Kontur	0.157	***	0.077	0.135	0.319	***	0.065	0.120
Kompression	Approximation	0.589	***	0.015	0.004	0.739	***	0.001	0.003
Kompression	Keine	0.576	***	0.018	0.007	0.738	**	0.005	0.004
Kontur	Approximation	0.380	***	0.047	0.010	0.383	***	0.013	0.019
Kontur	Keine	0.440	***	0.044	0.005	0.423	***	0.025	0.008
Approximation	Keine	0.633	o	0.243	0.266	0.776	***	0.086	0.129

es ebenfalls schwierig, die Annotationsstile zu quantifizieren, was sich durch absolute Fehler größer als $Q_{\Gamma_{sim}}/Q_{\Gamma_{sim,unc}} > 0.24$ ausdrückt. Die Tatsache, dass hier nur geringfügige Änderungen an den Annotationen vorliegen, die schwer zu erkennen sind, erklärt dieses Resultat. Für alle anderen Kombinationen können mittels der vorgeschlagenen Inspektions-Pipeline die Annotationsstile ohne große Fehler erfolgreich analysiert werden. Allerdings lässt sich, über alle Experimente betrachtet, kein dominierender Vorteil durch den Ansatz mit Unsicherheit erkennen (vgl. $Q_{\Gamma_{sim,unc}}$ zu $Q_{\Gamma_{sim}}$). Es zeigt sich jedoch auch kein Nachteil durch den Einbezug von Unsicherheit.

5.3 Verarbeitung von Bilddaten im Hochdurchsatz

5.3.1 Prozessierung Makroebene

Die nachfolgenden Ergebnisse im Hinblick auf die Prozessierung innerhalb der Makroebene basieren auf der im Rahmen dieser Arbeit entstandenen Publikation in [Sch21b].

5.3.1.1 Segmentierung von Elementen

Zur Evaluation der Genauigkeit des DNN wird die Bewertungsmetrik Q_{DSC} herangezogen. Es wird die Leistungsfähigkeit bezüglich des DMA-Spot- und Wellplate-Datensatzes auf Testdaten ausgewertet. Dabei kann $Q_{\text{DSC}} = 97.63\%$ für den DMA-Datensatz bzw. $Q_{\text{DSC}} = 96.59\%$ im Falle des Wellplate-Datensatzes erreicht werden. Die leicht reduzierte Metrik im Fall des Wellplate-Datensatzes lässt sich einerseits auf den geringeren Datensatzumfang zurückführen (vgl. Tabelle 5.1). Andererseits stellt diese Bildverarbeitungsaufgabe ein komplexeres Problem im Vergleich zum DMA-Problem dar, weil das DNN beispielsweise mit Schatten in Bildern umgehen muss (vgl. Abbildung 5.1).

Allerdings kann zusammenfassend festgestellt werden, dass das vorgeschlagene DNN eine universelle Methode zur Erkennung von Gitterelementen darstellt, die agnostisch bezüglich der Elementform ist. Dies unterstreicht $Q_{\text{DSC}} > 95\%$, auch bei einem kleinen Trainingsdatensatz im Falle der Wellplate. Die Probleme von etablierten Bildverarbeitungsmethoden wie Hough-Transformationen (manuelle Parametrierung, Beschränkung auf spezielle Formen) können gelöst werden, wodurch sich generische Gitter mit der erforderlichen Genauigkeit detektieren lassen. Ein Nachteil besteht jedoch darin, dass für das überwachte Lernen (vgl. Abschnitt 2.1.2) annotierte Trainingsdatensätze erforderlich sind.

Betrachtet man die Verarbeitungszeit auf handelsüblicher Hardware, so folgt beispielsweise im Falle einer gesamten DMA-Platte mit 672 Elementen ($6379 \text{ px} \times 18992 \text{ px}$) für eine Central Processing Unit (CPU) (Intel Core i7-10750H) 34.52 s, wohingegen bei einer GPU (NVIDIA Quadro RTX 3000) mit 2.41 s eine Beschleunigung von mehr als Faktor zehn möglich ist. Ergo erleichtert die Nutzung einer GPU die Bildanalyse deutlich, ist jedoch nicht zwingend für Anwender erforderlich.

5.3.1.2 Gitterschätzung

Genauigkeit Die entwickelte Methode zur automatisierten Gitterschätzung wird bezüglich zwei kompletten Experimenten pro Gitterart evaluiert. Für die Evaluierung werden die eingeführten Metriken zur Bewertung der korrekten Bestimmung der Elementmittelpunkte $Q_{\Delta p}$ bzw. des Formparameters $Q_{\Delta d}$ (siehe Abschnitt 3.2) herangezogen. In allen sechs verschiedenen Testfällen wurden die Gitterparameter mit ausreichender Genauigkeit geschätzt, was die Metriken quantitativ durch $Q_{\Delta p} \leq 0.053$ bzw. $Q_{\Delta d} \leq 0.037$ belegen. Ferner konnte die Gitterform in allen Fällen korrekt bestimmt werden, wodurch eine Genauigkeit von $Q_{\text{Top-1}} = 100\%$ folgt. Weitere Details zur Gitterschätzung finden sich in verschiedenen Darstellungen unter Anhang G.

Vergleich zum Status Quo Zur Evaluierung der entwickelten Methodik im Vergleich zum Status Quo¹ werden die eingeführten zeitbasierten Metriken $Q_{t_{\text{grid, user}}}$ und $Q_{t_{\text{grid}}}$ sowie Genauigkeitsmetriken ($Q_{\Delta d}$, $Q_{\Delta p}$) genutzt. Es wird ein Vergleich zwischen Gitterschätzung (automatisiert bzw. semi-automatisiert) gegenüber einer manuellen Auswertung präsentiert. Um zuverlässige Aussagen zu gewährleisten, sind zehn Nutzer in die Untersuchung eingeschlossen worden. Als Beispielerperiment wird ein DMA-Experiment mit 672 Elementen (quadratische Form der Elemente) betrachtet. Um den Aufwand im Rahmen des Experiments bei manueller Erkennung zu begrenzen, wurde die durchschnittliche Zeit gemessen, welche für die Markierung und Extraktion von jeweils zehn einzelnen Gitterelementen benötigt wird. Diese wurde dann linear auf 672 Elemente extrapoliert. Analog wurde die Genauigkeitsbetrachtung der Gitterschätzung ebenfalls auf eine Teilmenge von zehn Elementen reduziert.

Die Tabelle 5.7 zeigt die erzielten Resultate im Hinblick auf die Gitterschätzung. Für eine manuelle Auswertung wird $Q_{t_{\text{grid, user}}} = 835$ s benötigt, wobei

¹ Der Status Quo bezeichnet in diesem Kontext eine manuelle Bestimmung des Gitters.

Tabelle 5.7: Bewertung der Gitterschätzung. Der erforderliche Zeitaufwand für Nutzer beschrieben durch $Q_{t_{\text{grid, user}}}$, die Gesamtverarbeitungszeit $Q_{t_{\text{grid}}}$ sowie die eingeführten Genauigkeitsmetriken ($Q_{\Delta d}$, $Q_{\Delta p}$) werden präsentiert. Es wird ein DMA-Experiment mit 672 Elementen in quadratischer Form betrachtet. Die manuelle Gitterbestimmung wird mit der entwickelten Methodik (automatisiert, semi-automatisiert) verglichen. In Fettdruck sind die überlegenen Methoden markiert [Sch21b].

	Manuell	Entwickelte Methoden	
		semi-automatisiert	automatisiert
$Q_{t_{\text{grid, user}}}$ ↓ in s	835	61	< 0.01
$Q_{t_{\text{grid}}}$ ↓ in s	835	61.01	3.88¹/35.99²
$Q_{\Delta p}$ ↓	0.01	0.02	0.01
$Q_{\Delta d}$ ↓	0.02	0.02	0.03

¹ GPU (NVIDIA Quadro RTX 3000)

² CPU (Intel Core i7-10750H)

keine zusätzliche Berechnungszeit anfällt. Damit entspricht der aufgeführte Zahlenwert der Gesamtverarbeitungszeit $Q_{t_{\text{grid}}}$. Für die entwickelte semi-automatisierte Methode wird $Q_{t_{\text{grid, user}}} = 61$ s erzielt. Die Gesamtverarbeitungszeit beträgt in diesem Fall $Q_{t_{\text{grid}}} = 61.01$ s, da die Berechnungszeit mit 0.01 s gering ist. Es folgt damit eine Zeitersparnis um mehr als Faktor 13 im Vergleich zur manuellen Gitterbestimmung.

Der vollautomatisierte Ansatz erfordert quasi keinen Aufwand für Nutzer ($Q_{t_{\text{grid, user}}} < 0.01$ s). Die Gesamtverarbeitungszeit hängt jedoch stark von der Hardware ab. Im Falle einer handelsüblichen GPU (NVIDIA Quadro RTX 3000) beträgt diese $Q_{t_{\text{grid}}} = 3.88$ s, wohingegen die Verarbeitung ohne GPU (Verwendung einer CPU: Intel Core i7-10750H) mit $Q_{t_{\text{grid}}} = 35.99$ s erheblich länger dauert. Die erhöhte Rechenzeit liegt primär an den aufwändigen Berechnungen innerhalb des genutzten DNN zur Erkennung der Gitterelemente. Folglich sind die entwickelten automatisierten und semi-automatisierten Ansätze bezogen auf den Zeitaufwand immer der manuellen Gitterbestimmung überlegen.

Ferner sind im Hinblick auf die Genauigkeit der Verfahren ($Q_{\Delta d}$, $Q_{\Delta p}$) keine beachtlichen Unterschiede zwischen den verschiedenen Varianten ersichtlich (siehe Tabelle 5.7). Der normierte relative Fehler des Formparameters $Q_{\Delta d}$ liegt bei maximal 3 %. Ferner ist die auf den Gitterabstand normierte Abweichung der Gitterzentroide $Q_{\Delta p} \leq 2$ %.

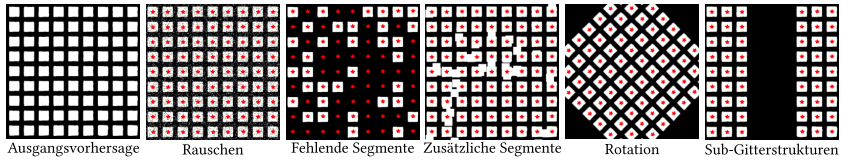
Damit ist die entwickelte Methode zur Gitterschätzung bezogen auf alle Kriterien der leistungsfähigste Ansatz, da der Aufwand für Benutzer reduziert werden konnte, ohne Einbußen in puncto Genauigkeit zu verzeichnen. Es gilt $Q_{\Delta p} \ll 0.1$ bzw. $Q_{\Delta d} \ll 0.1$, wodurch die in Abschnitt 3.2 empirisch definierten Schwellen zum Fehlschlagen der Detektion in keinem Fall überschritten werden. Folglich kann mittels der entwickelten Methoden ein wichtiger Beitrag zur schnellen Auswertung von HDEs gemacht werden.

5.3.1.3 Robustheit und Anwendungsgrenzen

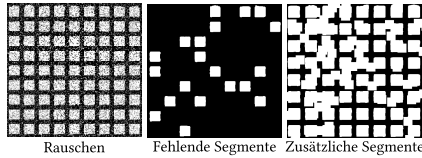
Auszüge aus den Untersuchungen im Hinblick auf die Robustheit und jeweilige Anwendungsgrenzen finden sich in Abbildung 5.11. In Abbildung 5.11a sind synthetische Veränderungen gezeigt, bei welchen die entwickelte Methodik zur automatisierten Gitterschätzung erfolgreich ist. Im Gegensatz dazu findet sich in Abbildung 5.11b eine Übersicht von Fällen, in welchen der Algorithmus nach synthetischer Veränderung der Ausgangsvorhersage fehlschlägt.

Ein Rauschpegel von 30 % führt zu Fehlern, was jedoch im Falle des verwendeten Convolutional Neural Network (CNN) U-Net zur Detektion der Elemente sehr unwahrscheinlich ist. Mittels der im Netzwerk enthaltenen Faltungsschichten werden die Vorhersagen geglättet. Zudem führen mehr als 30 % fehlende Segmente zu einem Problem, da nicht mehr ausreichend Repräsentanten für die Zeilen-/Spaltenbestimmung im Rahmen des vorgeschlagenen Clusterverfahrens Density-Based Spatial Clustering of Applications with Noise (DBSCAN) vorhanden sind. Außerdem resultiert ein Fehlerfall durch das Hinzufügen von mehr als 80 % Zusatzelementen, da Zeilen/Spalten zu sehr verschmelzen und damit nicht mehr differenziert werden können. Mittels einer Rotation oder durch das Generieren von Sub-Gitterstrukturen konnte die automatisierte Gitterschätzung nicht zum Fehlschlagen gebracht werden.

Zudem stellt Abbildung 5.12 als Ergänzung die korrespondierenden quantitativen Analysen zu den genannten Untersuchungsfällen im Kontext von Robustheit bzw. Anwendungsgrenzen der Gitterschätzung dar. Dazu sind die



(a) Robustheit. Die Gittervorhersage ist trotz Verunreinigungen im Vergleich zur ursprünglichen Vorhersage mit Rauschen, gelöschten oder hinzugefügten Segmenten, gedrehter Segmentierungsmaske oder erzeugten Sub-Gitterstrukturen möglich. Rote Kreuze stellen dabei die prädierten Mittelpunkte der Gitterschätzung dar.



(b) Anwendungsgrenzen. Fehlerfälle treten bei einem Rauschpegel von 30 %, bei 30 % fehlenden Segmenten oder im Falle von 80 % zusätzlichen Segmenten auf.

Abbildung 5.11: Robustheit und Anwendungsgrenzen der Gitterschätzung. Robustheit (a) und Anwendungsgrenzen (b) werden bezüglich unterschiedlicher synthetischer Veränderungen gegenübergestellt. Aktualisierte Version auf Basis von [Sch21b].

Metriken $Q_{\Delta p}$ und $Q_{\Delta d}$ für unterschiedlich starke Veränderungen gezeigt. Das Beispiel aus Abbildung G.3b wird für die Analyse verwendet.

Aus den Ergebnissen in Abbildung 5.11 bzw. Abbildung 5.12 kann zusammenfassend eine große Robustheit des Verfahrens für gewöhnliche Verunreinigungen in den Vorhersagen nachgewiesen werden.

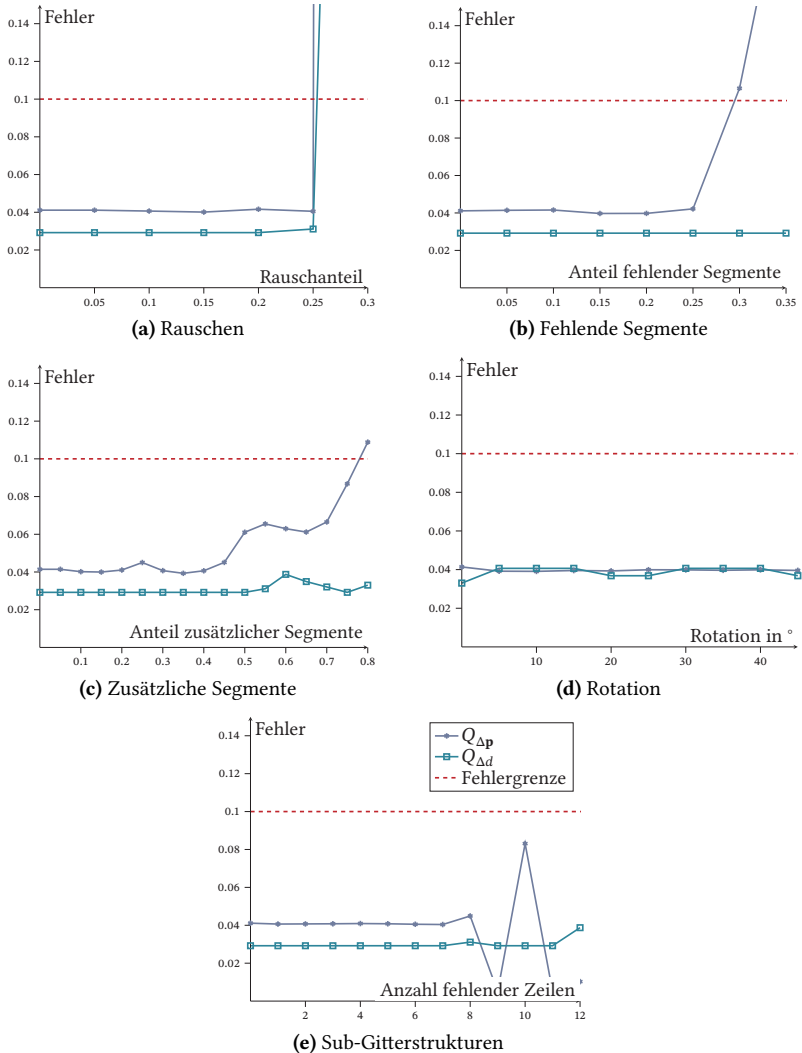


Abbildung 5.12: Quantitative Analyse der Anwendungsgrenzen der Gitterschätzung. Die Untersuchungsfälle „Rauschen“, „fehlende Segmente“, „zusätzliche Segmente“, „Rotation“ und „Sub-Gitterstrukturen“ sind bezüglich verschiedener Stärken der Veränderung gezeigt. Dazu sind die Qualitätsmetriken $Q_{\Delta P}$ und $Q_{\Delta d}$ zur Bewertung der Gitterschätzung dargestellt. Die Analyse bezieht sich dabei auf das Beispiel in Abbildung G.3b.

5.3.1.4 Software-Tool Grid Screener

Ein Vergleich im Hinblick auf Software-Tools für die Hochdurchsatz (HD)-Verarbeitung basierend auf den eingeführten Kriterien in Abschnitt 3.2 findet sich in Tabelle 5.8. Es erfolgt eine Gegenüberstellung der Programme Fiji [Sch12], PlantCV [Geh17], CellProfiler [Car06], EVOS-Pipeline [Kli20] und HTS-Corrector [Mak06] zum im Rahmen dieser Arbeit entwickelten Programm Grid Screener [Sch21b].

Tabelle 5.8: Vergleich Software-Tools HD-Verarbeitung. Fiji, PlantCV, CellProfiler, EVOS-Pipeline, HTS-Corrector und Grid Screener werden anhand der in Abschnitt 3.2 eingeführten Bewertungskriterien gegenübergestellt. Dabei wird zwischen vollständiger (✓), partieller (x/✓) und fehlender (x) Zielerfüllung differenziert [Sch21b].

Kriterium	Tools					
	Fiji [Sch12]	PlantCV [Geh17]	CellProfiler [Car06]	EVOS-Pipeline [Kli20]	HTS-Corrector [Mak06]	Grid Screener [Sch21b]
Bildverarbeitung	✓	✓	✓	✓	x	✓
Gittererkennung	x	x/✓	x/✓	(✓)	x	✓
Elementweise Prozessierung	✓	✓	✓	x/✓	x/✓	✓
Zugänglichkeit	✓	✓	✓	✓	✓	✓
GUI/Handbuch	✓	✓	✓	✓	✓	✓
Erweiterbarkeit	✓	✓	✓	x	x	✓
Software- und Hardwareanforderungen	x/✓	x/✓	x/✓	x/✓	x/✓	x/✓

Zunächst ist festzustellen, dass alle Programme, mit Ausnahme von HTS-Corrector, „Bildverarbeitung“ prinzipiell unterstützen.

Eine vollautomatisierte „Gittererkennung“ ist lediglich über den im Rahmen dieser Arbeit entwickelten Grid Screener möglich. CellProfiler und PlantCV ermöglichen eine halbautomatisierte Gitterschätzung, die jedoch keine Drehwinkelkorrektur beinhaltet. Ferner ist im Falle der EVOS-Pipeline die Bestimmung von Elementen halbautomatisiert, da der grobe Ort eines Elements manuell eingegeben werden muss.

Im Falle von Fiji, PlantCV, CellProfiler und Grid Screener sind unterschiedliche Verfahren zur „elementweisen Prozessierung“ verfügbar. Konträr dazu

ermöglicht die EVOS-Pipeline lediglich das Zählen von Zellen bzw. der HTS-Corrector erfordert bereits extrahierte Merkmale aus den Bildern für statistische Analysen, da die Bildverarbeitung nicht direkt vom Tool unterstützt wird.

Alle aufgeführten Softwarepakete sind frei zugänglich und beinhalten GUIs inkl. Benutzerhandbücher (vgl. „Zugänglichkeit“).

Mit Ausnahme von den Programmen HTS-Corrector und EVOS-Pipeline ist die „Erweiterbarkeit“ bei allen anderen Tools gegeben.

Da bei der Software HTS-Corrector lediglich Windows als Betriebssystem unterstützt wird, kann nur eine teilweise Erfüllung des Ziels bezüglich „Hardware- und Softwareanforderungen“ attestiert werden. Fiji und CellProfiler sind PlantCV bzw. Grid Screener durch die Bereitstellung direkter ausführbarer Programme im Vergleich zur Bereitstellung von pip-Paketen überlegen, da python vorausgesetzt wird. Allerdings folgt für alle Tools eine nur partielle Zielerfüllung, da im Falle der Verwendung von DL leistungsfähige Hardware, wie eine GPU, für das schnelle Prozessieren notwendig ist. Die EVOS-Pipeline setzt CellProfiler als Basisinstallation voraus, wodurch der direkte Einsatz auch nicht möglich ist.

Somit kann bei der Berücksichtigung aller Kriterien die Überlegenheit des entwickelten Grid Screener zu den Vergleichswerkzeugen festgehalten werden.

5.3.2 Prozessierung Mikroebene

Im Nachfolgenden erfolgt eine Vorstellung der Ergebnisse unterschiedlicher Methoden im Bereich der elementweisen Auswertung in der Mikroebene. Dabei wird auf verschiedene heterogene Bilddatensätze (vgl. Abschnitt 5.1) zurückgegriffen. Die Ergebnisse basieren auf der im Rahmen der Arbeit entstandenen Veröffentlichung in [Sch23b].

Seed Detektion Zunächst sollen die verschiedenen Methoden (Blob, Otsu, DL) auf vier Testdatensätzen bezüglich der Metrik Q_{DSC} verglichen werden.

Es ist anzumerken, dass sich hierbei Q_{DSC} , angelehnt an [Sch20b], auf eine Auswertung bezüglich der Detektion und nicht bezüglich der Segmentierungsmaske bezieht. In Tabelle 5.9 sind die Ergebnisse für die Datensätze DMA-Zellen-Fluoreszenz, Insects2022, LIVECell und Microglia2023 dargestellt [Sch23b].

Es zeigt sich deutlich, dass ein DL-basierter Ansatz für alle Datensätze die überlegene Methode darstellt. Zur qualitativen Veranschaulichung der Leistungsfähigkeit der trainierten DNNs im Vergleich zu den Methoden Otsu und Blob sind die Vorhersagen für jeweils ein Beispielbild pro Datensatz in Abbildung 5.13 illustriert.

Tabelle 5.9: Vergleich Seed Detektion. Die Güte der Methoden Blob, Otsu und DL wird anhand der Metrik $Q_{DSC} \uparrow$ für die Datensätze DMA-Zellen-Fluoreszenz, Insects2022, LIVECell und Microglia2023 evaluiert. Die überlegene Methode ist jeweils mittels Fettdruck markiert [Sch23b]. Eine beispielhafte Visualisierung der Ergebnisse findet sich in Abbildung 5.13.

Method	DMA-Zellen-Fluoreszenz	Insects2022	LIVECell	Microglia2023
Blob	0.384	0.051	0.506	0.060
Otsu	0.413	0.089	0.158	0.151
DL	0.850	0.775	0.841	0.904

In den Vorhersagen der Verfahren nach Otsu bzw. über Blob sind deutlich erkennbare Fehler vorzufinden. Konträr dazu können unter der Verwendung von DL komplexe Bildverarbeitungsaufgaben gelöst werden. Beispiele sind das Erkennen von Insekten oder der Immuneffektorzellen Mikroglia sowie die Unterscheidung von sich berührenden Zellen. Nutzer profitieren ferner durch das Bereitstellen der Seed-Detektion-Analysepipeline von einem reduzierten Annotationsaufwand zur Erzeugung eines annotierten Datensatzes, da lediglich die Mittelpunkte der Instanzen anstelle von Konturen markiert werden müssen.

Segmentierung Es soll detaillierter auf die Ergebnisse der Segmentierungen für jeweils zwei Datensätze pro Segmentierungsaufgabe eingegangen werden [Sch23b]. Ein quantitativer Vergleich zwischen dem klassischen Bildverarbeitungsalgorithmus Otsu und DL findet sich in Tabelle 5.10 für die Datensätze DMA-Zellen-Fluoreszenz, Insects2022, DMA-Spheroid-Hellfeld

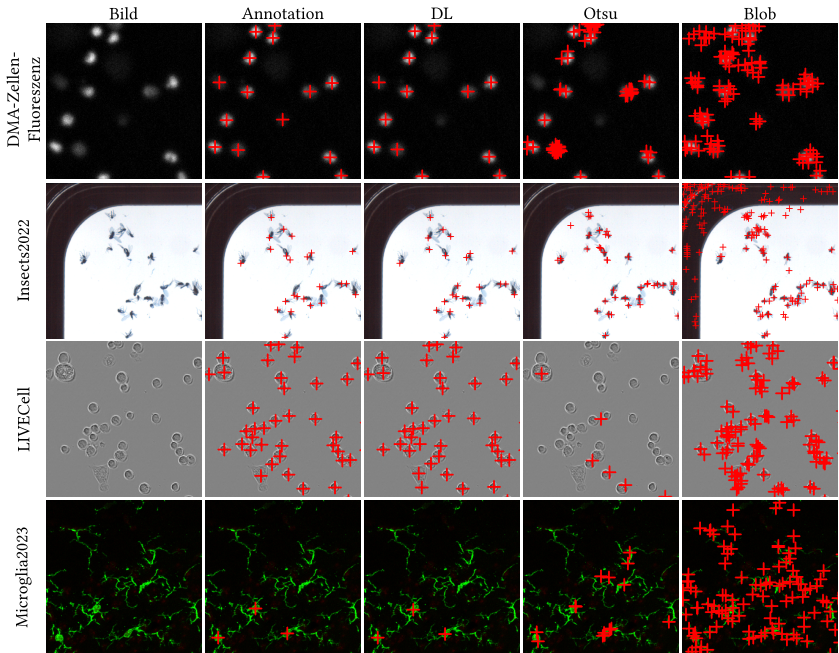


Abbildung 5.13: Vorhersagen der Seed-Detektion-Analysepipeline. Für ein Beispielbild der vier Datensätze (DMA-Zellen-Fluoreszenz, Insects2022, LIVECell, Microglia2023) sind das Bild inklusive Annotation sowie die Resultate der verschiedenen Methoden (DL, Otsu, Blob) der Seed-Detektion-Analysepipeline gezeigt. Rote Kreuze markieren dabei die Mittelpunkte der detektierten Objekte [Sch23b]. Es ist anzumerken, dass keine der Methoden eine Nachverarbeitung integriert hat, die Vorwissen miteinbezieht.

und DMA-Sphäroid-Fluoreszenz, die aus unterschiedlichen Praxisprojekten stammen. Bei einem Vergleich der Datensätze, dargestellt in Abschnitt 5.1, können diese in absteigender Reihenfolge hinsichtlich deren Komplexität¹ eingeordnet werden: Insects2022 > DMA-Sphäroid-Hellfeld > DMA-Sphäroid-Fluoreszenz > DMA-Zellen-Fluoreszenz.

¹ Die Sortierung ist empirisch erfolgt, indem die Bilder der Datensätze mit der Zielstellung, die in der Annotation dargestellt ist, verglichen worden sind.

Tabelle 5.10: Vergleich Segmentierung. Die Güte der Methoden Otsu und DL wird anhand des Kriteriums $Q_{DSC} \uparrow$ im Falle der semantischen Segmentierung bzw. mittels der Metrik $Q_{All+} \uparrow$ für die Instanzsegmentierung quantifiziert, wobei die überlegene Methode in Fettdruck markiert ist. Es werden die Testbilder der Datensätze DMA-Zellen-Fluoreszenz, Insects2022, DMA-Spheroid-Hellfeld und DMA-Spheroid-Fluoreszenz analysiert [Sch23b]. Exemplarische Veranschaulichungen der Methoden sind in Abbildung 5.14 gezeigt.

Method	DMA-Zellen-Fluoreszenz	Insects2022	DMA-Spheroid-Hellfeld	DMA-Spheroid-Fluoreszenz
Otsu	0.582	0.054	0.143	0.766
DL	0.676	0.481	0.925	0.916

Es zeigt sich, dass DNNs für alle Datensätze die überlegene Methode darstellen. Mit steigender Komplexität der Aufgabe, wie es bei dem Insects2022- oder DMA-Spheroid-Hellfeld-Datensatz der Fall ist, zeigt sich eine größere Überlegenheit von DL zum Otsu-Verfahren.

Um die Güte der DNNs im Vergleich zum Otsu-Schwelwertverfahren zudem qualitativ bewerten zu können, sind in Abbildung 5.14 jeweils Vergleiche zwischen Annotation und Prädiktionen der Methoden für ein Beispielbild der vier Datensätze gezeigt. Der visuelle qualitative Vergleich untermauert die Ergebnisse aus Tabelle 5.10. Im Falle der semantischen Segmentierung ist die DL-Pipeline fähig, erfolgreich zwischen Bildaufnahmeartefakten am Rand der Elemente und Sphäroiden (kreisförmige Elemente in der Nähe des Bildzentrums) zu differenzieren, wodurch die Annotationen und Prädiktionen kaum abweichen. Im Gegensatz dazu segmentiert das Otsu-Schwelwertverfahren die Randbereiche fehlerhaft.

Bei einer Betrachtung der Instanzsegmentierungsaufgaben ist festzuhalten, dass der DMA-Zellen-Fluoreszenz-Datensatz mit hoher Genauigkeit vom DNN gelöst wird. Allerdings gelingt diese Bildverarbeitungsaufgabe ebenfalls mit dem Otsu-Verfahren, welches lediglich Fehler bei sich berührenden Zellen aufweist. Andererseits ist im Falle des Insects2022-Datensatzes zu erkennen, dass beide Verfahren diesen Datensatz nicht vollständig korrekt lösen. Zwar weist das DL-Verfahren eine deutlich höhere Güte als das Otsu-Verfahren auf, dennoch werden zum Teil kleinere Insekten nicht korrekt vom DNN erkannt.

Eine Erweiterung des Datensatzumfangs oder eine Anpassung der Schwellwerte im Post-Processing sind Optionen für zukünftige Optimierungen. Meist skaliert der erforderliche Annotationsumfang für das erfolgreiche Lösen einer

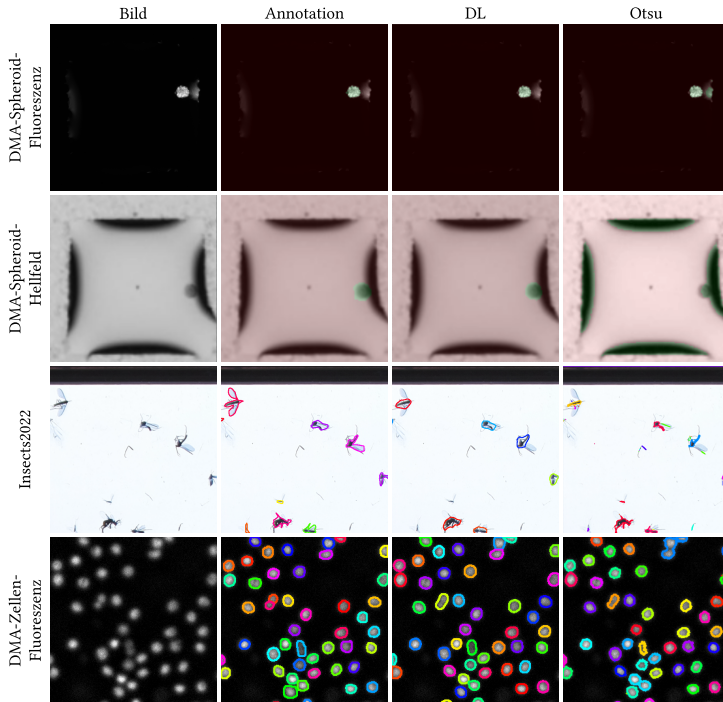
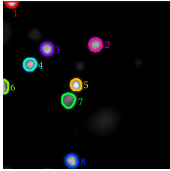


Abbildung 5.14: Vorhersagen der Segmentierung. Für jeweils ein Beispiel aus den verschiedenen Datensätzen (DMA-Spheroïd-Hellfeld, DMA-Spheroïd-Fluoreszenz, Insects2022, DMA-Zellen-Fluoreszenz) ist ein Vergleich zwischen originalem Bild, Annotation und Prädiktion des DNN bzw. Otsu-Schwelleverfahren gezeigt. Segmente sind dabei hellgrün hinterlegt, wohingegen der Hintergrund hellrot markiert ist. Instanzen sind über verschiedene Farbkonturen dargestellt [Sch23b].

Bildverarbeitungsaufgabe mit deren Komplexität. Ferner muss als Limitierung festgehalten werden, dass alle DL-basierten Ansätze einen annotierten Datensatz erfordern. Damit resultiert ein erhöhter Aufwand für Nutzer im Vorfeld, verglichen zum parameterlosen Verfahren nach Otsu.

Zudem findet sich in Tabelle 5.11 ein Beispiel für eine Merkmalsliste $\mathcal{F}_{i,j}$ des gezeigten Elements $\mathbf{x}_{i,j}$ aus dem DMA-Zellen-Fluoreszenz-Datensatz, um die

Tabelle 5.11: Exemplarische Merkmalsliste $\mathcal{F}_{i,j}$. Für das links dargestellte Element $\mathbf{x}_{i,j}$ aus dem DMA-Zellen-Fluoreszenz-Datensatz sind die Merkmale $\mathcal{F}_{i,j}$ (Fläche, Exzentrizität, mittlerer Grauwert, Längen der Hauptachsen/Nebenachsen) der Instanzen extrahiert (Tabelle rechts) [Sch23b].



ID	Fläche	Exzentrizität	Mittlerer Grauwert	Hauptachse	Nebenachse ...
1	153	0.835	136.784	19.394	10.683
2	320	0.238	110.491	20.482	19.893
3	304	0.304	121.214	20.16	19.206
4	296	0.347	113.936	20.054	18.81
5	246	0.399	140.801	18.496	16.962
6	169	0.902	104.538	22.767	9.835
7	363	0.38	83.59	22.412	20.735
8	305	0.476	121.069	21.018	18.486

individuelle Extraktion der für die jeweilige Anwendung relevanten Merkmale zu zeigen. Es sind hierbei beispielhaft die Merkmale Fläche, Exzentrizität, mittlerer Grauwert sowie die Längen der Hauptachsen/Nebenachsen pro Instanz dargestellt, die automatisiert generiert und gespeichert werden. Aus der Tabelle 5.11 wird deutlich, dass im Falle von vielen Objekten bzw. zu charakterisierenden Merkmalen ein vollautomatisierter Ansatz zur schnellen und reproduzierbaren Auswertung der Experimente, z. B. wie in [Cui23], notwendig ist.

Zusammenfassend kann festgehalten werden, dass mit der Integration von DL eine generische Segmentierungsmethode für unterschiedliche Prozessierungen im Rahmen der Mikroebene bereitgestellt wird. Die Methode zeichnet sich zudem durch das Bereitstellen von beliebigen Merkmalslisten aus. Dadurch können HDEs mit erhöhter Genauigkeit sowie Reproduzierbarkeit im Vergleich zu klassischen Methoden wie Otsu automatisiert und ganzheitlich ausgewertet werden.

Mehrdimensionale Instanzanalyse Im Weiteren sollen die Resultate der entwickelten Methode im Kontext der mehrdimensionalen Instanzanalyse für das Beispiel „Ermittlung der Zellviabilität“ diskutiert werden. Diese gibt den Anteil lebender Zellen zur Gesamtzellanzahl an. Die Gütebestimmung der Instanzsegmentierung, die einen Teilprozessierungsschritt der Gesamtmethode darstellt, ist bereits in Tabelle 5.10 quantifiziert und diskutiert worden.

Abbildung 5.15 zeigt die exemplarischen Ergebnisse der in Abschnitt 3.5.3 entwickelten Analysepipeline für einen Bildausschnitt. Es ist anzumerken, dass ein Gesamtelement $\mathbf{x}_{i,j}$ um Faktor 64 größer ist, worin jedoch eine Diskussion der Details bzw. eine geeignete Visualisierung des Gesamtbilds erschwert ist.

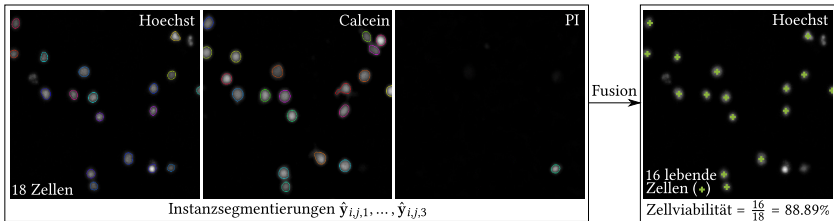


Abbildung 5.15: Ergebnisse Zellviabilität. Links sind die Instanzsegmentierungen $\hat{\mathbf{y}}_{i,j,1}, \dots, \hat{\mathbf{y}}_{i,j,3}$ für die Einfärbungen Hoechst, Calcein und PI über eine Markierung der Konturen gezeigt. Rechts sind alle als lebend eingestuft Zellen mittels grünen Kreuzen markiert, wodurch final auf die Zellviabilität geschlossen werden kann [Sch23b]. Die Darstellung dient zur beispielhaften Visualisierung der Funktionsweise des entwickelten Algorithmus (vgl. Abschnitt 3.5.3). Zudem zeigt das Schaubild auch die Komplexität einer manuellen mehrdimensionalen Bildanalyse, da der Vergleich zwischen den Bildern schwierig ist.

Die Resultate der Instanzsegmentierungen $\hat{\mathbf{y}}_{i,j,1}, \dots, \hat{\mathbf{y}}_{i,j,3}$ der Einfärbungen Hoechst, Calcein und PI decken sich mit den Ergebnissen aus dem vorherigen Abschnitt. Mittels DNNs können die Zellkerne robust in allen Farbkanälen erkannt werden. Die Betrachtung des Fusionschritts zeigt, dass lediglich Zellen als lebend kategorisiert werden, wenn diese in Hoechst und gleichzeitig Calcein erkannt worden sind, allerdings eine Detektion in PI ausbleibt. Für den exemplarischen Elementausschnitt ist zudem die Zellviabilität als Quotient aus lebenden Zellen und Zellen im Hoechst-Kanal zur Illustration berechnet worden.

Folglich kann mit der entwickelten Bildverarbeitungspipeline die Zellviabilität automatisiert bestimmt werden. Betrachtet man im Vergleich eine manuelle Analyse, so müssen für ein beispielhaftes DMA-Einzelexperiment in Summe 129024 (64 Segmente pro Bild \cdot 3 Kanäle \cdot 672 Elemente) der gezeigten Ausschnitte analysiert werden. Zudem wäre eine gleichzeitige, mehrdimensionale Überlagerung zwischen den Kanälen notwendig. Die entwickelte Methode stellt damit einen großen Benefit für Nutzer in der Praxis dar, da eine

manuelle Analyse aus oben beschriebenen Gründen nicht praktikabel sowie schwer reproduzierbar ist.

Farb- und Intensitätsanalyse Im Weiteren soll auf beispielhafte Fragestellungen aus Experimenten der Praxis eingegangen werden, um die Relevanz der Analyse von Farbverläufen oder Intensitäten für die Auswertung von HDEs zu verdeutlichen.

Zunächst sollen die Resultate der entwickelten Methode zur Farbanalyse diskutiert werden. Abbildung 5.16 (a) zeigt ein Hellfeldbild und Abbildung 5.16 (b) die zugehörigen errechneten Quantifizierungen, die über den Sättigungskanal im Hue-Saturation-Value (zu Deutsch: Farbwert, Farbsättigung, Helligkeit) (HSV)-Farbraum generiert worden sind. Es gilt, eine Quantifizierung zu finden, mit der die potentia Hydrogenii (pH)-Werte automatisiert über den Farbumschlag (transparent zu orange) ermittelt werden können.

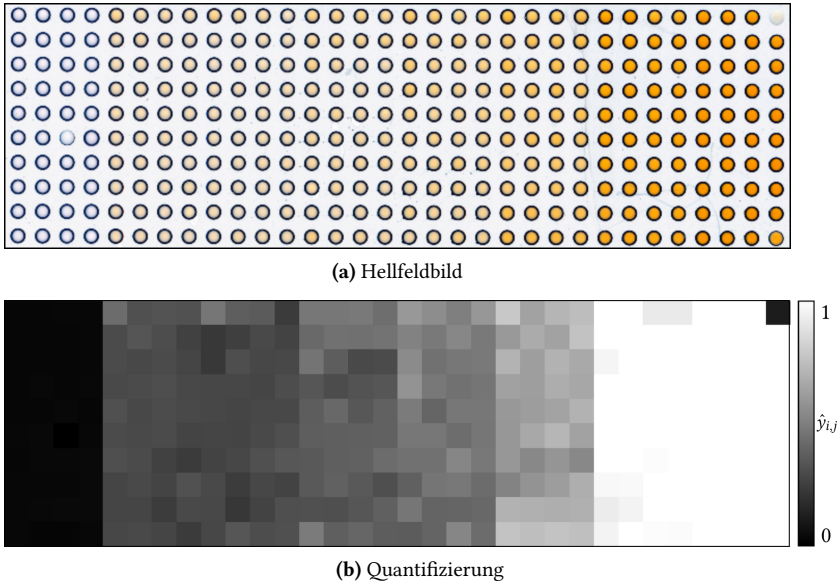


Abbildung 5.16: Farbanalyse DMA I. Das Hellfeldbild (a) und die Quantifizierung pro Element $\hat{y}_{i,j}$ (b), codiert über Grauwerte, sind visualisiert (weiß: $\hat{y}_{i,j} \rightarrow 0$, orange: $\hat{y}_{i,j} \rightarrow 1$) [Sch23b].

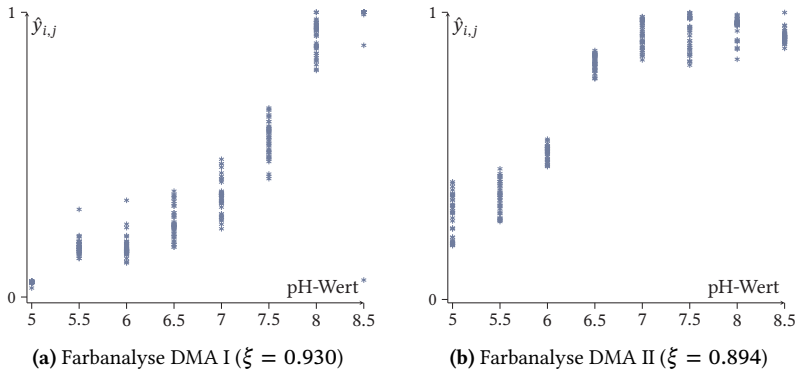


Abbildung 5.17: Vergleich zwischen Quantifizierung und pH-Wert. Für die Experimente Farbanalyse DMA I (a) und Farbanalyse DMA II (b) wird der Zusammenhang zwischen Quantifizierung $\hat{y}_{i,j}$ und pH-Wert gezeigt. Zudem ist die lineare Korrelation beider Größen über ξ dargestellt [Sch23b].

Da der Zusammenhang zwischen Readout (Farbe) und pH-Wert durch eine Kalibrierungsmessung bekannt ist, kann für eine Evaluierung der Ergebnisse die Korrelation zwischen Quantifizierung $\hat{y}_{i,j}$ und gemessenem pH-Wert berechnet werden. Die Zusammenhänge sind in Abbildung 5.17 (a) illustriert. Es folgt als Korrelationskoeffizient $\xi = 0.930$ für das in Abbildung 5.16 gezeigte Experiment. Ferner kann bei einem visuellen Vergleich zwischen Hellfeldbild in Abbildung 5.16 (a) und Quantifizierung in Abbildung 5.16 (b) deren Korrelation festgestellt werden. Es zeigt sich deutlich, dass der Farbumschlag über die berechnete Kennzahl $\hat{y}_{i,j}$ charakterisiert werden kann. Somit ist eine automatisierte Experimentauswertung zur Ermittlung des pH-Werts unter Verwendung der entwickelten Analyseverfahren möglich.

Der Farbverlauf in Abbildung 5.18 unterscheidet sich zu Abbildung 5.16 in der Art des Farbumschlags, da ein Wechsel von gelb zu violett numerisch beschrieben werden soll.

Im Rahmen dieser Aufgabenstellung basiert die Quantifizierung auf einer Transformation, die den blauen Farbkanal positiv bzw. den grünen Farbkanal negativ überlagert, wobei beide Signale vorab normiert werden. Analog

finden sich die Zusammenhänge in Abbildung 5.17 (b). Es folgt ein Korrelationskoeffizient von $\xi = 0.894$. Demnach ist es ebenfalls möglich, den Farbumschlag automatisiert zu quantifizieren. Allerdings ist die Güte im Vergleich zum Experiment in Abbildung 5.16 minimal geringer. Eine Erklärung dazu ist die größere Komplexität im Falle des Farbumschlags zwischen zwei Farben im Vergleich zur Farbveränderung von transparent zu einer Farbe. Die Gegenüberstellung von Farbverlauf bzw. Quantifizierung $\hat{y}_{i,j}$ zeigt hier analog, dass durch die angegebene Transformation eine erfolgreiche quantitative Beschreibung des Farbumschlags über die bestimmten Kennzahlen möglich ist.

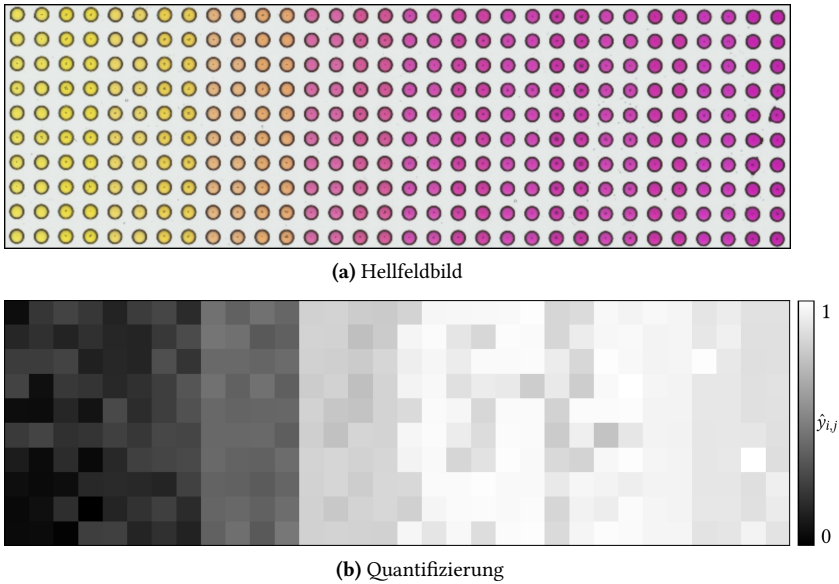
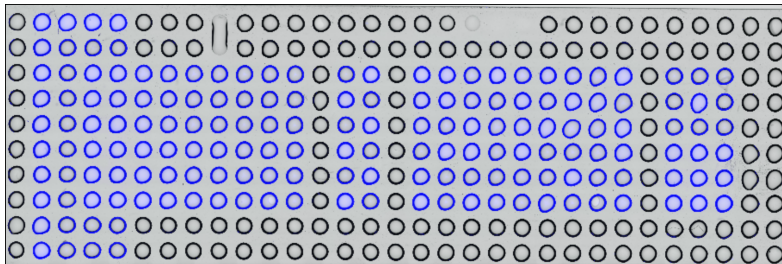


Abbildung 5.18: Farbanalyse DMA II. Analog zu Abbildung 5.16 sind das Hellfeldbild (a) und die korrespondierende Quantifizierungen $\hat{y}_{i,j}$ (b) dargestellt (gelb: $\hat{y}_{i,j} \rightarrow 0$, violett: $\hat{y}_{i,j} \rightarrow 1$) [Sch23b].

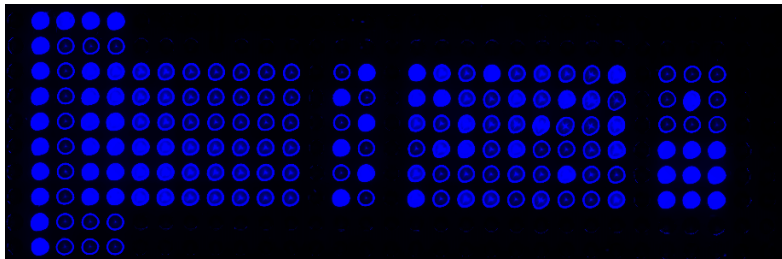
Zusammenfassend kann festgehalten werden, dass die Transformationsfunktion zur Umrechnung von RGB-Werten in eine Kennzahl stark von der Problemstellung abhängt und damit der in der Methodik vorgeschlagene variable

Ansatz sinnvoll ist. Der Einsatz von DNNs kann als Alternative zur heuristischen Quantifizierung dienen und sollte daher in zukünftigen Projekten als weitere generische Transformationsfunktion berücksichtigt werden.

Die Ergebnisse der beschriebenen Intensitätsanalyse aus Abschnitt 3.5.3 finden sich in Abbildung 5.19. In der Visualisierung sind ein Hellfeldoverlay, das zugehörige Fluoreszenzsignal sowie die berechnete Quantifizierung pro Element $\hat{y}_{i,j}$ vergleichend gegenübergestellt. Die im Rahmen der Arbeit entstandenen Veröffentlichungen [Sch23b, Wie22] nutzen die Methode beispielsweise zur Erforschung neuer Medikamentenwirkstoffe.

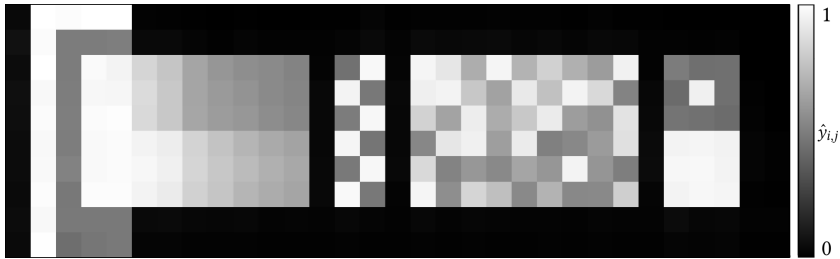


(a) Hellfeldoverlay



(b) Fluoreszenzsignal

Abbildung 5.19: Intensitätsanalyse DMA. Eine Gegenüberstellung von Hellfeldoverlay (a), Fluoreszenzsignal (b) und resultierender Quantifizierung pro Element $\hat{y}_{i,j}$ (c), codiert über Grauwerte, ist gezeigt (kein Fluoreszenzsignal: $\hat{y}_{i,j} \rightarrow 0$, großes Fluoreszenzsignal: $\hat{y}_{i,j} \rightarrow 1$) [Sch23b, Wie22].



(c) Quantifizierung

Abbildung 5.19: Intensitätsanalyse DMA. Fortsetzung.

Da in den hier vorliegenden Experimenten keine Referenz $y_{i,j}$ existiert, werden die Ergebnisse vergleichend analysiert. Der direkte Vergleich zwischen Fluoreszenzsignal und Kennzahlen $\hat{y}_{i,j}$ zeigt klar, dass die sichtbaren Unterschiede im Signal über die quantitative Kennzahl beschrieben werden können. Z. B. folgen bei ausbleibenden Fluoreszenzsignalen Quantifizierungen von $\hat{y}_{i,j} \approx 0$. Ferner zeigen sich größere Kennwerte $\hat{y}_{i,j}$, falls Kreise gefüllt sind im Vergleich zu Kreisen, bei denen lediglich der Rand wahrgenommen werden kann.

Folglich ist auch in diesem Anwendungsfall eine automatisierte Auswertung der HDEs über die vorgeschlagene Analysemethodik möglich.

5.4 Effizientes Lernen

Im Weiteren werden die Methoden im Bereich des „annotations-effizienten Lernens“ aus der Anwendungsperspektive vergleichend gegenübergestellt (siehe Abschnitt 5.4.1 bis 5.4.4). Der Vergleich erfolgt anhand der in Abschnitt 3.6 eingeführten Übergruppen. Dazu werden die in Abschnitt 3.2 vorgestellten Kriterien verwendet. Generell sollen die Ergebnisse jeweils anhand zweier semantischer Segmentierungsdatensätze (ISIC-2017-Melanoma, DMA-Spheroid-Hellfeld) sowie ebenfalls zweier Instanzsegmentierungsdatensätze (MoNuSeg, LIVECell) besprochen werden. Abschließend fasst Abschnitt 5.4.5 die neuen Erkenntnisse zusammen.

5.4.1 Transfer Learning

Um die Größe des Pools an Datensätzen pro Bildverarbeitungsaufgabe bei der Betrachtung von Transfer Learning (TL) zu erhöhen, werden Instanzsegmentierung und semantische Segmentierung nicht separat betrachtet. Der Informationsgehalt ist im Falle einer binären semantischen Segmentierung geringer im Vergleich zu Instanzsegmentierungen. Daher werden alle Annotationen der Instanzsegmentierungsdatensätze in Annotationen für binäre semantische Segmentierungsprobleme überführt. Somit fokussiert sich die weitere Analyse im Bereich TL auf die semantische Segmentierung. Die nachfolgenden Ergebnisse stellen Erweiterungen der im Rahmen der Arbeit entstandenen Veröffentlichungen in [Mol23a, Mol23b, Rod23] dar.

Es wird das Potenzial von $K = 12$ annotierten Datensätzen (${}^k\mathcal{D}^a$) $_{k=1, \dots, K}$ zur Lösung der nicht annotierten Datensätzen untersucht. Elf Datensätze stammen aus der biomedizinischen Domäne und ein Datensatz bestehend aus Alltagsbildern (${}^8\mathcal{D}^a$) ist als Ausreißer ebenfalls in den Vergleich mit aufgenommen worden. Nähere Informationen inklusive Beispiele finden sich unter Anhang C.4. Als nicht annotierte Datensätze werden, die bereits diskutierten, vier zu lösenden Datensätze (ISIC-2017-Melanoma, DMA-Spheroid-Hellfeld, MoNuSeg, LIVECell) beleuchtet. Zu Beginn wird der Zusammenhang analysiert, wie sich i) die geschätzten Ähnlichkeiten (Γ) zwischen gelösten und unbekanntem Datensätzen zur ii) Genauigkeit im Sinne Q_{DSC} bezüglich des zu lösenden Datensatzes verhalten. Dazu wird ein DNN auf den bekannten Datensätzen trainiert, welches zur Lösung des nicht annotierten Datensatzes verwendet wird.

In Tabelle 5.12 finden sich die Ergebnisse für eine Ähnlichkeitsschätzung für alle genannten Datensatzkombinationen auf Basis des entwickelten Ansatzes, der die *Unsicherheit von DNNs* als Vergleichsmetrik nutzt. Ferner sind in Tabelle 5.13 die Übersichtsresultate der Methoden zu finden. Hier werden sowohl die Korrelationen ξ zwischen Lösungsgüte des unbekanntem Datensatzes bzw. vorhergesagter Ähnlichkeit als auch die Genauigkeiten der korrekten Auswahl des optimalen Datensatzes (Q_{Top-1}, Q_{Top-3}) dargestellt.

Tabelle 5.12: Quantifizierung der Ähnlichkeit von Datensätzen auf Basis der Unsicherheit. Es erfolgt eine vergleichende Darstellung der vorhergesagten Ähnlichkeit Γ , basierend auf prädizierten Unsicherheiten der DNNs, zwischen den zu lösenden Datensätzen und als bekannt angenommen annotierten Datensätzen (${}^k\mathcal{D}^a$) $_{k=1,\dots,K}$. Ferner wird die Segmentierungsgüte anhand von Q_{DSC} analysiert, wenn ein auf dem bekannten Datensatz ${}^k\mathcal{D}^a$ vortrainiertes DNN zur Lösung des Problems genutzt wird. In Fettdruck ist jeweils der als am ähnlichsten vorhergesagte Datensatz (1. Zeile) bzw. Datensatz mit höchster Segmentierungsgüte (2. Zeile) markiert.

	Metrik	${}^1\mathcal{D}^a$	${}^2\mathcal{D}^a$	${}^3\mathcal{D}^a$	${}^4\mathcal{D}^a$	${}^5\mathcal{D}^a$	${}^6\mathcal{D}^a$	${}^7\mathcal{D}^a$	${}^8\mathcal{D}^a$	${}^9\mathcal{D}^a$	${}^{10}\mathcal{D}^a$	${}^{11}\mathcal{D}^a$	${}^{12}\mathcal{D}^a$
ISIC-2017-Melanoma	Γ	0.845	0.860	0.802	0.760	0.823	0.732	0.717	0.822	0.634	0.442	0.815	0.937
	$Q_{DSC} \uparrow$	0.187	0.059	0.223	0.160	0.120	0.176	0.148	0.353	0.139	0.021	0.497	0.791
DMA-Spheroid-Hellfeld	Γ	0.888	0.779	0.790	0.792	0.485	0.809	0.908	0.728	0.453	0.467	0.882	0.791
	$Q_{DSC} \uparrow$	0.168	0.107	0.020	0.016	0.009	0.023	0.182	0.076	0.078	0.028	0.120	0.154
MoNuSeg	Γ	0.926	0.898	0.814	0.729	0.906	0.796	0.951	0.783	0.529	0.650	0.894	0.877
	$Q_{DSC} \uparrow$	0.726	0.558	0.143	0.022	0.013	0.075	0.811	0.301	0.000	0.050	0.558	0.359
LIVECell	Γ	0.881	0.859	0.814	0.721	0.891	0.847	0.943	0.802	0.785	0.678	0.975	0.845
	$Q_{DSC} \uparrow$	0.707	0.296	0.298	0.020	0.329	0.618	0.549	0.766	0.567	0.220	0.863	0.306

Es zeigt sich, dass für alle zu lösenden Datensätze mittels der Ähnlichkeit Γ der optimale Datensatz für TL automatisiert ausgewählt wird ($Q_{Top-1} = 1.00$). Ferner folgt aus der Korrelation $\xi > 0.6$ zwischen der prädizierten Ähnlichkeit Γ und der Segmentierungsgüte Q_{DSC} , dass generell über die vorgeschlagene Methode ein sinnvoller Zusammenhang zur Datensatzauswahl gefunden werden kann. Allerdings muss als Limitierung angemerkt werden, dass vereinzelt ebenfalls starke Ausreißer in den Ergebnissen vorzufinden sind, d. h. es kommt zu hohen vorhergesagten Ähnlichkeiten (vgl. $\Gamma = 0.860$, ISIC-2017-Melanoma zu ${}^2\mathcal{D}^a$), obwohl der Datensatz sich mit $Q_{DSC} = 0.059$ sehr schlecht zur Lösung des jeweiligen Problems eignet. Die Unsicherheits-schätzung hat Probleme, bei zu großen Unterschieden zwischen den Domänen sinnvolle Aussagen zu treffen (z. B. ISIC-2017-Melanoma: Farbaufnahmen aus Kamera, ${}^2\mathcal{D}^a$: Graustufenbilder aus Mikroskop). Ähnliches lässt sich im Falle des Ausreißerdatensatzes ${}^8\mathcal{D}^a$ feststellen. Hier werden hohe Ähnlichkeitswerte zu den Vergleichsdatsätzen ($\Gamma \in [0.728, 0.822]$) vorhergesagt, aber die

Tabelle 5.13: Vergleich der Ansätze. Die drei vorgeschlagenen Methoden werden hinsichtlich der Korrelation ξ von Ähnlichkeit Γ und Segmentierungsgüte Q_{DSC} bzw. Korrektheit der besten (Q_{Top-1})/korrekten Auswahl unter den drei besten Vorschlägen (Q_{Top-3}) verglichen.

Methode	$\xi(Q_{DSC}, \Gamma)$	Q_{Top-1}	Q_{Top-3}
Unsicherheit	0.605	1.00	1.00
Bildmerkmale	0.632	0.75	1.00
Kombination ($\chi = 0.4$)	0.694	0.75	1.00

deutlichen Schwankungen hinsichtlich Q_{DSC} zeigen an, dass dies nicht der Realität entspricht und somit der Ansatz hier fehlschlägt.

Ferner sind in Tabelle 5.14 die Ergebnisse unter Verwendung der konzipierten Methodik dargestellt, die *Bildmerkmale* berücksichtigen.

Tabelle 5.14: Quantifizierung der Ähnlichkeit von Datensätzen auf Basis von Bildmerkmalen. Analog zu Tabelle 5.12 werden die Ergebnisse des Ansatzes basierend auf den Bildmerkmalen gezeigt.

	Metrik	$1\mathcal{D}^a$	$2\mathcal{D}^a$	$3\mathcal{D}^a$	$4\mathcal{D}^a$	$5\mathcal{D}^a$	$6\mathcal{D}^a$	$7\mathcal{D}^a$	$8\mathcal{D}^a$	$9\mathcal{D}^a$	$10\mathcal{D}^a$	$11\mathcal{D}^a$	$12\mathcal{D}^a$
ISIC-2017-Melanoma	Γ	0.545	0.334	0.205	0.190	0.253	0.214	0.674	0.795	0.244	0.413	0.352	0.943
	$Q_{\text{DSC}} \uparrow$	0.187	0.059	0.223	0.160	0.120	0.176	0.148	0.353	0.139	0.021	0.497	0.791
DMA-Spheroid-Hellfeld	Γ	0.649	0.219	0.090	0.085	0.127	0.116	0.842	0.675	0.119	0.243	0.247	0.651
	$Q_{\text{DSC}} \uparrow$	0.168	0.107	0.020	0.016	0.009	0.023	0.182	0.076	0.078	0.028	0.120	0.154
MoNuSeg	Γ	0.927	0.192	0.107	0.101	0.147	0.123	0.882	0.695	0.129	0.217	0.222	0.558
	$Q_{\text{DSC}} \uparrow$	0.726	0.558	0.143	0.022	0.013	0.075	0.811	0.301	0.000	0.050	0.558	0.359
LIVECell	Γ	0.119	0.867	0.112	0.138	0.180	0.095	0.180	0.256	0.305	0.078	0.895	0.233
	$Q_{\text{DSC}} \uparrow$	0.707	0.296	0.298	0.020	0.329	0.618	0.549	0.766	0.567	0.220	0.863	0.306

Es kann mit dieser Methode lediglich in drei von vier Fällen der optimale Datensatz ausgewählt werden ($Q_{\text{Top-1}} = 0.75$). Allerdings unterscheiden sich im Falle von MoNuSeg die Segmentierungsgüten des am ähnlichsten ($1\mathcal{D}^a$: $\Gamma = 0.927$, $Q_{\text{DSC}} = 0.726$) und zweit-ähnlichst vorhergesagten Datensatzes ($7\mathcal{D}^a$: $\Gamma = 0.910$, $Q_{\text{DSC}} = 0.811$) nur minimal, was über $Q_{\text{Top-3}} = 1.00$ belegt wird. Damit befindet sich der optimale Datensatz ($7\mathcal{D}^a$) unter den drei am ähnlichsten vorhergesagten Datensätzen ($1\mathcal{D}^a$, $7\mathcal{D}^a$, $8\mathcal{D}^a$). Ferner lassen sich mit diesem Ansatz auf Basis der Bildinhalte höhere Korrelationen mit $\xi = 0.632$ erreichen, wodurch diese Methode global gesehen der Unsicherheits-Methode mit $\xi = 0.605$ überlegen ist (vgl. Tabelle 5.13).

Im letzten Schritt finden sich in Tabelle 5.15 die Resultate, falls eine *Kombination* beider vorher genannten Ansätze herangezogen wird.

Es zeigt sich bei der Kombination beider Ansätze, dass die höchste Korrelation ($\xi = 0.694$) zwischen der vorhergesagten Ähnlichkeit Γ und jeweiligen Segmentierungsgüte Q_{DSC} erreicht wird. Die Verbesserung bei Nutzung des kombinierten Ansatzes liegt im Vergleich zu den einzelnen Methoden bei knapp 15 % (im Vergleich zur Methode auf Basis der Unsicherheit) bzw. 10 % (bezüglich des Ansatzes über die Bildinhalte), was Tabelle 5.13 zu entnehmen ist.

Tabelle 5.15: Kombinierte Quantifizierung der Ähnlichkeit von Datensätzen. Entsprechend zu Tabelle 5.12 und Tabelle 5.14 sind hier die Resultate der kombinierten Methode dargestellt. Der gewählte Überlagerungsparameter lautet $\chi = 0.4$.

Metrik	$1\mathcal{D}^a$	$2\mathcal{D}^a$	$3\mathcal{D}^a$	$4\mathcal{D}^a$	$5\mathcal{D}^a$	$6\mathcal{D}^a$	$7\mathcal{D}^a$	$8\mathcal{D}^a$	$9\mathcal{D}^a$	$10\mathcal{D}^a$	$11\mathcal{D}^a$	$12\mathcal{D}^a$	
ISIC-2017-Melanoma	Γ	0.665	0.544	0.444	0.418	0.481	0.421	0.691	0.806	0.400	0.425	0.537	0.941
	$Q_{DSC} \uparrow$	0.187	0.059	0.223	0.160	0.120	0.176	0.148	0.353	0.139	0.021	0.497	0.791
DMA-Spheroid-Hellfeld	Γ	0.744	0.443	0.370	0.368	0.270	0.393	0.868	0.696	0.252	0.333	0.501	0.707
	$Q_{DSC} \uparrow$	0.168	0.107	0.020	0.016	0.009	0.023	0.182	0.076	0.078	0.028	0.120	0.154
MoNuSeg	Γ	0.927	0.474	0.390	0.352	0.451	0.392	0.910	0.730	0.289	0.390	0.490	0.686
	$Q_{DSC} \uparrow$	0.726	0.558	0.143	0.022	0.013	0.075	0.811	0.301	0.000	0.050	0.558	0.359
LIVECell	Γ	0.424	0.864	0.392	0.371	0.464	0.396	0.485	0.474	0.497	0.318	0.927	0.478
	$Q_{DSC} \uparrow$	0.707	0.296	0.298	0.020	0.329	0.618	0.549	0.766	0.567	0.220	0.863	0.306

Dennoch bleibt das beschriebene Problem, dass im Falle von MoNuSeg immer noch ein minimal schlechterer Datensatz ausgewählt wird. Nichtsdestoweniger ist der kombinierte Ansatz zu präferieren, da dieser die global besten Zusammenhänge (vgl. Korrelationen ξ) liefert.

Zudem zeigt Tabelle 5.16 die Ergebnisse für die verschiedenen in Abschnitt 3.6 eingeführten Fine-Tuning-Varianten (ALR, GVLR, LVLR, ALR+Mix, GVLR+Mix, LVLR+Mix), wenn der über die kombinierte Methodik bestimmte ähnlichste bekannte Datensatz mit dem teilweise annotierten unbekanntem Datensatz verknüpft wird. Um verschiedene Annotationsraten ρ zu beleuchten, wird das integrale Gütekriterium $Q_{int, DSC}$ (vgl. Gleichung (3.6)) genutzt. Ferner sind die p -Werte eines einseitigen Zweistichproben- t -Tests (unter der Annahme gleicher Varianzen) gezeigt, um zu prüfen, ob signifikante Verbesserungen ($p < 0.05$) im Vergleich zur Referenz vorliegen. Zudem wird die normierte Rechenzeit $Q_{t, norm}$ (vgl. Gleichung (3.7)) gezeigt. Benchmark bildet dabei das Training ohne Berücksichtigung eines bekannten Datensatzes (Ohne), also keine Verwendung von TL.

Es zeigt sich, dass die TL-Methoden im Allgemeinen signifikante Verbesserung der DNN-Güte ($Q_{int, DSC}$) im Vergleich zum Training ohne Wissen aus einer anderen Domäne liefern. Allerdings kann bei einer Gegenüberstellung der Fine-Tuning-Methoden keine einzelne global überlegene Methode gefunden werden. Sowohl ALR als auch GVLR liefern zusammenfassend sinnvolle Ergebnisse und sind bei den Untersuchungen der LVLR-Methode überlegen ($p < 0.05$ für drei von vier Datensätzen). Die Verknüpfung von alten und neuen Trainingsdaten (Mix) führt nur bezüglich des ISIC-2017-Melanoma-Datensatzes zu deutlichen Verbesserungen ($p < 0.001$).

Tabelle 5.16: Vergleich der datensatzspezifischen Metriken für TL. Basierend auf den Metriken $Q_{\text{int, DSC}}$ (inklusive p -Werte eines einseitigen Zweistichproben- t -Tests, unter der Annahme gleicher Varianzen, zur Überprüfung einer signifikanten Verbesserung im Vergleich zur Referenz) bzw. $Q_{t, \text{norm}}$ werden die Fine-Tuning-Ansätze ALR, GVLR, LVLR, GVLR+Mix sowie LVLR+Mix zum Ansatz ohne TL verglichen. Dabei erfolgt die Gegenüberstellung für vier verschiedene Datensätze (ISIC-2017-Melanoma, DMA-Spheroid-Hellfeld, MoNuSeg, LIVECell). Die jeweils überlegene Methode ist mittels Fettdruck markiert.

	Metrik	Ohne	ALR	GVLR	LVLR	ALR+Mix	GVLR+Mix	LVLR+Mix
ISIC-2017-Melanoma	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	0.961 (-)	0.989 (***)	0.981 (***)	0.975 (**)	1.003 (***)	1.005 (***)	0.999 (***)
	$Q_{t, \text{norm}} \downarrow$	1.000	3.463	3.460	3.464	3.818	3.789	3.725
DMA-Spheroid-Hellfeld	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	0.932 (-)	0.943 (o)	0.954 (*)	0.938 (o)	0.908 (o)	0.930 (o)	0.948 (*)
	$Q_{t, \text{norm}} \downarrow$	1.000	20.223	20.203	20.314	20.657	20.494	20.800
MoNuSeg	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	0.984 (-)	0.991 (*)	0.995 (***)	0.989 (*)	0.993 (***)	0.994 (***)	0.992 (***)
	$Q_{t, \text{norm}} \downarrow$	1.000	10.647	10.640	10.661	11.239	11.189	11.088
LIVECell	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	0.998 (-)	0.999 (**)	0.998 (o)	0.998 (o)	0.998 (o)	0.998 (o)	0.997 (o)
	$Q_{t, \text{norm}} \downarrow$	1.000	4.165	4.156	4.150	4.807	4.812	4.748

Die Rechenzeiten $Q_{t, \text{norm}}$ sind in einem Bereich zwischen Faktor drei und 21 länger, da aufwändige Berechnungen im Vorfeld zur Bestimmung der Ähnlichkeiten notwendig sind. Relativ machen sich diese primär fixen Rechenkosten am stärksten beim DMA-Spheroid-Hellfeld-Datensatz bemerkbar, da dieser von geringerem Umfang ist und dementsprechend zügig trainiert ($Q_{t, \text{norm}} > 20$).

Tabelle 5.17 vergleicht die verschiedenen Methoden anhand der eingeführten datensatzübergreifenden Metriken.

Tabelle 5.17: Vergleich der datensatzübergreifenden Metriken für TL. Es werden die verschiedenen Methoden zum Fine-Tuning anhand der Kriterien „zusätzliche Hyperparameter“, „Implementierungsaufwand“ und „Hardwareanforderungen“ verglichen. Für qualitative Kennzahlen sind Vorteile mit +/++ und Nachteile mittels -/- dargestellt bzw. neutrale Eigenschaften über o.

Kriterium	Methoden					
	Ohne	ALR	GVLR	LVLR	GVLR+Mix	LVLR+Mix
Zusätzliche Hyperparameter \downarrow	0	3	4	4	5	5
Implementierungsaufwand	++	-	-	--	--	--
Hardwareanforderung	++	++	++	++	++	++

Jede der auf TL-basierten Methoden beinhaltet mindestens drei Hyperparameter für die automatisierte Auswahl des Ursprungsdatensatzes (Unsicherheitsmethode, Methode auf Basis der Bildinhalte und deren Gewichtung bei der Kombination). Im Falle von GVLR oder LVLR muss zudem festgelegt werden, wie stark die Lernrate reduziert werden soll. Ferner resultiert im Falle der Ansätze, die Mix beinhalten, ein weiterer Parameter zum Einstellen des

Umfangs der Trainingsdaten aus dem ursprünglichen Datensatz. Der Implementierungsaufwand ist für alle Methoden hoch, da die automatisierte Methode zur Auswahl des ähnlichsten Datensatzes programmiert werden muss. Die Komplexität erhöht sich zudem im Falle des kombinierten Trainings beim Mix-Ansatz. Allerdings gibt es keine speziellen Hardwareanforderungen für die verglichenen Methoden.

Damit sind generell bei beschränkten Ressourcen im Rahmen der Implementierung bzw. Parametrierung der Methoden die weniger komplexen Ansätze (ALR, GVLR) zu präferieren.

5.4.2 Deep Active Learning

Nachfolgend sollen die Resultate der Methoden im Bereich DAL (Uncertainty, Core Set - ResNet/U-Net/Autoencoder (AE)) beschrieben und diskutiert werden. In Tabelle 5.18 finden sich die Ergebnisse bezüglich der datensatzspezifischen Metriken. Die Referenz bildet dabei ein zufälliges Ziehen der zu annotierenden Bilder (Random). Es wird hier zwischen einem fortgesetzten Training (F) und Trainingsneubeginn (N) differenziert.

Tabelle 5.18: Vergleich der datensatzspezifischen Metriken für DAL. Analog zu Tabelle 5.16 erfolgt eine vergleichende Darstellung unterschiedlicher Methoden im Kontext DAL (Uncertainty, Core Set - ResNet/U-Net/AE) zum zufälligen Ziehen der Bilder (Random) für die semantische Segmentierung und die Instanzsegmentierung. Hierbei wird zwischen den Konfigurationen Trainingsfortsetzung (F) und Trainingsneubeginn (N) unterschieden (vgl. Typ).

Datensatz	Metrik	Typ	Methoden				
			Random	Uncertainty	Core Set - ResNet	Core Set - U-Net	Core Set - AE
ISIC-2017-Melanoma	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	F	0.926 (-)	0.877 (o)	0.917 (o)	0.932 (o)	0.912 (o)
		N	0.952 (-)	0.914 (o)	0.955 (o)	0.967 (*)	0.960 (o)
	$Q_{t, \text{norm}} \downarrow$	F	1.000	1.664	0.666	1.017	0.680
		N	1.733	2.073	1.118	1.347	1.181
DMA-Spheroid-Hellfeld	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	F	0.928 (-)	0.915 (o)	0.880 (o)	0.931 (o)	0.913 (o)
		N	0.946 (-)	0.929 (o)	0.942 (o)	0.957 (o)	0.953 (o)
	$Q_{t, \text{norm}} \downarrow$	F	1.000	0.549	0.528	0.461	1.001
		N	1.185	0.631	0.537	0.561	1.107
MoNuSeg	$Q_{\text{int, AJI+}} \uparrow (p \downarrow)$	F	0.936 (-)	0.949 (*)	0.936 (o)	0.934 (o)	0.926 (o)
		N	0.963 (-)	0.971 (o)	0.962 (o)	0.959 (o)	0.961 (o)
	$Q_{t, \text{norm}} \downarrow$	F	1.000	3.720	0.827	0.756	0.816
		N	1.336	4.058	1.020	1.063	1.059
LIVECell	$Q_{\text{int, AJI}} (p \downarrow)$	F	1.000 (-)	0.996 (o)	1.000 (o)	0.995 (o)	1.001 (o)
		N	1.009 (-)	1.007 (o)	1.011 (o)	1.000 (o)	1.006 (o)
	$Q_{t, \text{norm}} \downarrow$	F	1.000	7.281	0.883	0.945	0.833
		N	1.164	7.419	1.018	1.125	1.157

Zunächst lässt sich festhalten, dass im Sinne der Güte des DNN (vgl. $Q_{\text{int, DSC/AJI+}}$) ein Trainingsneubeginn (N) deutlich überlegen zu einer Fortsetzung des Trainings (F) ist. Allerdings folgen längere Rechenzeiten $Q_{t, \text{norm}}$ als Nachteil im Falle des Trainingsneubeginns. Im direkten Vergleich der Methoden zum zufälligen Ermitteln der Reihenfolge von annotierten Bildern sind lediglich in zwei Fällen (ISIC-2017-Melanoma mit Uncertainty/N, MoNuSeg mit Core Set - U-Net/F) schwach signifikante Verbesserungen ($p < 0.05$) festzuhalten, die jedoch zwischen den Datensätzen variieren. Keine der DAL-Methoden zeigt eine globale Überlegenheit im direkten Vergleich zur Referenz/zu anderen Konkurrenzmethoden. Zudem folgt im Falle des unsicherheitsbasierten Ansatzes ein deutlicher Nachteil in puncto Rechenzeit $Q_{t, \text{norm}}$, insbesondere im Rahmen der Instanzsegmentierung. Dies lässt sich durch die erforderlichen mehreren Inferenzen zur Erzeugung einer Stichprobe erklären. Im Speziellen ist die Berechnung bei vielen Instanzen pro Bild sehr aufwändig. Z. B. steigt der Rechenaufwand im Falle des LIVECell-Datensatzes mit $Q_{t, \text{norm}} > 7$ oder bei MoNuSeg mit $Q_{t, \text{norm}} > 3.7$ stark an.

In Tabelle 5.19 erfolgt eine zusammenfassende Bewertung der eingeführten datensatzübergreifenden Metriken für die genannten Verfahren.

Tabelle 5.19: Vergleich der datensatzübergreifenden Metriken für DAL. Ähnlich zu Tabelle 5.17 werden die verschiedenen DAL-Methoden anhand der datensatzübergreifenden Bewertungsmetriken gegenübergestellt.

Kriterium	Methoden				
	Random	Uncertainty	Core Set - ResNet	Core Set - U-Net	Core Set - AE
Zusätzliche Hyperparameter ↓	0	2	2	2	2
Implementierungsaufwand	++	--	0	0	-
Hardwareanforderung	++	++	++	++	++

Zunächst lässt sich festhalten, dass keine der Methoden besondere Hardwareanforderungen voraussetzt. Allerdings ist ein zusätzlicher Implementierungsaufwand für alle DAL-Methoden zu verzeichnen. Beispielsweise sind unsicherheitsbewusste DNNs inklusive Metriken (vgl. Uncertainty), Netze zur Merkmalsextraktion (Core Set - AE) oder Optimierungsalgorithmen für das Bestimmen heterogener Beispiele auf Basis extrahierter Bildmerkmale (Core Set) zusätzlich zu programmieren. Es gibt ferner jeweils zwei zusätzliche Hyperparameter, die es einzustellen gilt. Im Falle der Core-Set-Methoden

müssen die latente Dimension und Vergleichsmetrik definiert werden, wohingegen im Falle von Uncertainty das Verfahren zur Schätzung der Unsicherheit sowie die zu nutzende Berechnungsmetrik auszuwählen ist.

5.4.3 Semi-Supervised Learning

Zur Analyse der Segmentierungsqualität (in Abhängigkeit verschiedener Annotationsraten ρ) $Q_{\text{int, DSC/AJL+}}$ bzw. der normierten Rechenzeit $Q_{t, \text{norm}}$ zeigt Tabelle 5.20 die Ergebnisse der Semi-Supervised Learning (SemSL)-Methoden (FixMatch, MeanTeacher, ST, ST++, CCT) für die vier eingeführten Datensätze. Das klassische Supervised Learning (SL) (Supervised) stellt hierbei die Referenz im Vergleich dar. Die Resultate bauen auf der Veröffentlichung in [Kep22] auf, die im Rahmen dieser Arbeit entstanden ist.

Tabelle 5.20: Vergleich der datensatzspezifischen Metriken für SemSL. Die verschiedenen SemSL-Methoden (FixMatch, MeanTeacher, ST, ST++, CCT) werden analog zu Tabelle 5.16 der Referenz-Methode (Supervised) gegenübergestellt.

Datensatz	Metrik	Methoden					
		Supervised	FixMatch	MeanTeacher	ST	ST++	CCT
ISIC-2017-Melanoma	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	0.943 (-)	0.929 (o)	0.957 (*)	0.965 (***)	0.946 (o)	0.954 (*)
	$Q_{t, \text{norm}} \downarrow$	1.000	4.147	5.734	3.766	7.231	7.945
DMA-Spheroid-Hellfeld	$Q_{\text{int, DSC}} \uparrow (p \downarrow)$	0.926 (-)	0.944 (o)	0.956 (o)	0.879 (o)	0.950 (o)	0.966 (*)
	$Q_{t, \text{norm}} \downarrow$	1.000	3.706	2.678	3.407	4.996	3.430
MoNuSeg	$Q_{\text{int, AJL+}} \uparrow (p \downarrow)$	0.910 (-)	0.913 (o)	0.932 (*)	0.531 (o)	0.886 (o)	0.926 (*)
	$Q_{t, \text{norm}} \downarrow$	1.000	2.241	3.927	3.578	3.937	5.085
LIVECell	$Q_{\text{int, AJL+}} \uparrow (p \downarrow)$	0.987 (-)	0.994 (o)	1.002 (*)	0.637 (o)	0.936 (o)	1.001 (*)
	$Q_{t, \text{norm}} \downarrow$	1.000	2.647	2.485	2.163	3.518	6.786

Es kann festgehalten werden, dass CCT und MeanTeacher immer die Referenz des reinen überwachten Lernens (Supervised) übertreffen. Bis auf eine Ausnahme (DMA-Spheroid-Datensatz, $p = 0.097$) im Falle von MeanTeacher sind die Verbesserung alle signifikant ($p < 0.05$). Bei ST/ST++ bzw. FixMatch sind, abhängig vom Datensatz, Verbesserungen im Vergleich zur Referenz festzustellen. Aber es zeigen sich teilweise auch negative Effekte, wodurch unter Verwendung der SemSL-Methoden auch ein schlechteres Optimum nach Ende des Trainings erreicht werden kann. ST kann lediglich im Falle des ISIC-2017-Melanoma-Datensatzes zu signifikanten Verbesserungen

führen, bei allen anderen Datensätzen sind Verschlechterungen zu verzeichnen ($p > 0.95$). Beispielsweise kann dies der Fall sein, wenn die Vorhersagen im Rahmen des Selbsttrainings (ST/ST++) zu schlecht sind. Dementsprechend wird der Trainingsprozess analog zu verrauschten Annotationen (vgl. Abschnitt 5.2.7) negativ beeinflusst. Hinsichtlich der Rechenzeiten $Q_{t,\text{norm}}$ sind deutliche Nachteile im Vergleich zur Referenz erkennbar. Das Training im Falle von SemSL benötigt mindestens doppelt so lange wie die Referenz des überwachten Lernens, kann sich jedoch bis zu zirka Faktor acht verlängern (ISIC-2017-Melanoma-Datensatz mit ST++ oder CCT).

In Tabelle 5.21 werden die Ergebnisse im Hinblick auf die datensatzübergreifenden Metriken präsentiert.

Tabelle 5.21: Vergleich der datensatzübergreifenden Metriken für SemSL. Analog zu Tabelle 5.17 erfolgt der datensatzübergreifende Vergleich der SemSL-Methoden.

Kriterium	Methoden					
	Supervised	FixMatch	MeanTeacher	ST	ST++	CCT
Zusätzliche Hyperparameter ↓	0	4	4	2	3	4
Implementierungsaufwand	++	-	o	o	-	--
Hardwareanforderung	++	++	--	++	--	--

Für die vorgestellten Methoden resultieren zusätzliche Hyperparameter zur Gewichtung des unüberwachten Teils der Fehlerfunktion inklusive der Auswahl einer Rampenfunktion zu Beginn des Trainings (FixMatch, MeanTeacher, CCT). Ferner müssen die Größenverteilungen von annotierten/nicht annotierten Datensatzelementen (alle Methoden) eingestellt werden bzw. gilt es die zusätzlichen starken Augmentierungen (FixMatch, ST, ST++) zu parametrieren. Ferner muss bei FixMatch der Schwellwert im Rahmen des Curriculum Learnings, bei ST++ die Anzahl der Wiederholungs-Zyklen nach Pseudo-Annotation, bei MeanTeacher ein Parameter zur Gewichtung von neuen und vorherigen DNN-Parametern und bei CCT die Anzahl der Decoder gewählt werden. Der Zusatzimplementierungsaufwand hält sich bei MeanTeacher (Mittelung der neuen und vorherigen Parameter) und ST (Abspeicherung der Pseudo-Annotationen) in Grenzen. Bei ST++ und FixMatch müssen zusätzlich Elemente des Curriculum Learnings integriert bzw. zwischen starken/schwachen Augmentierungen differenziert werden.

Im Falle CCT ist eine Veränderung des DNN notwendig, um Veränderungen der latenten Bildmerkmale zu erzeugen, was die Komplexität der Implementierung steigert. Die Hardwareanforderungen hinsichtlich GPU sind im Falle von MeanTeacher (zwei Modelle), ST++ (mehrere Modellzustände zu verschiedenen Trainingsstadien für Curriculum Learning) und CCT (mehrere Decoder) erhöht.

Es lässt sich als neue Erkenntnis festhalten, dass im Falle fehlender Einschränkungen hinsichtlich Rechenzeit oder Hardware die Methoden CCT oder MeanTeacher für Anwender empfohlen werden können. Wenn eine einfache Implementierung angestrebt wird, ist es ratsam, MeanTeacher zu verwenden. Allerdings kann bei limitierten Rechenzeiten bzw. Hardwareeinschränkungen der Falle sein, dass die Methoden nicht anwendbar sind.

5.4.4 Self-Supervised Learning

Im Weiteren werden die Ergebnisse der Self-Supervised Learning (SSL)-Methoden (AE, Simple Framework for Contrastive Learning of Visual Representations (SimCLR), Barlow Twins, Bootstrap Your Own Latent (BYOL), MoCo, DenseCL und ImageNet) gezeigt und diskutiert. Die zufällige Initialisierung (Random) des Encoders kann hier als Referenz genannt werden. Die gezeigten Ergebnisse stellen Erweiterungen der Publikationen in [Ret21, Ret23] dar, welche im Rahmen der vorliegenden Arbeit entwickelt worden sind. Eine Vorstellung der datensatzspezifischen Resultate findet sich in Tabelle 5.22. Dabei wird zudem ein Vergleich zwischen der Fixierung (F) bzw. einem Training (N) der Encoder-Parameter präsentiert.

Es kann festgestellt werden, dass die Fixierung des Encoders (F) im Vergleich zum weiteren Training (N) in allen Experimenten zu einer Verschlechterung der DNNs führt. Zum anderen kann keine SSL-Methode gefunden werden, die für alle Datensätze global zu signifikant besseren Ergebnissen bezüglich $Q_{\text{int, DSC/AJL+}}$ führt. SimCLR löst zwei der acht Fälle am besten. Hingegen zeigt sich, dass ein auf ImageNet vortrainierter Encoder in vier von acht Fällen die beste Methode darstellt und in diesem Fall die anderen SSL-Methoden schlägt. Folglich stellt die Verwendung eines auf ImageNet trainierten Encoders eine

Tabelle 5.22: Vergleich der datensatzspezifischen Metriken für SSL. Die verschiedenen SSL-Methoden (AE, SimCLR, Barlow Twins, BYOL, MoCo, DenseCL und ImageNet) werden wie in Tabelle 5.16 pro Datensatz gegenübergestellt. Als Referenz kann die zufällige Initialisierung (Random) des Encoders aufgeführt werden. Es wird zwischen der Fixierung (F) bzw. Training (N) von Encoder-Parametern unterschieden.

Datensatz	Metrik	Typ	Methoden							
			Random	AE	SimCLR	Barlow Twins	BYOL	MoCo	DenseCL	ImageNet
ISIC-2017-	$Q_{int, DSC} \uparrow (P \downarrow)$	F	0.946 (-)	0.949 (o)	0.951 (o)	0.970 (**)	0.964 (**)	0.972 (***)	0.974 (***)	0.954 (o)
		N	0.968 (-)	0.980 (**)	0.972 (o)	0.977 (o)	0.968 (o)	0.985 (**)	0.982 (**)	0.974 (o)
		F	1.000	1.542	6.607	6.715	7.116	6.626	6.359	1.226
Melanoma	$Q_{L, norm} \downarrow$	F	1.000	1.573	7.183	7.081	7.611	7.100	6.924	1.172
		N	0.931 (+)	0.928 (o)	0.930 (o)	0.912 (o)	0.910 (o)	0.900 (o)	0.917 (o)	0.933 (+)
		N	0.958 (-)	0.952 (o)	0.963 (o)	0.922 (o)	0.952 (o)	0.943 (o)	0.948 (o)	0.961 (o)
DMA-Spheroid-Hellfeld	$Q_{L, norm} \downarrow$	F	1.000	1.086	2.605	2.654	2.667	2.572	2.479	0.624
		N	1.000	1.040	2.606	2.608	2.612	2.499	2.442	0.622
		F	0.888 (-)	0.914 (***)	0.887 (o)	0.865 (o)	0.883 (o)	0.900 (*)	0.910 (***)	0.936 (***)
MoNuSeg	$Q_{int, AJI} \uparrow (P \downarrow)$	N	0.960 (-)	0.954 (o)	0.961 (o)	0.922 (o)	0.959 (o)	0.959 (o)	0.958 (o)	0.967 (*)
		F	1.000	2.025	4.516	4.751	5.283	5.851	4.149	0.906
		N	1.000	1.879	4.948	4.328	4.948	5.469	3.897	0.995
LIVECell	$Q_{int, AJI} \uparrow (P \downarrow)$	F	0.966 (-)	0.989 (o)	0.987 (***)	0.985 (***)	0.982 (***)	0.992 (***)	0.991 (***)	0.994 (***)
		N	1.005 (-)	0.998 (o)	1.006 (o)	0.996 (o)	1.005 (o)	1.005 (o)	1.005 (o)	1.004 (o)
		F	1.000	2.017	3.767	3.755	4.188	3.265	2.228	0.882
	$Q_{L, norm} \downarrow$	N	1.000	1.675	3.099	2.950	3.420	2.791	1.961	0.778

universelle Methode dar, die in vielen Fällen erfolgreich und zudem rechen- effizient ist.

Darüber hinaus kann festgehalten werden, dass sich die Rechenzeit von Random, AE und ImageNet in einer ähnlichen Bandbreite bewegen, wohingegen die anderen SSL-Verfahren durch den aufwändigeren Pre-Text-Task deutlich höhere Rechenzeiten verursachen.

Bei einer Betrachtung der datensatzübergreifenden Metriken (vgl. Tabelle 5.23) sind die Zufallsinitialisierung und das Verwenden eines auf ImageNet trainierten Encoders gegenüber den anderen SSL-Methoden überlegen. Das Einbinden eines AE zudem ist schnell ohne große Hardwareanforderungen möglich. Insbesondere die Implementierungen von MoCo und DenseCL sind deutlich komplexer bzw. durch größere Modelle steigen die GPU-Hardwareanforderungen. Die größten Hardwareanforderungen bestehen bei BYOL, da sowohl eine erhöhte Modellgröße vorliegt und große Batchgrößen notwendig sind.

Tabelle 5.23: Vergleich der datensatzübergreifenden Metriken für SSL. Es werden analog zu Tabelle 5.17 die unterschiedlichen SSL-Methoden gegenübergestellt.

Kriterium	Methoden							
	Random	AE	SimCLR	Barlow Twins	BYOL	MoCo	DenseCL	ImageNet
Zusätzliche Hyperparameter \downarrow	0	0	1	1	1	2	3	0
Implementierungsaufwand	++	+	o	o	o	--	--	++
Hardwareanforderung	++	++	o	-	--	-	-	++

Es folgt damit als neue Erkenntnis für Anwender, dass in vielen Situationen ein Encoder, der zuvor auf ImageNet trainiert wurde, als die beste Wahl im Bereich der SSL-Methoden betrachtet werden kann. Dies liegt hauptsächlich daran, dass ein auf ImageNet vortrainierter Encoder oft Vorteile in Bezug auf die Qualität der Segmentierung bietet und gleichzeitig wenige Anpassungen, zusätzliche Rechenzeit oder spezielle Hardwareanforderungen erfordert.

5.4.5 Resümee

Unter der Verwendung von TL oder SemSL sind mehrheitlich beachtliche Verbesserungen der Güte von DNNs möglich, wenn nicht annotierte Daten zur Verfügung stehen.

Die neu entwickelte Methode zur Bewertung von Ähnlichkeiten zwischen Datensätzen hilft Nutzern bei einer vereinfachten Anwendung von TL. Die klassischen Fine-Tuning-Verfahren der global reduzierten/ursprünglichen Lernrate (vgl. ALR/GVLR) im Kontext TL sind oftmals elaborierten Ansätzen (LVLR bzw. Mix) in den gezeigten Untersuchungen überlegen. Die Methoden CCT oder MeanTeacher sind im Rahmen von SemSL universelle Ansätze, die in den meisten Fällen zum Erfolg führen. Allerdings steigt bei allen Methoden im Bereich TL und SemSL der Implementierungs-, Rechen-, Parametrierungs- und Hardwareaufwand.

Im Bereich von DAL konnte kein Verfahren identifiziert werden, das eindeutig und global als überlegen gegenüber der Referenzmethode des überwachten Lernens eingestuft werden kann. Ferner zeigt sich, dass elaborierte SSL-Methoden oftmals einem auf ImageNet trainierten Encoder nicht überlegen sind. Ähnlich zu dem Ergebnis im Falle von DAL kann auch keine dominierende SSL-Methode gefunden werden. Ein auf ImageNet trainierter Encoder lässt sich ohne Aufwand durch Setzen eines Parameters in gängigen DL-Programmpaketen¹ direkt verwenden.

¹ <https://pytorch.org/vision/stable/models.html>

Anwendern wird durch den umfangreichen Vergleich des Stands der Technik zu „annotations-effizientem Lernen“ die Methodenauswahl im Falle von Segmentierungsproblemen deutlich erleichtert. Mithilfe der umfassenden Metriken zusätzlich zur Betrachtung der Güte können Anwender unter den jeweiligen Rahmenbedingungen die sinnvollste Methode auswählen. Ferner wurde über das eingeführte integrale Gütekriterium $Q_{\text{int},(\cdot)}$ eine Metrik gefunden, die den gesamten Verlauf der Lernkurve, also verschiedene Größen von partiell annotierten Datensätzen berücksichtigt. Generell entstehen bei Verwendung der untersuchten Methoden höhere Rechenkosten, jedoch bringen diese meist (mit wenigen Ausnahmen) keine zusätzlichen Anforderungen hinsichtlich Hardware mit sich. Zudem resultiert durch die Implementierung und Einstellung der Methode in der Regel ein deutlicher Zusatzaufwand für Nutzer.

5.5 Anwendung des entwickelten Konzepts

Abschließend werden in Tabelle 5.24 die verschiedenen Kooperationsprojekte beschrieben, die mit den entwickelten Methoden unterstützt werden konnten. Außerdem sind in Abbildung 5.20 Beispielbilder dargestellt, um die Projektvielfalt bildlich zu zeigen. Für eine detaillierte Beschreibung der Projekte soll auf die entsprechenden Veröffentlichungen [Böh23, Cha22, Cui23, Krü22, Sch23d, Str21a, Wie22] verwiesen und an dieser Stelle verzichtet werden.

Zusammenfassend kann festgehalten werden, dass die entwickelten Werkzeuge (KaIDA, AI²Seg und Grid Screener) in der Praxis Anwendung finden. Die Einsatzgebiete sind dabei sehr vielseitig, was sich auf die Erweiterbarkeit und Modularität der neu entwickelten Werkzeuge zurückführen lässt (vgl. Abschnitt 4.2). Die Werkzeuge sind größtenteils über die beschriebene Infrastruktur (vgl. Abschnitt 4.2) den Projektpartnern zur Verfügung gestellt worden, um den Partnern einen eigenständigen Einsatz der Softwarepakete zu ermöglichen.

Tabelle 5.24: Überblick der Kooperationen. Es sind verschiedene Projekte mit den jeweiligen Kooperationspartnern inklusive der Projektbeschreibung bzw. verwendeten Werkzeugen dargestellt. Zugehörigkeiten: ¹BioQuant, Universität Heidelberg; ²Institut für Automation und angewandte Informatik, Karlsruher Institut für Technologie; ³Institut für Biologische und Chemische Systeme - Funktionelle molekulare Systeme, Karlsruher Institut für Technologie; ⁴Pharmakologisches Institut, Universität Heidelberg; ⁵Institut für Biologische und Chemische Systeme - Biologische Informationsprozessierung, Karlsruher Institut für Technologie.

ID	Projektname	Projektpartner	Projektbeschreibung	Werkzeug
a	MiCATS [Sch23d]	Svenja Schmelzer ¹ , Ilka Bischofs ¹	Mikroskopie-gekoppeltes Bildanalyse-Tool für Sporen; Annotation und Inspektion von Bildern mit bakteriellen Sporen; Training eines DNN zur Segmentierung von Instanzen	KaIDA, AI ² Seg
b	Chamber cropper	Svenja Schmelzer ¹ , Ilka Bischofs ¹	Annotation und Training eines U-Nets zum Beschneiden der Kultivierungskammer von Mikrofluidiksystemen	KaIDA
c	GAN Selection [Böh23]	Moritz Böhland ²	Training Instanzsegmentierung; Analyse von Abbruchkriterien für GANs	KaIDA
d	Phoridae	Christian Pylatiuk ² , Luca Rettenberger ² , Lorenz Wühl ²	Segmentierung der Körperteile von Phoroiden (Insekten)	KaIDA
e	Fluoreszenzintensität [Wie22]	Janne Wiedemann ³ , Pavel Levkin ³	Bestimmung der Fluoreszenzintensität; Enzymatischer Inhibitions- assay mit blauer Fluoreszenzanzeige zur Charakterisierung einer auf dem Chip synthetisierten Inhibitorbibliothek	Grid Screener
f	Microglia	Theresa Roth ⁴ , Manuela Simonetti ⁴ , Svenja Schmelzer ¹ , Rohini Kuner ⁴	Detektion von Immuneffektorzellen Mikroglia, Flächensegmentierung und Berechnung des Dorsalhorns des Rückenmarks, Intensitätsmessung der Färbung	KaIDA
g	Farbanalyse	Anna Popova ³ , Pavel Levkin ³	Bestimmung des pH-Werts auf Basis von Farbinformation	Grid Screener
h	Zählung von Zellen	Anna Popova ³ , Pavel Levkin ³	Detektion der Spots innerhalb eines Scans und Bestimmung der Zellviabilität pro Element über die Detektion von Zellkernen in Fluoreszenzeinfärbungen	KaIDA, Grid Screener
i	Organoide	Steffen Sonntag ³ , Véronique Orian-Rousseau ³	Segmentierung von Organoiden und Analyse der korrespondierenden Fluoreszenzintensität des Segments	KaIDA
j	Zellteilung	Christina Cramer von Clausbruch ⁵ , Carsten Weiss ⁵	Detektion von Zellen und Klassifikation des Zellteilungsstadiums	KaIDA

Weiter auf der nächsten Seite

Tabelle 5.24: Überblick der Kooperationen (Fortsetzung).

ID	Projektname	Projektpartner	Projektbeschreibung	Werkzeug
k	Spheroïd I [Cui23]	Haijun Cui ³ , Anna Popova ³ , Pavel Levkin ³	Segmentierung und Charakterisierung von Sphäroiden; Umwidmung von FDA-zugelassenen Medikamenten für Temozolomid-resistente IDH1-mutierte Gliome	KaIDA
l	cDNA [Cha22]	Shraddha Chakraborty ³ , Anna Popova ³ , Pavel Levkin ³	Zellkerndetektion; Zellen zu cDNA: Phänotypische Bewertung und Genexpressionsanalyse von lebenden Zellen	KaIDA
m	Rhodamine	Janne Wiedemann ³ Michelle Iwohn ³ , Pavel Levkin ³	Gittererkennung und Farbauswertung zur Bestimmung der Rhodaminextraktionsrate	Grid Screener
n	CLL [Str21a]	Razan El Khaled El Faraj ³ , Annika Strauß ³ , Anna Popova ³ , Pavel Levkin ³	Validierung der DMA-Plattform für DSRT-Test bei Blutkrebs; Berechnung der Zellviabilität von CLL-Zellen; Annotation und Analyse von mehrdimensionalen Bildern	KaIDA
o	DSRT I	Maryam Salarian ³ , Anna Popova ³ , Pavel Levkin ³	Medikamentenscreening an SCC-25- und HaCaT-Zelllinien in 3D-Zell-Sphäroiden; Annotation und Analyse über Instanzsegmentierung	KaIDA
p	DSRT II	Maryam Salarian ³ , Anna Popova ³ , Pavel Levkin ³	Medikamentenscreening an SCC-25- und HaCaT-Zelllinien in Monolayer; Annotation und Analyse über Segmentierung	KaIDA
q	Spheroïd BF	Charlotte Luchena ³ Anna Popova ³ , Pavel Levkin ³	Segmentierung von Sphäroiden in Hellfeldaufnahmen	KaIDA, Grid Screener
r	Grid Crop, [Krü22]	Martin Krüger ³ , Joaquin Urrutia Gómez ³ , Anna Popova ³ , Pavel Levkin ³	Mechanismus zur kontaktlosen Übertragung von Tropfen; Ausschneidefunktion einzelner Gitterelemente in verschiedenen Bildkanälen	Grid Screener
s	Spots	Joaquin Urrutia Gómez ³ , Anna Popova ³ , Pavel Levkin ³	Detektion von Zellen in Hellfeldaufnahmen	Grid Screener
t	Insects [Wüh22]	Lorenz Wührl ² , Christian Pylatiuk ²	Instanzsegmentierung von Insekten; Aufnahme aus DiversityScanner	KaIDA

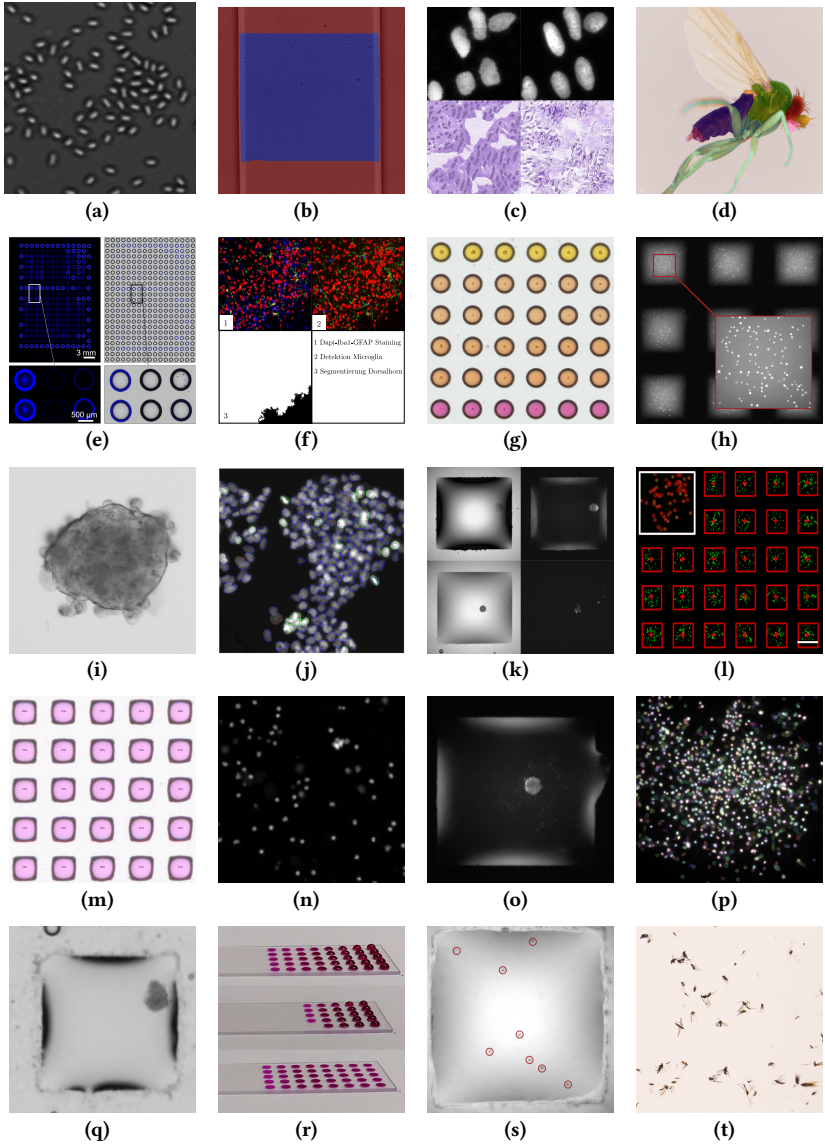


Abbildung 5.20: Beispielbilder zu Kooperationen. Exemplarische Beispielbilder sind zu den Projekten aus Tabelle 5.24 dargestellt [Böh23, Cha22, Cui23, Krü22, Sch23d, Str21a, Wie22, Wüh22].

6 Zusammenfassung und Ausblick

Hochdurchsatzexperimente (HDEs) sind von großer Relevanz für viele Versuche in verschiedenen wissenschaftlichen Disziplinen wie Chemie, Biologie, Pharmazie und Medizin. Gitterförmige Plattformen, z. B. Droplet Microarray oder Mikrotiterplatten, werden dabei häufig genutzt, um diese in der Praxis durchzuführen. Die nachgelagerte Analyse der Experimente erfolgt in der Regel durch die Interpretation von Bilddaten.

Allerdings fehlen Algorithmen für eine vollautomatisierte Auswertung der gitterförmigen HDEs. Insbesondere ist die Nutzung von datengetriebenen Methoden, wie beispielsweise Deep Learning (DL), erschwert, da annotierte Datensätze für die speziellen Bildverarbeitungsprobleme fehlen. Ferner mangelt es Forschern an Informationen hinsichtlich der Anwendung von annotations-effizientem Lernen. Im Allgemeinen ist die Anwendbarkeit von Algorithmen für Forscher in interdisziplinären Projekten beeinträchtigt, da entsprechende Software-, Hardware- oder Datenmanagementlösungen nicht existieren.

Die Kernpunkte und zentralen Ergebnisse dieser Arbeit umfassen:

- Die Entwicklung eines umfassenden Konzepts zur Lösung der genannten Probleme, das mittels heterogenen Bilddatensätzen aus dem Bereich Biomedizin hinsichtlich des Nutzens in der Praxis evaluiert worden ist.
- Die gezeigten Methoden im Bereich des assistierten Annotierens helfen Nutzern effizient, also mit möglichst geringem Zeitaufwand, Annotationen von hoher Qualität zu generieren.
- Der Einfluss von verschiedenen Typen verrauschter Annotationen ist untersucht worden. Ferner erfolgte die Präsentation von entwickelten

Methoden zur automatisierten Inspektion bezüglich verrauschten Datensätzen. Es konnte gezeigt werden, dass der Einfluss von Annotationsfehlern von der Fehlerart bzw. dem Umfang der falschen Annotationen abhängt. Die bereitgestellten Methoden zur automatisierten Inspektion waren in der Lage unter den getroffenen Annahmen die Inspektion erfolgreich durchzuführen und damit Nutzer zu unterstützen.

- Zudem ermöglichte die entwickelte Methodik eine vollautomatisierte Auswertung der HDEs unter Berücksichtigung deren gitterförmigen Anordnung. Mittels des umfangreichen Portfolios zur Analyse einzelner Elemente konnten verschiedenste Probleme aus der Anwendung in hoher Genauigkeit gelöst werden. Dies lag unter anderem an der Entwicklung von maßgeschneiderten DL-Analysepipelines, die sich als generische Methoden zur Lösung heterogener Bildverarbeitungsprobleme eigneten.
- Der umfangreiche Methodenvergleich im Bereich „effizientes Lernen“ leistete einen wichtigen Beitrag für Anwender bei der Methodenauswahl unter den jeweiligen projektspezifischen Rahmenbedingungen. Verschiedene qualitative und quantitative Metriken wurden gezeigt, um die Methoden aus verschiedenen Perspektiven bewerten zu können. Es konnte jedoch festgehalten werden, dass insbesondere der Einsatz von Semi-Supervised Learning und Transfer Learning (TL) großes Potenzial für die Nutzung in der Praxis gezeigt hat. In diesem Zusammenhang bildete die entwickelte Methodik auf Basis von Bildmerkmalen sowie Unsicherheiten von Deep Neural Networks zur Quantifizierung der Ähnlichkeit zwischen Datensätzen einen wichtigen Baustein, um TL in der Praxis sinnvoll nutzen zu können.
- Mittels des Konzepts zur Anwendbarkeit der Methoden ist Forschern eigenständiges Handeln im Rahmen der Auswertung von HDEs ermöglicht worden. Dazu sind verschiedenste Softwarepakete inklusive grafischer Benutzeroberflächen entwickelt, unterschiedliche Lösungen zur Allokation von Hardwareressourcen bereitgestellt sowie Strategien zum Datenmanagement bzw. der Ausrollung von Softwareupdates diskutiert worden.

Die vorgestellten Methoden sind generisch hinsichtlich deren Anwendbarkeit bei unterschiedlichen Bildverarbeitungsproblemen. Allerdings haben sich die Untersuchungen im Rahmen dieser Arbeit meist auf zweidimensionale Bildsegmentierungen fokussiert. Die Beleuchtung von anderen Aufgabentypen, wie beispielsweise die Objekterkennung oder 3D-Daten, ist ein Punkt für zukünftige Arbeiten.

Bei der Verwendung von neuen Untersuchungstypen werden häufig ebenfalls neue, auf die Untersuchung zugeschnittene Auswertungen benötigt. Damit stellt die Entwicklung weiterer Methoden zur Analyse einzelner Elemente im Rahmen gitterförmiger HDEs ein fortlaufendes Arbeitspaket dar.

Zudem sind insbesondere Self-Supervised Learning und Deep Active Learning in weiteren Arbeiten zu beleuchten, da diese bei den Untersuchungen im Rahmen der Arbeit bisher kaum Vorteile im Vergleich zum überwachten Lernen gezeigt haben.

Darüber hinaus ist die Kombination verschiedener Typen verrauschter Annotationen im Bereich der Intra-Experten-Variabilität ein weiteres Arbeitspaket für zukünftige Arbeiten. Des Weiteren gilt es im Bereich verrauschter Annotationen auch neue Problemklassen zu erschließen, wie beispielsweise die panoptische Segmentierung.

Das vorgeschlagene Architekturkonzept nimmt an, dass Rechencluster oder Graphics Processing Unit-Server remote verfügbar sind. Ein Schritt weiter ohne das Erfordernis von Hardware im Allgemeinen hin zu vollständigen cloud-basierten Lösungen, die über Browser genutzt werden können, sind als Untersuchungspunkt in der Zukunft zu nennen.

Darüber hinaus beschränkt sich das vorgeschlagene Konzept in seinem Grundgerüst nicht auf die Hochdurchsatz-Bildverarbeitung. Untersuchungen bezüglich dessen Anwendbarkeit für andere Szenarien, wie z. B. der Datenanalyse von Zeitreihen im Energiekontext [Hei20] oder der eindimensionalen Untersuchung von Materialspektren [Sch21d], gilt es zukünftig durchzuführen.

Literatur

- [Abd21] ABDAR, Moloud u. a.: „A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges“. In: *Information Fusion* 76 (2021), S. 243–297. DOI: [10.1016/j.inffus.2021.05.008](https://doi.org/10.1016/j.inffus.2021.05.008) (siehe S. 12, 31).
- [Ach19] ACHILLE, Alessandro; LAM, Michael; TEWARI, Rahul; RAVICHANDRAN, Avinash; MAJI, Subhransu; FOWLKES, Charles; SOATTO, Stefano und PERONA, Pietro: „Task2Vec: Task Embedding for Meta-Learning“. In: *IEEE/CVF International Conference on Computer Vision*. Seoul, Südkorea, 2019, S. 6429–6438. DOI: [10.1109/ICCV.2019.00653](https://doi.org/10.1109/ICCV.2019.00653) (siehe S. 38).
- [Ada94] ADAMS, R. und BISCHOF, L.: „Seeded Region Growing“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.6 (1994), S. 641–647. DOI: [10.1109/34.295913](https://doi.org/10.1109/34.295913) (siehe S. 25).
- [Ahm21] AHMAD, Hafiz Mughees; RAHIMI, Afshin und HAYAT, Khizer: „Deep Learning Transforming the Manufacturing Industry: A Case Study“. In: *IEEE International Conference on High Performance Computing & Communications*. Haikou, China, 2021, S. 1286–1291. DOI: [10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00196](https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00196) (siehe S. 2).
- [Ala14] ALAIN, Guillaume und BENGIO, Yoshua: „What Regularized Auto-Encoders Learn from the Data-Generating Distribution“. In: *Journal of Machine Learning Research* 15.1 (2014), S. 3563–3593. DOI: [10.5555/2627435.2750359](https://doi.org/10.5555/2627435.2750359) (siehe S. 32).
- [Alp10] ALPAYDIN, Ethem: *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2010 (siehe S. 9).

- [Als16] ALSHUT, Rüdiger: „Konzept für Bildanalysen in Hochdurchsatz-Systemen am Beispiel des Zebrafischblümlings“. Diss. Karlsruhe, Deutschland: Karlsruher Institut für Technologie (KIT), 2016. DOI: [10.5445/IR/1000069231](https://doi.org/10.5445/IR/1000069231) (siehe S. 1, 21).
- [An10] AN, W. Frank und TOLLIDAY, Nicola: „Cell-Based Assays for High-Throughput Screening“. In: *Molecular Biotechnology* 45.2 (2010), S. 180–186. DOI: [10.1007/s12033-010-9251-z](https://doi.org/10.1007/s12033-010-9251-z) (siehe S. 1, 22).
- [Ash20] ASH, Jordan T.; ZHANG, Chicheng; KRISHNAMURTHY, Akshay; LANGFORD, John und AGARWAL, Alekh: „Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds“. In: *International Conference on Learning Representations*. Verfügbar: <https://openreview.net/forum?id=ryghZJBKPS>. Zugriff: 12.04.2023. Virtuelles Event, Äthiopien, 2020 (siehe S. 32).
- [Att14] ATTENE-RAMOS, Matias; AUSTIN, Christopher P. und XIA, Menghang: „High Throughput Screening“. In: *Encyclopedia of Toxicology*. Hrsg. von WEXLER, Philip. Academic Press, 2014, S. 916–917. DOI: [10.1016/B978-0-12-386454-3.00209-8](https://doi.org/10.1016/B978-0-12-386454-3.00209-8) (siehe S. 1).
- [Aul20] AULD, Douglas S. u. a.: „Microplate Selection and Recommended Practices in High-throughput Screening and Quantitative Biology“. In: *Assay Guidance Manual* (2020). Verfügbar: <https://www.ncbi.nlm.nih.gov/books/NBK558077/>. Zugriff: 11.04.2023 (siehe S. 22).
- [Ban19] BANDARU, Praveen u. a.: „A Microfabricated Sandwiching Assay for Nanoliter and High-Throughput Biomarker Screening“. In: *Small* 15.15 (2019), S. 1900300. DOI: [10.1002/smll.201900300](https://doi.org/10.1002/smll.201900300) (siehe S. 1, 22).
- [Bar19] BARTSCHAT, Andreas: ImageLabelingTool. [Online]. Verfügbar: <https://bitbucket.org/abartschat/imagelabelingtool>. Zugriff: 28.04.2022. 2019 (siehe S. 15, 16, 115).
- [Bäu21] BÄUERLE, Simon; BÖHLAND, Moritz; BARTH, Jonas; REISCHL, Markus; STEIMER, Andreas und MIKUT, Ralf: „CAD-to-real: enabling deep neural networks for 3D pose estimation of electronic control units: A Transferable and Automated Approach for Industrial

- Use Cases“. In: *at - Automatisierungstechnik* 69.10 (2021), S. 880–891. DOI: [10.1515/auto-2021-0020](https://doi.org/10.1515/auto-2021-0020) (siehe S. 2).
- [Bel57] BELLMAN, Richard Ernest: *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957. DOI: [10.1126/science.127.3304.976.a](https://doi.org/10.1126/science.127.3304.976.a) (siehe S. 36).
- [Ben15] BENGEL, Günther; BAUN, Christian; KUNZE, Marcel und STUCKY, Karl-Uwe: *Masterkurs Parallele und Verteilte Systeme: Grundlagen und Programmierung von Multicore-Prozessoren, Multiprozessoren, Cluster, Grid und Cloud*. 2. Aufl. Wiesbaden, Deutschland: Springer Vieweg, 2015. DOI: [10.1007/978-3-8348-2151-5](https://doi.org/10.1007/978-3-8348-2151-5) (siehe S. 86).
- [Ben20] BENZ, Maximilian; ASPERGER, Arndt; HAMESTER, Meike; WELLE, Alexander; HEISLER, Stefan und LEVKIN, Pavel A.: „A Combined High-Throughput and High-Content Platform for Unified on-Chip Synthesis, Characterization and Biological Screening“. In: *Nature Communications* 11.1 (2020), S. 5391. DOI: [10.1038/s41467-020-19040-0](https://doi.org/10.1038/s41467-020-19040-0) (siehe S. 1, 21).
- [Ber12] BERGSTRA, James und BENGIO, Yoshua: „Random Search for Hyper-Parameter Optimization“. In: *Journal of Machine Learning Research* 13.2 (2012). DOI: [10.5555/2188385.2188395](https://doi.org/10.5555/2188385.2188395) (siehe S. 92).
- [Ber19] BERG, Stuart u. a.: „Ilastik: Interactive Machine Learning for (Bio)Image Analysis“. In: *Nature Methods* 16.12 (2019), S. 1226–1232. DOI: [10.1038/s41592-019-0582-9](https://doi.org/10.1038/s41592-019-0582-9) (siehe S. 16).
- [Beu79] BEUCHER, Serge und LANTUÉJOUL, Christian: „Use of Watersheds in Contour Detection“. In: *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation*. Bd. 132. Verfügbar: <http://cmm.enscm.fr/~beucher/publi/watershed.pdf>. Zugriff: 11.04.2023. Rennes, Frankreich, 1979, S. 17–21 (siehe S. 16, 25, 27).
- [Bie20] BIEWALD, Lukas: *Experiment Tracking with Weights and Biases*. [Online]. Verfügbar: <https://www.wandb.com/>. Zugriff: 24.04.2021. 2020 (siehe S. 91).

- [Böh23] BÖHLAND, Moritz; BRUCH, Roman; LÖFFLER, Katharina und REISCHL, Markus: „Unsupervised GAN epoch selection for biomedical data synthesis“. In: *Current Directions in Biomedical Engineering (akzeptiert)* (2023) (siehe S. 155, 156, 158).
- [Bra00] BRADSKI, Gary: „The openCV Library.“ In: *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25.11 (2000), S. 120–123 (siehe S. 16, 56, 91, 116).
- [Bré17] BRÉHÉRET, Amaury: Pixel Annotation Tool. [Online]. Verfügbar: <https://github.com/abreheret/PixelAnnotationTool>. Zugriff: 28.04.2022. 2017 (siehe S. 15, 16).
- [Bri01] BRIECHLE, Kai und HANEBECK, Uwe D.: „Template Matching Using Fast Normalized Cross Correlation“. In: *Optical Pattern Recognition*. Bd. 4387. Orlando, FL, USA, 2001, S. 95–102. DOI: [10.1117/12.421129](https://doi.org/10.1117/12.421129) (siehe S. 25).
- [Bro22] BROWN, Sarah: Why it's time for 'data-centric artificial intelligence'. MIT Sloan School of Management. [Online]. Verfügbar: <https://mitsloan.mit.edu/ideas-made-to-matter/why-its-time-data-centric-artificial-intelligence>. Zugriff: 26.05.2023. 2022 (siehe S. 2).
- [Bus20] BUSLAEV, Alexander; IGLOVIKOV, Vladimir I.; KHVEDCHENYA, Eugene; PARINOV, Alex; DRUZHININ, Mikhail und KALININ, Alexandr A.: „Albumentations: Fast and Flexible Image Augmentations“. In: *Information - an International Interdisciplinary Journal* 11.2 (2020). DOI: [10.3390/info11020125](https://doi.org/10.3390/info11020125) (siehe S. 91, 256).
- [Cai19] CAICEDO, Juan C. u. a.: „Nucleus Segmentation across Imaging Experiments: The 2018 Data Science Bowl“. In: *Nature Methods* 16.12 (2019), S. 1247–1253. DOI: [10.1038/s41592-019-0612-7](https://doi.org/10.1038/s41592-019-0612-7) (siehe S. 99, 101).
- [Car06] CARPENTER, Anne E. u. a.: „CellProfiler: Image Analysis Software for Identifying and Quantifying Cell Phenotypes“. In: *Genome Biology* 7.10 (2006). DOI: [10.1186/gb-2006-7-10-r100](https://doi.org/10.1186/gb-2006-7-10-r100) (siehe S. 28, 130).

- [Cha06] CHAN, Tammy Pui Shan; MALIK, Preeti und SINGH, Rahul: „An Interactive Visualization-Based Approach for High Throughput Screening Information Management in Drug Discovery“. In: *International Conference of the IEEE Engineering in Medicine and Biology Society*. New York, NY, USA, 2006, S. 5794–5797. DOI: [10.1109/IEMBS.2006.259448](https://doi.org/10.1109/IEMBS.2006.259448) (siehe S. 28).
- [Cha13] CHANG, Kyle u. a.: „The Cancer Genome Atlas Pan-Cancer Analysis Project“. In: *Nature Genetics* 45.10 (2013), S. 1113–1120. DOI: [10.1038/ng.2764](https://doi.org/10.1038/ng.2764) (siehe S. 37).
- [Cha20a] CHAN, Yao-Chun; LI, Mingchen und OYMAK, Samet: On the Marginal Benefit of Active Learning: Does Self-Supervision Eat Its Cake? arXiv:2011.08121, [Online]. Verfügbar: <https://arxiv.org/pdf/2011.08121>. Zugriff: 21.02.2023. 2020 (siehe S. 30).
- [Cha20b] CHARNG, Jason u. a.: „Deep Learning Segmentation of Hyperautofluorescent Fleck Lesions in Stargardt Disease“. In: *Scientific Reports* 10.1 (2020), S. 16491. DOI: [10.1038/s41598-020-73339-y](https://doi.org/10.1038/s41598-020-73339-y) (siehe S. 26).
- [Che10] CHECHIK, Gal; SHARMA, Varun; SHALIT, Uri und BENGIO, Samy: „Large Scale Online Learning of Image Similarity through Ranking“. In: *Journal of Machine Learning Research* 11 (2010), S. 1109–1135. DOI: [10.5555/1756006.1756042](https://doi.org/10.5555/1756006.1756042) (siehe S. 243).
- [Che17] CHEPLYGINA, Veronika; MOESKOPS, Pim; VETA, Mitko; BOZORG, Behdad Dasht und PLUIM, Josien P. W.: „Exploring the Similarity of Medical Imaging Classification Problems“. In: *Joint MICCAI Workshop on Computing and Visualization for Intravascular Imaging and Computer Assisted Stenting*. Quebec City, Kanada, 2017, S. 59–66. DOI: [10.1007/978-3-319-67534-3_7](https://doi.org/10.1007/978-3-319-67534-3_7) (siehe S. 36).
- [Che18] CHEN, Xiaoling u. a.: „DataMed – an Open Source Discovery Index for Finding Biomedical Datasets“. In: *Journal of the American Medical Informatics Association* 25.3 (2018), S. 300–308. DOI: [10.1093/jamia/ocx121](https://doi.org/10.1093/jamia/ocx121) (siehe S. 36).

- [Che20] CHEN, Ting; KORNBLITH, Simon; NOROUZI, Mohammad und HINTON, Geoffrey: „A Simple Framework for Contrastive Learning of Visual Representations“. In: *International Conference on Machine Learning*. Bd. 119. Virtuelles Event, 2020, S. 1597–1607. DOI: [10.5555/3524938.3525087](https://doi.org/10.5555/3524938.3525087) (siehe S. 33, 84).
- [Che22] CHENG, Heng-Tze: LaMDA: Towards Safe, Grounded, and High-Quality Dialog Models for Everything. Google Research Blog. [Online]. Verfügbar: <https://ai.googleblog.com/2022/01/lamda-towards-safe-grounded-and-high.html>. Zugriff: 03.04.2023. 2022 (siehe S. 36).
- [Che79] CHEN, Patrick C und PAVLIDIS, Theodosios: „Segmentation by Texture Using a Co-Occurrence Matrix and a Split-and-Merge Algorithm“. In: *Computer Graphics and Image Processing* 10.2 (1979), S. 172–182. DOI: [10.1016/0146-664X\(79\)90049-2](https://doi.org/10.1016/0146-664X(79)90049-2) (siehe S. 25).
- [Che95] CHENG, Yizong: „Mean shift, mode seeking, and clustering“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8 (1995), S. 790–799. DOI: [10.1109/34.400568](https://doi.org/10.1109/34.400568) (siehe S. 25).
- [Chi20] CHI, Weicheng; MA, Lin; WU, Junjie; CHEN, Mingli; LU, Weiguo und GU, Xuejun: „Deep Learning-Based Medical Image Segmentation with Limited Labels“. In: *Physics in Medicine & Biology* 65.23 (2020), S. 235001. DOI: [10.1088/1361-6560/abc363](https://doi.org/10.1088/1361-6560/abc363) (siehe S. 15, 29).
- [Cho19] CHOUHAN, Siddharth Singh; SINGH, Uday Pratap; KAUL, Ajay und JAIN, Sanjeev: „A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology“. In: *4th International Conference on Information Systems and Computer Networks*. Mathura, Indien, 2019, S. 700–707. DOI: [10.1109/ISCON47742.2019.9036158](https://doi.org/10.1109/ISCON47742.2019.9036158) (siehe S. 37).
- [Cho20] CHOLLET, Francois: Transfer learning and fine-tuning. Keras Developer Guides. [Online]. Verfügbar: https://keras.io/guides/transfer_learning/#the-typical-transferlearning-workflow. Zugriff: 16.01.2023. 2020 (siehe S. 38, 83).

- [Çiç16] ÇIÇEK, Özgün; ABDULKADIR, Ahmed; LIENKAMP, Soeren S.; BROX, Thomas und RONNEBERGER, Olaf: „3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation“. In: *Medical Image Computing and Computer-Assisted Intervention*. Athen, Griechenland, 2016. DOI: [10.1007/978-3-319-46723-8_49](https://doi.org/10.1007/978-3-319-46723-8_49) (siehe S. 26).
- [Cod18] CODELLA, Noel C. F. u. a.: „Skin Lesion Analysis toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging, Hosted by the International Skin Imaging Collaboration“. In: *International Symposium on Biomedical Imaging*. Washington, DC, USA, 2018, S. 168–172. DOI: [10.1109/ISBI.2018.8363547](https://doi.org/10.1109/ISBI.2018.8363547) (siehe S. 99, 101).
- [Col20] COLEMAN, Cody; YEH, Christopher; MUSSMANN, Stephen; MIRZASOLEIMAN, Baharan; BAILIS, Peter; LIANG, Percy; LESKOVEC, Jure und ZAHARIA, Matei: „Selection via Proxy: Efficient Data Selection for Deep Learning“. In: *International Conference on Learning Representations*. Verfügbar: <https://openreview.net/forum?id=HJg2b0VYDr>. Zugriff: 12.04.2023. Virtuelles Event, Äthiopien, 2020 (siehe S. 32).
- [Cor19] CORBIÈRE, Charles; THOME, Nicolas; BAR-HEN, Avner; CORD, Matthieu und PÉREZ, Patrick: „Addressing Failure Prediction by Learning Model Confidence“. In: *Advances in Neural Information Processing Systems*. Vancouver, Kanada, 2019, S. 2902–2913. DOI: [10.5555/3454287.3454548](https://doi.org/10.5555/3454287.3454548) (siehe S. 12, 31).
- [Den09] DENG, Jia; DONG, Wei; SOCHER, Richard; LI, Li-Jia; LI, Kai und FEI-FEI, Li: „ImageNet: A Large-Scale Hierarchical Image Database“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, USA, 2009, S. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848) (siehe S. 10, 38, 39, 55, 104).
- [Des11] DESERNO, Thomas M.: „Fundamentals of Biomedical Image Processing“. In: *Biomedical Image Processing*. Hrsg. von DESERNO, Thomas M. Heidelberg/Berlin, Deutschland: Springer, 2011, S. 1–51. DOI: [10.1007/978-3-642-15816-2_1](https://doi.org/10.1007/978-3-642-15816-2_1) (siehe S. 21–23).

- [Doe19] DOERR, Allison; VOGT, Nina; TANG, Lin; TANG, Lei; SINGH, Arunima; MUKHOPADHYAY, Madhura; MARX, Vivien; AKHTAR, Anam; INGLIS, George und ELDRIDGE, Cara: „Deep Learning Gets Scope Time“. In: *Nature Methods* 16.12 (2019), S. 1195. DOI: [10.1038/s41592-019-0670-x](https://doi.org/10.1038/s41592-019-0670-x) (siehe S. 2, 15, 22, 26).
- [Don19] DONAHUE, Jeff und SIMONYAN, Karen: „Large Scale Adversarial Representation Learning“. In: *Advances in Neural Information Processing Systems* 32 (2019), S. 10541–10551. DOI: [10.5555/3454287.3455233](https://doi.org/10.5555/3454287.3455233) (siehe S. 32).
- [Dös16] DÖSSEL, Olaf: *Bildgebende Verfahren in der Medizin: Von der Technik zur medizinischen Anwendung*. 2. Auflage. Wiesbaden, Deutschland: Springer Vieweg, 2016. DOI: [10.1007/978-3-642-54407-1](https://doi.org/10.1007/978-3-642-54407-1) (siehe S. 22).
- [Dos21] DOSOVITSKIY, Alexey u. a.: „An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale“. In: *International Conference on Learning Representations*. [Online]. Verfügbar: <https://openreview.net/pdf?id=YicbFdNTTy/>. Zugriff: 21.02.2023. Virtuelles Event, 2021 (siehe S. 27).
- [Edl21] EDLUND, Christoffer; JACKSON, Timothy R.; KHALID, Nabeel; BEVAN, Nicola; DALE, Timothy; DENGEL, Andreas; AHMED, Sheraz; TRYGG, Johan und SJÖGREN, Rickard: „LIVECell—A Large-Scale Dataset for Label-Free Live Cell Segmentation“. In: *Nature Methods* 18.9 (2021), S. 1038–1045. DOI: [10.1038/s41592-021-01249-6](https://doi.org/10.1038/s41592-021-01249-6) (siehe S. 96, 99, 101, 246, 247).
- [Eht17] EHTESHAMI BEJNORDI, Babak; VETA, Mitko; JOHANNES VAN DIEST, Paul; VAN GINNEKEN, Bram; KARSEMEIJER, Nico; LITJENS, Geert; VAN DER LAAK, Jeroen A. W. M. und AND THE CAMELYON16 CONSORTIUM: „Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer“. In: *JAMA* 318.22 (2017), S. 2199–2210. DOI: [10.1001/jama.2017.14585](https://doi.org/10.1001/jama.2017.14585) (siehe S. 37).

- [Eng21] ENGLBRECHT, Fabian; RUIDER, Iris E. und BAUSCH, Andreas R.: „Automatic Image Annotation for Fluorescent Cell Nuclei Segmentation“. In: *PLOS ONE* 16.4 (2021), S. 1–13. DOI: [10.1371/journal.pone.0250093](https://doi.org/10.1371/journal.pone.0250093) (siehe S. 16).
- [Eve10] EVERINGHAM, Mark; VAN GOOL, Luc; WILLIAMS, Christopher K. I.; WINN, John und ZISSERMAN, Andrew: „The Pascal Visual Object Classes (VOC) Challenge“. In: *International Journal of Computer Vision* 88.2 (2010), S. 303–338. DOI: [10.1007/s11263-009-0275-4](https://doi.org/10.1007/s11263-009-0275-4) (siehe S. 246, 247).
- [Eykh18] EYKHOLT, Kevin; EVTIMOV, Ivan; FERNANDES, Earleence; LI, Bo; RAHMATI, Amir; XIAO, Chaowei; PRAKASH, Atul; KOHNO, Tadayoshi und SONG, Dawn: „Robust Physical-World Attacks on Deep Learning Visual Classification“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA, 2018, S. 1625–1634. DOI: [10.1109/CVPR.2018.00175](https://doi.org/10.1109/CVPR.2018.00175) (siehe S. 12).
- [Fal19a] FALCON, William: PyTorch Lightning. [Online]. Verfügbar: <https://www.pytorchlightning.ai/index.html>. Zugriff: 24.04.2021. 2019 (siehe S. 91).
- [Fal19b] FALK, T. u. a.: „U-Net – Deep Learning for Cell Counting, Detection, and Morphometry“. In: *Nature Methods* 16 (2019), S. 67–70. DOI: [10.1038/s41592-018-0261-2](https://doi.org/10.1038/s41592-018-0261-2) (siehe S. 28).
- [For07] FORLINES, Clifton; WIGDOR, Daniel; SHEN, Chia und BALAKRISHNAN, Ravin: „Direct-Touch vs. Mouse Input for Tabletop Displays“. In: *Conference on Human Factors in Computing Systems*. San Jose, CA, USA, 2007, S. 647–656. DOI: [10.1145/1240624.1240726](https://doi.org/10.1145/1240624.1240726) (siehe S. 17).
- [Gal15] GALLERT, C.; LEHMANN, R.; RODDELKOPF, T.; JUNGINGER, S. und THUROW, K.: „High Throughput Screening System for Screening of 3D Cell Cultures“. In: *IEEE International Instrumentation and Measurement Technology Conference*. Pisa, Italien, 2015, S. 1302–1307. DOI: [10.1109/I2MTC.2015.7151461](https://doi.org/10.1109/I2MTC.2015.7151461) (siehe S. 21).

- [Gal16] GAL, Yarin und GHAHRAMANI, Zoubin: „Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning“. In: *International Conference on Machine Learning*. Bd. 48. New York, NY, USA, 2016, S. 1050–1059. DOI: [10.5555/3045390.3045502](https://doi.org/10.5555/3045390.3045502) (siehe S. 12, 31).
- [Gal17] GAL, Yarin; ISLAM, Riashat und GHAHRAMANI, Zoubin: „Deep Bayesian Active Learning with Image Data“. In: *International Conference on Machine Learning*. Sydney, Australien, 2017, S. 1183–1192. DOI: [10.5555/3305381.3305504](https://doi.org/10.5555/3305381.3305504) (siehe S. 31, 84).
- [Gam19] GAMPER, Jevgenij; ALEMI KOOHBANANI, Navid; BENET, Ksenija; KHURAM, Ali und RAJPOOT, Nasir: „PanNuke: An Open Pan-Cancer Histology Dataset for Nuclei Instance Segmentation and Classification“. In: *Digital Pathology*. Warwick, England, 2019, S. 11–19. DOI: [10.1007/978-3-030-23937-4_2](https://doi.org/10.1007/978-3-030-23937-4_2) (siehe S. 246, 247).
- [Gan22] GANGULI, Swetava; IYER, C. V. Krishnakumar und PANDEY, Vipul: „Reachability Embeddings: Self-supervised Representation Learning from Spatiotemporal Motion Trajectories for Multimodal Geospatial Computer Vision“. In: *23rd IEEE International Conference on Mobile Data Management*. Paphos, Cyprus, 2022, S. 44–53. DOI: [10.1109/MDM55031.2022.00028](https://doi.org/10.1109/MDM55031.2022.00028) (siehe S. 32).
- [Gar22] GARCIA SANTA CRUZ, Beatriz u. a.: „Generalising from Conventional Pipelines Using Deep Learning in High-Throughput Screening Workflows“. In: *Scientific Reports* 12.1 (2022), S. 11465. DOI: [10.1038/s41598-022-15623-7](https://doi.org/10.1038/s41598-022-15623-7) (siehe S. 1, 21).
- [Gat16] GATYS, Leon A.; ECKER, Alexander S. und BETHGE, Matthias: „Image Style Transfer Using Convolutional Neural Networks“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA, 2016, S. 2414–2423. DOI: [10.1109/CVPR.2016.265](https://doi.org/10.1109/CVPR.2016.265) (siehe S. 36).
- [Geh17] GEHAN, Malia A. u. a.: „PlantCV v2: Image Analysis Software for High-Throughput Plant Phenotyping“. In: *PeerJ* 5 (2017), e4088. DOI: [10.7717/peerj.4088](https://doi.org/10.7717/peerj.4088) (siehe S. 23, 28, 130).

- [Gho17] GHOSH, Aritra; KUMAR, Himanshu und SASTRY, P. S.: „Robust Loss Functions under Label Noise for Deep Neural Networks“. In: *AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA, 2017, S. 1919–1925. DOI: [10.5555/3298483.3298518](https://doi.org/10.5555/3298483.3298518) (siehe S. 19).
- [God21] GODAU, Patrick und MAIER-HEIN, Lena: „Task Fingerprinting for Meta Learning in Biomedical Image Analysis“. In: *Medical Image Computing and Computer Assisted Interventions*. Straßburg, Frankreich, 2021. DOI: [10.1007/978-3-030-87202-1_42](https://doi.org/10.1007/978-3-030-87202-1_42) (siehe S. 37).
- [Góm21a] GÓMEZ-DE-MARISCAL, Estibaliz; GARCÍA-LÓPEZ-DE-HARO, Carlos; OUYANG, Wei; DONATI, Laurène; LUNDBERG, Emma; UNSER, Michael; MUÑOZ-BARRUTIA, Arrate und SAGE, Daniel: „DeepImageJ: A User-Friendly Environment to Run Deep Learning Models in ImageJ“. In: *Nature Methods* 18.10 (2021), S. 1192–1195. DOI: [10.1038/s41592-021-01262-9](https://doi.org/10.1038/s41592-021-01262-9) (siehe S. 2).
- [Góm21b] GÓMEZ-DE-MARISCAL, Estibaliz; GARCÍA-LÓPEZ-DE-HARO, Carlos; OUYANG, Wei; DONATI, Laurène; LUNDBERG, Emma; UNSER, Michael; MUÑOZ-BARRUTIA, Arrate und SAGE, Daniel: „DeepImageJ: A User-Friendly Environment to Run Deep Learning Models in ImageJ“. In: *Nature Methods* 18.10 (2021), S. 1192–1195. DOI: [10.1038/s41592-021-01262-9](https://doi.org/10.1038/s41592-021-01262-9) (siehe S. 28).
- [Goo15] GOODFELLOW, Ian; SHLENS, Jonathon und SZEGEDY, Christian: „Explaining and Harnessing Adversarial Examples“. In: *International Conference on Learning Representations*. Verfügbar: <https://arxiv.org/pdf/1412.6572>. Zugriff: 12.04.2023. 2015 (siehe S. 13).
- [Goo16] GOODFELLOW, Ian; BENGIO, Yoshua und COURVILLE, Aaron: *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016 (siehe S. 10, 13, 14, 26).
- [Gra19] GRAHAM, Simon; VU, Quoc Dang; RAZA, Shan E Ahmed; AZAM, Ayesha; TSANG, Yee Wah; KWAK, Jin Tae und RAJPOOT, Nasir: „Hover-Net: Simultaneous Segmentation and Classification of

- Nuclei in Multi-Tissue Histology Images“. In: *Medical Image Analysis* 58 (2019), S. 101563. DOI: [10.1016/j.media.2019.101563](https://doi.org/10.1016/j.media.2019.101563) (siehe S. 1, 48).
- [Gri19] GRINBERG, Michael: Dokumentenvorlage für die Schriftenreihe „Karlsruher Schriften zur Anthropomatik“. [Online]. Verfügbar: <https://gitlab-ext.iosb.fraunhofer.de/grinbe/Dissertation-Template-KIT>. Zugriff: 09.07.2021. 2019 (siehe S. vi).
- [Gri20] GRILL, Jean-Bastien u. a.: „Bootstrap Your Own Latent - a New Approach to Self-Supervised Learning“. In: *Advances in Neural Information Processing Systems*. Bd. 33. Vancouver, Kanada, 2020, S. 21271–21284. DOI: [10.5555/3495724.3497510](https://doi.org/10.5555/3495724.3497510) (siehe S. 33, 84).
- [Guo17] GUO, Chuan; PLEISS, Geoff; SUN, Yu und WEINBERGER, Kilian Q.: „On Calibration of Modern Neural Networks“. In: *International Conference on Machine Learning*. Verfügbar: <http://proceedings.mlr.press/v70/guo17a/guo17a.pdf>. Zugriff: 11.04.2023. Sydney, Australien, 2017, S. 1321–1330 (siehe S. 12).
- [Guo18] GUO, Sheng; HUANG, Weilin; ZHANG, Haozhi; ZHUANG, Chenfan; DONG, Dengke; SCOTT, Matthew R. und HUANG, Dinglong: „CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images“. In: *European Conference on Computer Vision*. München, Deutschland, 2018, S. 139–154. DOI: [10.1007/978-3-030-01249-6_9](https://doi.org/10.1007/978-3-030-01249-6_9) (siehe S. 19).
- [Guo19] GUO, Yunhui; SHI, Honghui; KUMAR, Abhishek; GRAUMAN, Kristen; ROSING, Tajana und FERIS, Rogerio: „SpotTune: Transfer Learning through Adaptive Fine-Tuning“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, USA, 2019, S. 4800–4809. DOI: [10.1109/CVPR.2019.00494](https://doi.org/10.1109/CVPR.2019.00494) (siehe S. 39).
- [He15] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing und SUN, Jian: „Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification“. In: *International Conference on Computer Vision*. Santiago, Chile, 2015, S. 1026–1034. DOI: [10.1109/ICCV.2015.123](https://doi.org/10.1109/ICCV.2015.123) (siehe S. 32, 92).

- [He16] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing und SUN, Jian: „Deep Residual Learning for Image Recognition“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA, 2016, S. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90) (siehe S. 26, 55, 104, 242, 255).
- [He17] HE, Kaiming; GKIOXARI, Georgia; DOLLÁR, Piotr und GIRSHICK, Ross B.: Mask R-CNN. arXiv:1703.06870, [Online]. Verfügbar: <https://arxiv.org/pdf/1703.06870>. Zugriff: 21.02.2023. 2017 (siehe S. 27).
- [He20] HE, Kaiming; FAN, Haoqi; WU, Yuxin; XIE, Saining und GIRSHICK, Ross: „Momentum Contrast for Unsupervised Visual Representation Learning“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA, 2020, S. 9726–9735. DOI: [10.1109/CVPR42600.2020.00975](https://doi.org/10.1109/CVPR42600.2020.00975) (siehe S. 33, 84).
- [Hei20] HEIDRICH, Benedikt; TUROWSKI, Marian; LUDWIG, Nicole; MIKUT, Ralf und HAGENMEYER, Veit: „Forecasting energy time series with profile neural networks“. In: *e-Energy: ACM International Conference on Future Energy Systems*. Virtuelles Event, Australien, 2020, S. 220–230. DOI: [10.1145/3396851.3397683](https://doi.org/10.1145/3396851.3397683) (siehe S. 161).
- [Hol20] HOLLANDI, Réka; DIÓSDI, Ákos; HOLLANDI, Gábor; MOSHKOV, Nikita und HORVÁTH, Péter: „AnnotatorJ: An ImageJ Plugin to Ease Hand Annotation of Cellular Compartments“. In: *Molecular Biology of the Cell* 31.20 (2020), S. 2179–2186. DOI: [10.1091/mbc.E20-02-0156](https://doi.org/10.1091/mbc.E20-02-0156) (siehe S. 16, 20).
- [Hou60] HOUGH, Paul V.: „Method and Means for Recognizing Complex Patterns“. US Patent US3069654A. Verfügbar: <https://patents.google.com/patent/US3069654A/en>. Zugriff: 27.02.2023. 1960 (siehe S. 25).
- [Hua21] HUANG, Ziyi; LAM, Henry und ZHANG, Haofeng: Quantifying Epistemic Uncertainty in Deep Learning. arXiv:2110.12122, [Online]. Verfügbar: <https://arxiv.org/pdf/2110.12122>. Zugriff: 12.04.2023. 2021 (siehe S. 12).

- [Hub62] HUBEL, David H. und WIESEL, Torsten N.: „Receptive Fields, Binocular Interaction and Functional Architecture in the Cat’s Visual Cortex“. In: *The Journal of Physiology* 160.1 (1962), S. 106–154. DOI: [10.1113/jphysiol.1962.sp006837](https://doi.org/10.1113/jphysiol.1962.sp006837) (siehe S. 10).
- [Hub64] HUBER, Peter J.: „Robust Estimation of a Location Parameter“. In: *The Annals of Mathematical Statistics* 35.1 (1964), S. 73–101. DOI: [10.1214/aoms/1177703732](https://doi.org/10.1214/aoms/1177703732) (siehe S. 257).
- [Hül21] HÜLLERMEIER, Eyke und WAEGEMAN, Willem: „Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods“. In: *Machine Learning* 110.3 (2021), S. 457–506. DOI: [10.1007/s10994-021-05946-3](https://doi.org/10.1007/s10994-021-05946-3) (siehe S. 12).
- [Ing08] INGLESE, James und AULD, Douglas S.: „High Throughput Screening (HTS) Techniques: Applications in Chemical Biology“. In: *Wiley Encyclopedia of Chemical Biology*. Hrsg. von BEGLEY, Tadhg P. John Wiley & Sons, Ltd, 2008, S. 1–15. DOI: [10.1002/9780470048672.wecb223](https://doi.org/10.1002/9780470048672.wecb223) (siehe S. 1).
- [Iqb18] IQBAL, Haris: Plot Neural Net. [Online]. Verfügbar: <https://github.com/HarisIqbal88/PlotNeuralNet>. Zugriff: 17.05.2023. 2018 (siehe S. 254, 255).
- [Ise21b] ISENSEE, Fabian; JAEGER, Paul F.; KOHL, Simon A. A.; PETERSEN, Jens und MAIER-HEIN, Klaus H.: „nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation“. In: *Nature Methods* 18.2 (2021), S. 203–211. DOI: [10.1038/s41592-020-01008-z](https://doi.org/10.1038/s41592-020-01008-z) (siehe S. 19, 26).
- [Jaa22] JAAKS, Patricia u. a.: „Effective Drug Combinations in Breast, Colon and Pancreatic Cancer Cells“. In: *Nature* 603.7899 (2022), S. 166–173. DOI: [10.1038/s41586-022-04437-2](https://doi.org/10.1038/s41586-022-04437-2) (siehe S. 1, 21).
- [Jäh12] JÄHNE, Bernd: Digitale Bildverarbeitung. 5. Auflage. Berlin/Heidelberg, Deutschland: Springer, 2012. DOI: [10.1007/978-3-642-04952-1](https://doi.org/10.1007/978-3-642-04952-1) (siehe S. 21, 24, 25).

- [Jej12] JEJKAL, Thomas; HARTMANN, Volker; STOTZKA, Rainer; OTTE, Jens; GARCIA, Ariel; VAN WEZEL, Jos und STREIT, Achim: „LAMBDA – the LSDF Execution Framework for Data Intensive Applications“. In: *Euromicro International Conference on Parallel, Distributed and Network-Based Processing*. München, Deutschland, 2012, S. 213–220. DOI: [10.1109/PDP.2012.69](https://doi.org/10.1109/PDP.2012.69) (siehe S. 28, 91, 271).
- [Jen19] JENA, Rohit und AWATE, Suyash P.: „A Bayesian Neural Net to Segment Images with Uncertainty Estimates and Good Calibration“. In: *Information Processing in Medical Imaging*. Hong Kong, China, 2019, S. 3–15. DOI: [10.1007/978-3-030-20351-1_1](https://doi.org/10.1007/978-3-030-20351-1_1) (siehe S. 13).
- [Jia18] JIANG, Heinrich; KIM, Been; GUAN, Melody und GUPTA, Maya: „To Trust or Not to Trust a Classifier“. In: *Advances in Neural Information Processing Systems*. Bd. 31. Montreal, Kanada, 2018. DOI: [10.5555/3327345.3327458](https://doi.org/10.5555/3327345.3327458) (siehe S. 12).
- [Jin21] JING, Longlong und TIAN, Yingli: „Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2021), S. 4037–4058. DOI: [10.1109/TPAMI.2020.2992393](https://doi.org/10.1109/TPAMI.2020.2992393) (siehe S. 32, 33).
- [Jon08] JONES, Thouis R.; KANG, In Han; WHEELER, Douglas B.; LINDQUIST, Robert A.; PAPALLO, Adam; SABATINI, David M.; GOLAND, Polina und CARPENTER, Anne E.: „CellProfiler Analyst: Data Exploration and Analysis Software for Complex Image-Based Screens“. In: *BMC Bioinformatics* 9.1 (2008), S. 482. DOI: [10.1186/1471-2105-9-482](https://doi.org/10.1186/1471-2105-9-482) (siehe S. 23).
- [Jor15] JORDAN, Michael I. und MITCHELL, Tom M.: „Machine Learning: Trends, Perspectives, and Prospects“. In: *Science* 349.6245 (2015), S. 255–260. DOI: [10.1126/science.aaa841](https://doi.org/10.1126/science.aaa841) (siehe S. 9).
- [Jug14] JUG, Florian; PRIETZSCH, Tobias; PREIBISCH, Stephan und TOMANCAK, Pavel: „Bioimage Informatics in the Context of Drosophila Research“. In: *Methods* 68.1 (2014), S. 60–73. DOI: [10.1016/j.ymeth.2014.04.004](https://doi.org/10.1016/j.ymeth.2014.04.004) (siehe S. 37).

- [Kar20] KARIMI, Davood; DOU, Haoran; WARFIELD, Simon K. und GHOLIPOUR, Ali: „Deep Learning with Noisy Labels: Exploring Techniques and Remedies in Medical Image Analysis“. In: *Medical Image Analysis* 65 (2020), S. 101759. DOI: [10.1016/j.media.2020.101759](https://doi.org/10.1016/j.media.2020.101759) (siehe S. 17, 18).
- [Ken17] KENDALL, Alex und GAL, Yarin: „What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?“ In: *Advances in Neural Information Processing Systems*. Long Beach, CA, USA, 2017, S. 5580–5590. DOI: [10.5555/3295222.3295309](https://doi.org/10.5555/3295222.3295309) (siehe S. 12, 13, 31).
- [Kha22] KHAN, Salman; NASEER, Muzammal; HAYAT, Munawar; ZAMIR, Syed Waqas; KHAN, Fahad Shahbaz und SHAH, Mubarak: „Transformers in Vision: A Survey“. In: *ACM Computing Surveys* 54.10 (2022). DOI: [10.1145/3505244](https://doi.org/10.1145/3505244) (siehe S. 27).
- [Kin07] KING, Gary: „An Introduction to the Dataverse Network as an Infrastructure for Data Sharing“. In: *Sociological Methods and Research* 36 (2007), S. 173–199. DOI: [10.1177/0049124107306660](https://doi.org/10.1177/0049124107306660) (siehe S. 28).
- [Kin15] KINGMA, Diederik P. und BA, Jimmy: „Adam: A Method for Stochastic Optimization“. In: *International Conference on Learning Representations*. Verfügbar: <https://arxiv.org/pdf/1412.6980>. Zugriff: 12.04.2023. San Diego, CA, USA, 2015, S. 1–15 (siehe S. 11, 255).
- [Kir17] KIRKPATRICK, James u. a.: „Overcoming Catastrophic Forgetting in Neural Networks“. In: *Proceedings of the National Academy of Sciences* 114.13 (2017), S. 3521–3526. DOI: [10.1073/pnas.1611835114](https://doi.org/10.1073/pnas.1611835114) (siehe S. 38).
- [Kir19] KIRILLOV, Alexander; HE, Kaiming; GIRSHICK, Ross; ROTHER, Carsten und DOLLAR, Piotr: „Panoptic Segmentation“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, USA, 2019, S. 9404–9413. DOI: [10.1109/CVPR.2019.00963](https://doi.org/10.1109/CVPR.2019.00963) (siehe S. 15).

- [Kli20] KLIMAJ, Stefan Donovan; LICON MUNOZ, Yamhilette; DEL TORO, Katelyn und HINES, William Curtis: „A High-Throughput Imaging and Quantification Pipeline for the EVOS Imaging Platform“. In: *PLoS ONE* 15.8 (2020), S. 1–13. DOI: [10.1371/journal.pone.0236397](https://doi.org/10.1371/journal.pone.0236397) (siehe S. 22, 23, 28, 130).
- [Koh18] KOHL, Simon; ROMERA-PAREDES, Bernardino; MEYER, Clemens; DE FAUW, Jeffrey; LEDSAM, Joseph R.; MAIER-HEIN, Klaus; ESLAMI, S. M. Ali; JIMENEZ REZENDE, Danilo und RONNEBERGER, Olaf: „A Probabilistic U-Net for Segmentation of Ambiguous Images“. In: *Advances in Neural Information Processing Systems*. Montreal, Kanada, 2018, S. 6965–6975. DOI: [10.5555/3327757.3327800](https://doi.org/10.5555/3327757.3327800) (siehe S. 20).
- [Köh19] KÖHLER, Jan M.; AUTENRIETH, Maximilian und BELUCH, William H.: „Uncertainty Based Detection and Relabeling of Noisy Image Labels“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*. Verfügbar: <https://arxiv.org/pdf/1906.11876>. Zugriff: 11.05.2023. Long Beach, CA, USA, 2019, S. 33–37 (siehe S. 20).
- [Kom21] KOMARNICKA, Urszula K.; PUCELIK, Barbara; WOJTALA, Daria; LESIÓW, Monika K.; STOCHEL, Grażyna und KYZIOŁ, Agnieszka: „Evaluation of Anticancer Activity in Vitro of a Stable Copper(I) Complex with Phosphine-Peptide Conjugate“. In: *Scientific Reports* 11.1 (2021), S. 23943. DOI: [10.1038/s41598-021-03352-2](https://doi.org/10.1038/s41598-021-03352-2) (siehe S. 78).
- [Kou18] KOUW, Wouter M. und LOOG, MARCO: An Introduction to Domain Adaptation and Transfer Learning. arXiv:1812.11806, [Online]. Verfügbar: <https://arxiv.org/pdf/1812.11806>. Zugriff: 12.04.2023. 2018 (siehe S. 35).
- [Kov22] KOVAL, Alina: How to Use Pre-Labeled Data for AI Algorithms With High-Quality Requirements. Datafloq. [Online]. Verfügbar: <https://datafloq.com/read/use-pre-labeled-data-ai-algorithms-high-quality-requirements>. Zugriff: 17.01.2023. 2022 (siehe S. 18).

- [Kri12] KRIZHEVSKY, Alex; SUTSKEVER, Ilya und HINTON, Geoffrey E: „ImageNet Classification with Deep Convolutional Neural Networks“. In: *Advances in Neural Information Processing Systems*. Bd. 25. Lake Tahoe, NV, USA, 2012. DOI: [10.5555/2999134.2999257](https://doi.org/10.5555/2999134.2999257) (siehe S. 10, 25).
- [Krü22] KRÜGER, Martin: „Konzeption eines Mechanismus zum kontaktlosen Übertragen von Tröpfchen für Droplet Microarrays“. Bachelorarbeit. Karlsruhe, Deutschland: Karlsruher Institut für Technologie, Fakultät für Maschinenbau, 2022 (siehe S. 155, 157, 158).
- [Kum20] KUMAR, Neeraj u. a.: „A Multi-Organ Nucleus Segmentation Challenge“. In: *IEEE Transactions on Medical Imaging* 39.5 (2020), S. 1380–1391. DOI: [10.1109/TMI.2019.2947628](https://doi.org/10.1109/TMI.2019.2947628) (siehe S. 48, 99, 101, 251).
- [Lai17] LAINE, Samuli und AILA, Timo: „Temporal Ensembling for Semi-Supervised Learning“. In: *International Conference on Learning Representations*. Verfügbar: <https://openreview.net/forum?id=BJ6oOfqge>. Zugriff: 12.04.2023. Toulon, Frankreich, 2017 (siehe S. 34).
- [Lak17] LAKSHMINARAYANAN, Balaji; PRITZEL, Alexander und BLUNDELL, Charles: „Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles“. In: *Advances in Neural Information Processing Systems*. Long Beach, CA, USA, 2017, S. 6405–6416. DOI: [10.5555/3295222.3295387](https://doi.org/10.5555/3295222.3295387) (siehe S. 13).
- [Lak21] LAKSHMANAN, V.; GÖRNER, M. und GILLARD, R.: *Practical Machine Learning for Computer Vision: End-to-end Machine Learning for Images*. Sebastopol, CA, USA: O’Reilly Media, 2021 (siehe S. 9, 10).
- [Law79] LAWS, Kenneth Ivan: „Texture Energy Measures“. In: *Image Understanding Workshop*. Verfügbar: <https://apps.dtic.mil/sti/pdfs/ADA077568.pdf#page=57>. Zugriff: 12.04.2023. Los Angeles, CA, USA, 1979, S. 47–51 (siehe S. 25).

- [LeC15] LECUN, Yann; BENGIO, Yoshua und HINTON, Geoffrey: „Deep Learning“. In: *Nature* 521.7553 (2015), S. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539) (siehe S. 10, 26).
- [LeC98] LECUN, Yann; BOTTOU, Leon; BENGIO, Yoshua und HAFFNER, Patrick: „Gradient-Based Learning Applied to Document Recognition“. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791) (siehe S. 25).
- [Lee13] LEE, Dong-Hyun: „Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks“. In: *International Conference on Learning Representations*. Verfügbar: <https://scholar.google.co.kr/scholar?oi=bibs&cluster=16547318329102522555&btnI=1&hl=ko>. Zugriff: 12.04.2023. Scottsdale, AZ, USA, 2013 (siehe S. 35).
- [Lee18] LEE, Kuang-Huei; HE, Xiaodong; ZHANG, Lei und YANG, Linjun: „CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA, 2018, S. 5447–5456. DOI: [10.1109/CVPR.2018.00571](https://doi.org/10.1109/CVPR.2018.00571) (siehe S. 19).
- [Lei20] LEI, Wenxi; DEMIR, Konstantin; OVERHAGE, Joerg; GRUNZE, Michael; SCHWARTZ, Thomas und LEVKIN, Pavel A.: „Droplet-Microarray: Miniaturized Platform for High-Throughput Screening of Antimicrobial Compounds“. In: *Advanced Biosystems* 4.10 (2020), S. 2000073. DOI: [10.1002/adbi.202000073](https://doi.org/10.1002/adbi.202000073) (siehe S. 1).
- [Lim16] LIM, Ji Won; SHIN, Kwang Soo; MOON, Jaemin; LEE, Sung Kuk und KIM, Taesung: „A Microfluidic Platform for High-Throughput Screening of Small Mutant Libraries“. In: *Analytical Chemistry* 88.10 (2016), S. 5234–5242. DOI: [10.1021/acs.analchem.6b00317](https://doi.org/10.1021/acs.analchem.6b00317) (siehe S. 23).
- [Lin15] LIN, Tzuta: LabelImg. [Online]. Verfügbar: <https://github.com/tzutalin/labelimg>. Zugriff: 18.05.2022. 2015 (siehe S. 21, 115).
- [Lin98] LINDBERG, Tony: „Feature Detection with Automatic Scale Selection“. In: *International Journal of Computer Vision* 30.2 (1998), S. 79–116. DOI: [10.1023/A:1008045108935](https://doi.org/10.1023/A:1008045108935) (siehe S. 25, 76).

- [Lit17] LITJENS, Geert; KOOI, Thijs; BEJNORDI, Babak Ehteshami; SETIO, Arnaud Arindra Adiyoso; CIOMPI, Francesco; GHAFORIAN, Mohsen; DER LAAK, Jeroen A.W.M. van; VAN GINNEKEN, Bram und SÁNCHEZ, Clara I.: „A Survey on Deep Learning in Medical Image Analysis“. In: *Medical Image Analysis* 42 (2017), S. 60–88. DOI: [10.1016/j.media.2017.07.005](https://doi.org/10.1016/j.media.2017.07.005) (siehe S. 2, 15, 36).
- [Liu20] LIU, Yang und GUO, Hongyi: „Peer loss functions: Learning from noisy labels without knowing noise rates“. In: *International Conference on Machine Learning*. Verfügbar: <http://proceedings.mlr.press/v119/liu20e/liu20e.pdf>. Zugriff: 12.04.2023. Virtuelles Event, 2020, S. 6226–6236 (siehe S. 19).
- [Liu22] LIU, Ziquan; XU, Yi; XU, Yuanhong; QIAN, Qi; LI, Hao; CHAN, Antoni B. und JIN, Rong: „Improved Fine-Tuning by Leveraging Pre-Training Data: Theory and Practice“. In: *Advances in Neural Information Processing Systems*. Bd. 35. Verfügbar: https://proceedings.neurips.cc/paper_files/paper/2022/file/d1c88f9790765146ec8fb5d02e5653a0-Paper-Conference.pdf. Zugriff: 12.04.2023. New Orleans, LA, USA, 2022, S. 32568–32581 (siehe S. 39, 83).
- [Ljo12] LJOSA, Vebjorn; SOKOLNICKI, Katherine L und CARPENTER, Anne E: „Annotated High-Throughput Microscopy Image Sets for Validation“. In: *Nature Methods* 9.7 (2012), S. 637–637. DOI: [10.1038/nmeth.2083](https://doi.org/10.1038/nmeth.2083) (siehe S. 107, 108).
- [Löf21] LÖFFLER, Katharina; SCHERR, Tim und MIKUT, Ralf: „A Graph-Based Cell Tracking Algorithm with Few Manually Tunable Parameters and Automated Segmentation Error Correction“. In: *PLOS ONE* 16.9 (2021), S. 1–28. DOI: [10.1371/journal.pone.0249257](https://doi.org/10.1371/journal.pone.0249257) (siehe S. 18, 62).
- [Löf23] LÖFFLER, Katharina: „Multiple Object Tracking in Light Microscopy Images Using Graph-based and Deep Learning Methods“. Dissertation. Karlsruhe, Deutschland: Karlsruher Institut für Technologie, Fakultät für Maschinenbau, 2023. DOI: [10.5445/IR/1000158917](https://doi.org/10.5445/IR/1000158917) (siehe S. 18, 62).

- [Luo22] LUO, Qiwu; SU, Jiaojiao; YANG, Chunhua; GUI, Weihua; SILVÉN, Olli und LIU, Li: „CAT-EDNet: Cross-attention Transformer-Based Encoder–Decoder Network for Salient Defect Detection of Strip Steel Surface“. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), S. 1–13. DOI: [10.1109/TIM.2022.3165270](https://doi.org/10.1109/TIM.2022.3165270) (siehe S. 2).
- [Mac67] MACQUEEN, John B.: „Some Methods for Classification and Analysis of MultiVariate Observations“. In: *Berkeley Symposium on Mathematical Statistics and Probability*. Verfügbar: <https://projecteuclid.org/ebook/Download?urlid=bsmsp%2F1200512992&isFullBook=False>. Zugriff: 12.04.2023. Berkeley, CA, USA, 1967, S. 281–297 (siehe S. 25).
- [Mah19] MAHONY, Niall O’; CAMPBELL, Sean; CARVALHO, Anderson; HARAPANAHALLI, Suman; VELASCO-HERNÁNDEZ, Gustavo Adolfo; KRPALKOVA, Lenka; RIORDAN, Daniel und WALSH, Joseph: „Deep Learning vs. Traditional Computer Vision“. In: *Advances in Computer Vision*. Las Vegas, NV, USA, 2019, S. 128–144. DOI: [10.1007/978-3-030-17795-9_10](https://doi.org/10.1007/978-3-030-17795-9_10) (siehe S. 2, 15, 22, 29, 31).
- [Mah21] MAHBOD, Amirreza; SCHAEFER, Gerald; BANCHER, Benjamin; LÖW, Christine; DORFFNER, Georg; ECKER, Rupert und ELLINGER, Isabella: „CryoNuSeg: A Dataset for Nuclei Instance Segmentation of Cryosectioned H&E-stained Histological Images“. In: *Computers in Biology and Medicine* 132 (2021), S. 104349. DOI: [10.1016/j.combiomed.2021.104349](https://doi.org/10.1016/j.combiomed.2021.104349) (siehe S. 246, 247).
- [Mai23] MAIER-HEIN, Lena u. a.: Metrics Reloaded: Pitfalls and Recommendations for Image Analysis Validation. arXiv:2206.01653, [Online]. Verfügbar: <https://arxiv.org/pdf/2206.01653>. Zugriff: 25.05.2023. 2023 (siehe S. 248, 250).
- [Mak06] MAKARENKOV, Vladimir; KEVORKOV, Dmytro; ZENTILLI, Pablo; GAGARIN, Andrei; MALO, Nathalie und NADON, Robert: „HTS-Corrector: Software for the Statistical Analysis and Correction of

- Experimental High-Throughput Screening Data“. In: *Bioinformatics* 22.11 (2006), S. 1408–1409. DOI: [10.1093/bioinformatics/btl126](https://doi.org/10.1093/bioinformatics/btl126) (siehe S. 28, 130).
- [McI18] McINNES, Leland; HEALY, John; SAUL, Nathaniel und GROßBERGER, Lukas: „UMAP: Uniform Manifold Approximation and Projection“. In: *Journal of Open Source Software* 3.29 (2018), S. 861. DOI: [10.21105/joss.00861](https://doi.org/10.21105/joss.00861) (siehe S. 55).
- [Mik08] MIKUT, Ralf: *Data Mining in der Medizin und Medizintechnik*. Karlsruhe, Deutschland: KIT Scientific Publishing, 2008. DOI: [10.5445/KSP/1000008476](https://doi.org/10.5445/KSP/1000008476) (siehe S. 21, 28, 248, 249).
- [Mir19] MIRIKHARAJI, Zahra; YAN, Yiqi und HAMARNEH, Ghassan: „Learning to Segment Skin Lesions from Noisy Annotations“. In: *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. Shenzhen, China, 2019, S. 207–215. DOI: [10.1007/978-3-030-33391-1_24](https://doi.org/10.1007/978-3-030-33391-1_24) (siehe S. 19).
- [Mit97] MITCHELL, Tom M.: *Machine Learning*. New York City, USA: McGraw-Hill, 1997 (siehe S. 9, 10).
- [Moe19] MOEN, Erick; BANNON, Dylan; KUDO, Takamasa; GRAF, William; COVERT, Markus und VAN VALEN, David: „Deep Learning for Cellular Image Analysis“. In: *Nature Methods* 16.12 (2019), S. 1233–1246. DOI: [10.1038/s41592-019-0403-1](https://doi.org/10.1038/s41592-019-0403-1) (siehe S. 10, 36).
- [Moh12] MOHRI, Mehryar; ROSTAMIZADEH, Afshin und TALWALKAR, Ameet: *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012 (siehe S. 9, 10, 32, 249).
- [Mos18] MOSAAD, E. O.; CHAMBERS, K. F.; FUTREGA, K.; CLEMENTS, J. A. und DORAN, M. R.: „The Microwell-mesh: A High-Throughput 3D Prostate Cancer Spheroid and Drug-Testing Platform“. In: *Scientific Reports* 8.1 (2018), S. 253. DOI: [10.1038/s41598-017-18050-1](https://doi.org/10.1038/s41598-017-18050-1) (siehe S. 22).

- [Nai20] NAIR, Tanya; PRECUP, Doina; ARNOLD, Douglas L. und ARBEL, Tal: „Exploring Uncertainty Measures in Deep Networks for Multiple Sclerosis Lesion Detection and Segmentation“. In: *Medical Image Analysis* 59 (2020), S. 101557. DOI: [10.1016/j.media.2019.101557](https://doi.org/10.1016/j.media.2019.101557) (siehe S. 14).
- [Ng23] NG, Andrew: Data-centric AI Resource Hub. [Online]. Verfügbar: <https://datacentricai.org/>. Zugriff: 21.02.2023. 2023 (siehe S. 2).
- [Nor17] NORTHCUTT, Curtis G.; WU, Tailin und CHUANG, Isaac L.: „Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels“. In: *Conference on Uncertainty in Artificial Intelligence*. Verfügbar: <https://arxiv.org/pdf/1705.01936>. Zugriff: 11.05.2023. Sydney, Australien, 2017 (siehe S. 19).
- [Nor21a] NORTHCUTT, Curtis G.; ATHALYE, Anish und MUELLER, Jonas: „Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks“. In: *Advances in Neural Information Processing Systems, Track on Datasets and Benchmarks*. Verfügbar: <https://openreview.net/pdf?id=XccDXrDNLeK>. Zugriff: 12.04.2023. Virtuelles Event, 2021 (siehe S. 18).
- [Nor21b] NORTHCUTT, Curtis G.; JIANG, Lu und CHUANG, Isaac L.: „Confident Learning: Estimating Uncertainty in Dataset Labels“. In: *Journal of Artificial Intelligence Research (JAIR)* 70 (2021), S. 1373–1411. DOI: [10.1613/jair.1.12125](https://doi.org/10.1613/jair.1.12125) (siehe S. 19).
- [Noy20] NOY, Natasha: An Analysis of Online Datasets Using Dataset Search. Google Research Blog. [Online]. Verfügbar: <https://ai.googleblog.com/2020/08/an-analysis-of-online-datasets-using.html>. Zugriff: 16.01.2023. 2020 (siehe S. 36, 83).
- [Ots79] OTSU, Nobuyuki: „A Threshold Selection Method from Gray-Level Histograms“. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), S. 62–66. DOI: [10.1109/TSMC.1979.4310076](https://doi.org/10.1109/TSMC.1979.4310076) (siehe S. 16, 25).

- [Oua20] OUALI, Yassine; HUDELLOT, Céline und TAMI, Myriam: An Overview of Deep Semi-Supervised Learning. arXiv:2006.05278, [Online]. Verfügbar: <https://arxiv.org/pdf/2006.05278>. Zugriff: 12.04.2023. 2020 (siehe S. 33).
- [Ouy19] OUYANG, Wei; MUELLER, Florian; HJELMARE, Martin; LUNDBERG, Emma und ZIMMER, Christophe: „ImJoy: An Open-Source Computational Platform for the Deep Learning Era“. In: *Nature Methods* 16.12 (2019), S. 1199–1200. DOI: [10.1038/s41592-019-0627-0](https://doi.org/10.1038/s41592-019-0627-0) (siehe S. 28).
- [Ouy22] OUYANG, Wei u. a.: BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis. bioRxiv: 2022.06.07.495102, [Online]. Verfügbar: <https://www.biorxiv.org/content/10.1101/2022.06.07.495102v1>. Zugriff: 21.02.2023. 2022 (siehe S. 28).
- [Pan10] PAN, Sinno Jialin und YANG, Qiang: „A Survey on Transfer Learning“. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), S. 1345–1359. DOI: [10.1109/TKDE.2009.191](https://doi.org/10.1109/TKDE.2009.191) (siehe S. 35).
- [Pas19] PASZKE, Adam u. a.: „PyTorch: An Imperative Style, High-Performance Deep Learning Library“. In: *Advances in Neural Information Processing Systems*. Bd. 32. Vancouver, Kanada, 2019. DOI: [10.5555/3454287.3455008](https://doi.org/10.5555/3454287.3455008) (siehe S. 91).
- [Pat17] PATRINI, Giorgio; ROZZA, Alessandro; MENON, Aditya Krishna; NOCK, Richard und QU, Lizhen: „Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA, 2017, S. 2233–2241. DOI: [10.1109/CVPR.2017.240](https://doi.org/10.1109/CVPR.2017.240) (siehe S. 19).
- [Pen21] PENG, Jialin und WANG, Ye: Medical Image Segmentation with Limited Supervision: A Review of Deep Network Models. arXiv:2103.00429, [Online]. Verfügbar: <https://arxiv.org/pdf/2103.00429>. Zugriff: 21.02.2023. 2021 (siehe S. 29, 30, 34).

- [Pet21] PETROV, Dmitry und SHCHEKLEIN, Ivan: Data Version Control - Open-source Version Control System for Machine Learning Projects. [Online]. Verfügbar: <https://dvc.org/>. Zugriff: 21.10.2021. 2021 (siehe S. 60, 91).
- [Pha21] PHAM, Hieu; DAI, Zihang; XIE, Qizhe und LE, Quoc V.: „Meta Pseudo Labels“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Nashville, TN, USA, 2021, S. 11552–11563. DOI: [10.1109/CVPR46437.2021.01139](https://doi.org/10.1109/CVPR46437.2021.01139) (siehe S. 35).
- [Pic23] PICHAI, Sundar: An important next step on our AI journey. Google Blog. [Online]. Verfügbar: <https://blog.google/technology/ai/bard-google-ai-search-updates/>. Zugriff: 03.04.2023. 2023 (siehe S. 36).
- [Pop16] POPOVA, Anna A.; DEMIR, Konstantin; HARTANTO, Titus Genisius; SCHMITT, Eric und LEVKIN, Pavel A.: „Droplet-Microarray on Superhydrophobic–Superhydrophilic Patterns for High-Throughput Live Cell Screenings“. In: *RSC Advances* 6.44 (2016), S. 38263–38276. DOI: [10.1039/C6RA06011K](https://doi.org/10.1039/C6RA06011K) (siehe S. 1, 22).
- [Pop19] POPOVA, Anna A.; TRONSER, Tina; DEMIR, Konstantin; HAITZ, P.; KUODYTE, Karolina; STARKUVIENE, Vytaute; WAJDA, Piotr und LEVKIN, Pavel A.: „Facile One Step Formation and Screening of Tumor Spheroids Using Droplet-Microarray Platform“. In: *Small* 15.25 (2019). DOI: [10.1002/smll.201901299](https://doi.org/10.1002/smll.201901299) (siehe S. 1, 24).
- [Räd23] RÄDSCH, Tim; REINKE, Annika; WERU, Vivienn; TIZABI, Minu D.; SCHRECK, Nicholas; KAVUR, A. Emre; PEKDEMIR, Bünyamin; ROß, Tobias; KOPP-SCHNEIDER, Annette und MAIER-HEIN, Lena: „Labelling Instructions Matter in Biomedical Image Analysis“. In: *Nature Machine Intelligence* 5.3 (2023), S. 273–283. DOI: [10.1038/s42256-023-00625-5](https://doi.org/10.1038/s42256-023-00625-5) (siehe S. 18).
- [Rei06] REISCHL, Markus: „Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen“. Diss. Karlsruhe, Deutschland, 2006. DOI: [10.5445/KSP/1000004247](https://doi.org/10.5445/KSP/1000004247) (siehe S. 28).

- [Rei17] REISCHL, Markus: „Eine umfassende Methodik zur automatisierten Auswertung von bildgebenden Hochdurchsatzverfahren“. Habilitation. Karlsruhe, Deutschland: Karlsruher Institut für Technologie (KIT), 2017 (siehe S. 1, 22, 23, 25, 52).
- [Ren21] REN, Pengzhen; XIAO, Yun; CHANG, Xiaojun; HUANG, Po-Yao; LI, Zhihui; GUPTA, Brij B.; CHEN, Xiaojiang und WANG, Xin: „A Survey of Deep Active Learning“. In: *ACM Computing Surveys* 54.9 (2021). DOI: [10.1145/3472291](https://doi.org/10.1145/3472291) (siehe S. 16, 30, 31).
- [Riv22] RIVERBANK COMPUTING LIMITED, THE QT COMPANY: PyQt5 Reference Guide. [Online]. Verfügbar: <https://www.riverbankcomputing.com/static/Docs/PyQt5/index.html>. Zugriff: 19.09.2022. 2022 (siehe S. 92, 95, 97).
- [Rob51] ROBBINS, Herbert E.: „A Stochastic Approximation Method“. In: *Annals of Mathematical Statistics* 22 (1951), S. 400–407. DOI: [10.1214/aoms/1177729586](https://doi.org/10.1214/aoms/1177729586) (siehe S. 11).
- [Rol17] ROLNICK, David; VEIT, Andreas; BELONGIE, Serge J. und SHAVIT, Nir: Deep Learning Is Robust to Massive Label Noise. arXiv:1705.10694, [Online]. Verfügbar: <https://arxiv.org/pdf/1705.10694>. Zugriff: 12.04.2023. 2017 (siehe S. 18).
- [Ron15] RONNEBERGER, O.; P.FISCHER und BROX, T.: „U-Net: Convolutional Networks for Biomedical Image Segmentation“. In: *Medical Image Computing and Computer-Assisted Intervention*. Bd. 9351. München, Deutschland, 2015, S. 234–241 (siehe S. 26, 81, 108, 109, 254).
- [Rou22] ROUILLARD, Tristan; PROSKUDIN, Kostiantyn und WENNMAN, Alexander: Hasty.ai. [Online]. Verfügbar: <https://hasty.ai>. Zugriff: 18.05.2022. 2022 (siehe S. 16, 21, 115).
- [Roy19] ROY, Abhijit Guha; CONJETI, Sailesh; NAVAB, Nassir und WACHINGER, Christian: „Bayesian QuickNAT: Model Uncertainty in Deep Whole-Brain Segmentation for Structure-Wise Quality Control“. In: *NeuroImage* 195 (2019), S. 11–22. DOI: [10.1016/j.neuroimage.2019.03.042](https://doi.org/10.1016/j.neuroimage.2019.03.042) (siehe S. 14, 238, 252).

- [Sai21] SAIDU, Isah Charles und CSATÓ, Lehel: „Active Learning with Bayesian UNet for Efficient Semantic Image Segmentation“. In: *Journal of Imaging* 7.2 (2021), S. 37. DOI: [10.3390/jimaging7020037](https://doi.org/10.3390/jimaging7020037) (siehe S. 31, 84).
- [Sam88] SAMET, Hanan und TAMMINEN, Markku: „Efficient Component Labeling of Images of Arbitrary Dimension Represented by Linear Bintrees“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.4 (1988), S. 579–586. DOI: [10.1109/34.3918](https://doi.org/10.1109/34.3918) (siehe S. 25).
- [Sch03] SCHULTZ, Matthew und JOACHIMS, Thorsten: „Learning a Distance Metric from Relative Comparisons“. In: *Advances in Neural Information Processing Systems*. Bd. 16. Whistler, Kanada, 2003, S. 41–48. DOI: [10.5555/2981345.2981351](https://doi.org/10.5555/2981345.2981351) (siehe S. 33).
- [Sch12] SCHINDELIN, Johannes; ARGANDA-CARRERAS, Ignacio; FRISE, Erwin; KAYNIG, Verena; LONGAIR, Mark; PIETZSCH, Tobias; PREIBISCH, Stephan; RUEDEN, Curtis; SAALFELD, Stephan; SCHMID, Benjamin u. a.: „Fiji: An Open-Source Platform for Biological-Image Analysis“. In: *Nature Methods* 9.7 (2012), S. 676–682. DOI: [10.1038/nmeth.2019](https://doi.org/10.1038/nmeth.2019) (siehe S. 16, 23, 28, 130).
- [Sch15] SCHMIDHUBER, Jürgen: „Deep Learning in Neural Networks: An Overview“. In: *Neural Networks* 61 (2015), S. 85–117. DOI: [10.1016/j.neunet.2014.09.003](https://doi.org/10.1016/j.neunet.2014.09.003) (siehe S. 10).
- [Sch19] SCHUTERA, Mark; JUST, Steffen; GIERTEN, Jakob; MIKUT, Ralf; REISCHL, Markus und PYLATIUK, Christian: „Machine Learning Methods for Automated Quantification of Ventricular Dimensions“. In: *Zebrafish* 16.6 (2019), S. 542–545. DOI: [10.1089/zeb.2019.1754](https://doi.org/10.1089/zeb.2019.1754) (siehe S. 24).
- [Sch20a] SCHERR, Tim; LÖFFLER, Katharina; BÖHLAND, Moritz und MIKUT, Ralf: „Cell Segmentation and Tracking Using CNN-Based Distance Predictions and a Graph-Based Matching Strategy“. In: *PLoS One* 15.12 (2020), S. 1–22. DOI: [10.1371/journal.pone.0243219](https://doi.org/10.1371/journal.pone.0243219) (siehe S. 19, 27, 257).

- [Sch20b] SCHERR, Tim; STREULE, Karolin; BARTSCHAT, Andreas; BÖHLAND, Moritz; STEGMAIER, Johannes; REISCHL, Markus; ORIAN-ROUSSEAU, Véronique und MIKUT, Ralf: „BeadNet: Deep Learning-Based Bead Detection and Counting in Low-Resolution Microscopy Images“. In: *Bioinformatics* 36.17 (2020), S. 4668–4670. DOI: [10.1093/bioinformatics/btaa594](https://doi.org/10.1093/bioinformatics/btaa594) (siehe S. 16, 26, 76, 132).
- [Sch21d] SCHUETZKE, Jan; BENEDIX, Alexander; MIKUT, Ralf und REISCHL, Markus: „Enhancing deep-learning training for phase identification in powder X-ray diffractograms“. In: *International Union of Crystallography Journal* 8.3 (2021), S. 408–420. DOI: [10.5445/IR/1000131192](https://doi.org/10.5445/IR/1000131192) (siehe S. 161).
- [Sch21e] SCHUTERA, Mark; HUSSEIN, Mostafa; ABHAU, Jochen; MIKUT, Ralf und REISCHL, Markus: „Night-to-Day: Online Image-to-Image Translation for Object Detection Within Autonomous Driving by Night“. In: *IEEE Transactions on Intelligent Vehicles* 6.3 (2021), S. 480–489. DOI: [10.1109/TIV.2020.3039456](https://doi.org/10.1109/TIV.2020.3039456) (siehe S. 36, 38).
- [Sch22f] SCHULMAN, John u. a.: ChatGPT: Optimizing Language Models for Dialogue. [Online]. Verfügbar: <https://chat.openai.com/chat>. Zugriff: 05.01.2023, 2022 (siehe S. 36, 37).
- [Sch22g] SCHUTERA, Mark; RETTENBERGER, Luca; PYLATIUK, Christian und REISCHL, Markus: „Methods for the Frugal Labeler: Multi-class Semantic Segmentation on Heterogeneous Labels“. In: *PLOS ONE* 17.2 (2022), S. 1–14. DOI: [10.1371/journal.pone.0263656](https://doi.org/10.1371/journal.pone.0263656) (siehe S. 24).
- [Sch23a] SCHERR, Tim: „Deep learning-based particle detection and instance segmentation for microscopy images“. Diss. Karlsruhe, Deutschland: Karlsruher Institut für Technologie (KIT), 2023. DOI: [10.5445/IR/1000163187](https://doi.org/10.5445/IR/1000163187) (siehe S. 16, 27).
- [Sch23e] SCHUTERA, Mark: „A Concept for Deployment and Evaluation of Unsupervised Domain Adaptation in Cognitive Perception Systems“. Diss. Karlsruhe, Deutschland: Karlsruher Institut für Technologie (KIT), 2023. DOI: [10.5445/IR/1000162895](https://doi.org/10.5445/IR/1000162895) (siehe S. 35, 38).

- [Sei23] SEIFERMANN, Maximilian; REISER, Patrick; FRIEDERICH, Pascal und LEVKIN, Pavel A.: „High-Throughput Synthesis and Machine Learning Assisted Design of Photodegradable Hydrogels“. In: *Small Methods* (2023), S. 2300553. DOI: [10.1002/smt.202300553](https://doi.org/10.1002/smt.202300553) (siehe S. 23).
- [Sek20] SEKACHEV, Boris; MANOVICH, Nikita; ZHILTSOV, Maxim; ZHAVORONKOV, Andrey; KALININ, Dmitry und HOFF, Ben: Computer Vision Annotation Tool (CVAT). [Online]. Verfügbar: <https://github.com/openvinotoolkit/cvat>. Zugriff: 28.04.2022. 2020 (siehe S. 16, 21, 115).
- [Sen18] SENNER, Ozan und SAVARESE, Silvio: „Active Learning for Convolutional Neural Networks: A Core-Set Approach“. In: *International Conference on Learning Representations*. Verfügbar: <https://openreview.net/forum?id=H1aIuk-RW>. Zugriff: 12.04.2023. Vancouver, Kanada, 2018 (siehe S. 31, 55, 84, 104).
- [Set10] SETTLES, Burr: „Active Learning Literature Survey“. In: *Computer Sciences Technical Report, University of Wisconsin-Madison* (2010). Verfügbar: <https://burrsettles.com/pub/settles.activelearning.pdf>. Zugriff: 12.04.2023 (siehe S. 30).
- [Sha21] SHARMA, Manu; RASMUSON, Daniel und RIEGER, Brian: Labelbox. [Online]. Verfügbar: <https://labelbox.com>. Zugriff: 28.05.2021. 2021 (siehe S. 16).
- [Sin06] SINGH, R.; VELASQUEZ, E.; VIJAYANT, P. und YERA, E.: „Free-FlowDB: Storage, Querying and Interacting with Structure-Activity Information from High-Throughput Drug Discovery“. In: *19th IEEE Symposium on Computer-Based Medical Systems*. Salt Lake City, UT, USA, 2006, S. 75–80. DOI: [10.1109/CBMS.2006.89](https://doi.org/10.1109/CBMS.2006.89) (siehe S. 28).
- [Sir17] SIRINUKUNWATTANA, Korsuk u. a.: „Gland Segmentation in Colon Histology Images: The Glas Challenge Contest“. In: *Medical Image Analysis* 35 (2017), S. 489–502. DOI: [10.1016/j.media.2016.08.008](https://doi.org/10.1016/j.media.2016.08.008) (siehe S. 37).

- [Smi18] SMITH, Lewis und GAL, Yarin: „Understanding Measures of Uncertainty for Adversarial Example Detection“. In: *Conference on Uncertainty in Artificial Intelligence*. Verfügbar: <http://auai.org/uai2018/proceedings/papers/207.pdf>. Zugriff: 12.04.2023. Monterey, CA, USA, 2018, S. 560–569 (siehe S. 14).
- [Soh20] SOHN, Kihyuk; BERTHELOT, David; CARLINI, Nicholas; ZHANG, Zizhao; ZHANG, Han; RAFFEL, Colin A; CUBUK, Ekin Dogus; KURAKIN, Alexey und LI, Chun-Liang: „FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence“. In: *Advances in Neural Information Processing Systems*. Bd. 33. Virtuelles Event, 2020, S. 596–608. DOI: [10.5555/3495724.3495775](https://doi.org/10.5555/3495724.3495775) (siehe S. 35, 85).
- [Son20] SONG, Hwanjun; KIM, Minseok; PARK, Dongmin und LEE, Jae-Gil: Learning from Noisy Labels with Deep Neural Networks: A Survey. arXiv:2007.08199, [Online]. Verfügbar: <https://arxiv.org/pdf/2007.08199>. Zugriff: 12.04.2023. 2020 (siehe S. 18).
- [Spa16] SPANHOL, Fabio A.; OLIVEIRA, Luiz S.; PETITJEAN, Caroline und HEUTTE, Laurent: „A Dataset for Breast Cancer Histopathological Image Classification“. In: *IEEE Transactions on Biomedical Engineering* 63.7 (2016), S. 1455–1462. DOI: [10.1109/TBME.2015.2496264](https://doi.org/10.1109/TBME.2015.2496264) (siehe S. 37).
- [Sri14] SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya und SALAKHUTDINOV, Ruslan: „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“. In: *Journal of Machine Learning Research* 15.56 (2014), S. 1929–1958. DOI: [10.5555/2627435.2670313](https://doi.org/10.5555/2627435.2670313) (siehe S. 13).
- [Ste14] STEGMAIER, Johannes; OTTE, Jens C.; KOBITSKI, Andrei; BARTSCHAT, Andreas; GARCIA, Ariel; NIENHAUS, G. Ulrich; STRÄHLE, Uwe und MIKUT, Ralf: „Fast Segmentation of Stained Nuclei in Terabyte-Scale, Time Resolved 3D Microscopy Image Stacks“. In: *PLOS ONE* 9.2 (2014), S. 1–11. DOI: [10.1371/journal.pone.0090036](https://doi.org/10.1371/journal.pone.0090036) (siehe S. 25).

- [Sti21] STIRLING, David R; CARPENTER, Anne E und CIMINI, Beth A: „CellProfiler Analyst 3.0: Accessible Data Exploration and Machine Learning for Image Analysis“. In: *Bioinformatics* 37.21 (2021), S. 3992–3994. DOI: [10.1093/bioinformatics/btab634](https://doi.org/10.1093/bioinformatics/btab634) (siehe S. 23).
- [Str21a] STRAUß, Annika: „Establishment of the workflow for high-throughput drug testing of patient-derived chronic lymphocytic leukemia cells on Droplet Microarray chip“. Bachelorarbeit. Karlsruhe, Deutschland: Karlsruher Institut für Technologie, Fakultät für Biologie, 2021 (siehe S. 155, 157, 158).
- [Str21b] STRINGER, Carsen; WANG, Tim; MICHAELOS, Michalis und PACHITARIU, Marius: „Cellpose: A Generalist Algorithm for Cellular Segmentation“. In: *Nature Methods* 18.1 (2021), S. 100–106. DOI: [10.1038/s41592-020-01018-x](https://doi.org/10.1038/s41592-020-01018-x) (siehe S. 16, 27, 56, 108, 109, 115).
- [Suk15] SUKHBAATAR, Sainbayar; BRUNA, Joan; PALURI, Manohar; BOURDEV, Lubomir und FERGUS, Rob: „Training Convolutional Networks with Noisy Labels“. In: *International Conference on Learning Representations*. Verfügbar: <https://nyuscholars.nyu.edu/en/publications/training-convolutional-networks-with-noisy-labels>. Zugriff: 12.04.2023. San Diego, CA, USA, 2015 (siehe S. 19).
- [Szy12] SZYMAŃSKI, Paweł; MARKOWICZ, Magdalena und MIKICIUK-OLASIK, Elżbieta: „Adaptation of High-Throughput Screening in Drug Discovery—Toxicological Screening Tests“. In: *International Journal of Molecular Sciences* 13.1 (2012), S. 427–452. DOI: [10.3390/ijms13010427](https://doi.org/10.3390/ijms13010427) (siehe S. 1, 22).
- [Tah15] TAHA, Abdel Aziz und HANBURY, Allan: „Metrics for Evaluating 3D Medical Image Segmentation: Analysis, Selection, and Tool“. In: *BMC Medical Imaging* 15.1 (2015), S. 29. DOI: [10.1186/s12880-015-0068-x](https://doi.org/10.1186/s12880-015-0068-x) (siehe S. 48).
- [Taj20] TAJBAKHS, Nima; JEYASEELAN, Laura; LI, Qian; CHIANG, Jeffrey; WU, Zhihao und DING, Xiaowei: „Embracing Imperfect Datasets:

- A Review of Deep Learning Solutions for Medical Image Segmentation“. In: *Medical Image Analysis* 63 (2020), S. 101693. DOI: [10.1016/j.media.2020.101693](https://doi.org/10.1016/j.media.2020.101693) (siehe S. 18).
- [Tan19] TANNO, Ryutaro; SAEEDI, Ardavan; SANKARANARAYANAN, Swami; ALEXANDER, Daniel und SILBERMAN, Nathan: „Learning from Noisy Labels by Regularized Estimation of Annotator Confusion“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, USA, 2019, S. 11236–11245. DOI: [10.1109/CVPR.2019.011150](https://doi.org/10.1109/CVPR.2019.011150) (siehe S. 20).
- [Tar17] TARVAINEN, Antti und VALPOLA, Harri: „Mean Teachers Are Better Role Models: Weight-averaged Consistency Targets Improve Semi-Supervised Deep Learning Results“. In: *Advances in Neural Information Processing Systems*. Bd. 30. Long Beach, CA, USA, 2017. DOI: [10.5555/3294771.3294885](https://doi.org/10.5555/3294771.3294885) (siehe S. 34, 85).
- [Tor05] TORVALDS, Linus: Git. [Online]. Verfügbar: <https://github.com/git>. Zugriff: 27.02.2023. 2005 (siehe S. 60, 91, 271).
- [Tro18] TRONSER, Tina; DEMIR, Konstantin; REISCHL, Markus; BASTMEYER, Martin und LEVKIN, Pavel A.: „Droplet Microarray: Miniaturized Platform for Rapid Formation and High-Throughput Screening of Embryoid Bodies“. In: *Lab on a chip* 18.15 (2018), S. 2257–2269. DOI: [10.1039/C8LC00450A](https://doi.org/10.1039/C8LC00450A) (siehe S. 1).
- [Tsc18] TSCHANDL, Philipp; ROSENDAHL, Cliff und KITTLER, Harald: „The HAM10000 Dataset, a Large Collection of Multi-Source Dermoscopic Images of Common Pigmented Skin Lesions“. In: *Scientific Data* 5.1 (2018), S. 180161. DOI: [10.1038/sdata.2018.161](https://doi.org/10.1038/sdata.2018.161) (siehe S. 246, 247).
- [Ulm17] ULMAN, Vladimír u. a.: „An Objective Comparison of Cell-Tracking Algorithms“. In: *Nature Methods* 14.12 (2017), S. 1141–1152. DOI: [10.1038/nmeth.4473](https://doi.org/10.1038/nmeth.4473) (siehe S. 20, 37, 246, 247).
- [Van08] VAN DER MAATEN, Laurens und HINTON, Geoffrey: „Visualizing Data Using T-SNE“. In: *Journal of Machine Learning Research* 9.86 (2008). Verfügbar: <http://jmlr.org/papers/v9/vandermaaten08a.html>. Zugriff: 27.02.2023, S. 2579–2605 (siehe S. 55).

- [Van14] VAN DER WALT, Stefan; SCHÖNBERGER, Johannes L; NUNEZ-IGLESIAS, Juan; BOULOGNE, François; WARNER, Joshua D; YAGER, Neil; GOUILLART, Emmanuelle und YU, Tony: „Scikit-Image: Image Processing in Python“. In: *PeerJ* (2014). DOI: [10.7717/peerj.453](https://doi.org/10.7717/peerj.453) (siehe S. 91).
- [Vas17] VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N; KAISER, Łukasz und POLOSUKHIN, Illia: „Attention Is All You Need“. In: *Advances in Neural Information Processing Systems*. [Online]. Verfügbar: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. Zugriff: 21.02.2023. Long Beach, CA, USA, 2017, S. 6000–6010 (siehe S. 27).
- [Vie22] VIERING, Tom und LOOG, Marco: „The Shape of Learning Curves: A Review“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), S. 1–20. DOI: [10.1109/TPAMI.2022.3220744](https://doi.org/10.1109/TPAMI.2022.3220744) (siehe S. 12).
- [Vio01] VIOLA, P. und JONES, M.: „Rapid Object Detection Using a Boosted Cascade of Simple Features“. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Bd. 1. Kauai, HI, USA, 2001, S. 511–518. DOI: [10.1109/CVPR.2001.990517](https://doi.org/10.1109/CVPR.2001.990517) (siehe S. 25).
- [von21] VON CHAMIER, Lucas u. a.: „Democratising Deep Learning for Microscopy with ZeroCostDL4Mic“. In: *Nature Communications* 12.1 (2021), S. 2276. DOI: [10.1038/s41467-021-22518-0](https://doi.org/10.1038/s41467-021-22518-0) (siehe S. 2).
- [Vu19] VU, Quoc Dang u. a.: „Methods for Segmentation and Classification of Digital Microscopy Tissue Images“. In: *Frontiers in Bioengineering and Biotechnology* 7 (2019), S. 53. DOI: [10.3389/fbioe.2019.00053](https://doi.org/10.3389/fbioe.2019.00053) (siehe S. 48).
- [Vya10] VYAWAHARE, Saurabh; GRIFFITHS, Andrew D. und MERTEN, Christoph A.: „Miniaturization and Parallelization of Biological and Chemical Assays in Microfluidic Devices“. In: *Chemistry & Biology* 17.10 (2010), S. 1052–1065. DOI: [10.1016/j.chembiol.2010.09.007](https://doi.org/10.1016/j.chembiol.2010.09.007) (siehe S. 1, 22).

- [Wad16] WADA, Kentaro: Labelme: Image Polygonal Annotation with Python. [Online]. Verfügbar: <https://github.com/wkentaro/labelme>. Zugriff: 28.04.2022. 2016 (siehe S. 15, 16).
- [Wad18] WADA, Kentaro: labelme: Image Polygonal Annotation with Python. [Online]. Verfügbar: <https://github.com/wkentaro/labelme>. Zugriff: 18.05.2022. 2018 (siehe S. 115).
- [Wan15] WANG, Liantao; HU, Xuelei; YUAN, Bo und LU, Jianfeng: „Active Learning via Query Synthesis and Nearest Neighbour Search“. In: *Neurocomputing* 147 (2015), S. 426–434. DOI: [10.1016/j.neucom.2014.06.042](https://doi.org/10.1016/j.neucom.2014.06.042) (siehe S. 31).
- [Wan18] WANG, Tian; CHEN, Yang; QIAO, Meina und SNOUSI, Hichem: „A Fast and Robust Convolutional Neural Network-Based Defect Detection Model in Product Quality Control“. In: *The International Journal of Advanced Manufacturing Technology* 94.9 (2018), S. 3465–3471. DOI: [10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00196](https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00196) (siehe S. 2).
- [Wan19] WANG, Guotai; LI, Wenqi; AERTSEN, Michael; DEPREST, Jan; OURSELIN, Sébastien und VERCAUTEREN, Tom: „Aleatoric Uncertainty Estimation with Test-Time Augmentation for Medical Image Segmentation with Convolutional Neural Networks“. In: *Neurocomputing* 338 (2019), S. 34–45. DOI: [10.1016/j.neucom.2019.01.103](https://doi.org/10.1016/j.neucom.2019.01.103) (siehe S. 14).
- [Wan21a] WANG, Xinlong; ZHANG, Rufeng; SHEN, Chunhua; KONG, Tao und LI, Lei: „Dense Contrastive Learning for Self-Supervised Visual Pre-Training“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Nashville, TN, USA, 2021, S. 3024–3033. DOI: [10.1109/CVPR46437.2021.00304](https://doi.org/10.1109/CVPR46437.2021.00304) (siehe S. 33, 84).
- [Wan21b] WANG, Yanke; MARCATO, Daniel; TIRUMALASETTY, Vani; KANAGARAJ, Naveen Krishna; PYLATIUK, Christian; MIKUT, Ralf; PERAVALI, Ravindra und REISCHL, Markus: „An Automated Experimentation System for the Touch-Response Quantification of Zebrafish Larvae“. In: *IEEE Transactions on Automation Science and*

- Engineering* 19.4 (2021), S. 3007–3019. DOI: [10.1109/TASE.2021.3104507](https://doi.org/10.1109/TASE.2021.3104507) (siehe S. 24).
- [Wil16] WILKINSON, Mark D. u. a.: „The FAIR Guiding Principles for Scientific Data Management and Stewardship“. In: *Scientific Data* 3.1 (2016), S. 160018. DOI: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18) (siehe S. 28, 54).
- [Wil20] WILSON, Andrew Gordon und IZMAILOV, Pavel: „Bayesian Deep Learning and a Probabilistic Perspective of Generalization“. In: *Advances in Neural Information Processing Systems*. Virtuelles Event, Kanada, 2020, S. 4697–4708 (siehe S. 13).
- [Wu19] WU, Yuxin; KIRILLOV, Alexander; MASSA, Francisco; LO, Wan-Yen und GIRSHICK, Ross: Detectron2. [Online]. Verfügbar: <https://github.com/facebookresearch/detectron2>. Zugriff: 05.01.2023. 2019 (siehe S. 27).
- [Wüh22] WÜHRL, Lorenz; PYLATIUK, Christian; GIERSCH, Matthias; LAPP, Florian; VON RINTELEN, Thomas; BALKE, Michael; SCHMIDT, Stefan; CERRETTI, Pierfilippo und MEIER, Rudolf: „DiversityScanner: Robotic Handling of Small Invertebrates with Machine Learning Methods“. In: *Molecular Ecology Resources* 22 (2022), S. 1626–1638. DOI: [10.1111/1755-0998.13567](https://doi.org/10.1111/1755-0998.13567) (siehe S. 24, 99, 101, 157, 158).
- [Xia17] XIAO, Junwei; LU, Jianfeng und LI, Xiangyu: „Davies Bouldin Index Based Hierarchical Initialization K-Means“. In: *Intelligent Data Analysis* 21.6 (2017), S. 1327–1338. DOI: [10.3233/IDA-163129](https://doi.org/10.3233/IDA-163129) (siehe S. 244).
- [Xia22] XIAO, Hui; LI, Dong; XU, Hao; FU, Shuibo; YAN, Diqun; SONG, Kangkang und PENG, Chengbin: „Semi-Supervised Semantic Segmentation with Cross Teacher Training“. In: *Neurocomputing* 508 (2022), S. 36–46. DOI: [10.1016/j.neucom.2022.08.052](https://doi.org/10.1016/j.neucom.2022.08.052) (siehe S. 34, 85).
- [Yan22] YANG, Lihe; ZHUO, Wei; QI, Lei; SHI, Yinghuan und GAO, Yang: „ST++: Make Self-Training Work Better for Semi-Supervised Semantic Segmentation“. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. New Orleans, LA, USA, 2022, S. 4268–4277. DOI: [10.1109/CVPR52688.2022.00423](https://doi.org/10.1109/CVPR52688.2022.00423) (siehe S. 35, 85).

- [Yoo03] YOO, Andy B.; JETTE, Morris A. und GRONDONA, Mark: „SLURM: Simple Linux Utility for Resource Management“. In: *Job Scheduling Strategies for Parallel Processing*. Seattle, WA, USA, 2003, S. 44–60. DOI: [10.1007/10968987_3](https://doi.org/10.1007/10968987_3) (siehe S. 91).
- [Yoo19] YOO, Donggeun und KWEON, In So: „Learning Loss for Active Learning“. In: *Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, USA, 2019, S. 93–102. DOI: [10.1109/CVPR.2019.00018](https://doi.org/10.1109/CVPR.2019.00018) (siehe S. 32).
- [Yu20] YU, Shaode; CHEN, Mingli; ZHANG, Erlei; WU, Junjie; YU, Hang; YANG, Zi; MA, Lin; GU, Xuejun und LU, Weiguo: „Robustness Study of Noisy Annotation in Deep Learning Based Medical Image Segmentation“. In: *Physics in medicine and biology* 65.17 (2020). DOI: [10.1088/1361-6560/ab99e5](https://doi.org/10.1088/1361-6560/ab99e5) (siehe S. 18).
- [Zbo21] ZBONTAR, Jure; JING, Li; MISRA, Ishan; LECUN, Yann und DENY, Stéphane: Barlow Twins: Self-supervised Learning via Redundancy Reduction. arXiv:2103.03230, [Online]. Verfügbar: <https://arxiv.org/pdf/2103.03230>. Zugriff: 12.04.2023. 2021 (siehe S. 33, 84).
- [Zha18] ZHANG, Jing; SHENG, Victor S.; LI, Tao und WU, Xindong: „Improving Crowdsourced Label Quality Using Noise Correction“. In: *IEEE Transactions on Neural Networks and Learning Systems* 29.5 (2018), S. 1675–1688. DOI: [10.1109/TNNLS.2017.2677468](https://doi.org/10.1109/TNNLS.2017.2677468) (siehe S. 17).
- [Zha19] ZHANG, Leicheng; CHEN, Peng; ZHOU, Zhuoyue; HU, Yawei; SHA, Qiuyue; ZHANG, Houjin; LIU, Xin; DU, Wei; FENG, Xiaojun und LIU, Bi-Feng: „Agarose-Based Microwell Array Chip for High-Throughput Screening of Functional Microorganisms“. In: *Talanta* 191 (2019), S. 342–349. DOI: [10.1016/j.talanta.2018.08.090](https://doi.org/10.1016/j.talanta.2018.08.090) (siehe S. 22).
- [Zho18] ZHOU, Zongwei; SIDDIQUEE, Md Mahfuzur Rahman; TAJBAKSHI, Nima und LIANG, Jianming: „UNet++: A Nested u-Net Architecture for Medical Image Segmentation“. In: *International Workshop*

on Deep Learning in Medical Image Analysis. Granada, Spanien, 2018, S. 3–11. DOI: [10.1007/978-3-030-00889-5_1](https://doi.org/10.1007/978-3-030-00889-5_1) (siehe S. 26).

- [Zhu22] ZHU, Zhaowei; DONG, Zihao und LIU, Yang: „Detecting Corrupted Labels without Training a Model to Predict“. In: *International Conference on Machine Learning*. Verfügbar: <https://proceedings.mlr.press/v162/zhu22a/zhu22a.pdf>. Zugriff: 12.04.2023. Virtuelles Event, 2022 (siehe S. 19).

Eigene Publikationen

Veröffentlichungen mit Autoren, die Beiträge zu gleichen Anteilen beigetragen haben, sind über Autor[†] gekennzeichnet.

- [Ise20] ISELE, Simon T.[†]; **SCHILLING, MARCEL P.**[†]; KLEIN, Fabian E. und ZOELLNER, J. Marius: „Annotating Radar Data Efficiently: Semantic Radar Labeling Framework (SeRaLF)“. In: *Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. Verfügbar: [https://ml4ad.github.io/files/papers2020/Annotating%20Automotive%20Radar%20efficiently:%20Semantic%20Radar%20Labeling%20Framework%20\(SeRaLF\).pdf](https://ml4ad.github.io/files/papers2020/Annotating%20Automotive%20Radar%20efficiently:%20Semantic%20Radar%20Labeling%20Framework%20(SeRaLF).pdf). Zugriff: 12.04.2023. Virtuelles Event, 2020 (siehe S. 2).
- [Ise21a] ISELE, Simon T.[†]; **SCHILLING, MARCEL P.**[†]; KLEIN, Fabian E.[†]; SARALAJEW, Sascha und ZOELLNER, J. Marius: „Radar Artifact Labeling Framework (RALF): Method for Plausible Radar Detections in Datasets“. In: *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems*. Virtuelles Event, 2021, S. 22–33. DOI: [10.5220/0010395100220033](https://doi.org/10.5220/0010395100220033) (siehe S. 2).
- [Mün21] MÜNKE, Friedrich R.; **SCHILLING, MARCEL P.**; MIKUT, Ralf und REISCHL, Markus: „Evaluierung von Merkmalen Zur Abbildung von Veränderungen in Ungeordneten Bilddaten“. In: *at - Automatisierungstechnik* 69.10 (2021), S. 892–902. DOI: [10.1515/ato-2021-0038](https://doi.org/10.1515/ato-2021-0038) (siehe S. 2).
- [Sch21a] **SCHILLING, MARCEL P.**; RETTENBERGER, Luca; MÜNKE, Friedrich; CUI, Haijun; POPOVA, Anna A.; LEVKIN, Pavel A.; MIKUT,

- Ralf und REISCHL, Markus: „Label Assistant: A Workflow for Assisted Data Annotation in Image Segmentation Tasks“. In: *Proceedings - 31. Workshop Computational Intelligence*. Berlin, Deutschland, 2021, S. 211–234. DOI: [10.5445/IR/1000139715](https://doi.org/10.5445/IR/1000139715) (siehe S. 49, 52, 53, 90).
- [Sch21b] **SCHILLING, MARCEL P.**; SCHMELZER, Svenja; GÓMEZ, Joaquín Eduardo Urrutia; POPOVA, Anna A.; LEVKIN, Pavel A. und REISCHL, Markus: „Grid Screener: A Tool for Automated High-Throughput Screening on Biochemical and Biological Analysis Platforms“. In: *IEEE Access* 9 (2021), S. 166027–166038. DOI: [10.1109/ACCESS.2021.3135709](https://doi.org/10.1109/ACCESS.2021.3135709) (siehe S. 44, 48, 52, 68, 69, 71, 74, 75, 86, 89, 97, 98, 123, 126, 128, 130, 235, 236, 265, 268, 269).
- [Böh22] BÖHLAND, Moritz[†]; NEUMANN, Oliver; **SCHILLING, MARCEL P.**; REISCHL, Markus; MIKUT, Ralf; LÖFFLER, Katharina und SCHERR, Tim[†]: „Ciscnet - a Single-Branch Cell Nucleus Instance Segmentation and Classification Network“. In: *IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*. Kalkutta, Indien, 2022, S. 1–5. DOI: [10.1109/ISBIC56247.2022.9854734](https://doi.org/10.1109/ISBIC56247.2022.9854734) (siehe S. 1).
- [Cha22] CHAKRABORTY, Shraddha; LUCHENA, Charlotte; ELTON, Jonathan J.; **SCHILLING, MARCEL P.**; REISCHL, Markus; ROUX, Margaux; LEVKIN, Pavel A. und POPOVA, Anna A.: „Cells-to-cDNA on Chip”: Phenotypic Assessment and Gene Expression Analysis from Live Cells in Nanoliter Volumes Using Droplet Microarrays“. In: *Advanced Healthcare Materials* (2022), S. 2102493. DOI: [10.1002/adhm.202102493](https://doi.org/10.1002/adhm.202102493) (siehe S. 1, 155, 157, 158).
- [Neu22] NEUMANN, Oliver; **SCHILLING, MARCEL P.**; REISCHL, Markus und MIKUT, Ralf: „EasyMLServe: Easy Deployment of REST Machine Learning Services“. In: *Proceedings - 32. Workshop Computational Intelligence*. Berlin, Deutschland, 2022, S. 11–30. DOI: [10.5445/IR/1000154156](https://doi.org/10.5445/IR/1000154156) (siehe S. 86).
- [Ret22] RETTENBERGER, Luca; **SCHILLING, MARCEL P.** und REISCHL, Markus: „Annotation Efforts in Image Segmentation Can Be

- Reduced by Neural Network Bootstrapping“. In: *Current Directions in Biomedical Engineering* 8.2 (2022), S. 329–332. DOI: [10.1515/cdbme-2022-1084](https://doi.org/10.1515/cdbme-2022-1084) (siehe S. 57).
- [Sch22a] SCHERR, Tim; SEIFFARTH, Johannes; WOLLENHAUPT, Bastian; NEUMANN, Oliver; **SCHILLING, MARCEL P.**; KOHLHEYER, Dietrich; SCHARR, Hanno; NÖH, Katharina und MIKUT, Ralf: „microbeSEG: A Deep Learning Software Tool with OMERO Data Management for Efficient and Accurate Cell Segmentation“. In: *PLOS ONE* 17.11 (2022), S. 1–14. DOI: [10.1371/journal.pone.0277601](https://doi.org/10.1371/journal.pone.0277601).
- [Sch22b] **SCHILLING, MARCEL P.**; AHUJA, Niket; RETTENBERGER, Luca; SCHERR, Tim und REISCHL, Markus: „Impact of Annotation Noise on Histopathology Nucleus Segmentation“. In: *Current Directions in Biomedical Engineering* 8.2 (2022), S. 197–200. DOI: [10.1515/cdbme-2022-1051](https://doi.org/10.1515/cdbme-2022-1051) (siehe S. 49, 61, 62, 118, 119).
- [Sch22c] **SCHILLING, MARCEL P.**; NEUMANN, Oliver; SCHERR, Tim; CUI, Haijun; POPOVA, Anna A.; LEVKIN, Pavel A.; GÖTZ, Markus und REISCHL, Markus: „A Computational Workflow for Interdisciplinary Deep Learning Projects Utilizing bwHPC Infrastructure“. In: *Proceedings of the 7th bwHPC Symposium*. Virtuelles Event, 2022, S. 69–74. DOI: [10.18725/OPARU-46164](https://doi.org/10.18725/OPARU-46164) (siehe S. 24, 86, 90).
- [Sch22d] **SCHILLING, MARCEL P.**; SCHERR, Tim; MÜNKE, Friedrich R.; NEUMANN, Oliver; SCHUTERA, Mark; MIKUT, Ralf und REISCHL, Markus: „Automated Annotator Variability Inspection for Biomedical Image Segmentation“. In: *IEEE Access* 10 (2022), S. 2753–2765. DOI: [10.1109/ACCESS.2022.3140378](https://doi.org/10.1109/ACCESS.2022.3140378) (siehe S. 49, 50, 52, 61, 65, 67, 89, 118, 119, 237, 239, 252).
- [Sch22e] **SCHILLING, MARCEL P.**; SCHMELZER, Svenja; KLINGER, Lukas und REISCHL, Markus: „KaIDA: a modular tool for assisting image annotation in deep learning“. In: *Journal of Integrative Bioinformatics* 19.4 (2022), S. 20220018. DOI: [10.1515/jib-2022-0018](https://doi.org/10.1515/jib-2022-0018) (siehe S. 49, 52, 53, 86, 89, 90, 92–95, 107, 108, 112, 114, 115, 258, 271).

- [Wie22] WIEDMANN, Janne J. u. a.: „Combining Miniaturized Chemical Synthesis with Biochemical Screening for High-Throughput Discovery“. In: *ELRIG DRUG DISCOVERY*. London, England, 2022 (siehe S. 24, 141, 155, 156, 158).
- [Cui23] CUI, Haijun; SUN, Xueyuan; **SCHILLING, MARCEL P.**; HEROLD-MENDE, Christel; REISCHL, Markus; LEVKIN, Pavel A.; POPOVA, Anna A. und TURCAN, Sevin: „Repurposing FDA-Approved Drugs for Temozolomide-resistant IDH1 Mutant Glioma using High-Throughput Miniaturized Screening on Droplet Microarray Chip“. In: *Advanced Healthcare Materials* 12.24 (2023), S. 2300591. DOI: [10.1002/adhm.202300591](https://doi.org/10.1002/adhm.202300591) (siehe S. 1, 102, 136, 155, 157, 158).
- [Gra23] GRAHAM, Simon u. a.: „CoNIC Challenge: Pushing the frontiers of nuclear detection, segmentation, classification and counting“. In: *Medical Image Analysis* (2023), S. 103047. DOI: [10.1016/j.media.2023.103047](https://doi.org/10.1016/j.media.2023.103047).
- [Mol23a] MOLINA-MORENO, Miguel; **SCHILLING, MARCEL P.**; REISCHL, Markus und MIKUT, Ralf: „ASAP: Automated Style-Aware Selection of Pre-training Datasets in 2D Biomedical Imaging“. In: *Unter Review* (2023) (siehe S. 80, 82, 143, 242, 245).
- [Mol23b] MOLINA-MORENO, Miguel[†]; **SCHILLING, MARCEL P.**[†]; REISCHL, Markus und MIKUT, Ralf: „Automated Style-aware Selection of Annotated Pre-training Databases in Biomedical Imaging“. In: *IEEE International Symposium on Biomedical Imaging (ISBI)*. Cartagena, Kolumbien, 2023, S. 1–5. DOI: [10.1109/ISBI53787.2023.10230659](https://doi.org/10.1109/ISBI53787.2023.10230659) (siehe S. 51, 80, 82, 143, 242, 245).
- [Ret23] RETTENBERGER, Luca[†]; **SCHILLING, MARCEL P.**[†]; ELSER, Stefan; BÖHLAND, Moritz und REISCHL, Markus: „Self-Supervised Learning with Limited Data for Medical Image Segmentation“. In: *IEEE Transactions on Biomedical Engineering* 70.9 (2023), S. 2519–2528. DOI: [10.1109/TBME.2023.3252889](https://doi.org/10.1109/TBME.2023.3252889) (siehe S. 51, 80, 152).

- [Sch23b] **SCHILLING, MARCEL P.**; EL KHALED EL FARAJ, Razan; URRUTIA GÓMEZ, Joaquín Eduardo; SONNENTAG, Steffen J.; WANG, Fei; NESTLER, Britta; ORIAN-ROUSSEAU, Véronique; POPOVA, Anna A.; LEVKIN, Pavel A. und REISCHL, Markus: „Automated High-Throughput Image Processing as Part of the Screening Platform for Personalized Oncology“. In: *Scientific Reports* 13.1 (2023), S. 5107. DOI: [10.1038/s41598-023-32144-z](https://doi.org/10.1038/s41598-023-32144-z) (siehe S. 22, 41, 42, 75, 76, 78, 103, 104, 108, 109, 131–141).
- [Sch23c] **SCHILLING, MARCEL P.**; KLINGER, Lukas; SCHUMACHER, Ulrike; SCHMELZER, Svenja; BORDALLO LÓPEZ, Miguel; NESTLER, Britta und REISCHL, Markus: „AI²Seg: A Method and Tool for AI-based Annotation Inspection of Biomedical Instance Segmentation Datasets“. In: *Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. Sydney, Australien, 2023, S. 1–6. DOI: [10.1109/EMBC40787.2023.10341074](https://doi.org/10.1109/EMBC40787.2023.10341074) (siehe S. 49, 52, 61, 62, 65, 89, 95, 96, 118, 119, 238, 264).

Patente

- [Ise21] ISELE, Simon T. und **SCHILLING, MARCEL P.**: „Verfahren und System zum automatischen Labeling von Radardaten“. DE 10 2020 123 920 B3 (Deutschland). 2021.
- [Bis22] BISCHOF, Fabian; ISELE, Simon T.; KLEIN, Fabian und **SCHILLING, MARCEL P.**: „Kraftfahrzeug mit subjektiv angepasster Dynamik eines Fahrbeeinflussungssystems, System aufweisend eine Mehrzahl von Kraftfahrzeugen, Verfahren zum Betrieb eines Kraftfahrzeugs und Verfahren zum Betrieb eines Systems“. DE 10 2021 106 226 A1 (Deutschland). 2022.
- [Ise22a] ISELE, Simon T. und **SCHILLING, MARCEL P.**: „Labeling Method“. KR 102022077082 A1 (Korea). 2022.
- [Ise22b] ISELE, Simon T. und **SCHILLING, MARCEL P.**: „Marking Method“. CN000114578373A (China). 2022.
- [Ise22c] ISELE, Simon T. und **SCHILLING, MARCEL P.**: „Method and system for automatically labeling radar data“. US 20220083841 A1 (USA). 2022.
- [Roo22] ROOS, Stefan; ISELE, Simon T.; **SCHILLING, MARCEL P.** und BENTZ, Björn: „Verfahren zur dreidimensionalen Rekonstruktion einer Szene in einem Vorfeld eines Fahrzeugs“. DE 10 2021 102 818 B3 (Deutschland). 2022.
- [Sch23] **SCHILLING, MARCEL P.** und RUNFT, Marc: „Verfahren, System und Computerprogrammprodukt zur Erkennung der Umgebung eines Kraftfahrzeugs“. DE 10 2021 119 951 A1 (Deutschland). 2023.

Betreute studentische Arbeiten

- [Kli21] KLINGER, Lukas: „Hyperparameter-Tuning zur Optimierung von Deep Learning Methoden“. Projektarbeit. Karlsruhe, Deutschland: Duale Hochschule Baden-Württemberg, 2021 (siehe S. 91).
- [Ret21] RETTENBERGER, Luca: „Evaluation and Comparison of Methods to Increase Data Efficiency in the Area of Deep Learning for Processing Biomedical Imaging Data“. Masterarbeit. Weingarten, Deutschland: Hochschule Ravensburg-Weingarten University of Applied Sciences, Fakultät für Informatik, 2021 (siehe S. 47, 51, 80, 152).
- [Sch21c] SCHMELZER, Svenja: „Entwicklungsbeitrag zu den Bildverarbeitungstools Grid Screener und KalDA“. Praktikumsprotokoll. Heidelberg, Deutschland: Universität Heidelberg, Fakultät für Ingenieurwissenschaften (Molekulare Biotechnologie), 2021 (siehe S. 52).
- [Kep22] KEPPLER, Gustav: „Evaluation and Comparison of Semi-Supervised Learning Methods for Biomedical Image Processing“. Masterarbeit. Karlsruhe, Deutschland: Karlsruher Institut für Technologie, Fakultät für Maschinenbau, 2022 (siehe S. 51, 80, 150).
- [Kli23] KLINGER, Lukas: „Konzept zur Erweiterung von selbstkonfigurierenden Deep Learning Methoden über annotationseffizientes Lernen“. Masterarbeit. Heidelberg, Deutschland: Universität Heidelberg, Fakultät für Informatik, 2023.

- [Rod23] RODRIGUES, Fabio: „Development of Novel Methods for Evaluating Datasets for Transfer Learning Using Uncertainty and Heterogeneity“. Masterarbeit. Karlsruhe, Deutschland: Karlsruher Institut für Technologie, Fakultät für Mathematik, 2023 (siehe S. [51](#), [80](#), [82](#), [143](#), [241](#), [245](#)).
- [Sch23d] SCHMELZER, Svenja: „MiCATS: Development of a Novel Microscopy-Coupled Analysis Tool for Spores Based on Deep Learning“. Masterarbeit. Heidelberg, Deutschland: Universität Heidelberg, Fakultät für Ingenieurwissenschaften (Molekulare Biotechnologie), 2023 (siehe S. [24](#), [155](#), [156](#), [158](#)).

Abbildungsverzeichnis

1.1	Visualisierung der Forschungsfrage	3
1.2	Übersicht der offenen Teilprobleme.	4
1.3	Aufbau der Arbeit	7
2.1	Supervised Learning	11
2.2	Gittertypen.	22
2.3	Fragestellungen im Bereich der Analyse von Elementen.	24
2.4	Übersicht effizientes Lernen.	29
2.5	Antworten von ChatGPT.	37
3.1	Konzeptüberblick	42
3.2	Notation.	45
3.3	Überblick Datensätze.	47
3.4	Illustration integrales Gütekriterium	51
3.5	Überblick assistiertes Annotieren.	53
3.6	Synthetischer Datensatz zur semantischen Segmentierung inklusive Verunreinigungen	61
3.7	Synthetischer Datensatz der Instanzsegmentierung inklusive Verunreinigungen	62
3.8	Simulation der Experten-Variabilität.	64
3.9	Automatisierte Inspektion der Intra-Experten-Variabilität.	65
3.10	Automatisierte Inspektion der Inter-Experten-Variabilität.	67
3.11	Gesamtmethodik zur automatisierten Bildverarbeitung von Gitterstrukturen in Hochdurchsatz.	69
3.12	Rotationschätzung.	71
3.13	Robustheit und Anwendungsgrenzen.	75
3.14	Prozessierung Mikroebene.	75

3.15	Analysepipeline zur Segmentierung in Hochdurchsatzexperimenten.	76
3.16	Analysepipeline für die mehrdimensionale Instanzanalyse.	78
3.17	Methodenvergleich bezüglich Verfahren zum effizienten Lernen.	81
3.18	Pipeline zur automatisierten Auswahl ähnlicher Datensätze für TL.	82
3.19	Überblick der Architekturkonzeption für interdisziplinäre Datenanalyseprojekte.	86
4.1	KaIDA.	93
4.2	Aufbau der Implementierung von Bildverarbeitungsaufgaben in KaIDA.	94
4.3	Aufbau der Implementierung von Methoden in KaIDA.	95
4.4	AI ² Seg GUI.	96
4.5	Grid Screener.	98
4.6	Aufbau der elementweisen Prozessierung im Grid Screener.	98
5.1	Auszüge aus den Datensätzen	100
5.2	Selektion Heterogenität.	104
5.3	Selektion Unsicherheit.	105
5.4	Vergleich Segmentierungsfehler und Unsicherheit.	106
5.5	„Smart Picking“.	106
5.6	Pre-Processing.	108
5.7	Pre-Annotation.	109
5.8	Vergleich Annotationszeit und Annotationsqualität Pre-Annotation.	111
5.9	Post-Annotation-Processing.	113
5.10	Annotation-Inspektion.	114
5.11	Robustheit und Anwendungsgrenzen der Gitterschätzung.	128
5.12	Quantitative Analyse der Anwendungsgrenzen der Gitterschätzung.	129
5.13	Vorhersagen der Seed-Detektion-Analysepipeline.	133

5.14	Vorhersagen der Segmentierung.	135
5.15	Ergebnisse Zellviabilität.	137
5.16	Farbanalyse DMA I.	138
5.17	Vergleich zwischen Quantifizierung und potentia Hydrogenii (pH)-Wert.	139
5.18	Farbanalyse DMA II.	140
5.19	Intensitätsanalyse DMA.	141
5.20	Beispielbilder zu Kooperationen.	158
C.1	Architektur Ähnlichkeitsschätzer.	242
C.2	Vergleichsdatensätze TL	247
D.1	Metriken semantische Segmentierung.	250
D.2	Illustration Q_{IDaIS}	252
E.1	U-Net Architektur.	254
E.2	ResNet50 Architektur.	255
F.1	Benutzeroberfläche zur Projekterstellung in KaIDA.	259
F.2	Oberfläche für Training und Anwendung in KaIDA.	260
F.3	Benutzeroberfläche für das assistierte Annotieren in KaIDA.	262
F.4	Benutzeroberfläche für die Inspektion eines Datensatzes in AI ² Seg.	264
F.5	Benutzeroberfläche Grid Screener.	265
F.6	Eingabefenster für die semi-automatisierte Gitterschätzung im Werkzeug Grid Screener.	267
G.1	Gitterschätzung Wellplate.	268
G.2	Gitterschätzung DMA - Kreis.	268
G.3	Gitterschätzung DMA - Quadrat.	269

Tabellenverzeichnis

5.1	Überblick Datensätze.	99
5.2	Vergleich der Software-Tools zur Annotation	115
5.3	Resultate Intra-Experten-Variabilität semantische Segmentierung	119
5.4	Resultate Intra-Experten-Variabilität Instanzsegmentierung	119
5.5	Resultate Inter-Experten-Variabilität semantische Segmentierung	122
5.6	Resultate Inter-Experten-Variabilität Instanzsegmentierung	123
5.7	Bewertung der Gitterschätzung.	126
5.8	Vergleich Software-Tools Hochdurchsatzverarbeitung	130
5.9	Vergleich Seed Detektion	132
5.10	Vergleich Segmentierung	134
5.11	Exemplarische Merkmalsliste $\mathcal{F}_{i,j}$	136
5.12	Quantifizierung der Ähnlichkeit von Datensätzen auf Basis der Unsicherheit.	144
5.13	Vergleich der Ansätze.	144
5.14	Quantifizierung der Ähnlichkeit von Datensätzen auf Basis von Bildmerkmalen.	145
5.15	Kombinierte Quantifizierung der Ähnlichkeit von Datensätzen.	146
5.16	Vergleich der datensatzspezifischen Metriken für TL.	147
5.17	Vergleich der datensatzübergreifenden Metriken für TL.	147
5.18	Vergleich der datensatzspezifischen Metriken für DAL.	148
5.19	Vergleich der datensatzübergreifenden Metriken für DAL.	149
5.20	Vergleich der datensatzspezifischen Metriken für SemSL.	150

5.21	Vergleich der datensatzübergreifenden Metriken für SemSL.	151
5.22	Vergleich der datensatzspezifischen Metriken für SSL.	153
5.23	Vergleich der datensatzübergreifenden Metriken für SSL.	153
5.24	Überblick der Kooperationen	156
E.1	Details zur Optimierung.	255
E.2	Details zum Training.	256
E.3	Details zu den Zielfunktionen.	257
H.1	Details zu den verwendeten python-Softwarepaketen.	270
I.1	Details zum prototypischen Bildverarbeitungsserver.	271

Liste der Algorithmen

1	Rotations- und Abstandsschätzung	235
2	Gitterschätzung	236
3	Inspektionsalgorithmus Intra-Experten-Variabilität für semantische Segmentierung	237
4	Inspektionsalgorithmus Intra-Experten-Variabilität für Instanzsegmentierung	238
5	Inspektionsalgorithmus Inter-Experten-Variabilität	240
6	Ähnlichkeitsschätzung mittels Unsicherheit	241
7	Ähnlichkeitsschätzung auf Basis der Bildmerkmale	244
8	Ähnlichkeitsschätzung mit Unsicherheit und Bildmerkmalen	246
9	Aggregated Jaccard Index Q_{AJI}	251
10	inverted Aggregated Jaccard Index Agreement in Sample Q_{iAAiS}	252

Abkürzungsverzeichnis

Adam	Adaptive Moment Estimation
AE	Autoencoder
AI²Seg	AI-based Annotation Inspection for Segmentation
AJI	Aggregated Jaccard Index
AL	Active Learning
ALR	Alte Lernrate
API	Application Programming Interface
AUC	Area Under the Curve
BYOL	Bootstrap Your Own Latent
CCT	Cross Teacher Training
CL	Contrastive Learning
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DAL	Deep Active Learning

DBI	Davies Bouldin Index
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DE	Deep Ensemble
DL	Deep Learning
DMA	Droplet Microarray
DNN	Deep Neural Network
DoG	Difference of Gaussian
DoH	Determinant of Hessian
DSC	Dice-Sørensen Coefficient
DVC	Data Version Control
FAIR	Findable Accessible Interoperable Reusable
GPU	Graphics Processing Unit
GUI	Graphical User Interface
GVLRL	Global verringerte Lernrate
HD	Hochdurchsatz
HDE	Hochdurchsatzexperiment
HPC	High-Performance Computing

HSV	Hue-Saturation-Value (zu Deutsch: Farbwert, Farbsättigung, Helligkeit)
iAAiS	inverted Aggregated Jaccard Index Agreement in Sample
iDAiS	inverted Dice Agreement in Sample
ILT	Image Labeling Tool
IoU	Intersection over Union
ISIC	International Skin Imaging Collaboration
JI	Jaccard Index
KaIDA	Karlsruhe Image Data Annotation Tool
KI	Künstliche Intelligenz
KIT	Karlsruher Institut für Technologie
KNN	k-Nearest-Neighbor
LoG	Laplacian of Gaussian
LSDF	Large Scale Data Facility
LVLR	Linear verringerte Lernrate
MCD	Monte Carlo Dropout
MICCAI	Medical Image Computing and Computer Assisted Intervention
Mix	Gemischte Datensätze im Training

ML	Maschinelles Lernen
MLP	Multi-Layer Perceptron
MoCo	Momentum Contrast
MoNuSeg	Multi-Organ Nucleus Segmentation
NN	Neural Network
PCA	Principal Component Analysis
pH	potentia Hydrogenii
PI	Propidium Iodide
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
REST	Representational State Transfer
ROC	Receiver Operating Characteristic
SemSL	Semi-Supervised Learning
SGD	Stochastic Gradient Descent
SimCLR	Simple Framework for Contrastive Learning of Visual Representations
SL	Supervised Learning
SLURM	Simple Linux Utility for Resource Management

SNE	Stochastic Neighbor Embedding
SSL	Self-Supervised Learning
TL	Transfer Learning
TTA	Test Time Augmentation
UL	Unsupervised Learning
UMAP	Uniform Manifold Approximation and Projection

Notation

Allgemeine Notation

Skalare	Kursive Kleinbuchstaben	a
Mengen	Kalligrafische Großbuchstaben	\mathcal{A}
Vektoren	Kleinbuchstaben in Fettdruck	a
Tensoren	Großbuchstaben in Fettdruck	A
Zahlenmengen	Großbuchstaben mit doppelten Linien	\mathbb{A}

Lateinische Buchstaben

a_1, \dots, a_n	Stichprobenmenge einer Zufallsvariable A
\mathcal{A}	Indexmenge aller Bilder mit vorliegender Annotation
$\mathcal{A}_{\text{insp}}$	Geordnete Menge der Bilder zur Inspektion
A_{circle}	Flächeninhalt kreisförmiges Element
A_{square}	Flächeninhalt quadratisches Element
b_1, \dots, b_n	Stichprobenmenge einer Zufallsvariable B
C	Bildtiefe
c_u	Clusterzuordnung horizontal
c_v	Clusterzuordnung vertikal
\mathbb{C}	Komplexe Zahlen
d	Formparameter des Gitterelements
\mathcal{D}	Generischer Datensatz

$\tilde{\mathcal{D}}$	Verunreinigter Datensatz
\mathcal{D}^a	Annotierter Datensatz
\mathcal{D}^{na}	Nicht annotierter Datensatz
\mathcal{D}^{pa}	Partiell annotierter Datensatz
${}^A\mathcal{D}, {}^B\mathcal{D}$	Subdatensatz A bzw. B
\mathbf{e}_i	Encoder-Stylemerkmal
f	Allgemeine Funktion
f_θ	Parametrierbare Funktion mit Parametern θ
$f_{\theta,insp}$	Inspektionsfunktion
$f_{\theta,micro}$	Bildverarbeitungsfunktion Mikroebene
$f_{\theta,spot}$	Detektionsfunktion Elemente
$f_{\theta,total}$	Gesamtbildverarbeitungsipeline
f_{trafo}	Abstrakte Transformationsfunktion
$f_{Q(\cdot),lin}$	Lineare Approximation der Gütefunktion $Q(\cdot)$ bezüglich der Annotationsrate ρ , Lernkurve
$\mathcal{F}_{i,j}$	Merkmalsliste des Elements $\mathbf{x}_{i,j}$
f_n	Falsch-negative Vorhersagen
f_p	Falsch-positive Vorhersagen
FPR	Falsch-Positiv-Rate
H	Bildhöhe
I	Anzahl der Zeilen einer Gitterstruktur
$J_{insp,i}$	Geordnete Menge der Instanzen zur Inspektion bezüglich Bild \mathbf{x}_i
J	Anzahl der Spalten einer Gitterstruktur
K	Anzahl der annotierten Datensätze
L	Anzahl der nicht annotierten Datensätze
l	Fehlerfunktion überwachtes Lernen
∇l	Gradient der Fehlerfunktion
$l_{triplet}$	Triplet-Fehlerfunktion basierend auf latenten Merkmalen

$l_{\text{triplet-style}}$	Triplet-Fehlerfunktion basierend auf Stylemerkmale
l_{mse}	Mittlerer quadratischer Fehler der Bildmerkmale
$l_{\text{mse-style}}$	Mittlerer quadratischer Fehler der Stylemerkmale
\mathcal{L}_u	Listen der geschätzten horizontalen Linien
\mathcal{L}_v	Listen der geschätzten vertikalen Linien
m_i	Annotations-effiziente Methode i
\mathcal{M}	Menge der Methoden im Kontext annotations-effizientes Lernen
M	Anzahl der annotierten Elemente eines Datensatzes oder Gesamtanzahl der annotations-effizienten Lernverfahren
\mathbb{N}	Natürliche Zahlen
N	Anzahl der Elemente eines Datensatzes
N_{emb}	Dimension des latenten Merkmalsraum
\emptyset	Leere Menge
\mathcal{P}	Menge der Mittelpunktskoordinaten
$\mathcal{P}_{\text{neighbors}}$	Menge der Mittelpunktskoordinaten der nächsten Nachbarn
$\mathbf{p}_{i,j}$	Mittelpunktvektor des Elements i,j
p	p -Wert eines t -Tests
Q	Qualitätskriterium, Metrik
$Q_{\Delta p}$	Metrik zur Bewertung der Schätzung der Gitterpositionen
$Q_{\Delta d}$	Metrik zur Bewertung der Schätzung der Gitterabmessung
$Q_{t_{\text{grid, process}}}$	Prozessierungszeit Gitterschätzung
$Q_{t_{\text{grid, user}}}$	Zeitaufwand Benutzer Gitterschätzung
$Q_{t_{\text{grid}}}$	Gesamtzeitaufwand Gitterschätzung
$Q_{t_{\text{element, process}}}$	Prozessierungszeit Elementverarbeitung
$Q_{t_{\text{element, user}}}$	Zeitaufwand Benutzer Elementverarbeitung

$Q_{t_{\text{element}}}$	Gesamtzeitaufwand Elementverarbeitung
Q_t	Rechenzeit
$Q_{t,\text{norm}}$	Normiertes Gütekriterium Rechenzeit
$Q_{t_{\text{annotation}}}$	Annotationszeitdauer
Q_{DSC}	Dice-Sørensen Coefficient (DSC)-Score
ΔQ_{DSC}	Veränderung des DSC-Scores zwischen zwei Annotationen
Q_{DSC2}	DSC-Score auf Instanzsegmentierung verändert
Q_{AJI}	Aggregated Jaccard Index (AJI)-Score
$Q_{\text{AJI+}}$	AJI-Score ohne Überbestrafung
$\Delta Q_{\text{AJI+}}$	Veränderung des AJI-Scores ohne Überbestrafung zwischen zwei Annotationen
Q_{IoU}	Intersection over Union (IoU)-Score
Q_{DBI}	Davies Bouldin Index (DBI) Metrik
$Q_{\Gamma_{\text{sim, unc}}}$	Metrik zur Bewertung der Ähnlichkeitsschätzung zwischen Datensätzen unter Einbezug der Unsicherheit von Vorhersagen
$Q_{\Gamma_{\text{sim}}}$	Metrik zur Bewertung der Ähnlichkeitsschätzung zwischen Datensätzen ohne Einbezug der Unsicherheit von Vorhersagen
$Q_{\text{int},(\cdot)}$	Integrales Gütekriterium
$Q_{\text{Top-1}}$	Top-1 Genauigkeit
$Q_{\text{Top-3}}$	Top-3 Genauigkeit
$Q_{\text{Top-R}}$	Top-R Genauigkeit
Q_{iDAiS}	Unsicherheitsmaß Segmentierung (inverted Dice Agreement in Sample (iDAiS))
Q_{iAAiS}	Unsicherheitsmaß Instanzsegmentierung (inverted Aggregated Jaccard Index Agreement in Sample (iAAiS))
Q_{AUC}	Area Under the Curve (AUC)
$q_{i,\text{insp}}$	Inspektionsscores der Instanzen in Bild \mathbf{x}_i

\mathcal{Q}	Liste der Unsicherheiten eines Datensatzes, Liste der Übereinstimmungsscores zweier annotierter Datensätze
\mathbb{R}	Reelle Zahlen
R	Parameter Top- R -Genauigkeit $Q_{\text{Top-}R}$
r	Betrag/Abstand in Nachbarschaftsbeziehung von Elementen
r_n	Richtig-negative Vorhersagen
r_p	Richtig-positive Vorhersagen
RPR	Richtig-Positiv-Rate
\mathcal{S}_i	Liste aller Zustände von Instanzen bezüglich des Bildes \mathbf{x}_i
T	Anzahl der Einzelvorhersagen
\mathcal{T}	Liste von Übereinstimmungen von Einzelvorhersagen
triplet	Triplet-Zielfunktion
θ	Parametervektor Bildverarbeitungsipeline
\mathcal{U}	Liste horizontaler Bildkoordinaten
u	Horizontale Bildkoordinate
u_i	Skalare Unsicherheit bezüglich des Bildes \mathbf{x}_i
\mathbf{u}_i	Unsicherheit der Instanzen/pro Pixel in Bild \mathbf{x}_i , Decoder-Stylemerkmal
\mathcal{V}	Liste vertikale Bildkoordinaten
v	Vertikale Bildkoordinate
w_j	Parameter zur Gewichtung der Schichttiefe in der Zielfunktion des Ähnlichkeitsschätzers
W	Bildbreite
\mathcal{X}	Menge aller Bilder \mathbf{x}
\mathbf{x}	Eingangsbild
$\hat{\mathbf{x}}$	Rekonstruiertes Bild
$\check{\mathbf{x}}_{i,j}$	Vorverarbeitetes Eingangsbild des Elements i,j

$\mathbf{x}_{i,j}$	Eingangsbild des Elements i,j
\mathcal{y}	Menge aller Annotationen bzw. Ausgabevektoren \mathbf{y}
\mathbf{y}	Ausgabevektor
$\hat{\mathbf{y}}$	Verunreinigter Ausgabevektor
$\mathbf{y}_{i,j}$	Ausgabevektor bezüglich Element i,j
$\check{\mathbf{y}}_{i,j}$	Nachbearbeiteter Ausgabevektor bezüglich Element i,j
y_{shape}	Form des Gitterelements
$\hat{\mathbf{y}}$	Vorhersage
$\hat{\mathbf{y}}^1, \dots, \hat{\mathbf{y}}^T$	Stichprobenmenge mit $T \in \mathbb{N}$ Vorhersagen
$\hat{\mathbf{y}}_{\text{spot}}$	Vorhersage der Gitterelemente
$\check{\mathbf{y}}_{\text{spot}}$	Nachbearbeitete Vorhersage der Gitterelemente
$\check{\mathbf{y}}_{\text{spot},\hat{\alpha}}$	Nachbearbeitete und um Drehung korrigierte Vorhersage der Gitterelemente
\mathbb{Z}	Ganze Zahlen
\mathcal{Z}	Liste der Bildmerkmale eines Datensatzes
\mathbf{z}_i	Latenter Merkmalsvektor

Griechische Buchstaben

α	Drehwinkel einer Gitterstruktur
β	Rate der verunreinigten Instanzen pro Bild
γ	Rate der verunreinigten Bilder pro Datensatz
Γ	Ähnlichkeit zwischen Datensätzen
Γ_{het}	Ähnlichkeit zwischen Datensätzen auf Basis von Bildinhalten
Γ_{unc}	Ähnlichkeit zwischen Datensätzen auf Basis von Unsicherheit
Γ_{sim}	Ähnlichkeit zwischen Annotationen von Datensätzen

φ	Winkel in Nachbarschaftsbeziehungen von Elementen
θ	Parametervektor eines Deep Neural Network (DNN)
ρ	Annotationsrate eines Datensatzes
ξ	Lineare Korrelation nach Pearson
ζ	Parameter zur Gewichtung der Unsicherheit eines DNNs
ν	Parameter zur Rekonstruktionsgewichtung
κ	Parameter zur Stylegewichtung
η	Parameter zur Skalierung des Einflusses der Instanzgröße
χ	Parameter zur Gewichtung der Anteile von Unsicherheit und Bildinhalten bei der Ähnlichkeitsschätzung von Datensätzen
ψ_k	Dichte der Merkmale des Datensatzes $^k\mathcal{D}$ im latenten Merkmalsraum
σ_{kl}	Abstand der Merkmalsmittelpunkte der Datensätze $^k\mathcal{D}$ bzw. $^l\mathcal{D}$
ω	Parameter zur Skalierung der geschätzten Ähnlichkeit zwischen Datensätzen

Operatoren

$\arg \min(\cdot)$	Argument des Minimums
$d(\cdot, \cdot)$	Distanzfunktion
$\dim(\cdot)$	Dimension eines Vektors
$(\bar{\cdot})$	Mittelwert der Größe
$\text{card}(\cdot)$	Kardinalität
$\min(\cdot)$	Minimumsfunktion

$\max(\cdot)$	Maximumsfunktion
$(\hat{\cdot})$	Prädiktion
$(\tilde{\cdot})$	Verrauschte oder verunreinigte Größe
$(\check{\cdot})$	Vor- oder nachprozessierte Größe
$(\cdot)^*$	Optimale Größe
\approx	Ähnlichkeit
\mapsto	Abbildung

Funktionen in Algorithmen

<code>region_props</code>	Berechnung der Eigenschaften von Segmenten
<code>k_nearest_neighbors</code>	k-Nearest-Neighbor (KNN)-Algorithmus
<code>complex</code>	Transformation in \mathbb{C} -Zahlenraum
<code>feasibility_check</code>	Überprüfung der Gültigkeit der Nachbarschaftsbeziehung
<code>estimator</code>	Schätzung des Rotationswinkels bzw. Abstands
<code>clustering</code>	Clusterverfahren
<code>estimate_line</code>	Linienschätzung
<code>get_intersection</code>	Bestimmung von Linienschnittpunkten
<code>predict_w_uncertainty</code>	Vorhersage der Segmentierung inklusive deren Unsicherheit
<code>overall_inspection</code>	Inspektion des Gesamtdatensatzes aus Basis der Einzelbildinspektionen
<code>image_wise_inspection</code>	Inspektion der Annotation im Falle der Instanzsegmentierung
<code>state_assignment</code>	Zuweisung von Zuständen zu allen Instanzen eines Bildes
<code>matching</code>	Berechnung der Übereinstimmung

<code>overall_rating</code>	Berechnung der Ähnlichkeit bezüglich der Annotationsstile zweier Datensätze
<code>similarity_estimation</code>	Schätzung der Ähnlichkeit zweier Bild-Datensätze
<code>predict_uncertainty</code>	Schätzung der Unsicherheit einer Vorhersage
<code>outlier_removal</code>	Bereinigung der Ausreißer
<code>median</code>	Medianberechnung
<code>average</code>	Mittelwertberechnung

A Algorithmen zur automatisierten Gittererkennung

Algorithmus 1 Rotations- und Abstandsschätzung

```

Input:  $\check{Y}_{\text{spot}}$ 
Output:  $\hat{\alpha}, \hat{r}$ 
 $\hat{\alpha}, \hat{r} \leftarrow \text{init\_list}(), \text{init\_list}()$ 
 $\mathcal{P} \leftarrow \text{region\_props}(\check{Y}_{\text{spot}})[\text{centroids}]$  ▷ bestimme Zentroide
for  $i = 1, \dots, \text{length}(\mathcal{R})$  do
   $p_i^u, p_i^v \leftarrow \mathcal{R}[i]$ 
   $\mathcal{P}_{\text{neighbors}} \leftarrow \text{k\_nearest\_neighbors}(k = 4, \mathcal{P} \setminus \mathcal{P}[i])$  ▷ ermittle die 4 nächsten Nachbarn des Zentroids i
   $\mathbf{r}_{\text{neighbor}}, \varphi \leftarrow \text{init\_list}(), \text{init\_list}()$ 
  for  $j = 1, \dots, 4$  do
     $p_j^u, p_j^v \leftarrow \mathcal{P}_{\text{neighbors}}[j]$ 
     $r_j, \varphi_j \leftarrow \text{complex}(u_j - u_i, v_j - v_i)$  ▷ komplexer Zahlenraum  $\mathbb{C}$  zur Nachbarschaftsbeschreibung
     $\mathbf{r}_{\text{neighbor}}.\text{append}(r_j), \varphi.\text{append}(\varphi_j)$ 
  end for
  if  $\text{feasibility\_check}(\mathbf{r}_{\text{neighbor}}, \varphi)$  then ▷ prüfe, ob Nachbarschaft gültig ist
     $\hat{\alpha}_i, \hat{r}_i \leftarrow \text{estimator}(\mathbf{r}_{\text{neighbor}}, \varphi)$  ▷ schätze Winkel und Abstand der lokalen Nachbarschaft
     $\hat{\alpha}.\text{append}(\hat{\alpha}_i), \hat{r}.\text{append}(\hat{r}_i)$ 
  end if
end for
 $\hat{\alpha}, \hat{r} \leftarrow \text{median}(\hat{\alpha}), \text{median}(\hat{r})$  ▷ nutze Median für Robustheit

```

Der Algorithmus 1 stellt die entwickelte Methode zur Schätzung des Drehwinkels bzw. Abstands dar, die auf der Vorveröffentlichung [Sch21b] basiert. Eine ausführliche Erklärung findet sich im Hauptteil der Arbeit unter Abschnitt 3.5.2.

Wenn das Gitter nur eine oder zwei Zeilen bzw. Spalten enthält, wird die Anzahl der nächsten Nachbarn automatisch reduziert. Die automatische Reduktion der betrachteten Nachbarschaftssegmente erfolgt für den Fall, dass keine gegebene Winkeldifferenz von etwa $\varphi \approx \frac{\pi}{2}$ zwischen allen vier Nachbarn besteht. Kann dieses Winkelkriterium auch bei drei benachbarten Elementen

nicht erfüllt werden, wird die Anzahl der Nachbarschaftssegmente auf zwei Nachbarn bzw. eine Winkeldifferenz von π (ungefähr, wegen Detektionsungenauigkeiten) reduziert.

Algorithmus 2 Gitterschätzung

Input: $\check{Y}_{\hat{\alpha}, \text{spot}}, \hat{r}$
Output: $\hat{I}, \hat{J}, \hat{p}_{i,j}$

```

 $\mathcal{P} \leftarrow \text{region\_props}(\check{Y}_{\hat{\alpha}, \text{spot}})$  [centroids] ▷ bestimme Zentroide
 $\mathcal{U}, \mathcal{V} \leftarrow \text{init\_list}(), \text{init\_list}()$ 
for  $i = 1, \dots, \text{length}(\mathcal{P})$  do
     $p_i^u, p_i^v \leftarrow \mathcal{P}[i]$ 
     $\mathcal{U}.\text{append}(p_i^u), \mathcal{V}.\text{append}(p_i^v)$  ▷ teile Detektion nach Koordinatenrichtungen auf
end for
 $\mathcal{C}_u, \mathcal{C}_v \leftarrow \text{clustering}(\mathcal{U}, \hat{r}), \text{clustering}(\mathcal{V}, \hat{r})$  ▷ bestimme Cluster-Zugehörigkeiten
 $\mathcal{L}_u, \mathcal{L}_v \leftarrow \text{init\_list}(), \text{init\_list}()$ 
 $\hat{I}, \hat{J} \leftarrow \max(\mathcal{C}_v) + 1, \max(\mathcal{C}_u) + 1$ 
for  $j = 0, \dots, \hat{J} - 1$  do
     $\mathcal{L}_u.\text{append}(\text{estimate\_line}(\mathcal{P}[\mathcal{C}_u = j]))$  ▷ schätze vertikale Linien
end for
for  $k = 0, \dots, \hat{I} - 1$  do
     $\mathcal{L}_v.\text{append}(\text{estimate\_line}(\mathcal{P}[\mathcal{C}_v = k]))$  ▷ schätze horizontale Linien
end for
 $\hat{p}_{i,j} \leftarrow \text{get\_intersections}(\mathcal{L}_u, \mathcal{L}_v)$  ▷ bestimme Schnittpunkte,  $i = 1, \dots, \hat{I}, j = 1, \dots, \hat{J}$ 

```

In Algorithmus 2 ist die konzipierte Vorgehensweise zur Gitterschätzung dargestellt, welche im Rahmen der Arbeit in [Sch21b] vorgestellt worden ist. In Abschnitt 3.5.2 des Hauptteils der Arbeit werden die einzelnen Bestandteile des Algorithmus erklärt.

Wenn das Gitter nur eine einzige Zeile oder Spalte enthält, kann keine Linie geschätzt werden. In diesem Fall wird das Gitter direkt durch die Mittelpunkte der Segmente beschrieben, die Teil des einzelnen Clusters sind.

B Algorithmen zur automatisierten Inspektion der Expertenvariabilität

B.1 Intra-Experten-Variabilität

Nachfolgend soll auf die Details der entwickelten Algorithmen zur Inspektion im Kontext von Intra-Experten-Variabilität (vgl. Abschnitt 3.4) eingegangen werden. Es werden Methoden für die Bildverarbeitungsaufgaben der semantischen Segmentierung (siehe Algorithmus 3) und Instanzsegmentierung (siehe Algorithmus 4) vorgestellt.

Algorithmus 3 Inspektionsalgorithmus Intra-Experten-Variabilität für semantische Segmentierung

Input: $\tilde{\mathcal{D}}, f_{\theta, \text{insp}}, \zeta, \eta$

Output: $\mathcal{A}_{\text{insp}}$

for $i = 1, \dots, N$ do

$\hat{\mathbf{y}}_i, \hat{u}_i \leftarrow \text{predict_w_uncertainty}(f_{\theta, \text{insp}}, \mathbf{x}_i)$

$q_{i, \text{insp}} \leftarrow (1 - \zeta)(1 - Q_{\text{DSC}}(\hat{\mathbf{y}}_i, \mathbf{y}_i)) + \zeta(1 - \hat{u}_i)$

end for

$\mathcal{A}_{\text{insp}} \leftarrow \text{overall_inspection}(\{\dots, q_{i, \text{insp}}, \dots\})$

Semantische Segmentierung Algorithmus 3 zeigt die Details hinsichtlich der Inspektion im Falle der semantischen Segmentierung für Intra-Experten-Variabilität und basiert auf der im Rahmen der Arbeit entstandenen Publikation in [Sch22d]. Eingänge sind das trainierte DNN $f_{\theta, \text{insp}}$, der verunreinigte Datensatz $\tilde{\mathcal{D}}$ und der Parameter $\zeta \in [0, 1]$ zur Einstellung der Unsicherheitsgewichtung. Für jedes Bild \mathbf{x}_i wird eine Vorhersage $\hat{\mathbf{y}}_i$ inklusive geschätzter

Unsicherheitsquantifizierung $\hat{\mathbf{u}}_i$ bezüglich des Gesamtbilds erzeugt. Im Anschluss wird damit ein Inspektionscore $q_{i,\text{insp}}$ über die gewichtete Summe aus Diskrepanz zwischen Vorhersage $\hat{\mathbf{y}}_i$ und verrauschter Annotation $\tilde{\mathbf{y}}_i$ bzw. Sicherheit der Vorhersage gebildet. Abschließend erfolgt basierend auf den Inspektionscores $q_{i,\text{insp}}$ die Generierung einer geordneten Liste $\mathcal{A}_{\text{insp}}$, welche eine priorisierte Inspektion von besonders kritischen Annotationen ermöglicht (vgl. `overall_inspection`).

Algorithmus 4 Inspektionsalgorithmus Intra-Experten-Variabilität für Instanzsegmentierung

Input: $\hat{\mathcal{D}}, f_{\theta,\text{insp}}, \zeta, \eta$
Output: $\mathcal{A}_{\text{insp}}, \{\dots, \mathcal{J}_{i,\text{insp}}, \dots\}$
for $i = 1, \dots, N$ **do**
 $\hat{\mathbf{y}}_i, \hat{\mathbf{u}}_i \leftarrow \text{predict_w_uncertainty}(f_{\theta,\text{insp}}, \mathbf{x}_i)$
 $\mathcal{S}_i \leftarrow \text{state_assignment}(\hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i)$
 $\mathcal{J}_{i,\text{insp}} \leftarrow \text{image_wise_inspection}(\mathcal{S}_i, \hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i, \hat{\mathbf{u}}_i, \zeta, \eta)$
end for
 $\mathcal{A}_{\text{insp}} \leftarrow \text{overall_inspection}(\{\dots, \mathcal{J}_{i,\text{insp}}, \dots\})$

Instanzsegmentierung Der Pseudocode der entwickelten Methodik zur Inspektion in puncto Intra-Experten-Variabilität von Instanzsegmentierungsdatensätzen findet sich in Algorithmus 4 und basiert auf der im Rahmen der Arbeit entstandenen Veröffentlichung in [Sch23c].

Zunächst wird das trainierte DNN $f_{\theta,\text{insp}}$ verwendet, um eine Vorhersage $\hat{\mathbf{y}}_i$ für das Bild $\mathbf{x}_i \in \hat{\mathcal{D}}$ zu generieren. Um das Potenzial einer Betrachtung der Unsicherheiten von Prädiktionen zu eruieren, wird die Unsicherheit $\hat{\mathbf{u}}_i$ bezüglich aller Instanzen parallel vorhergesagt (vgl. `predict_w_uncertainty`). Die Arbeit von Roy et al. [Roy19] definiert eine Metrik zur Berechnung der Unsicherheit über einen Quervergleich im Falle der semantischen Segmentierung (siehe Gleichung (D.7)). Angelehnt daran wird dieser Gedanke auf einzelne Instanzen j erweitert. Deren Unsicherheit bezogen auf die finale Vorhersage ergibt sich durch den Quervergleich der Übereinstimmungen in jeder DNN-Vorhersage.

Anschließend vergleicht die Funktion `state_assignment` die verrauschte Annotation $\tilde{\mathbf{y}}_i \in \tilde{\mathcal{D}}$ mit der Vorhersage $\hat{\mathbf{y}}_i$. Es soll zwischen den drei Zuständen der Instanzen j in $\tilde{\mathbf{y}}_i/\hat{\mathbf{y}}_i$ differenziert werden: i) *nicht übereinstimmend* (Schnittmenge zwischen der Vorhersage $\hat{\mathbf{y}}_i$ und der Annotation $\tilde{\mathbf{y}}_i$ vorhanden, aber nicht die gleiche Form), ii) *zusätzlich* (Instanz nur in der Annotation $\tilde{\mathbf{y}}_i$ präsent) und iii) *fehlend* (Instanz nur in der Vorhersage $\hat{\mathbf{y}}_i$ vorhanden). Die Zustände werden in der Liste \mathcal{S}_i zusammengefasst.

Daraufhin folgt über die Funktion `image_wise_inspection` die Inspektionsreihenfolge $\mathcal{J}_{i,\text{insp}}$ bezüglich aller Instanzen in Bild \mathbf{x}_i . Im Allgemeinen wird mithilfe von Q_{IoU} zwischen annotierten und vorhergesagten Instanzen die Abweichung pro Instanz berechnet ($1 - Q_{\text{IoU}}$). Um die Größe einer Instanz im Fall von $Q_{\text{IoU}} = 0$ (*fehlend, zusätzlich*) zu berücksichtigen, wird eine Skalierung der Abweichung ermöglicht. Mittels Parameter $\eta \in [0,1]$ kann die Abweichungsmetrik in Bezug auf die Segmentfläche skaliert werden. Die Normalisierung schwankt zwischen der maximalen Größe aller gegebenen Instanzen ($\eta = 0$) oder der maximalen Größe aller sich nicht überschneidenden Instanzen ($\eta = 1$). Darüber hinaus ist die endgültige Bewertung pro Instanz eine gewichtete Summe aus fehlender Übereinstimmung zwischen Annotation bzw. Vorhersage und der Sicherheit der vorhergesagten Instanz. Der Parameter $\zeta \in [0,1]$ steuert den Einfluss der Unsicherheit, wobei $\zeta = 0$ bedeutet, dass die Unsicherheit nicht berücksichtigt wird.

Die Inspektionsreihenfolge bezüglich aller Bilder des Datensatzes $\tilde{\mathcal{D}}$, ausgedrückt durch $\mathcal{A}_{\text{insp}}$, ergibt sich aus der Mittelung aller Inspektionsbewertungen der Listen $\mathcal{J}_{i,\text{insp}}$ pro Bild \mathbf{x}_i , die als Gesamtbewertung bezeichnet wird (vgl. `overall_inspection`).

B.2 Inter-Experten-Variabilität

Eine Beschreibung der konzipierten Methodik zur Inspektion der Inter-Experten-Variabilität findet sich in Algorithmus 5, die sich an [Sch22d] anlehnt. Eingangsgrößen stellen dabei zum einen die von Experten A/B annotierten Datensätze ${}^A\tilde{\mathcal{D}}/{}^B\tilde{\mathcal{D}}$ dar. Zum anderen sind die DNNs ${}^A f_{\theta,\text{insp}}/{}^B f_{\theta,\text{insp}}$

Algorithmus 5 Inspektionsalgorithmus Inter-Experten-Variabilität

Input: ${}^A\tilde{\mathcal{D}}, {}^B\tilde{\mathcal{D}}, {}^A f_{\theta, \text{insp}}, {}^B f_{\theta, \text{insp}}, \zeta$
Output: $\hat{\Gamma}_{\text{sim}}({}^A\tilde{\mathcal{D}}, {}^B\tilde{\mathcal{D}})$

```

 $\mathcal{Q} \leftarrow \emptyset$ 
for each  $k \in \{A, B\}$  do
     $l \leftarrow \{A, B\} \setminus k$ 
    for  $i = 1, \dots, {}^k N$  do
         ${}^l \hat{\mathbf{y}}_i, \hat{u}_i \leftarrow \text{predict\_w\_uncertainty}({}^l f_{\theta, \text{insp}}, {}^k \mathbf{x}_i)$ 
         $\mathcal{Q}.\text{append}(\text{matching}({}^l \hat{\mathbf{y}}_i, {}^k \hat{\mathbf{y}}_i), \hat{u}_i)$ 
    end for
end for
 $\hat{\Gamma}_{\text{sim}}({}^A\tilde{\mathcal{D}}, {}^B\tilde{\mathcal{D}}) \leftarrow \text{overall\_rating}(\mathcal{Q}, \zeta)$ 

```

die auf den genannten Datensätzen ${}^A\tilde{\mathcal{D}}/{}^B\tilde{\mathcal{D}}$ trainiert worden sind, weitere Eingangsgrößen. Ferner kann über den Parameter $\zeta \in [0,1]$ der Einbezug von DNN-Unsicherheiten eingestellt werden.

Für jedes Bild ${}^k \mathbf{x}_i \in {}^k \tilde{\mathcal{D}}$ liegt eine zugehörige Annotation ${}^k \hat{\mathbf{y}}_i$ des entsprechenden Experten vor. Mittels ${}^l f_{\theta, \text{insp}}$ kann im Allgemeinen eine fiktive Annotation im Stil von Experte l über die Vorhersage ${}^l \hat{\mathbf{y}}_i$ generiert werden (vgl. `predict_w_uncertainty`), um einen Vergleich zu ermöglichen. Zudem erfolgt eine Vorhersage der Gesamtunsicherheit pro Bild, notiert über \hat{u}_i .

Über die `matching`-Funktion können die Annotationsstile über eine zur Aufgabe passende Vergleichsmetrik (z. B. $Q_{\text{AJI+}}, Q_{\text{DSC}}$) verglichen werden. Die Ergebnisse der Vergleiche und vorliegenden Unsicherheiten bei der Bewertung werden in die Liste \mathcal{Q} gespeichert.

Final erfolgt dann über die Funktion `overall_rating` die Vorhersage $\hat{\Gamma}_{\text{sim}}({}^A\tilde{\mathcal{D}}, {}^B\tilde{\mathcal{D}})$ hinsichtlich der Ähnlichkeit der Annotationsstile. Im naiven Fall werden die numerischen Ergebnisse aus den Vergleichen gemittelt. Allerdings kann im Falle von $\zeta > 0$ zusätzlich eingestellt werden, dass Vergleiche mit sicheren Vorhersagen größeren Einfluss auf das Endresultat haben. Dabei wird ein auf die Sicherheit der Vorhersage gewichteter Mittelwert genutzt.

C Methoden zur Schätzung der Ähnlichkeit von Datensätzen

C.1 Unsicherheit

Algorithmus 6 Ähnlichkeitsschätzung mittels Unsicherheit

Input: $(\mathcal{D}_k^a)_{k=1,\dots,K}$, \mathcal{D}^{na} , $(f_{\theta, \mathcal{D}_k^a})_{k=1,\dots,K}$
Output: $\Gamma(\mathcal{D}_i^a, \mathcal{D}^{\text{na}})_{k=1,\dots,K}$

- 1: **for each** annotated dataset \mathcal{D}_k^a **do**
- 2: $\mathcal{Q} \leftarrow \emptyset$
- 3: **for each** image ${}^k\mathbf{x}_j$ in \mathcal{D}_k^a **do**
- 4: $\mathcal{Q}.\text{append}(\text{predict_uncertainty}({}^k\mathbf{x}_j, f_{\theta, \mathcal{D}_k^a}))$
- 5: **end for**
- 6: $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{\text{na}}) \leftarrow \text{similarity_estimation}(\mathcal{Q})$
- 7: **end for**

In Algorithmus 6 findet sich der entwickelte Ansatz zur Schätzung der Ähnlichkeit von Datensätzen auf der Basis von Unsicherheiten. Diese Methode basiert auf [Rod23], die im Rahmen dieser Arbeit entstanden ist.

Zunächst wird angenommen, dass für die bereits annotierten Datensätze $(\mathcal{D}_k^a)_{k=1,\dots,K}$ unsicherheitsbewusste DNNs $(f_{\theta, \mathcal{D}_k^a})_{k=1,\dots,K}$ trainiert worden sind. Für den zu lösenden bzw. nicht annotierten Datensatz wird jeweils für alle Bilder ${}^k\mathbf{x}_j$ deren Unsicherheit bezüglich des Netzes vorhergesagt, welches auf dem annotierten Datensatz \mathcal{D}_k^a trainiert worden ist (vgl. `predict_uncertainty`). Die genaueste Methode zur Schätzung der Unsicherheit stellt dabei Test Time Augmentation (TTA) in Kombination mit der Metrik $Q_{\text{ID AIS}}$ (vgl. Gleichung (D.7)) dar. Die Liste aller Unsicherheiten \mathcal{Q} bildet die

Basis zur Ähnlichkeitsschätzung mittels Funktion der `similarity_estimation`. Dabei wird der Median aller Unsicherheiten ermittelt und anschließend über dessen Negation auf die vorhergesagte Ähnlichkeit $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{na})$ umgerechnet.

C.2 Bildmerkmale

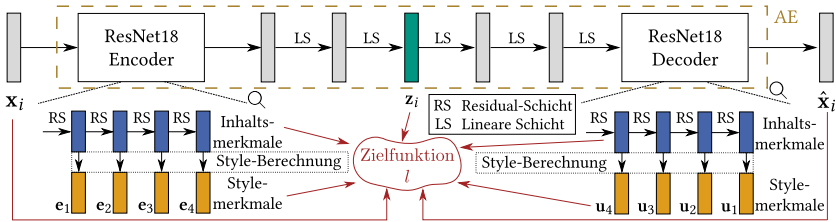


Abbildung C.1: Architektur Ähnlichkeitsschätzer. Es wird ein Autoencoder (AE) verwendet, der jeweils ein ResNet18 als Netzarchitektur verwendet, um das Bild x_i zu \hat{x}_i zu rekonstruieren. Zwei lineare Schichten (LS) ermöglichen eine zusätzliche Merkmalsreduktion, um die finalen latenten Merkmale z_i zu erhalten. Über eine Style-Berechnung erfolgt die Überführung der Encoder- bzw. Decoder-Merkmale in Stylemerkmale $(e_i/u_i)_{i=1,\dots,4}$ [Mol23a, Mol23b]. ©2023 IEEE.

Gemäß der im Rahmen der Arbeit entstanden Veröffentlichungen [Mol23a, Mol23b] zeigt Abbildung C.1 die entwickelte Architektur, die Ähnlichkeitsschätzungen von Datensätzen basierend auf deren Bildmerkmalen durchführt. Für ein Bild x_i können mittels der Merkmale eines ResNet18-Encoders [He16] die Stylemerkmale $(e_i)_{i=1,\dots,4}$ über die Berechnung der Gram-Matrizen ermittelt werden.

Im Nachgang wird der latente Merkmalsvektor $z_i \in \mathbb{R}^{N_{\text{emb}}}$ berechnet, indem über zwei lineare Schichten die Dimensionen weiter reduziert werden. Es gilt für die Dimension des latenten Merkmalsraums $N_{\text{emb}} \in \mathbb{N}^+$. Analog zum Encoder ergeben sich die Decoder-Stylemerkmale $(u_i)_{i=1,\dots,4}$. Mittels $\hat{x}_i(x_i)$ wird die vorhergesagte Bildrekonstruktion beschrieben.

Die neu entwickelte Architektur stellt eine Komposition aus Berücksichtigung von Bildrekonstruktion bzw. von Inhalts- und Stylemerkmalen dar.

Die Zielfunktion bezüglich der Bildrekonstruktion kann über

$$l_{\text{mse}} = (\hat{\mathbf{x}}(\mathbf{x}) - \mathbf{x})^2 \quad (\text{C.1})$$

beschrieben werden, wobei sich diese analog auch für die Stylemerkmale über

$$l_{\text{mse-style}} = \sum_{j=1}^4 w_j (\mathbf{e}_j(\mathbf{x}) - \mathbf{u}_j(\mathbf{x}))^2 \quad (\text{C.2})$$

berechnen lässt. Die Wichtungsfaktoren $w_j \in \mathbb{R}^+$ sind einzustellende Hyperparameter. Die AE-ähnliche Architektur kann dadurch vom Stil ähnliche Datensätze im latenten Merkmalsraum näher zueinander bringen bzw. Abstände im Falle von unterschiedlichen Datensätzen aufbauen.

Ähnlich dazu folgt der Triplet-Anteil der Zielfunktion [Che10]

$$l_{\text{triplet}} = \text{triplet}(\mathbf{z}, \mathbf{z}^p, \mathbf{z}^n), \quad (\text{C.3})$$

wobei \mathbf{z}^p den positiven Anker (Bild des gleichen Datensatzes) und \mathbf{z}^n den negativen Anker (Bild eines anderen Datensatzes) bezüglich des latenten Merkmalsvektors \mathbf{z}_i darstellt. Damit wird in den AE die Fähigkeit integriert, Bilder der verschiedenen Datensätze im latenten Raum zu trennen.

Der Triplet-Anteil bezüglich der Stylemerkmale berechnet sich analog über

$$l_{\text{triplet-style}} = \sum_{j=1}^4 w_j \text{triplet}(\mathbf{e}_l(\mathbf{x}), \mathbf{e}_l^p(\mathbf{x}), \mathbf{e}_l^n(\mathbf{x})). \quad (\text{C.4})$$

Damit folgt die zusammenfassende Zielfunktion

$$l = \nu (\kappa l_{\text{mse}} + (1 - \kappa) l_{\text{mse-style}}) + (1 - \nu) (\kappa l_{\text{triplet}} + (1 - \kappa) l_{\text{triplet-style}}), \quad (\text{C.5})$$

wobei die Parameter $\nu \in [0,1]$ den Rekonstruktionsanteil bzw. $\kappa \in [0,1]$ die Styleanteile ins Verhältnis setzen. Aus dem Training des AE kann im Anschluss der Merkmalsextraktor $f_{\theta, \text{FE}}$ bestimmt werden, der zur Ähnlichkeits-schätzung genutzt wird.

Es soll hervorgehoben werden, dass die entwickelte Methode keine Annotationen der Datensätze erfordert und unabhängig von der eigentlich zu lösenden Aufgabe ist. Somit lässt sich die konzipierte Methode auf verschiedene Bildverarbeitungsproblemtypen anwenden. Folglich können Datensätze ohne die Notwendigkeit von aufwändig zu generierenden Annotationen in den latenten Merkmalsraum für einen Vergleich eingebunden werden.

Algorithmus 7 Ähnlichkeitsschätzung auf Basis der Bildmerkmale

Input: $(\mathcal{D}_k^a)_{k=1,\dots,K}$, \mathcal{D}^{na} , $f_{\theta,\text{FE}}$
Output: $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})_{k=1,\dots,K}$

- 1: $\mathcal{Z}^{\text{na}} \leftarrow f_{\theta,\text{FE}}(\mathcal{D}^{\text{na}})$
- 2: $\mathcal{Z}^{\text{na}} \leftarrow \text{outlier_removal}(\mathcal{Z}^{\text{na}})$
- 3: **for each** annotated dataset \mathcal{D}_k^a **do**
- 4: $\mathcal{Z}_k^a \leftarrow f_{\theta,\text{FE}}(\mathcal{D}_k^a)$
- 5: $\mathcal{Z}_k^a \leftarrow \text{outlier_removal}(\mathcal{Z}_k^a)$
- 6: $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{\text{na}}) \leftarrow \text{similarity_estimation}(\mathcal{Z}^{\text{na}}, \mathcal{Z}_k^a)$
- 7: **end for**

In Algorithmus 7 findet sich der Pseudocode zur Ermittlung der Ähnlichkeiten auf Basis der Bildmerkmale

Zunächst wird der Merkmalsextraktor $f_{\theta,\text{FE}}$ verwendet, um die Abbildung \mathcal{Z}^{na} des zu lösenden Datensatzes \mathcal{D}^{na} im latenten Merkmalsraum zu ermitteln. Dies erfolgt analog für die annotierten Datensätze, wodurch die Liste \mathcal{Z}_k^a bestehend aus Bildmerkmalen resultiert. Für die beiden Beschreibungen im latenten Merkmalsraum wird eine Ausreißerkorrektur durchgeführt, um die Robustheit zu erhöhen (vgl. `outlier_removal`). Im Nachgang erfolgt dann basierend darauf die Ähnlichkeitsschätzung in der Funktion `similarity_estimation`. Dazu wird der DBI [Xia17]

$$Q_{\text{DBI}}(\mathcal{D}_k, \mathcal{D}_l) = \frac{\psi_k + \psi_l}{\sigma_{kl}} \tag{C.6}$$

genutzt, um die Ähnlichkeit zwischen zwei Datensätzen \mathcal{D}_i und \mathcal{D}_k zu bestimmen. Dort wird über

$$\psi_k = \frac{1}{N_k} \sum_{i=1}^{N_k} d(\mathbf{z}^{(k)} \mathbf{x}_i, {}^k \bar{\mathbf{z}}) \quad (\text{C.7})$$

die Dichte der Merkmale des Datensatzes im latenten Merkmalsraum bestimmt, wobei $d(\cdot, \cdot)$ eine beliebige Distanzfunktion (z. B. euklidischer Abstand oder Kosinusähnlichkeit) und ${}^l \bar{\mathbf{z}}$ den Mittelpunkt der Merkmale des Datensatzes \mathcal{D}_k charakterisieren. Ferner erfolgt über

$$\sigma_{kl} = d({}^k \bar{\mathbf{z}}, {}^l \bar{\mathbf{z}}) \quad (\text{C.8})$$

die Berechnung des Abstands beider Mittelpunkt der Datensätze. Um Ähnlichkeiten im Bereich von $\Gamma \in [0,1]$ sicherzustellen, wird über eine verschobene logistische Sigmoidfunktion ($[0, \infty] \mapsto [0,1]$)

$$\Gamma(\mathcal{D}_k, \mathcal{D}_l) = 1 - 2 \left(\frac{1}{1 + \exp(-\omega(1/Q_{\text{DBI}}(\mathcal{D}_k, \mathcal{D}_l)))} - \frac{1}{2} \right) \quad (\text{C.9})$$

eine Skalierung der Ähnlichkeiten durchgeführt, wobei $\omega \in \mathbb{R}^+$ zur Parametrierung der Skalierung dient.

C.3 Kombination aus Unsicherheit und Bildmerkmalen

Die entwickelte Methodik zur Kombination von Algorithmus 6 und Algorithmus 7 ist in Algorithmus 8 dargestellt, die im Rahmen der Arbeit in [Mol23a, Mol23b, Rod23] veröffentlicht worden ist. Der Vorteil ist in diesem Fall, dass ein Vergleich auf Basis von verschiedenen Messgrößen (Unsicherheit und Bildinhalt) zu genaueren Ähnlichkeitsvorhersagen führen kann (vgl. Abschnitt 5.4.1).

Prinzipiell werden jeweils zunächst getrennt die Ähnlichkeiten $\Gamma_{\text{unc}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})$ (vgl. Algorithmus 6) bzw. $\Gamma_{\text{het}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})$ (vgl. Algorithmus 7) berechnet.

Algorithmus 8 Ähnlichkeitsschätzung mit Unsicherheit und Bildmerkmalen

Input: $(\mathcal{D}_k^a)_{i=1,\dots,K}$, \mathcal{D}^{na} , $f_{\theta, \text{FE}}$, $(f_{\theta}, \mathcal{D}_k^a)_{k=1,\dots,K}$, χ

Output: $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})_{k=1,\dots,K}$

- 1: $\mathcal{Z}^{\text{na}} \leftarrow f_{\theta, \text{FE}}(\mathcal{D}^{\text{na}})$
- 2: **for each** annotated dataset \mathcal{D}_k^a **do**
- 3: $\mathcal{Z}_k^a \leftarrow f_{\theta, \text{FE}}(\mathcal{D}_k^a)$
- 4: $\mathcal{Z}_k^a \leftarrow \text{outlier_removal}(\mathcal{Z}_k^a)$
- 5: $\Gamma_{\text{het}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}}) \leftarrow \text{similarity_estimation}(\mathcal{Z}^{\text{na}}, \mathcal{Z}_k^a)$
- 6: $\mathcal{Q} \leftarrow \emptyset$
- 7: **for each** image ${}^k\mathbf{x}_j$ in \mathcal{D}_k^a **do**
- 8: $\mathcal{Q}.\text{append}(\text{predict_uncertainty}({}^k\mathbf{x}_j, f_{\theta}, \mathcal{D}_k^a))$
- 9: **end for**
- 10: $\Gamma_{\text{unc}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}}) \leftarrow \text{similarity_estimation}(\mathcal{Q})$
- 11: $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{\text{na}}) \leftarrow (1 - \chi)\Gamma_{\text{het}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}}) + \chi\Gamma_{\text{unc}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})$
- 12: **end for**

Über den Hyperparameter $\chi \in [0,1]$ zur Gewichtung können die einzelnen Ähnlichkeiten $\Gamma_{\text{unc}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})$ und $\Gamma_{\text{het}}(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})$ ins Verhältnis gesetzt, um die finale Ähnlichkeit $\Gamma(\mathcal{D}_k^a, \mathcal{D}^{\text{na}})$ zu berechnen.

C.4 Vergleichsdatensätze

In Abbildung C.2 findet sich eine Übersicht der Datensätze, die für alle Untersuchungen im Bereich TL als bereits annotiert angenommen werden und damit die Ausgangsdomänen darstellen. Dabei sind zwei Vertreter aus dem Bereich Histopathologie (vgl. CryoNuSeg [Mah21], PanNuKe [Gam19]), Vertreter der Cell Tracking Challenge [Ulm17] (vgl. DIC-C2DH-HeLa, Fluo-C2DL-Huh7, Fluo-N2DH-GOWT, Fluo-N2DH-SIM+, Fluo-N2DL-HeLa, PhC-C2DH-U373, PhC-C2DL-PSC) bzw. BV-2-Mikrogliazellen einer Hellfeldmikroskopie (vgl. LIVECell-BV2 [Edl21]) und ein Dermatologie-Datensatz (vgl. HAM10000 [Tsc18]) enthalten. Zudem ist der Echtweltdatensatz Pascal VOC [Eve10], als Vertreter außerhalb der biomedizinischen Domäne, Bestandteil der untersuchten Datensätze.

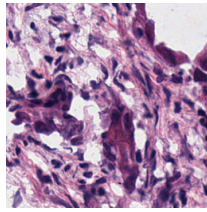
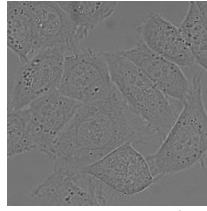
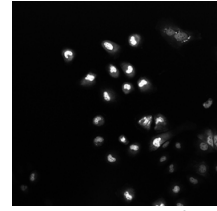
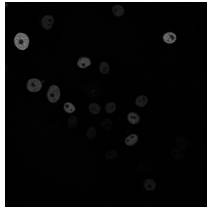
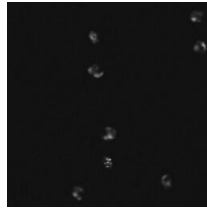
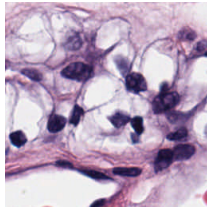
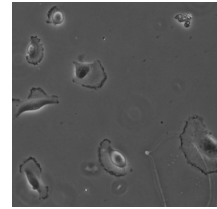
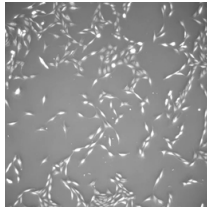
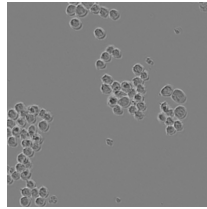
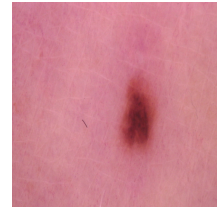
(a) CryoNuSeg (1^{a} \mathcal{D}^{a})(b) DIC-C2DH-HeLa (2^{a} \mathcal{D}^{a})(c) Fluo-C2DL-Huh7 (3^{a} \mathcal{D}^{a})(d) Fluo-N2DH-GOWT1 (4^{a} \mathcal{D}^{a})(e) Fluo-N2DH-SIM+ (5^{a} \mathcal{D}^{a})(f) Fluo-N2DL-HeLa (6^{a} \mathcal{D}^{a})(g) PanNuKe (7^{a} \mathcal{D}^{a})(h) PASCAL VOC (8^{a} \mathcal{D}^{a})(i) PhC-C2DH-U373 (9^{a} \mathcal{D}^{a})(j) PhC-C2DL-PSC (10^{a} \mathcal{D}^{a})(k) LIVECell-BV2 (11^{a} \mathcal{D}^{a})(l) HAM10000 (12^{a} \mathcal{D}^{a})

Abbildung C.2: Vergleichsdatensätze TL. Beispielbilder der Datensätze (a) CryoNuSeg [Mah21], (b) DIC-C2DH-HeLa [Ulm17], (c) Fluo-C2DL-Huh7 [Ulm17], (d) Fluo-N2DH-GOWT [Ulm17], (e) Fluo-N2DH-SIM+ [Ulm17], (f) Fluo-N2DL-HeLa [Ulm17], (g) PanNuKe [Gam19], (h) PASCAL VOC [Eve10], (i) PhC-C2DH-U373 [Ulm17] (j) PhC-C2DL-PSC [Ulm17], (k) LIVECell-BV2 [Edl21] und (l) HAM10000 [Tsc18] sind dargestellt.

D Metriken

Eine Übersicht zu Metriken im Bereich der Bildverarbeitung findet sich beispielsweise in [Mai23]. Im Weiteren soll auf die Metriken eingegangen werden, die speziell im Rahmen dieser Arbeit wichtig sind.

D.1 Korrelationen

Um den Zusammenhang zweier Stichprobenmengen $a_1, \dots, a_n \in \mathbb{R}$ and $b_1, \dots, b_n \in \mathbb{R}$ der Zufallsvariablen A and B zu untersuchen, kann der Pearson-Korrelationskoeffizient

$$\xi_{a,b} = \frac{\sum_{i=1}^n (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2} \sqrt{\sum_{i=1}^n (b_i - \bar{b})^2}}, \quad (\text{D.1})$$

genutzt werden [Mik08], wobei \bar{a} bzw. \bar{b} die entsprechenden Mittelwerte beschreiben. Die Metrik misst den linearen Zusammenhang der beiden Stichprobenmengen. Es gilt $\xi_{a,b} \in [-1, 1]$, wobei $\xi_{a,b} \rightarrow -1$ einen exakt negativen linearen bzw. $\xi_{a,b} \rightarrow 1$ einen exakt positiven linearen Zusammenhang anzeigt. Allerdings ist die Metrik im Allgemeinen für nicht-monotone Zusammenhänge ungeeignet.

D.2 Klassifikation

Gemäß [Mik08, Moh12] kann im Falle einer binären Klassifikation zwischen richtig-positiven r_p , richtig-negativen r_n , falsch-positiven f_p und falsch-negativen f_n Vorhersagen unterschieden werden. Für die Genauigkeit resultiert dann

$$Q_{\text{Top-1}} = \frac{r_p + r_n}{r_p + r_n + f_p + f_n}. \quad (\text{D.2})$$

Falls der Prädiktionsalgorithmus Schätzwerte pro Klasse liefert, kann die Top- R -Genauigkeit ($R \in \mathbb{N}$)

$$Q_{\text{Top-}R} = \frac{R r_p + R r_n}{R r_p + R r_n + R f_p + R f_n} \quad (\text{D.3})$$

analog berechnet werden, wobei $R(\cdot)$ die entsprechenden Größen bezeichnen, bei denen Vorhersagen als korrekt gelten, wenn sich die Schätzwerte der korrekten Klasse in den R besten (englisch „top“) Klassen der Vorhersage enthalten sind. Es gilt hierbei $Q_{\text{Top-}R} \geq Q_{\text{Top-1}}$ für $R > 1$.

Es folgt als Richtig-Positiv-Rate $\text{RPR} = \frac{r_p}{r_p + f_n}$ bzw. Falsch-Positiv-Rate $\text{FPR} = \frac{f_p}{r_n + f_p}$, welche innerhalb der Receiver Operating Characteristic (ROC) für jegliche Schwellwerte hinsichtlich der Vorhersagen gegenübergestellt werden. Vorteil der ROC ist, dass sich das Qualitätskriterium unabhängig von einem definierten Schwellwert bezüglich der Entscheidung berechnen lässt. Zur Beurteilung eines Klassifikators eignet sich die AUC, die über

$$Q_{\text{AUC}} = \int_0^1 \text{RPR}(\text{FPR}) \, d\text{FPR} \quad (\text{D.4})$$

bestimmt wird und dementsprechend dem Flächeninhalt der Kurve entspricht. Es soll angemerkt werden, dass $Q_{\text{AUC}} > 0.5$ die Überlegenheit des Klassifikators gegenüber einer zufälligen Klassifikation ($Q_{\text{AUC}} = 0.5$) charakterisiert, wohingegen $Q_{\text{AUC}} < 0.5$ eine vorwiegend falsche Klassifikation darstellt.

D.3 Segmentierung

Semantische Segmentierung Bezeichne \mathbf{y} die Annotation und $\hat{\mathbf{y}}$ die zugehörige Vorhersage in Form von Pixelmatrizen, so folgt die IoU gemäß

$$Q_{IoU} = \frac{|\mathbf{y} \cap \hat{\mathbf{y}}|}{|\mathbf{y} \cup \hat{\mathbf{y}}|}. \tag{D.5}$$

Teilweise wird diese Metrik auch Jaccard Index (JI) bezeichnet. Ähnlich dazu ergibt sich der DSC über

$$Q_{DSC} = \frac{2 |\mathbf{y} \cap \hat{\mathbf{y}}|}{|\mathbf{y}| + |\hat{\mathbf{y}}|}. \tag{D.6}$$

Es ist anzumerken, dass diese Metrik in der Literatur häufig als F1-Score bezeichnet wird [Mai23]. In Abbildung D.1 findet sich eine bildhafte Veranschaulichung der genannten Metriken.

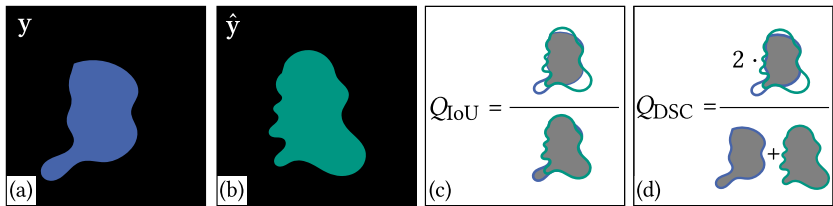


Abbildung D.1: Metriken semantische Segmentierung. Für eine exemplarische Annotation \mathbf{y} (a) und zugehörige Vorhersage $\hat{\mathbf{y}}$ (b) sind die Metriken Q_{IoU} (c) und Q_{DSC} (d) skizziert. Die zur Berechnung genutzten Flächen in (c-d) sind grau hinterlegt. Im Falle von Q_{IoU} wird der Quotient aus Schnittfläche und Vereinigung berechnet. Im Gegensatz dazu gibt Q_{DSC} das Verhältnis der zweifachen Schnittfläche zu den aufsummierten Einzelflächen an.

Nach Maier-Hein et al. [Mai23] sind die Q_{IoU} und Q_{DSC} sehr ähnlich, wodurch eine kombinierte Angabe der Metriken keinen Mehrwert liefert. Der DSC wird eher im Umfeld der medizinischen Bildverarbeitung verwendet, wohingegen die IoU vermehrt bei der Verarbeitung von Realweltdaten genutzt wird [Mai23].

Algorithmus 9 Aggregated Jaccard Index Q_{AJI} **Input:** $\mathbf{y}, \hat{\mathbf{y}}$ **Output:** Q_{AJI} $C \leftarrow 0, U \leftarrow 0$

▷ Initialisierung korrekte Pixel bzw. Vereinigung

for each ground truth instance \mathbf{y}_i **do** $j \leftarrow \arg \max_k (|\mathbf{y}_i \cap \hat{\mathbf{y}}_k| / |\hat{\mathbf{y}}_i \cup \hat{\mathbf{y}}_k|)$ Update $C \leftarrow C + |\mathbf{y}_i \cap \hat{\mathbf{y}}_j|, U \leftarrow U + |\mathbf{y}_i \cup \hat{\mathbf{y}}_j|$ Mark $\hat{\mathbf{y}}_j$ as *used***end for****for each** instance $\hat{\mathbf{y}}_l$ **do****if** $\hat{\mathbf{y}}_l$ is not *used* **then** $U \leftarrow U + |\hat{\mathbf{y}}_l|$ **end if****end for** $Q_{AJI} \leftarrow C/U$

Instanzsegmentierung Zur Quantifizierung der Güte im Rahmen der Instanzsegmentierung kann der Aggregated Jaccard Index (AJI) genutzt werden, welcher in Algorithmus 9 beschrieben wird und in [Kum20] zu finden ist.

Zu Beginn wird für jede Instanz $\mathbf{y}_i \in \mathbf{y}$ der Annotation \mathbf{y} nach einer Instanz $\hat{\mathbf{y}}_j \in \hat{\mathbf{y}}$ in der Vorhersage $\hat{\mathbf{y}}$ gesucht, die sich maximal überschneidet. Dementsprechend wird pro Instanz die Überschneidungsfläche und Vereinigungsfläche aktualisiert. Für alle vorhergesagten Instanzen, die keine Überschneidung zur Annotation haben, erfolgt die Addition deren Fläche zur Vereinigungsmenge, um dies negativ in der Metrik zu berücksichtigen.

D.4 Unsicherheit

Semantische Segmentierung Bezeichne $\{\hat{\mathbf{y}}^1, \dots, \hat{\mathbf{y}}^T\}$ eine Stichprobenmenge an $T \in \mathbb{N}$ Vorhersagen zu einem Bild \mathbf{x} , so folgt

$$Q_{iDAIS} = 1 - \frac{2}{T(T-1)} \sum_{i=1}^T \sum_{j=i+1}^T Q_{DSC}(\hat{\mathbf{y}}^i, \hat{\mathbf{y}}^j) \quad (\text{D.7})$$

Algorithmus 10 inverted Aggregated Jaccard Index Agreement in Sample Q_{iAAiS}

Input: $\hat{y}, \{\hat{y}^1, \dots, \hat{y}^T\}$
Output: Q_{iAAiS}

```

 $\mathcal{Q} \leftarrow \emptyset$ 
for each instance in overall prediction  $\hat{y}_i$  do
   $\mathcal{T} \leftarrow \emptyset$ 
  for each prediction  $\hat{y}^j$  do
     $k \leftarrow \arg \max_l (|\hat{y}_i \cap \hat{y}_l^j| / |\hat{y}_i \cup \hat{y}_l^j|)$ 
     $\mathcal{T}.append(Q_{JI}(\hat{y}_i, \hat{y}_k^j))$ 
  end for
   $\mathcal{Q}.append(1 - \text{average}(\mathcal{T}))$ 
end for
 $Q_{iAAiS} \leftarrow 1 - \text{average}(\mathcal{Q})$ 

```

zur Berechnung der bildweisen Unsicherheit gemäß den Ausführungen von Roy et al. [Roy19]. Exemplarisch findet sich zudem in Abbildung D.2 eine Veranschaulichung der Metrik.

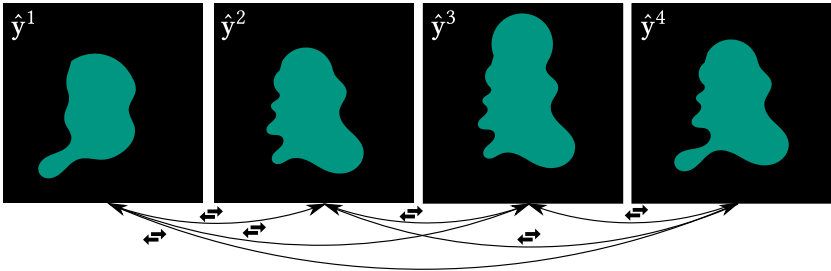


Abbildung D.2: Illustration Q_{iDAiS} . Für eine exemplarische Stichprobenmenge $\{\hat{y}^1, \hat{y}^2, \hat{y}^3, \hat{y}^4\}$ an $T = 4$ Vorhersagen ist die Berechnung der Unsicherheitsmetrik Q_{iDAiS} (kreuzweiser Vergleich) veranschaulicht [Sch22d].

Instanzsegmentierung Analog zu iDAiS soll die Berechnung des entwickelten Kriteriums für die Instanzsegmentierung, genannt inverted Aggregated Jaccard Index Agreement in Sample (iAAiS), anhand von Algorithmus 10 erläutert werden. Zu Beginn wird für jede Instanz \hat{y}_i der Gesamtvorhersage \hat{y} nach der bestmöglichen Überschneidung in den $T \in \mathbb{N}$ Vorhersagen des

Stichprobenumfangs $\{\hat{\mathbf{y}}^1, \dots, \hat{\mathbf{y}}^T\}$ bezüglich des Bilds \mathbf{x} gesucht. Im Nachgang wird der JI für die Instanz $\hat{\mathbf{y}}_k^j$ mit der größten Überlappung berechnet. Diese Metriken werden dann zum einen über alle T Vorhersagen gemittelt. Zum anderen wird danach über eine Umkehrung ($1 - \text{average}(\mathcal{J})$) das Maß für die Unsicherheit pro Instanz in die Liste \mathcal{Q} übernommen. Die Gesamtunsicherheit Q_{iAAIS} bezüglich aller Instanzen bildet sich aus dem Mittel aller Unsicherheiten der gegebenen Instanzen in Bild \mathbf{x} .

E Deep Neural Networks

E.1 Architekturen

Die nachfolgenden Abbildungen [E.1/E.2](#)¹ enthalten die zentralen Netzarchitekturen, auf die im Rahmen der Arbeit zurückgegriffen wird.

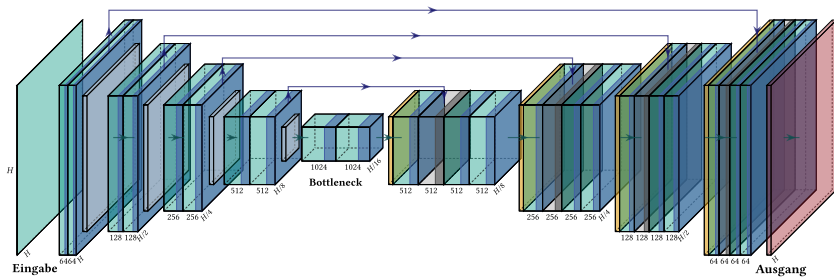


Abbildung E.1: U-Net Architektur. Das DNN besteht aus einem Encoder und Decoder, wobei diese Elemente über Überbrückungsverbindungen miteinander verknüpft sind. Generell ist die Architektur aus Basiselementen aufgebaut, die aus einer zweifachen Wiederholung von Faltung und Rectified Linear Unit (ReLU)-Aktivierungsfunktion bestehen. Im Encoder wird die Information durch Max-Pooling-Schichten verdichtet, wohingegen der Decoder mittels Upsampling-Schichten die Ausgabe in der Eingangsgröße erzeugt. Es erfolgt eine Verknüpfung der Merkmale aus dem Bottleneck (engste Stelle des DNN) und Informationen aus den Überbrückungsverbindungen. Eine Faltungsschicht am Ende des DNN generiert die problemspezifische Anzahl der jeweiligen Klassen am Ausgang. Oftmals finden hierbei die Sigmoid- oder Softmax-Funktionen Anwendung, um die Ausgabe entsprechend zu skalieren [Ron15, Iqb18].

¹ Die gezeigten Darstellungen basieren auf der Implementierung PlotNeuralNet [Iqb18].

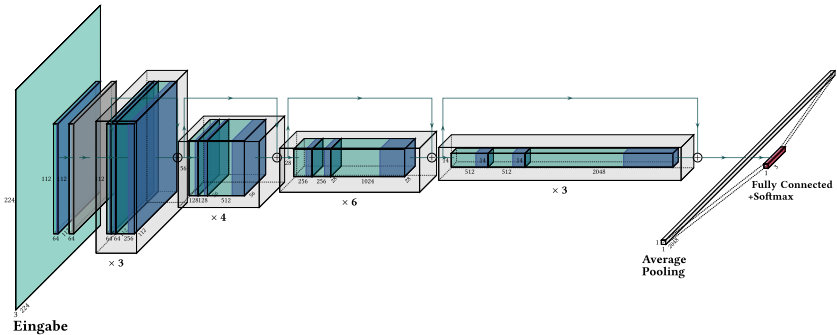


Abbildung E.2: ResNet50 Architektur. Das DNN besteht am Beginn aus einer Aneinanderreihung von Faltungsschicht, Batch-Normalisierung, ReLU-Funktion und Max-Pooling. Im Anschluss folgen ähnlich dazu die Residuen-Blöcke, mit einer Wiederholung des Basisaufbaus, jedoch lediglich Max-Pooling am Ende eines Blockes. Die Eingänge der Residuen-Blöcke werden zudem immer überbrückt weitergeleitet, um dem Verschwinden der Gradienten vorzubeugen. Am Ende wird über Mittelung der Merkmalschichten (Average Pooling) ein Merkmalsvektor erzeugt, der über vollständig verbundene Neuronen und Softmax-Funktion interpretiert wird, um Vorhersagen zu generieren [He16, Iqb18].

E.2 Standard-Konfigurationen

E.2.1 Optimierungsverfahren

Eine Beschreibung aller Elemente im Rahmen der verwendeten Optimierungsverfahren inklusive deren Parametrierung findet sich in Tabelle E.1.

Tabelle E.1: Details zur Optimierung.

Element	Parametrierung
Optimierungsalgorithmus	Adam ¹ [Kin15]
Lernrate	0.001
Koeffizienten Mittelung (Moment erster/zweiter Ordnung)	0.9/0.999
Lernraten-Reduktionsverfahren	ReduceLRonPlateau ²
Anzahl der Warte-Epochen	20
Faktor der Reduktion	0.5

E.2.2 Training

Erläuterung zu den Einzelheiten bezüglich des Trainings von DNNs sind in Tabelle E.2 gezeigt.

Tabelle E.2: Details zum Training.

Element	Parametrierung
Anzahl der Epochen	200
Batchgröße	4
Abbruch bei ausbleibender Verbesserung ¹	aktiv
Wartezeit	30 Epochen
Zugehörige Metrik zur Beurteilung	Validierungsgüte

E.2.3 Vorverarbeitungen

Nachfolgende Methoden (je nach Datensatz empirisch ausgewählt über Parameterstudien) werden zur Vorverarbeitung der Bilder verwendet:

- Min-Max-Normalisierung (optional: vorgeschaltete Normalisierung bezüglich 1. Perzentil/ 99. Perzentil im Falle von Ausreißern) oder
- z-Normalisierung (Mittelwert null, Standardabweichung eins).

E.2.4 Augmentierungen

Die Augmentierung der Daten erfolgt mittels der Bibliothek `Albumentations` [Bus20]. Folgende Transformationen² finden im Rahmen der Arbeit standardmäßig Anwendung:

- `VerticalFlip` (vertikale Drehung),
- `HorizontalFlip` (horizontale Drehung),
- `ShiftScaleRotate` (Translation, Skalierung und Rotation),

² https://albumentations.ai/docs/api_reference/augmentations/transforms bzw. https://albumentations.ai/docs/api_reference/augmentations/geometric

- GaussianBlur (Unschärfe durch Gauß-Filter) und
- RandomBrightnessContrast (zufällige Änderung von Helligkeit bzw. Kontrast).

E.2.5 Nachverarbeitungen

Zur Nachverarbeitung der Vorhersagen von DNNs wird nachfolgendes Vorgehen angewandt:

- Semantische Segmentierung - Argument des Maximums bei Mehrklassenproblemen oder Schwellwert von 0.5 im Falle der binären Segmentierung
- Instanzsegmentierung gemäß [Sch20a] - Schwellwert zur Bestimmung der Objekte: 0.4, Schwellwert zur Bestimmung der Objektgrenze: 0.09
- Seed Detektion - Schwellwert von 0.5

E.2.6 Zielfunktionen

Eine Übersicht bezüglich der verwendeten Zielfunktionen in Abhängigkeit der Anwendungsfälle findet sich in Tabelle E.3.

Tabelle E.3: Details zu den Zielfunktionen.

Anwendungsfall	Zielfunktion
Semantische Segmentierung	DiceLoss (1- Q_{DSC})
Instanzsegmentierung	SmoothL1Loss ¹ , ähnlich zu [Hub64]
Seed Detektion	DiceLoss (1- Q_{DSC})
Ähnlichkeitsschätzung	Zielfunktionen siehe Gleichung (C.5)
Annotationseffiziente Lernverfahren	Zielfunktionen aus Manuskripten siehe Abschnitt 2.4

F Schnellstart

Bedienungsanleitungen

F.1 KaIDA

Im Weiteren wird eine ausführliche Bedienungsanleitung für die schnelle Verwendung des Programms Karlsruhe Image Data Annotation Tool (KaIDA) bereitgestellt [Sch22e]. Dabei zeigt Abbildung F.1 Details zum Erstellen eines neuen Projekts. Ferner ist in Abbildung F.2 die Nutzung von KaIDA für ein Training bzw. die Anwendung von DNNs gezeigt. Außerdem präsentiert Abbildung F.3 eine Erklärung zu den Optionen im Rahmen des assistierten Annotierens.

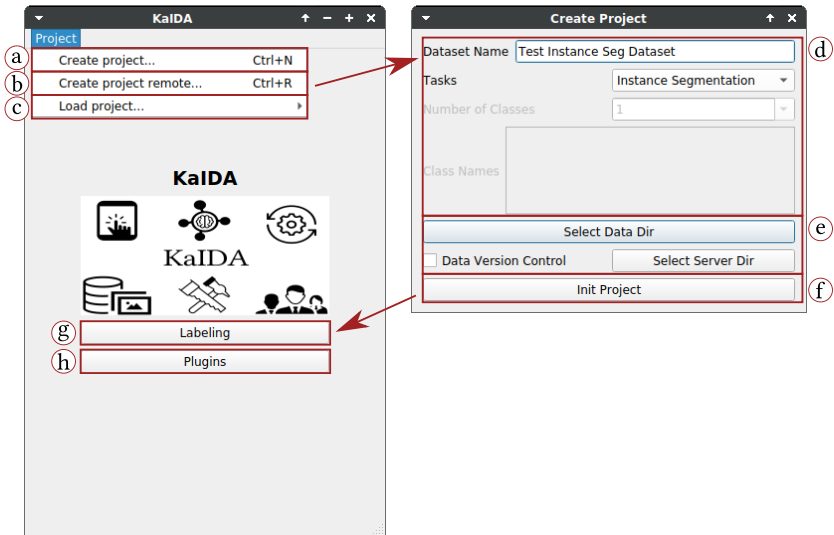


Abbildung F.1: Benutzeroberfläche zur Projekterstellung in KaIDA.

- (a) Erstellung eines neuen Projekts.
- (b) Erstellung und Import eines neuen Projekts, welches auf einem anderen Computer angelegt worden ist bzw. auf dem Datenserver abgelegt ist.
- (c) Laden eines bereits erstellten Projekts.
- (d) Eingabefenster zur Erstellung des Projekts, in dem Meta-Daten zum Projekt angegeben werden.
- (e) Auswahlfenster hinsichtlich des Speicherorts der Daten. Optional kann die Datensatzversionierung und die Sicherung der Daten auf einem Datenserver aktiviert werden.
- (f) Knopf zum finalen Erstellen des neuen Projekts.
- (g) Knopf zum Starten der Annotation.
- (h) Öffnen des Plugin-Menüs.

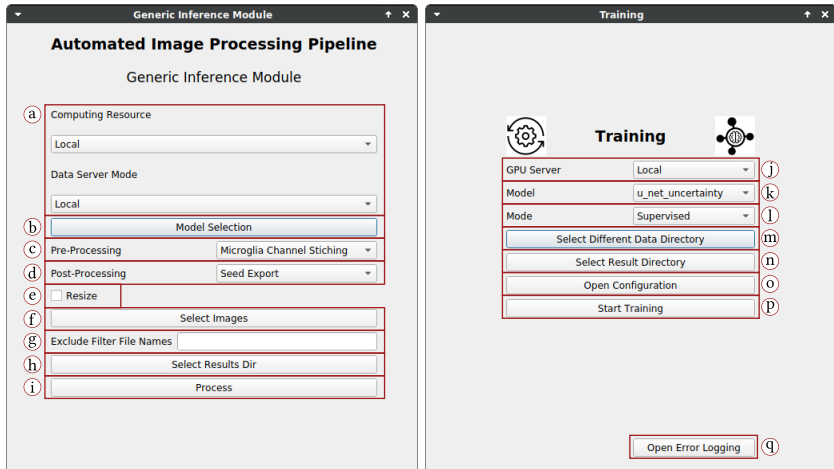


Abbildung F.2: Oberfläche für Training und Anwendung in KaIDA.

- Ⓐ Definition der Rechenmaschine bzw. des Datenservers.
- Ⓑ Auswahl des Modells.
- Ⓒ Optionale Auswahl einer speziellen Vorverarbeitung der Bilder im Rahmen der Anwendung.
- Ⓓ Auswahl der Methode zur Nachverarbeitung der Vorhersagen (z. B. Visualisierungen).
- Ⓔ Option zur Anpassung der Bildgröße.
- Ⓕ Menü zur Auswahl der Bilder.
- Ⓖ Möglichkeit zum Ausschluss spezieller Bilder (z. B. alle Bilder des Hellfeldkanals, die einen speziellen Namen tragen).
- Ⓗ Auswahl des Zielordners zum Speichern der Resultate.
- Ⓘ Start der Anwendung.
- Ⓙ Auswahl der Hardware zum Training.
- Ⓚ Selektion des gewünschten zu trainierenden Modells.

- ① Auswahl des Lernverfahrens.
- Ⓜ Option, um ein abweichendes Datensatzverzeichnis zu wählen.
- Ⓝ Festlegung des Speicherorts von Modellen.
- Ⓞ Definition der Trainingsparameter.
- Ⓟ Start des Trainings.
- Ⓠ Möglichkeiten, um das gesamte Logging während des Trainings zu betrachten.

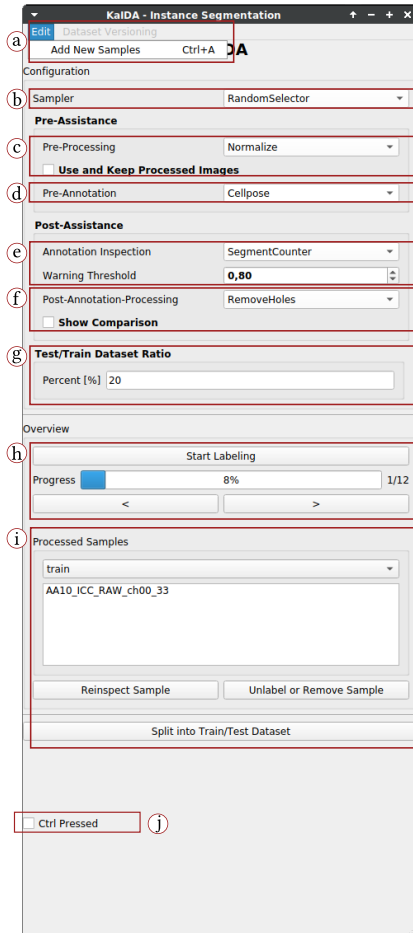


Abbildung F.3: Benutzeroberfläche für das assistierte Annotieren in KaIDA.

- Ⓐ Erweiterung des Datensatzes mit neuen Bildern.
- Ⓑ Auswahl der Selektionsstrategie.
- Ⓒ Menü zum Festlegen der Bildvorverarbeitung. Optional können die prozessierten Bilder anstelle der Rohdaten in den Datensatz übernommen werden.

- Ⓓ Auswahl einer möglichen Methode zur Pre-Annotation.
- Ⓔ Aktivierung der Annotation-Inspektion inklusive Definition der Warnschwelle.
- Ⓕ Auswahl einer automatisierten Nachverarbeitung der Annotationen. Optional können sich Nutzer die Änderungen anzeigen lassen.
- Ⓖ Definition des Verhältnisses von Trainings- und Test-Datensatz.
- Ⓗ Feedback zum Fortschritt, Wahlkosten für den Start der Annotationen bzw. der Auswahl von Bildern (vor/zurück). Beim Drücken der Starttaste wird dem Nutzer die zugehörige Eingabeoberfläche (siehe Abbildung 4.2) angezeigt, um die eigentliche Annotation durchzuführen.
- Ⓘ Übersicht der Datensätze inklusive Optionen zum Löschen der Annotationen, der erneuten Betrachtung eines annotierten Bildes bzw. dem Aufteilen in Trainings- und Test-Datensatz.
- Ⓙ Fiktives Drücken der Steuerungstaste (relevant für Tablet-Modus ohne Tastatur)

F.2 AI²Seg

In Abbildung F.4 ist das Graphical User Interface (GUI) zur Nutzung von AI-based Annotation Inspection for Segmentation (AI²Seg) gezeigt [Sch23c].

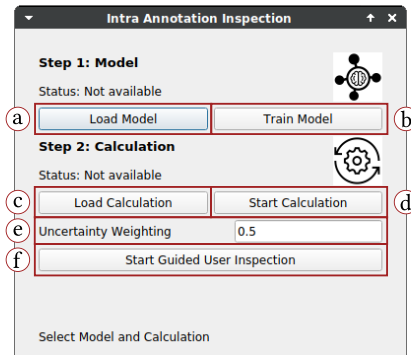


Abbildung F.4: Benutzeroberfläche für die Inspektion eines Datensatzes in AI²Seg.

- Ⓐ Laden eines trainierten Modells zur Inspektion.
- Ⓑ Training eines neuen Modells für die Inspektion.
- Ⓒ Laden von bereits berechneten Inspektionen des Datensatzes.
- Ⓓ Berechnung der Inspektionen für die Bilder des Datensatzes.
- Ⓔ Festlegung des Parameters zum Einbezug der Unsicherheit.
- Ⓕ Start der assistierten Datensatzinspektion. Fenster aus Abbildung 4.4 öffnet sich.

F.3 Grid Scener

Eine ausführliche Beschreibung der Oberfläche zur Software Grid Scener findet sich im Überblick in Abbildung F.5 [Sch21b]. Details zur semi-automatisierten Gittererkennung sind in Abbildung F.6 dargestellt.

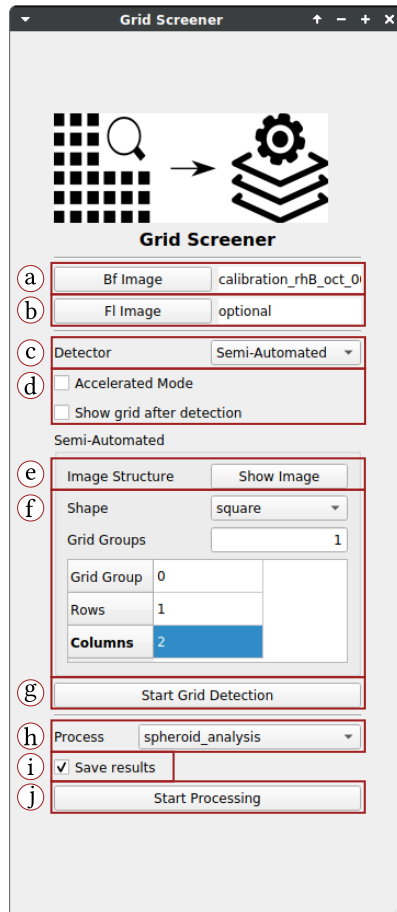


Abbildung F.5: Benutzeroberfläche Grid Scener.

- Ⓐ Auswahl eines Hellfeldbildes.
- Ⓑ Optionale Auswahl eines Fluoreszenzbildes.
- Ⓒ Konfiguration des Detektors (automatisiert/semi-automatisiert).
- Ⓓ Zusatzeinstellungen hinsichtlich der Detektion.
- Ⓔ Möglichkeit zur Anzeige des Bildes, um die Parameter auszulesen, die im Anschluss eingegeben werden müssen.
- Ⓕ Eingabemaske der notwendigen Parameter im Falle der semi-automatisierten Erkennung.
- Ⓖ Start der Gittererkennung.
- Ⓗ Menü zur Auswahl der Bildverarbeitungsmethode pro Element.
- Ⓘ Auswahl zur Speicherung.
- ⓵ Knopf zum Starten der elementweisen Auswertung.

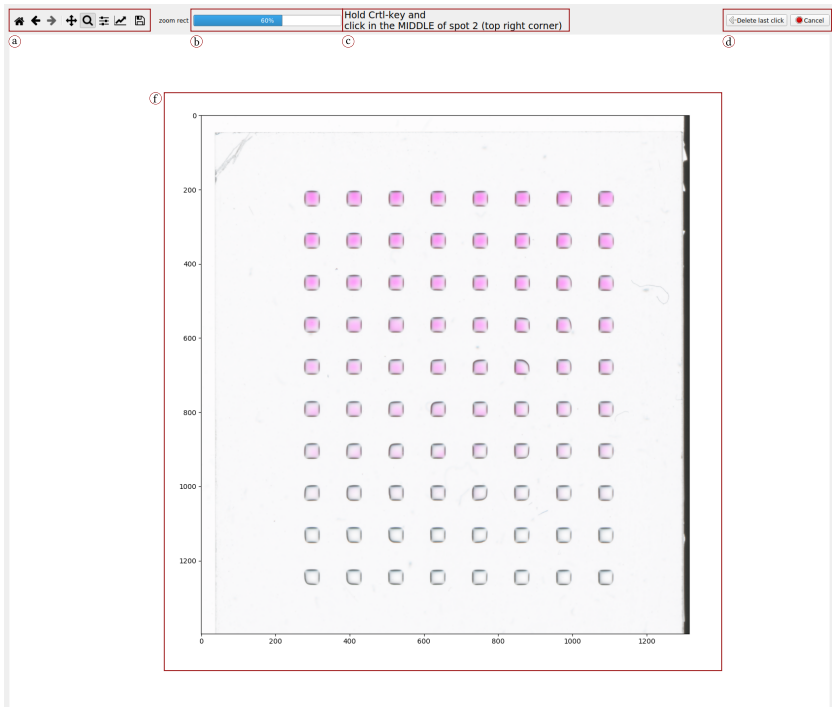
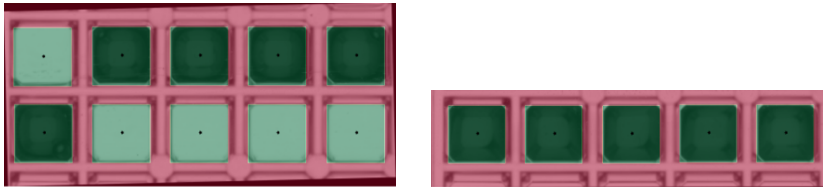


Abbildung F.6: Eingabefenster für die semi-automatisierte Gitterschätzung im Werkzeug Grid Screener.

- Ⓐ Einstellungsmöglichkeiten hinsichtlich der Darstellung.
- Ⓑ Fortschrittsanzeige.
- Ⓒ Anweisungen für Nutzer bezüglich der nächsten erforderlichen Eingabe.
- Ⓓ Möglichkeiten zum Rückgängigmachen von Eingaben bzw. Abbruch der Detektion.

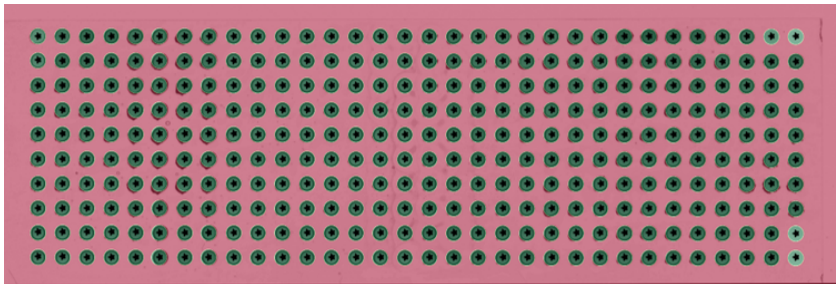
G Weitere Ergebnisse der Gitterschätzung

Ergänzend zum Hauptteil (vgl. Abschnitt 5.3.1) finden sich in Abbildung G.1 die Resultate der Gitterschätzung für Wellplates sowie für DMA-Experimente mit kreisförmigen (vgl. Abbildung G.2) bzw. quadratischen (vgl. Abbildung G.3) Elementen.



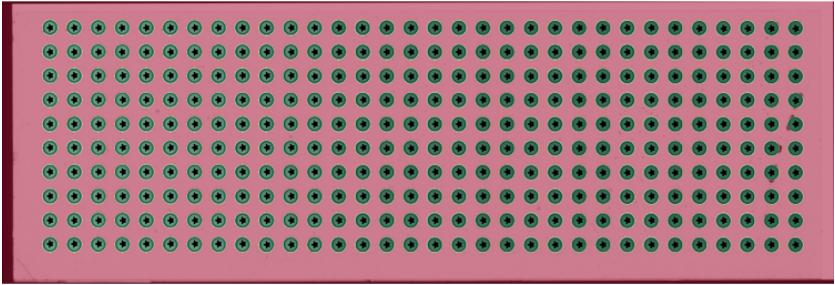
(a) Erstes Experiment ($Q_{\Delta p} = 0.012, Q_{\Delta d} = 0.015$). (b) Zweites Experiment ($Q_{\Delta p} = 0.032, Q_{\Delta d} = 0.037$).

Abbildung G.1: Gitterschätzung Wellplate. Vorhersagen sind für Wellplates dargestellt. Grüne Overlays zeigen Elemente bzw. schwarze Kreuze deren Mittelpunkte [Sch21b].



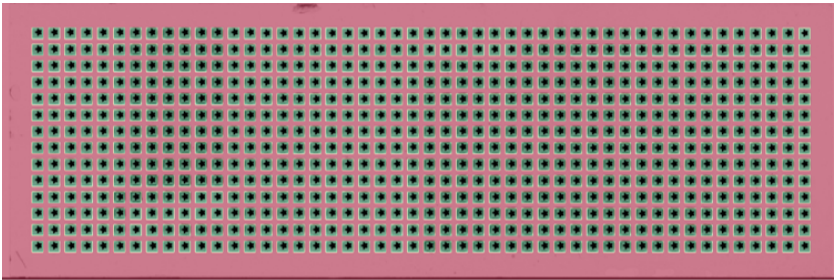
(a) Erstes Experiment ($Q_{\Delta p} = 0.047, Q_{\Delta d} = 0.029$).

Abbildung G.2: Gitterschätzung DMA - Kreis. Die vorhergesagten Gitter sind für DMA-Experimente mit Kreisform gezeigt. Die Visualisierung ist analog zu Abbildung G.1 [Sch21b].

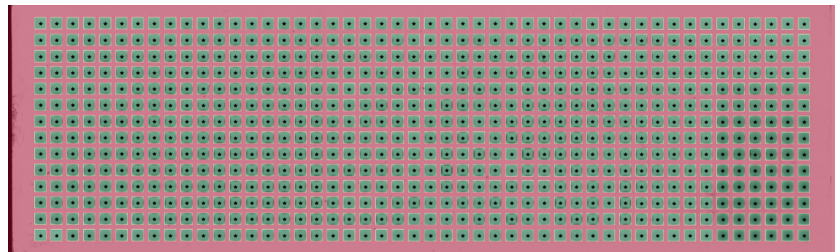


(b) Zweites Experiment ($Q_{\Delta p} = 0.030$, $Q_{\Delta d} = 0.019$).

Abbildung G.2: Gitterschätzung DMA - Kreis. Fortsetzung.



(a) Erstes Experiment ($Q_{\Delta p} = 0.053$, $Q_{\Delta d} = 0.007$).



(b) Zweites Experiment ($Q_{\Delta p} = 0.041$, $Q_{\Delta d} = 0.029$).

Abbildung G.3: Gitterschätzung DMA - Quadrat. Es wird die Vorhersage des Gitters für DMA-Experimente mit Quadraten illustriert. Die Darstellung entspricht Abbildung G.1 [Sch21b].

H Softwarepakete

Eine ausführliche Darstellung der verwendeten python-Softwarepaketen inklusive deren Version ist in Tabelle H.1 dargestellt. Im Rahmen der Arbeit ist conda¹ zur Paketverwaltung eingesetzt worden.

Tabelle H.1: Details zu den verwendeten python-Softwarepaketen².

Softwarepaket	Version
albumentations	0.5.2
cellpose[all]	0.7.2
datatoolkit	11.3.1
dvc	2.9.3
mxnet	2021.4.0
numpy	1.21.5
opencv-python	4.5.5.62
pandas	1.3.4
paramiko	2.8.1
pyqt5	5.12.3
pytorch	1.10.0
pytorch-lightning	1.4.9
torchvision	0.11.0
tables	3.6.1
ttach	0.0.3
tifffile	2020.10.1
scikit-image	0.18.3
scikit-learn	1.0.1
scipy	1.7.1
setuptools	59.5.0
slidingwindow	0.0.14
umap-learn	0.5.1
wandb	0.12.15

¹ <https://docs.conda.io/en/latest/miniconda.html>

I Prototypischer Bildverarbeitungsserver

Eine Beschreibung des prototypischen Bildverarbeitungservers ist in Tabelle I.1 aufgeführt.

Tabelle I.1: Details zum prototypischen Bildverarbeitungsserver. Die Elemente des eingerichteten prototypischen Bildverarbeitungservers werden im Detail dargestellt [Sch22e].

Elemente	Beschreibung
Server	Betriebssystem: Windows Server 2019
CPU	Intel Xeon 4210R
GPU	NVIDIA Quadro RTX 4000 (8 GB)
Arbeitsspeicher	32 GB
Lokaler Speicher	512 GB SSD
Software	miniconda ¹
Update	Git Versionskontrolle [Tor05]
Datenserver	LSDF [Jej12]

