

# **Prozessautomatisierung in der additiven Fertigung mit Robotik**

Zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften  
(Dr.-Ing.)

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

genehmigte  
DISSERTATION

von

**M.Sc. Pascal Becker**

Tag der mündlichen Prüfung: 11. Juli 2023  
Referent: Prof. Dr.-Ing. Rüdiger Dillmann  
Korreferentin: Prof. Dr.-Ing. Gisela Lanza

# Affirmation

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Graben-Neudorf, Mai 2023

M.Sc. Pascal Becker

# Danksagung

**Auf den Schultern von Riesen** Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Abteilung Interaktive Diagnose- und Servicesysteme am FZI Forschungszentrum Informatik in Karlsruhe. Die Zeit war geprägt von vielen abwechslungsreichen und spannenden Themen, Diskussionen und Projekten. Auch wenn sie nicht immer direkt im Zusammenhang mit dieser Arbeit standen, so brachten mich dennoch immer wieder Teile davon weiter. Hierbei ist vor allem dem Team rund um Dr.-Ing. Arne Rönnau und Georg Heppner zu danken. Trotz harten Projektdeadlines, nächtlichen Demo-Vorbereitungen und viel zu wenig Platz für all die Robotikdemonstratoren möchte ich diese lehrreiche Zeit in unglaublicher Arbeitsatmosphäre nicht missen. Dazu hat auch das restliche FZI-Team beigetragen.

Herrn Prof. Dillmann danke ich besonders für die wissenschaftliche Betreuung dieser Arbeit. In etlichen Diskussionen, im Rahmen der Betreuung von Abschlussarbeiten oder auf Klausurtagungen half er mir das Thema weiter zu verfeinern, und den roten Faden nicht zu verlieren. Des Weiteren möchte ich Frau Prof. Gisela Lanza vom Institut für Produktionstechnik am KIT für die Übernahme des Koreferats danken. Ohne eine Reihe an herausragenden Studenten wäre diese Arbeit aber auch nicht möglich gewesen. Hier möchte ich Iain Ahrens, Moritz Bartens, Cassandra Dahms, Christian Eichmann, Etienne Henger, Joshua Gebert, Jonas Keller, Anastasia Maklashevskikh, Christian Roth, Dario Schmid, David Schneider, Niklas Spielbauer und Robert Wilbrandt meinen Dank aussprechen. Sei es nur durch die Wartung und das Kümmern um die 3D-Drucker, sowie die ganze Infrastruktur darum, oder im Rahmen von Projekten, Bachelor- oder Masterarbeiten. Ihr habt alle in der ein oder anderen Form an dieser Arbeit mitgewirkt.

Mein größter Dank gilt aber meiner Familie und insbesondere meinen Eltern Heike und Stefan Becker, die mir diesen Weg ermöglicht haben. Angefangen beim Kosmos Elektronik-Kasten, Pausenbrot für die Schulpause, den ersten PC über fleißiges Üben und Mitleiden für Klassenarbeiten bis hin zur vollen Unterstützung beim Studium in Karlsruhe und Espoo. Sowie meiner Schwester Hannah für die viele Ablenkung bei hunderten, wenn nicht tausenden Runden Siedler von Catan.

Gewidmet ist diese Arbeit meiner Freundin Mareike Trappen. Du hattest immer ein offenes Ohr und konntest mich mit deiner wissenschaftlichen und technischen Neugier und Expertise immer wieder aufs Neue motivieren. Ohne dich hätte ich nicht das Durchhaltevermögen gehabt diese Arbeit fertigzustellen.

Minä rakastan sinua!

# Abstract (English Version)

Additive manufacturing is currently revolutionizing the limits of manufacturability. This novel process makes it possible to manufacture complex component geometries on-demand, locally and specifically to the needs at hand. Another advantage is the possibility to implement geometries that provide the required stability while at the same time using an optimal amount of material. Especially in view of the increasing individualization of products additive manufacturing is a key technology. In the present work, various approaches are developed and investigated that will help to make additive manufacturing more automated in the future.

Therefore, the printing process has to be monitored and the final print result has to be post-processed. These include in-situ observation of the print with the help of optical and acoustic sensors. The sensor values of these are evaluated both with classical image processing, but also with neural networks. Making it possible to detect errors at an early stage in the printing process.

For the post-processing several steps are necessary. While the object is printed, the exact position of the objects on the build plate can be calculated and an unloading sequence can be determined. The order is important to remove the printed objects one after the other without damage the objects or the printer itself. Using specifically designed gripper jaws, the printed objects are removed from the build plate by a robot without harming it.

After the unloading is completed, the printer can start a new job and the post-processing of the objects can be performed. In the vast majority of cases, the objects are printed with support structures, which are needed to support overhanging elements so that they remain stable in shape during the fabrication process. For this purpose, based on the CAD model file and the print data, those areas on the printed object are determined that were exclusively printed in order to obtain a high-quality print result. Two different approaches are implemented and investigated. In one approach the post-processing tool is mounted at the robot tool center point and the object is fixed and in the other approach it is vice versa. In both cases, path planning is used to plan a path that allows the structures to be removed without damaging the part, thus allowing the object to be used directly.

# Kurzfassung (German Version)

Aktuell revolutioniert die additive Fertigung die Grenzen der Fertigbarkeit. Durch dieses neuartige Verfahren ist es möglich, komplexe Bauteilgeometrien on-demand, lokal und speziell auf die vorliegenden Bedürfnisse angepasst zu fertigen. Ein weiterer Vorteil ist die Möglichkeit Geometrien herzustellen, die die benötigte Stabilität bei gleichzeitig optimalem Materialeinsatz bietet. Gerade in Anbetracht der steigenden Individualisierung von Produkten ist die additive Fertigung eine Schlüsseltechnologie. In der vorliegenden Arbeit werden verschiedene Verfahren entwickelt und untersucht, die dazu beitragen, dass die additive Fertigung in Zukunft automatisierter ablaufen kann.

Dazu zählen unter anderem die in-situ Beobachtung des Drucks sowie die Nachbearbeitung nach Fertigstellung des Objekts. Bei der Drucküberwachung kommen sowohl optische als auch akustische Sensoren zum Einsatz. Die Sensorwerte werden sowohl mit klassischer Bildverarbeitung, aber auch mit neuronalen Netzen ausgewertet. Somit wird es möglich Fehler frühzeitig im Druckprozess zu erkennen.

Die Druckdauer wird genutzt, um die exakte Position der Bauteile auf dem Druckbett zu berechnen, sowie eine Entnahmereihenfolge zu bestimmen, um die Bauteile nacheinander zu entnehmen. Hierbei dürfen weder die gedruckten Objekte noch der Drucker selbst beschädigt werden. Mit Hilfe von speziell entworfenen Greiferbacken werden die Objekte von einem Roboter aus dem Drucker von der Druckplatte genommen.

Nach der abgeschlossenen Entnahme kann der Drucker einen neuen Auftrag starten und die Nachbearbeitung der Objekte kann durchgeführt werden. In den allermeisten Fällen werden die Objekte mit Stützstrukturen gedruckt. Diese werden dazu benötigt überhängende Elemente zu stützen, damit diese während des Prozesses stabil in Form bleiben. Hierfür wird auf Basis der CAD- und der Druckdaten diejenigen Stellen am gedruckten Objekt ermittelt, die ausschließlich gedruckt wurden, um ein hochwertiges Druckergebnis zu erhalten. Zwei unterschiedliche Ansätze zur Entfernung der Stützstruktur werden umgesetzt und untersucht. Einmal befindet sich das Werkzeug am Roboter Endeffektor und das Objekt ist fest montiert und einmal ist es genau andersrum. In beiden Fällen wird mit Hilfe einer Pfadplanung ein Pfad geplant, der es ermöglicht die Strukturen ohne Beschädigung des Bauteils zu entfernen und so einen direkten Einsatz des Objektes ermöglicht.

# Inhaltsverzeichnis

<b>1. Einleitung und Überblick</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Forschungsfragen . . . . .	5
1.3. Einordnung und wissenschaftlicher Beitrag . . . . .	6
1.4. Gliederung der Abhandlung . . . . .	8
<b>2. Stand der Forschung</b>	<b>9</b>
2.1. Additive Fertigung . . . . .	9
2.1.1. Fused Filament Fabrication (FFF) . . . . .	10
2.1.2. Laser Powder Bed Fusion (L-PBF) . . . . .	11
2.2. Robotik . . . . .	12
2.3. In-Situ Fehlererkennung in der additiven Fertigung . . . . .	13
2.3.1. Optische Überwachung des Druckprozesses . . . . .	13
2.3.2. Akustische Überwachung des Druckprozesses . . . . .	19
2.4. Automatisierte Entnahme aus einem 3D-Drucker . . . . .	23
2.5. Automatisierte Nachbearbeitung gedruckter Objekte . . . . .	26
<b>3. In-situ Fehlererkennung im Druckprozess</b>	<b>28</b>
3.1. Optische Fehlererkennung mit klassischer Bildverarbeitung . . . . .	29
3.1.1. Vorverarbeitung der Bilddaten . . . . .	29
3.1.2. Erkennung der verschiedenen Fehlertypen . . . . .	31
3.1.3. Ergebnis und Evaluation . . . . .	36
3.2. Optische Fehlererkennung und Klassifikation mit Deep Learning . . . . .	41
3.2.1. Fehlerkategorien . . . . .	44
3.2.2. Erstellung, Klassifikation und Exploration des Datensatzes . . . . .	45
3.2.3. Umsetzung des Konzepts . . . . .	47
3.2.4. Topologie des modularen neuronalen Netzes . . . . .	51
3.2.5. Verbesserung des Datensatzes und der Netzarchitekturen . . . . .	56
3.2.6. Bewertung und Evaluation des modularen Ansatzes . . . . .	66
3.3. Akustische Fehlererkennung mit Hilfe von Deep Learning . . . . .	73
3.3.1. Datenaugmentation für erweiterten Datensatz . . . . .	75
3.3.2. Experimente und Ergebnisse . . . . .	79
3.3.3. Transfer des Verfahrens auf den PBF Druck . . . . .	82
3.3.4. Experimente und Ergebnisse . . . . .	88
3.4. Fazit . . . . .	92

<b>4. Automatisierte Entnahme gedruckter Objekte</b>	<b>94</b>
4.1. Greifen der gedruckten Objekte . . . . .	95
4.1.1. Analyse von 3D-gedruckten Objekten . . . . .	95
4.1.2. Entwurf der Greiffinger . . . . .	96
4.1.3. Strategien zum Lösen des Objekts von der Druckplatte . . . . .	100
4.1.4. Experimente und Evaluation zum Greiferdesign . . . . .	100
4.1.5. Analyse der Ablösekraft . . . . .	100
4.1.6. G-Code Analyse zur Positionsbestimmung . . . . .	102
4.1.7. Belastungstest . . . . .	103
4.2. Sequentielle Entnahme mehrerer gedruckter Objekte von der Druckplatte . .	105
4.2.1. Bestimmung der Objektanzahl, -position und Entnahmereihenfolge anhand des G-Codes . . . . .	107
4.2.2. Bestimmung der Anzahl der Objekte . . . . .	107
4.2.3. Bestimmung der Greifmöglichkeit . . . . .	110
4.2.4. Erstellung der Objekthypothese . . . . .	112
4.2.5. Verifizierung der Objekthypothese auf der Grundlage einer Punktwolke	113
4.2.6. Prüfung der Erfassbarkeit und Entfernung der Objekte . . . . .	113
4.2.7. Experimente und Ergebnisse . . . . .	114
4.2.8. Visuelle Datenanalyse . . . . .	115
4.2.9. Test des Gesamtsystems . . . . .	118
4.3. Fazit . . . . .	118
<b>5. Post-Processing der gedruckten Bauteile</b>	<b>120</b>
5.1. Nachbearbeitungswerkzeug am Roboter TCP . . . . .	120
5.1.1. Berechnung der Stützstrukturen . . . . .	123
5.1.2. Experimente und Evaluierung . . . . .	126
5.2. Gedrucktes Objekt am Roboter TCP . . . . .	131
5.2.1. Generierung des Werkzeugpfades . . . . .	132
5.2.2. Berechnung der Robotertrajektorie . . . . .	134
5.2.3. Ergebnisse und Evaluierung . . . . .	134
5.2.4. Ausführung der Trajektorie zur Nachbearbeitung . . . . .	134
5.2.5. Fräsgenauigkeit . . . . .	136
5.2.6. Entfernen der Stützstruktur . . . . .	137
5.2.7. Fazit . . . . .	140
<b>6. Zusammenfassung und Ausblick</b>	<b>141</b>
6.1. Zusammenfassung und Beitrag . . . . .	141
6.2. Diskussion . . . . .	143
6.3. Ausblick . . . . .	144

<b>Anhang</b>	<b>146</b>
<b>A. Appendix</b>	<b>147</b>
A.1. Verwendete Hardware und Software . . . . .	147
A.1.1. 3D-Drucker Prusa MK3S+ . . . . .	147
A.1.2. 3D-Drucker Raise3D N2 Plus . . . . .	148
A.1.3. Intel RealSense . . . . .	148
A.1.4. the t.bone SC 450 . . . . .	150
A.1.5. Universal Robots Roboter . . . . .	151
A.1.6. Robot Operating System (ROS) . . . . .	152
A.2. Design der Greiferhardware . . . . .	152
<b>Glossar</b>	<b>155</b>
<b>Akronyme</b>	<b>157</b>



# 1. Einleitung und Überblick

Sowohl die additive Fertigung, als auch die Robotik, sind gerade dabei die industrielle Produktion von Gütern aller Art zu revolutionieren. Nachdem mit Hilfe klassischer Industrierobotik viele einfach zu automatisierende Handhabungsaufgaben zufriedenstellend gelöst wurden, sind heutige Ansätze in der Lage dynamisch auf Messdaten zu reagieren und ihre Aufgabenbearbeitung entsprechend flexibel anzupassen. Die additive Fertigung wiederum ermöglicht das Produzieren von Geometrien, die bisher nur schwer oder unmöglich zu fertigen waren. Vor allem bei kleiner Stückzahl und hoher Bauteilkomplexität spielt die additive Fertigung ihre Vorteile gegenüber herkömmlichen Produktionsverfahren aus.

## 1.1. Motivation

Die additive Fertigung, umgangssprachlich auch 3D-Druck genannt, hat in den letzten Jahren einige bahnbrechende Entwicklungen hervorgebracht. So wurde 2019 an der Universität Tel-Aviv ein menschliches Herz in Miniaturgröße aus Stammzellen in einem 3D-Drucker hergestellt [1]. Bisher setzten vor allem technologieführende Industriezweige, zum Beispiel die Luft- und Raumfahrt [2] oder der Motorsport [3], die verschiedenen 3D-Druckverfahren ein. Der Fokus liegt dabei vor allem auf Bauteilen deren Geometrie eine optimale Kraftübertragung bei gleichzeitig minimalstem Materialeinsatz sicherstellt.

Die kontinuierliche Verbesserung der Druckverfahren und -materialien erlaubt es heutzutage jedoch auch kleinen und mittleren Unternehmen bis hin zu Privatpersonen die additive Fertigung kostengünstig und effizient für ihre Anwendungen einzusetzen. Neben diversen Polymeren und Metallen, gibt es mittlerweile ein weitreichendes Sortiment an druckbaren Materialien. So zum Beispiel Glas, Keramik, flexible, transparente oder funktionale Materialien, die aufgrund zum Beispiel bei einer Temperaturänderung ihre Eigenschaften ändern. Des Weiteren gibt es Unternehmen, die Gebäude innerhalb von wenigen Stunden drucken [4] oder Sportartikelhersteller wie Adidas, die Schuhsohlen für ihr futuristisches Schuhmodell Futurecraft 4D [5] für den Massenmarkt drucken.

Neben Campbell [6] und Berman [7] sind noch viele weitere Experten der Meinung, dass der 3D-Druck mit der entscheidende Teil der nächsten Industriellen Revolution sein wird: Angefangen beim privaten Druck eines Ersatzteils, Rapid Prototyping in den Entwicklungsabteilungen, angepassten Herzklappen für Patienten im Gesundheitsbereich [8] bis hin zum Druck von ganzen Gebäuden [9].

## 1. Einleitung und Überblick

Neben den konkreten Produktvorteilen bringt die additive Fertigung auch allgemeine Vorteile mit sich [10]. Die Reduktion der eingesetzten Materialien macht das Produkt nachhaltiger im Vergleich zu einem konventionell gefertigtem. Außerdem können Produkte additiv on-demand gedruckt werden, das heißt nur dann, wenn sie wirklich benötigt oder bestellt werden. Das bedeutet preiswertere Produkte und auch eine Reduzierung des Ressourcenverbrauchs.

Neue Produkte weisen mittlerweile eine große Anzahl an Individualisierungsmöglichkeiten auf. Angefangen bei Fahrradsatteln [11], welche auf die Körpergeometrie des Fahrers angepasst gedruckt werden, über die Variantenvielfalt in der Automobilbranche bis hin zu auf Körperproportionen angepasste Gegenstände, wie Brillengestelle [12], ist die Anzahl der wählbaren Möglichkeiten immens. Bei bisherigen Produktionsverfahren wie dem Druckguss wird pro Variante eine eigene Gussform benötigt. Eine steigende Variantenvielfalt bis hin zur persönlichen Individualisierung führt somit zu einer nahezu unbegrenzten Anzahl an benötigten Formen, was wirtschaftlich nicht attraktiv ist. Genau diese Lücke schließt die additive Fertigung, die aktuell die Grenzen der Fertigbarkeit, sowie die Geschäftsmodelle in der industriellen Produktion revolutioniert [13]. Mit vorliegenden CAD-Daten können komplexe Bauteilgeometrien on-demand, lokal und speziell auf die vorliegenden Bedürfnisse gefertigt werden. Hierfür ist nur eine passende Maschine, sowie das entsprechende Material in Filament-, Pulver- oder Resin-Form benötigt.

Im Rahmen dieser Arbeit wird der Fokus vor allem auf das Fused Filament Fabrication (FFF), auch Schichtschmelzverfahren genannt, gelegt. Dieses Verfahren ist sehr weit verbreitet, da die Drucker zum einen preiswert, zum anderen aber auch einfach bedienbar und schnell sind. Da im Bereich der Hobbynutzer dieses Druckverfahren sehr weit verbreitet ist, gibt es die benötigte Software zum Ansteuern des Druckers (Marlin, Klipper) und Erstellen der Druckdaten (Slic3r, Cura, PrusaSlicer, etc.) öffentlich zugänglich in Form von Open Source Software. Das ist leider nicht bei allen Druckerherstellern und -verfahren so, so dass hier viele proprietären Systeme und Dateiformate zum Einsatz kommen. Deren Parameter und Einstellungen sind perfekt auf den jeweiligen Drucker abgestimmt, schränken jedoch die Möglichkeit der Anpassung und Erweiterung im Rahmen einer Forschungsarbeit stark ein.

Bevor mit der additiven Fertigung Produkte jedoch einfach und in größerer Stückzahl hergestellt werden können, gibt es noch Herausforderungen zu bewältigen [14, 15]. Die Anzahl der benötigten manuellen Arbeitsschritte ist bisher noch sehr hoch im Vergleich zu z.B. subtraktiven Verfahren wie z.B. dem Fräsen.

Neben einer Vorbereitung des Druckers muss der Prozess regelmäßig auf Fehler überprüft werden. Hier kann es z.B. durch Luft einschüsse, thermischen Verzug oder mechanischen Einflüssen zu Fehlern kommen, sodass das gedruckte Objekt für eine spätere Verwendung nicht mehr einsetzbar ist. Ein Fehler im Druckprozess schadet unter Umständen nicht nur dem gewünschten Objekt, sondern kann auch die Hardware des Druckers selbst schädigen. Löst sich das aktuell im Druck befindliche Objekt vom Druckbett, so kann es dazu führen, dass sich um die Druckdüse, den Extruder, Druckmaterial sammelt und zu Schäden am Drucker selbst führt. Eine in-situ Fehlererkennung gibt es im Allgemeinen nicht und findet bisher nur manuell statt.

## 1. Einleitung und Überblick

Je früher ein Fehler im Prozess erkannt wird, desto eher kann darauf reagiert werden. Dies umfasst z.B. die Anpassung von Parametern, wie Temperatur und Geschwindigkeit, oder auch die Entscheidung, das der Druck neu gestartet werden muss. Ein fehlgeschlagener Druck hat nicht nur die Maschine unwirtschaftlich lange belegt, sondern das Objekt ist hinterher üblicherweise auch Ausschuss.

Nach erfolgreichem Druck muss der Drucker zuerst entladen werden, bevor der nächste Auftrag gestartet werden kann. Hierfür sind je nach Druckverfahren verschiedene Schritte benötigt. Durch die verschiedenen Druckmaterialien wie Filament, Pulver oder Resin sind hierbei auch verschiedene Maßnahmen einzuhalten. Das fertiggestellte Objekt wird abschließend einer aufwändigen Nachbearbeitung unterzogen, um so die verfahrenstechnisch bedingten Hilfsstrukturen, wie zum Beispiel Support- und Stützstrukturen, zu entfernen. Beim Pulverdruck muss das überschüssige Pulver entfernt werden, was z.B. mit einer Saugeinrichtung durchgeführt werden kann. Beim Resindruck gilt es das Objekt erst z.B. mit Isopropanol zu reinigen, dann unter UV-Licht auszuhärten, um anschließend die Stützstrukturen händisch zu entfernen. Erst danach ist das Objekt einsatzbereit.

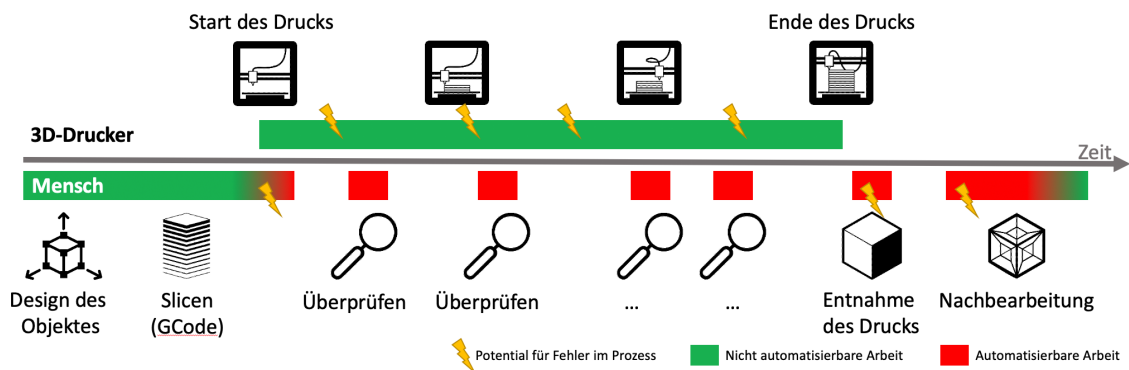


Abbildung 1.1.: Übersicht über die Prozesskette in der additiven Fertigung, sowie deren Arbeitsschritte und Möglichkeit zur Automatisierung.

In Abbildung 1.1 ist eine Übersicht über die im Druckprozess benötigten Arbeitsschritte dargestellt sowie mögliches Fehlerpotential und ob der jeweilige Prozessschritt prinzipiell automatisierbar ist. So ist zum Beispiel die regelmäßige Überprüfung des aktuellen Druckzustandes bisher unumgänglich und durch den Einsatz von entsprechenden Sensoren und Algorithmen automatisierbar. Auch in der Nachbearbeitung gibt es ein hohes Potenzial zur Automatisierung. Durch die Vielfältigkeit der druckbaren Geometrien ist die automatisierte Nachbearbeitung jedoch beschränkt. Die Herstellung von rotierten Bohrungen ist mit der additiven Fertigung kein Problem. Das Entfernen der sich darin befindlichen Stützstruktur stellt für eine Roboteranwendung eine Herausforderung dar. Dies beginnt schon mit der Wahl des richtigen Werkzeugs, als auch der aufwändigen Regelung zur eigentlichen Entfernung der Stützstruktur.

## 1. Einleitung und Überblick

Ebenfalls gibt es im Bereich der Robotik große technische Fortschritte. Die Hardware aktueller Leichtbauroboter und neue Normen wie die ISO/TS 15066 [16] ermöglichen eine einfache und sichere Zusammenarbeit von Mensch und Roboter ohne Sicherheitseinrichtungen, wie beispielsweise bisher benötigte Sicherheitszäune und -zellen. Zudem ermöglichen mobile Plattformen den flexiblen Einsatz der Roboter an dem Arbeitsplatz, an dem er momentan benötigt wird und kann so effektiver eingesetzt werden. Dies ist gerade für Produktionen mit geringer Stückzahl interessant, da dafür der Aufbau einer spezialisierten Roboterzelle selten rentabel ist.

Im Industrieumfeld existieren bereits viele Prozesse, die dank moderner Robotik voll automatisiert ablaufen und in denen der Mensch nur noch zur Überprüfung, zur Wartung und zur Fehlerkorrektur eingebunden ist. Vor allem repetitive und gefährliche Aufgaben mit hohen Lasten können von Robotern deutlich effizienter erledigt werden. Im Gegensatz zum Menschen sind die Belastungen durch z.B. Feinstaub für den Roboter weniger schädlich, weshalb sie gerade für solche Arbeiten, wie der Umgang mit Druckmaterialien, prädestiniert sind. Obwohl die additive Fertigung [17] und die Robotik beide als wichtige und zukunftssträchtige Technologien eingeschätzt werden, werden sie bisher kaum in Anwendungen zusammen eingesetzt. Es gibt einzelne Anwendungen, in denen ein Roboterarm als Druckkopf genutzt wird [18] und auch diverse Konstruktionsprojekte, die Roboter für die Herstellung von z.B. Brücken in Amsterdam einsetzen. Am Nächsten an einer automatisierten Fertigungsstraße für die additive Fertigung ist das Produkt „Form Cell“ der Firma Formlabs [19]. Bei dieser manipuliert eine Drei-Achsen-Kinematik mehrere Stereolithographie-Drucker (SLA) und entnimmt hierbei das komplette Druckbett inklusive des gedruckten Objekts.

Dies löst aber nur einen kleinen Teil der Automatisierung, da die noch offenen Nachbearbeitungsschritte weiterhin von einem Menschen durchgeführt werden müssen.

Insgesamt bietet die Kombination aus additiver Fertigung und Robotik, zusammen mit aktuellen Algorithmen der Sensorverarbeitung, die Möglichkeit, ein System zu entwickeln, das die manuellen Arbeitsschritte in der additiven Fertigung reduziert und gleichzeitig den Druckprozess mit Hilfe von Verfahren der künstlichen Intelligenz überwacht und auf Ereignisse reagiert. Dies wird im Rahmen dieses Promotionsvorhabens untersucht und entsprechend untersucht.

## 1.2. Forschungsfragen

Die Forschungsfragen in dieser Arbeit sind auf die drei folgenden, überwiegend manuellen Arbeitsschritte innerhalb der Prozesskette der additiven Fertigung fokussiert. Die Fehlererkennung im Druckprozess, die Entnahme aus dem Drucker sowie die Nachbearbeitung nach erfolgter Entnahme. Ziel dieser Arbeit ist es diese Prozessschritte weitestgehend zu automatisieren oder zumindest den Anteil an manueller Zuarbeit deutlich zu reduzieren. Bei allen genannten Schritten liegt der Fokus auf dem Fused Filament Fabrication (FFF) Druckverfahren, sowie auf der kompletten Prozesskette der additiven Fertigung.

**In-situ Fehlererkennung** Die Prozessfehlererkennung wird mit Hilfe geeigneter Sensoren umgesetzt die so integriert werden, dass sie den Druckprozess optimal erfassen können. Neben verschiedenen Sensoren werden zudem mehrere unterschiedliche Verfahren der Sensordatenverarbeitung eingesetzt. Die Komplexität, die sich durch die Geometrievielfalt, den Druckprozess, sowie die möglichen Parameter, ergibt, ist ein idealer Anwendungsfall für heutige Lernverfahren.

Verschiedene Verfahren des maschinellen Lernens werden untersucht, um an Hand von Trainingsdaten zu lernen, wie ein Fehler klassifiziert werden muss. Neben der Umsetzung der Deep Learning Verfahren ist auch die Erfassung der notwendigen Trainingsdaten ein Teil der Forschungsfrage. Mit der Güte der Trainingsdaten geht auch die Güte des Lernverfahrens einher und bestimmt so maßgeblich die Qualität der Fehlererkennung.

Insgesamt ergeben sich hieraus zwei Forschungsfragen:

*Mit welchen Sensoren ist eine zuverlässige in-situ Fehlererkennung möglich?*

*Welche Verfahren zur Auswertung der Sensordaten eignen sich für diesen Prozessschritt?*

**Entnahme des gedruckten Objekts** Mehrere Schritte sind notwendig, um ein Objekt erfolgreich aus dem Drucker zu entnehmen. Zuerst werden Druckdaten analysiert, um ein Greifwerkzeug zu entwickeln, das die Objekte schadfrei entnehmen kann. Hierbei spielt vor allem die Geometrievielfalt, sowie die Haftung des Objekts am Druckbett eine Rolle. Zur Entnahme eines einzelnen Objektes wird zuerst berechnet, an welcher Position im Drucker das Objekt von einem Roboter gegriffen werden muss. Sind mehrere unabhängige Objekte gleichzeitig gedruckt worden, so muss zudem noch die Reihenfolge bestimmt und geplant werden. Neben der Berechnung der Greifposition anhand der Druckdaten wird mit Hilfe von 3D Kameradaten die Griffposition verifiziert. Abschließend wird das Objekt an einer definierten Stelle abgelegt.

Daraus ergeben sich die folgende Forschungsfragen:

*Ist es möglich gedruckte Objekte direkt aus dem Drucker mit Hilfe eines Roboters zu entnehmen? Welche Randbedingungen sind zu berücksichtigen, um eine zerstörungsfreie Entnahme durchzuführen?*

**Nachbearbeitung des gedruckten Objektes** Durch den Vergleich des Maschinencodes mit den ursprünglichen CAD-Daten werden die für den Druckprozess benötigten zusätzlichen Strukturen klassifiziert, die nicht zum eigentlichen Bauteil gehören. Diese Strukturen werden dann mit Hilfe eines Roboters und eines Werkzeugs entfernt. Dabei ist in einem Szenario das Werkzeug am Roboter montiert und das gedruckte Objekt fixiert und in einem weiteren Szenario das Werkzeug fest montiert und der Roboter greift das zu bearbeitende Objekt. Neben der Anordnung von Werkzeug und Objekt ist auch die Trajektorienplanung zu berücksichtigen. Eine sorgfältige Positionierung des Objektes relativ zum Werkzeug, sowie eine Planung ermöglicht das Bauteil mit wenig Umgreifvorgängen und eleganten Roboterbewegungen zuverlässig zu bearbeiten.

So lautet die Forschungsfrage zur Nachbearbeitung:

*Wie kann ein gedrucktes Objekt mit Hilfe eines Roboters automatisiert nachbearbeitet werden?*

### 1.3. Einordnung und wissenschaftlicher Beitrag

Obwohl die additive Fertigung bereits seit mehr als dreißig Jahren eine etablierte Produktionstechnik ist [20] und in Zukunft eine größere Rolle spielen soll [21, 22], gibt es bisher kaum wissenschaftliche Arbeiten zur Automatisierung in diesem Bereich [23]. Der Hauptentwicklungsaufwand im Bereich der additiven Fertigung fließt in die Verbesserung der Hardware, sowie der druckbaren Materialien [24]. Jedoch ist die Hardware bisher kaum mit internen Sensoren zur Prozessüberwachung ausgestattet und somit kaum Fehlererkennung integriert. Verschiedene Forschungsgruppen und Wissenschaftler haben sich daher dem Thema angenommen und verschiedenste Drucker mit Sensoren ausgestattet [25, 26, 27].

Mit Hilfe von Algorithmen und Verfahren wie Methoden der künstlichen Intelligenz werden die Daten verarbeitet und eine Klassifizierung durchgeführt [28, 29, 30]. Jedoch sind die Lösungsansätze teilweise für eigenentwickelte Drucker oder erfordern einen größeren Umbau der eigentlichen Druckerhardware, sodass dies nur schwer replizierbar ist.

In dieser Arbeit wird die komplette Prozesskette der additiven Fertigung betrachtet und hierfür ein Werkzeugkasten entwickelt, der die einzelnen automatisierbaren Teilprozesse berücksichtigt und entsprechende Lösungen bereitstellt. Angefangen bei der Prozessbeobachtung, über den Planungsvorgang der Entnahme der gedruckten Objekte, bis hin zur Nachbearbeitung zur finalen Geometrie werden die Schritte analysiert und Lösungen erarbeitet.

Für die Prozessbeobachtung werden mehrere Sensoren, wie 2D Kameras oder Mikrofone eingesetzt, die flexibel an verschiedenen Punkten innerhalb des Druckers eingesetzt werden können. Durch die Anwendung aktueller Verfahren der Datenverarbeitung wie zum Beispiel Deep Learning Verfahren werden die erfassten Daten ausgewertet und entsprechend klassifiziert.

## *1. Einleitung und Überblick*

Bei der automatisierten Entnahme aus einem 3D-Drucker gibt es bisher nahezu keine wissenschaftlichen Beiträge. Erst seit kurzem sind einige Unternehmen dabei ihre Produkte entsprechend zu automatisieren beziehungsweise selbst ganze Drucker-Farmen anzubieten [19, 31, 32]. Jedoch gibt es im Bereich der klassischen Robotik sehr viele und umfangreiche Arbeiten zum Thema des robotischen Greifens [33], der Pfadplanung [34, 35], sowie der Kollisionsberechnung [36].

Im Rahmen dieser Arbeit wird die Entnahme der gedruckten Objekte aus dem Druckraum ganzheitlich betrachtet. Zu Beginn wird ein adaptiver Greifer für verschiedenste Geometrien entwickelt. Des Weiteren wird aus den vorhandenen Prozessdaten die Position der einzelnen gedruckten Objekte ermittelt, damit hiermit eine Trajektorie für die Entnahme mit dem Roboter berechnet werden kann.

Bei der Nachbearbeitung der gedruckten Objekte gibt es ebenfalls keine thematisch naheliegenden wissenschaftlichen Arbeiten. Im Bereich der spanenden Verfahren, wie z.B. des Fräsens, gibt es jedoch etliche Vorarbeiten zur Bauteilverarbeitung [37, 38]. Diese Prozesse werden jedoch hauptsächlich mit dafür konstruierten Maschinen durchgeführt, da die Steifigkeit von Knickarm-Kinematiken in den industriellen Anwendungsfällen problematisch ist [39]. Da in der typischen Fräsanwendung hauptsächlich Metalle verarbeitet werden, sind hier hochpräzise und gleichzeitig steife Maschinen notwendig.

Jedoch sind die Polymere, die in dieser Arbeit verwendet werden, weich genug, sodass die Steifigkeit des Roboterarmes in dieser Arbeit nur eine untergeordnete Rolle spielt. Das Wissen aus dem Herstellungsprozess der Objekte, das bedeutet die eigentliche Geometrie, sowie die prozessbedingten Stützstrukturen, werden in dieser Arbeit verwendet, um diese mit Hilfe eines Roboterarmes, sowie eines Werkzeuges zu entfernen.

## 1.4. Gliederung der Abhandlung

Die Motivation für diese Arbeit, sowie die Abgrenzung findet vor allem im Kapitel Einleitung und Überblick statt. Im folgenden Kapitel 2 wird der Stand der Wissenschaft dargestellt und dargestellt, inwiefern sich der Beitrag dieser Arbeit von diesen unterscheidet.

In Kapitel 3 wird die Fehlererkennung im Prozess behandelt. Hierbei werden die verschiedenen Verfahren dargestellt, sowie auf deren Details eingegangen. In jedem der drei Verarbeitungsmethoden wird auf die Fehlertypen eingegangen, die von dem jeweiligen Verfahren erkannt werden soll. Abschließend wird jedes Verfahren evaluiert und qualitativ bewertet.

Das Kapitel 4 befasst sich mit der Entnahme der gedruckten Objekte aus dem Bauraum des Druckers. Zu Beginn werden mit Hilfe einer Analyse die geometrischen Eigenschaften typischer Drucke untersucht, um darauf aufbauend einen Greifer zur Entnahme zu entwickeln. Anschließend wird auf Basis der Daten ein Greifermodell konstruiert und gefertigt mit welchem die Objekte entnommen werden. Mit Hilfe des G-Codes wird ermittelt, an welcher Stelle sich die Objekte im Druckraum befinden und in welcher Reihenfolge sie entnommen werden müssen.

Kapitel 5 befasst sich mit der Nachbearbeitung der gedruckten Objekte. Hierbei kommen zwei verschiedene Ansätze zum Einsatz. Zum Einen ist das Werkzeug zur Nachbearbeitung, eine Art Fräse, am Roboter montiert und das Objekt ist fixiert. Zum Anderen wird das Objekt vom Roboter gegriffen und entlang einer fest montierten Fräse geführt, um es nachzubearbeiten. Abschließend werden die jeweiligen Verfahren evaluiert und bewertet.

Das Kapitel 6, fasst die erarbeiteten Inhalte zusammen und diskutiert die Ergebnisse und liefert einen Ausblick auf mögliche, zukünftige Arbeiten.



## 2. Stand der Forschung

Das Forschungsfeld rund um die additive Fertigung ist breit aufgefächert und erstreckt sich über viele Disziplinen. So werden neben dem Bereich der Materialwissenschaften auch in den Bereichen der Prozessüberwachung und des Maschinenbaus wissenschaftliche Ergebnisse publiziert, die die additive Fertigung betreffen.

Im folgenden Kapitel wird der aktuelle Stand der Wissenschaft für die einzelnen Aspekte dieser Arbeit untersucht. Hierbei sind vor allem die Überprüfung der Güte während des Drucks, die Entnahme des gedruckten Bauteils, sowie die abschließende Bearbeitung des Bauteils mit Hilfe maschineller Verfahren im Fokus.

### 2.1. Additive Fertigung

Das additive Fertigungsverfahren, das heute als 3D-Druck bekannt ist, wurde in den 1980er Jahren von Charles W. Hull entwickelt und patentiert [40]. Bei diesem Verfahren, das Stereolithographie (SLA) genannt wird, wird der Querschnitt des gewünschten Objekts Schicht für Schicht aus einem UV-lichtempfindlichen Harz hergestellt, das mit einem Laser ausgehärtet wird. Durch Vergrößerung des Abstands zwischen Bauplatte und Harz wird die nächste Harzschicht aufgetragen und ausgehärtet. Dieser Zyklus wird so lange wiederholt, bis das Objekt fertiggestellt ist.

Heutzutage ist das gängigste additive Fertigungsverfahren Fused Deposition Modelling (FDM) oder Fused Filament Fabrication (FFF), bei dem das zu fertigende Modell schichtweise aus einem schmelzbaren Material, z. B. Kunststoff, erstellt wird. Es zeichnet sich durch niedrige Herstellungskosten und eine große Varianz an verwendbaren Materialien aus. Ein Nachteil ist die hohe Anzahl an möglichen Prozessfehlern, die zum Beispiel durch Hardware- oder Materialeinflüsse auftreten können. So kann eine falsche Einstellung der Bewegungs-, Extrudier- und Temperaturwerte dazu führen, dass die Druckdüse verstopft, kein Material mehr extrudiert wird und der Druckprozess ab einem gewissen Zeitpunkt nicht mehr korrekt fortgesetzt wird. Dadurch wird das Werkstück unbrauchbar.

Neben der steigenden Zahl an Herstellern für Druckerhardware, sowie Material, nimmt auch die Zahl der Unternehmen und Dienstleister zu. Dies umfasst sowohl die Hersteller von 3D-Druck Systemen, als auch die Dienstleister und Nutzer selbst. Gleichzeitig ermöglichen Initiativen wie das RepRap-Projekt [41] Hobbyisten weltweit, 3D-Drucker zu niedrigen Einstiegspreisen selbst zu bauen und zu verbessern [42, 43].

## 2. Stand der Forschung

Parallel zu diesen endverbraucherorientierten Konzepten wächst die Nachfrage nach Serien- und Massenproduktion von additiven Teilen und damit der Bedarf an Prozesskontrolle und -optimierung. Die Entwicklung von Automatisierungsmöglichkeiten für 3D-Druckprozesse rückt daher immer mehr in den Fokus von Industrie und somit auch der Wissenschaft.

Obwohl die Basistechnologie bereits mehrere Jahrzehnte alt ist, gibt es immer noch Herausforderungen, die für den produktiven Einsatz der additiven Fertigung in der Industrie gelöst werden müssen [44, 45, 14], wie zum Beispiel die Reduktion der manuellen Arbeitsschritte.

Im Folgenden werden zwei verschiedene Druckverfahren vorgestellt, die in dieser Arbeit relevant sind. Weitere Druckverfahren sind z.B. im Artikel des Manufactur3d Magazines [46] genannt, werden hier aber nicht weiter erläutert.

### 2.1.1. Fused Filament Fabrication (FFF)

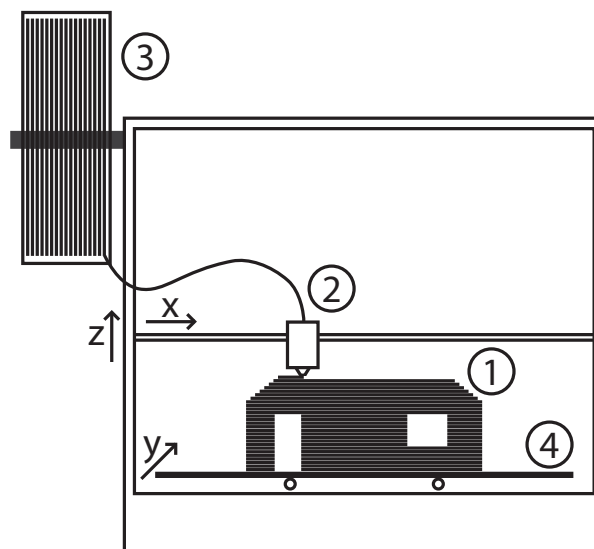


Abbildung 2.1.: Schematische Darstellung des Druckprozesses bei Fused Filament Fabrication.  
1) gedrucktes Objekt, 2) Extruder / Druckkopf, 3) Filamentrolle, 4) Druckbett.

Bei der Fused Filament Fabrication (FFF), auch unter dem Begriff Fused Deposition Modeling (FDM) bekannt, wird ein Filamentstrang geschmolzen und mit Hilfe einer beweglichen Düse, auch Extruder genannt, an einem bestimmten Punkt im Raum aufgebracht. Ursprünglich wurde das Verfahren von S. Scott Crump im Jahr 1989 erfunden und zum Patent beantragt [47]. Nach Ablauf des Patents wurde es vor allem von der Open-Source Community weiterentwickelt. Gedruckt wird auf ein Druckbett, das eine bestimmte Temperatur aufweist, um eine möglichst starke Haftung zu erreichen. Beim Druck kommen hauptsächlich Kunststoffe als Material zum Einsatz, da diese eine relative geringe Schmelztemperatur zwischen 200 und 300 Grad Celsius haben. Durch verschiedene Zusätze und Zusammensetzungen sind verschiedene Eigenschaften des Materials möglich. Von Stabilität über Transparenz bis hin zur Steifigkeit sind verschiedene

## 2. Stand der Forschung

Materialeigenschaften denkbar. Der Hardwareaufbau besteht aus drei Achsen, die zueinander beweglich sind, sodass der Extruder in X-, Y- und Z-Achse beweglich ist. Der Extruder auf der XY-Ebene bewegt sich entlang der Z-Achse vom Druckbett weg.

Schematisch ist dies in Abbildung 2.1 dargestellt.

Durch mehrere Druckköpfe oder Wechselsysteme ist es mit diesem Verfahren möglich mit mehreren Materialien gleichzeitig zu drucken oder mehrere Drucke parallel in einem Drucker herzustellen.

### 2.1.2. Laser Powder Bed Fusion (L-PBF)

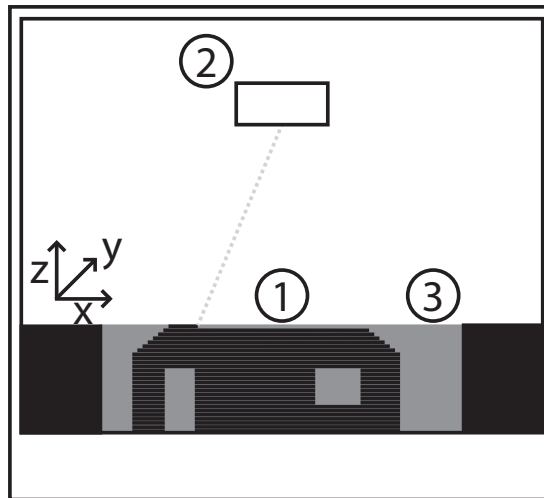


Abbildung 2.2.: Schematische Darstellung des Druckprozesses beim Powder Bed Fusion Druckprozess. 1) Gedrucktes Objekt, 2) Laserquelle, 3) Druckraum gefüllt mit Pulver

Bei dem Druckverfahren des Powder Bed Fusion wird das Objekt in einem Pulverbett hergestellt [48]. Hierfür wird eine Lage Pulver aufgebracht, wobei die Schichtdicke sowohl vom Material als auch von den gewählten Einstellungen, wie zum Beispiel Scan-Geschwindigkeit, Dicke der Pulverlage oder Laserleistung, abhängt. Nun wird mit Hilfe eines Lasers an den Stellen das Pulver aufgeschmolzen, an denen das Objekt in dieser Schicht entstehen soll. So verschmelzen einzelne Pulverbestandteile mit anderen in der gleichen Schicht, sowie den darunterliegenden Schichten und es entsteht Schicht für Schicht das gewünschte Objekt. Das gedruckte Objekt ist nach Fertigstellung allerdings komplett von Pulver umhüllt, wie in Abbildung 2.2 dargestellt. Hohlräume im Objekt sind ebenfalls mit dem Pulver gefüllt. Hierfür müssen beim Design bereits Öffnungen vorgesehen werden, damit das Pulver entfernt werden kann. Die Festigkeit und Stabilität der gedruckten Objekte hängt stark von der Parameterwahl des Lasers ab. Ist die Energiequelle zu stark eingestellt, so schmilzt zu viel und die Maßhaltigkeit stimmt nicht mehr oder das Pulver beginnt sogar zu kochen, wodurch Explosionen entstehen, die sich dann im Objekt als poröse Strukturen widerspiegeln. So ist auch bei diesem Druckverfahren die richtige Wahl der Parameter für ein qualitativ gutes Druckergebnis entscheidend.

### 2.2. Robotik

Nicht nur im Bereich der additiven Fertigung gab es in den letzten Jahren große Fortschritte, sondern auch im Bereich der Robotik, speziell in der Leichtbau- bzw. Servicerobotik. Leichtbauroboter sind auch unter dem Namen kollaborierende Roboter oder Cobots bekannt. Mehrere Normen beschäftigen sich ausführlich mit dieser Art der Robotik [16, 49, 50, 51]. Der Fokus ihrer möglichen Aufgaben liegt in der Kooperation mit dem Menschen. Konstruktiv sind die Cobots daher so gebaut, dass das Gefahrenpotential bei der Zusammenarbeit minimal ist. Dadurch sind die möglichen Tragkräfte ( $< 50$  kg) und Reichweiten ( $< 2$  m) des Roboterarmes zwar geringer als bei herkömmlichen Industrierobotern, allerdings kann z.B. auf einen Zaun als Sicherheitseinrichtung, je nach Anwendung verzichtet werden. Zudem werden oft Geschwindigkeit- und Beschleunigungswerte reduziert, damit der mögliche Impuls bei Kontakt zwischen Roboter und Mensch gering ausfällt. Neben der verringerten Unfallgefahr durch den Roboterarm, sind oft auch Kraft-Momenten-Sensoren verbaut, über die es möglich ist, auftretende Kräfte zu messen und den Roboter bei ungewolltem Kontakt zu verfahren oder abzuschalten.

Die im Rahmen dieser Arbeit verwendeten Robotertypen sind im Anhang A.1.5 aufgeführt.

Doch nicht nur die jüngsten Entwicklungen in der Hardware sind entscheidend für den Einsatz der Cobots in immer mehr Anwendungen, sondern auch die Möglichkeit der Verwendung von Open-Source Software zur Steuerung. Im Jahr 2009 wurde die erste Version des Robot Operating System (ROS) [52] online und als Open-Source Framework zur Verfügung gestellt. Ursprünglich war das Framework vor allem für den Alltag in der Forschung konzipiert, um den Aufwand in Forschungsinstituten zu reduzieren, da nicht jeder Forschende jeden Algorithmus von Neuem implementieren muss. Mit der Zeit entwickelte sich ROS immer weiter und so wurde auch eine Version für den industriellen Einsatz (ROS-Industrial) entwickelt, die sich durch besonders stabile Komponenten auszeichnet. Durch den weltweiten Einsatz an sehr vielen Forschungseinrichtungen im Bereich der Robotik wächst die Anzahl der Nutzer und der bereits implementierten Treiber und Algorithmen stetig. Laut Open Source Robotic Foundation (OSRF) sind mehr als 900 Firmen bekannt, die ROS für ihre Produkte einsetzen [53]. Dies führt auch dazu, dass Forschungsarbeiten direkt im ROS Framework umgesetzt und veröffentlicht werden, um anderen Forschenden eine einfache Verwendung zu ermöglichen. Andere bereits quelloffene Softwarebibliotheken, wie z.B. die Open Motion Planning Library (OMPL) [54], zur Planung von Robotertrajektorien, sind umfassend in ROS integriert und so können neu entwickelte Planungsalgorithmen unkompliziert an Robotern ausprobiert werden, ohne erst einen Adapter zur proprietären Programmiersprache jedes Roboterherstellers zu entwickeln.

## 2.3. In-Situ Fehlererkennung in der additiven Fertigung

Eine der größten Herausforderungen in der additiven Fertigung ist der Prozess selbst. Das finale Druckergebnis ist abhängig von einer Vielzahl an Parametern und Einflussfaktoren des Prozesses und ist von Druckverfahren zu Druckverfahren stark unterschiedlich. Je früher ein Fehler im Druckprozess erkannt werden kann, desto früher können Parameter angepasst, Einstellungen geändert oder der Druck neu gestartet werden. Hierfür muss der komplette Prozess aber durchgehend überwacht und die aufgenommenen Sensordaten zudem ausgewertet werden.

Die Fehler im Druckprozess können durch verschiedene Sensoren erkannt werden. So eignen sich vor allem optische Sensoren zur Erkennung, aber auch akustische Sensoren oder Beschleunigungssensoren, Temperaturwerte und Motorströme ermöglichen eine zuverlässige Fehlererkennung.

Mit Hilfe von verschiedenen Verfahren des maschinellen Lernens wird es möglich, auch komplexe Fehler zu erkennen. Die Basis für die Erstellung eines funktionierendes Lernverfahren ist der verwendete Datensatz beim Training. Je mehr Daten und Varianz in einem Trainingsdatensatz enthalten sind, desto besser können die späteren Fehlerfälle eindeutig klassifiziert werden.

### 2.3.1. Optische Überwachung des Druckprozesses

Um die beim FDM-Druck auftretenden Fehler besser zu verstehen, haben Wang *et al.* bereits 2006 die Einflüsse von thermischen Spannungszuständen, die dem Fehlerfall Warming zugrunde liegen, analysiert und Methoden zur Fehlerreduzierung entwickelt [55].

Liverani, Leali und Pellicciari stellten im Jahr 2010 einen Algorithmus vor, der aus einem kontinuierlichen Bildstrom einer monokularen Kamera das aufgenommene Objekt in 3D rekonstruieren kann [56]. Darauf aufbauend entwickelten Nuchitprasitchai, Roggemann und Pearce 2017 ein visuelles Kontrollsystem, das während eines laufenden 3D-Drucks den Druck in ausgewählten Intervallen anhält und die kamerabasierte 3D-Rekonstruktion des Teils im Bauraum mit dem theoretischen Fortschritt vergleicht [25]. Für dieses Verfahren müssen die Kamerapositionen relativ zueinander exakt bekannt sein und die Bilder mittels eines SIFT-Algorithmus entsprechend in Beziehung gesetzt werden. Der Algorithmus prüft zudem nur die volumetrische Ähnlichkeit auf Fehler und erkennt somit keine Fehler wie Warming und Layer Shifts.

Bereits 2012 stellten Kleszczynski *et al.* ein Fehlererkennungssystem zur Erhöhung der Prozesssicherheit in SLM-Drucksystemen (Selective Laser Melting ähnlich Powder Bed Fusion) vor. Bei diesem Verfahren werden Bauteile durch ein laserbasiertes, lokales, schichtweises Aufschmelzen von Metallpulver hergestellt. Zu diesem Zweck analysierten sie häufig auftretende Fehler und stellten mögliche Gegenmaßnahmen vor [57].

## 2. Stand der Forschung

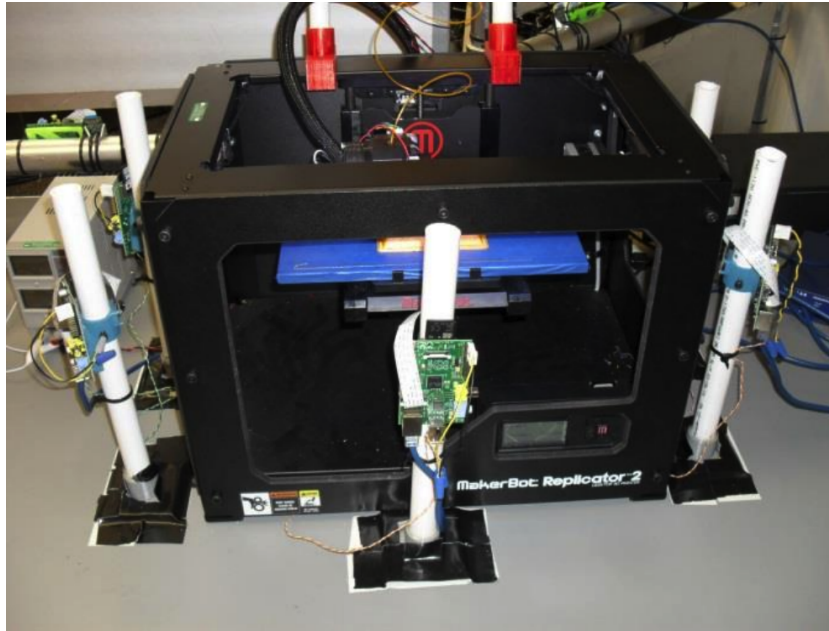


Abbildung 2.3.: Prozessüberwachung durch die Anordnung mehrerer Kameras (grünes Boards auf weißen Säulen) direkt vor dem Drucker auf Höhe des Extruders. So kann der komplette Entstehungsprozess aus verschiedenen Perspektiven beobachtet werden. Quelle [60].

Im Jahr 2013 veröffentlichten Dinwiddie, Love und Rowe eine Studie über Temperaturunterschiede im gedruckten Objekt während des Druckprozesses und die daraus resultierenden Unterschiede in den mechanischen Eigenschaften in einem industriellen FFF-Drucker [58]. Mit einem ähnlichen Ansatz untersuchten Krauss *et al.* 2014 die Druckprozesse in einem SLM-System. Krauss *et al.* untersuchten diese Schichten auf Temperaturspitzen, um Rückschlüsse auf Materialeigenschaften und Prozessführung zu ziehen [59].

Für Desktop-FFF-Drucker präsentierte Jeremy Straub 2015 einen Vorschlag für ein System mit mehreren Kameras rund um den Bauraum, siehe Abbildung 2.3, das zur vollständigen Prozessüberwachung aus verschiedenen Perspektiven eingesetzt werden könnte [60].

Im Jahr 2016 stellte das Forschungsteam um Faes *et al.* ihren Ansatz vor, bei dem sie einen 2D Laser Scanner in einen einfachen FFF-Drucker einbetteten [26]. Dadurch wurde es ihnen möglich die z-Höhe des aktuellen Drucks zu bestimmen, um so frühzeitig Fehler im Druckprozess festzustellen. Die Daten des über dem gedruckten Filament platzierten Laser Scanners wurden analysiert, um die Höhe der letzten Druckschicht zu berechnen. Diese Information wurde dann verwendet, um über ein Kontrollsystem Defekte zu kompensieren, wie z. B. eine schlechte Schichtverbindung oder eine Überextrusion, die durch eine Kombination von Fehlern bei der vertikalen Führung und der Extrusionsrate verursacht wurden.

Baumann *et al.* haben ein Überwachungssystem für einen FFF-Drucker entwickelt, das mit Hilfe einer Kamera das Zentrum des gedruckten Objekts lokalisiert und eine Freigabe des Drucks durch Vergleich der Position über die Zeit erkennt [27]. Darüber hinaus wird mit einem

## 2. Stand der Forschung

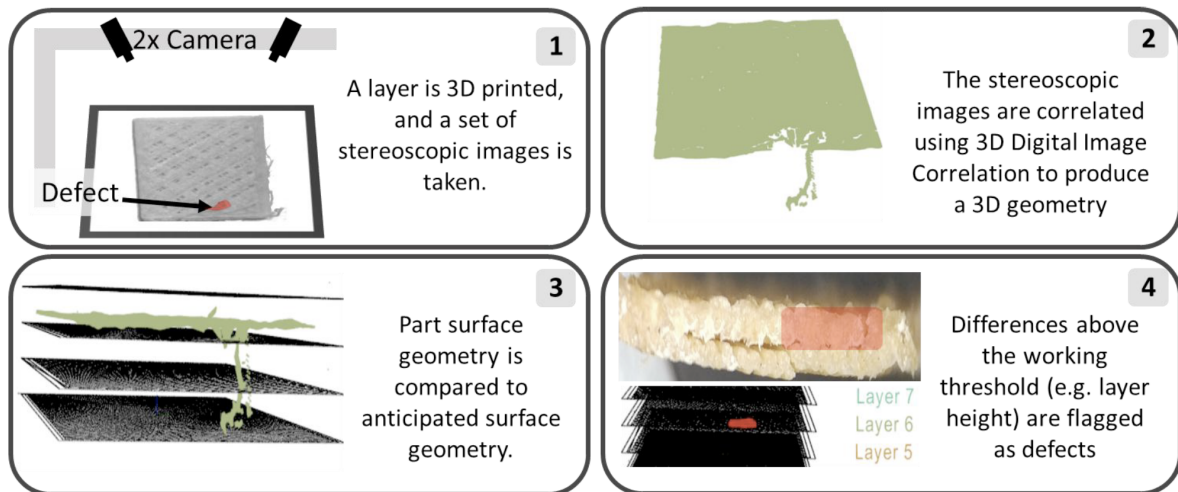


Abbildung 2.4.: Grafische Zusammenfassung der Arbeit von Holzmond *et al.* Ein Zwei-Kamera-Setup wurde in einem FFF-Drucker installiert, um den Druck zu verifizieren, während das Objekt gedruckt wurde. Mit digitaler 3D-Bildkorrelation (3D-DIC) und einem Vergleich mit dem Computermodell konnten sie Fehler während des Druckes erkennen. Quelle [61].

Canny-Filter die Oberkante des gedruckten Objekts bestimmt und der Abstand zum markierten Extruder gemessen. Überschreitet der gemessene Abstand einen bestimmten Grenzwert, wird eine unterbrochene Extrusion erkannt.

In dem Ansatz von Nuchitprasitchai *et al.* wird der FFF-Prozess an mehreren signifikanten Punkten angehalten und mit Hilfe von zwei Kameras ein 3D-Modell des aktuellen Produktionszustandes rekonstruiert [62]. Dieses rekonstruierte Objekt wird nun über ein perspektivisches Bild mit der STL-Datei des zu fertigenden Objekts verglichen. Weichen die Bilder um mehr als 5% voneinander ab, wird der Druck als fehlerhaft eingestuft. Bei der bisherigen Vorgehensweise wurden nur fertige Drucke analysiert und somit die in-situ-Fähigkeit nicht bestätigt. Außerdem ist es problematisch, dass gedruckte Objekte mit STL-Objekten verglichen werden, die eine viel bessere Oberflächenqualität aufweisen, selbst wenn der Druck fehlerfrei ist. Durch diese Differenz zwischen virtuellem und realen Objekt kann es zu falschen Eindrücken kommen und so die Erkennung von Fehlern verfälschen. Mit dem von Holzmond *et al.* vorgestellten Ansatz der dreidimensionalen digitalen Bildkorrelation wurde ebenfalls eine räumliche Rekonstruktion aus zwei Bildperspektiven erstellt, siehe Abb. 2.4 [61]. Im Gegensatz zu Nuchitprasitchai *et al.* wurde nicht das CAD-Modell für den Vergleich herangezogen, sondern der Pfad der Maschine (G-Code) für das Druckobjekt. Durch die Rekonstruktion des G-Codes als Punktwolke können die Oberflächen durch die Druckartefakte realistischer erzeugt werden. Ähnlich wie bei den oben beschriebenen Ansätzen wird ein Druck als fehlerhaft eingestuft, wenn seine Abweichung eine bestimmte Toleranzgrenze überschreitet. Mit dem vorgestellten System können lokale Defekte wie Filamenttropfen oder unzureichender Materialfluss identifiziert werden. Der Ansatz von Moretti *et al.* vergleicht ein im Bauprozess aufgenommenes Bild mit einem digitalen Zwilling des Druckobjektes. [63]. Da die Auflösung herkömmlicher Kameras oft zu gering ist,

## 2. Stand der Forschung

wird hier ein USB-Mikroskop eingesetzt. Aufgrund des kleinen Bildbereichs des Mikroskops wird für jede Schicht ein spezifischer Pfad festgelegt, auf dem das Mikroskop die Konturen des Druckobjekts nachzeichnet. Der Vergleich über den digitalen Zwilling erweist sich bei den hochauflösenden Bildern als vergleichsweise ressourcenschonend und effizient.

Liu *et al.* stellte einen Ansatz vor, bei dem ein am Extruder angebrachter Schallemissionssensor den Maschinenzustand eines FFF-Druckers überwacht [64]. Mithilfe eines Clustering-Ansatzes sind sie in der Lage, verschiedene Maschinenzustände wie das Be- und Entladen sowie eine unterbrochene Filamentzufuhr zu erkennen. Mit einem ähnlichen Versuchsaufbau konnte Wu *et al.* mit Hilfe von Support-Vektor-Maschinen (SVM) verstopfte Extruder von nicht verstopften unterscheiden [65]. Bei beiden Ansätzen basiert die Unterscheidung auf spezifischen Merkmalen im Frequenzbereich der Messung.

Tlegenov *et al.* brachte einen Beschleunigungssensor am Extruder an und konnte so einen verstopften Extruder erkennen [66]. Da die Extruder eines FFF-Druckers in der Regel beweglich sind, bestand der erste Schritt darin, einen Drucker mit einem festen Extruder und einer beweglichen Plattform zu entwickeln. Abhängig vom Material und der Temperatur wurden die üblichen Schwingungen ermittelt und ein Durchschnittswert berechnet. Mittels gezielter Fehlersimulation wurde ein Zusammenhang zwischen der Beschleunigungsamplitude und des verstopften Extruder dokumentiert. Ein starker Anstieg der am Extruder gemessenen Schwingungen kann demnach auf einen verstopften Extruder zurückgeführt werden. In einer weiteren Studie konnte Tlegenov *et al.* einen direkten Zusammenhang zwischen dem elektrischen Strom am Extrudermotor und eines verstopften Extruders herstellen [67]. Wenn der Extruder zu verstopfen beginnt, verringert sich der effektive Durchmesser des Extruders, was den Gegen- druck erhöht. Um die Extrusionsgeschwindigkeit konstant zu halten, muss auch der Strom am Schrittmotor erhöht werden. Dieser Anstieg wird durch ein Strommessgerät überwacht.

Da es in einem additiven Fertigungssystem meist mehr als eine Fehlerquelle gibt, liegt es nahe, die vorhandenen Sensoren in einem gemeinsamen Sensorsystem zu kombinieren. Zu diesem Zweck haben Rao *et al.* und Moretti *et al.* einen Drucker mit mehreren spezifischen Sensoren ausgestattet [68, 69]. Bei beiden Ansätzen werden Fehler durch eine Abweichung vom ermittelten Normalzustand erkannt. Rao *et al.* kombinierte Thermoelemente zur Messung der Extruder- und Betttemperatur mit Beschleunigungssensoren zur Messung der Extruder- und Bettschwingungen. Neben einem Endoskop zur Aufzeichnung des gesamten Druckprozesses erweiterten sie das System zur Temperaturmessung am Schmelzpunkt mit einem Infrarotthermometer. Mit einer Kombination aus Dirichlet-Prozessmischungsmodellen und der Evidenztheorie konnten Prozessfehler mit einer hohen Wahrscheinlichkeit mit einem F1-Score von 86% identifiziert werden. Der Ansatz von Moretti *et al.* kombiniert Positions- und Glühfadennmessgeräte mit Thermoelementen und einer Kamera. Es wird gezeigt, dass oft mehr als ein einzelner Sensor erforderlich ist, um einen Fehler zu erkennen. So kann beispielsweise ein oberflächlicher Fehler durch die Kombination von Positionsmessung und Vorschubgeschwindigkeit erkannt werden.

Die Ansätze von Rao *et al.* und Moretti *et al.* zeigen den großen Vorteil der visuellen Überwachung. Einige Fehlerkategorien lassen sich nicht den eindimensionalen Sensordaten zuordnen. So kann beispielsweise das Ablösen des Druckobjekts von der Bauplattform nicht durch die Düsentemperatur oder die Vorschubgeschwindigkeit erkannt werden. Ebenso hängt der Verzug



## 2. Stand der Forschung

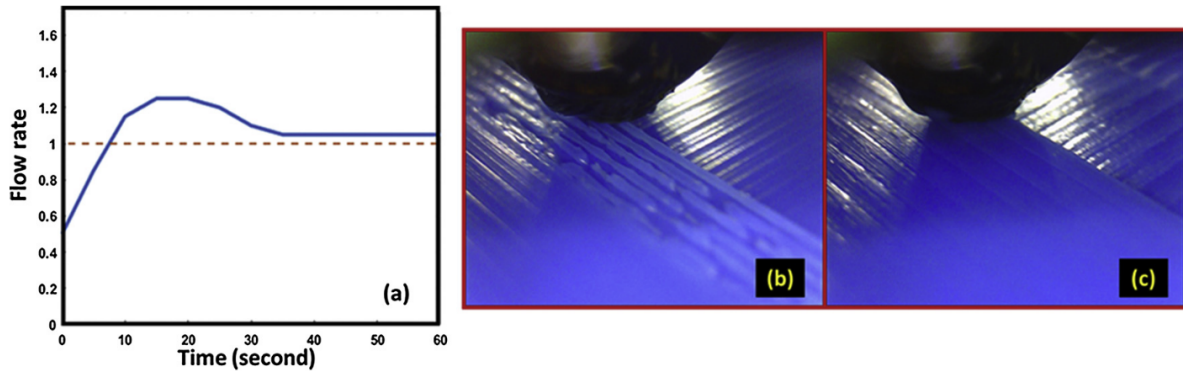


Abbildung 2.5.: Bildbasierte Fehlererkennung mit zwei an den Extruder montierten Digitalmikroskopen. Die Durchflussrate des Filaments wird angepasst, wenn ein Fehler erkannt wurde. Bild (a) zeigen die Durchflussrate während des Prozesses, das Bild der Mikroskope vor (b) und nach (c) der Regleranpassung. Diese Anpassung wurde durch ein k-NN gelernt und übertraf Alternativen wie LDA, SVM und NBC. Quelle [73].

des Objekts nicht nur von der Betttemperatur, sondern auch von der Temperatur der angrenzenden Schichten ab und kann daher nicht allein durch die internen Sensoren gemessen werden. Optische Kontrollmethoden basieren im Prinzip auf einer oder mehreren Kameras, die das Druckobjekt aus einem oder mehreren Winkeln überwachen.

Es gibt aber auch Ansätze wie den von Greeff *et al.*, bei dem der Filamentvorschub mit einer Kamera überwacht wird, um die Vorschubgeschwindigkeit und den Zustand des Filaments zu bestimmen [70]. Die vorgestellten Methoden beruhen auf numerischen und analytischen Modellen. Bei großen Datenmengen und insbesondere bei Bilddaten erfordern solche Ansätze eine enorme Rechenleistung. Beispielsweise ist es mit dem Ansatz von Moretti *et al.* nicht möglich, rechtzeitig auf den erkannten Fehler zu reagieren, da die Rechenzeit die Druckzeit einer Schicht übersteigt [63]. Um diese hochdimensionalen und hochfrequenten Daten kontinuierlich verarbeiten zu können, sind fortschrittliche Berechnungs- und Analysewerkzeuge erforderlich [71]. Hier bietet die Entwicklung im Bereich des maschinellen Lernens großes Potenzial. Mit maschinellem Lernen kann implizites Wissen entdeckt, Zusammenhänge in großen Produktionsdatensätzen identifiziert und so noch nie dagewesene Datenmengen in nutzbare und aufschlussreiche Informationen umgewandelt werden [28, 29]. Bei den Modellen handelt es sich nicht um eine lange Liste physikalisch basierter Gleichungen, sondern sie lernen automatisch die Beziehung zwischen Eingabemerkmalen und Ausgabezielen auf der Grundlage früherer Daten [72].

Im Gegensatz dazu verwendet Zhang *et al.* ein Convolutional Neural Network (CNN) für die Erkennung von FFF-Anomalien [30]. In ihrer Studie untersuchen sie die Eigenschaften der Deep-Learning-Anwendung und die Beziehung zwischen verschiedenen Parametern und deren Einfluss auf die Anzahl der richtigen Klassifizierungen. Mit einer Genauigkeit von 70% konnte gezeigt werden, dass der Algorithmus die Mehrheit der Bilder richtig klassifiziert. Um ein Überwachungssystem am FFF speziell für die Fehlerklasse Warping aufzubauen,

## 2. Stand der Forschung

setzte Saluja *et al.* ebenfalls ein CNN ein [74]. Um eine bestmögliche Überwachung des Fehlertyps Warping zu gewährleisten, konzentriert sich der Algorithmus nur auf den relevanten Teil an den unteren Ecken des Druckobjekts, wo der Fehler erwartet wird. In ihrer Studie untersuchten sie den Einfluss verschiedener Optimierer, Aktivierungsfunktionen und Dropouts. Mit dem vorgestellten Ansatz erreichen sie eine Genauigkeit von 97% und validieren die Ergebnisse anhand verschiedener Geometrien. Neben der bisher vorgestellten sogenannten Binär- oder Anomalieerkennung wurden verschiedene Mehrklassen-Klassifikationen in der Prozessüberwachung untersucht.

Liu *et al.* schlagen einen Ansatz zur Prozessüberwachung von FFF-Druckern vor, bei dem in einem ersten Schritt Texturinformationen aus Kamerabildern mittels einer Grauwertmatrix [73] extrahiert werden. Zu diesem Zweck werden zwei Digitalmikroskope direkt am Extruder angebracht, siehe Abb. 2.5. Aus der Texturauswertung können statistische Merkmalsvektoren ermittelt werden. Basierend auf diesen Merkmalen Kontrast, Korrelation, Energie und Homogenität, werden die Bilder mit Hilfe von k-NN den drei möglichen Klassen zugeordnet. Der vorgestellte Ansatz mit k-NN übertrifft in der Vorhersagewahrscheinlichkeit Alternativen wie Latent-Dirichlet-Allokation (LDA), Support Vector Machines (SVM) und Naive Bayes Classifier (NBC). In einer weiteren Veröffentlichung haben Liu *et al.* eine zweistufige Klassifikation aufgeführt [75]. In der ersten Stufe wird zwischen fehlerhaften und fehlerfreien Drucken unterschieden. Die fehlerhaften werden in einer zweiten Stufe der jeweiligen Fehlerklasse zugeordnet. So können Kapazitäten in der Mehrklassen-Klassifikation eingespart werden.

Wang *et al.* beispielsweise verwendet ein CNN mit zwei sogenannten Residualblöcken, um sechs oberflächlich auftretende Fehler zu klassifizieren [76]. Um Defekte auf dem gesamten Druckobjekt erkennen zu können, wird ein rotierendes Kamera-Setup gewählt, das den Druckaufbau aus jedem Winkel überwachen kann. Um eine möglichst große Fläche abzudecken, wird ein optimaler Positionsplan der Kamera in Abhängigkeit von der Geometrie des zu produzierenden Objekts erstellt. Die Bilder werden nicht als Ganzes klassifiziert, sondern in einzelne Segmente des relevanten Bereichs unterteilt. Die Auswahl des relevanten Bereichs erfolgt über eine Maske, die aus einem gerenderten Bild des CAD-Objekts aus derselben Perspektive generiert wird. Der Ansatz ist mit einer Genauigkeit von 91% vielversprechend und übertrifft in einem durchgeführten Vergleich bekannte Netze wie ResNet50 [77] oder Inception[78].

Banadaki *et al.* nutzt CNN zur Klassifizierung der Produktionsparameter Druckgeschwindigkeit und Filamenttemperatur des Druckprozesses auf Basis von im Prozess aufgenommenen Bildern [79]. So lässt sich anhand der über den Algorithmus ermittelten Parameter direkt feststellen, welche Parameter in der Vorbereitung falsch eingestellt sind. Das gesamte Druckobjekt wird aus einer festen Perspektive überwacht. In ihrem Ansatz verwenden sie das vortrainierte Modell Inception-3 [80], um 21 Klassen zu identifizieren, die sich aus Kombinationen von Temperatur und Geschwindigkeit ergeben. Mithilfe einer am Extruder angebrachten Kamera entwickelte Jin *et al.* ein System zur Unterscheidung der Fehlerkategorien Unter- und Überextrusion von fehlerfreien Druckbahnen [81]. Sie erweiterten das System um eine automatische Anpassung der Materialflussrate in Abhängigkeit von der erkannten Fehlerklasse. Zum Trainieren des Klassifikators verwendeten sie das vortrainierte Modell ResNet50. Um die geforderte Ausgabegröße von drei zu erreichen, modifizierten sie die letzte Schicht. Das System hat eine

## 2. Stand der Forschung

sehr hohe Genauigkeit von 98% und kann die erkannten Fehler entsprechend korrigieren. Jin *et al.* und Banadaki *et al.* verwendeten für das Transfer-Lernen die Modelle ResNet50 und Inception-3 für die Bildklassifikation [81, 79]. Wie Wang *et al.* herausfand, sind diese enorm tiefen Modelle (Inception-3 42 Layer, ResNet50 50 Layer) mit 1000 zu erkennenden Klassen sehr generalisiert [76]. Transferlernen funktioniert solange die zu erfüllenden Aufgaben der bisherigen ähnelt. Ob die Aufgabe, zwischen 1000 Klassen zu unterscheiden, und die Aufgabe, Fehler am Druckobjekt zu erkennen, hinreichend ähnlich sind, hängt auch von der Art und Anzahl der jeweils verfügbaren Trainingsdaten ab [82].

### Fazit

Die optische Fehlererkennung in der additiven Fertigung ist bisher neben der eigentlichen Materialforschung noch das mit am intensivsten untersuchte Gebiet. Allerdings ist durch die Anzahl an Druckverfahren und die Möglichkeit der Auswertung mit Hilfe von Algorithmen eine Vielzahl an Kombinationen möglich. Zudem kommt noch hinzu, dass je nach Druckertyp und -hersteller manche Verfahren nicht auf andere Drucker übertragbar sind oder speziell auf proprietäre Schnittstellen angewiesen sind. Die vorgestellten Arbeiten befassen sich nahezu ausschließlich mit der Drucktechnologie des Fused Filament Fabrication (FFF) 3D-Drucks. Durch die großen Unterschiede zwischen verschiedenen Druckverfahren und Druckern, ist es bisher nicht möglich einen allgemeinen Ansatz für alle Anwendungen zu erstellen.

Im Zuge dieser Arbeit soll ein Verfahren entwickelt und evaluiert werden, das innerhalb eines Druckverfahrens, der Fused Filament Fabrication, auf verschiedene Drucker übertragen werden kann. Zudem soll auch eine Übertragbarkeit auf andere Druckverfahren untersucht werden, dem Pulverbettdruck.

### 2.3.2. Akustische Überwachung des Druckprozesses

Einige der typischen Fehler im Prozess basieren auf mechanischer Einwirkung und sind somit hörbar für den Menschen. Mit Hilfe von Audiosensoren und aktuellen Methoden der Datenverarbeitung und des maschinellen Lernens können diese Fehler erkannt und der Benutzer benachrichtigt werden, sodass er entsprechend reagieren kann. Auf kritische Fehler kann auch sofort korrigiert werden, um Schäden oder Verschleiß am Drucker, Material- und Zeitverschwendung zu vermeiden.

Die Erkennung von Anomalien mit Hilfe akustischer Daten ist bereits ein aktuelles Forschungsgebiet. Im Vergleich zur visuellen Introspektion ist sie jedoch noch eine Nische. Erste moderne Ansätze verwenden Gaussian Mixture Models (GMM), um verschiedene Schussarten von Schusswaffen zu klassifizieren, mit dem Ziel, ein leistungsfähiges Überwachungssystem zu schaffen. Da Clavel *et al.* [83] dafür keine realistischen Daten hatten, erzeugten sie diese künstlich mit dem Prinzip der Data Augmentation. So mischten sie Schüsse und Umgebungsgeräusche zu einem Datensatz zusammen. Damit bildeten sie Datensätze mit unterschiedlichem Signal-Rausch-Verhältnis. Als Merkmale verwendeten sie die ersten acht

## 2. Stand der Forschung

Mel-Frequenz-Cepstral-Koeffizienten (MFCC) der einzelnen Samples, die Lautheit, die Schwerpunktswellenlänge und die ersten beiden Ableitungen dieser drei Merkmale. Metriken waren die False Rejection Ratio und die False Detection Ratio, für die sie mit ihrem System niedrige Werte erhielten.

Zhuang *et al.* verwenden ebenfalls GMM in Kombination mit Support Vector Machines (SVM) [84]. Als zweiten Klassifikator verwenden sie eine Kombination aus Multi-Layer Perceptrons (MLP) und Hidden Markov Models (HMM). Sie erreichen eine 45% bessere relative Leistung auf einem damals aktuellen Datensatz zur akustischen Ereigniserkennung (AED) als Methoden, die automatische Spracherkennungsmerkmale wie Mel-Frequenz-Cepstral-Koeffizienten und Filterbanken verwenden. Mit dieser hybriden Architektur erreichen sie einen F-Score von 41,2%.

Im Jahr 2014 verwendeten Rakotomamonjy *et al.* eine SVM für einen Datensatz von 19 Klassen. Da SVMs allein keine nicht-binäre Klassifizierung durchführen können, wurde ein Anywhere Against Anywhere-Ansatz verwendet. Bei diesem Ansatz werden  $\frac{k(k-1)}{2}$  SVM-Klassifikatoren erstellt, wobei jede dieser SVMs eine Unterscheidung zwischen zwei verschiedenen Klassen trifft [85]. Der Wilcoxon Sign Rank Test bescheinigt ihrer Methode eine hohe Funktionalität.

Der ESC-50-Datensatz besteht aus etwa 2000 annotierten Proben, die in 50 Klassen eingeteilt sind [86]. ESC steht für Environmental Sound Classification und kennzeichnet den Inhalt des Datensatzes, der aus verschiedenen Umgebungsgeräuschen besteht. Die ESC-10 ist eine Teilmenge davon. Sie umfasst zehn Klassen, die sich in drei Gruppen unterteilen lassen. Zu diesen Gruppen gehören akustische Ereignisse mit ähnlicher zeitlicher Struktur (z. B. Hundegebell), akustische Ereignisse mit starkem harmonischem Inhalt (z. B. Babygeschrei) und weniger strukturierte akustische Ereignisse (z. B. Hubschrauber, knisterndes Feuer). Der Urbansound-8K [87] besteht aus 10 Klassen und 8732 Proben verschiedener, typischer Geräusche, die in Städten in den USA auftreten. Dazu gehören Motorengeräusche, Pressluftschlämmer, Schüsse usw. Da Recurrent Neural Networks (RNN) in der automatischen Spracherkennung attraktive Ergebnisse liefern, wurde auch mit diesen Architekturen im AED experimentiert.

2017, Lundberg *et al.* [88] wählten einen anderen Ansatz. Sie wandten die modernste Technik der Bilderkennung, Convolutional Neural Networks (CNN), auf den Bereich der AED an. Sie extrahierten MFCCs, die Spektrogramme und Cross-Recurrence-Plots als Merkmale und gaben sie den CNN-Architekturen GoogLeNet [78] und AlexNet [89] als Input. Als Datensätze wurden der ESC-10, ESC-50 und der Urbansound-8K verwendet. Jedes der Merkmale wurde einzeln und in gemeinsamer Kombination aller drei verwendet. Diese Kombination wurde durch die Verwendung der Merkmale als verschiedene Farbkanäle desselben Bildes hergestellt. Mit den Spektrogrammen allein erzielten sie bei allen drei Datensätzen die besten Ergebnisse, aber eine Kombination brachte keinen weiteren Vorteil.

Vor allem die Long Short-Term Memory (LSTM) Modelle werden am häufigsten verwendet. Virtanen *et al.* [90] verwendeten LSTMs mit drei versteckten Schichten und je 32 Neuronen auf dem DCASE-2016-Datensatz und fanden damit eine wesentliche Verbesserung gegenüber GMM und MLP. Der DCASE-2016-Datensatz [91] wurde für die DCASE 2016 Challenge erstellt und verwendet. DCASE steht für Detection and Classification of Acoustic Scenes and

## 2. Stand der Forschung

Events (Erkennung und Klassifizierung akustischer Szenen und Ereignisse). Es besteht aus Umgebungsgeräuschen. Zusätzlich zu diesem Ansatz wurden auch Stereo- und Nicht-Mono-Aufnahmen verwendet. Dies führte ebenfalls zu einer Verbesserung der Ergebnisse.

In einem aktuellen Ansatz für FFF-Druck verwenden Wu *et al.* [92] ein Hidden Semi Markov Model (HSMM), um die verschiedenen Maschinenzustände zu erkennen - fehlerhafte (anormale) und nicht fehlerhafte Zustände. Da viele Drucker in der additiven Fertigung nicht über Sensoren zur Überwachung dieser Zustände verfügen, bringen sie einen akustischen Sensor am Extruder selbst an. Dieser Sensor misst die akustische Emission (AE). AE bezieht sich auf das Phänomen, bei dem elastische, transiente Wellen durch die Freisetzung von Energie in Materialien erzeugt werden. Dies wurde in vielen Anwendungen erfolgreich zur Überwachung der Lebensdauer von Geräten eingesetzt. Bei ihrem Ansatz führen sie zunächst eine Merkmalsextraktion aus den AE-Daten durch. Um die Anzahl der Merkmale zu reduzieren, führen sie eine Analyse segmentweise durch. Bei der Analyse wird das Signal in mehrere Intervalle mit einer bestimmten Länge unterteilt. Aus jedem dieser Intervalle werden die Merkmale extrahiert, deren Mittelwert berechnet und als neue Merkmale für die anschließende Principal Component Analysis (PCA) verwendet. Diese reduzierten Merkmale werden sowohl als Trainingsdaten für die HSMM als auch als Testdaten verwendet. Im Einzelnen wird in diesem Beitrag der Kompromiss zwischen Rechenzeit und Erkennungsrate diskutiert. Während eine hohe zeitliche Auflösung der einzelnen Samples eine höhere Erkennungsrate ermöglicht, leidet die Rechenzeit darunter, was den geforderten Echtzeitnutzen verringert.

In der Arbeit von Eschner *et al.* [93] wurde ein piezoelektrischer Sensor unterhalb des Druckraums beim Selective Laser Melting (SLM) Druckprozess integriert. Durch die Positionierung wird zum einen der Laser beim eigentlichen Druck nicht beeinträchtigt. Gleichzeitig bietet der direkte Kontakt zum Druckobjekt durch die Grundplatte eine Möglichkeit den Körperschall ohne weitere Beeinflussung durch Medienübergänge aufzuzeichnen. Der Fokus der Arbeit liegt auf der Korrelation zwischen Energiedichte beim Druck und Bauteilqualität bzw. -porosität an Hand akustischer Datenaufnahme. Die vier Parameter Laser Power, Scan Geschwindigkeit, Schichtdicke, sowie der Abstand der einzelnen Laserlinien (hatching distance) beeinflussen den Energieeintrag im Material entscheidend und bilden daher die Basis ihres Tests. Als Grundlage ihrer akustischen Erkennung ermittelten sie durch Variation der Parameter 18 Testdrucke. Diese werden anschließend mit Hilfe von Computertomographie und dem Archimedischen Prinzip evaluiert. Zusammen mit den akustischen Daten, sowie den Druckparametern wird dann ein neuronales Netz trainiert, das in der Lage ist die häufigsten Fehler zu klassifizieren: Überhitzung, Verklumpung (balling) und Porenbildung. Mit diesem Ansatz ist es ihnen möglich eine Klassifizierungsgenauigkeit von 93% zu erreichen.

In einer weiteren Arbeit untersuchten Eschner *et al.* [94] die Korrelation zwischen akustisch aufgenommenen Daten im Druckprozess (L-PBF) und Dichte des fertigen Bauteils. Auch hier kam ein Sensor unterhalb des Druckraums zum Einsatz, der direkt den Körperschall messen kann. Ein Unterschied zur vorhergehenden Veröffentlichung ist, dass nun nicht mehr nur einfache Würfel gedruckt werden, sondern die Würfel wurden auch auf komplexe Strukturen gedruckt, siehe Abbildung 2.6. Dies ist nötig um die Beeinflussung der akustischen Daten durch

## 2. Stand der Forschung

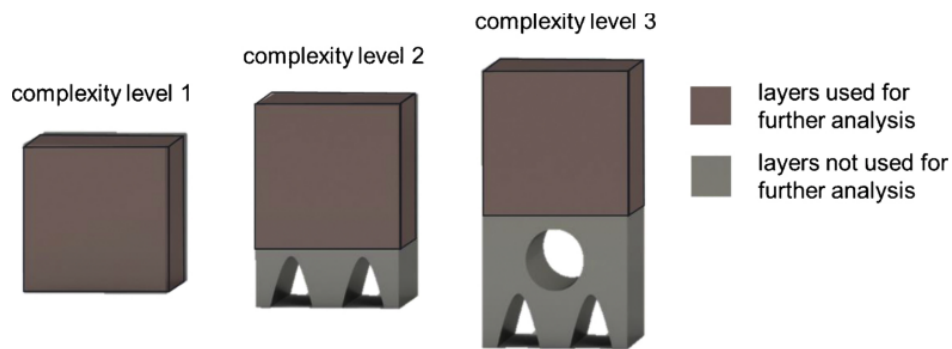


Abbildung 2.6.: Übersicht über die drei Komplexitätsklassen, die gedruckt wurden. Untersucht wurden letztendlich allerdings nur die Schichten oberhalb der zusätzlichen Geometrien, um eine Vergleichbarkeit herzustellen. Quelle [94].

die Bauteilgeometrie zu untersuchen. Als akustischer Sensor kam ein masseloser Piezokeramik-Sensor von QASS zum Einsatz. Mit Hilfe eines Multi Layer Perceptron (MLP) wurden die einzelnen Schichten, sowie die dazugehörigen Spektrogramme ausgewertet. Damit war es möglich einen F1-Score von 88% zu erreichen und das zeigt, dass eine Korrelation zwischen akustischen Signalen und Bauteilqualität vorhanden ist.

Die Autoren Koupryanoff *et al.* [95] nutzen statt dem Körperschall den Luftschall und nahmen akustische Daten im Druckraum eines Laser Powder Bed Fusion (L-PBF) Druckers mit Hilfe eines Mikrofons auf. Der Fokus lag hierbei auf dem Zusammenhang zwischen Schalldruckpegel, dem Frequenzbereich und die Form einer einzelnen, gedruckten Linie. Über eine einfache Vorverarbeitung der aufgenommenen Audiodaten war es ihnen anschließend möglich Probleme bei der Schichtdicke zu identifizieren, wie z.B. unebene oder zu dicke Schichten. Hierfür wurde eine Verarbeitungspipeline entwickelt, die das aktuelle akustische Signal mit verschiedenen Referenzdaten innerhalb eines Modells vergleicht, um auf dieser Basis zu entscheiden, ob im aktuellen Druckprozess Fehler vorliegen. Unterschieden werden können hier die Fehlerklassen „Kein Pulver“, „Dicke Pulverschicht“ und „Optimal“.

### Fazit

Die Untersuchung von akustischen Daten mit Hilfe von neuronalen Netzen wird hauptsächlich in der Spracheingabe und -verarbeitung eingesetzt, sowie auf Daten aus dem öffentlichen Raum. Die Anzahl an frei verfügbaren akustischen Datensätze aus Druckprozessen ist gering. Im Bereich des 3D-Drucks haben sich bisherige Verfahren eher auf die Verarbeitung von optischen Signalen fokussiert. Dies führt auch dazu, dass generell im Bereich der Anomalieerkennung und im Besonderen in der additiven Fertigung bisher hierzu nur wenig publiziert wurde.

In dieser Arbeit sollen akustische Emissionen im 3D-Druck näher untersucht werden und die entsprechenden Signale ausgewertet werden.

## 2.4. Automatisierte Entnahme aus einem 3D-Drucker

Zur vollständigen Automatisierung des 3D-Druckprozesses ist der Entnahmeprozess zu automatisieren [96].

Die Notwendigkeit der Prozessautomatisierung wird auch durch eine 2018 veröffentlichte Studie des Umweltbundesamtes [97] unterstrichen. Denn einige Schadstoffe können während des Druckprozesses austreten und eine manuelle Überwachung des Prozesses, selbst bei kleinen Losgrößen, ist nicht nur zeitaufwändig, sondern kann sogar ein Gesundheitsrisiko für die Bediener darstellen.

Um die Möglichkeiten und die Effizienz des 3D-Drucks durch Automatisierung weiter zu verbessern, haben einige Hersteller von 3D-Druckern mit der Umsetzung von (teil-)automatisierten Druckern begonnen. Dazu gehören die 3D-Druck-Unternehmen Stratasys Ltd. (mit dem Continuous Build Demonstrator), Formlabs (mit der Form Cell) [19] und Digital Metal (mit der „No-Hand“-Produktionslinie).

Der Continuous Build Demonstrator wurde für den kontinuierlichen Druck ohne die Ausfallzeiten, die durch das Warten auf einen Mitarbeiter entstehen, konzipiert. Um dies zu ermöglichen, hat Stratasys einen FFF-3D-Drucker entwickelt, der mit einem Förderband als Bauplatte ausgestattet ist [98]. Auf das Förderband wird eine durchsichtige Kunststoffolie gelegt, auf die das Objekt gedruckt wird. Nach Abschluss des Druckvorgangs wird die Folie zur Vorderkante der Bauplatte befördert, bis sich das gedruckte Objekt nicht mehr im Bauvolumen befindet. Danach wird die Folie abgeschnitten, sodass das gedruckte Objekt mit der Folie in einen Auffangbehälter fällt. So wird das Bauvolumen des 3D-Druckers geräumt und der nächste Druckauftrag kann beginnen.

Mit dem Projekt Form Cell hat Formlabs ebenfalls einen 3D-Drucker entwickelt, der automatisiert bedient werden kann [19]. Ähnlich wie Stratasys hat Formlabs eine abnehmbare Bauplatte eingeführt, um die Produktionseffizienz zu erhöhen und den Maschinenstundensatz zu senken. Die Anwendung wird mit mehreren SLA-3D-Druckern des Unternehmens und einem Industrieroboterarm realisiert. Dieser Arm wird verwendet, um die Bauplatte der fertigen Drucke aus ihrer Halterung zu entfernen und sie zur nächsten Nachbearbeitungsstation zu transportieren. So lässt sich die Kapazität der Zelle nahezu beliebig erweitern.

Auch der schwedische Metall-3D-Druck-Hersteller Höganäs Group Digital Metal hat eine automatisierte additive Fertigungsprozesskette angekündigt [31]. Durch den Einsatz eines Industrieroboterarms werden die 3D-Drucker mit einer mit Metallpulver gefüllten Baukastens versorgt, um die Produktion zu starten. Nach Beendigung des Binder-Jetting-Prozesses bereitet der Roboter die Entnahme des Baukastens vor und transportiert ihn zu verschiedenen Stationen zur Nachbearbeitung. Dazu gehören die Entfernung des Pulvers, das Abbinden des Kunststoffes und das abschließende Sintern. Nach der Entnahme des Baukastens kann ein neuer Baukasten mit dem gewünschten Material in den 3D-Drucker eingesetzt werden, sodass die Maschinenstillstandszeit sehr gering ist.

## 2. Stand der Forschung



Abbildung 2.7.: Der Creality CR-30 hat ein integriertes Förderband, welches als Build Plate dient. Somit ist es möglich Objekte beliebige Länge zu drucken. Gleichzeitig werden abgeschlossene Drucke automatisch vom Druckbett entfernt, da sie nach Erreichen des Ende des Druckbetts sich dort automatisch ablösen. Quelle [99].

Ein weiteres Unternehmen, das sich der Bewegung zur Automatisierung des 3D-Drucks angeschlossen hat, ist das 2017 gegründete Unternehmen Voodoo Manufacturing. Ihr Ziel ist es, den 3D-Druck zu nutzen, um dem Spritzguss kleinerer Mengen Konkurrenz zu machen. Um dies zu ermöglichen, hat Voodoo Manufacturing das Projekt Skywalker gestartet, bei dem ein Industrieroboterarm den fertigen Druck aus dem Bauvolumen für mehrere 3D-Drucker entfernt. Dazu wird eine abnehmbare Bauplatte verwendet, die nach Abschluss eines Druckauftrags mit Hilfe des Roboterarms aus dem Drucker herausgezogen wird. Nach der Entnahme aus dem Bauvolumen wird die Bauplatte zusammen mit dem daraufhin gedruckten Objekt auf ein Förderband gelegt, um zur Nachbearbeitung transportiert zu werden. Eine neue Bauplatte wird in den Drucker eingesetzt und der nächste Druckauftrag kann gestartet werden.

Es gibt einige weitere Unternehmen, die sich mit der Automatisierung des 3D-Drucks befassen, darunter 3D Systems mit seinem Produkt Figure 4 [32], das auf den selben Prinzipien wie die Form Cell von Formlabs beruht. Die meisten von ihnen verfolgen jedoch den selben Ansatz für die automatische Entfernung des 3D-Druckobjekts, indem sie die gesamte Bauplatte entfernen. Bei diesem Ansatz wird das Bauvolumen des Druckers freigegeben, und ein neuer Druckauftrag kann erteilt werden, wodurch die Ausfallzeit des Druckers erheblich reduziert wird. In diesen Fällen wird eine andere Bauplatte eingesetzt, was jedoch zu Problemen wie Kalibrierungs-



## 2. *Stand der Forschung*

fehlern führen kann. Durch die Entfernung der Bauplatte wird der Nachbearbeitungs- und Überwachungsaufwand durch einen Mitarbeiter nicht verringert, da das gedruckte Material von Hand von der Bauplatte entfernt werden muss. Daher sind die einzigen Vorteile der oben genannten Ansätze die Reduzierung der Druckkosten und der Ausfallzeiten des Druckers.

### **Fazit**

Bisher ist die automatisierte Entnahme hauptsächlich von Druckerherstellern vorgestellt worden. Die meisten Konzepte nutzen jedoch nicht direkt einen Roboterarm, sondern entnehmen das komplette Druckbett beziehungsweise setzen ein neues Druckbett ein. Jedoch ist das nicht mit jedem Druckprinzip möglich. Die Robotik kann dagegen mit flexiblen Ansätzen gerade bei solchen Anwendungsfällen ihre Stärken ausspielen. Obwohl einige Hersteller solche Verfahren bisher vorgestellt haben, sind viele davon nur prototypisch getestet, nicht aber kommerziell verfügbar. Hier besteht also offensichtlich noch Forschungsbedarf, um einen zielführenden Ansatz zu entwickeln und auf den Markt zu bringen.

In dieser Arbeit soll daher näher untersucht werden, wie eine zuverlässige Entnahme möglich ist.

## 2.5. Automatisierte Nachbearbeitung gedruckter Objekte

Robotersysteme in größeren Industrieanlagen sind heute fast üblich. Vor allem, wenn die Menge der hergestellten Produkte eine bestimmte Schwelle überschreitet, ist ein Roboter rentabel. Bei kleineren Produktionen kann die Programmierung und das Einlernen eines Roboters im Vergleich zur manuellen Arbeit jedoch leicht einen Mehraufwand darstellen. Die Planung und flexible Anpassung von Robotertrajektorien ist immer noch ein aktuelles Forschungsthema mit vielen Herausforderungen [35, 100]. Es gibt mehrere Methoden, um passende Trajektorien zu erreichen, wie Programmierung durch Vormachen [101, 102], instruktive oder lernende Systeme [103, 104] sowie eine Kombination dieser drei [105]. Neu verfügbare Technologien werden ebenfalls eingesetzt um die Art und Weise zu verbessern wie ein Benutzer Trajektorien teachen kann, wie zum Beispiel virtuelle und erweiterte Realität [106, 107] oder haptische Geräte [108].

Auch die Bahnplanung ist ein wichtiges Forschungsthema in der Zerspanung. In einem Artikel aus dem Jahr 2012 geben Chen und Dong einen Überblick über die aktuelle Forschung im Bereich der robotergestützten Bearbeitung [37]. Sie identifizieren drei Hauptthemen: Entwicklung von robotergestützten Bearbeitungssystemen, Bahnplanung für Bearbeitungsroboter und Analyse und Kompensation von Vibrationen und Schwingungen. Die Autoren nennen die Ungenauigkeit von Roboterarmen als die größte Schwäche. Industrielle Bearbeitungsaufgaben können strenge Toleranzen aufweisen, die für Roboter derzeit unerreichbar sind. Besonders deutlich wird dies bei Fräsanwendungen. Die geringe Steifigkeit von Roboterarmen im Vergleich zu klassischen Computerized-Numeric-Control (CNC)-Maschinen lässt sich auf die serielle Bauweise zurückführen. Dies kann auf die intrinsische Nachgiebigkeit von Zahnrädern und anderen mechanischen Komponenten jedes Robotergelenks zurückgeführt werden [109]. Andererseits sehen Chen und Dong die Flexibilität von Roboterarmen mit vielen Freiheitsgraden als einen großen Vorteil. Sie ermöglicht die Bearbeitung von komplexen Strukturen und nennen Objekte mit innenliegenden Merkmalen als eine Anwendung, bei der das Roboterfräsen traditionelle Methoden übertreffen kann. Für Aufgaben, die nicht mit strengen Toleranzen verbunden sind, sind Roboter bereits ein brauchbares Werkzeug. Chen und Dong nennen das Polieren, Schleifen und Entgraten als Aufgaben, bei denen Roboter nachweislich gute Leistungen erbringen. Roboter erreichen in diesen Bereichen eine hohe Arbeitsqualität, da sie in der Lage sind, ein Werkzeug am Ende des Arms auf die Oberflächennormale der bearbeiteten Oberfläche auszurichten.

Im Jahr 2019 veröffentlichten Ji und Wang einen Überblick über die industrielle Roboterbearbeitung, der mehr als 30 Jahre Forschung umfasst [38]. Sie konzentrieren sich dabei auf die Bearbeitung harter Materialien. Die bisherige Forschung zur Bearbeitung mit geringem Materialabtrag wie Polieren und Entgraten zielt hauptsächlich darauf ab, eine menschenähnliche Bedienung und eine hohe Oberflächenqualität zu erreichen. Bis heute sind die Stabilität des Roboters und die Bearbeitungsvibrationen ein wichtiges Forschungsthema für Prozesse mit hoher Zerspanungsrate wie das Fräsen. Die Autoren sehen die Notwendigkeit weiterer

## 2. Stand der Forschung

Verbesserungen, bevor die industrielle Roboterbearbeitung mit den etablierten CNC-Maschinen konkurrieren kann.

### **Fazit**

Die automatisierte Nachbearbeitung von additiv gefertigten Bauteilen ist bisher eine Nische in der Forschung. Im Bereich des Fräsens, der subtraktiven Fertigung, wurden bereits viele Untersuchungen durchgeführt, inwiefern dies mit Hilfe eines Roboters umsetzbar ist. hnliche Untersuchungen gibt es bisher nicht für die additive Fertigung. Dies macht deutlich, dass hier noch Potenzial für Verbesserung besteht.

Neben der Überwachung und der Entnahme des gedruckten Objektes ist die Nachbearbeitung ein sehr aufwändiger Prozess. Wird bei der Nachbearbeitung das Objekt beschädigt, muss es erneut gedruckt werden, daher ist hier besondere Vorsicht geboten. Im Rahmen dieser Arbeit werden zwei verschiedene Ansätze untersucht, die sich mit dem automatisierten Entfernen von Support Strukturen auseinandersetzen.

Auch wenn die additive Fertigung bereits mehrere Jahrzehnte produktiv eingesetzt wird, ist die Arbeit an automatisierten Verfahren bisher überschaubar. Sowohl im Bereich der Prozessbeobachtung, der automatisierten Entnahme oder Nachbearbeitung sind nahezu keine etablierten Verfahren bekannt oder gar Standards etabliert. Im Zuge dieser Arbeit soll daher die komplette Prozesskette der additiven Fertigung untersucht werden, um den Prozess insgesamt zu verbessern, sowie weiter zu automatisieren.

# 3. In-situ Fehlererkennung im Druckprozess

Durch die Komplexität des Druckverfahrens können während des Druckvorgangs verschiedene Fehler auftreten. Eine Schichtverschiebung tritt zum Beispiel auf, wenn eine der Achsen-Positionen im Druck gestört wird und die daraufhin gedruckte Struktur zu einer Seite verschoben wird. Dies kann zum Beispiel vorkommen, wenn die Zahnriemen der Achse zu lose oder zu fest sind. Dadurch entsteht ein Schlupf und die berechnete Motorumdrehung zur Bewegung des Extruders stimmt nicht mehr mit der realen Bewegung überein. Ein weiterer möglicher Fehler ist das Verstopfen der Düse, was dazu führt, dass die Extrusion während des Drucks stoppt und das Objekt nicht weitergeführt wird. Beide Druckfehler führen unabhängig zu einem mangelhaften Objekt und kann diese so nicht für den beabsichtigten Anwendungsfall verwendet werden.

Diese und weitere Fehlerfälle können auf verschiedene Arten im Druckprozess erkannt und verarbeitet werden. In den nächsten Kapiteln werden drei verschiedene untersucht, die sich jeweils voneinander in ihrer Verarbeitungsart oder der eingesetzten Sensorik unterscheiden.

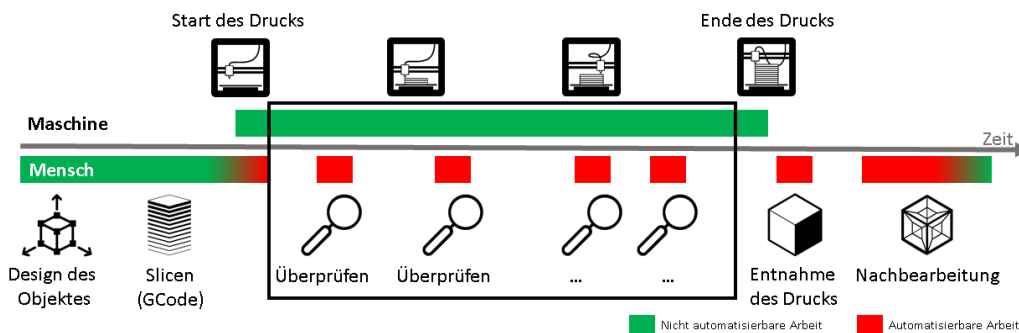


Abbildung 3.1.: In diesem Abschnitt der Arbeit liegt der Fokus auf der Überwachung des Druckprozesses live während des Prozesses (schwarzer Kasten). Hierfür werden verschiedene Verfahren untersucht und angewandt, um Fehler im Prozess frühzeitig zu klassifizieren.

## 3.1. Optische Fehlererkennung mit klassischer Bildverarbeitung

Ziel der optischen Fehlererkennung ist es, einen möglichen Fehler möglichst zeitnah zu identifizieren. Nur so besteht die Möglichkeit auf entsprechende Fehler zu reagieren und zum Beispiel Parameter zu ändern oder den Druck ohne großen Material- und Zeitverlust abzubrechen. In dieser Untersuchung wird der 3D-Drucker Raise Ideamaker eingesetzt, siehe Abb. 3.5. Dieser bietet den Vorteil, dass der Extruder sich ausschließlich auf der XY-Ebene bewegt und dadurch eine Anordnung der Kameras möglich ist, so dass sie immer auf der selben Höhe wie der Extruder ist. Mehrere Kameras aus der Intel RealSense-Produktreihe werden so angeordnet, dass bestimmte Teile des Druckers im Fokus liegen. Eine Kameraposition ist im Drucker auf Höhe der Drucknozzle, da auf dieser Höhe der entscheidende Teil des Druckprozess stattfindet und das Objekt hier an Größe gewinnt. Eine andere ist auf Höhe der Druckplatte, um so das Objekt auf Druckbettebene zu überwachen. Sowohl die Überwachung der Extruderebene, als auch der Druckbettebene sind entscheidend für eine Fehlererkennung. Nur dort können Fehler auftreten, da hier jeweils Übergänge zur neuesten Druckebene oder zur Druckerhardware vorhanden sind. Als Basis für diesen Ansatz zur Bildverarbeitung und Fehlererkennung wird die Open-Source-Bildverarbeitungsbibliothek OpenCV [110] verwendet.

### 3.1.1. Vorverarbeitung der Bilddaten

Um eine sinnvolle Weiterverarbeitung der zu prüfenden Bilder und eine möglichst geringe Rechenlast zu gewährleisten, werden die aufgenommenen Bilder zunächst für jeden der Fehlerfälle in Graustufenbilder (Schritt 1 in Abb. 3.2) umgewandelt. Vor dieser Umwandlung wird das Bild jedoch mit einem Gauß-Filter geglättet, was das Bildrauschen reduziert und eventuell auftretende Pixelfehler eliminiert sowie kleine Strukturen im Bild generell unterdrückt [111].

Im nächsten Schritt (Schritt 2 in Abb. 3.2) wird mit Hilfe des Sobel-Filters das Bild auf die darin enthaltenen Kanten reduziert. Dieser Filter wird verwendet, um die im Bild vorhandenen Informationen auf das absolut Notwendige zu beschränken. Da für die Erkennung der relevanten Druckfehler nur die Konturen des bedruckten Objekts relevant sind, ist es ausreichend, wenn nur diese im Bild erkennbar sind. Ein weiterer Vorteil der Umwandlung basierend auf dem Sobel-Filter ist, dass das resultierende Kantenbild unabhängig von der Beleuchtungsstärke ist. Dies ist der Fall, weil die Helligkeitsgradienten Unterschiede darstellen, die sich bei einer Änderung der Gesamtbeleuchtung des Bildes in ihrer Größe nicht ändern, im Gegensatz zum Helligkeitswert der einzelnen Pixel. Daher können Bilder, die unter unterschiedlichen Helligkeitsbedingungen aufgenommen wurden, mit denselben Algorithmen behandelt und sogar miteinander verglichen werden. So wird sichergestellt, dass die Kanten möglichst einfach erkannt und die Bilder möglichst einfach verarbeitet werden.

Anschließend wird im Verarbeitungsschritt 3 das Wissen über die Nozzle, in Form eines auf Kanten reduzierten Bildes, verwendet, um die beiden Nozzles im vorliegenden Bild zu finden. Dies ermöglicht das Auffinden der aktuellen Druckebene, da die untere Spitze der Nozzle immer

### 3. In-situ Fehlererkennung im Druckprozess

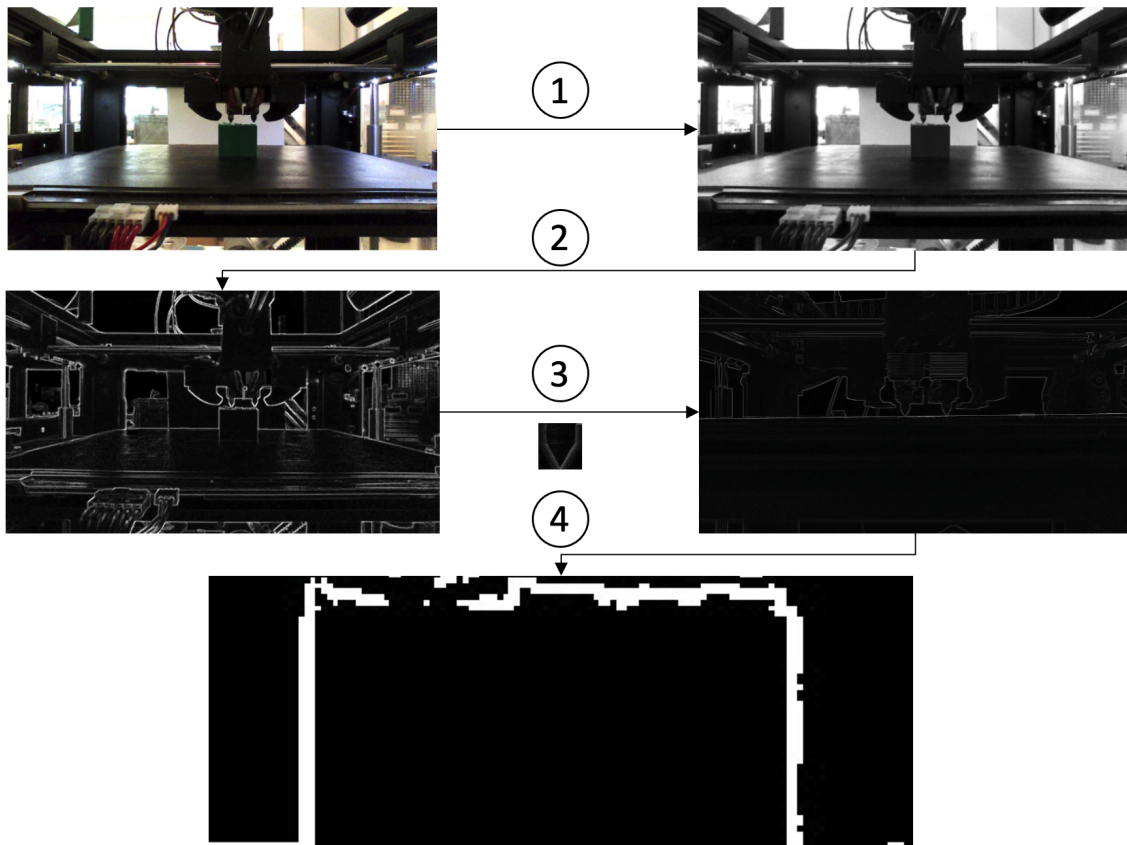


Abbildung 3.2.: Übersicht über die einzelnen Vorverarbeitungsschritte um den passenden Bildausschnitt im Bild der Kamera zu finden. 1) Umwandlung zu Graustufen-Bild, 2) Anwendung Sobel-Filter, 3) Suche nach Nozzle im Bild mit Detailbild von Nozzle, 4) Ausschneiden der Region of Interest unterhalb der Nozzle-Position.

auf der obersten Ebene des gedruckten Objektes ist. Abschließend werden die Helligkeitswerte normalisiert, um den ganzen Wertebereich (0 bis 255) des Schwarzweißbildes abzudecken, so dass es eine klare Unterscheidung zwischen den einzelnen Pixeln gibt.

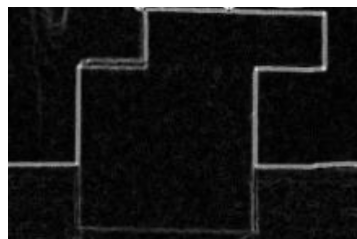


Abbildung 3.3.: Fehlerbild von einem Layershift. Dieser führt zu einer Verschiebung der einzelnen Schichten ab einem bestimmten Zeitpunkt. Quelle [192].

**Bestimmung der Region of Interest (ROI)** Für die weitere Verarbeitung der Bilder ist es ratsam, nur die Bildbereiche auszuwerten, in denen mögliche Fehler zu beobachten sind. Dies hat neben der Verkürzung der Laufzeit der Algorithmen auch den Vorteil, dass die Informationen auf das Notwendige reduziert werden kann, da durch den Zuschnitt der Bilder auf eine ROI viele Kanten im Hintergrund wegfallen, die sonst von den Fehlererkennungsalgorithmen fehlinterpretiert werden könnten.

Da die im System verwendeten Kameras unterschiedliche Bereiche des Drucks beobachten, unterscheidet sich die Bestimmung der ROI zwischen den Kameras auf Druckkopfebene von der auf Plattenebene.

Bei den Kameras, die auf der Druckplatte montiert sind, befindet sich der Druck in einer festen Höhenposition im Sichtfeld der Kamera. Die Höhe der ROI wird als Argument innerhalb der Software übergeben, da sie so an den jeweiligen Fehlerfall angepasst werden kann.

Bei Kameras, die auf Druckkopfhöhe angebracht sind, ist das Schneiden jedoch komplizierter, da die Position des Objekts, und damit dessen Oberkante, relativ zur Kamera nicht bekannt ist. Neben Fehlern im Druckprozess kann auch die Positionierung der Kamera am Drucker zu Abweichungen führen. Die Kenntnis der theoretischen Position der obersten Druckkante im Bild ist jedoch elementar für die Erkennung mehrerer auftretender Fehler.

Um diese Position exakt zu bestimmen, wird die Methode des Template Matching eingesetzt. Der Extruder wird mit Hilfe des Templates im Bild gesucht. Sowohl das Template, als auch das aktuell zu verarbeitende Bild, liegt vorverarbeitet vor. Für das Matching werden die Helligkeitsunterschiede der Pixel des Originals und des gleich großen Bereichs im Bild aufsummiert. Anschließend werden die berechneten Differenzen nach den kleinsten Abweichungen, d.h. den wahrscheinlichsten Positionen, gefiltert und aus diesen Positionen eine Gerade erstellt.

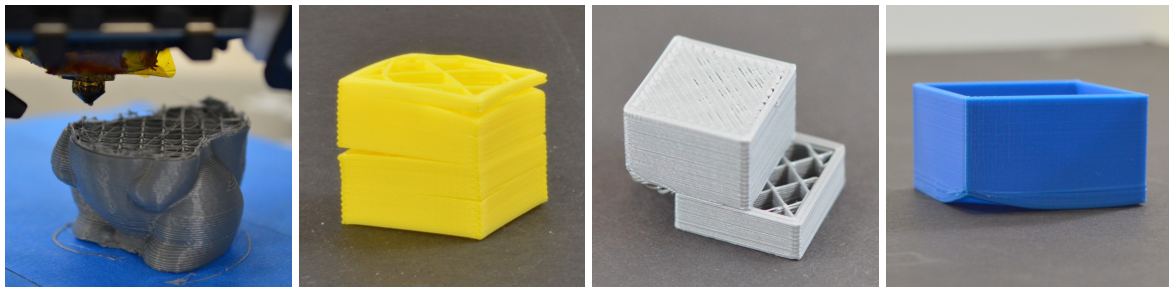
#### 3.1.2. Erkennung der verschiedenen Fehlertypen

Da mehr als zwanzig verschiedene Fehlerfälle im FFF Druckprozess möglich sind [112], wurden für diese Untersuchung die folgenden vier ausgesucht. Grundlage hierfür war, dass die Fehler zum einen optisch erkannt werden können und zum anderen sie sehr häufig in bisherigen Druckjobs aufgetreten sind. Gleichzeitig ist es auch möglich, die Fehler künstlich, z.B. durch die Anpassung der Druckparameter oder des G-Codes, zu erzeugen. Dies ermöglicht eine wiederholbare Untersuchbarkeit mit der Sicherheit, dass der Fehler innerhalb dieses Druckjobs auftreten wird. Dieses Wissen über das Auftreten des Fehlers ist natürlich dem entsprechenden Auswertungsalgorithmus nicht bekannt. Die Auswertung basiert rein auf Bilddaten der eingesetzten Sensoren.

### 3. In-situ Fehlererkennung im Druckprozess

Folgende vier Fehler werden erkannt:

- **Extrusion Stop** Im Druckvorgang wird kein weiteres Filament mehr aus der Nozzle gefördert, sodass das aktuelle Objekt nicht durch weitere Schichten weitergedruckt wird. Ursachen können zum Beispiel eine verstopfte Nozzle, falsche Temperatureinstellungen oder fehlendes Filament sein, siehe Abb. 3.4a.
- **Layer Separation** Die verschiedenen Druckschichten haften vor allem dann untereinander, wenn die eingestellte z-Höhe korrekt ist und die einzelnen Schichten die Chance haben miteinander zu verschmelzen. Ist dies nicht der Fall, ist es möglich, dass sich die gedruckten Schichten durch Verformung beim Abkühlen wieder voneinander lösen, siehe Abb. 3.4b.
- **Layer Shifts** Die Bewegungen des Druckkopfs sind durchgehend relativ. Zu Beginn eines Druckauftrags kalibriert sich der Drucker an allen Achsen auf einen Nullpunkten und von dort aus wird nur relativ verfahren. Ist die Hardware nicht exakt kalibriert oder verschlissen z.B. durch zu schwach gespannte Zahnriemen entlang der Achsen, so kann es passieren, dass der Druckkopf nicht an der Achsposition stoppt, an der er stoppen sollte. Dadurch entsteht eine Versatz auf der Achse, was sich in Form einer Verschiebung des kompletten weiteren Druckverlaufs auswirkt, siehe Abbildung 3.3 und 3.4c.
- **Warping** Beim Abkühlen des gedruckten Filaments kommt es zu thermischen Spannungen im Material. Dies ist vor allem direkt auf der Druckplatte problematisch. Durch eine zu schwache Haftung am Druckbett kann sich der Druck hierbei vom Druckbett lösen. Dies tritt vor allem an scharfen Ecken auf, da hier die thermischen Auswirkungen besonders stark sind. Dadurch hebt sich die Ecke an und biegt sich nach oben. Das Bauteil ist daher hinterher nicht mehr maßhaltig, siehe Abb. 3.4d.



(a) Extrusion Stop

(b) Layer Separation

(c) Layer Shift

(d) Warping

Abbildung 3.4.: Verschiedene Fehlerfälle visuell dargestellt. Quelle [112].



#### **Erkennung von Extrusion Stop**

Wenn die Extrusion gestoppt wird, entsteht eine Lücke zwischen dem Druckkopf und der obersten Schicht des gedruckten Objekts. Um zu bestimmen wo sich diese Schicht relativ zum Druckkopf befindet, reicht es aus den Abstand der obersten zusammenhängenden Pixelschicht vom oberen Rand des Bildes in der gewählten ROI zu bestimmen. Zu diesem Zweck wird das Bild Zeile für Zeile auf weiße Pixel überprüft. Wird ein weißes Pixel entdeckt, gilt die überprüfte Position als besetzt.

Außerdem wird gespeichert, ob sich der Suchalgorithmus gerade in einer Kante oder außerhalb befindet, was die erwartete Farbe des nächsten Pixels bestimmt - schwarz, wenn er sich außerhalb, weiß, wenn er sich innerhalb einer Kante befindet.

Da sich das Druckobjekt in der Mitte des Bildes befindet, beginnt die Suche außerhalb einer Kante. Sobald das erste belegte Pixel erkannt wird, wird seine Position gespeichert und der Status auf innerhalb einer Kante gesetzt. Wird dann ein unbesetzter Bildpunkt erkannt, bestimmt die gespeicherte Startposition die Länge der Kante.

Hat die Kante eine ausreichende Länge, wird eine Kante für diese Bildserie als erkannt gespeichert. Wird eine ausreichend lange Kante in mehreren aufeinanderfolgenden Zeilen des Bildes erkannt, wird die Nummer der aktuellen Zeile als obere Kante des Modells gespeichert.

Der Abstand wird von beiden Kameras auf Druckkopfebene gemessen und ein Durchschnittswert berechnet. Wenn dieser Durchschnittswert einen bestimmten Wert überschreitet, wird der Druckkopf als zu weit vom Druckobjekt entfernt erkannt und der Fehler wird gemeldet. Dies geschieht typischerweise, wenn entweder der Extruder verstopft ist oder kein Filament mehr nachgeliefert wird.

Die definierte Mindestlänge einer aufeinanderfolgenden Reihe von belegten Pixeln stellt sicher, dass keine im Binärbild verbleibenden Pixelfehler, sondern nur tatsächliche Kanten, erkannt werden. Die Notwendigkeit, tatsächliche Kanten in mehreren aufeinanderfolgenden Bildzeilen zu erkennen, stellt außerdem sicher, dass nicht eine Hintergrundkante, sondern das Objekt selbst erkannt wird, dessen Außenkanten aufgrund des geringeren Abstands des Objekts zur Kamera breiter erscheinen als die Kanten im Hintergrund des Bildes.

#### **Erkennung von Layer Separation**

Ähnlich wie bei der Erkennung eines Druckstopps geht es auch bei der Erkennung der Schichtentrennung um die Erkennung der oberen Objektkante. In diesem Fall ist es jedoch nicht nur notwendig, qualitativ zu erkennen, in welcher Bildreihe sich die oberste Schicht des Drucks befindet, sondern auch quantitativ zu bestimmen, wie gerade diese Schicht ist, da eine stark ungerade Schicht auf eine Trennung hinweist.

Um diese Quantifizierung vornehmen zu können, werden alle Bildspalten auf ihren höchsten belegten Punkt überprüft. Analog zu den vorherigen Fällen wird die Dicke der Objektkante verwendet, um sie von verbleibenden Pixelfehlern und Hintergrundkanten zu unterscheiden.

### 3. In-situ Fehlererkennung im Druckprozess

Um zu vermeiden, dass die Druckplatte oder fertige Teile des Drucks das Ergebnis verfälschen, wird der Suchbereich für die obersten belegten Punkte relativ zur erkannten Druckkopfebene begrenzt.

Nachdem alle obersten Punkte in einem Vektor gespeichert sind, wird die theoretische Position der horizontalen Oberkante durch Bildung eines arithmetischen Mittels bestimmt. Anschließend wird die Abweichung der tatsächlichen Oberkante von der theoretischen Oberkante ermittelt. Überschreitet diese in einer der beiden Kameraperspektiven auf Druckkopfebene einen Grenzwert, wird der Fehler Layer Separation als erkannt gemeldet.

#### **Erkennen von Layer Shifts**

Bei einem Layer Shift sind die rechte und linke Außenkante des von der Kamera erfassten Körpers in der gleichen Drucklage in die gleiche Richtung versetzt. Die Erkennung dieser Verschiebung in die gleiche Richtung stellt sicher, dass schmaler werdende Profile oder allmähliche Positionsänderungen der Außenkanten nicht unbeabsichtigt als Fehler gemeldet werden.

Um diese Parallelverschiebungen zu finden, müssen zunächst die rechte und linke Außenkante des Objekts erkannt und gespeichert werden. Dazu wird das Bild zeilenweise überprüft und die Position der äußersten belegten Pixel bestimmt. Um eine falsche Kantenlokalisierung zu vermeiden, wird durch die Überprüfung mehrerer aufeinander folgender Pixel sichergestellt, dass es sich um eine Kante und nicht um einen nicht erfolgreich gefilterten Pixelfehler handelt. Die Positionen der äußeren Kanten werden dann als Punkte in zwei separaten Vektoren gespeichert.

Anschließend werden die beiden Vektoren, in denen die Punkte der Außenkanten gespeichert sind, auf einen möglichen Layer Shift untersucht. Das heißt, es wird nach Abschnitten im Vektor gesucht, in denen plötzliche horizontale Veränderungen auftreten. Da eine Verschiebung nicht unbedingt zwischen zwei Zeilen eines Bildes auftreten muss, sondern sich aufgrund hoher Auflösungen auch über mehrere Zeilen erstrecken kann, werden Punkte mit einem Abstand von mehreren Pixeln miteinander verglichen, anstatt direkt aufeinander zu folgen. Die vertikale Nähe der verglichenen Einträge wird überprüft, um auszuschließen, dass es sich um einen kontinuierlichen und nicht um einen ruckartigen Übergang zwischen den Punkten handelt. Dies muss ausgeschlossen werden, da der Algorithmus zur Erkennung der Außenkanten, der die Vektoren erstellt, nicht unbedingt für jede Zeile des Bildes einen Wert speichert und somit Punkte mit einem großen vertikalen Abstand verglichen werden könnten. Wenn sichergestellt ist, dass die Punkte vertikal nahe beieinander liegen und eine mögliche Verschiebung nicht graduell sein kann, wird der horizontale Abstand zwischen den beiden Punkten ermittelt. Überschreitet dieser einen Grenzwert, wird der Vektoreintrag in einen zweiten Vektor geschrieben, der mögliche Positionen für Schichtverschiebungen speichert.

Die verschiedenen Kanten können jedoch eine unterschiedliche Anzahl von Kandidaten ergeben, da einseitige, sprunghafte Breitenänderungen auftreten können, beispielsweise in Form einer Treppe. Die Vektoren haben daher in der Regel eine unterschiedliche Länge und die Einträge an der gleichen Position beziehen sich auf unterschiedliche Zeilen des Bildes. Um

### 3. In-situ Fehlererkennung im Druckprozess

sicherzustellen, dass keine Sprünge auf unterschiedlichen Höhen als potenzielle Layer Shifts markiert werden, werden die Vektoren nach Paaren durchsucht. Dazu werden die Zeilenkoordinaten der potenziellen Punkte miteinander verglichen und ihre Positionen als Paare in unmittelbarer Nähe gespeichert.

Diese Paare, die durch ihre vertikale Nähe bestimmt werden, werden dann daraufhin überprüft, ob sie ein Layer Shift oder nur eine Veränderung der eigentlichen Bauteilgeometrie darstellen. Dazu werden die Beträge und Positionen der um die jeweiligen Punkte auftretenden Verschiebungen berechnet und verglichen. Sind diese gleich groß und zeigen in dieselbe Richtung, wird das Paar als tatsächlicher Layer Shift erkannt. Um zu verhindern, dass der Algorithmus bei jeder Prüfung wiederholt dieselbe Schichtverschiebung meldet, wird die Umgebung der potenziell neuen Verschiebung nach ihrer Erkennung gespeichert und mit der Umgebung der bereits gefundenen Verschiebungen verglichen. Wird die Umgebung der potenziell neuen Schicht nicht in den bereits gefundenen Umgebungen erkannt, wird angenommen, dass eine tatsächlich neue Schicht erkannt wurde. Dessen Umgebung wird in die bereits gefundenen Schichtumgebungen aufgenommen und der Fehler wird als neu gemeldet.

#### **Erkennung von Warping**

Warping wird das Abheben der unteren Schichten des gedruckten Objekts vom Druckbett genannt. Die Kameras auf der Druckplattenebene werden zur Erkennung des Fehlerfalls Warping eingesetzt. Die gewählte Position führt zu einer konstanten Position des Druckobjektes im Kamerabild, was den Vergleich mit einem gerenderten Soll-Bild des Objektes möglich macht. Tritt das Warping an den unteren Ecken des Objekts auf, sollten aufgrund der Kameraposition unbesetzte Pixel direkt auf und über der Druckplatte sichtbar sein.

Da jedoch im Allgemeinen nicht sicher ist, wie das Profil des gedruckten Objekts aussehen wird, wird das gerenderte Bild des gedruckten Modells verwendet, um einen Abgleich zwischen geplantem Modell und bisher gedruckten Modell zu erhalten. Zu diesem Zweck wird das gerenderte Bild analog zu den von Kameras aufgenommenen Bildern aufbereitet und beschnitten. Die ROI für den Warping-Fehlerfall enthält nur die untersten Zeilen des gedruckten Objekts, da Fehler und Abweichungen, die oberhalb dieser Zeilen auftreten, die Ergebnisse des Algorithmus verfälschen könnten. Die Bilder von Kamera und Rendering sollten daher übereinstimmen, sofern der Druck weit genug fortgeschritten ist, um die oberste Bildzeile zu belegen. Dies wird durch Zählen und Vergleichen der Anzahl der belegten Pixel der ersten Bildzeile von Render und Bild überprüft. Wenn die Anzahl der belegten Pixel ungefähr gleich ist, gilt der Druck als weit genug fortgeschritten.

Die Bilder werden dann voneinander subtrahiert. Das bedeutet, dass die Werte der einzelnen Pixel in beiden, idealerweise deckungsgleichen, Bildern voneinander subtrahiert werden und der Betrag der Differenz in ein neues Bild eingetragen wird. Pixel, die in den Bildern unterschiedliche Werte haben, erscheinen somit in dem durch die Subtraktion entstandenen Bild weiß. Durch Zählen dieser abweichenden Pixel kann die Kongruenz der Bilder unter Berücksichtigung eines gewissen Positionierungsfehlers bewertet und eine mögliche Verformung erkannt werden.

### 3.1.3. Ergebnis und Evaluation

Um beide horizontalen Richtungen für Layer Shift und Separation zu beobachten, müssen diese in einem Winkel von 90 Grad zueinander am Gehäuse des Druckers angebracht werden. Der größere horizontale Bildbereich der Intel RealSense R200-Kameras von 70 Grad im Vergleich zu 42,5 Grad der D435-Kameras und der geringe Platzbedarf der R200 machen sie für diesen Zweck besser geeignet, da die Position des Druckkopfs in einem anderen Feld erkannt werden kann. [113, 114].

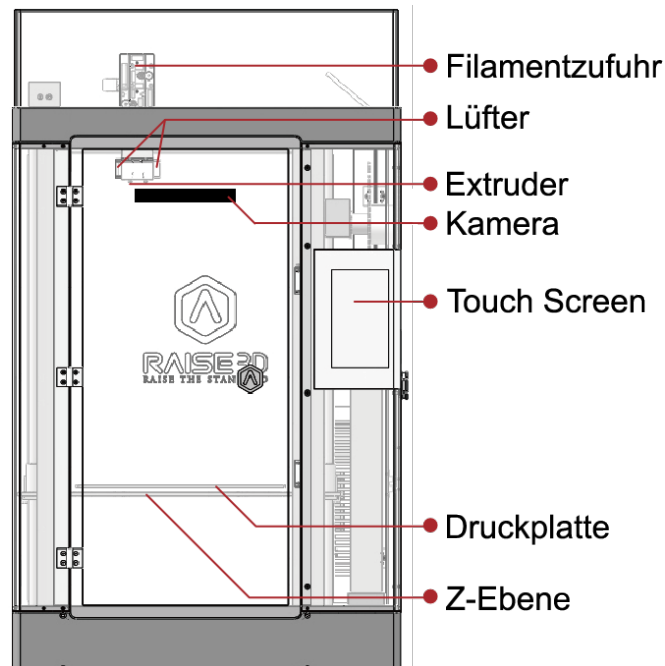


Abbildung 3.5.: Schematische Darstellung der Position der einzelnen Druckerkomponenten, sowie der Kamera auf Höhe des Extruders. Quelle basierend auf [115].

Der Testaufbau wird mit einem Raise3D ideaMaker N2 Plus durchgeführt, um die Allgemeingültigkeit des vorgestellten Ansatzes zu zeigen. Bei beiden Druckern handelt es sich um handelsübliche Geräte, die in der Industrie weit verbreitet sind. Obwohl beide Kameras (Intel RealSense D435 und R200) über Infrarot- und Tiefensensoren verfügen, werden in dem Ansatz nur die Farbbilder mit einem Seitenverhältnis von 16:9 und einer Auflösung von 1920x1080 Pixeln ausgewertet.

Für eine erfolgreiche Erkennung der verschiedenen Fehlerklassen ist die Position des Sensors wichtig. Um Lücken zwischen Extruder und Druckobjekt, Schichtverschiebungen und Extruderverstopfungen zu erkennen, werden Sensoren auf Extruderebene benötigt.

Der Raise3D-Drucker bewegt seinen Druckkopf in der X- und Y-Achse und das Bett mit seiner Z-Achse nach unten. Dies führt zu einem Aufbau, bei dem die Kamera auf einer festen Höhe montiert werden kann, da sich der Extruder nur horizontal bewegt.

### 3. In-situ Fehlererkennung im Druckprozess

Der Prusa-Drucker hingegen bewegt seinen Druckkopf sowohl in der X- als auch in der Z-Achse. So hebt sich der Druckkopf während des Drucks, während das Druckbett in der Y-Achse bewegt wird. Für die Prusa-Drucker wird ein Roboter eingesetzt, der die Position der Sensoren entsprechend der aktuellen Höhe der Extruder anpasst. Der Aufbau ist in Abbildung 3.6 dargestellt.

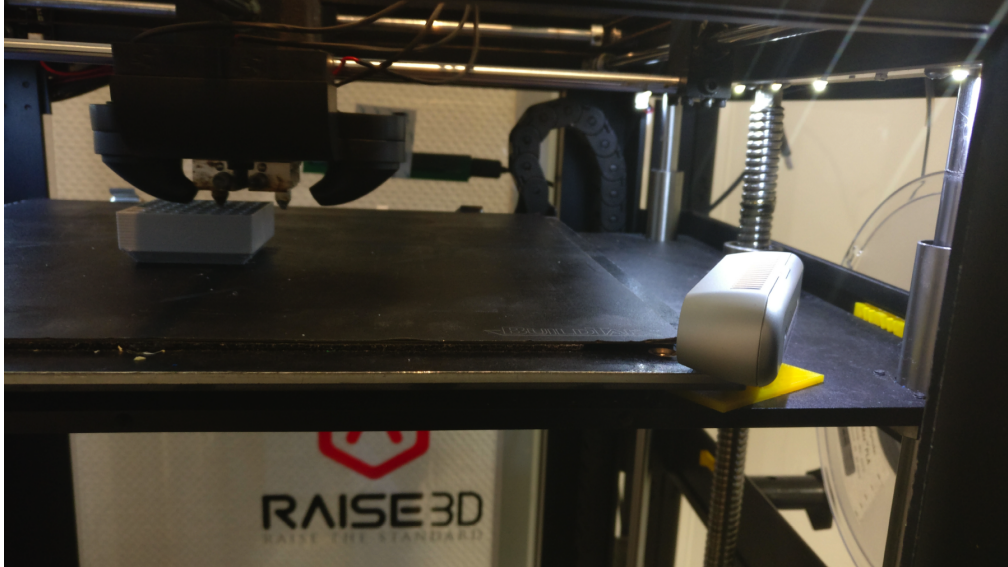


Abbildung 3.6.: Bild des Raise Druckers mit angebaute Kameras. Die Kameraposition ist diagonal zum Druckbett ausgerichtet und ermöglicht so eine Sicht sowohl auf die X- als auch auf die Y-Achsenbewegung des Druckers. Quelle [192].

Da sich der Druckkopf des ideaMaker-Druckers auch in der Y-Achse bewegt, ist die Erkennung des Extruders nicht immer perfekt. Um zu prüfen, ob der vorgestellte Algorithmus den Extruder an verschiedenen Positionen noch erkennen kann, wurde eine Auswertung der Erkennung und ihrer Position vorgenommen. Dazu wurde der Extruder in einem 10mm-Raster über den gesamten Arbeitsbereich in X- und Y-Achse bewegt und an jeder Position eine Erkennung durchgeführt. Die Ergebnisse sind in Abbildung 3.7 dargestellt. Da das Sichtfeld sowie der Fokuspunkt der Kamera begrenzt ist, stimmt der Erkennungsbereich nicht vollständig mit dem Druckbereich überein. Die Fehlererkennung deckt vor allem die Mitte des Druckbetts ab und vernachlässigt die Ränder, da standardmäßig das Bauteil immer mittig auf dem Druckbett vom Slicer positioniert wird. Selbst beim Druck eines großen Bauteils muss der Extruder den inneren Bereich des Druckbetts durchqueren und ist damit automatisch zu einem Großteil der Druckzeit immer im Erkennungsbereich.

Um die Leistung der beiden Algorithmen zur Erkennung von Layer Shift und Extrusion Stop zu bewerten, wurden insgesamt 1367 Messungen durchgeführt. 715 Messungen für Extrusion Stop und 652 Messungen für Layer Shifts.

Da Druckfehler schwer zu wiederholen sind, wurde der tatsächliche Fehler zum G-Code des gedruckten Objekts hinzugefügt. Für Layer Shift wurde ein verschobenes Modell gesliced

### 3. In-situ Fehlererkennung im Druckprozess



Abbildung 3.7.: Die Verteilung der Extrudererkennung in der XY-Ebene für die Druckplatte des ideaMaker-Druckers. Die Farben zeigen an, an welcher Stelle der Extruder zuverlässig erkannt werden konnte. In der Regel werden Objekte in der Mitte der Druckplatte gedruckt, so dass Druckvorgänge problemlos überwacht werden können. (Grün = zuverlässige Erkennung, Gelb = schwankende Erkennung, Rot = keine Erkennung). Quelle [192].

und für Extrusion Stop wurde der Extrusionsparameter im G-Code auf 0 gesetzt. Beide Fehler wurden bei etwa 39% des Druckvorgangs hinzugefügt.

Abbildung 3.8 zeigt den Druckfortschritt, zu welchem Zeitpunkt ein Fehler erkannt und der Druck abgebrochen wurde. Je früher ein Druck abgebrochen wird, desto mehr Zeit und Ressourcen werden gespart und der Druck kann wiederholt oder ein weiterer Druck gestartet werden. Da die Layer Shift Erkennung einige gedruckte Lagen nach dem Auftreten des Fehlers benötigt, benötigt die Erkennung mehr Zeit, in der das Objekt weiter gedruckt wird. Dies führt zu einem mittleren Abbruchfortschritt von 77 %. Der Nozzle Clogging Algorithmus basiert auf dem Abstand zwischen dem Extruder und dem Objekt. Sobald der Fehler auftritt, ist eine Erkennung möglich, sodass der mittlere Abbruchfortschritt bei 60 % liegt und somit früher erkannt wird als Schichtverschiebungen.

Beide Algorithmen erkennen echte negative Klassifizierungen, bis ein Fehler auftritt. Allerdings ist die Anzahl der falsch positiven Ergebnisse relativ hoch, da die Ansätze sehr empfindlich sind, siehe Abb. 3.10. Dies kann möglicherweise auf die unterschiedliche Beleuchtung der verschiedenen Drucker und die damit verbundene höhere Rate an Falscherkennungen zurückzuführen sein. Wenn ein falsch-positiver Fehler entdeckt wird, muss der Benutzer überprüfen, ob ein Fehler vorliegt und ob der Druck fortgesetzt werden kann. Im Gegensatz dazu ist die Anzahl der falsch-negativen Klassifizierungen gering. Da der Druck bei der nächsten Überprüfung fortgesetzt wird, kann der Fehler immer noch erkannt werden.

### 3. In-situ Fehlererkennung im Druckprozess

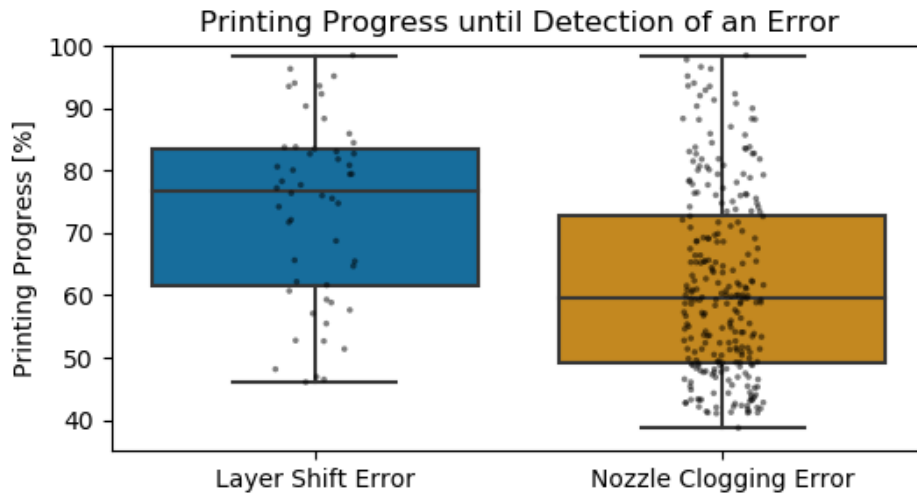


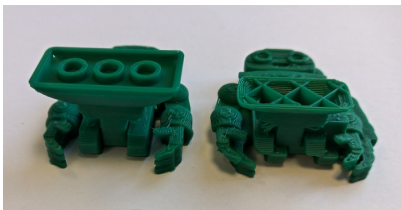
Abbildung 3.8.: Druckfortschritt in Prozent für die beide Fehler können während des Druckvorgangs erkannt werden. Bei der Schichtverschiebung dauert es im Durchschnitt länger, bis sie erkannt wird, da der Algorithmus mehrere verschobene Schichten benötigt, um einen Fehler zu erkennen. Es gibt keine Erkennung vor 40 %, da beide Fehler erst nach 39 % des Druckvorgangs erkannt wurden. Quelle [192].



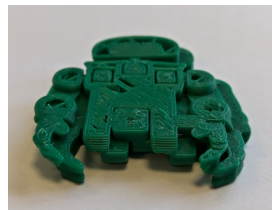
- (a) Ausmaß der unterschiedlichen Klassifizierungen während des Druckprozesses. Sobald ein echt positiver Fehler erkannt wird, wird der Druckvorgang abgebrochen. Insgesamt wurden 652 Tests durchgeführt.
- (b) Übersicht über die Fehlerklassifizierung beim Testen von sechs Druckern nach dem Zufallsprinzip auf Extruderverstopfung. Sobald der Algorithmus den Fehler als echt positiv einstuft, wurde der Druckvorgang abgebrochen. Insgesamt wurden 715 Tests durchgeführt.

Abbildung 3.9.: Klassifizierungsergebnisse während des Druckprozesses für die Fehlerfälle Layer Shift und Nozzle Clogging / Extruder verstopft. Quelle [192].

### 3. In-situ Fehlererkennung im Druckprozess



(a) Auf der linken Seite wurde ein Objekt wie vorgesehen gedruckt. Auf der rechten Seite ist ein Layer Shift aufgetreten und das Objekt wurde fortgesetzt, bis der Druckvorgang gestoppt wurde.



(b) Ein Layer Shift wurde kurz nach dem Auftreten gestoppt.



(c) Während des Drucks wurde der Extruder verstopft und die Extrusion stoppte. Der Druckkopf führt ohne Fehlererkennung die Druckbewegungen weiter fort ohne das Filament extrudiert wird.

Abbildung 3.10.: Verschiedene Aufnahmen von gedruckten Objekten mit und ohne aufgetretenem Fehler.



### 3. In-situ Fehlererkennung im Druckprozess

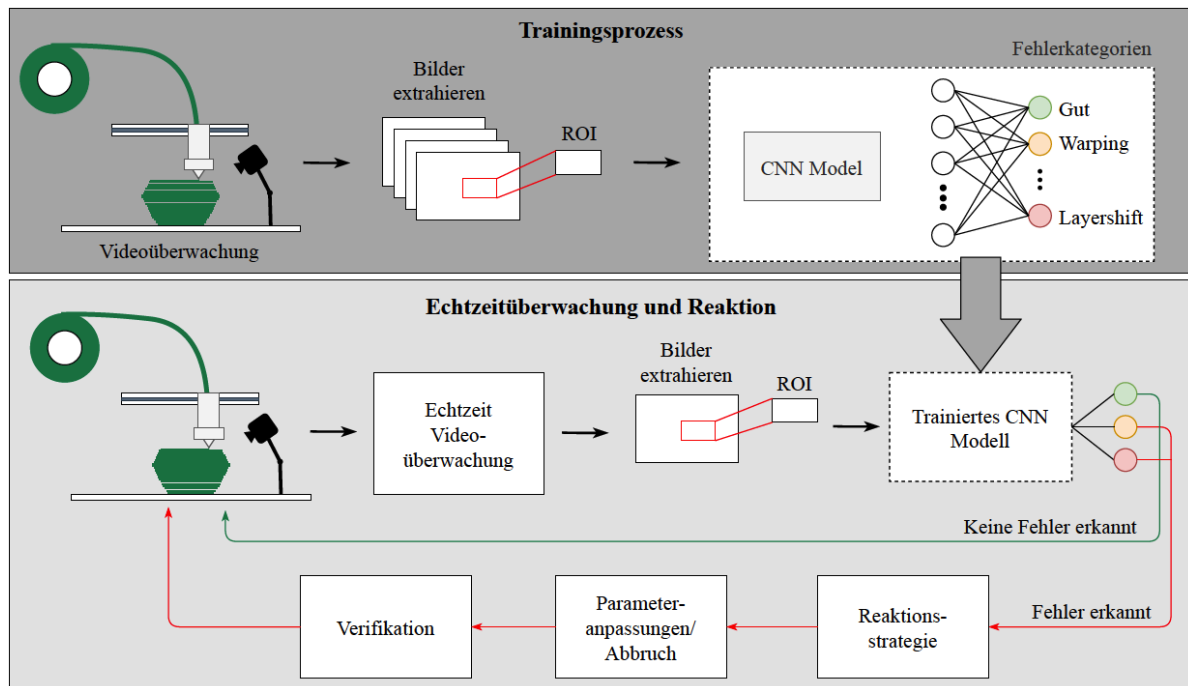


Abbildung 3.11.: Das Konzept der in-situ Prozessüberwachung bei einem Fused Filament Fabrication (FFF) Drucker. Hierbei wird ein Convolutional Neural Network (CNN) Modell verwendet, um Fehler im Druckprozess zu klassifizieren. Quelle [186].

## 3.2. Optische Fehlererkennung und Klassifikation mit Hilfe von Deep Learning

Um die hohen Material- und Zeitverluste zu minimieren, soll ein bildverarbeitendes Machine Learning-System eingesetzt werden, um den additiven Fertigungsprozess zu überwachen. Dabei müssen nicht nur fehlerhafte Drucke erkannt werden, sondern auch der jeweiligen Fehlerkategorie zugeordnet werden. Wenn das System als geschlossener Regelkreis konzipiert ist, kann eine Reaktionsstrategie automatisch ausgelöst werden, sobald ein Fehler erkannt und klassifiziert wurde. Abhängig von der Fehlerklasse kann dies z.B. ein Abbruch oder eine Anpassung der Druckparameter bedeuten. Diese Reaktionsstrategie kann entweder automatisch ausgeführt oder von einem Anwender verifiziert werden. Der Klassifikator der Fehlerklasse stellt die Hauptkomponente des Konzepts einer teilautonomen in-situ Prozessüberwachung dar.

Bevor ein Regelkreis entwickelt werden kann, müssen zuerst Trainingsdaten gesammelt und ein geeigneter Klassifikator trainiert werden. Da bisher kein Trainingsdatensatz öffentlich verfügbar ist, der für diesen Anwendungsfall geeignet ist, musste hierfür ein eigener Datensatz erstellt werden. Im Folgenden wird ein Konzept für die Entwicklung dieses Klassifikators vorgestellt. In der Konzeptionsphase wurden verschiedene Alternativen für die Erkennung und Klassifikati-

### 3. In-situ Fehlererkennung im Druckprozess

on von Fehlern untersucht und deren Leistungsfähigkeit in mehreren Tests bewertet. Zunächst wurden sogenannte 3D-Reconstruction-Module untersucht, die es ermöglichen, auf Basis von einer oder mehreren Kameraperspektiven aus zweidimensionalen Bildern dreidimensionale Voxeldateien zu erzeugen, die mit den CAD-Daten des herzustellenden Objekts verglichen werden können. Bei Bildern der gedruckten Objekte hat dieser Ansatz jedoch Schwierigkeiten, vermutlich da es Detailaufnahmen ohne viele unterschiedliche Tiefeebenen sind. Die rekonstruierten 3D-Modelle von fehlerfreien Drucken weichen bereits in großem Maße von den CAD-Daten ab und die Rekonstruktion erfordert einen hohen Entwicklungsaufwand, was die Technologie für diesen Anwendungsfall unattraktiv macht. Als Nächstes wurden sogenannte Re-Identification Netzwerke untersucht, die häufig in der Wiedererkennung von Autos oder Gesichtern eingesetzt werden. Diese Methode sucht über die Abstände der optischen Merkmale die ähnlichsten Bilder zueinander. Diese Methode war nicht geeignet, da sie nicht in der Lage ist, Fehlerklassen wiederzuerkennen, da die Fehler mit unterschiedlichen optischen Merkmalen auftreten und somit die Methode nicht in der Lage ist, zwischen Fehlerklassen zu unterscheiden.

Im letzten Test wurde der Ansatz des Ensemble Lernens bei tiefen Neuronalen Netzen (NN) anhand eines Beispieldatensatzes von drei zufällig verteilten Klassen untersucht. Der modulare Ansatz erzielte eine Accuracy von mehr als 5% höher gegenüber dem Einzelmodell und eignet sich somit für die Unterscheidung von Fehlerklassen. In Verbindung mit den für die Bildverarbeitung geeigneten Convolutional Neural Networks (CNNs) ist eine höhere Vorhersagegenauigkeit, strukturelle Vorteile und ein geringerer Trainingsaufwand zu erwarten. Bisher ist keine Studie bekannt, in der eine solche Herangehensweise für die Überwachung des additiven Fertigungsprozesses verwendet wurde. Daher soll das Modell, das die Zuordnung der Fehlerkategorien übernimmt, aus mehreren einzeln trainierten Modulen bestehen. Jedes Modul besteht aus einem möglichst schlanken und flachen CNN, das eine binäre Klassifikationsaufgabe löst. Dazu wird die Mehrklassen-Klassifikation der möglichen Fehlerkategorien in mehrere binäre Klassifikationen aufgeteilt. Es gibt verschiedene Möglichkeiten für diese Umwandlung, wie One-vs-One (OVO) und One-vs-All (OVA), die gegeneinander verglichen werden, siehe Abbildung 3.12.

Ein Konzept wird entwickelt, um geeignete Trainingsdaten für das Gesamtmodell zu erzeugen. Obwohl die Verwendung von einer Kombination aus CAD- und Kamerabildern für die Fehlererkennung erfolgreich war, werden lediglich Kamerabilder für die Klassifikation verwendet. Gerenderte CAD-Bilder tragen laut Ergebnissen von Keller [188] nicht zur Unterscheidung von verschiedenen Fehlerklassen bei.

Ein Versuchsaufbau wird benötigt, um die Trainingsdaten zu generieren. Mit einer geeigneten Anordnung der Komponenten müssen alle gedruckten Objekte auf den Bildern deutlich sichtbar sein. Die Bilder sollen aus einer festen Perspektive mit einer einzigen Kamera aufgenommen werden, die den gesamten Bereich des zu überwachenden Objekts erfasst. Um die Qualität der gedruckten Objekte nicht zu beeinträchtigen, müssen die Bilder ohne Eingriff in den Druckprozess aufgenommen werden. Nach der Identifizierung der relevanten Fehlerklassen werden verschiedene Objekte ausgewählt, um Trainingsdaten zu generieren. Diese Objekte werden in zwei Kategorien eingeteilt, um die Schwierigkeitsgrade und Varianzen zu erhöhen.

### 3. In-situ Fehlererkennung im Druckprozess

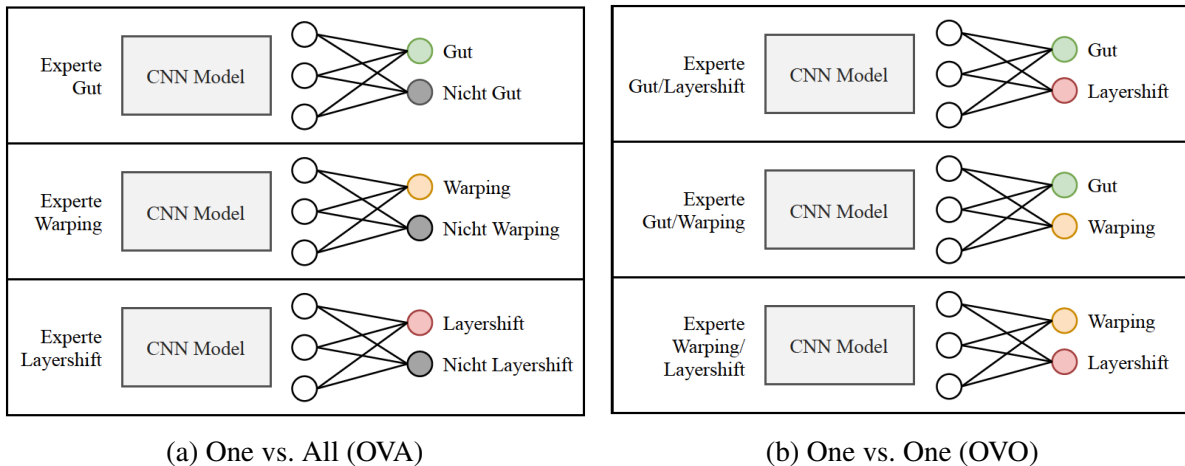


Abbildung 3.12.: Möglichkeiten der binären Klassifikation der Klasse Gut sowie den Fehlerklassen Warping und Layer Shift. Quelle [186].

Da Fehler im Druckprozess unregelmäßig und nicht vorhersehbar auftreten, werden für den Datensatz speziell präparierte Objekte gedruckt, die einen Fehler enthalten. Hierfür werden die Objekte beziehungsweise die Druckparameter angepasst, damit die entsprechend gewünschte Fehlerklasse gezielt im Druckprozess auftritt.

Ein Script wird verwendet, um regelmäßig Bilder während des Druckprozesses mit einer Kamera aufzunehmen. Diese Bilder werden vorverarbeitet, um sie für die Verarbeitung durch die Netze möglichst einfach zu machen. Die Bilder des Drucks werden den jeweiligen Klassen zugeordnet und entsprechend gelabelt. Die Trainingsdaten, die mit den einfachen Objekten erstellt wurden, werden verwendet, um die Systemtopologie, den Systemaufbau und verschiedene Grundstrukturen zu testen. Das Modell mit der höchsten Genauigkeit und dem geringsten Trainingsaufwand wird ausgewählt. Anschließend werden die Leistungsfähigkeit der Module durch Erweiterungen, Regularisierungen und Anpassung der Hyperparameter verbessert. Das entwickelte Gesamtmodell wird abschließend hinsichtlich seiner Vorhersagegenauigkeit, Generalisierungsfähigkeit und Robustheit evaluiert. Der Prozess des vorgestellten Entwicklungskonzepts ist in Abbildung 3.13 dargestellt.

Es wird untersucht, ob ein modulares Modell geeignet ist, um die anstehende Aufgabenstellung zu lösen. Um die tatsächliche Leistungsfähigkeit zu bewerten, wird das modulare Modell mit einem monolithischen Modell verglichen. Damit die industrielle Umsetzbarkeit beurteilt werden kann, werden auch der Trainings- und Rechenaufwand bewertet. Das vorgestellte Trainingskonzept des modularen Modells wird auch für die Entwicklung des monolithischen Modells verwendet, jedoch ohne die Schritte, die mit dem modularen Ansatz einhergehen und die Erhöhung des Schwierigkeitsgrades.

### 3. In-situ Fehlererkennung im Druckprozess

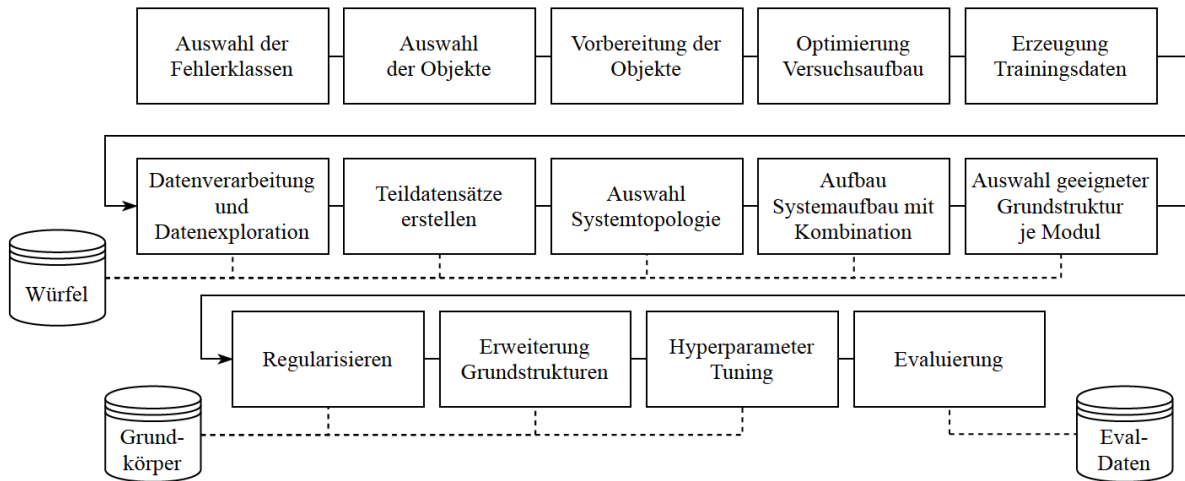


Abbildung 3.13.: Ablauf des vorgestellten Entwicklungskonzepts für das modulare System angefangen bei der Auswahl der passenden Fehlerklassen und möglichen Testobjekten über die Erzeugung von Trainingsdaten bis hin zur Implementierung der einzelnen Module sowie anschließenden Evaluierung. Quelle [186].

#### 3.2.1. Fehlerkategorien

Folgende vier Fehlerklassen wurden ausgewählt, um damit die einzelnen Netze zu trainieren: *Warping*, *Layer Shift*, *Spaghetti*, *Clogged Nozzle*. In Abbildung 3.14 sind die vier Fehler beim Druck einer Pyramide abgebildet. *Clogged Nozzle* steht hierbei stellvertretend für den Stop der Extrusion z.B. durch Verstopfung des Extruders, fehlendes Filament oder falsche Temperatureinstellungen oder Filamentwahl.

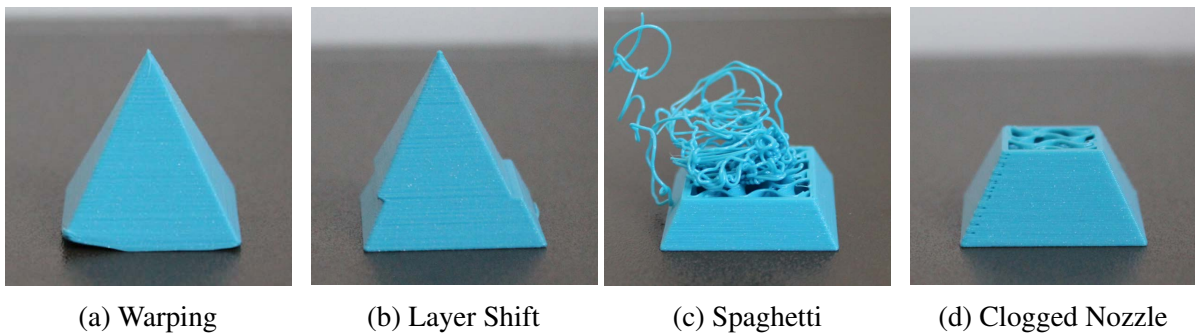


Abbildung 3.14.: Auswahl der relevanten Fehlerklassen am Beispiel einer Pyramide. Quelle [186].

### 3. In-situ Fehlererkennung im Druckprozess

Im Folgenden wird erläutert, wie die Fehlerklassen „simuliert“ werden, indem bestimmte Bedingungen geschaffen werden, die das Auftreten dieser Fehlerklassen fördern.

**Warping** Für die Fehlerkategorie Warping wird zum Beispiel das Material ABS verwendet, das einen höheren Schrumpfkoeffizient hat als PLA und daher eher zu Warping neigt. Die Temperatur der Bauplattform wird ab der zweiten Schicht reduziert und die Lüftergeschwindigkeit des Druckkopfes erhöht, um das Warping deutlicher zu machen.

**Layer Shift** Für die Fehlerklasse Layer Shift wird das CAD-Objekt ab einer bestimmten Höhe geschnitten und in der X- und Y-Ebene verschoben, um einen Versatz zu simulieren.

**Spaghetti** Für die Fehlerklasse Spaghetti wird die Bauplattform nach einer bestimmten Anzahl an Schichten abgekühlt, damit der Druck sich von der Bauplattform löst und der Abstand des Extruders von der Bauplattform erhöht, damit der Filamentstrang die Haftung auf dem Druckobjekt verliert.

**Clogged Nozzle** Für die Fehlerklasse Clogged Nozzle wird der Vorschub des Filaments im Extruder ab einer bestimmten Schicht gestoppt.

#### 3.2.2. Erstellung, Klassifikation und Exploration des Datensatzes

Die Grundlage für das überwachte Lernverfahren ist ein gelabelter Datensatz. Hierbei wird pro Bild eine oder mehrere Klassen identifiziert, mit denen das neuronale Netz das Bild klassifizieren soll. Obwohl das Labeln manuell durchgeführt werden muss und einen hohen Zeitaufwand erfordert, gibt es verschiedene Möglichkeiten, ein Bild zu kennzeichnen. Einige Beispiele sind das Zuweisen von einem oder mehreren Labels zu einem Bild oder das Markieren von Bereichen innerhalb eines Bildes, in denen der Fehler auftritt. Letzteres Vorgehen der Labelgenerierung ist jedoch sehr aufwendig und erfordert oft zusätzliche Software. Da bereits vorliegende Daten mit einem Label pro Bild gekennzeichnet sind und aufgrund des deutlich geringeren Aufwands, werden die Bilder in dieser Arbeit immer als Ganzes gekennzeichnet. Die Bilddaten eines Druckprozesses werden jeweils mit der Kennzeichnung *gut* versehen, bis eine der Fehlerklassen auftritt. Nachdem die Fehlerklasse erkannt wurde, entsteht ein Übergangsbereich, in dem die Unterscheidung zwischen den Klassen nicht klar ist. Da sich die optischen Merkmale der Fehlerklassen überschneiden, wird dieser Übergangsbereich nicht für das Training berücksichtigt.

Um einen aussagekräftigen Datensatz zu bekommen, wurden verschiedene Fehlerfälle inkludiert. Es wurde nicht nur der Fehlerfall selbst variiert, sondern auch verschiedene Filamentfarben und verschiedene Beleuchtungsszenarien umgesetzt.

#### Exploration der Bilddaten

Die Trennbarkeit der einzelnen Fehlerkategorien beeinflusst die Leistungsfähigkeit des Klassifikators. Während bei textuellen und strukturellen Daten die Unterscheidung der Klassen einfach durch den Vergleich von zwei Features erfolgen kann, ist es bei Bilddaten schwieriger. Um die Informationen und Merkmale von Bilddaten auf zwei Dimensionen zu reduzieren, können Methoden wie PCA [116] und t-Distributed Stochastic Neighbor Embedding (t-SNE) [117] verwendet werden.

Die PCA berechnet eine Linearkombination, um Eingangsvektoren in einen kleineren Unterraum zu reduzieren, indem korrelierende Variablen in eine kleinere Anzahl von unkorrelierten Variablen transformiert werden, die als Hauptkomponenten bezeichnet werden. Dabei versucht jede Hauptkomponente so viel unerklärte Varianz der höherdimensionalen Darstellung wie möglich zu erfassen, um den Informationsverlust während der Transformation zu minimieren [116]. t-SNE ist ein nicht lineares Verfahren zur Dimensionalitätsreduktion, das hochdimensionalen euklidischen Distanzen zwischen Datenpunkten in bedingte Wahrscheinlichkeiten umwandelt. Diese Wahrscheinlichkeiten repräsentieren Ähnlichkeiten zwischen den Dateninstanzen und steigen für nahe gelegene Dateninstanzen an [117].

Sowohl PCA, als auch t-SNE, kommen beide häufig in der Vorverarbeitung zu ML-Modellen zum Einsatz, wenn die Anzahl der Datensätze nicht mit dem Feature-Raum übereinstimmt. Die Anforderungen an die Menge der Datensätze für maschinelles Lernen steigen stark an, je mehr Merkmale pro Datensatz vorhanden sind [118]. Hier werden beide Verfahren vor allem zum Visualisieren und Explorieren der Daten eingesetzt.

In der Anfangsphase wurde eine PCA auf einer begrenzten Anzahl von 5.000 Bilddateninstanzen auf zwei Hauptkomponenten durchgeführt. Die Ergebnisse werden in Abbildung 3.15 (a) dargestellt, wobei die Fehlerklassen farblich gekennzeichnet sind. Im nächsten Schritt wurde der t-SNE-Algorithmus auf den gleichen Teildatensatz angewendet.

Es wurde festgestellt, dass die Instanzen bei Verwendung des t-SNE-Algorithmus klarer getrennt werden konnten und sich eindeutiger Cluster bildeten, siehe Abbildung 3.15 (b). Schließlich wurden beide Verfahren kombiniert, indem die PCA mit 50 Hauptkomponenten als Eingabe für den t-SNE-Algorithmus verwendet wurde, was das Ergebnis noch weiter verbesserte, siehe Abbildung 3.15 (c). Es wurde beobachtet, dass einige Fehlerklassen schwerer voneinander zu unterscheiden waren als andere, was mögliche Schwierigkeiten bei der Entwicklung des Systems anzeigen könnte. Zudem wird deutlich, dass sich die verschiedenen Fehlerklassen untereinander teilweise nur schwer unterscheiden lassen. Die Anwendung des Algorithmus hat gezeigt, dass bestimmte Fehlerklassen leichter von den anderen zu trennen sind. Insbesondere die Fehlerklassen Clogged Nozzle und Spaghetti konnten klar von den restlichen Dateninstanzen unterschieden werden. Im Gegensatz dazu hatte der Algorithmus Schwierigkeiten, die Fehlerklasse Warping von der Klasse Gut zu trennen, was auf Probleme des zu entwickelnden Systems hinweist, wenn es um diese Klassen geht. Es ist zu beachten, dass die Analyse nur in zwei Dimensionen durchgeführt wurde und eine dreidimensionale Darstellung mit einer weiteren Hauptkomponente möglicherweise andere Ergebnisse liefern würde.

### 3. In-situ Fehlererkennung im Druckprozess

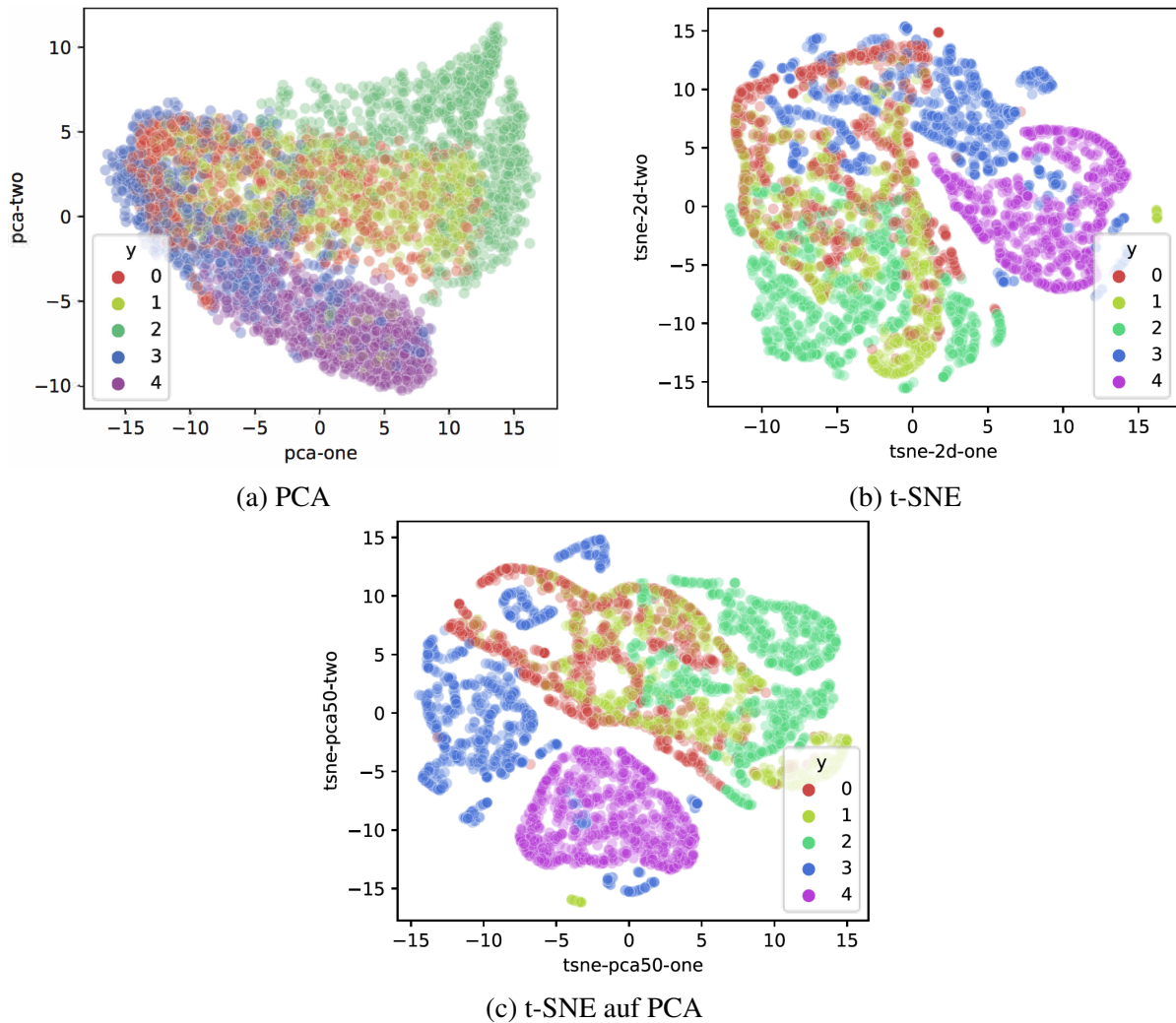


Abbildung 3.15.: Reduktion der Dimensionalität der Dateninstanzen anhand 5.000 Bildinstanzen für die fünf Fehlerklassen: 0 = Gut, 1 = Warping, 2 = Layer Shift, 3 = Spaghetti und 4 = Clogged Nozzle. Quelle [186].

#### 3.2.3. Umsetzung des Konzepts

Im folgenden Abschnitt wird beschrieben, wie das bisher beschriebene Konzept umgesetzt wurde. Zur Entwicklung und Optimierung des Systems wurden verschiedene Tests und Untersuchungen durchgeführt, die nachfolgend beschrieben werden.

#### Versuchsaufbau

Ein umfangreicher Datensatz ist notwendig, um das Machine Learning (ML)-Verfahren zuverlässig zu trainieren und zu evaluieren. Mit Hilfe eines Versuchsaufbaus wird es möglich die Daten kontrolliert aufzunehmen.

### 3. In-situ Fehlererkennung im Druckprozess

**Prusa MK3S** Der Prusa I3 MK3S ist ein flexibles FFF-Druckersystem, da viele seiner Bestandteile und Parameter als Open-Source Bauteil und Software verfügbar sind. Dies ermöglicht Modifikationen von Firmware, Ersatzteilen und Slicern. Es verwendet Standardformate wie G-Code und STL und ermöglicht daher einfache Manipulationen im Fertigungsprozess. Durch das offene Design des Druckers und das Fehlen eines umschließenden Gehäuses kann die Szenerie einfacher optimiert werden und ist für den Roboter einfacher zugänglich. Daher wird in der folgenden Untersuchung mit diesem 3D-Drucker gearbeitet.

**Intel RealSense D435** Die Wahl der Kameraposition ist von großer Bedeutung, um möglichst viel Oberfläche des herzustellenden Objekts zu erfassen und die Fehler zu erkennen. Um störende Inhalte im Hintergrund zu vermeiden und den Lernerfolg zu verbessern, wurde eine diagonale Position gewählt, die auf einer Ebene mit dem Druckbett liegt. Dies ermöglicht es, den kritischen Bereich nahe der Bauplattform ohne Störungen zu erfassen und alle vier untersuchten Fehlerklassen besser zu erkennen. Um den Abstand der Kamera zum Druckobjekt zu minimieren, wurde die Kamera am Roboterarm um 180 Grad gedreht montiert. Die finale Kameraposition ist in Abbildung 3.16 dargestellt.

**Beleuchtung** Zur gleichmäßigen Ausleuchtung der Kamerabilder wurde ein LED-Panel so angebracht, das der aufzunehmende Bereich gut ausgeleuchtet ist. So ließen sich Probleme mit schwankender Beleuchtung, rauschenden Bildern und verschwommenen Details verhindern.

#### **Datenerzeugung und -vorverarbeitung**

Die Bilder werden in einer festgelegten Frequenz aufgenommen, um Daten für den Druckprozess zu erzeugen. Hohe Frequenzen ermöglichen es, mehr Bilder pro Druckprozess aufzunehmen, jedoch kann es auch zu redundanter Information führen, die für das Training von CNNs nicht von Nutzen ist. In vergleichbaren Arbeiten werden die Bilder erst nach Beendigung einer Druckschicht aufgenommen, anstatt während des Druckprozesses [119, 120]. Um die Anzahl der benötigten Drucke für 15.000 Dateninstanzen bei einer Bauhöhe des Objekts von 2 cm zu minimieren, wird die Aufnahmefrequenz der Kamera auf ein Bild pro Sekunde gesetzt. Dadurch werden bei einer durchschnittlichen Druckgeschwindigkeit von 62,5 mm/s lediglich acht Drucke des gleichen Objekts benötigt. Im Vergleich dazu werden bei einer zeitlichen Synchronisation der Bildaufnahme mit dem Druckprozess und der Aufnahme von Bildern nach Beendigung einer Druckschicht etwa 112 Drucke benötigt. Obwohl durch die höhere Aufnahmefrequenz teilweise redundante Daten entstehen, ist der Aufwand für die Aufnahme der Bilder geringer als bei der zeitlichen Synchronisation. Der Prozess der Erzeugung der Bilddaten wird mithilfe eines automatischen Python-Scripts durchgeführt. Das Script legt die Bilder in nummerierten Ordnern ab und sorgt dafür, dass sie leicht identifizierbar sind. Vor der Weiterverarbeitung der Rohdaten, müssen diese jedoch vorverarbeitet werden, um die Merkmalsfindung der CNNs zu vereinfachen und effizienter zu gestalten. Dazu werden die Bilder um 180 Grad gedreht. Dies hat keine Auswirkungen auf den Informationsgehalt, vereinfacht



### 3. In-situ Fehlererkennung im Druckprozess

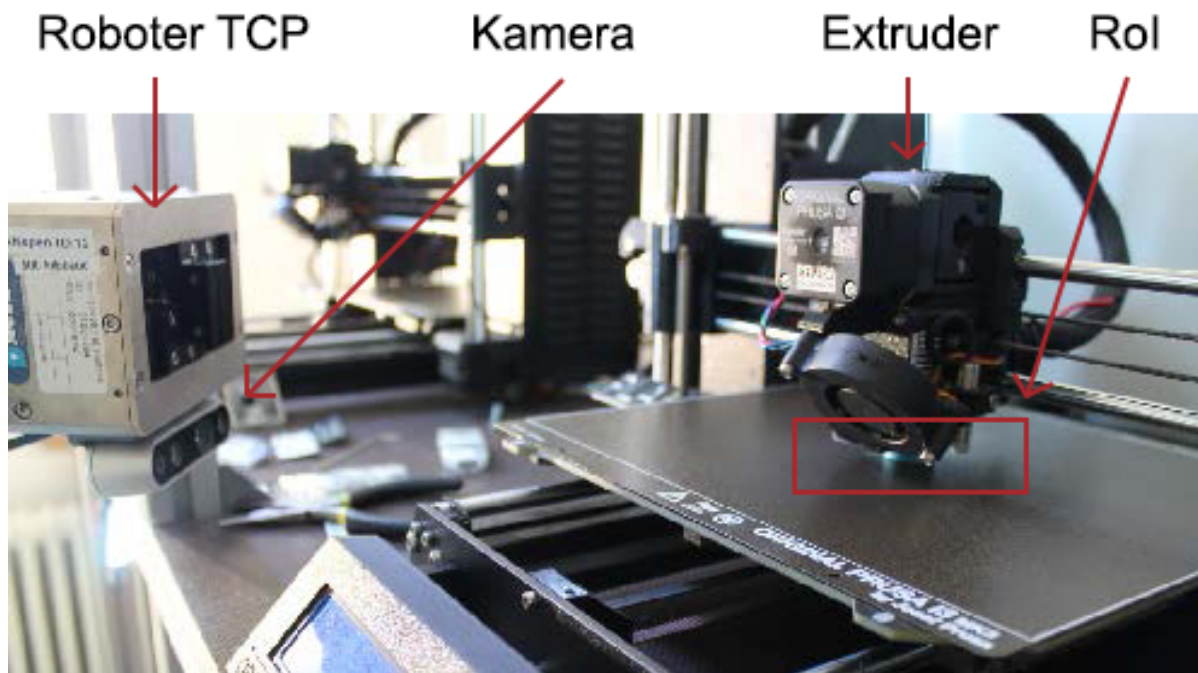


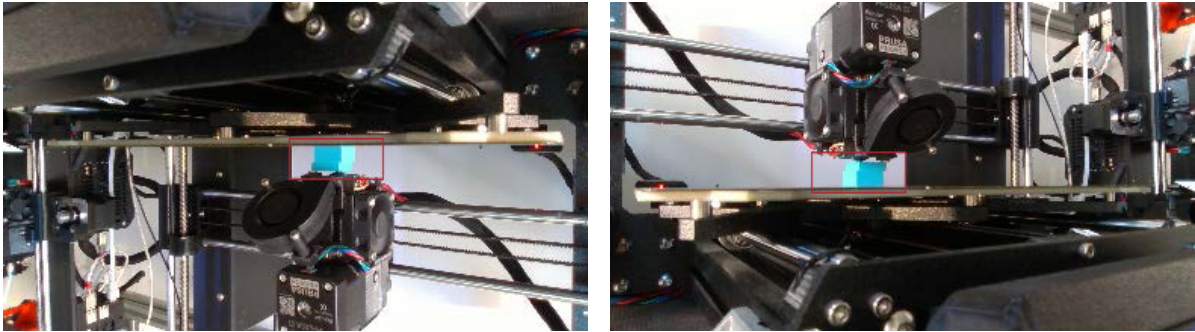
Abbildung 3.16.: Bild des Testaufbaus mit Roboterarm links im Bild und 3D-Drucker rechts. Die Kameraposition ist diagonal zum Druckbett ausgerichtet und ermöglicht so eine Sicht sowohl auf die X- als auch auf die Y-Achsenbewegung des Druckers. Sie ist um 180 Grad gedreht montiert. Zum einen, damit sie eine optimale Perspektive auf Höhe des Druckbetts einnehmen kann, zum anderen, um bei allen weiteren Prozessschritten die Bewegungen des Roboters nicht zu beeinträchtigen. Quelle [186].

jedoch das manuelle Labeln und Zuordnen der Daten. Ein Beispiel für ein Originalbild und ein gedrehtes Bild ist in Abbildung 3.17 zu sehen.

Um den Informationsgehalt zu minimieren und den Fokus auf relevante Informationen zu legen, werden die Bilder im nächsten Schritt beschnitten. Um die Datenmenge für die Netze zu minimieren und die Laufzeit zu optimieren, wird eine Region of Interest (ROI) gewählt. Dieser Bereich soll groß genug sein, um die sich bewegenden Objekte während des Druckprozesses abzubilden, gleichzeitig aber so wenig störende Elemente im Hintergrund enthalten wie möglich. Mit einer festen Größe von 310 x 140 Pixeln und einer Ausrichtung auf das Zentrum der Bauplatzform ermöglicht die ROI ein maximales Bauvolumen von 4 x 2,5 x 4,5 cm.

Zur weiteren Reduktion der Rechenlast wird aus den Farbbildern ein Graubild berechnet. Zum einen wird dadurch ein Bias auf eine bestimmte Farbe unterbunden, zum anderen ist so nur ein Drittel des Rechenaufwands notwendig. Die Abbildung 3.18 zeigt den ROI vor und nach einer Grauwertumwandlung. Durch die Umwandlung von Farbbildern in Grauwertbilder werden helle und farbige Filamente deutlich kontrastreicher dargestellt, da sich diese Farben stärker von dem Hintergrund abheben als Grautöne. Aus diesem Grund werden hauptsächlich

### 3. In-situ Fehlererkennung im Druckprozess

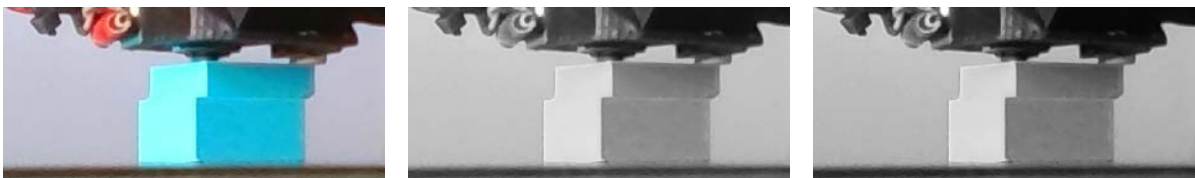


(a) Kamerabild. Um 180 Grad gedreht, da Kamera am Roboterarm kopfüber montiert ist, um Bewegungen des Roboterarms nicht zu beeinträchtigen.

(b) Durch Software gedrehtes Bild.

Abbildung 3.17.: Originalbild und gedrehtes Bild, welches später als Grundlage für die weiteren Vorverarbeitungsschritte dient. Die Region of Interest ist als roter Kasten eingezeichnet. Quelle [186].

rote und hellblaue Filamente für die Drucke verwendet. Im letzten Schritt der Verarbeitung werden die Grauwertbilder optimiert, um zum Beispiel den Grauwertbereich zu vergrößern oder den Kontrast zu erhöhen. Dies führt dazu, dass wichtige Merkmale wie Kanten auf den Bildern deutlicher hervorgehoben werden. Da die Aufnahmebedingungen, die verwendeten Filamentfarben und die Schattierungen variieren, sind die Datensätze qualitativ unterschiedlich. Dies erschwert die Suche nach einer optimalen Manipulationsstrategie. Um die beste Strategie zu finden, wurden verschiedene Verfahren wie die Lineare Skalierung, Clipping und die Gamma-Transformation mit unterschiedlichen Parametern an einer zufällig ausgewählten Stichprobe getestet. Die Gamma-Transformation mit einem Gamma-Wert von 0,8 hat die besten Ergebnisse geliefert, da sie das gesamte Bild dunkler macht und gleichzeitig den Kontrast über alle Grauwertbilder mit einem sich aus dem Bild ergebenden Faktor verstärkt.



(a) ROI in Farbe

(b) Grauwertumwandlung

(c) Gamma-Transformation

Abbildung 3.18.: Ausgewählte ROI nach den verschiedenen Vorverarbeitungsschritten. Quelle [186].

### 3. In-situ Fehlererkennung im Druckprozess

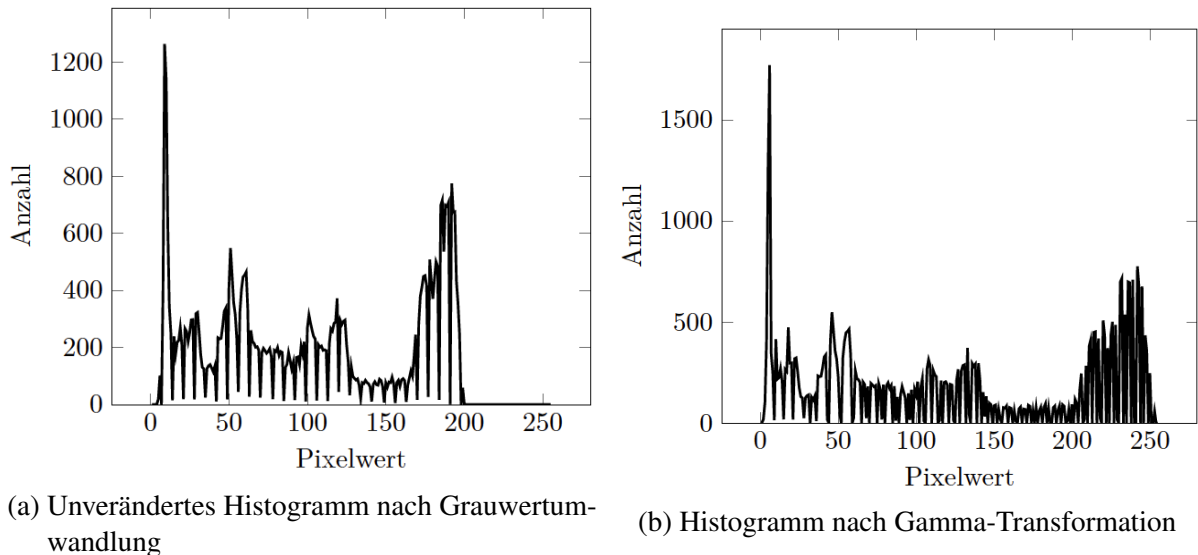


Abbildung 3.19.: Histogramm eines Grauwertbildes einmal ohne und einmal mit angewandter Gamma-Transformation. Quelle [186].

Abbildung 3.19 illustriert die Wirkung der Gamma-Transformation anhand eines Beispiel-Histogramms eines Grauwertbildes. Der Anteil an dunkleren Pixeln nimmt zu und gleichzeitig wird der maximale Grauwertebereich des Bildes optimal genutzt. Durch die dadurch erhöhte Kontrastwirkung lassen sich Kanten in den Bildern besser erkennen. Abschließend werden die vorverarbeiteten Bilddaten entsprechend den optischen Fehlermerkmalen gelabelt, siehe Abschnitt 3.2.2.

#### 3.2.4. Topologie des modularen neuronalen Netzes

Bis zur fertigen Topologie des modularen neuronalen Netzes wurden verschiedene Tests durchgeführt, um einen geeigneten Aufbau zu erarbeiten. Zu Beginn wurde ein Datensatz ausgewählt. In diesem befinden sich ausschließlich Aufnahmen von einem Würfel mit einer Kantenlänge von 20 mm. Hiermit wurde überprüft, ob CNNs grundsätzlich in der Lage sind, die gegebene Aufgabenstellung zu lösen. Die Geometrie des Würfels stellt die leichteste Aufgabe mit der geringsten Varianz dar. Nachdem ein grundlegender Aufbau des Netzes gefunden wurde, wird der Datensatz für die Regularisierung und das Hyperparameter-Tuning um weitere Objekte (Pyramiden, Halbkugel, Kegel, Zylinder) ergänzt. Im ersten Schritt wird die grundlegende Architektur des Systems untersucht. Um zu entscheiden, ob das System nach dem OVA- oder dem OVO-Ansatz aufgebaut werden soll, werden beide Ansätze mit einem gleichartigen CNN evaluiert. Es wird sowohl der modulare Ansatz als auch die Mehrklassen-Klassifikation des monolithischen Ansatzes bewertet. Die Netzstruktur, die hierfür verwendet wird, besteht aus zwei Convolutional- und zwei Max-Pooling-Layern sowie eines Fully Connected-Layers, siehe Abbildung 3.20. Die Convolutional Layer haben jeweils eine Kernelgröße von  $3 \times 3$ , verwenden 64 Filter und werden mit der ReLU-Aktivierungsfunktion (Rectified Linear Unit)

### 3. In-situ Fehlererkennung im Druckprozess

aktiviert. Der Max-Pooling Layer hat eine Kernelgröße von  $2 \times 2$  und das Fully Connected Layer enthält 64 Neuronen. Während beim binären Klassifikator lediglich ein Neuron mit der Sigmoid-Aktivierungsfunktion verwendet wird, verwendet die Mehrklassen-Klassifikation eine Ausgabeschicht von fünf Neuronen mit der Softmax-Aktivierungsfunktion. Das flattening wird über eine Global-Average-Pooling Operation umgesetzt. Als Loss-Funktion der Netze wird Binary Crossentropy bzw. Sparse Categorical Crossentropy und als Optimierer wird Adam verwendet. Alle Parameter wurden anhand von Erfahrungswerten ausgewählt und entsprechen einigen üblichen Modellen aus der Literatur.

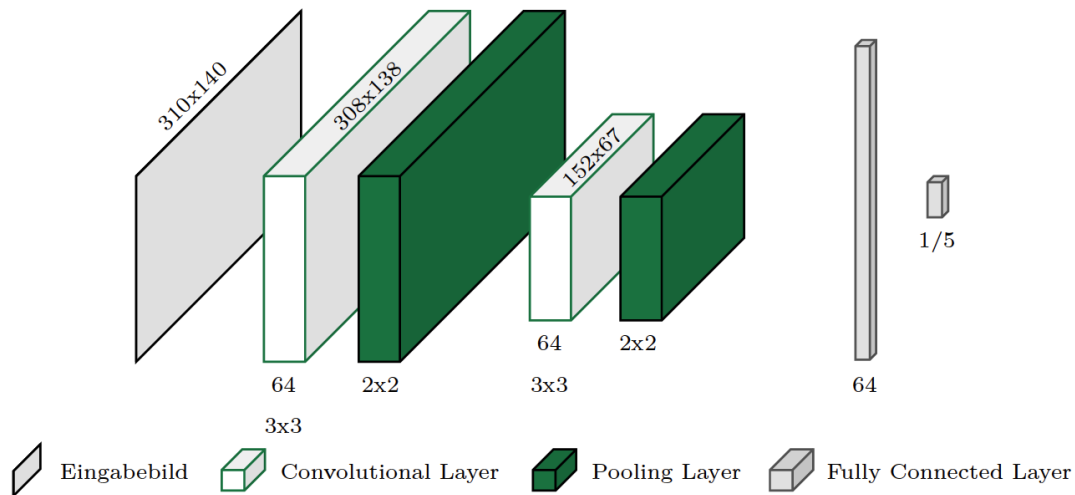


Abbildung 3.20.: Initiale Netzarchitektur des CNNs.

Ausgabeschicht Binär-Klassifikation: 1 Neuron,  
 Ausgabeschicht, Mehrklassen-Klassifikation: 5 Neuronen,  
 Loss-Funktion: Binary Crossentropy / Sparse Categorical Crossentropy, Opti-  
 mierer: Adam. Quelle [186].

Um die unterschiedlichen Modelle zu trainieren, werden zunächst die Trainingsdatensätze vorbereitet. Hierfür wird der Datensatz des Würfels verwendet, aus dem Bildbeispiele extrahiert werden. Der eingesetzte Datensatz enthält ausschließlich Bilder des Würfel-Drucks mit den Fehlerklassen Gut, Warping, Layer Shift, Spaghetti und Clogged Nozzle. Insgesamt sind 66 Druckvorgänge enthalten, was zu einer Gesamtgröße von 83.386 Bildern führte.

Der OVA-Ansatz erfordert fünf Modelle, die mit jeweils einem separaten Trainingsdatensatz trainiert werden. Diese Datensätze bestehen jeweils aus Daten einer Fehlerklasse und allen anderen Fehlerklassen. Hierbei werden insgesamt so viele Daten der „nicht-Fehlerklasse“ verwendet, dass die Anzahl mit der Datenanzahl der Fehlerklasse übereinstimmt. Beispielsweise werden bei 10.000 Warping-Daten jeweils 2.500 Layer Shift-, Gut-, Spaghetti- und Clogged Nozzle-Daten verwendet.

Der OVO-Ansatz setzt sich aus zehn Modellen zusammen und erfordert deshalb zehn Trainingsdatensätze. Diese Datensätze bestehen jeweils aus Daten zweier Fehlerklassen mit jeweils 2.500 Bildbeispielen. Ein Beispiel für die Zusammensetzung der Trainingsdatensätze ist in Abbildung 3.21 dargestellt.

### 3. In-situ Fehlererkennung im Druckprozess

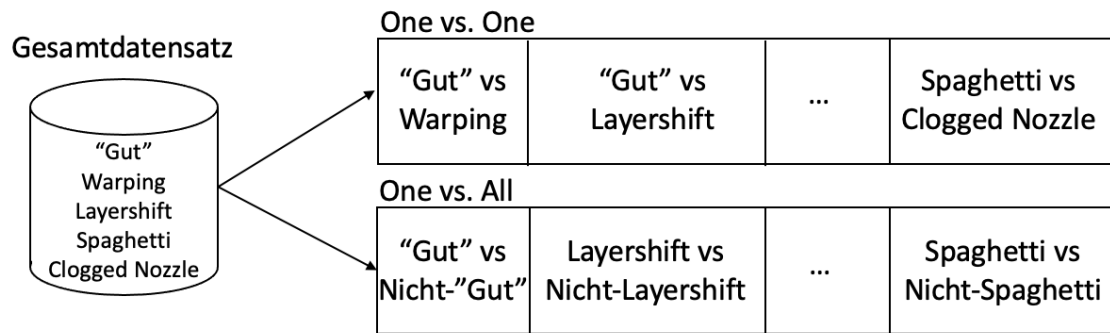


Abbildung 3.21.: Aufteilung der Trainingsdaten für den One vs. One- und den One vs. All-Ansatz. Quelle [186].

Die einzelnen Datensätze werden jeweils in Trainings-, Validierungs- und Testdatensätze unterteilt. Der Trainingsdatensatz wird verwendet, um die Muster und Merkmale der Bilder durch Anpassung der Gewichte zu lernen, während der Validierungsdatensatz verwendet wird, um die Hyperparameter zu optimieren. Mit dem entsprechenden Testdatensatz wird die tatsächliche Leistungsfähigkeit des Klassifikators bestimmt. Die binären Klassifikatoren sowie der Mehrklassen-Klassifikator werden mit einer Batchgröße von 32 für 50 Epochen mit den Trainingsdaten trainiert und mit den Testdaten validiert. Um die Vorhersagegenauigkeit zu bewerten, müssen die Ausgaben der beiden modularen Ansätze zunächst zu einer gesamtheitlichen Vorhersage kombiniert werden.

Beim OVO-Ansatz wird die Ausgabe ausgewählt, die am häufigsten vorhergesagt wurde. Es zeigt sich, dass binäre Klassifikatoren mit einem geringen Konfidenzniveau oft entscheidend für die Gesamtausgabe sind, was zu einer relativ schlechten Accuracy von weniger als 50% auf den Testdaten führt. Durch die Einführung eines Schwellenwerts können die Ausgaben solcher „unsicherer“ Klassifikatoren unterdrückt werden. Dadurch wurde die Accuracy auf 61% bis 72% gesteigert. Es zeigt sich, dass die Leistungsfähigkeit des Systems stark vom Schwellenwert abhängt und dieser die zentrale Optimierungsinstante darstellt.

Beim OVA-Ansatz wird jeweils der Klassifikator ausgewählt, der das höchste Konfidenzniveau aufweist, und damit eine Accuracy von 69% auf den Testdaten erreicht. Mit demselben Netzwerk wird zusätzlich ein Mehrklassen-Klassifikator als Ausprägung des monolithischen Ansatzes trainiert, der eine Accuracy von 62,5% auf den Testdaten erreicht.

In den untersuchten Systemen zeigt sich eine hohe Überanpassung, da die Accuracy auf den Trainingsdaten in allen Fällen 95% übersteigt. Es werden verschiedene Regularisierungstechniken benötigt, um die Überanpassung zu vermindern. Zudem erreichen die Systeme auf den Validierungsdaten eine ähnlich hohe Accuracy wie auf den Trainingsdaten, obwohl es sich um noch nicht gesehene Dateninstanzen handelt. Dies zeigt, dass sich die Daten innerhalb des Datensatzes zu sehr ähneln, um sie in Trainings-, Validierungs- und Testdaten aufzuteilen. Um unabhängige Testdaten zu erhalten, werden für die Testdaten Bilder verwendet, die in separaten Drucken erzeugt wurden. Die Variabilität in Bezug auf Licht, Szenerie und Fehlermerkmale reicht hierbei aus, um unabhängige Testdaten zu erzeugen.

### 3. In-situ Fehlererkennung im Druckprozess

Der durchgeführte Versuch zeigt, dass CNNs in der Lage sind, die Fehlerklassen erfolgreich zu unterscheiden. Modulare Systeme erzielen im Vergleich zum monolithischen Ansatz eine höhere Vorhersagegenauigkeit. Der OVO-Ansatz erfordert jedoch einen höheren Vorbereitungs- und Nachbereitungsaufwand und ist durch die Notwendigkeit eines Schwellenwerts und die Abhängigkeit der Vorhersagegenauigkeit von diesem Wert eingeschränkt. Es wurde auch festgestellt, dass der optimale Schwellenwert von den verwendeten Dateninstanzen abhängt. Das System sollte jedoch auf allen vergleichbaren Daten gleich gute Ergebnisse liefern. Aufgrund dieser Nachteile und der geringeren Vorhersagegenauigkeit wird für das in dieser Arbeit aufgebaute System der OVA-Ansatz gewählt.

Die Ergebnisse sind in Tabelle 3.1 zusammengefasst.

Tabelle 3.1.: Übersicht der verglichenen Ansätze.

	<b>Accuracy</b>	<b>Anzahl zu trainierender Modelle</b>
<b>OVA</b>	<b>69,2%</b>	<b>5</b>
<b>OVO</b>	66,2%	10
<b>Mehrklassen</b>	62,5%	<b>1</b>

#### Netzarchitektur der Fehlerklassen-Module

Im nächsten Schritt wird die Grundstruktur der einzelnen Komponenten des modularen Systems untersucht. Das Ziel ist es, ein System aus möglichst flachen und schlanken Einzelmodulen mit maximaler Vorhersagegenauigkeit aufzubauen. Es wird untersucht, ob die Vorhersagegenauigkeit bei gleicher Grundstruktur je nach Fehlerklasse variiert und ob die gleiche Netzstruktur für alle Einzelmodule verwendet werden sollte. Auf Basis dieser Analyse werden die Einzelmodule detailliert betrachtet. Um die Netzstrukturen miteinander zu vergleichen, sind die spezifischen Parameter der verschiedenen Netze weitgehend identisch. Um unabhängige Testdaten zu erhalten, werden Bilder verwendet, die in separaten Drucken erzeugt wurden. Die Validierungsmetrik ist die Vorhersagegenauigkeit auf dem Testdatensatz des Würfels.

### 3. In-situ Fehlererkennung im Druckprozess

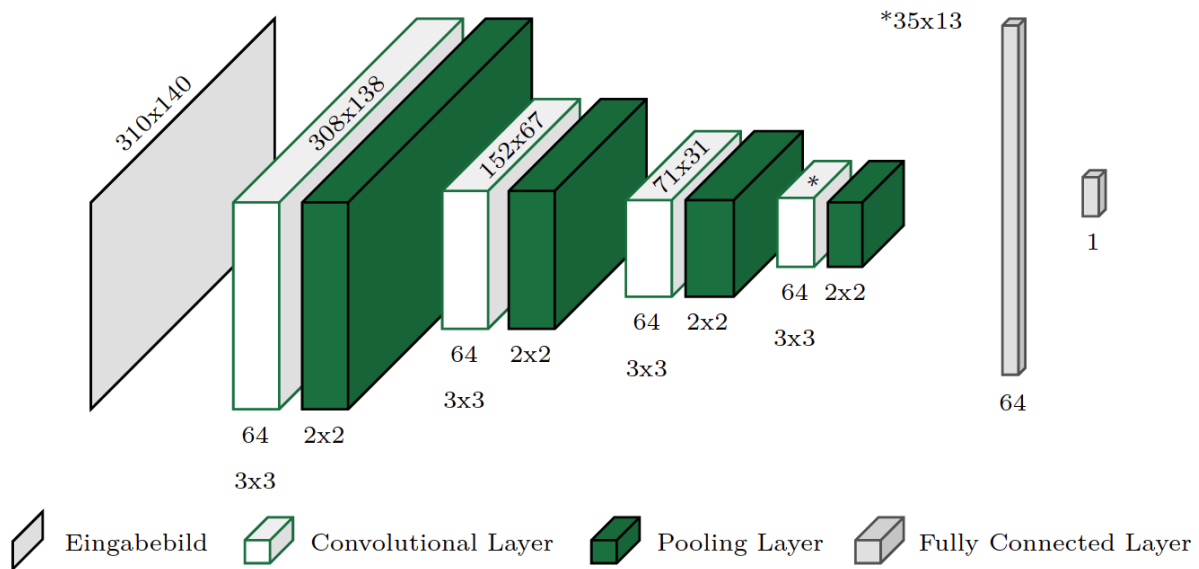


Abbildung 3.22.: Netzstruktur 4CNN-1Dense. Das Netz wird über den Adam-Optimierer und die Binary Crossentropy Loss-Funktion trainiert. Die Ausgabeschicht besteht aus einem Neuron. Quelle [186].

Tabelle 3.2.: Klassifizierungsergebnisse des 4CNN-1Dense Netzes

Netzstruktur	Gut	Warping	Layer Shift	Spaghetti	Clogged	Accuracy
4CNN-1Dense	67,5%	55,0%	95,8%	99,2%	91,7%	81,8%

### Konventionelle Netzstrukturen

**4CNN-1Dense** In der ersten untersuchten Netzstruktur wird die ursprüngliche Grundstruktur um zwei zusätzliche Convolution- und Max-Pooling-Layer erweitert. Durch diese zusätzlichen Faltungsschichten können räumliche Hierarchien von Mustern besser erkannt werden. Wenn die vorderen Layer einfache Muster wie Kanten lernen, können die tieferen Layer aus diesem Vorwissen einfachen Formen zusammensetzen. Mit den zusätzlichen Pooling-Layern wird die Anzahl der zu erlernenden Parameter reduziert. Sie tragen auch dazu bei, die räumlichen Hierarchien zu verstärken, indem die aufeinanderfolgenden Convolutional-Layer im Vergleich zur Anfangseingabe zunehmend größere Fenster betrachten. Die Anzahl der Filter sowie die Größe der Kernel werden in jedem Layer beibehalten, um die Aussagekraft gegenüber anderen getesteten Grundstrukturen nicht zu beeinträchtigen. In Abbildung 3.22 ist der gewählte Netzaufbau visualisiert.

Mit den gewählten Parametern von 30 Epochen und einer Batch-Größe von 32 im Systemaufbau konnte eine Gesamt-Accuracy von 81,77% erreicht werden. Ausreißer sind vor allem die Fehlerklassen Warping (55%) und Gut (67,5%). Alle Klassifizierungsergebnisse sind in Tabelle 3.2 aufgeschlüsselt.

**2CNN-3Dense und 4CNN-3Dense** Wie auch schon bei der Struktur von 4CNN-1Dense, wurden auch 2CNN-3Dense und 4CNN-3Dense evaluiert. Die jeweiligen Strukturen ähneln sich sehr stark und unterscheiden sich hauptsächlich in der Anzahl der einzelnen Layer-Typen, wie CNN- oder Dense-Layer. Einen Vergleich dieser drei verschiedenen Strukturen ist in Tabelle 3.3 dargestellt.

Tabelle 3.3.: Vergleich der konventionellen Netzstrukturen untereinander.

Netzstruktur	Gut	Warping	Layer Shift	Spaghetti	Clogged	Accuracy
4CNN-1Dense	<b>67,5%</b>	55,0%	<b>95,8%</b>	<b>99,2%</b>	<b>91,7%</b>	<b>81,8%</b>
4CNN-3Dense	30,8%	<b>90,0%</b>	<b>95,8%</b>	99,2%	86,7%	80,4%
2CNN-3Dense	53,3%	94,2%	63,3%	<b>96,7%</b>	63,4%	74,4%

### 3.2.5. Verbesserung des Datensatzes und der Netzarchitekturen

#### Erweiterung des Datensatzes

Die bisherigen Experimente zur Grundstruktur und Systemtopologie wurden mit einem Trainingsdatensatz bestehend aus dem Objekt Würfel durchgeführt und getestet. Im nächsten Schritt wird der Trainingsdatensatz um weitere vier Grundkörper erweitert, Beispiele dieser Objekte sind in Abbildung 3.23 gezeigt. Der Datensatz wurde neben dem Würfel um die Grundkörper Pyramide, Zylinder, Kegel und Halbkugel erweitert und so eine Gesamtanzahl von 153 Drucken mit daraus resultierenden 218.103 Bildern erreicht. Diese Grundkörper wurden ausgewählt, um weitere Oberflächenstrukturen wie konisch oder kugelförmig abzudecken und dadurch das Leistungsspektrum des Modells zu erweitern.

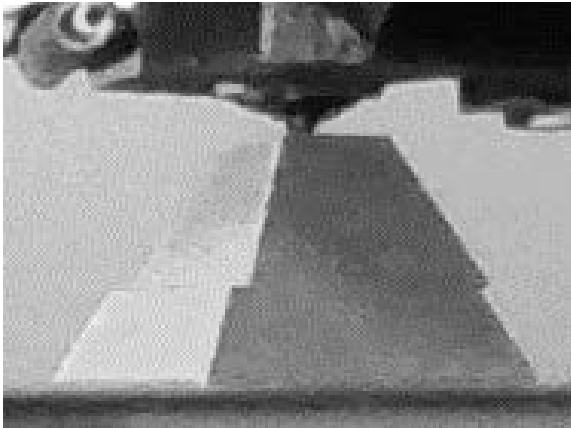
Um das Training zu beschleunigen und um die Schwankungen der Anzahl der Bildinstanzen je Objekt und Fehlerklasse zu minimieren, werden Teildatensätze aus dem Gesamtdatensatz für das Training der binären Klassifikatoren verwendet. Dies reduziert auch die Menge an redundanten Informationen im Trainingsdatensatz. Jeder so generierte Trainingsdatensatz besteht beispielsweise aus 10.000 Bildinstanzen für die Klasse Layer Shift und Nicht-Layer Shift. Die ausgewählten Daten bestehen zu 50% aus Würfeln und Grundkörper-Objekten und werden in einem separaten Trainingsdatensatz gesammelt. Diese Trainingsdatensätze werden verwendet, um die einzelnen binären Klassifikatoren zu trainieren und den Systemaufbau zu testen. Für die Testdaten werden separate Drucke erstellt. Mit einer Batch-Size von 32 und 30 Epochen erreicht der Systemaufbau eine Accuracy von 91,87% auf den Testdaten. siehe Tabelle 3.4.

Tabelle 3.4.: Ergebnisse mit erweitertem Datensatz

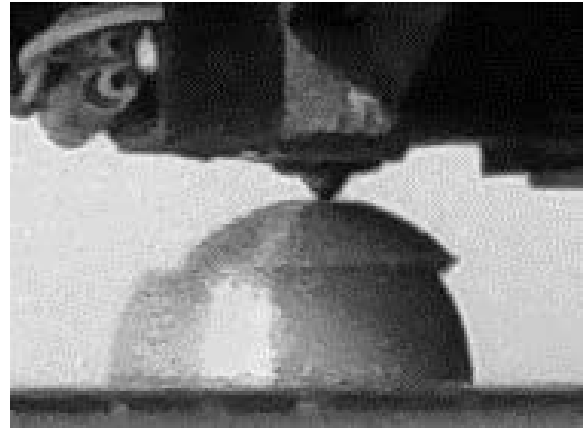
	Gut	Warping	Layer Shift	Spaghetti	Clogged	Accuracy
Würfel	67,5%	<b>95,8%</b>	95,8%	<b>99,2%</b>	<b>90,0%</b>	90,1%
Grundkörper	<b>89,1%</b>	89,5%	<b>98,3%</b>	98,8%	84,4%	<b>91,9%</b>



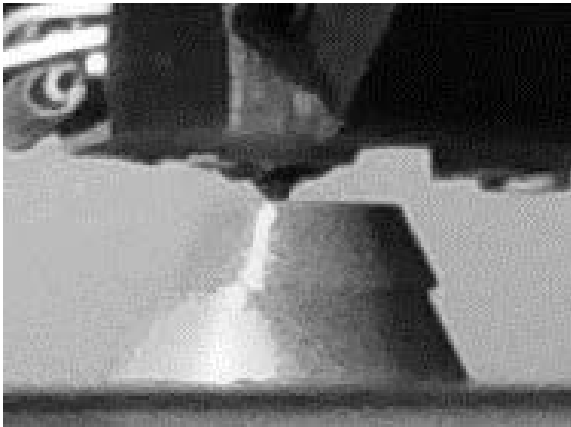
### 3. In-situ Fehlererkennung im Druckprozess



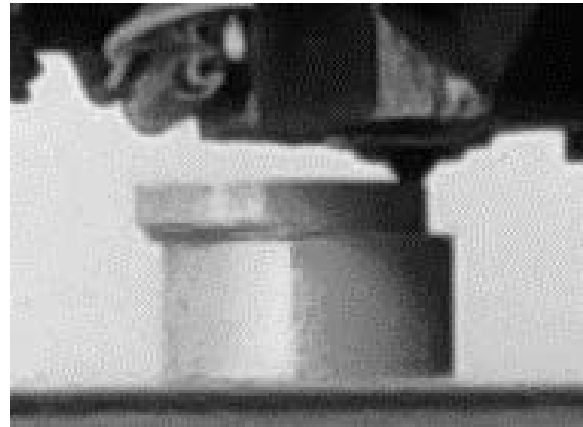
(a) Pyramide



(b) Halbkugel



(c) Kegel



(d) Zylinder

Abbildung 3.23.: Zusätzlich verwendete Objekte mit Fehler im Datensatz. Verschiedene Geometrien, die im bisherigen Datensatz noch nicht vertreten waren. Quelle [186].

Es hat sich gezeigt, dass die Vorhersagegenauigkeit des Systemaufbaus durch Erweiterung des Trainingsdatensatzes steigt, obwohl die Aufgabe bei verschiedenen zu klassifizierenden Objekten schwieriger wird. Die binären Klassifikatoren scheinen hauptsächlich von der höheren Anzahl an Drucken und der gleichzeitig steigenden Varianz in den Trainingsdatensätzen zu profitieren. Im nächsten Schritt soll untersucht werden, ob das Ergebnis durch die Verwendung von Regularisierungstechniken und Hyperparameter-Tuning verbessert werden kann.

#### **Class Activation Maps**

Trotz der stetigen Leistungssteigerung von neuronalen Netzen (NNs) sind sie aufgrund ihrer fehlenden Zerlegbarkeit in intuitive und verständliche Komponenten schwer zu interpretieren [121].

Dies erschwert es oft, zu erklären, warum ein intelligentes System, das auf den Trainingsdaten noch hohe Genauigkeit erzielte, plötzlich versagt. Ein Modell kann scheitern, wenn es sich beispielsweise an ein Objekt im Hintergrund anpasst, das nur bei Bildern einer bestimmten Klasse A zu sehen ist. Wenn das Objekt auf einem Bild der Klasse B zu sehen ist, klassifiziert das Modell das Bild fälschlicherweise als Klasse A. Um die Akzeptanz und das Vertrauen in solche intelligenten Systeme zu erhöhen, ist es wichtig, die Entscheidungen der Modelle transparenter zu machen und zu erklären, warum sie etwas vorhersagen.

CNNs bieten im Vergleich zu NNs den Vorteil, dass sie auf visuellen Konzepten basieren, die leicht zu visualisieren und zu interpretieren sind. Beispielsweise können die Filter der Convolution Layer visualisiert werden, um die visuellen Muster oder Merkmale zu erkennen, auf die jeder Filter empfindlich ist [122]. Eine weitere Möglichkeit ist die Visualisierung von Heatmaps, die anhand der Gradienteninformation der letzten Faltungsschicht die Wichtigkeit eines Neurons für eine bestimmte Entscheidung zeigen. So kann visualisiert werden, auf welchen Bereichen des Bildes eine Klasse vom Klassifikator ausgewählt wird. Dieser Ansatz wurde von Selvaraju *et al.* eingeführt und heißt Gradient-weighted Class Activation Mapping (Grad-CAM) [123].

Um sicherzustellen, dass sich das vorgestellte System tatsächlich auf die visuellen Merkmale der Fehlerklassen anpasst und nicht auf eine bestimmte Höhe des Druckkopfes, wird der Grad-CAM-Ansatz angewendet. Durch die Visualisierung kann auch die Überanpassung des Systems auf den Trainingsdaten erklärt werden. Mit diesem Ansatz werden die einzelnen binären Klassifikatoren der jeweiligen Fehlerklassen anhand einiger Beispielbilder getestet.

Abbildung 3.24 (a) zeigt, dass die Bereiche, in denen die optischen Merkmale des Layer Shifts auftreten, auch von den Class Activation Maps (CAMs) hoch gewichtet werden. Die höchste Gewichtung erhält jedoch der für die Fehlerklasse unwichtige Extruder. In Abbildung 3.24 (b) werden für den binären Klassifikator der Klasse Spaghetti hauptsächlich visuelle Merkmale des Fehlers markiert. Es zeigt sich, dass ein Objekt im Hintergrund ebenfalls eine hohe Gewichtung hat. Da dieser Effekt parallel zu den richtigen Bereichen auftritt, ist er in diesem Bild nicht besonders problematisch, kann sich aber negativ auf die Vorhersagegenauigkeit bei anderen Bildern auswirken. Gleiches gilt auch für die Abbildungen 3.24 (c) und (d).

In Zusammenfassung ergab die Untersuchung mit Grad-CAM, dass ein Großteil der binären Klassifikatoren sich auf die relevanten visuellen Merkmale der Fehlerklassen anpasst. Trotzdem werden häufig Bereiche im Hintergrund hervorgehoben oder Entscheidungen unabhängig vom Druckobjekt getroffen, siehe Abbildung 3.24. Das erklärt teilweise die starke Überanpassung des Systems auf den Trainingsdaten und zeigt, dass die Klassifikatoren nachjustiert werden müssen.

### 3. In-situ Fehlererkennung im Druckprozess

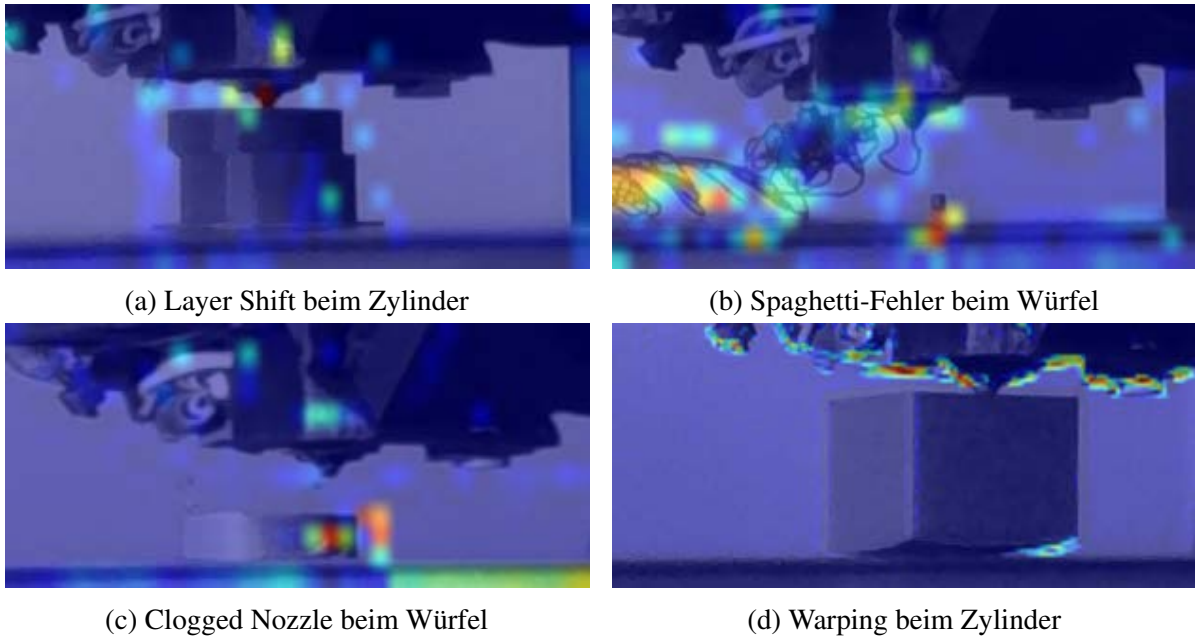


Abbildung 3.24.: Grad-CAM Visualisierung verschiedener Fehlerklassen. Die Relevanz des entsprechenden Bereichs ist durch die „Wärme“ in der Heatmap dargestellt. Je heller der Farbton eines Bereichs, desto relevanter ist dieser für die Klassifikation. Quelle [186].

#### Regularisieren

Die bisherigen Untersuchungen zeigen eine Überanpassung der binären Klassifikatoren. Obwohl eine Accuracy von über 95% auf den Trainingsdaten erreicht wird, sind die Ergebnisse auf den Testdaten, insbesondere bei den binären Experten der Fehlerklassen Warping (85,6%), Clogged Nozzle (81,5%) und Gut (71,9%), deutlich niedriger. Der Effekt wird im Systemaufbau zwar teilweise kompensiert, lässt aber trotzdem Handlungsspielraum für Verbesserungen.

Im Folgenden werden die Auswirkungen von verschiedenen Regularisierungstechniken untersucht. Beim Regularisieren wird die Leistungsfähigkeit sowohl der binären Klassifikatoren als auch des gesamten Systemaufbaus bewertet. Als Metrik werden die Accuracy und der Anteil der fehlerhaften Bildinstanzen, die fälschlicherweise als „Gut“ klassifiziert werden, verwendet. Diese Kennzahl ist relevant, da sie beschreibt, wie viele Prozent der fehlerhaften Bilder vom System nicht erkannt werden und wird als False Negative Rate (FNR) bezeichnet und berechnet sich durch die Summe der fälschlicherweise als „Gut“ klassifizierten Bilder, dividiert durch die Gesamtanzahl aller fehlerbehafteten Bilder.

$$\text{False Negative Rate (FNR)} = \frac{\sum \text{False Negative (FN)}}{\sum \text{True Positive} + \sum \text{False Negative}} \quad (3.1)$$

#### Data Augmentation

Um realistische Trainingsdaten zu generieren, werden Methoden der Data Augmentation angewendet, die einen Betrachter nicht erkennen lassen, welche Dateninstanzen generiert und welche nicht generiert wurden. Es ist wichtig, dass das visuelle Merkmal der Klassifikation erhalten bleibt. Aus diesem Grund werden Methoden wie Cropping und Drehung ausgeschlossen. In dieser Arbeit werden die Techniken Spiegelung und Helligkeitsvariation angewendet, siehe Abbildung 3.25. Um sicherzustellen, dass das Netz sich nicht auf die optischen Merkmale des Extruders einstellt, werden die Bilder um die vertikale Achse gespiegelt. Zur Helligkeitsvariation wird die Helligkeit der Bilder innerhalb eines Bereichs zufällig verändert. Um Speicherplatz zu sparen, werden diese künstlichen Daten erst während des Trainingsprozesses erzeugt und den Trainingsdaten hinzugefügt.

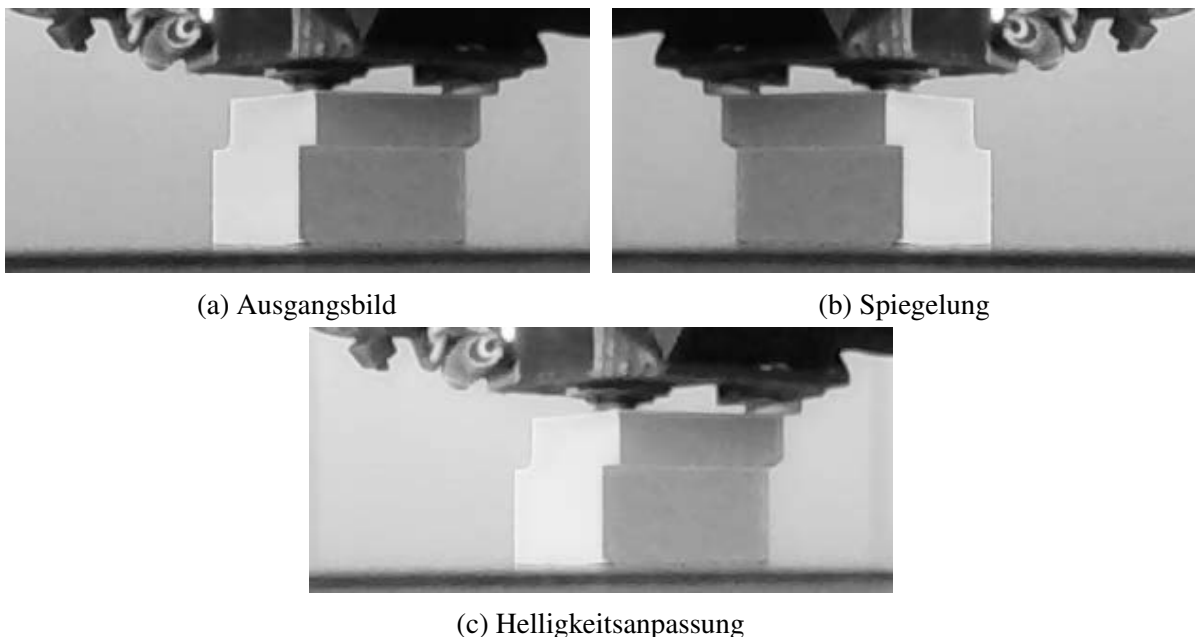


Abbildung 3.25.: Erweiterung der Trainingsdaten durch Data Augmentation wie z.B. durch Spiegelung oder Helligkeitsvariation. Quelle [186].

Zunächst werden die binären Klassifikatoren untersucht, indem die vorhandenen Netzstrukturen erneut mit künstlich erweiterten Trainingsdaten für 30 Epochen mit einer Batch-Größe von 32 trainiert werden. Tabelle 3.5 zeigt die Accuracy der binären Klassifikatoren zusammengefasst. Es zeigt sich, dass die Data Augmentation, außer beim Klassifikator der Fehlerklasse Spaghetti und Layer Shift, eine Verbesserung gegenüber der Basis bewirkt. In der Evaluierung im Systemaufbau ergibt sich mit der Data Augmentation eine Accuracy von 91,29%, was einer Verschlechterung von 0,58% gegenüber der Basis entspricht. Der Anteil der fehlerhaft als Gut klassifizierten Bildinstanzen sinkt im Systemaufbau von 6,6% auf 5,7%.

### 3. In-situ Fehlererkennung im Druckprozess

Tabelle 3.5.: Accuracy nach Erweitern des Datensatzes durch Data Augmentation.

	Gut	Warping	Layer Shift	Spaghetti	Clogged
Basis	79,3%	91,8%	<b>97,7%</b>	<b>98,9%</b>	87,4%
Augmentation	<b>83,1%</b>	<b>93,6%</b>	96,6%	97,1%	<b>89,3%</b>

#### Dropout

Dropout ist ein häufig verwendetes Verfahren zur Regularisierung von tiefen NNs und hat in der Praxis vor allem in Fully Connected Layern Erfolg gezeigt [124]). In CNNs war die Anwendung jedoch lange Zeit umstritten, da die Anzahl der Parameter für die Faltungsschichten im Verhältnis zu den Aktivierungen gering ist [125]. Studien von Wu und Gu [126] sowie Park und Kwak [125] konnten jedoch nachweisen, dass bei Convolutional Layer die Verwendung von Dropout positive Auswirkungen hat. Im Prädikationsblock wird vor jedem Dense Layer ein Dropout Layer mit einer Dropout Rate von 0,5 eingebaut. Eine Dropout Rate von 0,5 ist die übliche Einstellung bei Dense Layer [124]. Wu und Gu ([126] haben jedoch gezeigt, dass bei Convolutional Layer eine Dropout Rate von 0,5 zu einer Verschlechterung der Netzwerkleistung führt. Deshalb werden nach den Pooling Layer im Konstruktionsblock jeweils Dropout Layer mit einer Dropout Rate von 0,2 eingebaut. Die Accuracy der binären Klassifikatoren nach dem erneuten Training der angepassten Netzstrukturen ist in Tabelle 3.6 aufgeführt.

Tabelle 3.6.: Klassifizierungsergebnisse nach Kombination von Dropout und Data Augmentation.

	Gut	Warping	Layer Shift	Spaghetti	Clogged
Basis	<b>79,3%</b>	91,8%	<b>97,7%</b>	<b>98,9%</b>	87,4%
Dropout	75,5%	<b>94,8%</b>	96,0%	95,1%	<b>91,5%</b>

Die Anwendung von Dropout hat für einen Großteil der binären Klassifikatoren eine Verschlechterung der Accuracy zur Folge. Im Systemaufbau verschlechtert sich die Accuracy von 91,87% auf 90,55%. Der Anteil der nicht erkannten Fehler sinkt dagegen von 6,6% auf 5,6%. Es ist möglich, dass durch das genutzte Dropout in den Faltungsschichten bestimmte Neuronenbereiche, die für die Entscheidung wichtig sind, ausgeschlossen werden. Es könnte auch daran liegen, dass der Algorithmus durch die Verringerung der Parameter und Verbindungen die Merkmale nicht richtig lernt. Um dieser Unteranpassung entgegenzuwirken, könnte eine Erweiterung des Trainingsdatensatzes helfen. Hier bietet es sich an, die beiden Regularisierungstechniken Dropout und Data Augmentation zu kombinieren.

Die Kombination von Data Augmentation und Dropout führt zu einer Steigerung der Accuracy von binären Klassifikatoren wie Gut, Warping und Layer Shift im Vergleich zur Basis. Allerdings hat diese Kombination bei Spaghetti und Clogged Nozzle einen negativen Einfluss auf die Accuracy. Im Systemaufbau erzielt die Kombination (Augm-Drop) mit einer Accuracy von 93,37% eine höhere Erkennungsrate als einzelne Ansätze. Der Anteil nicht erkannten Fehler sinkt dabei auf 3,5%.

### 3. In-situ Fehlererkennung im Druckprozess

Die letzte Untersuchung betrachtet, wie sich der Systemaufbau verhält, wenn für jede Fehlerklasse die Regularisierungsmethode mit der höchsten Accuracy ausgewählt wird. Diese Mischung der Regularisierungsmethoden erzielt eine Accuracy von 94,4% und eine FNR von 3,4%, was alle bisherigen getesteten Systeme übertrifft.

#### Early Stopping

Bisher wurden die binären Klassifikatoren mit einer festen Anzahl von 30 Epochen trainiert. Es kann jedoch sein, dass einfachere zu erkennende Fehler bereits früher gelernt werden. Der binäre Klassifikator der Fehlerklasse Spaghetti erkennt beispielsweise Bilder ohne die bisher gezeigten Regularisierungstechniken mit höherer Genauigkeit, siehe Tabelle 6.15, was zeigt, dass zu viel Information die Vorhersagegenauigkeit beeinträchtigen kann. Eine zu hohe Anzahl an Epochen kann ebenfalls zu einer Überanpassung oder Verstärkung dieses Effekts führen.

Early Stopping ist eine Methode zur Verhinderung von Überanpassung, bei der das Training anhand der Vorhersagegenauigkeit auf den Validierungsdaten frühzeitig beendet wird. Hierfür können verschiedene Metriken verwendet werden. Das Training wird abgebrochen, wenn sich der Validierungsfehler über eine bestimmte Anzahl an Epochen  $e$  nicht mehr verbessert. Die Gewichte und Parameter werden nach dem Abbruch auf den Stand des Netzes bei minimalem Validierungsfehler zurückgesetzt. Die mit den bisherigen Verfahren regulierten Netze werden mit einer maximalen Anzahl von 50 Epochen trainiert und nach entweder drei, fünf oder acht Epochen ohne Minderung des Validierungsfehlers abgebrochen. Alle Ergebnisse sind in Tabelle 3.7 dargestellt.

Tabelle 3.7.: Ergebnisse mit Anwendung von Early Stopping

	Gut	Warping	Layer Shift	Spaghetti	Clogged
Augm-Drop	<b>83,5%</b>	<b>94,8%</b>	98,4%	94,7%	89,1%
$e = 8$	78,3%	94,1%	98,7%	98,3%	90,7%
$e = 5$	78,9%	91,4%	99,4%	<b>99,0%</b>	89,0%
$e = 3$	72,3%	93,4%	<b>99,4%</b>	98,0%	<b>96,0%</b>

Da sich die Accuracy für die Klassen Gut und Warping mit 30 Epochen nicht verbessert, scheint diese Anzahl gut geeignet zu sein. Bei den verbleibenden Fehlerklassen kann durch Early Stopping und den vorzeitigen Abbruch des Trainings eine Steigerung der Accuracy erreicht werden. Im Systemaufbau führt Early Stopping zu einer Gesamt-Accuracy von 94,61%, was einer Steigerung von 1,21% im Vergleich zu Augm-Drop entspricht. Zudem übertrifft dieser Ansatz auch die Mischung aus dem vorherigen Versuch um 0,21%. Die FNR sinkt mit Early Stopping auf 2,6%.

## Beurteilung der erzielten Ergebnisse

Die getesteten Regularisierungsmethoden erzielen die gewünschte Wirkung. Die Überanpassung der einzelnen binären Klassifikatoren konnte signifikant reduziert werden. Die Accuracy des Systemaufbaus erhöhte sich von 91,9% auf 94,6%, was einer Steigerung von 2,7% entspricht, siehe Tabelle 3.8.

Tabelle 3.8.: Übersicht der Erkennungsgenauigkeit, sowie der Falsch-Negativ-Rate, bei den verschiedenen, angewandten Schritten.

	Accuracy	FNR
Basis	91,9%	6,6%
Augmentation	91,3%	5,7%
Dropout	90,6%	5,6%
Augm-Drop	93,4%	3,5%
Mischung	94,4%	3,4%
Early Stopping-Augmentation-Dropout	<b>94,6%</b>	<b>2,6%</b>

Obwohl die Steigerung der Accuracy um 2,7% auf den ersten Blick nicht beeindruckend scheint, bedeutet dies eine Reduzierung der Fehlerrate des Systems um mehr als 30%. Die Regularisierungstechniken erzielen nicht bei jeder Fehlerklasse gleich positive Ergebnisse. So verschlechtert sich die Accuracy der vermeintlich einfachsten Klassifikation der Fehlerklasse Spaghetti mit der Anwendung von Data Augmentation und Dropout. Die Untersuchung mit Early Stopping zeigt, dass die binären Klassifikatoren der Klassen Gut und Warming im Vergleich zu den anderen Klassifikatoren nicht früher abgebrochen werden. Daraus ergibt sich, dass die Netzstrukturen für die Komplexität der Aufgabenstellung nicht ausreichend sind und sich das Netz in einem lokalen Minimum befindet. Mit einer finalen Accuracy nach der Regularisierung von 83,5% und 94,8% auf den binären Testdaten bieten diese Klassifikatoren das größte verbleibende Optimierungspotential, siehe Tabelle 3.7.

## Hyperparameter Tuning

Bisher wurden gute Ergebnisse mit den verwendeten Netzstrukturen erzielt, indem die Hyperparameter des binären Klassifikators nach Erfahrungswerten und „Best Practices“ ausgewählt wurden. Um die Vorhersagegenauigkeit des Systems weiter zu steigern, sollten die Hyperparameter entsprechend der gegebenen Aufgabenstellung optimiert werden. Eine manuelle Validierung aller möglichen Kombinationen von Parametern ist jedoch umständlich. Ein automatisierter und strukturierter Vorgang ist daher notwendig, um im immensen Hyperparameterraum effiziente Optimierungen durchzuführen. Beispielsweise werden in der Literatur bei CNNs die Hyperparameter Batch-Größe, Lernrate, Dropoutrate, Optimierer, Aktivierungsfunktion, Kernelgröße, Filteranzahl und die Anzahl der Neuronen in den Fully Connected Layers optimiert [127, 128, 129]. Da die Auswahl der richtigen Hyperparameter stark von der Aufgabenstellung abhängt, muss für diese Arbeit ein individuelles Set erstellt werden [130].

Es gibt zwei grundlegende Ansätze, um Hyperparameter zu optimieren: die Rastersuche und die Zufallssuche. Um den hohen Trainings- und Rechenaufwand zu minimieren, empfiehlt es sich, eine Kombination dieser beiden Ansätze zu verwenden. Zunächst wird eine grobe Rastersuche durchgeführt, um die relevanten und irrelevanten Hyperparameter für die Aufgabenstellung zu identifizieren. Dadurch wird der grobe Hyperparameterraum verständlicher und einfacher zu beurteilen. In einer zweiten Stufe werden die relevanten Hyperparameter erweitert und die irrelevanten ausgeschlossen. Da es in dieser Stufe deutlich mehr Parameter und somit Kombinationsmöglichkeiten gibt, wird hier die effizientere Zufallssuche angewendet. Dieser Ansatz wird für alle fünf binären Klassifikatoren angewendet. Für die Rastersuche werden die in Tabelle 3.9 aufgelisteten Parameter verwendet. Es entstehen insgesamt 64 Kombinationsmöglichkeiten, die nacheinander trainiert und getestet werden.

Tabelle 3.9.: Parameter zur Suchraumeingrenzung beim Hyperparameter Tuning am Beispiel Layer Shift

Hyperparameter	Abkürzung	Wertebereich
Aktivierungsfunktion	activation	['relu', 'elu']
Anzahl Dense Units	dense_1	[64,128]
Anzahl Filter	filter_1	[64,128]
Optimizer	optimizer	['adam', 'sgd']
Dropoutrate Conv	dropout	[0.2,0.3]
Dropoutrate Dense	dropout_1	[0.4,0.5]

Für jede Kombination von Hyperparametern wird die Accuracy auf dem binären Testdatensatz mit einem Algorithmus bestimmt und gespeichert. Anhand dieser Informationen können verschiedene Auswertungen durchgeführt werden, indem die Accuracy und die Hyperparameter betrachtet werden. Box-Whisker-Plots zeigen, welche Hyperparameter-Einstellungen sich am besten für die gegebene Aufgabenstellung eignen und welche nicht. In Abbildung 3.26 sind beispielhafte Box-Whisker-Plots für den binären Klassifikator der Fehlerklasse Layer Shift gezeigt. Durch die Rastersuche kann anhand dieser Ergebnisse lediglich der Optimierer



### 3. In-situ Fehlererkennung im Druckprozess

eingeschränkt werden. Der Median und der maximale Wert der Accuracy bei Verwendung des Adam-Optimierers sind deutlich höher als bei Verwendung des SGD-Optimierers. Daher kann der Optimierer für die folgende Zufallssuche fixiert werden. Obwohl der Median und der maximale Wert der Dropout rate des Convolutional Layers bei einer Einstellung von 0,3 höher sind als bei einer Einstellung von 0,2, wird die Differenz des maximalen Werts der Accuracy von 0,65% als zu gering eingeschätzt, um die Einstellung zu fixieren.

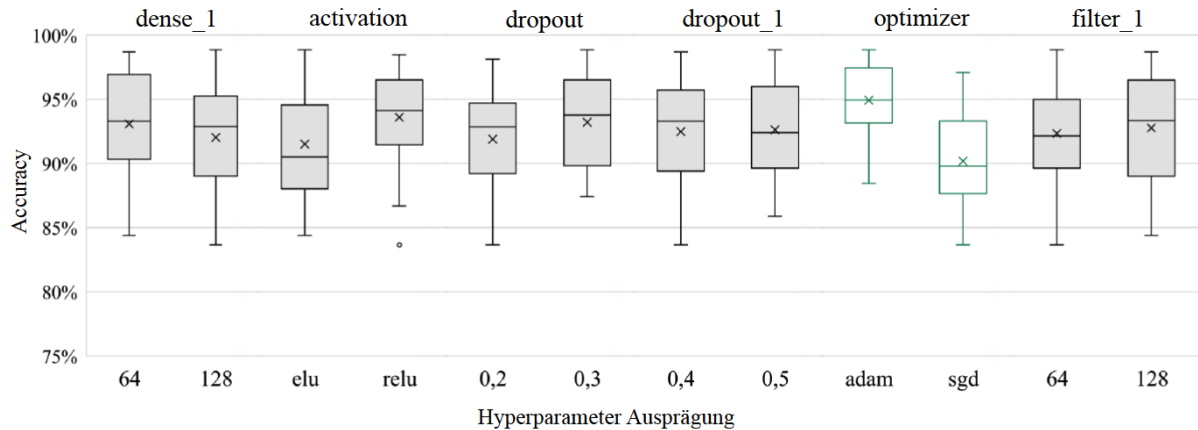


Abbildung 3.26.: Darstellung der Ergebnisse der Rastersuche am Beispiel Layer Shift. Quelle [186].

Die höchsten Werte der Accuracy, die durch die Rastersuche für die binären Klassifikatoren erreicht wurden, sind in Tabelle 3.10 aufgeführt. Die Ergebnisse zeigen die Verbesserung im Vergleich zu den Versionen ohne Early Stopping, da diese Regularisierungstechnik nicht in Verbindung mit der automatisierten Optimierung der Hyperparameter verwendet werden kann. Es zeigt sich, dass auch bei einem begrenzten Optimierungsraum von 64 Kombinationsmöglichkeiten eine Verbesserung erzielt wird. Im Vergleich zur vollständigen Regularisierung mit Early Stopping fällt die durchschnittliche Steigerung von 0,98% jedoch deutlich geringer aus.

Tabelle 3.10.: Ergebnisse der Klassifizierung nach erfolgter Rastersuche

	Gut	Warping	Layer Shift	Spaghetti	Clogged
Augm-Drop	95,2%	94,8%	98,4%	94,7%	89,1%
Rastersuche	<b>97,7%</b>	<b>97,0%</b>	<b>98,9%</b>	<b>98,2%</b>	<b>97,9%</b>

Um die Optimierung weiterzuführen, werden einige Hyperparameter, die in der Rastersuche verwendet wurden, erweitert. Da der Adam-Optimierer deutlich bessere Ergebnisse liefert als der SGD-Optimierer, wird dieser Hyperparameter fixiert und in der Optimierung nicht weiter untersucht. In der Literatur wurden insbesondere durch Anpassungen der Lernrate und der Anzahl an Filtern Verbesserungen erzielt. Daher werden bei der Zufallssuche für jedes Convolutional Layer separate Anzahlen an Filtern getestet. Außerdem wird der Hyperparameterraum um die Lernrate und die Größe der Kernel erweitert. Da sich mit diesen Erweiterungen bereits 26.244 Kombinationsmöglichkeiten ergeben, wird auf eine Unterscheidung der Kernel-Größe

### 3. In-situ Fehlererkennung im Druckprozess

je nach Convolutional Layer verzichtet. Die Auftretswahrscheinlichkeit der verschiedenen Ausprägungen der Hyperparameter ist jeweils gleich verteilt.

Die Zufallssuche ermöglicht eine weitere Verbesserung der Accuracy der binären Klassifikatoren, wie in Tabelle 3.11 dargestellt. Wie durch die Ergebnisse zu sehen ist trägt die Zufallssuche entscheidend dazu bei, das ein Optimum erreicht werden kann, insbesondere, wenn die Relevanz bestimmter Hyperparameter nicht klar ist.

Tabelle 3.11.: Ergebnisse der Klassifizierung nach erfolgter Zufallssuche

	Gut	Warping	Layer Shift	Spaghetti	Clogged
Augm-Drop	95,2%	94,8%	98,4%	94,7%	89,1%
Rastersuche	97,7%	97,0%	98,9%	98,2%	97,9%
Zufallssuche	<b>98,4%</b>	<b>97,7%</b>	<b>99,6%</b>	<b>99,6%</b>	<b>98,3%</b>

Es zeigt sich, dass mit Ausnahme der Aktivierungsfunktion und des Optimierers, die Ausprägungen der Hyperparameter für die gegebene Aufgabenstellung stark von der Fehlerklasse abhängen. Es ist auch zu beobachten, dass die CNNs durch die Anpassungen insgesamt komplexer werden. Eine Ausnahme ist die Fehlerklasse Spaghetti, bei der der binäre Klassifikator eine höhere Vorhersagegenauigkeit erzielt, wenn die Convolutional Layer weniger Filter verwenden. Dies zeigt, dass ein komplexeres Netz sich auch negativ auf die Leistungsfähigkeit auswirken kann. Ein besonderer Erfolg ist die Vergrößerung der Kernel, die eine deutliche Leistungssteigerung im Vergleich zur Ausgangssituation ermöglicht hat.

Da in beiden Varianten unterschiedliche Parameter verwendet wurden, lässt sich aus dieser Untersuchung nicht unmittelbar eine Aussage über die relative Leistungsfähigkeit der Optimierungsmethoden treffen. Trotzdem wird durch die schnellere Annäherung an das Optimum die Vorteilhaftigkeit der Zufallssuche deutlich.

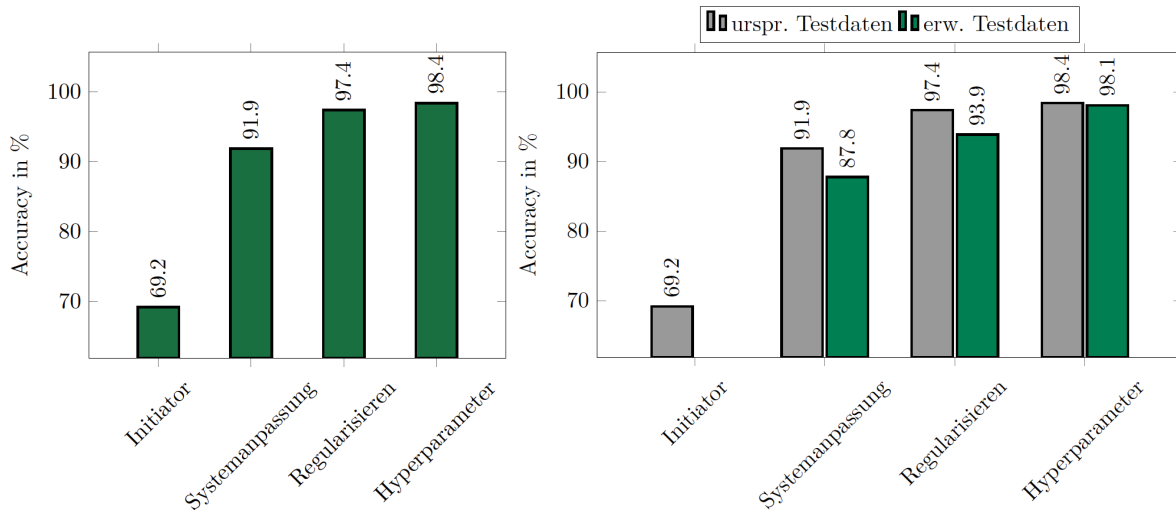
#### 3.2.6. Bewertung und Evaluation des modularen Ansatzes

Abbildung 3.27a zeigt die Entwicklung der Accuracy des modularen Modells. Durch die durchgeführten Maßnahmen konnte eine kontinuierliche Verbesserung erreicht werden. Der größte Effekt wurde durch die Systemanpassung mit einer Steigerung der Accuracy um 23% erzielt. Obwohl die Hyperparameteroptimierung nur eine Steigerung von 1% brachte, ist es auf dem höheren Leistungsniveau schwieriger, weitere Verbesserungen zu erzielen. Es sollte beachtet werden, dass die Accuracy mit einem einheitlichen Testdatensatz evaluiert wurde und das Leistungsniveau des entwickelten Modells möglicherweise von den verwendeten Testdaten abhängt.

Um die These zu überprüfen, wurde der Testdatensatz um weitere Drucke erweitert, obwohl die Anzahl der Testdaten gleich blieb. Dies führte jedoch dazu, dass die Varianz der optischen Fehlermerkmale im Testdatensatz erhöht wurde, was die Verallgemeinerbarkeit der Aussagen verbesserte. Besonders bei der Verwendung der erweiterten Testdaten führte die Hyperparameteroptimierung zu einer deutlichen Verschlechterung. In der Analyse zeigte sich, dass der

### 3. In-situ Fehlererkennung im Druckprozess

binäre Klassifikator der Klasse Gut nur noch 75% statt den vorherigen 94% richtig erkannte. Daher wurden die Hyperparameter erneut validiert und angepasst. Abbildung 3.27b zeigt die Ergebnisse. Der Initiator wurde nicht erneut geprüft, da bis zu diesem Entwicklungsschritt nur Bildinstanzen des Objekts Würfel verwendet wurden.



(a) Genauigkeit der Klassifizierung nach verschiedenen, angewandten Methoden. (b) Genauigkeiten im Vergleich ursprünglicher und erweiterter Datensatz.

Abbildung 3.27.: Grafische Darstellung der Einflüsse der verschiedenen Methoden auf das erzielte Ergebnis der Klassifizierungen. Quelle [186].

Im Vergleich zum ursprünglichen Testdatensatz führten die Systemanpassung und die Regularisierungstechniken zu einer Verschlechterung der Accuracy um 4,2% bzw. 3,5%. Der Leistungsanstieg, der durch die Hyperparameteroptimierung generiert wurde, wurde jedoch größer. Dadurch sind die Auswirkungen der Optimierung deutlich bedeutsamer als erwartet. Insgesamt führte die Erweiterung des Testdatensatzes zu einem linearen Anstieg der Accuracy über den Entwicklungsfortschritt.

Obwohl die Ergebnisse die positiven Auswirkungen der getroffenen Maßnahmen auf die Entwicklung des modularen Modells unterstreichen, zeigt sich auch, dass die verwendeten Testdaten einen Bias aufweisen, der die Ergebnisse beeinflussen kann. Um die tatsächliche Leistungsfähigkeit des Modells zu validieren, sollten deshalb weitere Daten erhoben werden. Der Bias der Testdaten kann dazu führen, dass bei der Verwendung von anderen Testdaten alternative Entscheidungen entlang des Entwicklungsprozesses getroffen worden wären.

Abbildung 3.28 zeigt den finalisierten Systemaufbau des modularen Modells. In der Darstellung hat der binäre Experte der Fehlerklasse Layer Shift für das Eingabebild die maximale Ausgabe. Das bedeutet, dass das Bild richtigerweise als Layer Shift klassifiziert wird.

### 3. In-situ Fehlererkennung im Druckprozess

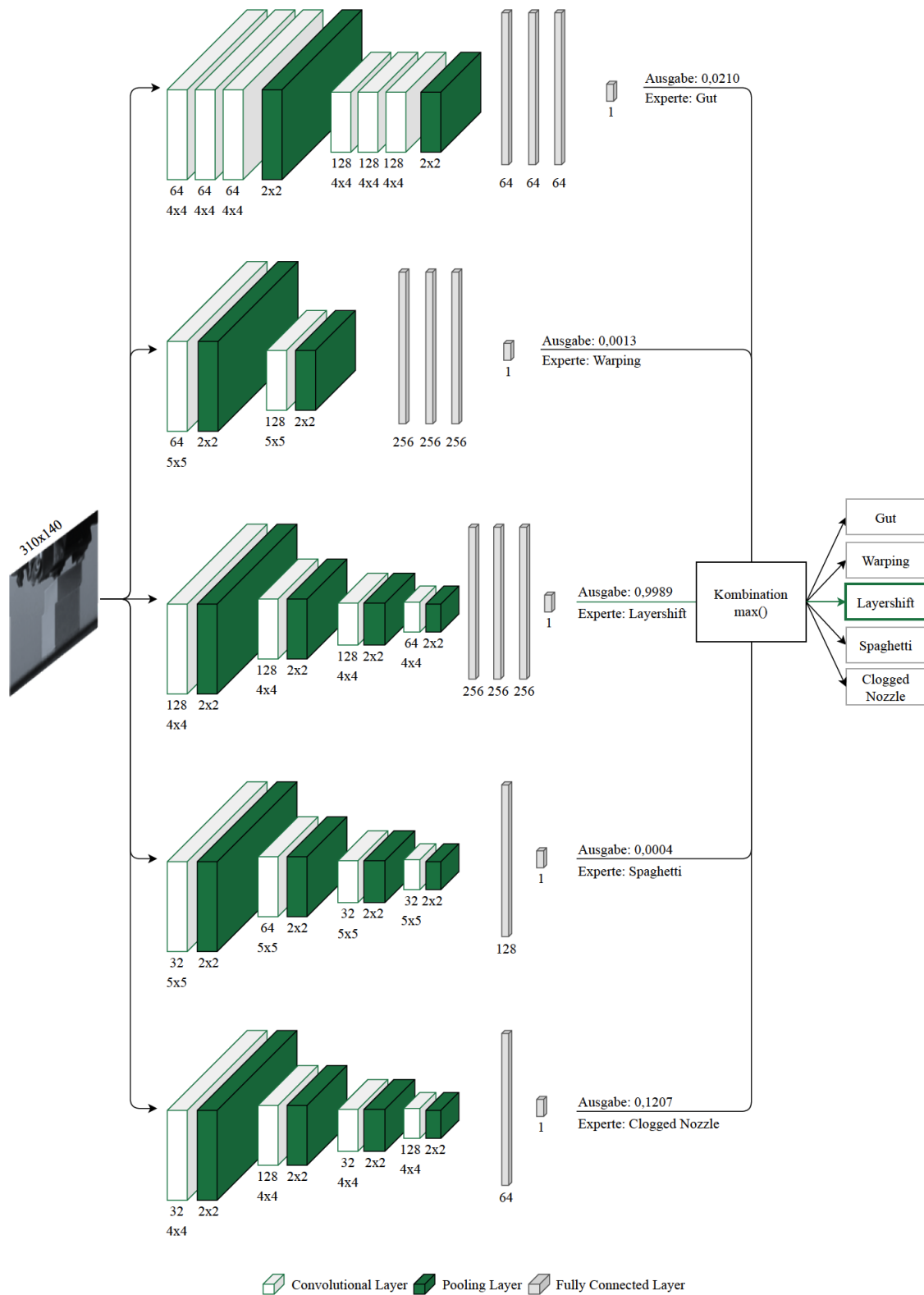


Abbildung 3.28.: Gesamtübersicht über alle fünf Modelle. Beim aktuellen Eingabebild handelt es sich um den Fehler Layer Shift, der auch vom Modell passend klassifiziert wird. Alle anderen vier Modelle klassifizieren die Bildeingabe nicht als für sich relevant. Quelle [186].

### 3. In-situ Fehlererkennung im Druckprozess

Im Folgenden wird der modulare Ansatz und die entwickelte Architektur evaluiert. Um die Leistungsfähigkeit des Modells zu bewerten, wird zunächst die Fehlerklassifikation auf einem neuen Testdatensatz untersucht. Hierbei werden neben der Accuracy und der FNR auch der  $F_\beta$ -Score als Makro-Wert und die Größe des Netzes sowie der Rechenaufwand des Trainings berücksichtigt. Danach wird die Generalisierungsfähigkeit des Modells in zwei Stufen bewertet. Dazu werden Bilder von bisher unbekanntem Objekten unterschiedlicher Ähnlichkeiten klassifiziert und die Robustheit des Modells gegen Veränderungen des Versuchsaufbaus untersucht. Abschließend wird geprüft, ob die Ansätze für eine Anwendung in Echtzeit geeignet sind und wie zeitnah das Modell Fehler klassifizieren kann. Die finale modulare Architektur wird mit Hilfe von 2,64 Millionen Parametern definiert was einem Speicherwaufwand von 31,5MB entspricht.

#### Fehlerklassifikation

Ein weiterer Testdatensatz, bestehend aus etwa 30 zusätzlichen Drucken, wird für die abschließende Evaluierung verwendet. Das Verhältnis von Trainingsdrucken zu Testdrucken beträgt etwa 90 zu 10. Mit dem neuen Testdatensatz sollen die Auswirkungen der bisher verwendeten Testdaten auf die Genauigkeit des Modells weiter untersucht werden. Gewählt wird  $\beta$ -Wert = 0,5, um den Recall entsprechend zu betonen. Das daraus resultierende Ergebnis ist in Tabelle 3.12 aufgeführt.

Tabelle 3.12.: Fehlerklassifikation mit erweitertem Datensatz.

	Accuracy	FNR	$F_\beta$
Modulare Architektur	97,5%	2,5%	0,977

Um zu verstehen, welche Fehler am häufigsten auftreten, wird die Konfusionsmatrix der Klassifikation zusätzlich zu den bisherigen Metriken ausgewertet. Tabelle 3.29 zeigt die prozentualen Verteilungen der Klassifikationen der verglichenen Modelle. Die Zeilen repräsentieren die tatsächliche Zuordnung der Klassen Gut (0), Warping (1), Layer Shift (2), Spaghetti (3) und Clogged Nozzle (4).

#### Generalisierungsfähigkeit

In der Regel wird die Generalisierungsfähigkeit eines Modells anhand der Aufteilung der Daten in Trainings-, Validierungs- und Testdaten beurteilt. Mit den Testdaten wird geprüft, wie gut das Modell unbekannte Daten klassifizieren kann. Abschließend werden ungesehene Objekte untersucht, um die Generalisierungsfähigkeit zu bewerten. Dies ist in Bezug auf das Anwendungsfeld der additiven Fertigung realistischer, da hier häufig individuelle Teile gedruckt werden, die dem Netz vorher nicht bekannt sind. Zusätzliche Druckobjekte, die sich in zwei Ähnlichkeitsstufen aufteilen lassen, werden verwendet. Während die erste Stufe mit den Grundobjekten vergleichbar ist, besteht die zweite Stufe aus komplexeren Objekten, die

### 3. In-situ Fehlererkennung im Druckprozess

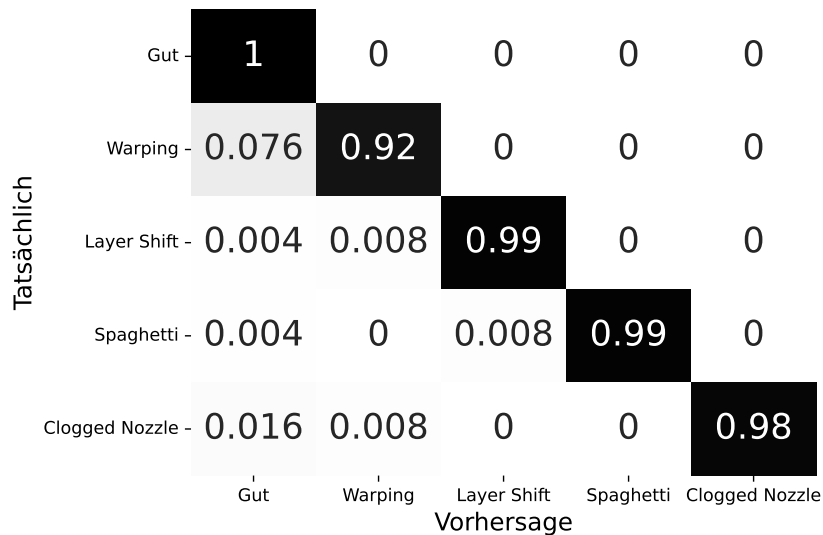


Abbildung 3.29.: Konfusionsmatrix für die Fehlerklassifikation. Hierbei wird deutlich, dass die Fehler deutlich voneinander unterschieden werden können und es nur zwischen den beiden Fehlerklassen Gut und Warping zu Verwechslungen kommt.

sich nur an den Grundkörpern orientieren. In Abbildung 3.30 sind die verwendeten Objekte dargestellt.

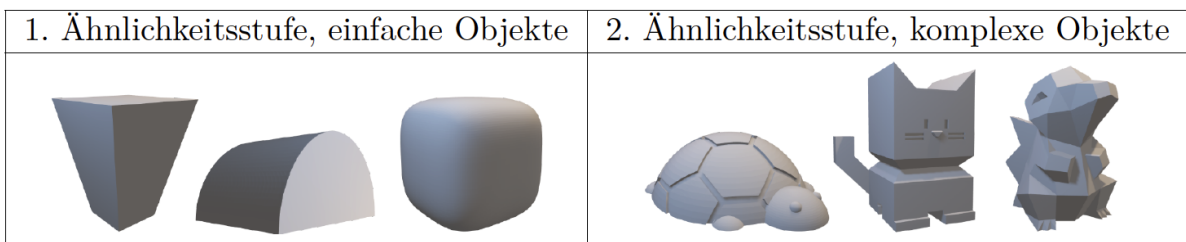


Abbildung 3.30.: Objekte in den beiden verwendeten Ähnlichkeitsstufen. Zur Evaluierung der Generalisierungsfähigkeit werden weitere Geometrien zu den Datensätzen hinzugefügt. Hierbei kommen sowohl weitere einfache Geometrien (Vase, Runddach, Gengon), aber auch erstmals komplexe Geometrien (Schildkröte, Calicat, Cyndaquil) zum Einsatz. Quelle [186].

Für jede Fehlerklasse wird mindestens ein Druck für jedes Objekt aufgenommen. Da die optischen Fehlermerkmale bei komplexeren Objekten manchmal schwer zu erkennen sind, wurden die Objekte je nach Fehlerklasse skaliert. Im ersten Schritt werden die Bildbeispiele von Vase, Runddach und Gengon mit verschiedenen Modellen klassifiziert. Die Ergebnisse sind als Konfusionsmatrix in Tabelle 3.31 aufgeführt.

In der nächsten Phase werden deutlich komplexere Objekte wie Schildkröte, Calicat und Cyndaquil untersucht. Diese Objekte bestehen aus den Grundobjekten, mit denen das Modell

### 3. In-situ Fehlererkennung im Druckprozess

Tatsächlich	Gut	0.71	0.21	0.024	0.016	0.04
	Warping	0.036	0.78	0.18	0	0
	Layer Shift	0	0.008	0.99	0	0
	Spaghetti	0.004	0	0	0.99	0.008
	Clogged Nozzle	0.0016	0.008	0	0	0.98
		Gut	Warping	Layer Shift	Spaghetti	Clogged Nozzle
		Vorhersage				

Abbildung 3.31.: Konfusionsmatrix für den modularen Ansatz bei Testung mit neuen, einfachen Objekten aus der 1. Ähnlichkeitsstufe, siehe Abbildung 3.30.

trainiert wurde. Während die Schildkröte auf dem Foto die optische Form einer Halbkugel hat, unterscheiden sich die anderen Objekte stark von den bekannten Objekten. Insbesondere die zusätzlichen Kanten können die Klassifizierung erschweren, insbesondere bei der Calicat und dem Cyndaquil. Die Ergebnisse in Tabelle 3.32 bestätigen dies. Das modulare System klassifiziert die Bildbeispiele mit 63,3%, und daher ergibt sich ein Unterschied von mehr als 30% im Vergleich zu den bekannten Grundobjekten.

Tabelle 3.13.: Generalisierungsfähigkeit

	Accuracy	FNR	F_beta
1. Ähnlichkeitsstufe	89,2%	1,4%	0,895
2. Ähnlichkeitsstufe	62,6%	0,1%	0,711

Das modulare Modell hat auf den Testdaten eine Genauigkeit von 97,5%. Die Evaluierung hat gezeigt, dass es vor allem bei der Klassifizierung der Klasse Gut Schwierigkeiten gibt. Mit dem modularen Ansatz kann die Konfiguration der binären Klassifikatoren verändert werden. So könnte der Klassifikator der Klasse Gut vor der eigentlichen Fehlerklassifikation ausgeführt werden. Dadurch müssten nur die fehlerhaften Bildbeispiele klassifiziert werden, was die Leistungsfähigkeit des Systems erhöhen würde, da es dann nur aus Experten der Fehlerklassen besteht. Außerdem kann bei einer modularen Aufteilung für die Klasse Gut weitere Informationen wie zum Beispiel ein gerendertes Bild der CAD-Daten separat eingegeben werden. Angenommen, dass der gewollte Versatz einer Treppenstufe dem optischen Merkmal des Layer Shifts nahezu identisch ist, erscheint dies ohnehin sinnvoll.

Die Analyse des Generalisierungsverhaltens auf unbekannte Objekte hat gezeigt, dass das modulare System mit einer Genauigkeit von 89,5% gut generalisiert. Es passt sich entscheidend

### 3. In-situ Fehlererkennung im Druckprozess

Tatsächlich	Gut	0.71	0.21	0.024	0.016	0.04
	Warping	0.036	0.78	0.18	0	0
	Layer Shift	0	0.008	0.99	0	0
	Spaghetti	0.004	0	0	0.99	0.008
	Clogged Nozzle	0.0016	0.008	0	0	0.98
		Gut	Warping	Layer Shift	Spaghetti	Clogged Nozzle
		Vorhersage				

Abbildung 3.32.: Konfusionsmatrix für die komplexen Modelle aus der 2. Ähnlichkeitsstufe.

an die optischen Merkmale der Fehlerklassen an und erkennt sie auch, wenn sich das betreffende Objekt verändert. Bei der zweiten Stufe werden jedoch die Grenzen des untersuchten Modells offensichtlich. Die komplexen Strukturen und Kanten scheinen das Modell zu verwirren, wodurch die Klassifikationsrate stark abnimmt. Ohne zusätzliche Informationen oder ohne dass das Modell die Objekte bereits kennen, ist somit keine ausreichende Vorhersagenauigkeit für eine Prozessüberwachung gewährleistet.



### 3.3. Akustische Fehlererkennung mit Hilfe von Deep Learning

Einige der Fehler die in der additiven Fertigung auftreten können, können optisch nicht oder kaum, dafür aber akustisch erfasst werden. Mit Hilfe von Audiosensoren und aktuellen Methoden der Datenverarbeitung und des maschinellen Lernens können diese Fehler erkannt und der Benutzer benachrichtigt werden, so dass er entsprechend reagieren kann. Kritische Fehler können auch sofort korrigiert werden, um Schäden oder Verschleiß am Drucker, Material- und Zeitverschwendung zu vermeiden.

In diesem Abschnitt wird eine Methode zur Erkennung von akustischen Ereignissen und Anomalien während eines Druckvorgangs mit einem 3D-Drucker mit Fused Filament Fabrication vorgestellt. Das neuronale Netzwerkmodell ist in der Lage, verschiedene Signale mit unterschiedlicher Schwierigkeit zu erkennen. Für Aufnahme von Audiodaten wird ein Mikrofon in der Nähe des Extruders des Druckers angebracht. Da es keine öffentlich zugänglichen Daten gibt, wurden Audio-Samples aufgezeichnet, annotiert und für eine bessere Verallgemeinerung des trainierten Modells mit verschiedenen Techniken erweitert. Hierfür kamen vor allem Pitching und Stretching zum Einsatz. Mel-Frequenz-Koeffizienten und Mel-Filterbank-Energien werden aus den aufgezeichneten und ergänzten Daten extrahiert und als Merkmale verwendet.

Zwei Aspekte sind wichtig, um aus den aufgenommenen Daten hochwertige Merkmale zu gewinnen. Zum einen ist die Auswahl eines geeigneten Mikrofons wichtig. Zum anderen bieten Audio-Aufnahmeprogramme unterschiedliche Formate, Abtastraten und Genauigkeiten an. Da nur die Geräusche am Drucker selbst, nicht aber die in der Umgebung relevant sind, wird ein Mikrofon mit Nierencharakteristik gewählt. Dies reduziert die irrelevanten Umgebungsgeräusche in den aufgenommenen Daten, zu denen vor allem menschliche Gespräche und Umgebungsgeräusche gehören. Um die besten Ergebnisse zu erhalten, sollte das Mikrofon daher auf den an einer Wand stehenden Drucker ausgerichtet werden und somit Geräusche von hinten nur sehr gedämpft wahrnehmen. Außerdem wurden die aufgenommenen Audiodaten im wav-Dateiformat gespeichert, das keine Kompression verwendet. Bei der Komprimierung könnten sonst wichtige Informationen aus dem Audiosignal verloren gehen.

In der additiven Fertigung gibt es eine ganze Reihe von möglichen Fehlern, die während eines Druckvorgangs auftreten können. Neben den bereits aus Kapitel 3.1 bekannten Fehlern Layer Shift und Extrusion Stop bzw. Clocked Nozzle, kommen hier noch zwei weitere dazu, die akustisch wahrnehmbar sind, optisch aber nur schwer zu erkennen sind. Eine akustische Erkennung der Fehler jedoch nur für einige wenige möglich, da z.B. ein zu geringer Filamentfluss oder eine falsche Temperatureinstellung, welche zu Warping führt, nicht hörbar sind. Die Auswahl der Fehler für diesen Ansatz, die akustisch erkennbar sind, basierte auf einer Liste möglicher Fehler der Firma Simplify3D [112]. Das Vorgehen zur Erstellung der Trainingsdaten ist in Abbildung 3.33 als Blockdiagramm dargestellt.

### 3. In-situ Fehlererkennung im Druckprozess

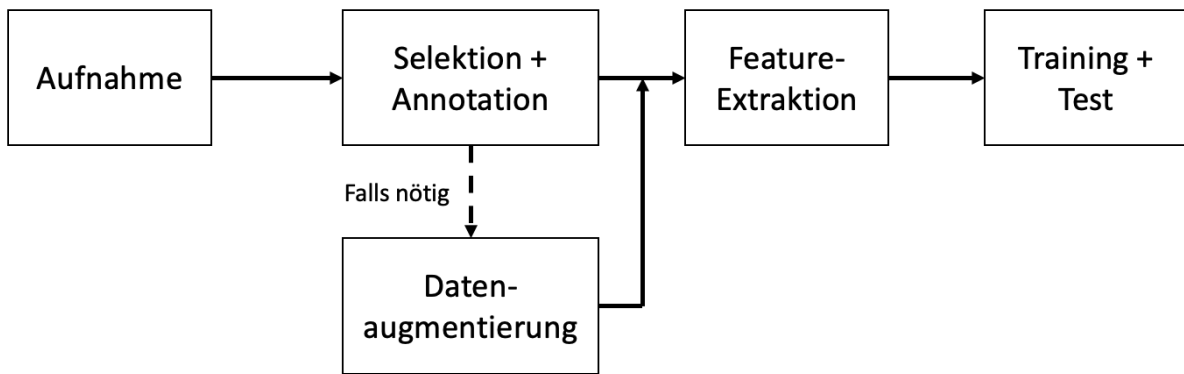


Abbildung 3.33.: Eingesetzte Machine Learning Pipeline zur Umsetzung der neuronalen Netze im Rahmen der akustischen Fehlererkennung. Quelle basierend auf [190].

**Falsche Extruderhöhe** Ein möglicher Fehler ist die falsche Kalibrierung der Düsenhöhe des Druckers. Wenn die Düse zu nah über das Objekt fährt, kommt sie damit in Berührung und hinterlässt Schleifspuren auf der Oberfläche. Dies führt dazu, dass die berechneten Höhen und der Materialfluss falsch sind, was dazu führen kann, dass das Objekt später falsche Abmessungen hat. Abbildung 3.34 zeigt einen Beispieldruck von oben. In der oberen Hälfte verläuft horizontal eine Spur aus geschmolzenem Kunststoff. Sie entstand, weil die zu tief kalibrierte heiße Düse über das Objekt kratzte.

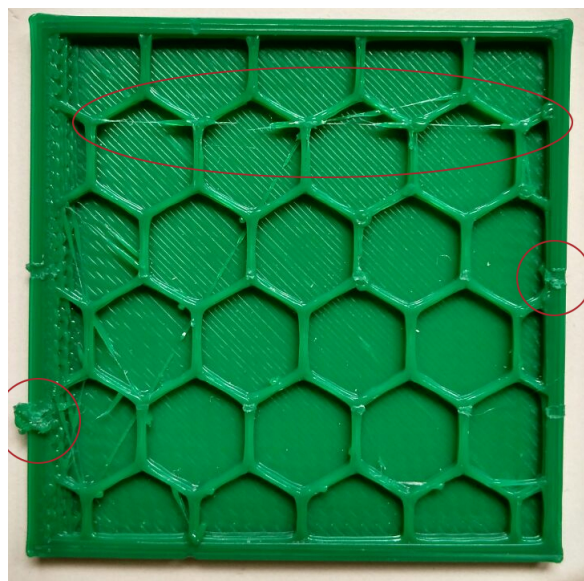


Abbildung 3.34.: Bild eines beschädigten Objekts, bei dem die Höhe der Düse während des Druckvorgangs nicht korrekt war und zu tief über das Objekt fuhr. Dadurch ergaben sich mehrere Unsauberkeiten, die akustisch detektierbar sind (rote Kreise). Quelle [190].

### 3. In-situ Fehlererkennung im Druckprozess

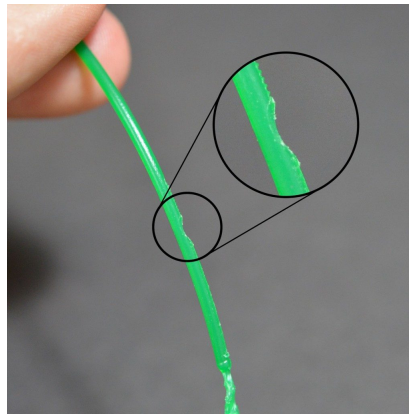


Abbildung 3.35.: Grinding Filament. Tritt entweder auf, wenn die Düse verstopft ist oder wenn der Durchmesser des Filaments falsch ist. Quelle [112].

**Grinding Filament** Während des Drucks wird das Filament durch scharfkantige Zahnräder zum Extruder geführt. Ist der Weg durch den Extruder für das Filament blockiert, drehen die Zahnräder dennoch weiter und reiben am Filament. Das bedeutet, dass immer mehr Filament abgerieben wird, bis das Filament und die Zahnräder nicht mehr ineinander greifen. Infolgedessen kann man beim Drehen der Schrittmotoren ein kratzendes Geräusch hören. Abbildung 3.35 zeigt den durch die vergrößerten Zahnräder verursachten Abrieb. Ein möglicher Grund für diesen Fehler kann eine verstopfte Düse oder eine falsche Temperatureinstellung sein.

#### Erkennung der Fehler

Die Erkennung kann zudem verwendet werden, um den aktuelle Zustand des Druckers erkennen. Hierfür werden zum Beispiel das Lüftergeräusch, das Druckgeräusch oder die Bewegungen der Z-Achse erfasst. Die beschriebenen Geräusche sind in Abbildung 3.37 dargestellt. Auch das Geräusch beim Öffnen und Schließen der Tür kann erkannt werden. Dies kann von entscheidender Bedeutung sein, wenn z. B. das aktuelle Objekt mit ABS gedruckt wird, das für ein gutes Druckergebnis eine stabile und hohe Temperatur benötigt. Wenn jemand die Tür während des Drucks öffnet und vergisst, sie zu schließen, könnte dies erkannt werden, um den Verantwortlichen zu warnen, damit der Druckprozess nicht durch instabile Temperaturen beeinträchtigt wird.

Alle trainierten Klassen sind in Tabelle 3.14 dargestellt.

#### 3.3.1. Datenaugmentation für erweiterten Datensatz

Die manuelle Erkennung von Fehlern in der Daten des 3D-Drucks ist recht mühsam, da die Fehler nur gelegentlich auftreten. Einige Fehler können manuell erzwungen werden, indem während dem Druckprozess eingegriffen wird. So können Fehler künstlich und zu bekannten Zeitpunkten erzeugt werden. Da dies aber für den Drucker schädlich und nicht vorgesehen ist,

### 3. In-situ Fehlererkennung im Druckprozess

Klassennummer	Klassenname
0	Lüftergeräusch
1	Druckvorgang
2	Tür öffnen
3	Schließen der Tür
4	Bewegung der Z-Achse
5	Falsche Düsenhöhe

Tabelle 3.14.: Verwendete Klassen für das Training des neuronalen Netzes.

wurde die Datenaugmentation genutzt, um einerseits mehr Daten zu erhalten, aber auch um ein allgemeineres Ergebnis mit dem neuronalen Netzmodell zu erzielen. Es wurden verschiedene Arten der Datenerweiterung verwendet, wie z. B. Zeitdehnung, Tonhöhenverschiebung oder -verstärkung.

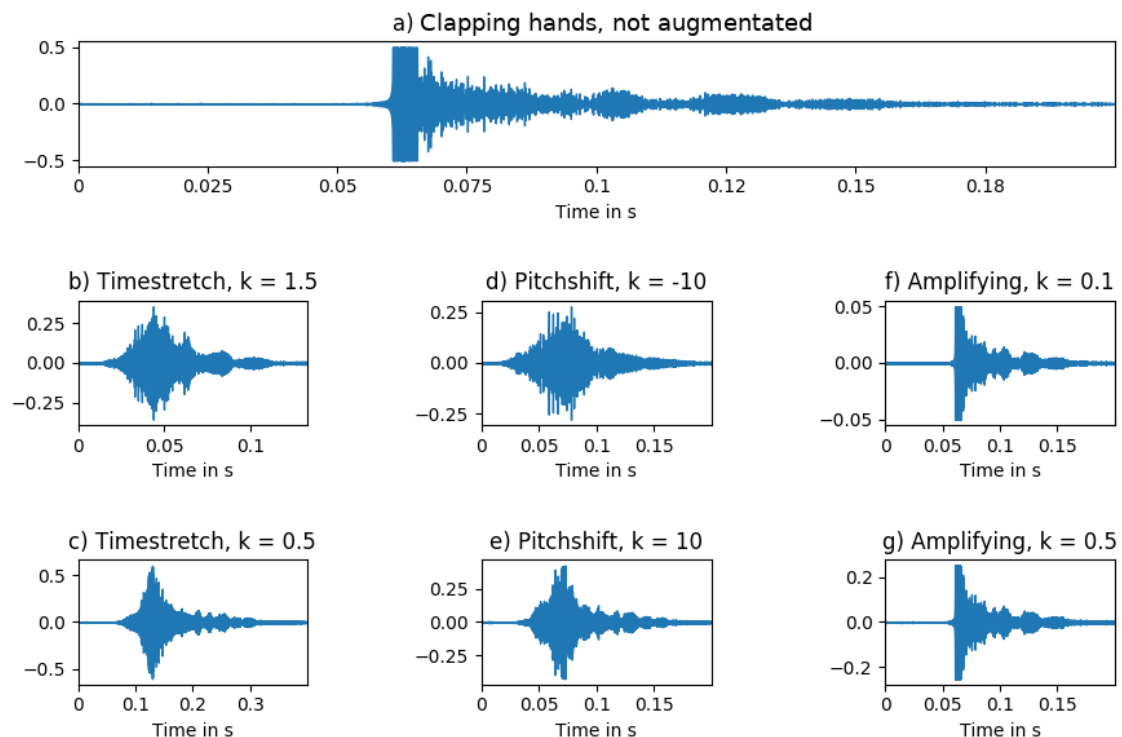


Abbildung 3.36.: Verschiedene Methoden der Datenaugmentation zur Erhöhung der Varianz der aufgezeichneten Daten. Drei Methoden wurden auf die Originaldaten angewandt: Zeitdehnung (Timestretch), Tonhöhenverschiebung (Pitchshift) und Verstärkung (Amplifying). Quelle [190].

Bei der zeitlichen Streckung wird das Signal in Zeitrichtung gestreckt oder gestaucht, während die Tonhöhe gleich bleibt. Abbildung 3.36a zeigt das ursprüngliche Audiosignal. Wendet man auf dieses Signal eine Dehnung mit verschiedenen Faktoren  $k$  an, wird es gestaucht (siehe Abbildung 3.36b) oder gedehnt (siehe Abbildung 3.36c).

### 3. In-situ Fehlererkennung im Druckprozess

Die Tonhöhe der einzelnen Samples wird um eine bestimmte Anzahl  $k$  Halbtöne nach oben oder unten verschoben. Abbildung 3.36d zeigt eine Verschiebung um 10 Halbtöne nach unten, Abbildung 3.36e zeigt eine Verschiebung um 10 Halbtöne nach oben.

Die Verstärkung stellt eine einfache Methode der Vergrößerung dar. Dazu wird jede Probe  $x_i$  mit einem Faktor  $a > 0$  multipliziert, wie es die Gleichung 3.2 beschreibt.

$$x_i = a * x_i, a \in \mathbb{R}_{>0} \quad (3.2)$$

Ist  $a > 1$ , wird das Signal verstärkt, ist  $1 > a > 0$ , wird das Signal abgeschwächt.

In Abbildung 3.36 sind zwei Augmentierungen eines Signals, dessen Original in Abbildung 3.36a dargestellt ist, mit den Faktoren  $k = 0, 1$  in Abbildung 3.36f und  $k = 0, 5$  in Abbildung 3.36g dargestellt.

Außerdem wurden die aufgezeichneten Audiodaten in überlappende Blöcke unterteilt, um sicherzustellen, dass ein komplettes Ereignis in mindestens einem Block enthalten ist, der als Eingabe für das neuronale Netz verwendet wird.

Bei diesem Ansatz wurde ein neuronales Netzwerkmodell mit Long Short-Term Memory verwendet, um die sechs verschiedenen Klangklassen zu lernen. Sie sind in Tabelle 3.14 aufgeführt. Nicht alle dieser Geräusche sind direkte Fehler, aber sie sind entscheidend für den Druckprozess, wie z. B. Bewegungen der Z-Achse oder das Geräusch des Druckerlüfters. Die Erkennung dieser Ereignisse hilft auch sicherzustellen, dass alles nach wie vor einwandfrei funktioniert. Einen Eindruck von den verschiedenen Geräuschsignalen gibt die Abbildung 3.37.

Tensorflow und Keras wurden verwendet, um das Netzwerk einzurichten und eine einfache Schnittstelle für das Lernen bereitzustellen. Das LSTM hat ein Layer mit 10 Nodes mit einem Dropout von 0,2. Darauf folgt ein Flatten-Layer, die die zweidimensionale Ausgabe der ersten Schicht auf eine eindimensionale abbildet. Die Ausgabe dieser Schicht wird wiederum von einer Ausgabeschicht aufgenommen, die durch Softmax aktiviert wird.

Aus den augmentierten und nicht-augmentierten Sample-Blöcken wurden die Mel-Frequenz-Cepstral-Koeffizienten (MFCC) extrahiert und die Maximalwerte der einzelnen MFCC-Frames bestimmt. Als weiteres Merkmal wurden die Mel-Filterbänke extrahiert und normalisiert. Diese Merkmale dienten als Eingangsvektoren für das LSTM-Modell. Das Training wurde mit der kategorialen Kreuzentropie als Loss-Funktion über 25 Epochen durchgeführt, wie in Abbildung 3.38 dargestellt.

### 3. In-situ Fehlererkennung im Druckprozess

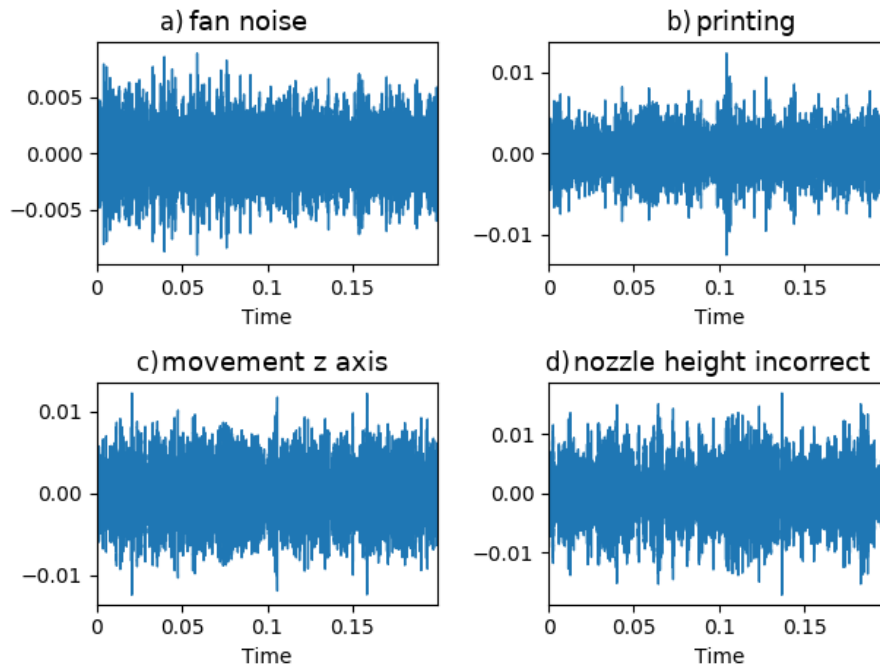


Abbildung 3.37.: Audiosignale im gegebenen Zeitfenster für mehrere Ereignisse wie das Lüftergeräusch (fan noise) (a), das Druckgeräusch (printing) (b), die Bewegung der Z-Achse (movement z-axis) (c) oder die falsche Extruderhöhe (nozzle height incorrect) (d). Quelle [190].

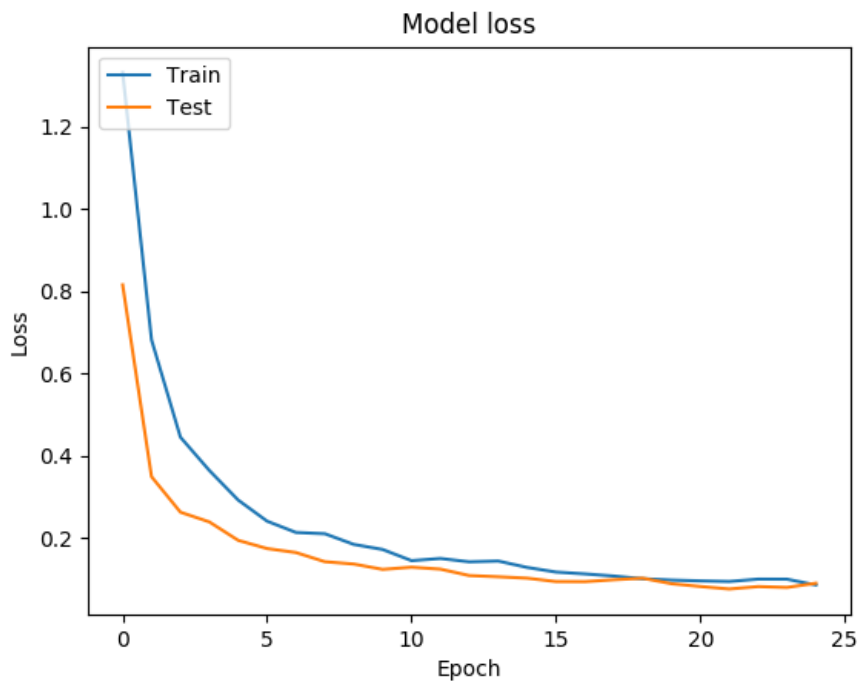


Abbildung 3.38.: Verlust von Training und Validierung über 25 Epochen. Quelle [190].

### 3.3.2. Experimente und Ergebnisse

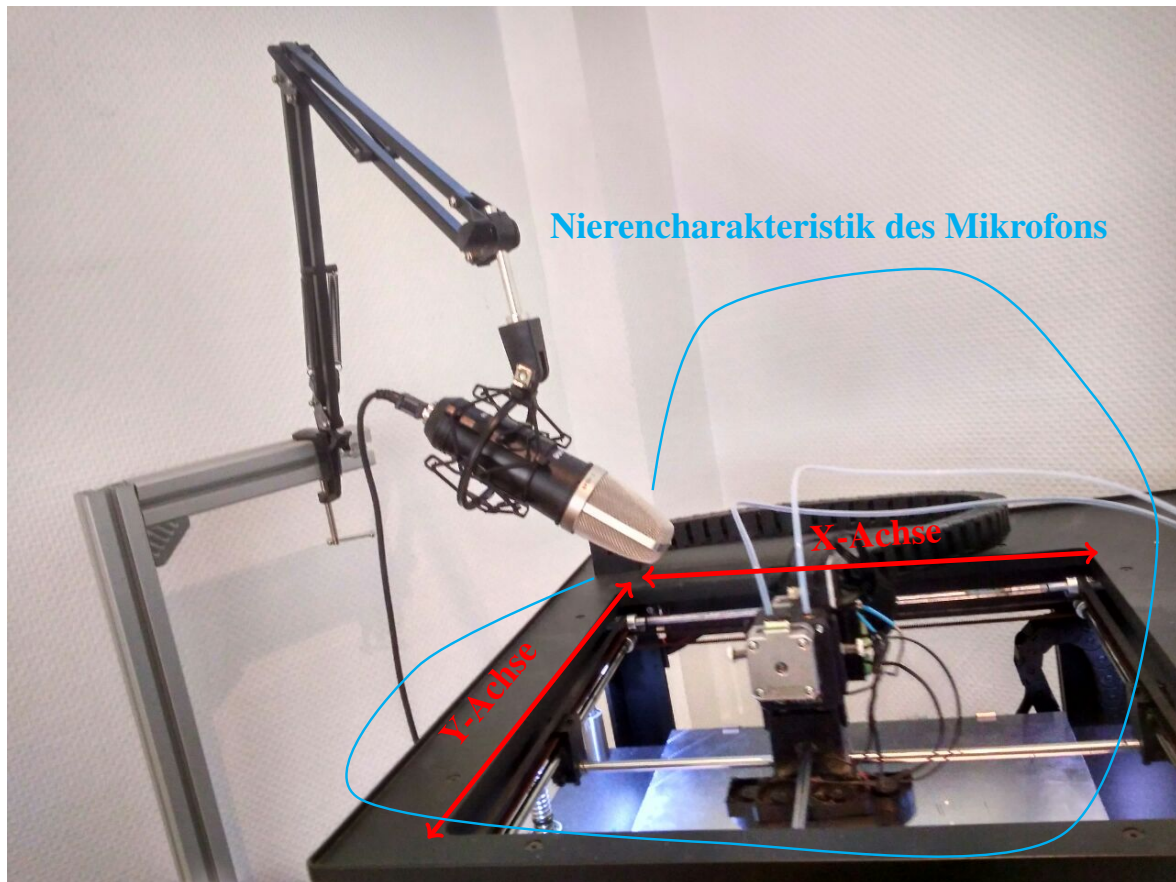


Abbildung 3.39.: Versuchsaufbau Druckbett: Das Mikrofon befindet sich während des Druckvorgangs in der Nähe des Extruders. Aufgrund der Nierencharakteristik des Mikrofons werden die meisten Umgebungsgeräusche nicht aufgenommen und der Fokus liegt auf dem Drucker selbst. Quelle [190].

Um den vorgestellten Ansatz zu validieren, wurde das Mikrofon wie in Abbildung 3.39 gezeigt am Drucker befestigt. Verwendet wurde ein t.bone SC450 Mikrofon mit Nierencharakteristik, um die Aufnahme von Umgebungsgeräuschen zu verhindern. Das Spektrum des Mikrofons ist für die Aufnahme von Ton und Sprache ausgelegt. In den Experimenten geht es vor allem darum, Geräusche im menschlichen Hörspektrum zu erfassen, daher eignet sich dieser Typ für die Versuche.

Es wurden mehrere Datensätze aufgenommen und ergänzt, um die Datenmenge zu erhöhen und eine Generalisierung für das neuronale Netz zu erhalten. Die Datensätze für das Lernen und für die Tests waren nicht identisch, um eine Verzerrung zu vermeiden.

Die zum Vergleich der Experimente verwendete Vorhersagemetrik ist der F1-Score. Der F1-Score ist definiert als

### 3. In-situ Fehlererkennung im Druckprozess

$$F1\text{-Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (3.3)$$

und ist eine der am weitesten verbreiteten Metriken im Bereich des maschinellen Lernens für die Mehrklassen-Klassifikation [131]. Für die Vorhersage von ungesehenen Daten ergibt das Experiment eine Genauigkeit von 86,25% und einen F1-Score von 67,08%. Abbildung 3.40 zeigt die Konfusionsmatrix der Vorhersage.

Tatsächlich	Lüfter -	119	0	0	0	0	7
	Druckprozess -	14	150	1	0	0	0
	Tür öffnen -	0	0	2	2	0	0
	Tür schließen -	1	0	1	10	0	0
	Bewegung z-Achse -	0	4	2	0	38	0
	Düsenhöhe nicht korrekt -	19	0	0	0	0	1
		Lüfter -	Druckprozess -	Tür öffnen -	Tür schließen -	Bewegung z-Achse -	Düsenhöhe nicht korrekt -
		Vorhersage					

Abbildung 3.40.: Konfusionsmatrix des LSTM auf ungesehene Daten.

Für den Klassifikator ist es schwierig, zwischen den Stichproben der Klassen Lüftergeräusch und falsche Extruderhöhe zu unterscheiden. In Abbildung 3.41 werden die Mel-Spektrogramme der Klassen Lüftergeräusch, Druck, Bewegung Z-Achse und Extruderhöhe nicht korrekt dargestellt, da die Mel-Bänke Ähnlichkeiten aufweisen. Da die Aufnahme der Trainingsdaten für diesen Fehler nicht während des Drucks erzeugt wurde wurde die Z-Achse kurzzeitig nicht bewegt ( $< 0, 1$  s). Daher kommt es zwischen dem Erkennen vom Lüftergeräusch und der falschen Extruderhöhe zu Verwechslung, siehe Konfusionsmatrix in Abbildung 3.40.

In Abbildung 3.42 sind zur genaueren Unterscheidung Proben der Klasse Lüftergeräusch und falsche Extruderhöhe nebeneinander dargestellt. Im hochfrequenten Bereich sind fast keine großen Unterschiede zu erkennen. Betrachtet man den niederfrequenten Bereich der grün dargestellten Spektrogramme, so gibt es ebenfalls keine großen Unterschiede, sondern nur geringe. Da auch die Klasse der Lüftergeräusche viele variantenreiche Samples aufweist, ist es für das Modell schwierig, zwischen diesen Merkmalen zu differenzieren.



### 3. In-situ Fehlererkennung im Druckprozess

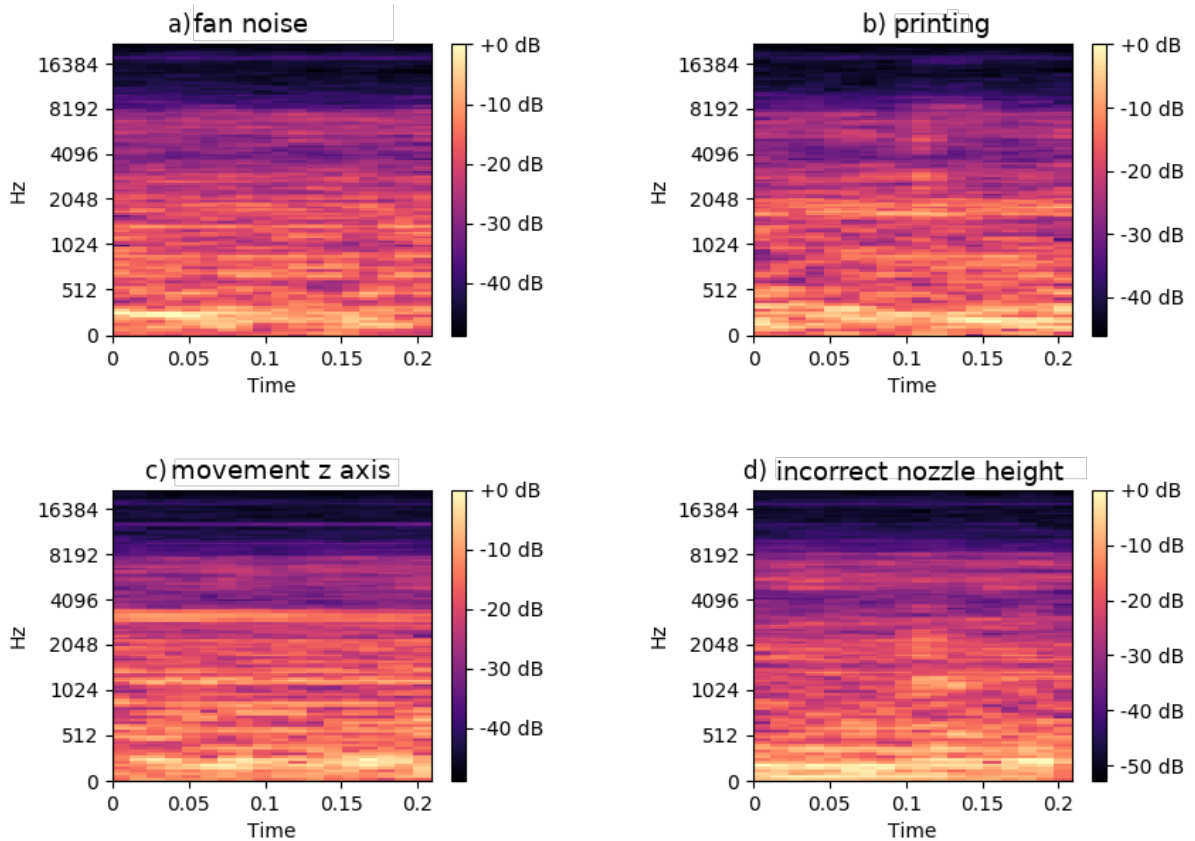


Abbildung 3.41.: Mel-Spektrogramme von Beispielen der Fehlerklassen Lüftergeräusch, Druck, Bewegung Z-Achse und falsche Extruderhöhe. Quelle [190].

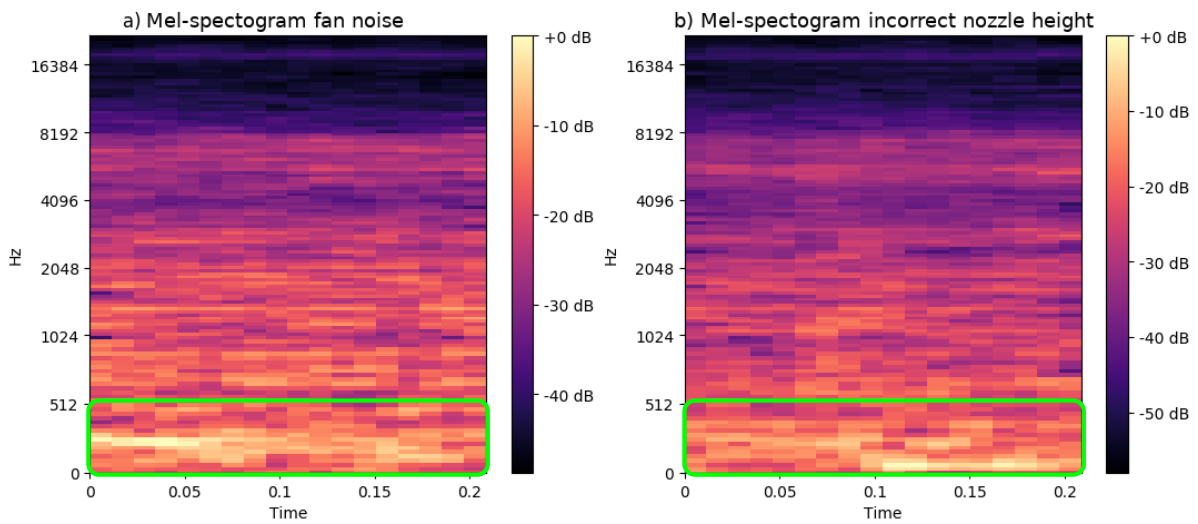


Abbildung 3.42.: Mel-Spektrogramme von Beispielen der Fehlerklassen Lüftergeräusch und falsche Extruderhöhe. Die Energien im Hochfrequenzbereich scheinen fast gleich zu sein, aber in den Energien im Niederfrequenzbereich sind kleine Unterschiede zu erkennen (vgl. grüne Kästen). Quelle [190].

### 3.3.3. Transfer des Verfahrens auf den PBF Druck

Das entwickelte Verfahren wird in diesem Abschnitt auf eine komplett andere Drucktechnologie angewandt, um die Transferierbarkeit zu zeigen.

Fehler im Powder Bed Fusion Druckverfahren können ebenfalls akustisch detektiert werden. Durch falsche Prozessparameter kann es zum Beispiel zu besonders porösen Bauteilen kommen, da es beim Schmelzen der Partikel zu Explosionen kommt und dadurch entstehen Luftlöcher, die sich letztendlich als Porosität im gedruckten Objekt widerspiegeln. Da die Porosität einen starken Einfluss auf die Bauteilstabilität hat, darf es hier keine Abweichungen zur eigentlich angedachten Materialverteilung geben. Allerdings sind diese Explosionen nicht im normalen menschlichen Hörspektrum, sondern befinden sich in deutlich höheren Frequenzen. So ist es notwendig, dass hier ein piezoelektrischer Sensor zum Einsatz kommt.

Neben dem Nachteil, dass PBF-Drucker eine höhere Leistung als andere AM-Verfahren benötigen, fehlt es den auf dem Markt befindlichen PBF-Druckern immer noch an einer zuverlässigen Prozessüberwachung, was dazu führt, dass viele Teile nicht verwendet werden können [132]. Zu den bei PBF auftretenden Defekten gehören Balling, Porosität, Rissbildung, geometrische Defekte und Sputter [133]. Diese haben einen erheblichen Einfluss auf die mechanischen Eigenschaften und damit auf die Qualität des zu verwendenden Teils [134]. Daher ist es notwendig, diese Defekte frühzeitig im Druckprozess zu erkennen, um den Aufwand für die Prüfung der Teile nach dem Druck zu reduzieren und sogar deren Weiterverarbeitung oder Verwendung zu vermeiden [94]. Da eine geringe Porosität entscheidend für ein qualitativ hochwertiges Druckergebnis ist, konzentriert sich der Transfer des Ansatzes ausschließlich auf die Erkennung der Porosität im Druckprozess.

Um während dem Druckprozess die Audiodaten aufzuzeichnen, konstruierte Niclas Eschner einen eigenen Druckeraufbau. Dieser ermöglichte ihm, die Vibrationen der Druckplatte zu messen, die während dem Druckprozess entstehen und sich von der Druckoberfläche durch das Objekt auf die Druckplatte auswirken. In mehreren Experimenten wurde das zu druckende Objekt um verschiedene Geometrien ergänzt, um die Auswirkung der Objektgeometrie auf die gemessenen Vibrationen zu untersuchen [94, 135]. Mit den aufgenommenen Daten aus den Druckvorgängen wurde ein neuronales Netz trainiert, das die Porosität an Hand der gemessenen Daten bestimmen konnte. Die in der Arbeit erstellten Daten wurden zudem veröffentlicht [136] und dieser Datensatz wird im folgenden verwendet, um den im vorherigen Kapitel 3.3 vorgestellten Ansatz vom FFF Druckverfahren auf das L-PBF Verfahren zu transferieren.

Insbesondere hat die Porosität als Defekt gravierende Auswirkungen auf das Bauteil durch Reiß- und Ermüdungsverhalten [133, 137]. Um den Grad der Porosität frühzeitig zu erkennen, wurde ein 2D Convolutional Neural Network auf in-situ Audioaufnahmen trainiert. Im Rahmen einer einfach zu bedienenden, anpassbaren Pipeline wurden Mel-Spektrogramme erzeugt und in das neuronale Netz zur Klassifizierung des Porositätsgrades eingespeist. Ein F1-Score von 98,5% zeigt, dass das Konzept die Porositätsdefekte von gedruckten Teilen tatsächlich durch neuronale Netze, die mit Audiospektrogrammen trainiert werden, in der Produktion effektiv erkannt werden können. Die Porosität kann somit direkt während des Druckprozesses abgeleitet

### 3. In-situ Fehlererkennung im Druckprozess

werden, was Kosten und Material spart, da ein poröser Druck frühzeitig gestoppt werden kann und eine weitere Untersuchung, wie z.B. durch Röntgen nach dem Druck, nicht mehr notwendig ist. Dieser Ansatz zeigt, dass die Integration von Sensoren in den Druckprozess einen großen Nutzen für die additive Fertigung in der Produktion bringen kann.

Der Originaldatensatz besteht aus den Aufnahmen des piezoelektrischen Sensors, während einzelne Objektschichten gedruckt wurden. Jedes Objekt besteht jeweils aus 125 Schichten. Jede Schicht entspricht einer Probe. Dies ergibt eine Gesamtzahl von 6.750 Proben, die insgesamt 90 GB an Rohdaten entsprechen. Die für die Datenerfassung variierten Parameter sind Laserleistung ( $W$ ,  $\in \{80, 130, 180\}$ ), Scangeschwindigkeit ( $mm/s$ ,  $\in \{200, 400, 1000\}$ ) und Spurbstand ( $\mu m$ ,  $\in \{40, 50\}$ ). Die Konfigurationen sind in Abbildung 3.43 dargestellt.

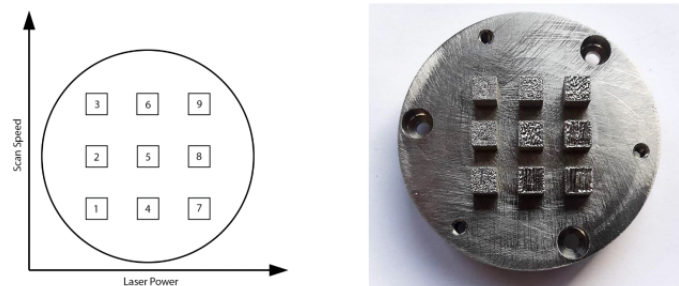


Abbildung 3.43.: Mehrere Testobjekte in Würfelform mit unterschiedlichen Druckeinstellungen wurden gedruckt, um die für das CNN benötigten Daten zu erzeugen. Quelle [138].

Jedes Objekt wird auf eine Komplexitätsklasse abgebildet, die der Klasse entspricht, die später für das maschinelle Lernen verwendet wird. Die Klassen sind gleichmäßig verteilt, wobei jede Klasse 2250 Proben umfasst. Abbildung 3.44 veranschaulicht die Komplexitätsklassen.

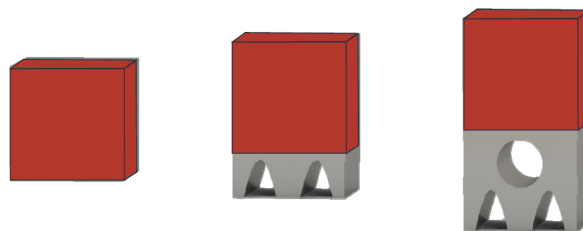


Abbildung 3.44.: Komplexitätsklassen der gedruckten Objekte. Von links nach rechts nimmt der Anteil der porösen Struktur zu, beginnend mit Klasse 1 ohne jegliche Porosität bis hin zu Klasse 3 mit einer hohen Wahrscheinlichkeit von Porosität. Der graue Teil des Bauteils ist die zusätzliche Struktur die zum Würfel hinzugefügt wurde, um die Auswirkung komplexer Geometrien zu untersuchen. Quelle [94].

Je höher die Klasse, desto höher ist die Wahrscheinlichkeit, dass das Objekt eine höhere Porosität aufweist. Das Ziel eines möglichen Klassifikators wäre es, eine korrekte Verbindung

### 3. In-situ Fehlererkennung im Druckprozess

zwischen dem akustischen Signal und der Porosität eines Objekts herzustellen. Daher ist es wichtig, aussagekräftige Merkmale zu wählen. In der einfachsten Form kann dies das akustische Rohsignal sein, aber auch stärker verarbeitete und ausgefeilte Merkmale wie Spektrogramme sind möglich. Darüber hinaus müssen die Merkmale auch entsprechend vorverarbeitet werden. Schließlich müssen die Eingabedaten zur Struktur des neuronalen Netzes kompatibel sein. Ein 2D-CNN kann zum Beispiel keine eindimensionalen Rohdaten verarbeiten. Selbst bei passenden Eingabedaten und Trainingsparametern kann der Trainingsprozess mehrere Tage in Anspruch nehmen, z. B. im Falle einer hohen Abtastrate in den Spektrogrammen. Ziel ist es hier, Einstellungen zu finden, die wenig Speicherplatz benötigen und zeitgleich ein schnelles Training, sowie eine hohe Klassifizierungsgenauigkeit zu ermöglichen.

Der hier vorgestellte Ansatz wird in zwei Phasen unterteilt: die Vorverarbeitungs- und die Trainingsphase. Die Vorverarbeitungsphase hat den Zweck, die einzelnen Schichten von Float-Zeitreihen zu Bildern vorzuverarbeiten und dadurch den späteren Lernprozess des Netzes zu beschleunigen. Die Beschleunigung ergibt sich daraus, dass die Vorverarbeitung nur einmal durchgeführt und dann auf der Festplatte gespeichert wird, im Vergleich zur Durchführung der Vorverarbeitung vor jedem einzelnen Training. In der Trainingsphase werden die Bilder aus dem Vorverarbeitungsordner geladen, in der Größe verändert und stapelweise in das neuronale Netz eingegeben. Die Vorverarbeitung wurde durchgeführt, um die Iterationsgeschwindigkeit beim Erstellen und Trainieren des Netzes zu erhöhen. Beide Phasen sind in der Abbildung 3.45 als eine gemeinsame Pipeline dargestellt.

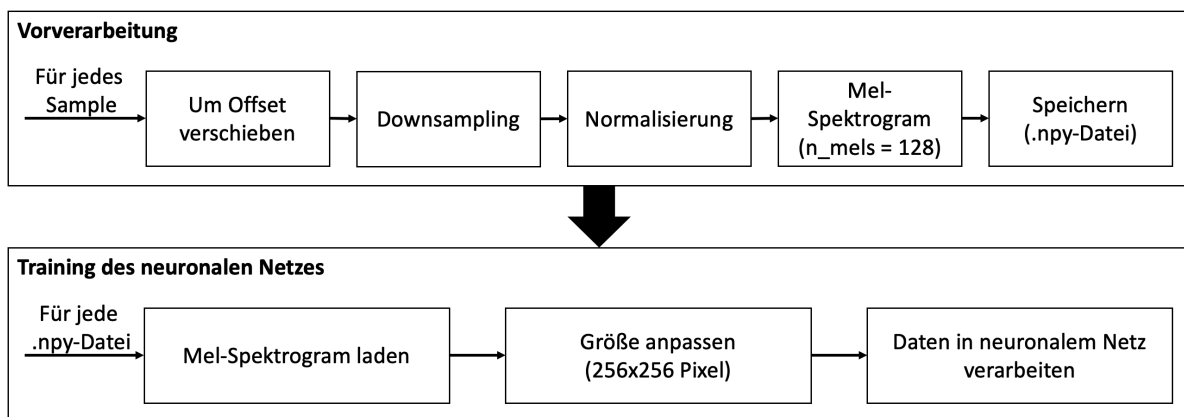


Abbildung 3.45.: Übersicht über die einzelnen Verarbeitungsschritte: Von den akustisch aufgezeichneten Daten jeder Schicht wird ein Offset subtrahiert, das Downsampling und die Normalisierung durchgeführt. Nach der Umwandlung der Daten in Mel-Spektrogramme werden sie als .npy gespeichert. Für die Trainingsphase werden die .npy erneut geladen, in der Größe verändert und dem CNN zugeführt.

Beginnend mit der Vorverarbeitungsphase werden die Schichten nacheinander im Dateiformat .h5py geladen und in ein einziges Array geflattet. Von diesem Array wird ein Offset subtrahiert. Der Offset wird als globales Minimum über alle Schichten berechnet. Dies ist notwendig, da der Sensor automatisch einen Offset zu allen Datenpunkten hinzufügt. Der Offset

### 3. In-situ Fehlererkennung im Druckprozess

$O$ , der von jeder Probe abgezogen wird, beträgt  $O = 8.220.261$ . Um die Anzahl der Proben zu reduzieren, wurde ein Downsampling durchgeführt. Obwohl dies zunächst kontraintuitiv erscheinen mag, erwies es sich später als effektiv, da selbst bei einer niedrigen Abtastrate die Vorhersagemetriken hoch waren, einschließlich des zusätzlichen Vorteils einer viel kürzeren Trainings- und Vorverarbeitungsdauer. Die Experimente wurden mit den folgenden Abtastraten durchgeführt:  $F_s \in \{176.400 \text{ Hz}, 352.800 \text{ Hz}, 705.600 \text{ Hz}, 1.411.200 \text{ Hz}, 2.822.400 \text{ Hz}\}$ . Die Experimente wurden zunächst mit 44.100 Hz durchgeführt, um ein absolutes Minimum für die Downsampling-Rate festzulegen. Das Modell ließ sich mit dieser Rate nicht gut trainieren. Daher wurde diese Rate verdoppelt, bis die Ergebnisse angemessen waren. Anschließend wird eine Normalisierung durchgeführt, um die Werte in einen gemeinsamen Bereich zu bringen und sie um Null zu zentrieren. Danach wird entweder das Spektrogramm und das Mel-Spektrogramm mit Hilfe der *LibROSA* Bibliotheksfunktionen [139] erstellt.

Ein Beispiel aus dem Datensatz ist in Abbildung 3.46 dargestellt.

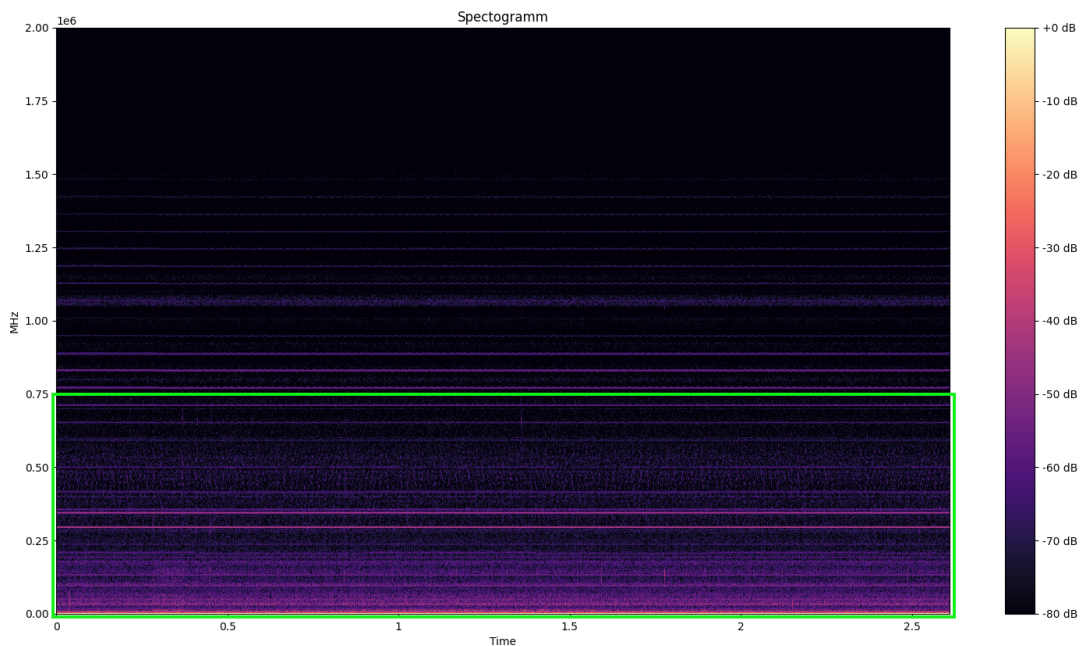


Abbildung 3.46.: Ein mit 4 MHz und  $n_{fft} = 2048$  abgetastetes Spektrogramm. Spektralenergien im unteren Frequenzbereich ( $< 500\text{kHz}$ , grün hervorgehoben) sind deutlich sichtbar. Quelle [190].

Die Skalierung der Mel-Skala ist dem Höreindruck des menschlichen Gehörs nachempfunden und wird daher bei vielen Herausforderungen eingesetzt, bei denen Anomalien oder Fehler tatsächlich für den Menschen hörbar sind. Zu diesen Herausforderungen gehören die Erkennung von Schallereignissen [140], die Erkennung von Anomalien [141] und Music Information Retrieval (MIR) [142]. Ein Beispiel für ein Mel-Spektrogramm aus dem Datensatz ist in Abbildung 3.47 zu sehen. Die X-Achse beschreibt die Zeit in Sekunden, die Y-Achse die logarithmisch skalierte Frequenz in Hz. Die Mels-Energie ist im unteren Spektrum recht konstant, im oberen Bereich sind spektrale Muster um 0,5 s zu erkennen.

### 3. In-situ Fehlererkennung im Druckprozess

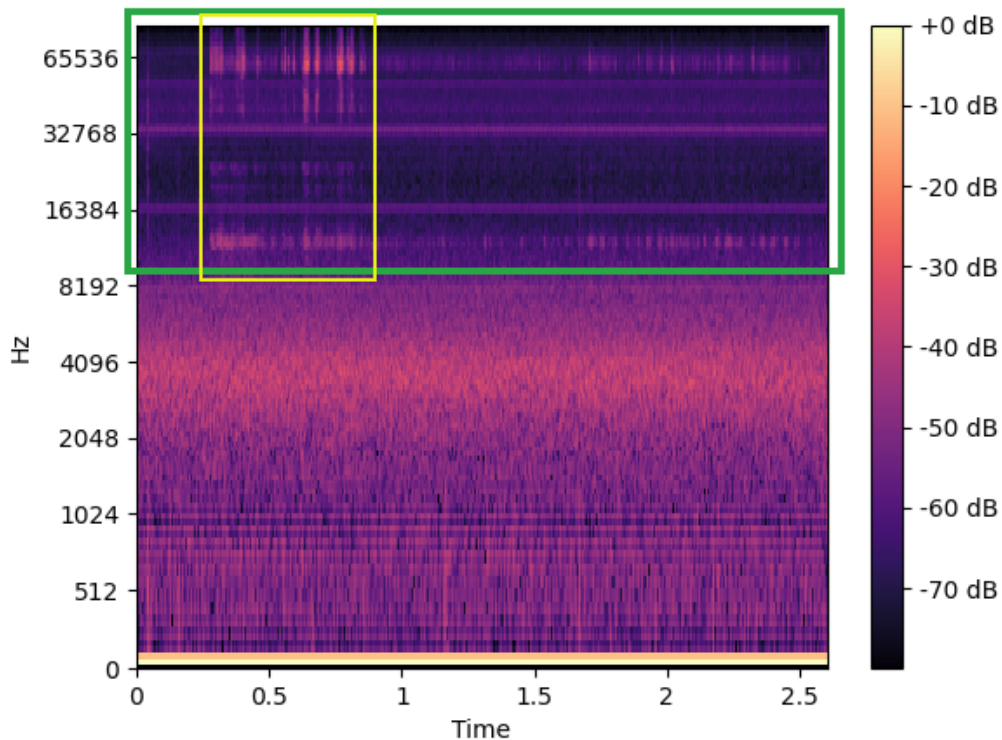


Abbildung 3.47.: Mel-Spektrogramm, abgetastet mit 176400 Hz, Achse ist logarithmisch skaliert. Gelb hervorgehobene Spektralmuster im oberen Frequenzbereich ( $> 10$  kHz, grün hervorgehobener Bereich). Der Rest ist größtenteils konstant. Quelle [190].

Beide Spektrogramme verwenden eine FFT-Fensterlänge von  $n_{fft} = 2.048$ . Für das Mel-Spektrogramm wird ein minimales Frequenzbin von  $f_{min} = 8.000$  Hz und  $n_{mels} = 128$  als Anzahl der Mel-Frequenzbins verwendet. Diese Spektrogramme werden schließlich als .numpy-Dateien für weitere Berechnungen gespeichert.

In der folgenden Trainingsphase wird zunächst eine Reihe von Schichten, die durch die Mel-Spektrogramme repräsentiert werden, durch Laden der entsprechenden .numpy-Dateien abgerufen. Durch die Verwendung von .numpy-Dateien wird der Ansatz beschleunigt, da die Dateien nur einmal vorverarbeitet werden müssen. Außerdem werden .numpy-Dateien schneller geladen als .h5py-Dateien. Bei der Iteration summiert sich diese Beschleunigung und spart mehrere Stunden an Rechenzeit. Die Spektrogramme werden dann entweder auf  $256 \times 256$  oder  $512 \times 512$  verkleinert, was als einheitliche Eingabegröße dient, bevor sie an das Netzwerk weitergeleitet werden. Mit einer größeren Eingabegröße können mehr Informationen übertragen werden, aber auch das Netzwerk muss gut genug sein, um die darin enthaltenen Informationen zu extrahieren. Aus diesem Grund wird in den Experimenten getestet, ob  $512 \times 512$  eine höhere Ergebnismetrik liefert als  $256 \times 256$ . Das stapelweise Laden erfolgt durch Python-Generatoren

### 3. In-situ Fehlererkennung im Druckprozess

mit dem Schlüsselwort `yield`, da nicht allzu viele Bilder gleichzeitig in den RAM passen. Die Generatoren holen iterativ nur die aktuell benötigten Bilder für den Stapel, füttern das Netzwerk mit ihnen und verwerfen sie anschließend aus dem RAM.

Für die Konstruktion und das Training der neuronalen Netze wird Tensorflow verwendet [143]. Für das Training der Modelle wird eine NVIDIA GeForce RTX 2080 Ti verwendet. Die Topologie des Convolutional Neural Network (CNN) ist in Abbildung 3.48 dargestellt.

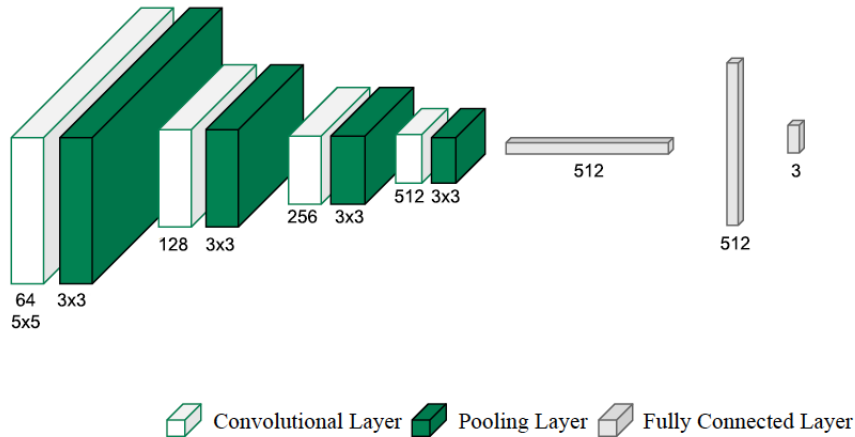


Abbildung 3.48.: Die Architektur des evolutionären neuronalen Netzes. Es hat insgesamt 1,8 Millionen trainierbare Parameter und besteht aus mehreren Pooling- und Faltungsschichten, die das Klassifizierungsergebnis verbessern sollen. Quelle [190].

Die einzelnen Netzschichten sind wie folgt aufgebaut: Conv2D  $\in \{64, 128, 256, 512\} \rightarrow$  MaxPooling  $\rightarrow$  Batch-Normalisierung  $\rightarrow$  Dropout  $\in \{0.2, 0.5\}$

Die Conv2D-Schichten verwenden die `relu`-Aktivierungsfunktionen (Rectified Linear Unit). In den letzten Klassifizierungsschichten wird eine Softmax-Funktion, Adam als Optimierer und kategoriale Kreuzentropie als Loss-Funktion verwendet. Das Modell verfügt über circa 1,8 Millionen Parameter. Jedes Training wird über 5 Epochen mit einer Stapelgröße von 8 iteriert. Um kostbare Zeit zu sparen und eine Überanpassung zu verhindern, wird Early Stopping mit einer Patience von zwei Epochen angewendet, was bedeutet, dass das Training gestoppt wird, sobald die Validierungsgenauigkeit der Modelle zwei Epochen lang nicht gestiegen ist. Um die Verzerrung zu verringern und das beste Modell zu erhalten, wird die  $k$ -fache Cross-Validation mit  $k = 5$  angewandt und es werden  $k$  Modelle erstellt. Der Wert  $k = 5$  wurde gewählt, weil er einen guten Mittelweg zwischen einem einzelnen Durchlauf und dem empfohlenen Wert  $k = 10$  darstellt, dessen Durchführung lange dauern würde. Tabelle 3.15 veranschaulicht, wie die Cross-Validation funktioniert. Bei diesem Ansatz wird jede Probe im Datensatz gegeneinander validiert.

Aus den berechneten Modellen wird das hinsichtlich der Validierungsgenauigkeit beste Modell für die Vorhersage ausgewählt.

### 3. In-situ Fehlererkennung im Druckprozess

Tabelle 3.15.: 5-fache Cross-Validation. Bei jeder Iteration werden 5 Foldings erstellt, von denen ein Fold als Validierungsset fungiert, die anderen vier gehören zum Trainingsset.

Iteration	Training set				Validation set
1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
2	Fold 5	Fold 1	Fold 2	Fold 3	Fold 4
3	Fold 4	Fold 5	Fold 1	Fold 2	Fold 3
4	Fold 3	Fold 4	Fold 5	Fold 1	Fold 2
5	Fold 2	Fold 3	Fold 4	Fold 5	Fold 1

Eschner *et al.* haben die jeweiligen Label entsprechend der Einschätzung der Porosität für jedes Objekt hinterlegt [94]. Das bedeutet, dass jede Schicht desselben Objekts das gleiche Klassenlabel hat. Abbildung 3.49 zeigt diese Zuordnung. Die Klassengrenzen liegen bei 1% und 6%. Die geschätzte Porosität wird nach Gleichung 3.4 berechnet, wobei  $\rho$  die Dichte,  $m$  die Masse und  $V$  das Volumen des Objekts ist.

$$\rho = \frac{m}{V} \quad (3.4)$$

#### 3.3.4. Experimente und Ergebnisse

Die Experimente wurden mit Spektrogrammen und Mel-Spektrogrammen als Bildtypen durchgeführt. Die Eingabegrößen waren  $256 \times 256$  und  $512 \times 512$ , um die Bilder auf eine einheitliche Größe zu bringen. Letzteres wird für den Fall getestet, dass durch das Downsampling auf  $256 \times 256$  zu viele Informationen verloren gehen. Vorausgesetzt, das Modell ist komplex genug, könnte es auch bessere Vorhersagemetriken als das erste Modell erreichen. Da ein Sensor verwendet wird, der Werte bis zu einer Frequenz von 4 MHz aufnehmen kann, ist zu erwarten, dass die Ergebnisse bei höheren Abtastraten besser sind, da sie mehr Informationen enthalten. Es wird auch erwartet, dass die Spektrogramme besser abschneiden als Mel-Spektrogramme, da die während des Druckvorgangs gehörten Geräusche weit außerhalb des akustischen Spektrums des Menschen liegen. Der Datensatz wurde in 10% Test-/Vorhersagedaten aufgeteilt, die restlichen 90% wurden zur Cross-Validation verwendet. Für jede unterschiedliche Abtastrate  $F_S$  wurde eine 5-fache Cross-Validation verwendet, wobei jedes CNN-Modell für 5 Epochen trainiert wurde.  $k = 5$  impliziert eine 80/20 Aufteilung von Training und Validierung. Der Mittelwert der Cross-Validation wurde gespeichert und als Metrik für die Stabilität des Trainingsprozesses und der Modelle verwendet. Aus jedem Fold wurde das Modell mit der besten Validierungsgenauigkeit ausgewählt, um anschließend eine Vorhersage für den Testdatensatz zu machen. Aus dieser Vorhersage wurden die entsprechenden Konfusionsmatrizen erstellt. Tabelle 3.19 zeigt die Konfusionsmatrix für  $F_S = 176.400$  Hz und eine Mel-Spektrogramm-Eingabe von  $256 \times 256$



### 3. In-situ Fehlererkennung im Druckprozess

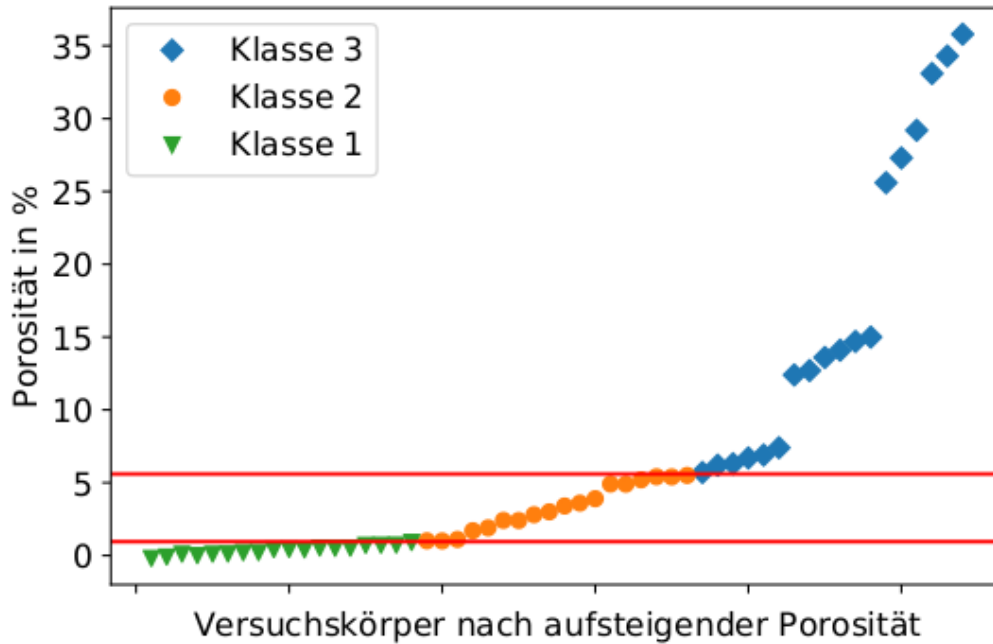


Abbildung 3.49.: Zuordnung der Porosität zur Klassenbezeichnung. Die geschätzte Porosität (in %) befindet sich auf der Y-Achse, die Objekte geordnet nach zunehmender Porosität auf der X-Achse. Quelle [94].

Punkten. Das Modell schneidet mit Ausnahme der Porositätsklasse 2 hervorragend ab. Dies war auch bei den Experimenten mit anderen Abtastraten und Eingabegrößen der Fall. Betrachtet man die Wahrscheinlichkeitsdichtefunktionen der jeweiligen Porositätsklassen in Abbildung 3.50, so ist zu erkennen, dass sich Klasse 2 hinsichtlich ihres Porositätsanteils mit den anderen Klassen überschneidet. Dies könnte es dem Modell erschweren, die richtige Porositätsklasse entsprechend der gegebenen Bodenwahrheitskennzeichnung eindeutig vorherzusagen.

Tabelle 3.16 zeigt die Ergebnisse der 5-fachen Cross-Validation nach einem Training von  $256 \times 256$  Mel-Spektrogrammen. Die Modelle mit  $F_S = 352.800$  Hz,  $705.600$  Hz haben eine hohe Validierungsgenauigkeit und einen guten Mittelwert, aber die Vorhersage kann nicht überzeugen. Die besten und stabilsten Modelle sind die übrigen mit  $F_S = 2.822.400$  Hz,  $1.411.200$  Hz,  $176.400$  Hz, da alle Metriken hoch sind.  $F_S = 176.400$  Hz wird als endgültiges Modell ausgewählt, da seine Vorhersage und sein Mittelwert nicht nur hoch sind, sondern vor allem das Training und die Vorhersage aufgrund der geringeren Abtastrate schneller waren. Die Möglichkeit, ein Modell schnell zu trainieren und einzusetzen, ist ein wichtiger Vorteil. Eine weitere interessante Beobachtung ist, dass die besten Validierungsgenauigkeiten nicht sehr unterschiedlich sind. Dies bedeutet, dass dank der Cross-Validation bei genügend Versuchen immer ein Modell mit der besten Leistung gefunden werden konnte.

Betrachtet man die Tabelle 3.17 für die Eingabegröße  $512 \times 512$  (bezeichnet als Experiment 2) und vergleicht sie mit der Tabelle 3.16 mit der Eingabegröße  $256 \times 256$  (Experiment 1),

### 3. In-situ Fehlererkennung im Druckprozess

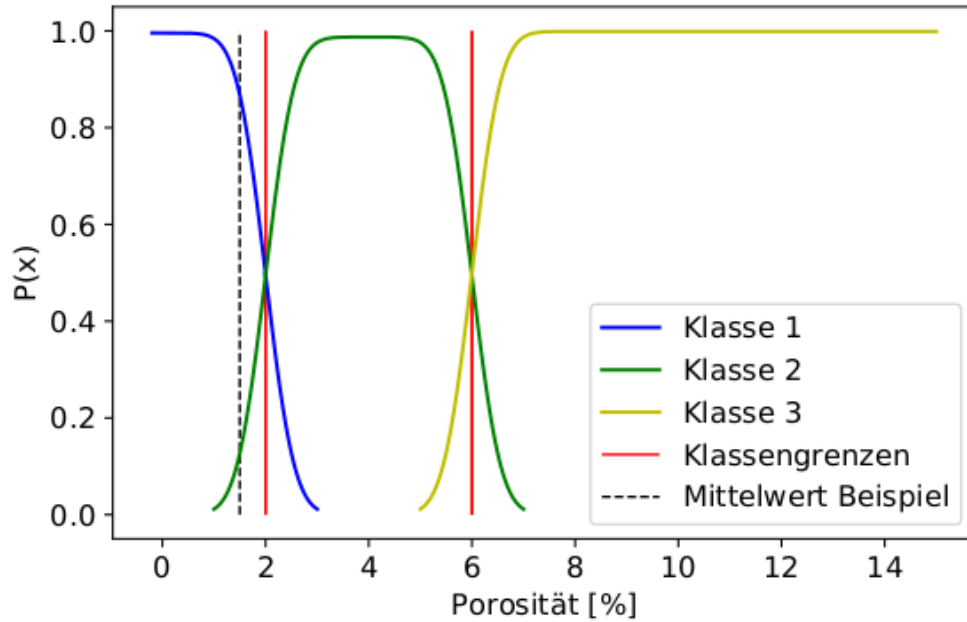


Abbildung 3.50.: Wahrscheinlichkeitsdichtefunktionen bezüglich der geschätzten Porositätsprozente. Die rot hervorgehobenen Linien stellen die strengen Klassengrenzen dar. Quelle [94].

lassen sich einige Beobachtungen machen. Die Mittelwerte der Cross-Validation unterscheiden sich stärker, ebenso wie die Validierungsgenauigkeiten. Keine Validierungsgenauigkeit ist so hoch wie die aus Experiment 1. Möglicherweise ist das Modell bereits mit den größeren Dimensionen überfordert und benötigt daher weitere, zusätzliche Filter. Auch die Mittelwerte unterscheiden sich stärker, um 10%. Aus den F1-Scores lässt sich keine Beobachtung oder Schlussfolgerung ziehen, außer dass das Modell mit einer Abtastrate von 176.400 Hz untauglich ist. Die Mel-Spektrogramme mit dieser Abtastrate hatten höchstens  $1152 \times 128 = 147.456$  Samples, im Vergleich zu  $512 \times 512 = 262.144$  Samples. Das bedeutet, dass ein großer Teil des Eingangsspektrogramms leer ist.

Um die Stabilität des Modells weiter zu betonen, werden in jeder Falte ein oder zwei Modelausreißer gefunden. Diese haben eine Validierungsgenauigkeit, die mehr als 20% vom besten Modell abweicht. Da die Cross-Validation nur fünf Faltungen umfasst, bedeutet dies, dass die Modelle nicht sehr stabil sind. Tabelle 3.18 veranschaulicht dies.

Um dieses Problem zu lösen, können die Modelle durch Datenerweiterung robuster gemacht werden. Tonhöhenverschiebung, Zeitdehnung und Verstärkung sind gängige Augmentierungsmethoden in der Akustik.

Um den Ansatz in einer industriellen Umgebung einsetzen zu können, muss die Ausführungszeit innerhalb der Anwendung schnell sein. Dazu gehört nicht nur die Vorhersagezeit, sondern auch die Trainingszeit, damit das Modell schnell eingesetzt werden kann. Tabelle 3.20 zeigt beispielhafte Ausführungszeiten mit den entsprechenden Vorverarbeitungsparametern. Die

### 3. In-situ Fehlererkennung im Druckprozess

Tabelle 3.16.: Experiment 1: Ergebnismetriken einer 5-fachen Cross-Validation für verschiedene Abtastraten und ein Mel-Spektrogramm  $256 \times 256$ . F1-Score ist für die Vorhersage nach Training und Validierung.

$F_S$ , Hz	Best ValAcc, %	Mean, %	F1-Score, %
176.400	98,3	86,9	98,5
352.800	98,5	85,4	52,5
705.600	98,8	89,6	82,1
1.411.200	98,6	88,1	99
2.822.400	98,6	85,4	98

Tabelle 3.17.: Experiment 2: Resultierende Metriken einer 5-fachen Cross-Validation für verschiedene Abtastraten und Mel-Spektrogramm der Größe  $512 \times 512$ . F1-Score ist für die Vorhersage nach Training und Validierung.

$F_S$ [Hz]	Accuracy	Best ValAcc, %	Mean, %	F1-Score [%]
176.400		99,7	74,7	41,4
352.800		97	78,9	97,4
705.600		93,4	74,3	91,8
1.411.200		97,1	89,7	82,6
2.822.400		95,9	89,3	95,3

Trainingszeit bezieht sich auf den gesamten Cross-Validationsprozess, die Vorhersage auf eine einzelne Probe. Eine Stichprobe ist 0,9 s lang. Die Trainingszeit erhöht sich, wenn eine höhere Abtastrate verwendet wird, und dauert im Fall von  $512 \times 512$  Spektrogrammen, die mit 2.822.400 Hz abgetastet wurden, sogar 11 Tage. Dies ist der einzige Fall, in dem eine Vorhersage in Echtzeit nicht möglich ist, so dass ein Einsatz nicht machbar ist. Die Spektrogramme mit einer Abtastrate von 4 MHz haben eine Größe von höchstens  $6.200 \times 1.025$ , die Spektrogramme mit 2.822.400 Hz haben also eine etwas geringere, aber ähnliche Größe, was zu dieser langen Dauer führt. Die anderen Modelle sind für diese Aufgabe geeignet.

Das Training mit den Spektrogrammen erwies sich im Vergleich zu den Mel-Spektrogrammen als unterdurchschnittlich, da es bei den meisten Metriken um 20% niedriger lag. Dies könnte auf die durchgeführte Vorverarbeitung zurückzuführen sein. Mit  $n_{mels} = 128$  wurde die Anzahl der Frequenzbins der Mel-Spektrogramme festgelegt und eher klein gehalten. Dies passt besser zu der folgenden Größenanpassung. Die Spektrogramme wurden im Allgemeinen größer. Zum Beispiel hatten Mel-Spektrogramme mit  $F_S = 176.400$  eine variable Größe von maximal  $1.152 \times 128$ . Die variable Größe ergibt sich aus der variablen Scangeschwindigkeit, wobei einige Schichten länger oder kürzer brauchen, was zu mehr oder weniger Samples führt. Im Gegensatz dazu hatte ein Spektrogramm mit der gleichen Abtastrate maximal  $1.025 \times 1.070$ .

### 3. In-situ Fehlererkennung im Druckprozess

Tabelle 3.18.: 5-fache Cross-Validation für verschiedene Abtastraten und Mel-Spektrogramm der Größe  $256 \times 256$

$F_S$ [Hz]	Fold #	1 [%]	2 [%]	3 [%]	4 [%]	5 [%]	Mean [%]
176.400		93	98,1	95	50,2	98,3	86,9
352.800		98,5	96,7	92,3	94,8	44,9	85,5
705.600		86,1	98,1	89,2	98,8	75,7	89,6
1.411.200		83,6	92,1	69,5	97	98,6	88,1
2.822.400		98,6	93,9	71,6	66,9	96,4	85,5

Tabelle 3.19.: Konfusionsmatrix für  $F_S = 176.400 Hz$

	Klasse 1	Klasse 2	Klasse3	F1-Score, %
Klasse 1	244	0	0	98
Klasse 2	10	192	0	97,5
Klasse 3	0	0	210	100
F1 Makro	98,5%			

1.025 ergibt sich aus der Kurzzeit-Fourier-Transformation (STFT), die ein Spektrogramm mit  $1 + n_{fft}/2$  Zeilen, d.h. Frequenzbins, erzeugt. Wenn diese Größe auf  $256 \times 256$  oder  $512 \times 512$  transformiert wird, gehen natürlich Informationen verloren. In weiteren Experimenten sollte dies berücksichtigt und  $n_{fft}$  niedriger angesetzt werden oder ein Modell mit einer größeren Eingabe und möglicherweise mehr Filtern verwendet werden, um sicherzustellen, dass das Modell die notwendigen Informationen extrahiert. Da die höchste Filtergröße 512 ist, siehe Abbildung 3.48, können zusätzliche Schichten mit den Größen 1.024, 2.048, usw. hinzugefügt werden, um die komplexe Struktur der Spektrogramme besser zu erfassen.

## 3.4. Fazit

Im Druckprozess integrierte Sensoren ermöglichen es Fehler während dem Druck zu erkennen und zu klassifizieren. In dieser Arbeit wurde das vorgestellte Verfahren mit dem Ansatz der Umwandlung der Audiodaten in Bilddaten mit Hilfe der MFCCs auf zwei verschiedene Drucktechnologien angewandt - der Fused Filament Fabrication und der Powder Bed Fusion. In beiden Verfahren konnten Fehler zuverlässig erkannt und klassifiziert werden. Die bisherigen Entwicklungen im Bereich des Deep Learnings für Bilddaten können auch erfolgreich genutzt werden, um damit Audiodaten zu verarbeiten. Gleichzeitig konnte gezeigt werden, dass es nur mit minimalen Aufwand möglich ist das Klassifizierungsverfahren auf einen anderen Drucker und eine andere Drucktechnologie zu übertragen.

### 3. In-situ Fehlererkennung im Druckprozess

Tabelle 3.20.: Ausführungszeiten für ausgewählte Experimente. Die Trainingszeit gilt für eine 5-fache Cross-Validation, d.h. 5 Trainings + Validierung. Die Vorhersage ist für eine Probe à 0.9 s. Es wurden zwei verschiedene Eingabeformate evaluiert. Die Mel-Spektrogramme und unveränderte Spektrogramme. Durch die lange Trainingsdauer bei gleichzeitig langer Inferenzzeit wurden Spektrogramme als Eingabe nicht weiter verfolgt, da sie im Vergleich zu den Mel-Spektrogrammen deutlich aufwändiger und zeitintensiver sind.

Bildtyp	Abtastrate	Eingabegröße	Training	Vorhersage
Mel-Spektrogramm	176.400	256	ca. 9 min	15 ms
Mel-Spektrogramm	176.400	512	ca. 23 min	15 ms
Mel-Spektrogramm	352.800	256	ca. 15 min	45 ms
Mel-Spektrogramm	2.822.400	512	ca. 7,5 h	48 ms
Mel-Spektrogramm	2.822.400	256	ca. 9,5 h	111 ms
Spektrogramm	176.400	256	ca. 3 h	88 ms
Spektrogramm	2.822.400	512	ca. 11 d	7,5 s

Jedoch müssen die entsprechenden Daten vorher in ein entsprechendes Eingabeformat, hier MFCC, umgewandelt werden. Da die Fehlererkennung in der additiven Fertigung allerdings keine Echtzeitkriterien erfüllen muss, ist diese Art von Datenverarbeitung hier problemlos anwendbar und kann zum Beispiel in einem Vorverarbeitungsschritt durchgeführt werden. Wie sich die Geometrie neuer, noch vom Netz bisher nicht gesehener Objekte auf die Auswertung auswirkt, ist bisher auch noch unbekannt.

# 4. Automatisierte Entnahme gedruckter Objekte

In diesem Kapitel wird das Greifen und Entnehmen der gedruckten Objekte behandelt. Einige Druckerhersteller, wie z.B. Creality mit dem CR-30 [99], haben ihren Drucker so konstruiert, das mit Hilfe eines Förderbandes das gedruckte Objekt bewegt werden kann. Das Förderband ist Teil des Achssystems und repräsentiert die Y-Achse. Ist ein Druck abgeschlossen, verfährt die Achse so lange, bis sich das gedruckte Bauteil am vorderen Ende vom Druckbett bzw. dem Förderband löst und zum Beispiel in eine Box fällt, aber nun ist der Drucker zwar wieder einsatzbereit. Der Nachteil an diesem Ansatz ist jedoch, dass das fertige Bauteil zum Einen nach abgeschlossenem Druck unkontrolliert vom Druckbett abgelöst wird, sprich es kann je nach Auffangbehältnis und Objektgeometrie leicht beschädigt werden. Zum Anderen ist die Lage und Orientierung durch den Fall des Objektes nun gänzlich unbekannt. Für eine weitere Verarbeitung mit Hilfe eines Roboters müsste für einen berechneten Griff zuerst das entsprechende Objekt z.B. mit Hilfe eines Kamerasystems identifiziert werden. Anschließend müsste an Hand der vorliegenden Position und Orientierung ein entsprechender Griff geplant werden. All diese Probleme entfallen, wenn das gedruckte Bauteil direkt von der Druckplatte entnommen wird, da die Position und Orientierung auf der Druckplatte bekannt sind.

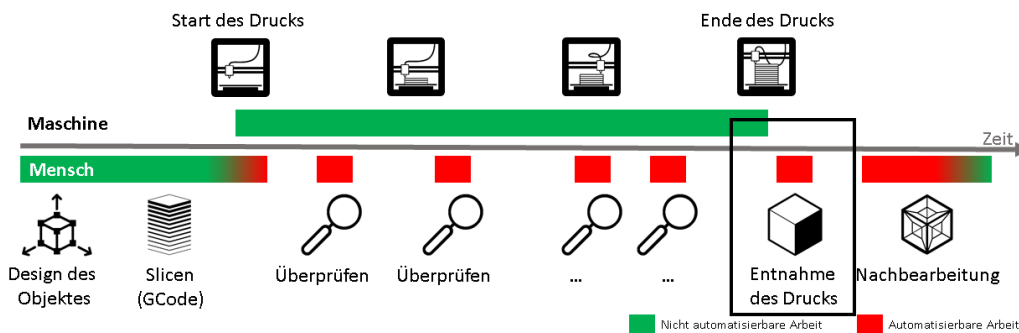


Abbildung 4.1.: In diesem Abschnitt der Arbeit liegt der Fokus auf der Entnahme des gedruckten Objekts, nachdem der Druck abgeschlossen wurde (schwarzer Kasten). Hierfür werden verschiedene Verfahren untersucht und angewandt, um das Objekt schadfrei und zuverlässig zu entnehmen.

## 4.1. Greifen der gedruckten Objekte

Um funktionierende universelle Greiffinger zu entwerfen, wurden 150.000 3D-Modelle analysiert, um einen angemessenen Bereich von Abmessungen zu berechnen. Um die genaue Position, Orientierung und Abmessungen des gedruckten Objekts zu bestimmen, analysiert ein einfacher, aber effektiver Algorithmus die vom Drucker bereitgestellte G-Code-Datei. Auf der Grundlage dieser Informationen wird eine Ablösestrategie gewählt, die eine Kombination aus einer oder mehreren der vier Grundbewegungen darstellt. Nach Erhalt dieser Informationen führt der Roboter den Entladevorgang aus. Des Weiteren werden fünf verschiedene Greiferansätze und -designs untersucht und mit einem Industriegreifer an einem Leichtbauroboter getestet.

### 4.1.1. Analyse von 3D-gedruckten Objekten

Um ein geeignetes Design für flexible Greiffinger zu entwickeln, sind einige Informationen über die gedruckten Objekte erforderlich. Deshalb wurden rund 150.000 3D-Objekte aus der Online-Community Thingiverse [144] ausgewertet. Die Abmessungen dieser STL-3D-Objekte wurden mit der Software ADMesh[145] ermittelt und gespeichert. Um die Verteilung der Objektbreiten zu bestimmen, wurde für jede Dimension (x, y, z) ein Histogramm erstellt. Die Dichte und die kumulative Verteilung wurden mit Hilfe von Standardverteilungsfunktionen mit minimaler Abweichung von den realen Werten approximiert, da die gegebenen Verteilungen unterschiedliche Approximationen erfordern.

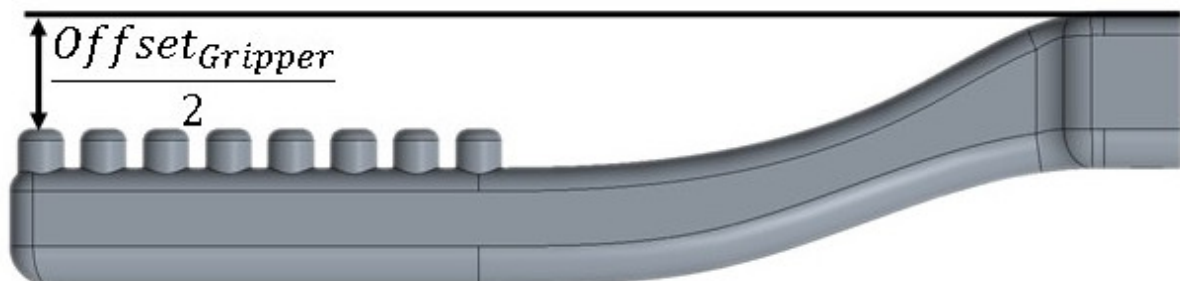


Abbildung 4.2.: Der Greiferoffset beschreibt die räumliche Differenz zwischen der Energieaufnahme und der Krafteinleitung. Quelle [187].

Dies geschah, um die häufigsten Dimensionen abzuschätzen und so eine effiziente Auswahl an greifbaren Objekten zu ermöglichen. Die Verteilung umfasst etwa 5% der durch Thingiverse verfügbaren Objekte; eine Erhöhung des Prozentsatzes verschiebt die Verteilung jedoch einfach weiter zur Y-Achse.

Es ist also offensichtlich, dass die häufigsten Abmessungen unter 80 mm liegen, wie in Abbildung 4.4a bis 4.4c zu sehen ist. Darüber hinaus wurde der minimal erforderliche Greiferoffset für einen bestimmten Bereich ermittelt, wie in Abbildung 4.3 dargestellt. Der Greiferversatz ist

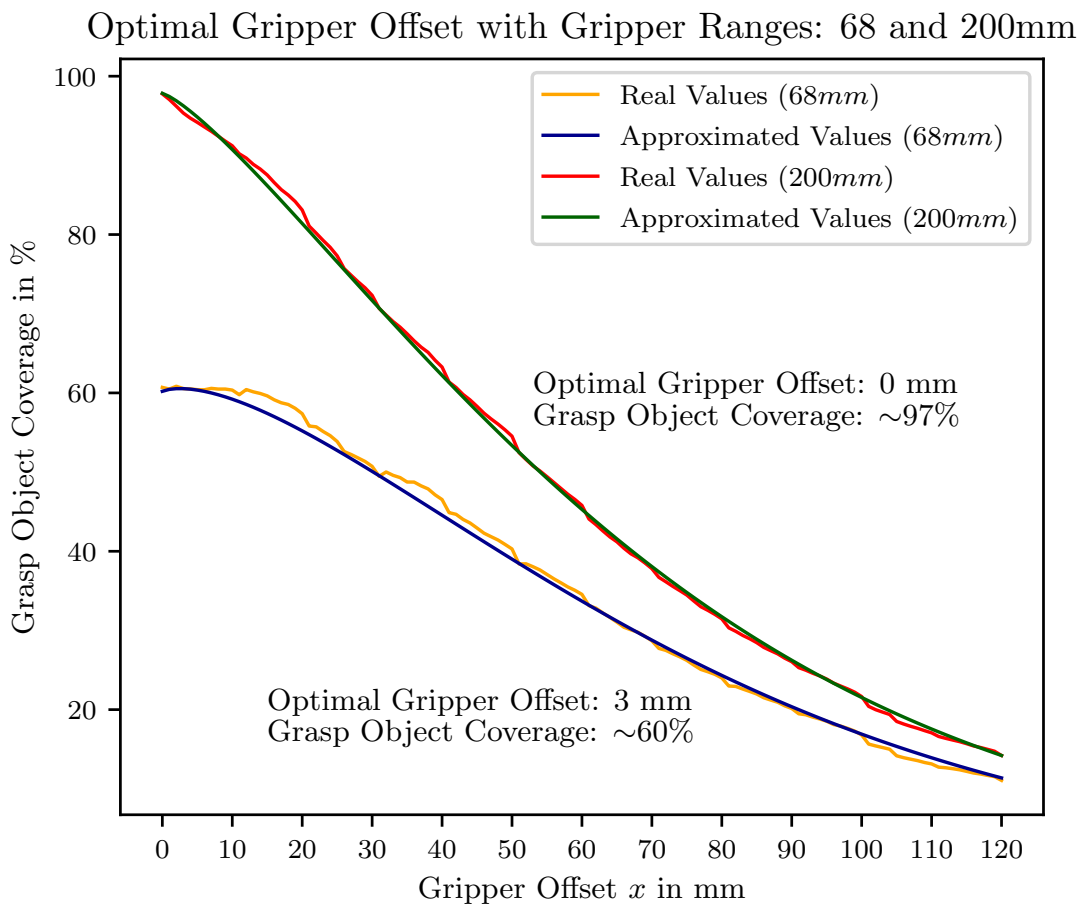


Abbildung 4.3.: Diese Grafik zeigt den Greifbereich von 68 und 200 mm, was zu einer 60%igen und 97%igen Greifobjektdeckung der analysierten Daten führt. Für die Approximierung wurde eine Kurve in die gemessenen Datenpunkte gelegt. Quelle [187].

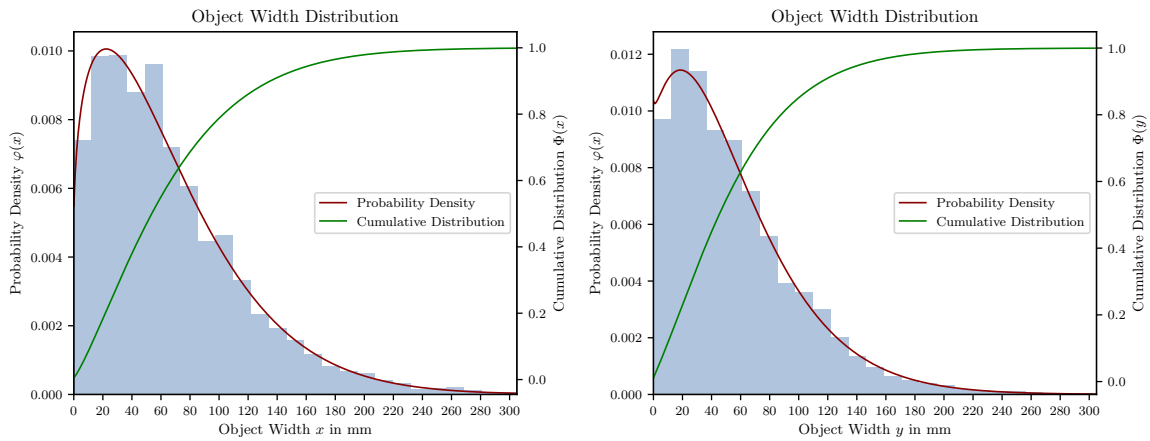
der Abstand zwischen den beiden Greifflächen, wenn der Greifer vollständig geschlossen ist, siehe Abbildung 4.2. Um auch kleine Objekte von der Druckplatte greifen zu können, sollte der Offset so klein wie möglich sein. Bei einem Offset von 0 mm und einem Bereich von 140 mm können mehr als 90% der Objekte gegriffen werden, siehe Abbildung 4.4a.

#### 4.1.2. Entwurf der Greiffinger

Geeignete Greiferfinger sind erforderlich, um 3D-gedruckte Objekte von der Druckplatte eines Druckers greifen und abnehmen zu können. Der Greifer muss eine Reichweite von 140 mm haben, um mehr als 90% aller analysierten Teile greifen zu können, und eine abriebfeste Greiffläche besitzen, um den Verschleiß zu verringern und die Lebensdauer des Greifers zu

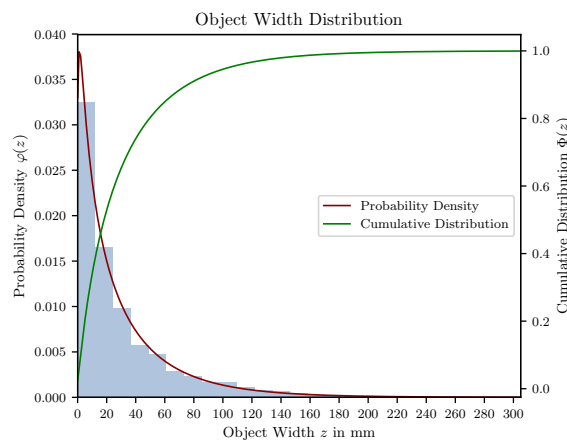


#### 4. Automatisierte Entnahme gedruckter Objekte



(a) Objektbreitenverteilung in x-Dimension

(b) Objektbreitenverteilung in y-Dimension



(c) Objektbreitenverteilung in z-Dimension

Abbildung 4.4.: Analyse der Dimensionen verschiedener Objekte in X-, Y- und Z-Achse. Es wird ersichtlich, dass ein Greifer mit 140 mm Öffnungsbreite 90% aller analysierten Objekte handhaben kann. Quelle [187].

erhöhen. Hierfür wurde mit Hilfe der Methodik des Morphologischen Kastens eine Vielzahl an Optionen und Varianten erstellt, die dann systematisch diskutiert und entsprechend hinterher hergestellt wurden. Der komplette Morphologische Kasten ist im Anhang A.6 abgebildet.

#### 4. Automatisierte Entnahme gedruckter Objekte



(a) 3D gedruckter Noppengreiferfinger. Versatz: 10 mm, Abdeckung: ~ 58%



(b) Gussnoppen-Greiferfinger. Versatz: 20 mm, Abdeckung: ~ 45%



(c) Luftgepolsterter Greiferfinger. Versatz: -10 mm, Abdeckung: ~ 56%



(d) Mit Polyurethanschaum gepolsterte Greiferfinger. Versatz: 0 mm, Abdeckung: ~ 63%



(e) Zangengreifer. Versatz: 0 mm, Abdeckung: ~ 97%

Abbildung 4.5.: Entworfen und bewertete Greiffinger mit unterschiedlichen Greifbereichen sowie unterschiedlichen flexiblen Greifflächen. Quelle [187].

#### 4. Automatisierte Entnahme gedruckter Objekte

Die Greiferfinger müssen sich auch an das 3D-gedruckte Objekt anpassen und es so weit wie möglich umschließen, um die erforderliche Ablösekraft über eine große Fläche zu übertragen und die Belastung des Objekts zu verringern. Dies verhindert eine Beschädigung der gedruckten Objekte während des Entladevorgangs und ermöglicht auch das Entladen von zerbrechlichen Objekten. An Hand des Ergebnisses des Morphologischen Kastens wurden fünf Greiferfingerentwürfe implementiert und getestet, um die Kosten und den Aufwand für eine 3D-Druckproduktion zu reduzieren.

Der in Abbildung 4.5a dargestellte Greiffinger besteht aus einer PLA-Basis und einer thermoplastischen Polyurethan-Noppen-Greiffläche. Die Noppenstruktur ermöglicht eine gute Kraffteinleitung, allerdings nicht über eine große Fläche, da das gedruckte flexible Material eine vielseitige Anpassung des Noppenprofils verhindert.

Daher wurde derselbe Greifer mit einer Noppenstruktur aus einem gegossenen Elastomerharz getestet, wie in Abbildung 4.5b zu sehen. Dies ermöglichte ein weitaus höheres Maß an Anpassung und erlaubte die erfolgreiche Entnahme mehrerer 3D-gedruckter Objekte.

Um die Anpassungsfähigkeit der Greiffläche weiter zu erhöhen, wurde ein luftgepolstertes Greiferfingerdesign angestrebt, siehe Abbildung 4.5c. Dieses Design erwies sich als zu anpassungsfähig und verhinderte eine korrekte Kraftübertragung auf das Objekt, was letztlich zum Scheitern des Ablösevorgangs führte. Mit einer Anpassung des verwendeten Elastomermaterials und der Kissenkonstruktion ist eine praktikable Lösung denkbar. Die geschätzten Kosten und die schlechte Effizienz des Luftkissens machten jedoch eine neue und verbesserte Konstruktion erforderlich.

Diese Verbesserung wurde durch den Austausch des Luftkissens gegen einen Polyurethanschaum realisiert. Wie in Abbildung 4.5d dargestellt, ist die Greiffläche zusätzlich mit einem abriebfesten Material überzogen, um die Lebensdauer des Greifers zu erhöhen. Diese Konstruktion hat sich als sehr erfolgreich erwiesen, da der Schaumstoff eine anpassungsfähige Oberfläche bietet, die in der Lage ist, die erforderliche Kraft über eine große Fläche zu übertragen, wodurch die Belastung des Objekts beim Entladen verringert wird.

Schließlich wurde die Reichweite des Greifers bei der in Abbildung 4.5e gezeigten Zangengreiferkonstruktion erhöht. Bei dieser Konstruktion erhöhte sich die Reichweite von 60%, mit einem Greifbereich von 68 mm, auf 97%, mit einer Reichweite von 200 mm. Diese Reichweite von 200 mm wurde in erster Linie gewählt, um die Reichweite des Greifers zu erhöhen und das Greifen von gedruckten Objekten aus jeder Position innerhalb des Bauvolumens zu ermöglichen. Die Greiffläche wurde aus demselben Elastomergießharz hergestellt, jedoch in Form eines konkaven Reifenprofils. Mit dem zusätzlichen Bonus eines größeren Greifbereichs und dem neuen Greifflächenprofil übertraf der Zangengreifer die gewünschten Kriterien.

Die Greiferdesigns wurden gegeneinander getestet, wobei der Finger aus Polyurethanschaum und der Zangengreifer am besten abschnitten. Bei den entwickelten Designs handelt es sich jedoch um Prototypen, die weiter optimiert und aus einem steiferen, leichteren Material wie Aluminium hergestellt werden müssen.

### 4.1.3. Strategien zum Lösen des Objekts von der Druckplatte

Wenn das geschmolzene Filament auf die Druckplatte aufgetragen wird, haftet das Material aufgrund der rauen Oberfläche auf der Platte. Um das fertige Objekt abzulösen, müssen diese Bindungskräfte überwunden werden. Eine optimierte Strategie zum Ablösen des Objekts von der Druckplatte des Druckers hat mehrere Vorteile. Erstens wird das Objekt weniger geschädigt, da die Belastung während des Ablösevorgangs reduziert wird. Dies ist erforderlich, um eine zerstörungsfreie Ablösung zu gewährleisten. Zweitens wird auch die Roboterhardware geschont, da die Kräfte und Drehmomente geringer sind.

Drei Translationsachsen (X-, Y- und Z-Achse) und drei Rotationen (um die X-, Y- oder Z-Achse) sowie eine Kombination dieser Achsen können zum Lösen des Objekts durchgeführt werden. Ein Mitarbeiter würde immer die Höhe des Objekts ausnutzen, um bei einer Drehung um die X- oder Y-Achse Hebelkräfte zu erzeugen. Nach einigen Tests wurde diese Methode auch in Kombination mit einer Verschiebung in der Z-Achse implementiert. Da einige Modelle zerbrechlich sein können, wird die Drehung über eine ihrer Kanten ausgeführt, um das Objekt nicht zu beschädigen.

Die Adhäsionskräfte zwischen der Druckplatte und dem Objekt können die Tragfähigkeit des Roboters übersteigen, was eine erfolgreiche Entnahme des Objekts behindert, wenn die falsche Strategie verwendet wird. Die richtige Strategie hingegen nutzt Hebelkräfte und Drehmomente, wodurch die vom Roboter maximal aufzubringende Kraft reduziert wird.

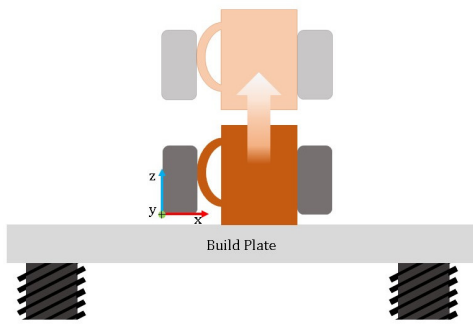
### 4.1.4. Experimente und Evaluation zum Greiferdesign

Geeignete Techniken wurden zur Evaluation eingesetzt, um die notwendigen Ablösekräfte zu bestimmen und die Ablösestrategien genau zu bewerten und zu modellieren. Um eine optimale Strategie zu wählen, die für das aktuelle Objekt verwendet wird, wird eine Analyse des gegebenen G-Codes durchgeführt. Nach der Analyse wird die eingesetzte Hard- und Software kurz beschrieben und ein Lebensdauertest mit den vielversprechenden Greiffingeransätzen durchgeführt.

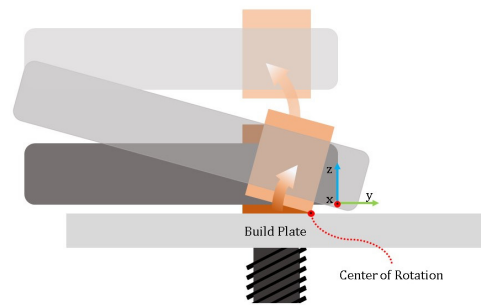
### 4.1.5. Analyse der Ablösekraft

Um eine Abschätzung der erforderlichen Ablösekraft zu ermitteln, die benötigt wird, um ein gedrucktes Objekt mit einer gegebenen Oberfläche von einer Druckplatte abzulösen, wurde eine Analyse ähnlich der von Teliskova *et al.* durchgeführt. [146]. Wie in Abbildung 4.7 dargestellt, wurden fünf Objekte auf die Druckplatte gedruckt. Diese wurden mit Hilfe eines Kraftmessers und eines vertikalen und horizontalen Zugs von der Druckplatte abgelöst. Das Worst-Case-Szenario einer vertikalen Ablösung von der Druckplatte ist in Abbildung 4.7 dargestellt und zeigt mehrere Positionen und unterschiedliche Oberflächenbereiche. Bei einer Kontaktfläche von  $1.000 \text{ mm}^2$  wurde nur Position drei weiterverfolgt, da das Bauvolumen des

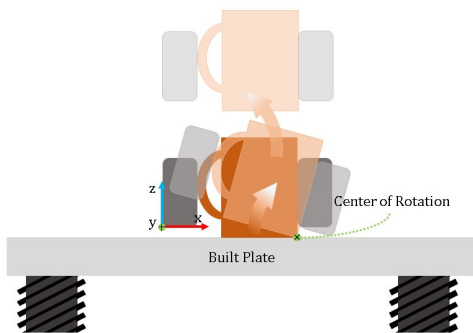
#### 4. Automatisierte Entnahme gedruckter Objekte



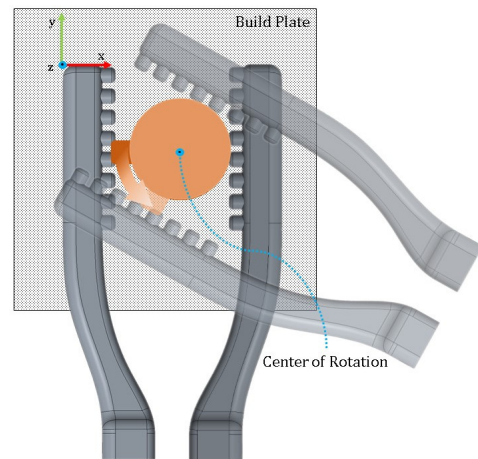
(a) Translation entlang der Z-Achse



(b) Rotation um die X-Achse



(c) Rotation um die Y-Achse



(d) Rotation um die Z-Achse

Abbildung 4.6.: Schematische Darstellung der vier Grundelemente zur Erzeugung der Objekt-ablösestrategien: Ausführen einer Translation entlang der Z-Achse (a), einer Rotation um eine der Achsen X (b), Y (c) oder Z (d). Quelle [187].

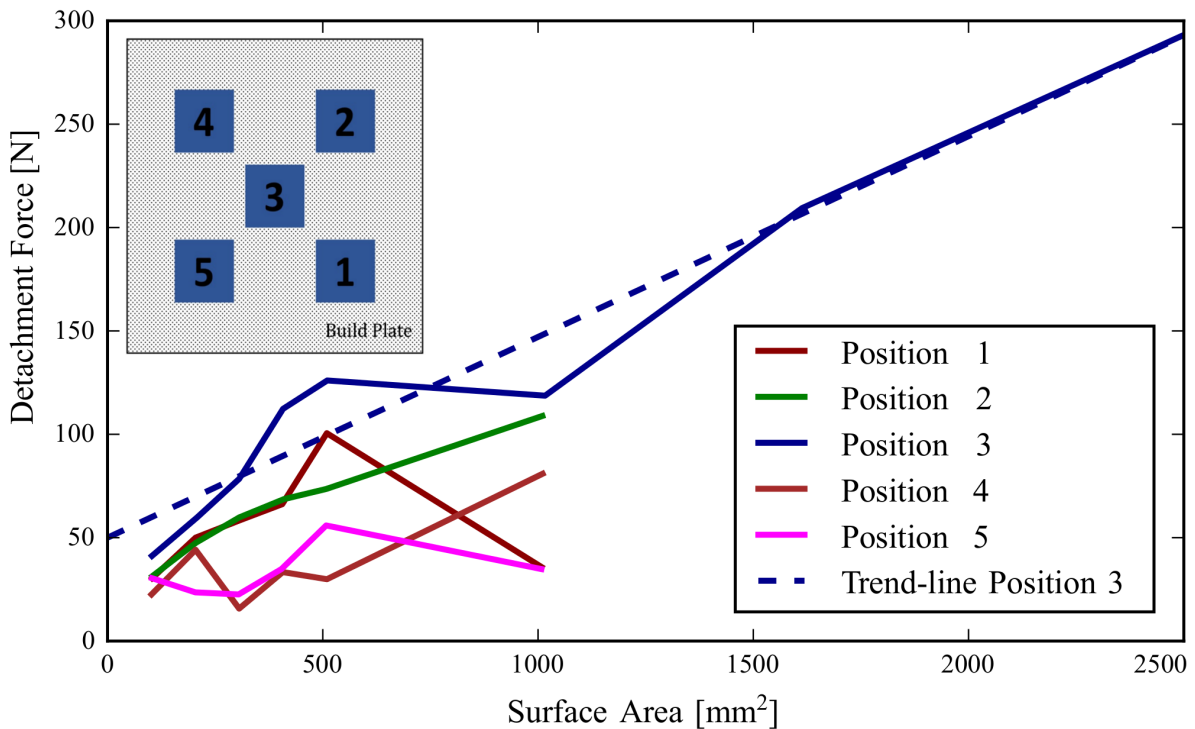


Abbildung 4.7.: Die Bewertung der Ablösekraft erfolgte anhand der Position und der Kontaktfläche zwischen dem Objekt und der Druckplatte, um das Entlastungsverhalten bei zukünftigen Ablösungen abzuschätzen (wie bei der Trendlinie Position 3 beschrieben). Hierfür wurden die gedruckten Objekte mit Hilfe einer Waage von der Druckplatte abgezogen und die dabei auftretende Kraft gemessen. Quelle [187].

gewählten Druckers für alle fünf Positionen zu klein war und die anderen Positionen einen Reduzierung der erforderlichen Kräfte aufwiesen. Nach einer Analyse lässt sich sagen, dass das einfachere Ablösen auf die unebene Druckplatte zurückzuführen ist, die nicht perfekt parallel zur Extruderebene liegt. Geringfügige Unterschiede führen dazu, dass sich die Adhäsionskräfte ändern und meist abnehmen, weil das Material nicht richtig an der Druckplatte haftet. Daher sollte die vorgestellte Trendlinie nur dazu verwendet werden, eine ungefähre erforderliche Ablösekraft in  $N$  für das Worst-Case-Szenario eines vertikalen Zugs abzuschätzen.

#### 4.1.6. G-Code Analyse zur Positionsbestimmung

Bei den verschiedenen vorgeschlagenen Ablösestrategien muss dem Roboter die Position des Objekts bekannt sein. Diese wird in der Regel vom Benutzer während des Slicings festgelegt, ebenso wie der Maßstab des Objekts selbst. In Verbindung mit den hinzugefügten Stützstrukturen weichen die Abmessungen des gedruckten Objekts von denen des entworfenen Objekts ab. Um die tatsächliche Position und die Abmessungen auf der Druckplatte zu bestimmen,

#### 4. Automatisierte Entnahme gedruckter Objekte

analysiert ein Algorithmus den G-Code des Druckauftrags. G-Code ist die am weitesten verbreitete Programmiersprache für numerische Steuerungen [147]. Sie wird zur Steuerung des Druckers verwendet und enthält alle für die Herstellung des Objekts erforderlichen Parameter und Bewegungen.

Für jede Schicht überprüft der Algorithmus die Bahnen des Druckkopfes, um die Position auf der Druckplatte sowie die Abmessungen in X, Y und Z zu berechnen. Sobald die Position und die Ausrichtung bestimmt sind, berechnet der Algorithmus die erforderliche Greifposition sowie die Öffnungs- und Schließbreite des Greifers, um eine optimale Entnahme des 3D-Druckteils zu gewährleisten. Dies geschieht, um die Belastung der beteiligten Teile zu reduzieren und den Entnahmeprozess so effizient wie möglich zu gestalten. Mit den ermittelten Informationen kann das Objekt mit Hilfe einer optimierten Ablösestrategie unter Verwendung der inversen Kinematik aus dem Bauvolumen des Druckers entfernt werden, wie im Abschnitt 4.1.3 beschrieben.

##### 4.1.7. Belastungstest

Zur Anwendung des Konzepts und zur Nutzung der entworfenen Greiffinger wurde ein Universal Robots UR5 (A.1.5), mit einer maximalen Traglast von 5 kg, in Kombination mit einem Schunk PG 70-Plus Greifer, mit 68 mm Greifbereich, verwendet. Beide werden mit Robot Operating System (ROS) gesteuert. Durch die modulare Softwarearchitektur von ROS ist es möglich mit geringer Vorlaufzeit einen solchen Versuchsaufbau aufzubauen. Da die maximale Nutzlast des Roboterarms 5 kg beträgt und der Greifer mit Zubehör etwa 1,5 kg wiegt, bleiben 3,5 kg für die Handhabung der gedruckten Objekte übrig.

Die meisten der gedruckten Objekte sind klein und leicht, aber um das Objekt von der Druckplatte zu lösen, wird das Drehmoment des Arms benötigt. In einer Zustandsmaschine prüft die Software über eine Schnittstelle periodisch den aktuellen Zustand des Druckers. Wenn der Auftrag erledigt ist, lädt der G-Code-Analysealgorithmus den G-Code vom Drucker-Backend herunter und berechnet die Position sowie die Abmessungen des Objekts. Nun ist die Greifposition bekannt und der Roboter kann beginnen, das Objekt zu lösen. Dazu wird der Greifer auf die spezifische Öffnungsweite eingestellt. Anschließend bewegt sich der Roboter innerhalb des Bauvolumens des Druckers. Wenn er sein Ziel erreicht hat, schließt sich der Greifer und nimmt Kontakt mit dem Objekt auf. Eine Überprüfung, ob der Greifer geschlossen ist und ein Objekt zwischen seinen Greiferfingern hat, bestimmt, ob der Roboter seine Ablösestrategie ausführen soll. Ist dies der Fall, wird die Strategie ausgeführt und das Objekt von der Druckplatte gelöst; es wird geprüft, ob die Ablösung erfolgreich war, indem sichergestellt wird, dass sich das Objekt in den Greiferfingern befindet; und bei erfolgreicher Ablösung wird es automatisch in einem bestimmten Bereich abgelegt.

Da der 3D-Druck von Objekten viel Zeit in Anspruch nimmt und die entwickelten Greifer und Ablösestrategien ausgiebig getestet werden müssen, wurde ein Versuchsaufbau entwickelt, der weniger Zeit in Anspruch nimmt, da nicht gewartet werden muss, bis jeder Druck abgeschlossen ist. Ziel dieses Aufbaus ist es, die Ablösestrategien zusammen mit dem Verhalten der Greifer

#### 4. Automatisierte Entnahme gedruckter Objekte

zu testen, wie z. B. die Abnutzung der Greiferfinger, die Benutzerfreundlichkeit, die Effizienz und die Lebensdauer. Das Experiment wurde mit Hilfe einer ferromagnetischen Oberfläche und Magneten durchgeführt, die an einem 3D-gedruckten Objekt befestigt wurden, siehe Abbildung 4.8. Um die Adhäsionskräfte darzustellen, wurden Magnete unterschiedlicher Stärke an einem 3D-gedruckten Vergleichsobjekt, Benchy, befestigt, das auf der ferromagnetischen Oberfläche platziert wurde. Die maximale Kraft, die mit jeder Greifereinrichtung, jedem Greifer und jeder Ablösestrategie überwunden werden kann, wurde anhand des oben dargestellten Worst-Case-Szenarios gründlich analysiert. Es wurde festgestellt, dass der Aufbau nur als Schätzung des Systemverhaltens während des Ablösevorgangs akzeptabel ist, da die Magnete eine translatorische Bewegung auf der ferromagnetischen Oberfläche ermöglichen, im Gegensatz zu den Haftkräften, die auf das Objekt und die Druckplatte wirken. Außerdem stieß der UR5 mit der analysierten Magnetkraft von etwa 42 N an seine Grenzen.

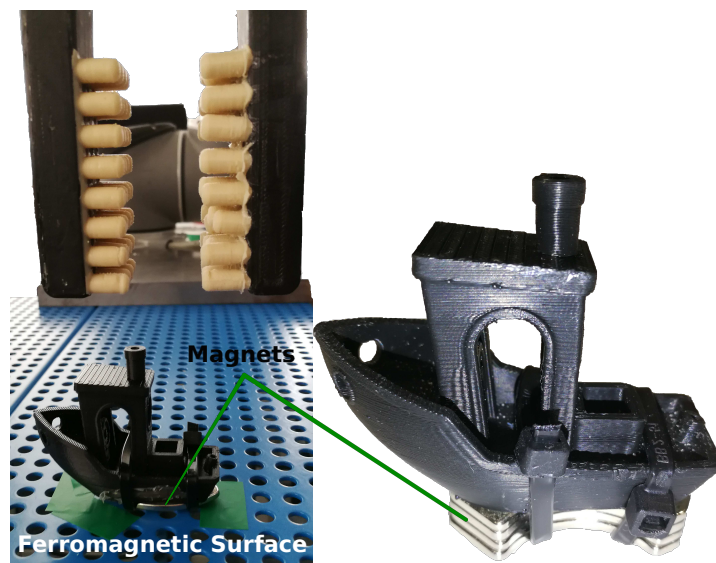


Abbildung 4.8.: Aufbau zur Simulation der Adhäsionskräfte zwischen dem gedruckten Objekt und der ferromagnetischen Oberfläche. Die auftretenden Kräfte werden durch das Hinzufügen von Magneten zum 3D-gedruckten Objekt simuliert, um die vorgeschlagenen Greiferfinger und Ablösestrategien mit kurzen Zykluszeiten bewerten zu können. Quelle [187].

Um einen Lebensdauertest durchzuführen und die Abnutzung der Greiferfinger zu bewerten, wurde das Referenzobjekt mit den angebrachten Magneten auf die ferromagnetische Oberfläche gelegt, mit einer bestimmten Ablösestrategie und einem Greifer von der Oberfläche entfernt und erneut auf die Oberfläche gelegt. Dieser Vorgang wurde für jede Ablösestrategie und jeden Greifer mehr als zweihundertmal wiederholt. Der daraus resultierende Verschleiß der Gussgratgreiferfinger ist in Abbildung 4.9 dargestellt. Bei diesen Greiferfingern ist der Verschleiß minimal. Die Greiferfinger mit Schaumstoff hingegen zeigten einen hohen Verschleiß. Um dem entgegenzuwirken, wurde eine Schutzhülle hinzugefügt, die die Lebensdauer der Greifer deutlich erhöht.



#### 4. Automatisierte Entnahme gedruckter Objekte



(a) Noppen-Elastomerprofil vor dem Lebensdauertest.



(b) Gegossenes Elastomerprofil nach wiederholter Nutzung

Abbildung 4.9.: Eine Nahaufnahme vor und nach der Bewertung des gegossenen, entgrateten Greiferfingers, der nach mehr als 2000 Benutzungszyklen wenig Verschleiß zeigt. Quelle [187].

Nach Abschluss der Tests mit der ferromagnetischen Oberfläche wurde die Ablösung eines 3D-gedruckten Objekts von der Druckplatte des Druckers mehrmals mit dem besten Greifersetup getestet, das den Gussgrat-, den schaumstoffgepolsterten und den Zangengreifer umfasste. Diese Tests wurden mit einer Vielzahl von 3D-gedruckten Objekten unterschiedlicher Größe und Form durchgeführt. Die meisten der getesteten Entladevorgänge waren erfolgreich, was die Effizienz des Verfahrens und der getesteten Greiferfinger bestätigte. Somit stellt dieses automatisierte Greifen von 3D-gedruckten Teilen einen der ersten erfolgreichen Ansätze dar.

## 4.2. Sequentielle Entnahme mehrerer gedruckter Objekte von der Druckplatte

Im Folgenden wird ein Ansatz vorgestellt, der die Anzahl, Positionen und Größen aller gedruckten Objekte durch Analyse der G-Code-Datei des aktuellen Druckauftrags bestimmt. Es wird ermittelt, ob und in welcher Reihenfolge die Objekte durch einen Roboterarm entfernt werden können.

Die vorgestellte Methode nutzt die Druckdaten in Form von G-Code-Daten, um die Berechnung der aktuell gedruckten Objekte und ihrer Position und Größe durchzuführen. Eine zusätzliche, optische Überprüfung ist notwendig bevor die Objekte entnommen werden, um mögliche Veränderungen der gedruckten Strukturen aufgrund von Fehlern während des Druckvorgangs

#### 4. Automatisierte Entnahme gedruckter Objekte

zu erkennen. Dies ist auf Fehler wie Schichtverschiebung, Spaghetti oder einem verstopften Extruder zurückzuführen. Hierfür wird zusätzlich eine Stereo-Kamera eingesetzt, um die Hypothese direkt nach dem Druckvorgang zu verifizieren. Der Roboter ist in der Lage, die Kamera in verschiedene Positionen zu bewegen, um die beste Perspektive für den aktuellen Druckauftrag zu erhalten, und die Objekte mit seinem Greifer von der Druckplatte zu entfernen. Durch die Analyse der 3D-Punktwolke wird die Hypothese bestätigt und die Objektentnahme durchgeführt, andernfalls wird die Greifposition angepasst und, falls ein Mensch eingreifen muss, wird die entsprechende Person benachrichtigt. Im letzten Schritt werden die Objekte automatisch durch einen Roboter von der Druckplatte entfernt.

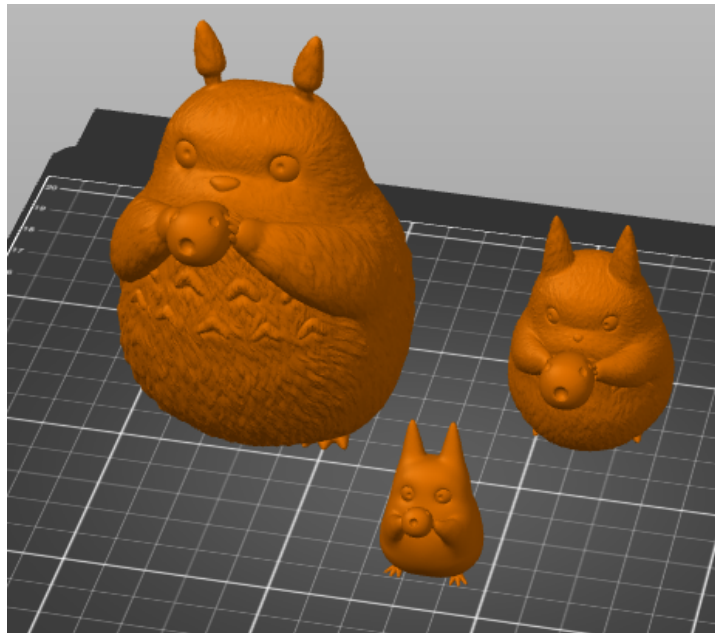


Abbildung 4.10.: Beispiel aus der Slicing-Software mit 3 stl-Dateien, die auf der Druckplatte platziert werden und die Grundlage für den generierten G-Code bilden. Quelle [189].

Die Ausnutzung der gesamten Druckplatte eines Druckers durch das Drucken mehrerer Objekte auf einmal ist effektiv und durch die größere Anzahl an zu druckenden Objekten steigt die Druckdauer. So kann es schnell mehrere Stunden dauern, bis der Drucker den kompletten Auftrag erledigt hat. Derzeit gibt es jedoch keine Möglichkeit, diese Objekte nach einem erfolgreichen Druck automatisch zu entfernen, ohne spezielle Druckerhardware zu verwenden, und direkt danach den nächsten Auftrag zu starten. Dies wird bisher manuell durchgeführt.

Der vorgeschlagene Ansatz ist in der Lage, die Position jedes Objekts aus dem G-Code zu berechnen und einen Roboter so zu bewegen, dass alle Objekte automatisch nacheinander entfernt werden. Er besteht aus zwei Aspekten: Erstellung der Hypothese der gedruckten Objekte auf der Grundlage des G-Codes und Überprüfung der Hypothese anhand einer gemessenen Punktwolke. Danach ist der Serviceroboter in der Lage, nacheinander alle Objekte zu entfernen, je nach der Hypothese für jedes Objekt.

### 4.2.1. Bestimmung der Objektanzahl, -position und Entnahmereihenfolge anhand des G-Codes

Zu einer erfolgreichen Entnahme eines Objektes sind mehrere Parameter notwendig.

- Die Position des Objekts auf der Druckplatte
- Die Griffbreite, die für den Greifvorgang notwendig ist
- Ob das Objekt direkt entnommen werden kann, oder ob noch weitere Objekte vorher entnommen werden müssen, damit der Roboter das Objekt zuverlässig greifen kann.

Diese drei Punkte können alle durch eine Analyse des G-Code bestimmt werden.

### 4.2.2. Bestimmung der Anzahl der Objekte

Die Struktur einer G-Code-Datei ermöglicht eine sehr einfache Extraktion der  $(X, Y, Z)$ -Koordinaten, die zu den Objekten gehören, mit Hilfe regulärer Ausdrücke. Um die Position aller Objekte zu ermitteln, werden alle  $(X, Y, Z)$ -Koordinaten auf die  $(X, Y)$ -Ebene projiziert. Dabei bleibt der Datensatz aussagekräftig, kann aber mit Standardalgorithmen leicht geclustert werden.

Clusteranalysen sind Verfahren zur Feststellung von Ähnlichkeiten einzelner Datenpunkte innerhalb meist sehr großer Datenmengen. Im Zusammenhang mit dem Parsen einer G-Code-Datei ist Ähnlichkeit definiert als die Zugehörigkeit von  $(X, Y)$ -Punkten zum selben Objekt. Dem System ist im Voraus nicht bekannt, wie viele Objekte gleichzeitig auf einer Druckplatte gedruckt werden. Daher eignen sich für dieses Clustering nur die Algorithmen, die die Anzahl der Cluster automatisch erkennen. Die Algorithmen MeanShift [148], DBSCAN [149] und OPTICS [150] erwiesen sich als geeignet. Durch Iteration wurde herausgefunden, dass der Mindestabstand zwischen den gedruckten Objekten mindestens 20 mm in der  $(X, Y)$ -Ebene betragen soll. Tabelle 4.1 zeigt die Ergebnisse des Clusters mit verschiedenen Algorithmen und Abstandsbreiten. In der ersten Zeile wurde kein Mindestabstand festgelegt. Im zweiten Fall betrug der Mindestabstand 20 mm. Bei weiteren Tests erwies sich der Mindestabstand von 20 mm als optimal. Bei 10 oder 15 mm Abstand führt das Clustering immer noch zu falschen Ergebnissen.

### Bestimmung der Kameraausrichtung

Um später eine optimale Kameraperspektive auf das Objekt zu erhalten, wird im Vorfeld eine geeignete Kameraausrichtung berechnet. Mittels linearer Regression wird eine Gerade gefunden, die die Clusterpunkte auf der  $(X, Y)$ -Ebene am besten approximiert. Dabei können entweder alle  $(X, Y)$ -Koordinaten oder, nur die Koordinaten der Clusterzentren approximiert werden. Die Steigung  $k$  der berechneten Geraden ist in beiden Fällen annähernd gleich, wie in Abbildung 4.11 gezeigt. Der Parameter  $k$  wird für die Positionierung des TCP übernommen:

#### 4. Automatisierte Entnahme gedruckter Objekte

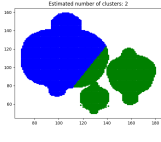
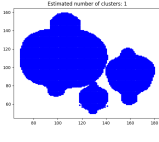
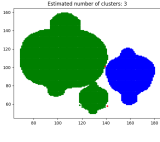
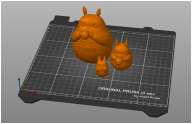
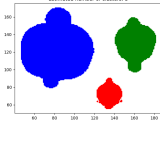
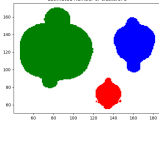
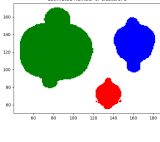
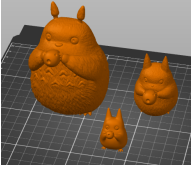
	MeanShift	DBScan	OPTICS	STL-file
Anzahl der Cluster	2	1	2 (+ Ausreißer)	
Graph				
Anzahl der Cluster	3	3	3	
Graph				

Tabelle 4.1.: Unterschiedliche Clustering-Algorithmen ohne (obere Zeile) und mit (untere Zeile) Mindestabstand zwischen den Objekten in der G-Code-Datei. Quelle [189].

die X-Achse des Kamerakoordinatensystems sollte möglichst parallel zur berechneten Linie liegen.

#### Bestimmung des geometrischen Objektzentrums oder des Massenschwerpunkts

Bei der MeanShift-Methode werden die Clusterzentren automatisch berechnet, sodass keine zusätzliche Bestimmung erforderlich ist. Somit sind die  $(X, Y)$ -Koordinaten des jeweiligen Objektzentrums bereits vorhanden. Bei der Verwendung von DBSCAN oder OPTICS müssen die Clusterzentren explizit berechnet werden. Die entsprechende  $Z$ -Koordinate wird durch Re-Parsen der G-Code-Datei ermittelt. Dabei wird die Suche auf die Koordinaten der Mitglieder des entsprechenden Clusters beschränkt, was durch die Festlegung der Constraints wie  $x_{max}$ ,  $y_{max}$ ,  $x_{min}$ ,  $y_{min}$  möglich ist.

Nach Beendigung des Parsings sind die zum Objekt gehörenden  $(X, Y, Z)$ -Koordinaten verfügbar. Extrahiert man die  $Z$ -Koordinaten und fügt sie zu einer Menge zusammen, erhält man aufgrund der Struktur der G-Code-Datei eine aufsteigend sortierte Liste, in der jede  $Z$ -Koordinate einmal vorkommt. Der Mittelwert dieser Liste ist die gesuchte  $Z$ -Koordinate  $Z_{center}$ . Kombiniert man diese mit den  $(X, Y)$ -Koordinaten des Clusterzentrums, erhält man den Punkt, den der Werkzeugmittelpunkt des Roboters zum Greifen anfahren soll.

Der Objektmittelpunkt ist jedoch nicht unbedingt der Massenmittelpunkt. Um die Stabilität des Griffs zu erhöhen, ist es sinnvoll, den Massenschwerpunkt des Objekts zu finden, unter der Annahme, dass das Objekt nicht hohl ist. Zu diesem Zweck wird eine Koordinate  $Z_{mass} \in Z_{obj}$  so gefunden, dass die Differenz zwischen der Summe  $S_{unter}$  der Querschnittsflächen für  $Z_i < Z_{mass}$  und  $S_{over}$  der Querschnittsflächen für  $Z_i > Z_{mass}$  am kleinsten ist.

#### 4. Automatisierte Entnahme gedruckter Objekte

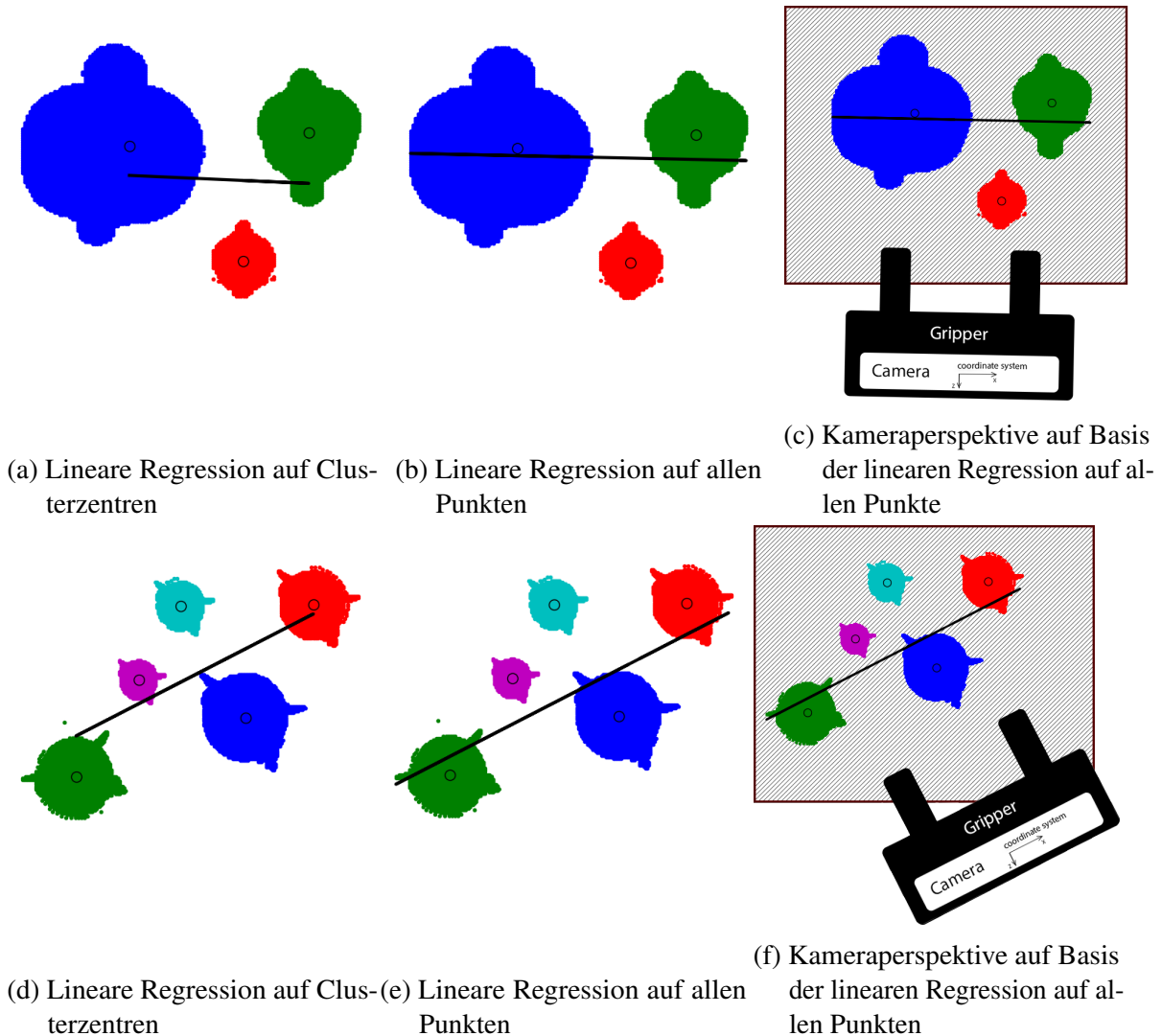


Abbildung 4.11.: Je nachdem welche Zentren für die Bestimmung der linearen Regression verwendet werden, unterscheidet sich die spätere Perspektive der Kamera am Roboterarm. Quelle [189].

Da beliebige komplizierte Objektgeometrien gedruckt werden, ist es nicht möglich, die genauen Querschnittsflächen zu berechnen. Daher werden die Flächen auf die umschriebenen Vierecke oder Kreise reduziert. Für jede  $Z$ -Koordinate des Objekts werden  $x_{max}$ ,  $y_{max}$ ,  $x_{min}$ ,  $y_{min}$  bestimmt.

Wenn die Fläche auf ein Viereck reduziert werden soll, wird die Fläche wie folgt berechnet.

$$A = (x_{max} - x_{min}) \cdot (y_{max} - y_{min}). \quad (4.1)$$

Wenn die Fläche auf einen Kreis reduziert werden soll, ist der Radius des Kreises

$$r = \frac{\max((x_{max} - x_{min}), (y_{max} - y_{min}))}{2} \quad (4.2)$$

#### 4. Automatisierte Entnahme gedruckter Objekte

und die Fläche somit

$$A = \pi \cdot r^2. \quad (4.3)$$

Die Bestimmung von  $Z_{mass}$  und die Kombination mit den  $(X, Y)$ -Koordinaten des Clusterzentrums,  $(X_{mass}, Y_{mass}, Z_{mass})$ , liefert die vorhergesagten Koordinaten für den TCP des Roboterarms.

#### Bestimmung der Dimension des Objekts auf der Greifebene

Nachdem  $Z_{mass}$  gefunden ist, wird die Schließbreite für den Greifer bestimmt. Beim Start des Softwarestacks wird der G-Code-Analyse als einer der Parameter die Höhe der Greiferfinger  $h$  übergeben.

Sei  $d$  die Breite des Objekts auf der Ebene  $Z = Z_{mass}$  und  $h$  die Höhe der Finger des Greifers.  $d_1$  sei die Breite des Objekts auf  $Z_1 = Z_{mass} - \frac{h}{2}$  und  $d_2$  sei die Breite des Objekts auf  $Z_2 = Z_{mass} + \frac{h}{2}$ . Die Breiten  $d_3, \dots, d_n$  seien die Breiten auf allen  $Z_n$  zwischen  $Z_1$  und  $Z_2$  ( $Z_n \in Z_{obj}$ ). Die endgültige Schließbreite für den Greifer ist dann  $Breite = \max(d_1, \dots, d_n)$ . Abbildung 4.12 und 4.13 verdeutlichen die Notwendigkeit dieser Berechnung. Abbildung 4.13 zeigt auch, warum der Massenschwerpunkt dem Objektzentrum vorgezogen werden sollte.

Die Breite  $d_i$  für  $Z = Z_i$  kann dann wie folgt bestimmt werden:  $d_i = X_{max,i} - X_{min,i}$ , wobei  $(X_{max,i}, Y, Z_i)$  und  $(X_{min,i}, Y, Z_i)$  Punkte des Objekts auf der Ebene  $Z = Z_i$  sind und  $Y \leq Y_{mass}$  gilt. Wenn die Breite des Objekts erlaubt ist, bedeutet dies nicht unmittelbar, dass der Massenschwerpunkt des Objekts durch den TCP angefahren werden kann. Ist der Abstand zwischen  $y_{center}$  und  $y_{min}$  größer als die Länge der Greifflächen der Greiffinger, muss der anzufahrende Punkt entlang der y-Achse verschoben werden.

#### 4.2.3. Bestimmung der Greifmöglichkeit

Bei der Annäherung gibt es hauptsächlich folgende Situationen, in denen die Entnahme eines Gegenstandes nicht möglich ist:

- Wenn die erforderliche Schließbreite `width_closed` des Greifers minus 16 mm größer ist als die maximale Greiferöffnung, kann das Objekt nicht entnommen werden. Bei der Annäherung an das Objekt müssen die Greiferbacken weiter geöffnet werden als die Objektbreite, um eine Kollision mit dem Objekt zu vermeiden. Ein Abstand von  $2 \cdot 8 \text{ mm} = 16 \text{ mm}$  ist eingerechnet, um Schäden zu vermeiden.
- Wenn die Schließbreite des Greifers kleiner ist als die minimale Greiferöffnung, kann das Objekt nicht entnommen werden. Die minimale Greiferöffnung beträgt 0 mm, aber der Abstand zwischen den Greiferbacken ist in dieser Situation nicht unbedingt 0. Bei den hier verwendeten Greiferbacken beträgt der Abstand zwischen den Greiferbacken im geschlossenen Zustand 6,5 mm.

#### 4. Automatisierte Entnahme gedruckter Objekte

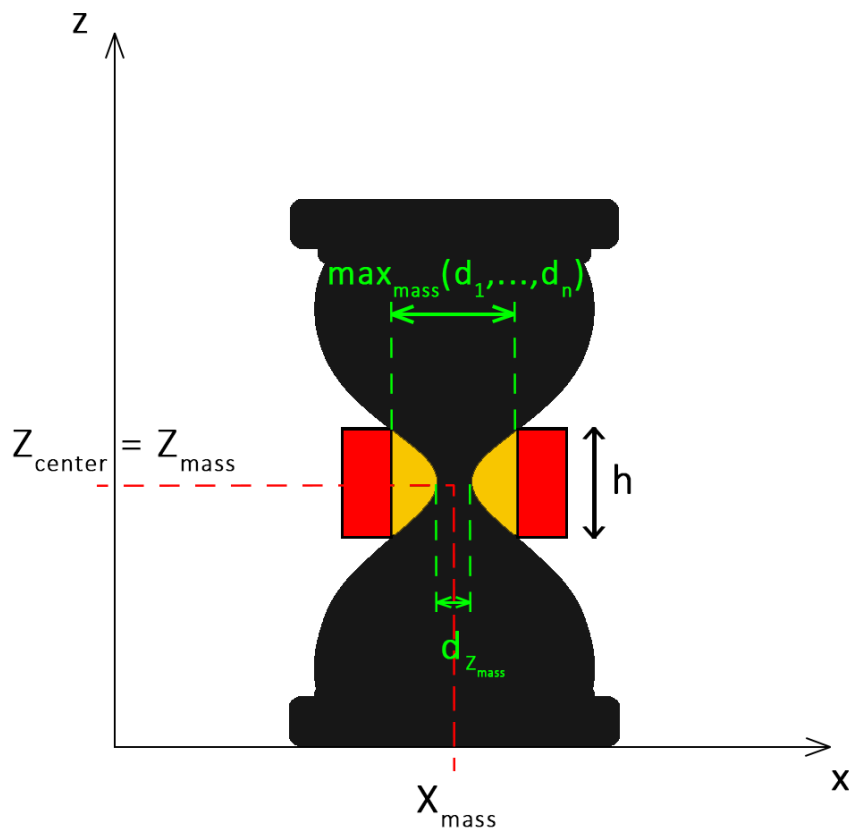


Abbildung 4.12.: Beispiel eines Druckobjekts in der XZ-Ebene: Keine Möglichkeit, den Greifer in Richtung  $d_{z_{mass}}$  zu bewegen, da die Greiferbacken zu breit für die berechnete Position sind. Quelle [189].

- Beträgt die Höhe des Objekts weniger als 20 mm, kann das Objekt nicht entnommen werden, da der Roboter den Drucker durch Anschlagen der Druckplatte beschädigen könnte.
- Wenn das Objekt A mit zulässiger Breite in der Projektion auf die XZ-Ebene von einem anderen Objekt B verdeckt wird, das nicht gegriffen werden kann, kann das Objekt A auch nicht entnommen werden.

Wenn das aktuelle GreifszENARIO keinen der obigen Fälle darstellt, kann der Roboter das Objekt automatisch entladen.

#### Bestimmung der Reihenfolge der Objektentnahme

Da die Objekte parallel zur X-Achse entnommen werden, lässt sich die Reihenfolge der Entnahme der Objekte anhand der Y-Koordinaten der Objektzentren sehr einfach bestimmen. Dazu werden die Y-Koordinaten einschließlich der Labels der Clusterzentren extrahiert und

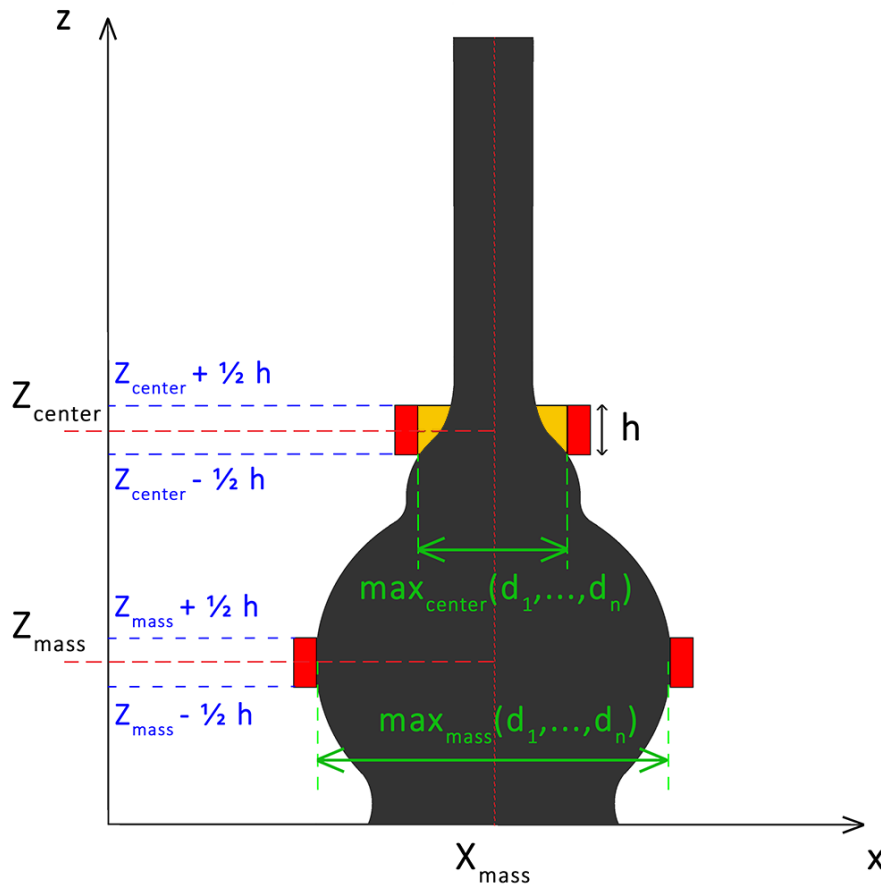


Abbildung 4.13.: Vergleich der Greifpunkte auf  $Z_{center}$  und  $Z_{mass}$ : Der Punkt auf  $Z_{center}$  hat weniger Kontaktpunkte und ist daher weniger stabil. Quelle [189].

in aufsteigender Reihenfolge bezüglich der Y-Koordinaten sortiert. Die daraus resultierende Reihenfolge der Labels ist die Reihenfolge der Extraktion der Objekte.

#### 4.2.4. Erstellung der Objekthypothese

Nach Abschluss der G-Code-Analyse wird die Hypothese berechnet, die durch die Analyse des Punktwolken-Algorithmus weiter verifiziert wird. Für jedes Objekt werden die folgenden Informationen in einer Struktur zusammengeführt:

- $X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{max}$  - Array der Randkoordinaten des Objekts in jeder der drei Dimensionen;
- Koordinaten des Massenschwerpunkts des Objekts in mm - ein Tupel  $(X, Y, Z)_{mass}$ ;
- Entscheidung, ob das Objekt gegriffen werden kann  $can\_grasp$  - *true* oder *false*;
- Koordinaten zum Anfahren mit dem Mittelpunkt-Werkzeug in mm - ein Tupel  $(X, Y, Z)_{Griff}$ ;



#### 4. Automatisierte Entnahme gedruckter Objekte

- Objektbreite auf der Greifebene  $obj\_width$  in mm;
- Schließbreite für den Greifer  $width\_closed$  in mm.

Die Strukturen für jedes erkannte Objekt werden in der Reihenfolge der Extraktion angeordnet und bilden zusammen mit der Anzahl der Objekte und der Steigung der Regressionsgeraden eine Hypothesenmeldung, die an die Kameradatenanalyse weitergegeben wird.

##### 4.2.5. Verifizierung der Objekthypothese auf der Grundlage einer Punktwolke

Die Startposition wird mit der Kamera angefahren. Es wird ein Einzelbild aufgenommen und mit dem Cropbox-Algorithmus [151] verarbeitet, der alle Punkte entfernt, die außerhalb eines festen Rahmens liegen. Außerdem werden die größten flächigen Komponenten, die nicht zu den Zielobjekten gehören, entfernt und alle verbleibenden Punkte geclustert. Die Anzahl der resultierenden Clusterwolken entspricht der gemessenen Anzahl der Objekte auf der Druckplatte. Wenn diese Zahl mit der Zahl aus der Hypothese übereinstimmt, kann mit der Überprüfung der Berührbarkeit und der Entfernung der Objekte fortgefahren werden.

##### 4.2.6. Prüfung der Erfassbarkeit und Entfernung der Objekte

Zur Überprüfung der Greifbarkeit wird für jedes Objekt, für das  $can\_grasp = true$  ist, eine Kameraposition angefahren. Die X-Achse der Kamera ist parallel zur X-Achse der Druckplatte, der Koordinatenursprung der Kamera wird auf die X-Koordinate der Objektmitte geführt. Jedes Objekt sollte einzeln betrachtet werden, da kleinere Objekte im Hintergrund von den größeren Objekten im Vordergrund verdeckt werden können.

Nachdem die Kamera in die Inspektionsposition bewegt wurde, wird wieder ein Einzelbild aufgenommen und wie beschrieben verarbeitet. Weiterhin wird nur die Clusterwolke berücksichtigt, die die Punkte enthält, deren X- und Z-Koordinaten im Bereich der Objektgrenzpunkte liegen.

Es kann vorkommen, dass auf der vorgeschlagenen Greifebene feine Details hervorstehen, die beim Greifen möglicherweise brechen könnten. Um dies zu vermeiden, sollte die Punktwolke auf das Vorhandensein solcher Details untersucht werden. Experimentell wurde festgestellt, dass solche feinen Details nicht in der Punktwolke erscheinen. Sie erscheinen als Reflexionen am Rande der ursprünglichen Punktwolke und werden daher bereits beim Filtern entfernt.

Zunächst wird die Breite  $d$  des Objekts zwischen  $Z_{grip} - \frac{h}{2}$  und  $Z_{grip} + \frac{h}{2}$  aus der Clusterwolke berechnet. Aufgrund der Unvollständigkeit der Punktwolke wird diese Zahl von der Zahl aus der G-Code-Analyse abweichen. Wenn  $d \geq 0,8 \cdot obj\_width$ , kann das Objekt wie in der Hypothese vorgeschlagen erfasst werden. Wenn  $d < 0,8 \cdot obj\_width$ , wird die Greifebene um eine Greiferfingerbreite nach unten oder oben verschoben, wenn die Möglichkeit besteht. Dies ist notwendig, da der Greifer sonst beim Schließen nicht in Kontakt kommt. Die Objektbreite

#### 4. Automatisierte Entnahme gedruckter Objekte

für die neue Greifposition wird aus der Clusterwolke ermittelt. Wenn das Objekt auf der neuen Greifebene gegriffen werden kann, wird es entnommen und an den Ablageort gelegt. Der Vorgang wird für jedes Objekt wiederholt. Konnten alle Objekte entnommen werden, wird geprüft, ob die Druckplatte leer ist, und dann erhält der Drucker das Signal, dass ein neuer Druckauftrag gestartet werden kann.

#### 4.2.7. Experimente und Ergebnisse

Zunächst wurden mehrere Tests durchgeführt, um die Funktion des G-Code-Analysemoduls zu überprüfen. Die Anzahl der Objekte in jeder G-Code-Datei lag zwischen 1 und 25, und die Objektgrößen lagen zwischen 10 mm x 10 mm und 80 mm x 80 mm (Breite x Tiefe). Die Objekthöhen lagen zwischen 10 mm und 150 mm. In den folgenden Tabellen sind die Ergebnisse einiger Versuchsläufe dargestellt, siehe Tabelle 4.2.

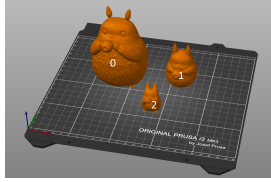
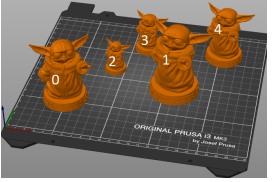
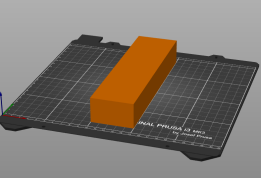
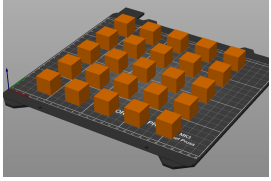
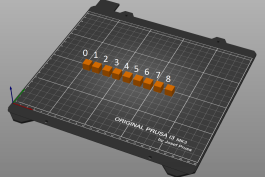
			
<b>Druckjob</b>			
<b>Anzahl der Cluster</b>	3	5	1
<b>Gradient der Regressionslinie</b>	-0,05070884	0,48840319	0,00567
<b>Entnahme erfolgreich</b>	Ja	Ja	Ja
			
<b>Druckjob</b>			
<b>Anzahl der Cluster</b>	25	4	
<b>Gradient der Regressionslinie</b>	0,003656	-	
<b>Entnahme erfolgreich</b>	Ja	Nein	

Tabelle 4.2.: Testdateien mit mehreren Objekten und unterschiedlichen Mengen an Objekten. Die Tests 1 bis 4 waren erfolgreich, während Test 5 fehlschlug, da der Abstand zwischen den Objekten zu gering war. Quelle [189].

#### 4. Automatisierte Entnahme gedruckter Objekte

Testdatei #1 enthält 3 Objekte mit Freiformen unterschiedlicher Größe, die auf der  $(X, Y)$ -Ebene einen Mindestabstand von 20 mm zueinander haben. Die Reihenfolge der Extraktion wurde mit  $[2, 0, 1]$  berechnet und beschreibt die Beschriftung der Objekte. Die Daten, die die G-Code-Analyse ergeben hatte, sind in Tabelle 4.2 aufgeführt. Für das Objekt mit der Bezeichnung 2 liegt der Massenschwerpunkt bei  $Z = 14,45$  mm, was unter dem Schwellenwert von 20 mm liegt. Die Höhe des Objekts beträgt jedoch 35,15 mm und liegt damit über dem Schwellenwert von 20 mm, sodass das Objekt doch gepickt werden kann. Es sollte bei  $Z = 20$  und nicht bei  $Z = 14,45$  mm aufgenommen werden. Die Analyse erkennt dies. Für das Objekt mit dem Label 2 wurde schließlich  $(X, Y, Z)_{center} = (134.83, 72.21, 14.45)$  mm,  $(X, Y, Z)_{grip} = (134.83, 72.21, 20)$  mm berechnet.

Die Testdatei #2 enthält 5 Objekte, die alle das gleiche Modell haben, aber in unterschiedlichen Größen in die STL-Datei eingefügt wurden. Auf der  $(X, Y)$ -Ebene wird ebenfalls der Mindestabstand von 20 mm eingehalten. Die Reihenfolge der Entfernung wurde mit  $[0, 1, 2, 3, 4]$  berechnet.

Die Testdatei #3 enthält ein Objekt mit einer einfachen Form (Block), dessen Tiefe viel größer ist als seine Breite, sodass der Abstand zwischen  $y_{center}$  und  $y_{min}$  größer ist als die Länge der Greiferbacken. Für das Objekt  $75 \text{ mm} < (y_{Zentrum} - y_{min} = 102 \text{ mm})$  und  $z_{Zentrum} < 20 \text{ mm}$ , daher werden sowohl  $y_{grip}$  als auch  $z_{grip}$  angepasst.

Die Testdatei #4 enthält 25 gleiche Würfel mit  $20 \text{ mm} \times 20 \text{ mm}$ , die in 5 Reihen zu je 5 Würfeln mit einem Mindestabstand von 20 mm angeordnet sind. Alle 25 Objekte wurden von der G-Code-Analyse erkannt. Sie sollten bei  $z_{grip} = 20 \text{ mm}$  gegriffen werden.

Testdatei #5 enthält 9 Würfel:  $10 \text{ mm} \times 10 \text{ mm} \times 10 \text{ mm}$ , die in einer Reihe mit einem Abstand von etwa 5 mm angeordnet sind. Die G-Code-Analyse erkennt nur 4 Objekte und fasst mehrere Würfel zu einem Objekt zusammen. Das falsche Ergebnis des Clustering führt zu einer falschen Hypothese. Um hier zu einem korrekten Ergebnis zu kommen, müssten die Parameter für das Clustering angepasst werden oder ein anderer Algorithmus eingesetzt werden.

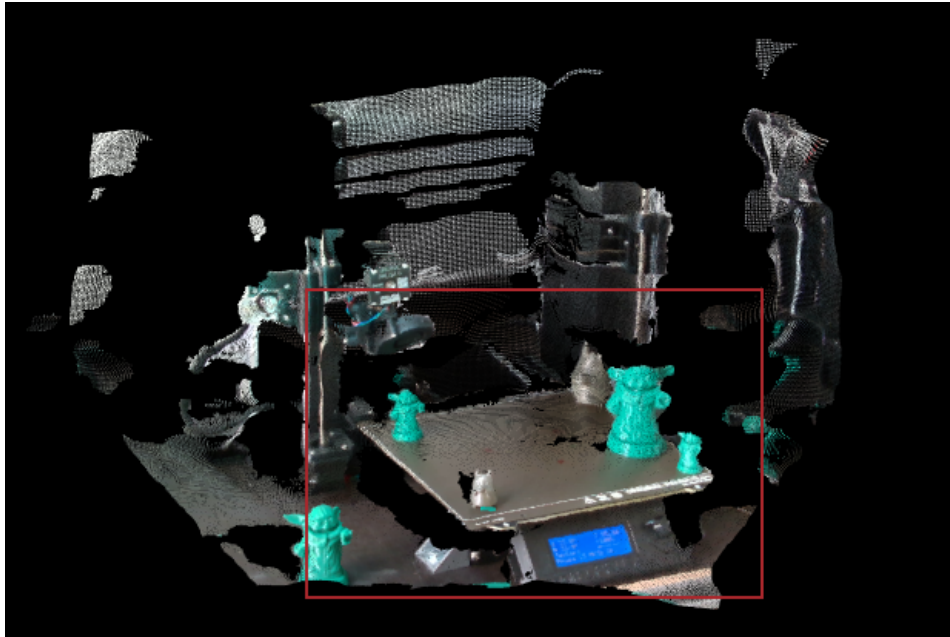
### 4.2.8. Visuelle Datenanalyse

Die Überprüfung der Hypothese sowie die Greifbarkeit wurden an mehreren Objektsätzen getestet. Die Testläufe wurden mit Objektsätzen durchgeführt, deren Anordnung auf der Druckplatte unterschiedliche Ausgangspositionen erfordert. Abbildung 4.14 zeigt die Bilder von drei Testszenen und die entsprechenden gefilterten Punktwolken. In allen Tests wurde die Anzahl der Objekte erfolgreich erkannt.

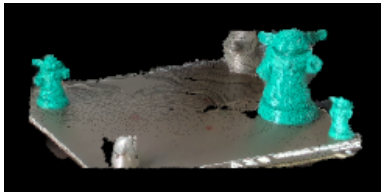
#### Untersuchung der Greifbarkeit

In diesem Test wurde das Objekt #1 aus dem Test #1 gedruckt und die aus der G-Code-Analyse bekannten Informationen wurden sowohl an den Roboter als auch an den CountObjects-Dienst übergeben. Die Testposition wurde angefahren und der Dienst wurde aufgerufen. Zur

#### 4. Automatisierte Entnahme gedruckter Objekte



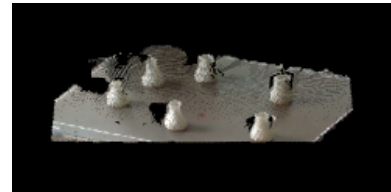
(a) Original Punktwolke mit eingezeichneter Region of Interest (roter Rahmen).



(b) 5 erkannte Objekte aus Abb. 4.14a.



(c) 9 erkannte Objekte aus einem weiteren Druck.



(d) 6 erkannte Objekte aus einem weiteren Druck.

Abbildung 4.14.: Die Punktwolke wird verarbeitet und an Hand von Hypothesen die darin enthaltene Anzahl von Objekten bestimmt. Quelle [189].

Verdeutlichung wurden die resultierenden Punktwolken gespeichert und visualisiert. Auf der mit CropBox gefilterten Wolke, siehe Abbildung 4.15a, ist leicht zu erkennen, dass die dünnen, feinen Teile des Objekts aufgrund der Einschränkungen des Sensors nicht in der Punktwolke zu finden sind. Anschließend wurde die ebene Komponente der Punktwolke entfernt und das Objekt als Clusterwolke dargestellt, siehe Abbildung 4.15b. Die durch die G-Code-Analyse berechneten Koordinaten  $x_{tcp}$  und  $y_{tcp}$  wurden verwendet, um festzustellen, ob der Cluster zum aktuell ausgewählten Objekt gehörte. Die Clusterwolke wurde dann beschnitten, siehe Abbildung 4.15c, und die Breite wurde berechnet.

Die durch die G-Code-Analyse berechnete Breite betrug 40 mm. Die in der Punktwolke gemessene Breite betrug 39 mm, was 97,5 % des Wertes aus der G-Code-Analyse entspricht. Das Objekt kann also wie in der Hypothese berechnet gegriffen werden.

#### 4. Automatisierte Entnahme gedruckter Objekte



(a) Punktwolke gefiltert mit CropBox.



(b) Objekt aus der Punktwolke segmentiert.



(c) Bereich zur Untersuchung der Greifbarkeit.

Abbildung 4.15.: Die Punktwolke des Objekts wird in mehreren Schritten vorverarbeitet, um die bestmögliche Eingabe für den Greifbarkeitsalgorithmus zu erhalten. Hierbei wird final der Ausschnitt gesucht, in dem der Griffpunkt liegt, um hierfür die Griffbreite zu bestimmen. Quelle [189].

### 4.2.9. Test des Gesamtsystems

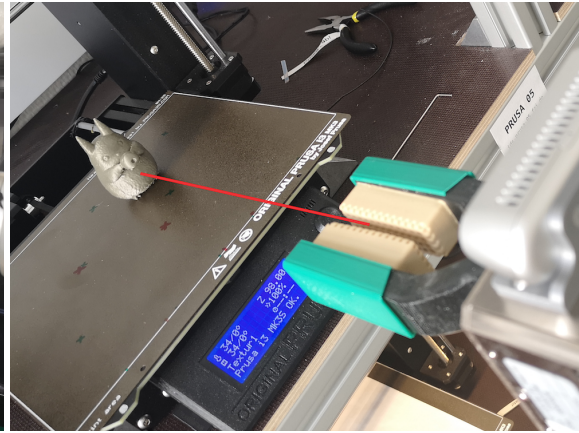
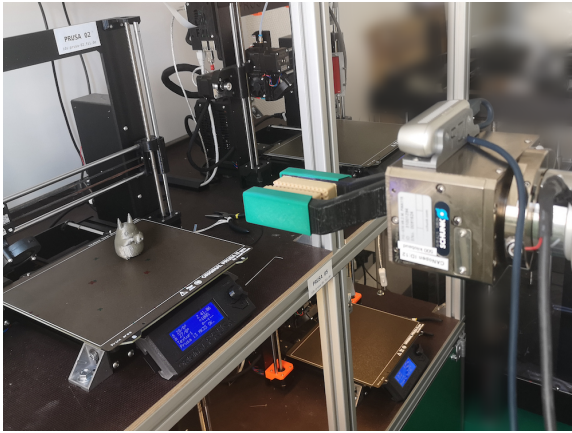
Um den gesamten Ansatz zu evaluieren, wurden alle vorgestellten Methoden zusammen getestet. Nach erfolgreicher Generierung der Ausgangshypothese wurde diese mit der Punktwolke verifiziert und der berechnete Greifpunkt an den Greifmechanismus übergeben. Dieser bewegte den Roboter mit der richtigen Greiferöffnungsweite an den richtigen Punkt und schloss den Greifer. Anschließend wurde das Objekt an der Ablageposition platziert. Die Tests waren erfolgreich und alle Objekte wurden ohne Beschädigung entnommen. Abbildung 4.16 stellt verschiedene Schritte des gesamten Systemtests dar. Die Verifikationsschritte werden vor dem Greifen jedes Objekts durchgeführt.

## 4.3. Fazit

Der Maschinencode des Druckers, der G-Code, enthält alle wichtigen Informationen um zu berechnen, an welcher Stelle sich wo im Bauraum welches Objekt befindet. Während des Druckvorgangs kann die finale Breite des Objekts bestimmt werden. Sind alle Objekte bestimmt können sie nun in einer ebenfalls berechneten Reihenfolge entnommen werden. Hierfür kommt der entwickelte Zweibackengreifer zum Einsatz, der sowohl über den Formschluss dank der elastischen Greiferoberfläche, als auch über Kraftschluss, das Objekt sicher entnehmen kann. Die Kombination aus Analyse der Druckdaten, sowie dem entwickelten Robotergreifer ermöglicht die automatische Entnahme gedruckter Objekte direkt aus dem 3D-Drucker. Der vorgestellte flexible Ansatz ist der erste Schritt zu einer wirtschaftlichen Handhabung von gedruckten Objekten von der Druckplatte und reduziert die nicht wertschöpfende Stillstandszeit des 3D-Drucks.

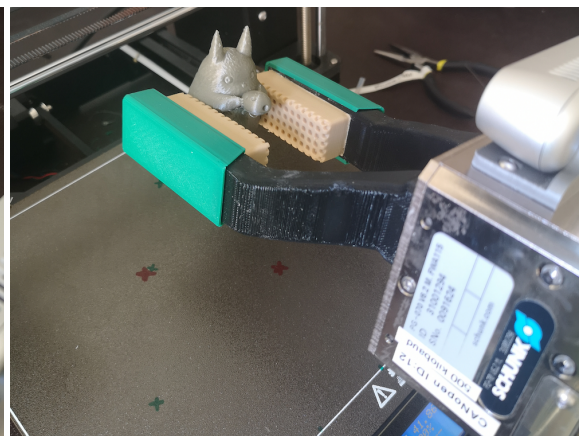
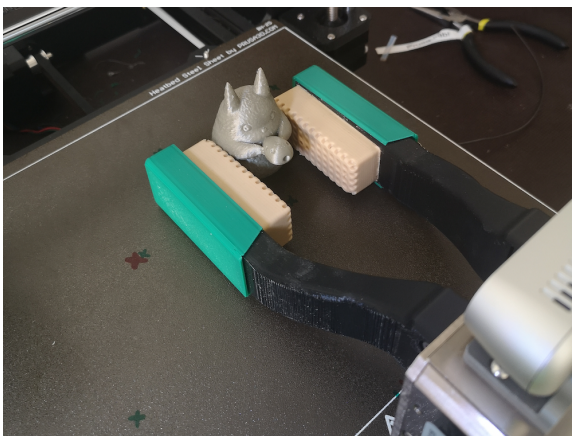
Es gibt jedoch Einschränkungen, was die minimale und maximale Größe des Objektes betrifft. Diese können aber zum Beispiel durch ein Greiferwechselsystem angegangen werden, damit auch zu kleine oder zu große Objekte gegriffen werden können.

#### 4. Automatisierte Entnahme gedruckter Objekte



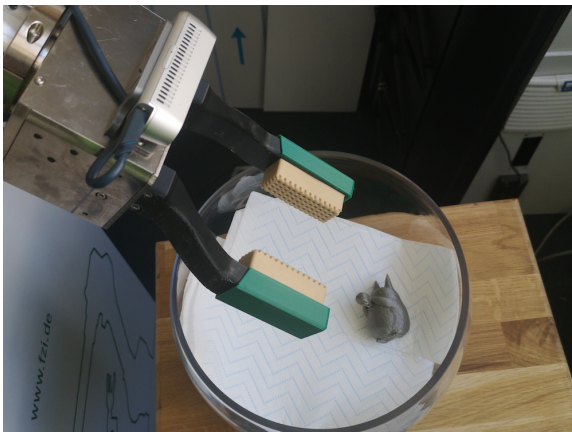
(a) Zuerst wird der Roboter zum Drucker bewegt, bevor die berechnete Kameraposition eingenommen wird.

(b) Sobald der Roboter die berechnete Kameraposition erreicht hat, wird die Greifweite aus dem Kamerabild berechnet.



(c) Das Objekt wird mit der berechneten Griffbreite gegriffen, um Beschädigungen zu vermeiden.

(d) Das Objekt wird durch Drehen von der Druckplatte gelöst, um weder sich selbst, das Objekt oder die Druckplatte zu beschädigen.



(e) Bewegen des Roboters in die Ablageposition und Öffnen des Greifers zur Freigabe des Objekts.

(f) Das Objekt wurde erfolgreich von der Druckplatte entfernt und der Roboter wartet auf den nächsten Auftrag.

Abbildung 4.16.: Bildsequenz der verschiedenen Zwischenschritte des vorgestellten Ansatzes zur automatischen Entnahme gedruckter Objekte. Quelle [189].

# 5. Post-Processing der gedruckten Bauteile

## 5.1. Nachbearbeitungswerkzeug am Roboter TCP

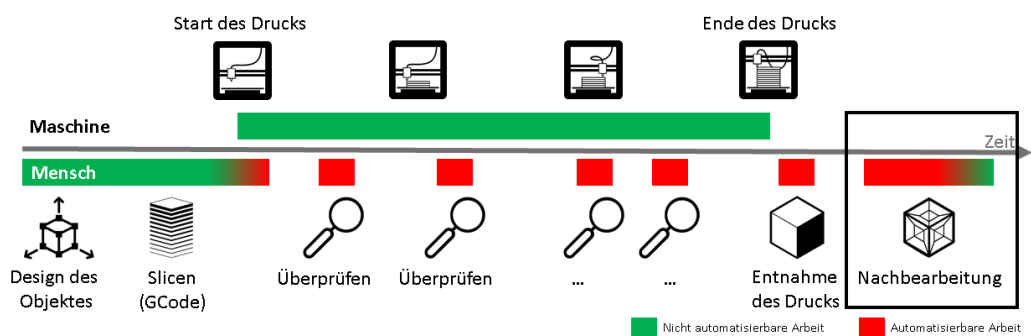


Abbildung 5.1.: In diesem Abschnitt der Arbeit liegt der Fokus auf der Nachbearbeitung gefertigter Objekte (schwarzer Kasten). Prozesstechnisch notwendige Stützstrukturen müssen nach dem Druck entfernt werden, um ein funktionales Bauteil zu bekommen.

3D-Druck, Fräsverfahren oder Spritzguss haben alle gemeinsam, dass das Objekt nach der Herstellung durch manuelle Arbeit überprüft und nachbearbeitet werden muss. Dies ist notwendig, um unerwünschte Teile wie Kanten oder Stützstrukturen aus den Fertigungsprozessen zu entfernen. Es gibt bereits Methoden diese Nachbearbeitung zu automatisieren, allerdings ist hier ein initial sehr hoher Aufwand nötig, weswegen es sich vor allem für Bauteile mit hoher Stückzahl eignet. Doch bei kleinen Stückzahlen, schnellen und flexiblen Produkthanpassungen oder gar einmalig hergestellten Objekten übersteigt der Aufwand, einem Roboter eine bestimmte Nachbearbeitungsbahn beizubringen, den Nutzen [152]. Deshalb wird diese Aufgabe trotz der heutigen Möglichkeiten immer noch manuell erledigt.

Dieser Ansatz beschäftigt sich mit der automatischen Generierung von Trajektorien für Roboter, die allein auf dem Unterschied zwischen den ursprünglichen computergestützten Designdaten (STL) und dem Maschinencode (G-Code) basieren. Die beiden Eingabedateien werden verwendet, um den Unterschied zwischen dem entworfenen und dem letztendlich produzierten Objekt zu berechnen, und auf der Grundlage dieser Abweichung wird eine Trajektorie für die



## 5. Post-Processing der gedruckten Bauteile

Nachbearbeitung generiert. Beim 3D-Druck müssen einige Teile von unten gestützt werden, damit sie ohne Fehler gedruckt werden können. Diese Strukturen haben nur den Zweck, eine Verformung im Druckprozess der oberen Schichten durch die Schwerkraft zu verhindern. Diese sogenannten Stützstrukturen sind zerbrechlich und lassen sich anschließend leicht entfernen.

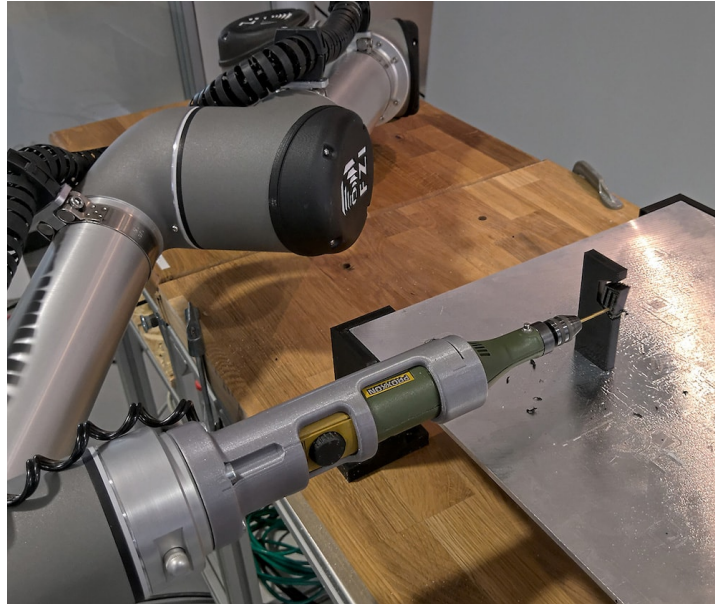


Abbildung 5.2.: Der Roboterarm führt die berechnete Trajektorie aus, um die Support Struktur vom eigentlichen Bauteil zu trennen. Hierbei steht das gedruckte Objekt auf der Druckplatte, die wiederum in eine vorher kalibrierte Position gebracht wurde. Quelle [185].

Nachdem der Druck abgeschlossen und das Material abgekühlt ist, müssen diese zusätzlichen Strukturen entfernt werden, um das ursprünglich gewünschte Objekt zu erhalten. Dieser Entfernungsprozess wird heutzutage immer noch manuell von Mitarbeitern während der Nachbearbeitung durchgeführt, was sowohl zeitaufwändig als auch gesundheitlich bedenklich ist.

Der Ansatz dieser Arbeit besteht darin, diesen Prozess zu automatisieren und nur vorhandene Daten zu verwenden, um eine Robotertrajektorie zur Ausführung dieser Aufgabe zu erstellen. Da das Framework auf dem Robot Operating System (ROS) [52] sowie der Open Motion Planning Library (OMPL) [153] basiert, kann es ohne großen Aufwand in industriellen Anwendungen eingesetzt werden. Zur Durchführung des Experiments verwendet das Framework die CAD-Daten und den Maschinencode des gedruckten Bauteils, um die Differenz zwischen diesen beiden Datensätzen zu berechnen. Eine Voxel-Datenstruktur stellt sicher, dass jede Art von dreidimensionalem Objekt dargestellt werden kann und die Auflösung je nach Anwendungsfall angepasst werden kann. Die gesamte Voxelverarbeitung und -berechnung wird von GPU-Voxels [154] übernommen, einem Open-Source-Framework für die Echtzeit-Datenverarbeitung auf Grafikprozessoren. Das eigentliche Objekt darf während der gesamten Ausführung der Trajektorie keinen Schaden nehmen. Deshalb verwendet der Bahnplaner eine Voxel-Darstellung

## 5. Post-Processing der gedruckten Bauteile

des Roboters und des Werkzeugs sowie eine vereinfachte Umgebung, um eine kollisionsfreie Bahn zu planen. Ein Kontakt mit der Stützstruktur des Objekts ist hier erwünscht, insbesondere während des Fräsvorgangs. Der Kontakt zwischen Werkzeug und Objekt ist jedoch nur innerhalb der Stützstrukturen zulässig. Daher werden die verschiedenen Strukturen bei der Bahnplanung unterschiedlich behandelt. Während der Ausführung der generierten Trajektorie wird die Voxeldarstellung basierend auf den aktuellen Gelenkwerten des Roboters und der daraus resultierenden Werkzeugposition aktualisiert. Um eine hohe Erreichbarkeit zu gewährleisten, werden verschiedene Orientierungen des robotermontierten Werkzeugs im Objektraum abgetastet. Die Orientierung, die die meisten Zielpunkte erreichen kann, wird verwendet, um so viel Material wie möglich zu entfernen. Bleibt nach einer ersten Iteration noch unerwünschtes Material übrig, wird der Prozess wiederholt und eine neue Orientierung errechnet. Durch die aktualisierte Hindernisdarstellung können mit jeder Iteration neue Lösungen ermöglicht werden. In Abbildung 5.2 ist der Roboter gerade dabei die Stützstrukturen aus einem gedruckten Objekt zu entfernen.

Abbildung 5.3 zeigt eine vereinfachte Architektur, die die wichtigsten Komponenten des Software-Stacks hervorhebt. Die erste Komponente ist die Objektsegmentierung, die aus den eingehenden Daten wie STL- und G-Code-Dateien oder sogar Punktwolken-Sensordaten eine Voxeldarstellung erzeugt. Das flexible Framework ist in der Lage, weitere Arten von CAD- oder Sensordaten zu verarbeiten, aber die erforderlichen Konverter sind derzeit noch nicht implementiert.

Die Objektdarstellung in Form mehrerer Voxelsätze wird von der Bahnplanung verwendet, um die gewünschten Post-Processing-Trajektorien zu erzeugen. Eine Visualisierung einer geparteten G-Code-Datei ist in Abbildung 5.4 zu sehen. Diese Art der Datendarstellung wurde gewählt, da sie der Darstellung der realen Welt sehr nahe kommt. Diese wird für die Kollisionserkennung bei der Planung der Robotertrajektorien benötigt, und die Datenstruktur ist außerdem in der Lage, Operationen wie das Hinzufügen und Subtrahieren mehrerer Voxel-Ebenen durchzuführen. Während des Materialabtrags wird die Objektdarstellung auf der Grundlage der Position des Werkzeugs aktualisiert. Dazu wird die Abtrennung von Kunststoffspänen durch das Werkzeug simuliert. Die Trajektorie kann mit einem realen Roboter manipulator ausgeführt werden oder indem der gesamte Prozess in einer Simulation abläuft.

GPU-Voxels [154] ist ein Kernstück für die in dieser Arbeit vorgestellte Trajektoriengenerierung, da es zur Modellierung des Arbeitsraums, in dem der Roboter arbeitet, und auch der verschiedenen Darstellungen des 3D-gedruckten Objekts verwendet wird. Es gibt mehrere Datenquellen, die zur Voxel-Umgebung beitragen: der STL-Import, der G-Code-Interpreter, die aktuelle Position des Roboters, die aus seinem Zustand abgeleitet wird, und die geplanten Trajektorien, die durch Sweep-Volumen dargestellt werden. Der OMPL-Planer [153] nutzt die detaillierte Objektdarstellung und den grobkörnigeren Roboterarbeitsbereich, um mögliche Roboterhaltungen und -bewegungen zu validieren. Dabei wird auf Selbstkollisionen und Kollisionen zwischen dem Roboter und seiner Umgebung geprüft. Der Planer ist in ein ROS-Nodelet verpackt, das eine Aktion für die Bewegungsplanung bereitstellt.

Geplante Pfade werden aus Sequenzen von gültigen Roboterposen zusammengesetzt. Um einen solchen Pfad auszuführen, wird das FZI-eigene Framework zur linearen Interpolation von

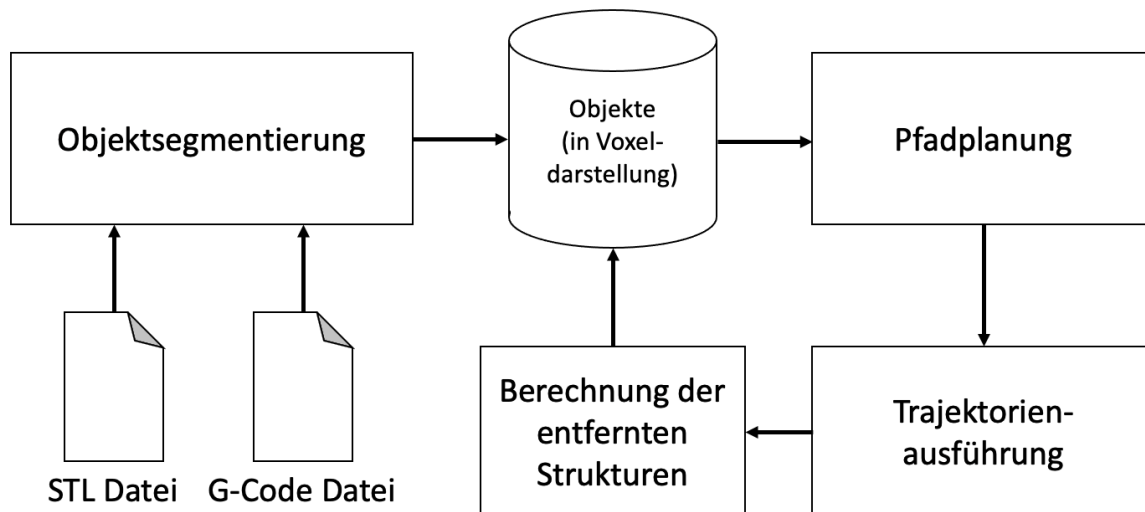


Abbildung 5.3.: Architekturdiagramm des vorgestellten Ansatzes. Die Eingabedaten werden segmentiert und voxelisiert und je nach Anwendungsfall wird die Trajektorie mit integrierter Kollisionsprüfung generiert. Die berechnete Trajektorie kann dann von einem Motion Stack wie der FZI Motion Pipeline oder MoveIt![155] ausgeführt werden. Basierend auf Quelle [185].

Robotertrajektorien, die Motion Pipeline, verwendet. Der inverse Kinematik-Solver TRAC-IK [156] wird verwendet, um kartesische Zielpositionen in Roboterposen zu transformieren. TRAC-IK kombiniert zwei gängige inverse kinematische Solver, um schnelle Lösungen für einfache Probleme zu liefern und gleichzeitig komplexe Probleme zu lösen.

Durch die Verwendung von ROS ist das System für Automatisierung und Wiederverwendung optimiert. Der Benutzer muss zur Laufzeit nicht eingreifen, die Eingabedateien können vorher als Parameter angegeben werden. ROS-Startkonfigurationen sind vorgegeben und bieten die Möglichkeit, verschiedene Parameter wie Build-Größe oder Voxel-Auflösung an einer Stelle anzupassen.

### 5.1.1. Berechnung der Stützstrukturen

Der gängigste Dateityp zur Darstellung von Objekten für den 3D-Druck ist STL. Für diese Arbeit muss ein Verfahren zur Voxelisierung von STL-Dateien entwickelt werden. Um ein Modell in eine GPU-Voxels-Datenstruktur zu importieren, wird die STL-Datei zunächst mit binvox [157, 158] konvertiert, einem Programm, das in der Lage ist, komprimierte binäre Voxel-Gitter aus STL-Dateien zu erstellen. Dieser Schritt ist in einem eigenständigen ROS-Knoten gekapselt, der eine ROS-Aktion zum Analysieren und Konvertieren einer gegebenen STL-Datei bereitstellt. Aus der erstellten binvox-Datei kann ein GPU-Voxels-Voxel-Set erstellt werden.

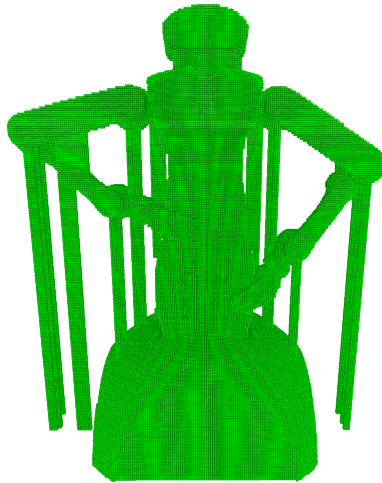


Abbildung 5.4.: Ergebnis der G-Code-Voxelisierung mit dem gewünschten Objekt und der Stützstruktur. Dies ist die Grundlage für die Pfadplanung mit OMPL sowie für die Kollisionsprüfung. Quelle [185].

Um mit den durch STL-Import und G-Code-Interpretation erhaltenen Voxel-Sets arbeiten zu können, müssen die Sets in ihrer Position und Rotation in der GPU-Voxels-Umgebung aufeinander abgestimmt werden. Dies kann automatisch erfolgen, indem die erforderliche Verschiebung aus der bekannten Position der Bounding Boxes des Objekts berechnet wird. Wird das Objekt vor dem Slicing in Bezug auf die STL gedreht, ist diese einfache Methode nicht anwendbar. Drehungen um Vielfache von 90 Grad um die X- oder Z-Achse des Objekts können mit Hilfe von Aufrufargumenten des `binvox`-Programms kompensiert werden. Andere, komplexere Drehungen erfordern fortgeschrittenere Anpassungsstrategien. Algorithmen für das Matching von Punktwolken wie Iterative Closest Point (ICP) [159, 160] könnten angewendet werden.

Zur Erleichterung der Supportentfernung müssen die Voxel in verschiedene Gruppen unterteilt werden. Von besonderem Interesse ist die Unterscheidung zwischen Teilen, die erwünscht sind, und Support Strukturen, die entfernt werden muss. In den folgenden Abschnitten wird die Aufteilung der Voxel in sinnvolle Gruppen erläutert.

Alle durch den STL-Import erzeugten Voxel bilden die Menge *STL*, während alle Voxel aus der G-Code-Interpretation die Menge *GCODE* bilden. `Binvox` voxelisiert eine gegebene STL als eine solide Voxelmenge, daher sind alle Voxel, die die äußere Hülle des Objekts (die Menge *SHELL*) und, falls vorhanden, das Füllmuster (*INFILL*) repräsentieren, enthalten:

$$SHELL \subseteq STL \quad \text{and} \quad INFILL \subseteq STL \quad (5.1)$$

Jedes Voxel aus der G-Code-Interpretation kann entweder als Objekthülle, Füllung oder Support Struktur klassifiziert werden, was zu Gleichung 5.2 führt. Jedes Voxel, das nicht erwünscht ist (nicht Teil der Hülle oder Füllung), aber trotzdem gedruckt wird, wird als Support Struktur

## 5. Post-Processing der gedruckten Bauteile

klassifiziert, wie in Gleichung 5.3 formalisiert, was auch bedeutet, dass das STL-Modell keine Support Struktur enthält, wie in Gleichung 5.4 gezeigt.

$$GCODE = SHELL \cup INFILL \cup SUPPORT \quad (5.2)$$

$$\begin{aligned} SUPPORT &= (GCODE \setminus SHELL) \setminus INFILL \\ &= GCODE \setminus (SHELL \cup INFILL) \end{aligned} \quad (5.3)$$

$$STL \cap SUPPORT = \emptyset \quad (5.4)$$

Aus den Gleichungen 5.2, 5.3 und 5.4 lässt sich schließen, dass eine Menge, die alle Support-Voxel enthält, das relative Komplement von  $STL$  in  $GCODE$  ist, wie in Gleichung 5.5 dargestellt.

$$SUPPORT = GCODE \setminus STL \quad (5.5)$$

In der Realität entstehen Quantisierungsfehler durch eine begrenzte Voxelauflösung und eine begrenzte Druckerauflösung. Die Schichthöhe wirkt als Grenze in Z-Richtung, während die kleinste druckbare Einheit in der XY-Ebene durch den Düsendurchmesser definiert ist. Daher ist es vorteilhaft,  $STL$  vor der Ausführung der oben genannten Gleichung 5.5 zu dilatieren, um die Anzahl der fälschlicherweise als Support klassifizierten Voxel zu reduzieren.  $Q$  in Gleichung 5.6 ist ein strukturierendes Element, das Quantisierungsfehlern entgegenwirken soll.

$$STL' = STL \oplus Q \quad (5.6)$$

Gleichungen 5.1 ist auch für  $STL'$  gültig. Gleichungen 5.4 und anschließend 5.5 sind im Allgemeinen nicht für  $STL'$  gültig, weil Voxel aus  $SUPPORT$ , die nahe an  $STL$  liegen, durch die Dilatation in  $STL'$  eingeschlossen werden. Diese Voxel können durch die in Gleichung 5.7 formulierte Fehlermenge ausgedrückt werden. Eine neue, reduzierte Menge  $SUPPORT'$  kann wie in Gleichung 5.8 definiert und in Abbildung 5.5a visualisiert werden.

$$ERROR = SUPPORT \cap STL' \quad (5.7)$$

$$SUPPORT' = SUPPORT \setminus ERROR = GCODE \setminus STL' \quad (5.8)$$

Experimente mit verschiedenen Objektmodellen zeigen, dass die vorgeschlagene Dilatation der  $STL$ -Menge ein gültiger Ansatz ist, um Quantisierungsfehlern zu begegnen.

Für einen gezielten Abtrag des verbindenden Polymers mit einem robotermontierten Werkzeug ist es von Vorteil, die Kontaktzonen zwischen dem STL-Modell und den Stützen zu segmentieren. Gleichung 5.9 zeigt, wie man diese Kontakte bestimmt, wobei  $N$  eine  $3 \times 3 \times 3$  Nachbarschaft ist und Abbildung 5.5b visualisiert das Ergebnis.

$$CONTACT = SUPPORT' \cap (STL' \oplus N) \quad (5.9)$$

Nachdem die Kontaktzonen extrahiert wurden, ist es von Vorteil, sie in Gruppen von zusammenhängenden Voxeln zu clustern. Auf diese Weise lassen sich die verteilten Kontaktvoxel nach räumlichen Beziehungen und nach der Stützstruktur, zu der sie gehören, segmentieren. Der

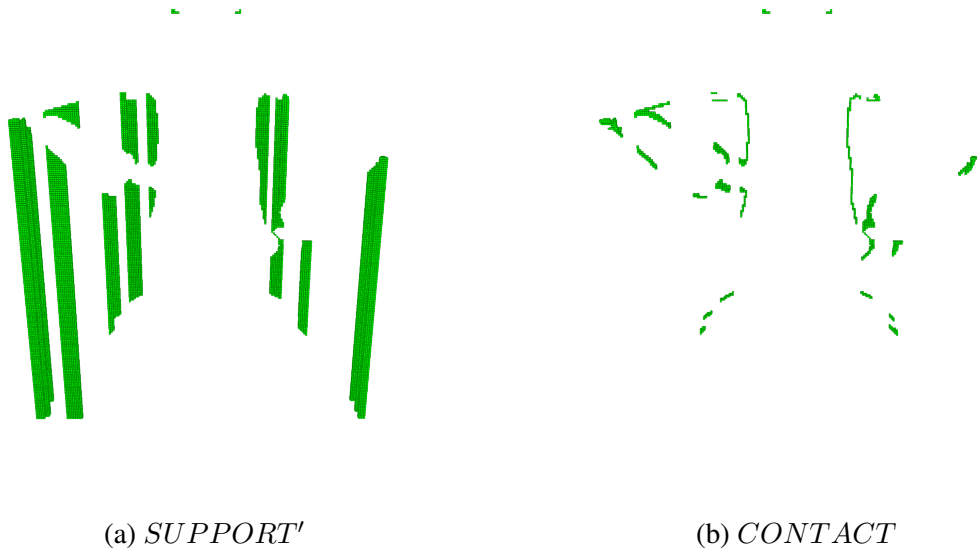


Abbildung 5.5.: Beispiel der Voxelsätze, die durch Gleichungen 5.8 und 5.9 aus den Robotermodell STL- und G-Code-Dateien erzeugt wurden. Quelle [185].

angewandte Clustering-Algorithmus folgt einem iterativen Ansatz des Regionswachstums.  $A$  ist die Voxelmenge, die geclustert werden soll, z.B. die Menge, die alle Kontaktvoxel enthält.  $B_0$  ist eine Menge, die ein beliebiges Voxel  $a$  mit  $a \in A$  enthält.  $B_{n+1}$  wird nach Gleichung 5.10 berechnet. Das strukturierende Element  $S_r$  wird als Kugel mit dem Radius  $r$  gewählt.  $r$  ist die Länge, um die eine Region in einem Schritt wächst und der minimale Abstand zwischen zwei verschiedenen Clustern.

$$B_{n+1} = (B_n \oplus S_r) \cap A \quad (5.10)$$

Ein Cluster ist gefunden, wenn  $B_{n+1} = B_n$ , woraufhin  $B_{n+1}$  aus  $A$  entfernt und der Vorgang mit einem neuen  $B_0$  aus dem nun reduzierten  $A$  wiederholt wird. Alle Cluster sind gefunden, wenn  $A = \emptyset$ .

Abbildung 5.6 zeigt sieben mit dem vorgestellten Algorithmus ermittelte Voxel-Cluster, sowohl allein als auch mit dem aus der STL-Datei generierten Modell.

### 5.1.2. Experimente und Evaluierung

Das System wurde mit einem UR5-Leichtbau-Roboterarm von Universal Robots (A.1.5) und einem Standardbohrer, der entlang der Z-Achse des Roboter-TCPs montiert ist, evaluiert. Der Roboter selbst ist auf einer Werkbank montiert, die auf gleicher Höhe auch Befestigungspunkte für die Bauplatte des 3D-Druckers enthält. Die Platte kann ohne zusätzliche Werkzeuge montiert werden, da die Plattenhalterung bereits im Robotersystem kalibriert ist.

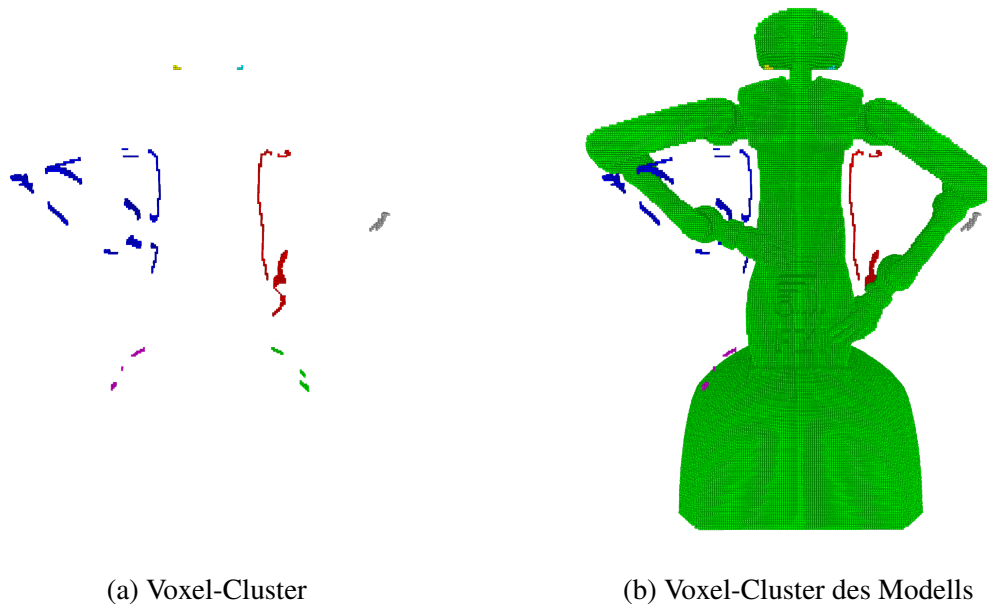


Abbildung 5.6.: Beispiel für Voxelcluster von Stützkontaktregionen in verschiedenen Farben. Quelle [185].

Der gesamte Aufbau kann in der Simulation mit Gazebo [161] zur Vorabkontrolle verwendet werden und ist auch als Hardware-Demonstrator implementiert.

Das vorgestellte System wird zur Nachbearbeitung eines 3D-gedruckten Objekts mit zusammenhängendem Stützvolumen verwendet. Abbildung 5.2 zeigt das Werkzeug in Kontakt mit der Stützstruktur, während die Frästrajektorie ausgeführt wird. Das Ergebnis dieses Prozesses ist in Abbildung 5.7 zusammen mit dem Originalmodell des gewünschten Objekts und einem manuell gefertigten Exemplar dargestellt. Zusätzlich wird ein Teil mit einem erzwungenen Fehler zum Vergleich mit dem unverfälschten Objekt bereitgestellt. Dies kann passieren, wenn das Werkzeug falsch ausgerichtet ist und die Seite des Bohrers das Objekt berührt, während die Bohrspitze zu den nächstgelegenen Punkten auf der inneren Rückseite fährt. Ein solcher Bearbeitungsfehler kann ein Indikator für eine fehlerhafte Kalibrierung sein.

Um die Vielseitigkeit des Ansatzes zu zeigen, wurden mehrere unterschiedliche Objekte mit jeweils eigenen Besonderheiten nachbearbeitet. Abbildung 5.8 zeigt zum Beispiel einen Trockenlauf in der Simulation. Dieser kann dazu verwendet werden, die geplante Trajektorie ohne Beschädigung zu validieren. Besonders bei schwierigen Objekten mit mehreren Clustern oder mit Abmessungen, die den Roboter an seine Gelenkgrenzen zwingen, kann dies hilfreich sein, um Schäden zu vermeiden. Der Algorithmus plant die Trajektorie so, dass die Kontaktflächen der Stützstrukturen zuerst entfernt werden, um die Zeit für die Nachbearbeitung zu reduzieren. Wenn sich die Stützstruktur direkt zwischen einem Teil der Bauplatte und dem Objekt befindet, wird der Roboter nicht versuchen, die untere Befestigung an der Bauplatte zu entfernen. Nur die an den Objekten befestigten Strukturen werden entfernt, sodass das gedruckte Objekt weiterhin von den unerwünschten Strukturen befreit ist.

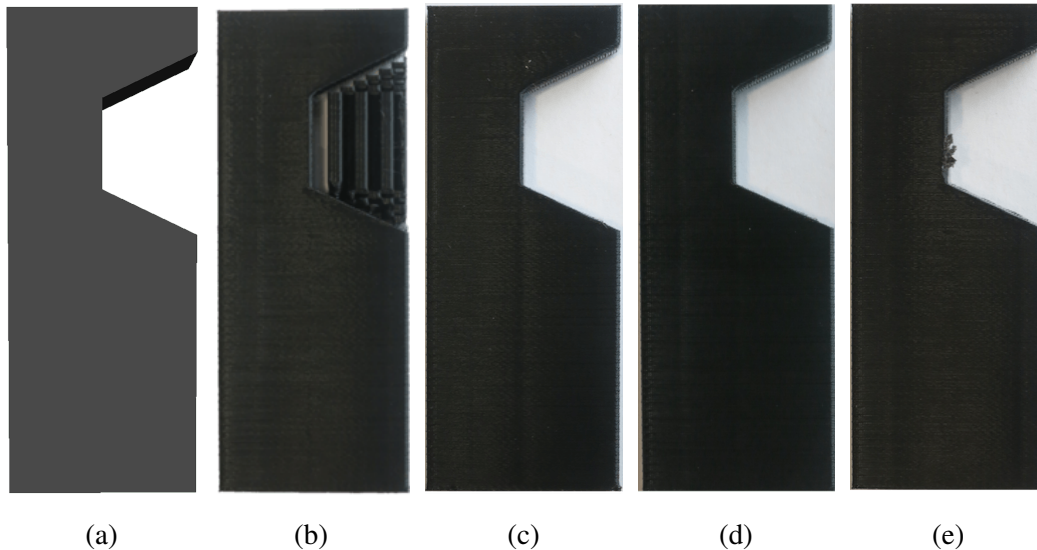


Abbildung 5.7.: Das Beispielobjekt als STL-Modell (a), gedruckt mit Support Struktur (b), nachbearbeitet durch den Roboter (c) und einen Menschen (d) und mit einem möglichen Bearbeitungsfehler (e). Quelle [185].

In Abbildung 5.8 ist der untere Teil der Struktur bereits entfernt worden, diese Voxel sind nicht mehr sichtbar. Die Voxel werden laufend aktualisiert, sodass dieser Freiraum in weiteren Prozessschritten von der Pfadplanung genutzt werden kann.

Das vorgeschlagene System ist in der Lage, Trajektorien zu erstellen, die mehrere disjunkte Stützstrukturen durch Clustering und iterative Verarbeitung entfernen. Die erzeugte Trajektorie führt Schnitte ohne Kollisionen aus. Ein komplexeres Beispiel ist die Entfernung der Stützstrukturen von einem kleinen humanoiden Roboterobjekt. Die Herausforderung liegt hier in der zerbrechlichen Struktur des Objekts und den zahlreichen Stützstrukturen, die unten an der Bauplatte und teilweise am Objekt selbst befestigt sind. Das nachbearbeitete Objekt ist in Abbildung 5.9 dargestellt.

Bei dieser robotergestützten Bearbeitung gibt es keine direkte Zeitersparnis gegenüber der manuellen Arbeit. Ein Vorteil besteht darin, dass das Robotersystem sowohl tagsüber als auch nachts arbeiten kann, ohne dass eine menschliche Aufsicht erforderlich ist. Dies und die geringeren Kosten im Vergleich zu einer menschlichen Arbeitskraft können auf lange Sicht die längere Bearbeitungsdauer ausgleichen.

Die Flexibilität der vorgestellten Lösung kann auch an einem anderen Anwendungsfall integriert werden. Ein Logo wird in eine Polystyrol-Schaumstoffkachel geätzt, wobei dieselbe Pipeline wie oben beschrieben verwendet wird. Die gewünschte Kachel mit dem Logo wird anstelle der STL-Datei voxelisiert und die aus dem G-Code generierte Voxelmenge durch eine aus einem Modell der unbearbeiteten Kachel ersetzt. Es reichte aus, die Eingabedaten entsprechend anzupassen. Eine Änderung des Quellcodes war nicht erforderlich. Es werden drei unabhängige Cluster des zu entfernenden Materials extrahiert, wie in Abbildung 5.10a dargestellt. Die Trajektorie wird vom Roboter ausgeführt und das Endprodukt ist in Abbildung 5.10b zu sehen.



## 5. Post-Processing der gedruckten Bauteile

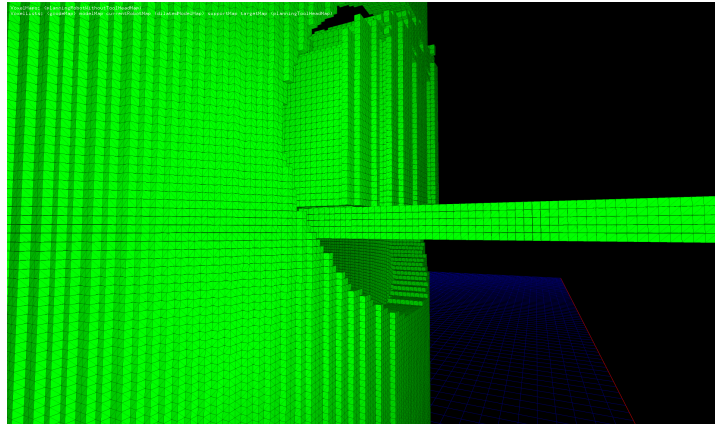


Abbildung 5.8.: Simulieren Sie die Entfernung der Stützstruktur im Inneren eines zylindrischen Objekts. Der untere Teil der Stützstruktur wurde bereits entfernt und die Voxeldarstellung entsprechend aktualisiert. Der obere Teil der Stützstruktur wurde bisher noch nicht vom Werkzeug angefahren und ist daher noch in den Daten repräsentiert. Quelle [185].

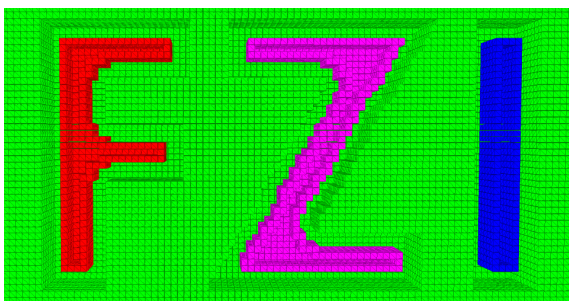
Das System generierte drei Trajektorien, die jeweils einen einzelnen Cluster berücksichtigten, sodass der Rest des Modells nicht beschädigt wurde. Zwischen den Clustern kehrt der Roboter zu einer bestimmten Position zurück, von der aus er sich zum nächsten Cluster bewegt.

Während der Experimente wurde deutlich, dass die Reichweite des Universal Robots UR5 die Art und Größe der verarbeiteten Objekte erheblich einschränkt. Um dieses Problem zu beheben, wird eine Änderung der Hardware vorgeschlagen. Es könnte ein Roboter mit einem größeren Arbeitsbereich verwendet werden, oder die Montage des Roboters auf einer linearen Achse könnte seine Reichweite ebenfalls erhöhen. Schließlich könnte der Arbeitsbereich einschließlich der Befestigungspunkte für die Bauplatte auf einem Drehtisch angeordnet werden, um den gegebenen Arbeitsbereich des Roboters besser auszunutzen.

## 5. Post-Processing der gedruckten Bauteile



Abbildung 5.9.: Das Robotermodell bei der Nachbearbeitung und Entfernung mehrerer Säulen der Stützstruktur aus dem Druck. Die ersten Säulen sind bereits entfernt worden. Quelle [185].



(a) Die Voxeldarstellung einer Materialabtragsbahn. Die verschiedenen vom Roboter zu entfernenden Voxel-Cluster sind rot, violett und blau eingefärbt. Quelle [185].



(b) Das mit dem vorgeschlagenen System gefräste Logo. Eine Anpassung des Quellcodes war nicht erforderlich, es wurden lediglich die Daten des Logos und des Blocks geändert. Quelle [185].

Abbildung 5.10.: Indem die beiden verwendeten Eingangsdaten CAD-Datei und G-Code entsprechend angepasst werden, ist es auch problemlos möglich mit dem Roboter zu Fräsen.

## 5.2. Gedrucktes Objekt am Roboter TCP

In diesem Abschnitt wird der vorgeschlagene Ansatz für die vollautomatische Entfernung von Stützstrukturen von 3D-gedruckten Teilen mit Hilfe eines Roboterarms beschrieben, der das gedruckte Objekt entlang einer festen Frässpindel bewegt. Er ist als zweistufiger Prozess konzipiert, der sich eng an einen normalen 3D-Druckvorgang anlehnt.

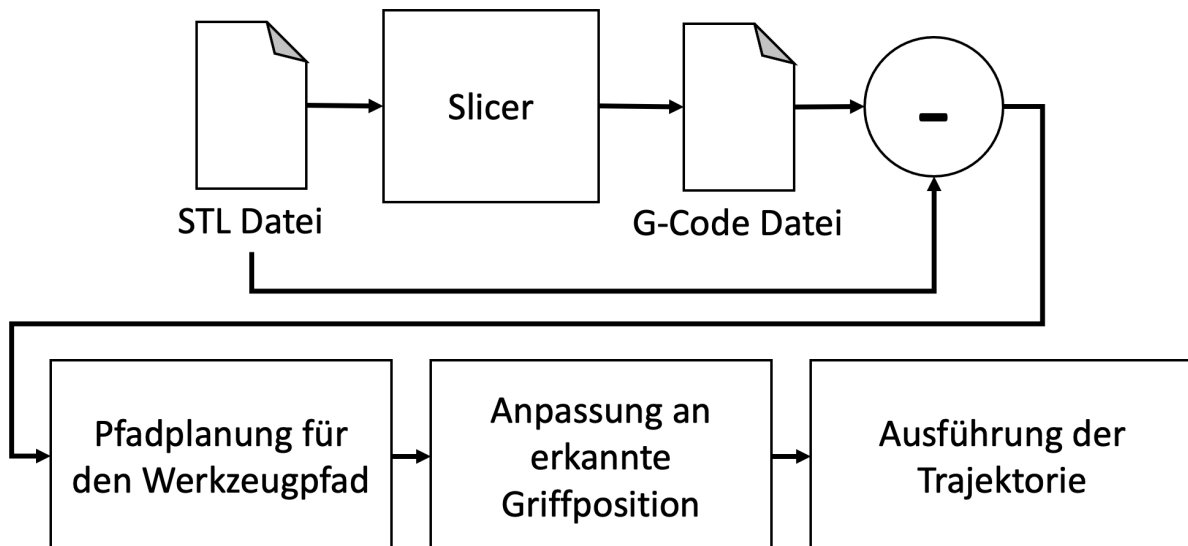


Abbildung 5.11.: Überblick über den vorgeschlagenen Nachbearbeitungsablauf. Nachdem das Teilemodell im CAD konstruiert und mit einem Slicer in G-Code-Maschinenanweisungen übertragen wurde, werden zusätzliche Stützstrukturen identifiziert und Trajektorien geplant und gespeichert. Nach der Fertigung des Teils wird es mit einer Tiefenkamera erfasst und vermessen, bevor die Support Struktur mit einer festen Frässpindel entfernt wird. Basierend auf Quelle [193].

Ein Überblick über den vorgeschlagenen Arbeitsablauf ist in Abbildung 5.11 zu sehen und beginnt mit dem Entwurf des 3D-gedruckten Teils im CAD. Sobald der Nutzer dieses mit Hilfe des Slicers in G-code Maschinenbefehle übertragen hat, werden das resultierende geometrische Modell und der G-code ausgewertet, um ein Volumenmodell aller hinzugefügten Stützstrukturen zu erhalten. Auf dieser Grundlage wird eine Reihe von Zielflächen ausgewählt und für jede davon werden Fräsbahnen erzeugt. Zu diesem Zeitpunkt wird das Objekt bereits gedruckt, was bedeutet, dass dieser Schritt viel Zeit in Anspruch nehmen kann, ohne den Fertigungsprozess zu verzögern.

Sobald der Druckvorgang abgeschlossen ist, wird das gedruckte Objekt von einem Roboterarm gehalten und zur genauen Greiferkalibrierung vor eine Tiefenkamera bewegt. Nachdem der zuvor berechnete Pfad um den Greiffehler korrigiert wurde, wird er ausgeführt, indem der Roboterarm vor eine feststehende Frässpindel bewegt wird.

### 5.2.1. Generierung des Werkzeugpfades

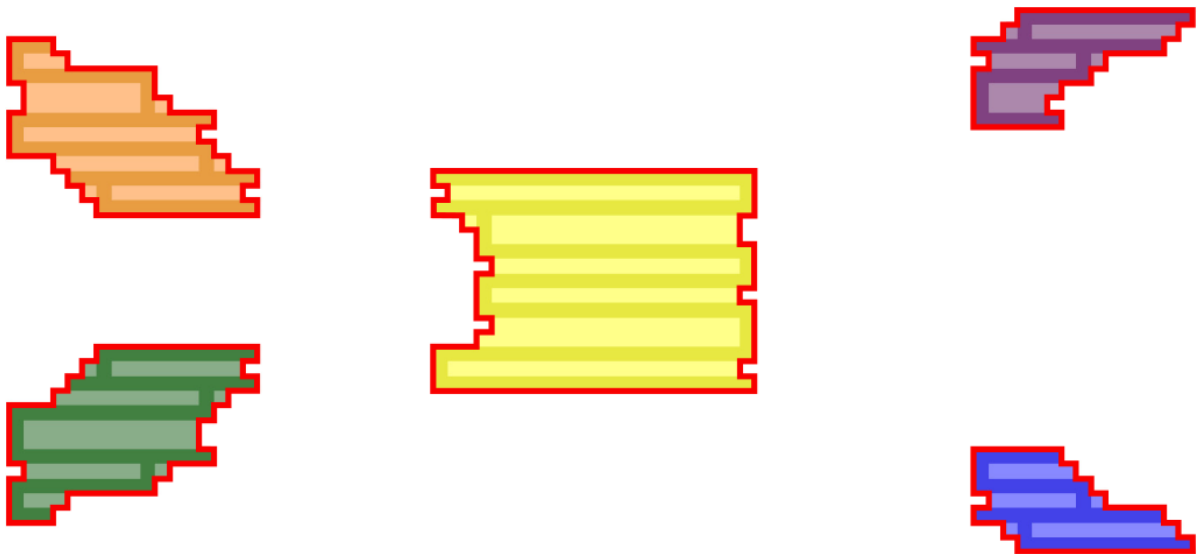


Abbildung 5.12.: Segmentierung einer einzelnen Extrusionsschicht. Die dunkleren Linien sind als Stütze klassifizierte Extrusionen, eine morphologische Schließung dieser ergibt die farbigen Regionen und die roten Grenzen werden über alle Schichten hinweg verbunden, um solide Stützmodelle zu erhalten. Die verschiedenen Farben geben an, dass die entsprechenden Teile zu einer Region gehören. Quelle [193].

Die Offline-Planung der Fräsbahnen beginnt mit der Analyse des CAD-Modells des Werkstücks und der entsprechenden G-Code-Druckeranweisungen. Der G-Code wird gescannt und alle Materialaustritte werden nach ihrer Höhe über dem Druckbett gruppiert. Für jede Schichthöhe wird ein Schnitt durch das CAD-Modell durchgeführt und eine Polygon-Approximation der Modellgrenzen verwendet, um die Extrusionen entweder als Modellmaterial oder als Träger zu klassifizieren. Alle Stützextrusionen werden in eine 2D-Bitmap eingefügt, und es wird eine morphologische Schließoperation durchgeführt, um zusammenhängende Stützregionen zu erhalten. Die Grenzen dieser Regionen werden über alle Modellebenen hinweg gesammelt, und überlappende Polygone benachbarter Ebenen werden miteinander verbunden, um solide Modelle der Stützregionen zu bilden. Ein Beispiel für diesen Prozess ist in Abbildung 5.12 zu sehen.

Die Pfadplanung wird auf den Kontaktflächen zwischen Stützstrukturen und dem Werkstückmodell durchgeführt. Deren Erkennung wurde im Rahmen dieser Arbeit nicht implementiert, sollte aber aus den generierten Stützmodellen gewonnen werden können. Die Planung für jede Oberfläche beginnt mit der Auswahl eines Satzes von Fräser-Kontakt-Pfaden durch isoparametrisches Streichen eines der Oberflächenparameter innerhalb seiner Grenzen. Da in dieser Arbeit nur Werkzeuge mit Kugelkopf betrachtet werden, kann der Werkzeugversatz für diese Bahnen durch einen einfachen Versatz entlang der Oberflächennormale implementiert werden.

## 5. Post-Processing der gedruckten Bauteile

Die Werkzeugausrichtungen werden nun für jeden Punkt entlang dieser Pfade zugewiesen, mit dem Ziel, Konfigurationen zu finden, die kollisionsfrei sind. Hierbei ist zu beachten, dass die Ausrichtung einen Winkel von mindestens  $30^\circ$  zur Oberfläche haben und in einem Winkel von mehr als  $60^\circ$  zum Greifer bleiben soll. Diese Konfigurationen werden ermittelt, indem zunächst in einem großen Kegel von  $60^\circ$  um die Oberflächennormale am Startpunkt der Bahn nach einer brauchbaren Orientierung gesucht wird, wobei Orientierungen nahe der Oberflächennormale bevorzugt werden. Für jeden nachfolgenden Punkt wird dann ein kleinerer Kegel von  $15^\circ$  durchsucht. Wird keine brauchbare Orientierung gefunden, so wird die Bahn an diesem Punkt geteilt und eine verbindende Bewegung eingefügt, bevor wieder auf die Abtastung innerhalb des großen Kegels zurückgegriffen wird. Nachdem eine brauchbare Orientierung für einen Punkt des Pfades gefunden wurde, wird auch ein einzelner Schritt von  $2,5^\circ$  in Richtung der Oberflächennormalen auf Machbarkeit getestet, um elegante Orientierungsänderungen entlang gekrümmter Oberflächen zu ermöglichen.

Nachdem allen Bahnsegmenten Orientierungen zugewiesen wurden, werden sie nun sortiert, indem das nächste Segment mit dem nächstgelegenen Start- oder Endpunkt ausgewählt wird. In einem letzten Schritt wird dann jedes Segment mit Hilfe einer kubischen B-Spline durch die Werkzeugpositionen und eine lineare Interpolation zwischen den entsprechenden Orientierungen interpoliert.

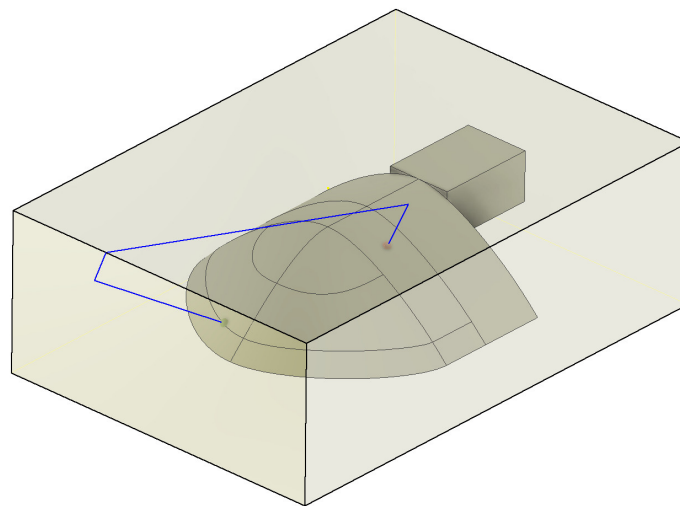


Abbildung 5.13.: Beispiel für eine verbindende Bewegung zwischen zwei Punkten auf einer Objektoberfläche. Der grüne Start- und der rote Endpunkt werden entlang der Werkzeugachse auf die safe box projiziert. Die Bewegung zwischen ihnen wird auf der Oberfläche der safe box ausgeführt. Quelle [193].

Das Ergebnis dieser Berechnungen ist eine Menge von Werkzeugbahnen entlang der Werkstückoberfläche. Um diese zu einer vollständigen Werkzeugbahn zu verbinden, werden nun Bewegungen geplant, die beide Punkte verbinden. Dies geschieht auf ähnliche Weise wie beim Ansatz *safe-box* und ist in Abbildung 5.13 zu sehen. Um mögliche Kollisionen zu vermeiden, werden Verbindungsbewegungen nur auf der Oberfläche eines vergrößerten Begrenzungsrahmens des Modells durchgeführt, und der Vorschub und der Rückzug bewegen sich nur entlang

der Werkzeugachse. Die Bewegung zwischen zwei Punkten wird durch lineare Interpolation auf der Oberfläche der Box erzielt.

Nach Abschluss der Werkzeugbahnplanung werden die Bahnen als G-Code gespeichert.

### 5.2.2. Berechnung der Robotertrajektorie

Nachdem das Werkstück auf dem 3D-Drucker gefertigt wurde, wird es vom Roboterarm gegriffen und vor die Tiefenkamera bewegt und dort positioniert. Das MoveIt-Framework [162] wird verwendet, um sechs verschiedene Posen anzufahren, und die Kamerapunktwolken werden zusammengeführt, um eine genaue Messung der Teilegeometrie zu erhalten. Diese wird nun mit dem CAD-Modell verglichen, indem zunächst gleichmäßig Punkte auf der Modelloberfläche abgetastet werden und dann die gesammelte Punktwolke mit Hilfe eines ICP-Algorithmus (*iterative closest point*) mit diesen abgeglichen wird, was die Transformation zwischen der beobachteten und der geplanten Teileposition ergibt.

Diese Abweichung wird nun in einer generierten Skriptdatei in der robotereigenen Programmiersprache *URscript* verwendet, die zunächst die entsprechenden Werkzeug- und Teiletransformationen relativ zum Roboter definiert und dann den Werkzeugweg lädt. Die Ausführung wird dann mit dem vom Hersteller bereitgestellten Paket *Remote TCP & Toolpath URcap* ausgelöst.

### 5.2.3. Ergebnisse und Evaluierung

Um den vorgeschlagenen Prozess zu evaluieren, wurde ein Hardware-Setup entworfen und hergestellt. Es wird ein Universal Robots UR16e-Roboterarm (A.1.5) mit einem Röhm REPG 2x50-Parallelbackengreifer verwendet, um die gedruckten Objekte zu greifen und zu bewegen. Auf einem Sockel montiert, wird eine stereoskopische Tiefenkamera Roboception rc\_visard 65m für die Greiferkalibrierung verwendet und eine Teknomotor COMC0350060 Frässpindel führt den eigentlichen Fräsvorgang durch. Der vollständige Aufbau ist in Abbildung 5.14 zu sehen.

Die Gesamtauswertung ist in drei Teile gegliedert: Zunächst wird die Ausführung der Werkzeugbahn im Hinblick auf den kinematischen Aufbau und die Bahnsteuerung untersucht. Dann wird die Bahnanpassung und die Basisfräsgenauigkeit getestet und quantifiziert. Abschließend werden komplette Fräsbahnen erzeugt, um die Support Struktur vom Testteil zu entfernen.

### 5.2.4. Ausführung der Trajektorie zur Nachbearbeitung

Die beiden in Abbildung 5.15 gezeigten Testteile wurden zur Bewertung der implementierten Werkzeugbahnausführung verwendet. Die Werkzeugbahnen wurden auf der Kugelspitze der Teile erzeugt und gingen von Werkzeuglängen von 25 mm, 50 mm, 75 mm und 100 mm aus.

## 5. Post-Processing der gedruckten Bauteile

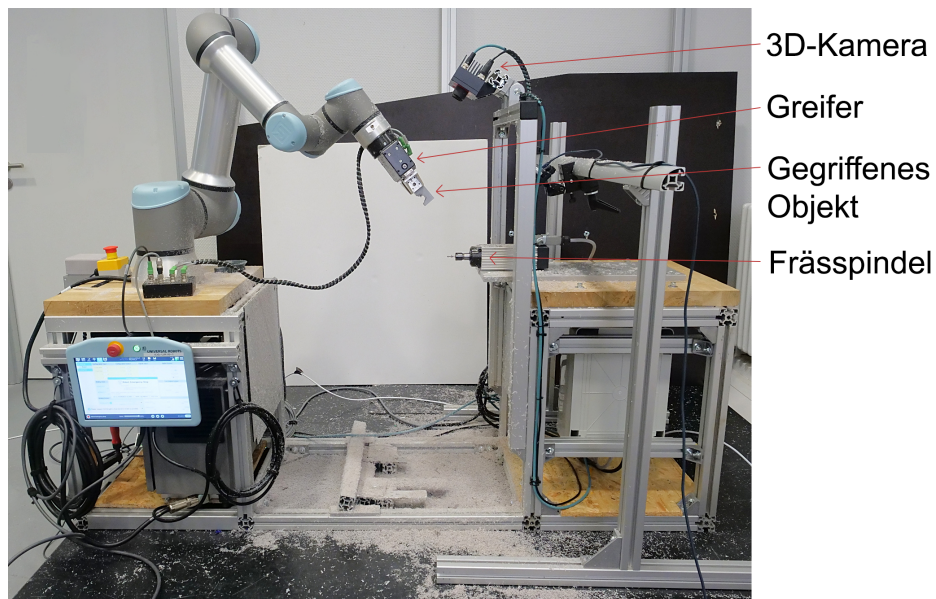


Abbildung 5.14.: Das Hardware-Setup, mit dem das vorgeschlagene Verfahren zum Entfernen von Stützen evaluiert wurde. Der UR16e-Roboterarm auf der linken Seite hält das Werkstück vor der Tiefenkamera, bevor er die Fräsbahn vor der unteren Frässpindel startet. Quelle [193].

Sie wurden dann unter Verwendung der vorgeschlagenen Bahnanpassungs- und Werkzeugbahnausführungsschritte ausgeführt.

Von allen generierten Werkzeugbahnen konnte nur die Konfiguration für das zweite Teil mit einer Werkzeuglänge von 100 mm zuverlässig ausgeführt werden. Alle anderen Konfigurationen scheiterten, indem sie entweder mit dem Frässpindelständer kollidierten oder durch eine Selbstkollision zum Stillstand kamen. Für diese Fehlschläge konnten zwei Hauptursachen identifiziert werden. Erstens wies das implementierte Handgelenkdesign einen großen Abstand zwischen dem letzten Roboterarmgelenk und dem zu bearbeitenden Objekt auf, was zu großen Roboterbewegungen für kleine Orientierungsänderungen und Kollisionen mit Strukturen hinter dem Schneidkopf führte. Zweitens konnte die Steuerung, die zur Ausführung der generierten Trajektorie verwendet wurde, große Orientierungsänderungen nicht adäquat handhaben. Insbesondere wurde der Arm regelmäßig vollständig ausgefahren und drehte sich dann schnell um die Spindelachse, bis er eine Kollision oder eine Gelenkgrenze erreichte. Die wiederholte Ausführung einiger Pfade zeigte auch, dass das Verhalten bei singulären Konfigurationen nicht deterministisch war, die immer wieder an der Vorderseite des Teststücks auftraten, wo das letzte Roboterarmgelenk mit der Spindelachse zusammenfiel. Dies führte zu stark divergierenden weiteren Trajektorien.

Das getestete Teil wurde speziell ausgewählt, um diese Art von Ausführungsproblemen zu zeigen, und weitere Teile, die in dieser Evaluierung verwendet wurden, wiesen diese Probleme nicht so stark auf. Diese Ergebnisse zeigen jedoch, dass ein überarbeiteter Greifaufbau und eine ausgefeiltere Ausführungssteuerung oder Planungsstrategie erforderlich sind.

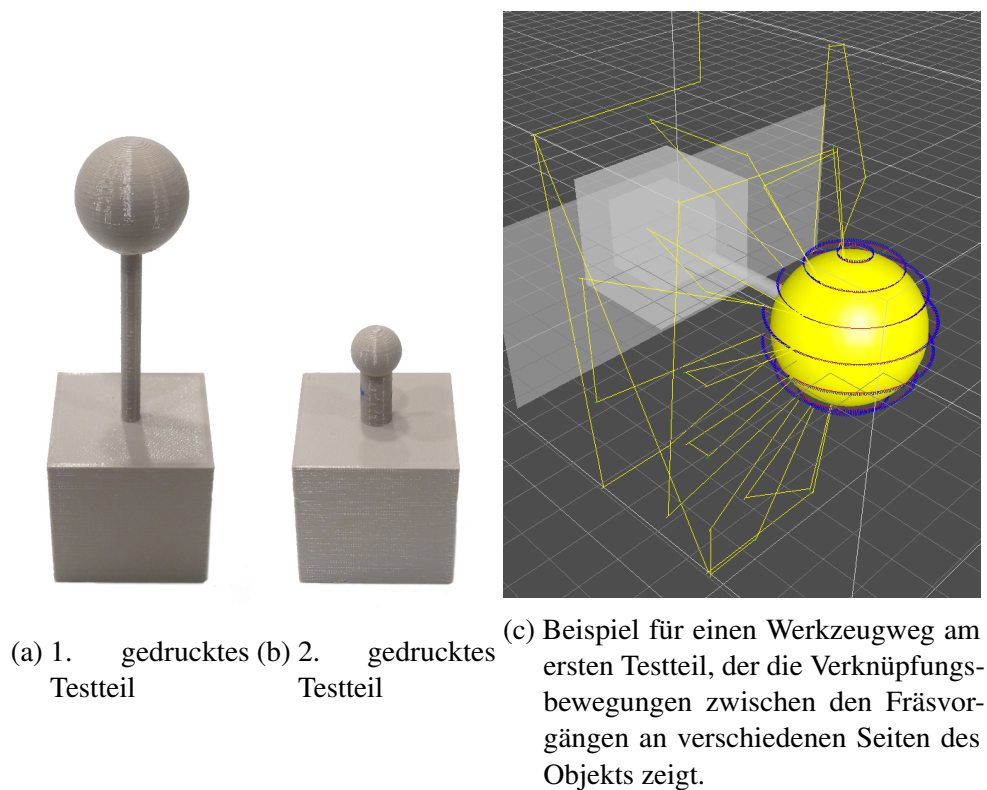


Abbildung 5.15.: Die Testobjekte, die für die Auswertung der Werkzeugwegausführung verwendet werden. Quelle [193].

### 5.2.5. Fräsgenauigkeit

Um die Grundgenauigkeit des vorgeschlagenen Fräsprozesses zu bewerten, wurde das in Abbildung 5.16 gezeigte Referenzteil aus PETG-Kunststoff gedruckt und entlang dreier Achsen A, B und C mit Hilfe von Werkzeugwegen gefräst, die mit der CAM-Software Autodesk Inventor erzeugt wurden. Nach dem Einlegen der Teile in den Greifer und der Durchführung der kamerabasierten Bahnanpassung wurden drei Messungen vorgenommen: Zunächst wurde die Abmessung des gedruckten Teils ohne jegliche Änderungen als Basislinie gemessen. Dann wurde der Werkzeugweg für eine Seite jeder Achse ausgeführt und die Zwischenmaße gemessen. Schließlich wurde der Werkzeugweg für die zweite Seite ausgeführt und die Endmaße wurden aufgezeichnet. Alle Messungen wurden während des Prozesses mit einem digitalen Messschieber ohne zwischenzeitliche Störungen durchgeführt. Für diesen Bewertungsschritt wurden 20 Teile verwendet.

Die Ergebnisse dieses Versuchs sind in Abbildung 5.17 dargestellt, wobei der Fehler zwischen der Messung und dem Nennmaß bei jedem Schritt aufgeführt ist. Wie erwartet sind die gemessenen Abmessungen entlang jeder Achse direkt nach dem Druck ziemlich genau. Nach dem Fräsen der ersten Seite ist der gemessene Fehler viel größer, sinkt aber nach dem Fräsen der zweiten Seite wieder auf einen mittleren Wert. Die Standardabweichung folgt einem ähnlichen



## 5. Post-Processing der gedruckten Bauteile

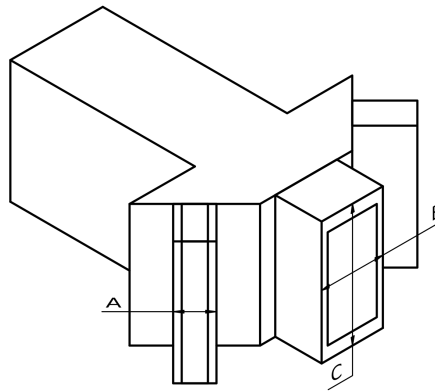


Abbildung 5.16.: Das Testobjekt zur Auswertung der Fräsgenauigkeit. Das Fräsen und Messen erfolgt entlang der hervorgehobenen Achsen A, B und C. Quelle [193].

Trend, ist aber für zwei der Achsen nach der vollständigen Bearbeitung sogar geringer als die des unbearbeiteten Teils.

Die Differenz zwischen dem Fehler der zweiten und dritten Messung lässt auf einen globalen Kalibrierungsfehler schließen. Dies erscheint plausibel, da für den Versuchsaufbau kein strukturierter Kalibrierungsprozess verwendet wurde. Der insgesamt geringe Fehler von weniger als 1 mm für die meisten Teile deutet darauf hin, dass ein solcher Schritt ausreichen könnte, um eine angemessene Genauigkeit für diese Anwendung zu erreichen. Die insgesamt geringe Standardabweichung zeigt, dass die auf der Tiefenkamera basierende Bahnanpassung eine ausreichende Wiederholbarkeit bietet, um Variationen in der Greifhaltung zu kompensieren.

Diese Ergebnisse legen nahe, dass der vorgestellte Aufbau und das Verfahren prinzipiell in der Lage sind, die Support Strukturen zu entfernen.

### 5.2.6. Entfernen der Stützstruktur

Als letzter Test wurde die Leistung der Entfernung der Stützstruktur an den beiden in Abbildung 5.18 gezeigten Testteilen bewertet. Für beide wurden Prüfkörper gedruckt und anschließend unter Verwendung der beschriebenen Schritte zur Einstellung und Ausführung des Greifers gefräst. Es wurden Werkzeugwege generiert, um zunächst den Großteil des Stützmaterials mit einem 8-mm-Werkzeug mit einem Versatz von 5 mm zu entfernen und dann den Rest mit einem feineren 4-mm-Werkzeug zu entfernen. Die Ausführungszeit wurde aufgezeichnet und die Abtragsleistung wurde durch Wiegen der Teile nach dem Abtragen und durch visuelle Inspektion des Ergebnisses beurteilt.

Das erste Testteil ähnelt einem Schaufelblatt und wies glatte Übergänge zwischen verschiedenen komplexen Verzahnungen auf. Drei Testobjekte wurden aus PETG-Kunststoff gedruckt.

## 5. Post-Processing der gedruckten Bauteile

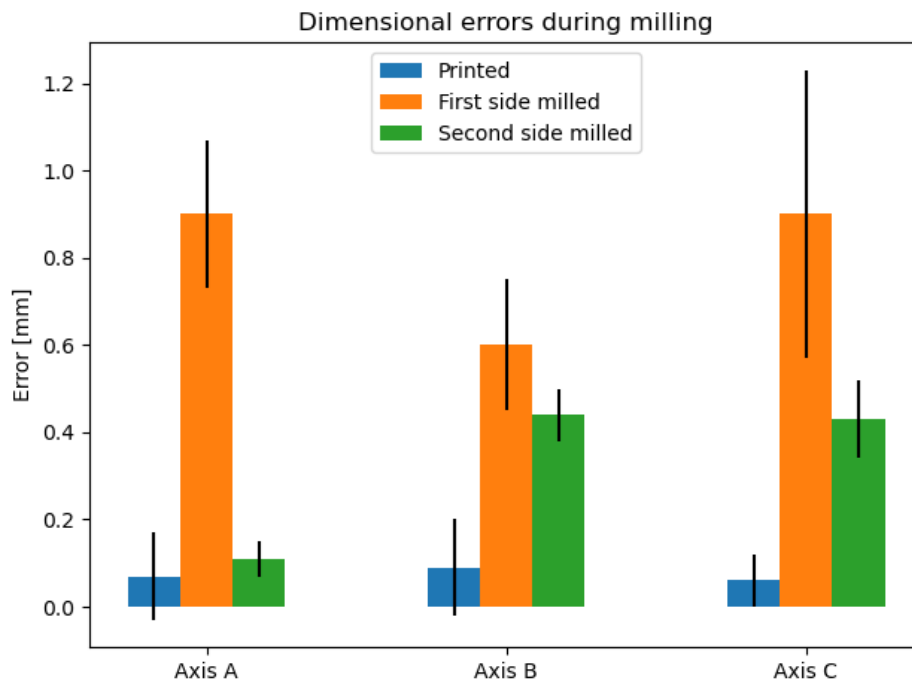


Abbildung 5.17.: Mittlere Fehler und Standardabweichungen der Maße entlang der Achsen A, B und C, siehe Abbildung 5.16 nach dem Druck, nach dem Fräsen der ersten Seite und nach dem Fräsen der zweiten Seite. Quelle [193].

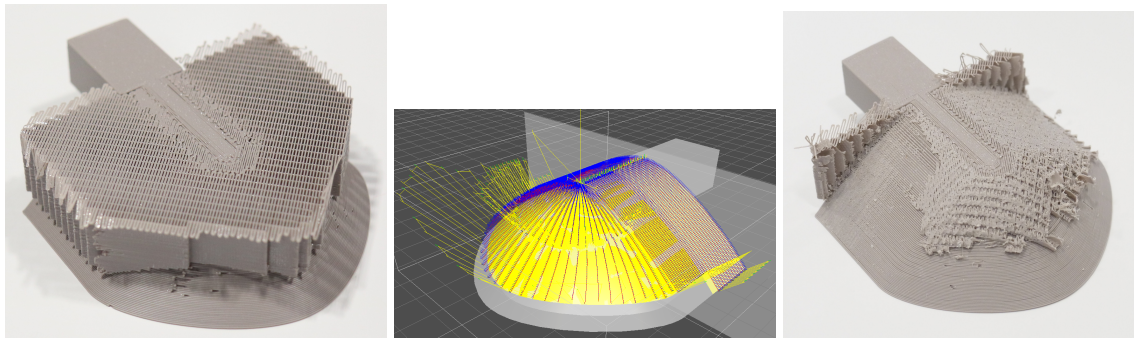
Das zweite Teil besteht aus einem Block mit einer Aussparung, die eine Stütze enthielt. Davon wurden fünf Probekörper aus PLA-Kunststoff gedruckt.

	Teil 1	Teil 2
Planungszeit [s]	943	11
Länge der ersten Werkzeugbahn [mm]	4032	980
Länge des zweiten Werkzeugwegs [mm]	8006	1768
Fräszeit der ersten Werkzeugbahn [s]	569,3	138,5
Fräszeit des zweiten Werkzeugwegs [s]	1146,7	269,5
Verbleibendes Stützmaterial [g]	0,9	0,2

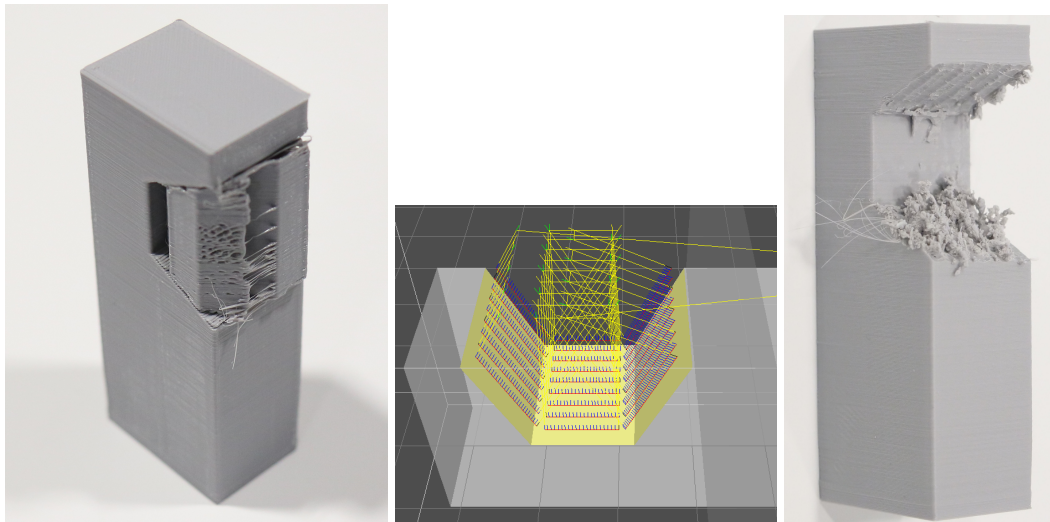
Tabelle 5.1.: Bewertungsergebnisse der Entfernung

Das Ergebnis der Experimente ist in Tabelle 5.1 zu sehen. Es ist sofort zu erkennen, dass die Planungszeit für das komplexere erste Testteil drastisch höher ist, obwohl die Anzahl der Kollisionsprüfungen aufgrund des Fehlens von Hindernissen im Werkzeugweg deutlich geringer sein sollte. Dies ist vor allem auf die Kosten für Abstands- und Kollisionsprüfungen zurückzuführen, die bei komplexeren Oberflächenmerkmalen deutlich mehr Zeit in Anspruch nehmen.

## 5. Post-Processing der gedruckten Bauteile



(a) Das erste Testteil vor der Ent- (b) Der zweite geplante Werk- (c) Erstes Teil nach dem Entfernen der Support Struktur  
fernung. zeugweg am ersten Teil



(d) Das zweite Testteil vor dem (e) Der zweite geplante Werk- (f) Zweites Teil nach  
Entfernen des Supports zeugweg am zweiten Teil Entfernen der Stütze

Abbildung 5.18.: Prozessablauf für zwei Testobjekte. Die bei der Auswertung der Entfernung der Support Struktur verwendeten Teile. a/d) gedrucktes Objekt, b/e) geplante Fräspfade entlang der Oberfläche, c/f) Objekt nach Fräsdurchgang. Sowohl das konkave, als auch das konvexe Testobjekt können beide vom vorgestellten Ansatz bearbeitet werden. Quelle [193].

Beim ersten Test kann die Ausrichtung der Werkzeugbahn reibungslos der Oberflächen-normale ohne Hindernisse folgen, während sich das Werkzeug beim zweiten Teil schrittweise drehen muss, während es sich in den Ausschnitt bewegt. Die iso-parametrische CC-Pfaderzeugung erzeugt für das zweite Teil schön parallele Werkzeugpfade, aber deutlich divergierende Pfade für die Vorderseite des ersten Teils. Die Werkzeugwege für beide Teile wiesen eine große Anzahl von Zwischenpfaden auf, wobei nur 49,6% und 16,7% der Prozesszeit für Fräsbewegungen aufgewendet wurden. Dies ist in erster Linie auf die Instabilität des Planers zurückzuführen, benachbarte Werkzeugbahnen auf der Oberfläche direkt zu verbinden, ohne dass vollständige Werkzeugrückzüge eingefügt werden müssen.

## 5. Post-Processing der gedruckten Bauteile

Die Ergebnisse der Abtragsoperationen sind in Abbildung 5.18 zu sehen. Das Wiegen der resultierenden Teile und der Vergleich mit einem manuell bearbeiteten Teil zeigt, dass der größte Anteil der Support Struktur entfernt wurde. Die Reste auf beiden Teilen sind visuell bei allen geprüften Teilen gleich. Sie zeigen erneut eine Abweichung in der Werkzeugkalibrierung, aber während beim ersten Teil nur eine Hülle an der Vorderseite verbleibt, wurde beim zweiten Teil auch Oberflächenmaterial an einer der Schrägen entfernt. Außerdem bleibt am Ende des ersten Teils eine Wand aus Material zurück, da das Werkzeug nicht zu nahe an den Greifer heranfahren konnte. Nur das zweite Testteil weist deutliche Schneidprobleme an der inneren Schräge auf, wo die kontinuierliche Orientierungsänderung große Roboterbewegungen erforderte, die zu einer geringeren Genauigkeit und Steifigkeit führten. Dieser Effekt sollte bei der Festlegung des Fräsvorschubs berücksichtigt werden.

Mit dieser Auswertung wurde die grundsätzliche Machbarkeit des vorgeschlagenen Entfernens der Support Struktur gezeigt. Der verwendete Trajektorienregler hatte zwar Probleme mit dieser Anwendung, diese sind jedoch vor allem auf Grund der Kinematik des Roboterarms aufgetreten. Eine bessere Positionierung des Armes zum montierten Werkzeug dürfte dieses Problem beheben.

### 5.2.7. Fazit

Beide Bearbeitungsansätze - Werkzeug am TCP und Objekt am TCP - sind erfolgreich evaluiert worden und zeigen, dass eine Nachbearbeitung gedruckter Objekte möglich ist. Aus den vorhandenen Daten des Druckprozesses kann die Stützstruktur als Differenz berechnet und basierend auf der Schnittkante daraus eine Robotertrajektorie generiert werden.

Entscheidend ist vor allem das Wissen über die genaue Position des Objektes relativ zum Roboterarm. Die Präzision des Roboterarms ist je nach Hersteller und Modell weitestgehend konstant und könnte in einer realen Anwendung auch regelmäßig kalibriert werden. Befindet sich jedoch das Objekt nicht genau an der Position, mit der die Trajektorienplanung den Pfad berechnet hat, dann kommt es zu Ungenauigkeiten und Fehlern beim Fräsvorgang. Hier wäre eine kontinuierliche Überwachung des Objektes von Vorteil oder ein Greifsystem, das ein Verrutschen des Bauteils verhindert. Jedoch ist gerade letzteres durch die große Vielzahl an druckbaren Geometrien nur schwer umsetzbar.

Insgesamt hat der Ansatz mit fixiertem Werkzeug das größere Potential, da es einige Vorteile bietet. Zum Einen ist das gedruckte Objekt im Vergleich zum Fräswerkzeug deutlich leichter und zum Anderen treten so keine Beschleunigungen und kein Ruck am Roboterarm durch das Werkzeug auf. Schwierig ist nach wie vor jedoch das Greifen des Objektes. Hier wäre ein geeigneter Greifer zu entwerfen, der in der Lage ist gedrucktes Bauteil in einer Lage festzuhalten. Eventuell könnte im Druckprozess eine Haltevorrichtung zum Druckobjekt hinzugefügt werden, mit der das Objekt dann im späteren Fräsvorgang sicher gehalten werden kann, ähnlich wie beim Drehen von Teilen.

# 6. Zusammenfassung und Ausblick

In diesem Abschnitt werden die Ergebnisse der Arbeit zusammengefasst und diskutiert. Zudem wird ein Ausblick auf mögliche Anschluss Themen aufgezeigt.

## 6.1. Zusammenfassung und Beitrag

Die additive Fertigung gilt als Schlüsseltechnologie für die individuelle Produktion von Gütern aller Art. Gedruckte Objekte sind mittlerweile aus der heutigen Welt nicht mehr wegzudenken, angefangen bei Prototypen in der Forschung und Entwicklung bis hin zu Bauteilen von Flugzeugen. Obwohl die grundlegende Technologie bereits recht alt ist, ist die manuelle Zuarbeit heutzutage noch sehr hoch.

In dieser Arbeit wurden mehrere Ansätze entwickelt und evaluiert, die die Automatisierung der additiven Fertigung ermöglichen, um die manuelle Arbeit zu reduzieren und gleichzeitig die Qualität der Drucke zu verbessern, als auch die Verschwendung von Rohmaterial vorzubeugen. Die Verfahren helfen die Produktionskapazitäten der Drucker effizienter zu nutzen und gleichzeitig den Ausschuss und damit die Verschwendung von Material, Energie und Maschinenzeit zu reduzieren.

Die Beiträge dieser Arbeit können in drei Blöcken strukturiert dargestellt werden.

**In-situ Prozessüberwachung** Eine zuverlässige und automatisierte Überprüfung des Druckprozesses wird möglich, wenn für das Verfahren geeignete Sensoren und Ansätze eingesetzt werden. Im Rahmen dieser Arbeit wurden verschiedene Ansätze Kombinationen aus Sensor und Algorithmus getestet. Gerade optische Sensoren im Zusammenspiel mit Algorithmen der künstlichen Intelligenz bieten hier großes Potenzial. Grundlage hierfür war vor allem ein großer Datensatz an Bildern verschiedenster Geometrien, sowie eine angepasste Architektur des neuronalen Netzes. So wurde für jeden Fehlerfall ein eigenes, kleines neuronales Netzwerk entwickelt und trainiert, um so als Experte für diesen einen Fehlerfall zu gelten. Die Kombination aus mehreren Experten für verschiedene Fehler ermöglichte die Erkennung verschiedener Fehlerklassen bei gleichzeitig effizientem Trainings- und Laufzeitaufwand. So ist es auch in Zukunft möglich die einzelnen Experten mit neuen Bilddaten schnell und effektiv weiterzuentwickeln, ohne ein großes Netz, das alle Fehlerklassen enthält, neu trainieren zu müssen. Dies gilt auch für die Erweiterung um weitere Fehlerklassen, die mit diesem modularen Ansatz einfach möglich ist.

Bei der Untersuchung der Anwendung der akustischen Sensoren hat sich herausgestellt, dass die Transformation der akustischen Daten in Bilddaten, um diese wiederum mit neuronalen Netzen aus der Bildverarbeitungen zu verarbeiten, eine sehr zielführende Variante darstellt. Hierfür werden die akustischen Daten mit Hilfe der Mel-Frequency-Cepstrum-Koeffizienten grafisch aufbereitet und diese Bilddaten dann anschließend ausgewertet. So ist es möglich verschiedene Fehlerfälle sowohl im Fused Filament Fabrication Verfahren, als auch im pulverbasiertem Druckverfahren, zu lernen und zu klassifizieren.

Insgesamt hat sich gezeigt, dass eine optische und akustische Fehlererkennung im Druckprozess möglich ist und die aufgenommenen Sensordaten zielführend mit neuronalen Netzen klassifiziert werden können. Jedoch sind die Verfahren nur sehr schwer zwischen verschiedenen Druckertypen und Herstellungsverfahren portierbar, da die grundlegende Unterschiede zu groß sind.

**Entnahme der gedruckten Objekte** Für die Entnahme der gedruckten Objekte wurde zuerst eine Analyse der Druckdaten, basierend auf der Online-Plattform Thingiverse verfügbaren Druckdaten durchgeführt. Darauf aufbauend wurde mit der Methodik des Morphologischen Kastens Optionen und Varianten ermittelt, die in verschiedenen Tests evaluiert wurden. Die Greiferkonzepte mit dem höchsten Potential wurden anschließend gefertigt und weiteren Tests unterzogen. Hierbei wurden vor allem Langzeittests durchgeführt, um die Abnutzung des Materials zu überprüfen und auch um verschiedene Ablösestrategien der Objekte von der Druckplatte zu untersuchen. Des Weiteren wurde der Maschinencode dazu verwendet die Position der einzelnen, gedruckten Objekte zu ermitteln. Daraus kann sowohl die genaue Greifposition für die spätere Entnahme, als auch die Entnahmereihenfolge berechnet werden. So ist eine Entnahme aus dem Drucker möglich, ohne die Objekte zu beschädigen.

Durch die Formfreiheit, die die additive Fertigung bietet, sind die möglichen druckbaren Geometrien endlos. Dies erschwert die Entwicklung eines universellen Greifkonzepts. Eine erste Generalisierung im Rahmen einer Objektgröße z.B. von 2 bis 10 cm kann mit Hilfe eines Greifkonzeptes umgesetzt werden. Gerade sehr kleine oder sehr große Objekte sind schwer zu greifen. Hierfür könnte dann zum Beispiel ein Greiferwechselkonzept eingesetzt werden, damit für verschiedene Größen, immer das passende Greifkonzept eingesetzt werden kann. Um diese Limitierung zu überkommen, könnte in Zukunft ein Greiferwechselkonzept eingesetzt werden, welches einen größeren Einsatzbereich ermöglicht.

**Post-Processing der gedruckten Objekte** Nach erfolgter Entnahme müssen verfahrenstypische Strukturen entfernt werden. Dazu gehören unter anderem Stützstrukturen, die das Drucken von Überhängen ermöglichen. Aus der Differenz der CAD- und Maschinendaten kann genau bestimmt werden, welche Punkte zum eigentlichen Objekt und welche zu den Stützstrukturen gehören. Dieses Wissen wird verwendet, um anschließend Pfade zu planen, die mit einem Roboter für den Materialabtrag ausgeführt werden können. Zwei verschiedene Ansätze wurden betrachtet: Zum einen wurde Werkzeug direkt am Roboter montiert und das gedruckte Objekt fixiert. Zum anderen wurde Objekt mit Hilfe eines Greifers am Roboter montiert bzw. eingespannt und das Werkzeug blieb fest montiert. Zudem wurde mit Hilfe einer 3D-Kamera die genaue Position des Objekts in den Greiferfingern ermittelt, um so die

geplante Trajektorie anzupassen. Dies ist notwendig da nicht immer ein gleichbleibender Griff gewährleistet ist und es dadurch zu Fehlern in der Nachbearbeitung kommen kann.

Die automatisierte Nachbearbeitung ist im Allgemeinen möglich. Sie weißt jedoch Grenzen auf, sobald Geometrien gedruckt werden, die für den hier entwickelten Ansatz kinematisch nicht nachbearbeitbar sind. Darunter fallen zum Beispiel tiefergehende Bohrungen oder innenliegende Flächen, die außerhalb der Reichweite des Werkzeuges liegen.

Des Weiteren wurde im Rahmen dieser Arbeit ein Druckerarray aufgebaut, das es ermöglicht hat in kurzer Zeit viele Testdaten zu erzeugen. Insgesamt sechs Drucker druckten parallel, während durch einen davor montierten Roboterarm die Sensorik sowie das Entnahmewerkzeug an jedem Drucker eingesetzt werden konnte. So war es möglich ununterbrochen Testdaten zu erzeugen, sowie Testobjekte effizient herzustellen.

## 6.2. Diskussion

Die vorgestellten Ansätze wurden evaluiert und für die Drucktechnologie Fused Filament Fabrication und eine Vielzahl an Objektgeometrien wurden die Anforderungen erfüllt. So ist es möglich Fehler im Druckprozess mit einer hohen Genauigkeit zu erkennen und zu klassifizieren, die gedruckten Objekte nach Fertigstellung schadfrei vom Druckbett zu entnehmen und entsprechend nachzubearbeiten.

Allerdings ist einer der größten Vorteile der additiven Fertigung - die Möglichkeit nahezu jede Geometrie zu fertigen - gleichzeitig auch die größte Herausforderung. So spielt zum Beispiel gerade die Wahl des Materials eine große Rolle und beeinflusst dadurch die möglichen Fehlerfälle im Druckprozess. Je nach Material müssten daher angepasste Algorithmen zum Einsatz kommen, da sich sowohl Fehlerklasse, als auch Fehlerausprägung unterscheiden können. Im Rahmen der Erstellung der Trainings- und Testdaten wurde zwar eine große Anzahl an verschiedenster Geometrien mit berücksichtigt, diese können aber keine Vollständigkeit beanspruchen. Idealerweise würde das System durchgehend mit jedem Druck weiterlernen, um so immer mehr verschiedene Geometrien gesehen zu haben.

Je nach Sensor spielt natürlich auch der Ort des Fehlers eine wichtige Rolle. Gerade bei optischen Sensoren ist die Verdeckung ein Problem, welches sich nur durch den Einsatz von mehreren Sensoren lösen lässt. Hierfür könnten jedoch die gleichen Netze eingesetzt werden, da auch die Trainingsdaten bereits verschiedene Perspektiven beinhaltet haben. Jedoch ist es bei kommerziellen Systemen nahezu unmöglich Sensoren in den Druckraum zu integrieren oder auf die Live-Daten der Hardware zuzugreifen, da diese hinter proprietären Schnittstellen unzugänglich sind. Jedoch ergibt sich für Druckerhersteller hier hohes Potential eine maßgeschneiderte Lösung zu integrieren.

Die Vielzahl der Geometrie-Möglichkeiten ist natürlich auch eine Herausforderung für den Greifer des Roboters. Von Objekten im Submillimeterbereich bis hin zu Objekten mit einer Kantenlänge von einem Meter und größer ist heutzutage je nach Druckverfahren alles möglich.

Ein einziges Greiferdesign für alle Anwendungen und Verfahren gibt es daher nicht. Jedoch konnte hier aufgezeigt werden, welche Aspekte zu berücksichtigen sind und wie es beispielhaft aussehen könnte. Basierend auf diesen Erkenntnissen kann prinzipiell für jede Druckgröße ein Greifer konzipiert werden.

Bei der Nachbearbeitung der Objekte ist das eingesetzte Werkzeug entscheidend und limitiert gleichzeitig auch die Möglichkeiten. So sind gewundene und innenliegende Hohlräume für das Werkzeug und die Roboterkinematik nur schwierig bis unmöglich zu erreichen. Eine Kombination aus verschiedenen Werkzeugen und Werkzeugwechslern könnte hier allerdings Abhilfe schaffen und den Spielraum entsprechend erweitern.

Insgesamt spielt auch die eingesetzte Drucktechnologie eine wichtige Rolle. Durch die Vielzahl an Technologien muss immer für die entsprechend vorliegende ein passender Ansatz eingesetzt werden. Dies birgt natürlich auch immer neue Herausforderungen. Gerade im Bereich der Materialwissenschaften ist die Entwicklungsgeschwindigkeit sehr hoch und es ist durchaus auch denkbar, dass in zukünftigen Druckverfahren die Fehler während des Druckes nicht mehr auftreten oder die Nachbearbeitung komplett anders stattfinden muss. Allerdings ist durch die flexible und modulare Robotik gegeben, dass die Anpassung hierfür überschaubar ist und zukünftig schnell umgesetzt werden kann.

### 6.3. Ausblick

Bei der Entwicklung und Evaluation dieser Ansätze wurde klar, dass die Automatisierung der additiven Fertigung prinzipiell möglich ist. Im Rahmen dieser Arbeit wurde hauptsächlich das Fused Filament Fabrication Verfahren betrachtet, welches nur eines von vielen weiteren Verfahren ist. Die entwickelten Verfahren könnten daher auch in anderen Verfahren eingesetzt beziehungsweise dafür angepasst werden. Je nach Verfahren ist hier auch die zwingende Mitarbeit des Herstellers notwendig, da sowohl die Sensor- als auch die Datenintegration ansonsten schwierig umzusetzen ist.

Durch die steigende Leistung der Hardware im Bereich der neuronalen Netze, wie GPUs oder sogar neuronaler Hardware, wird es möglich, immer komplexere Netzarchitekturen zum einen örtlich nah an der Maschine, zum anderen in Echtzeit zu nutzen. Das ermöglicht eine direkte Auswertung der aktuellen Sensordaten und somit eine schnelle Reaktion auf einen möglichen Fehler. Gleichzeitig kann die Anzahl der Fehlerklassen gesteigert werden, ohne an Evaluierungsgeschwindigkeit einzubüßen. Die Produktionsdaten könnten direkt verwendet werden, um die Verfahren weiter zu verbessern. So könnten die Druckdaten direkt mit den jeweiligen Sensordaten genutzt werden, um entsprechende Klassifikatoren weiter zu verbessern. Dadurch würde die Maschine mit der Zeit immer besser werden und die Auswertung genauer zu den jeweiligen Anforderungen des Nutzers passen. Mit Hilfe aktueller Cloud-Technologien könnten alle Drucker voneinander profitieren und mit Ansätzen wie Federated Learning alle gegenseitig verbessern und gleichzeitig produktionskritische Informationen entsprechend geschützt bleiben. Bisher existiert kein standardisierter Datensatz, der eine Evaluation aller Ansätze untereinander



## 6. Zusammenfassung und Ausblick

möglich macht. Ein solcher könnte dabei helfen, generalisierte Ansätze zu entwickeln, die dann wiederum in mehr Maschinen zum Einsatz kommen könnten.

Auch in der Robotik gibt es immer neue Fortschritte. Eine der Limitierung ist die Anzahl an Achsen beziehungsweise Tragkraft und Reichweite der Roboterarme. Mit neuen Kinematiken, ermöglicht durch neue Materialien und Technologien, kann es möglich werden, dass auch bisher schwer bis unmöglich erreichbare Stellen an gedruckten Objekten in Zukunft einfacher erreicht werden können. Prinzipiell sind die vorgestellten Ansätze hardwareagnostisch, sodass zukünftige Robotikhardware schnell und ohne viel Integrationsaufwand evaluiert werden könnte.

Im Hinblick auf ökonomische Aspekte ist ein Roboter für die Entnahme und Nachbearbeitung pro Drucker natürlich nicht optimal. Gerade durch die langen Druckzeiten würde ein einzelner Roboter pro Drucker die meiste Zeit stillstehen. Allerdings ist es heutzutage möglich Roboter auch mobil zu betreiben, sodass sie mehrere Stationen gleichzeitig betreuen können. In zukünftigen Arbeiten könnten die Verfahren auf mobile Roboter übertragen werden, damit der Roboter in eine ganze Farm von Druckern integriert werden könnte und so mehrere Drucker parallel betreut. Dies würde technisch, als auch ökonomisch Sinn ergeben und gleichzeitig die Attraktivität des Ansatzes steigern. Mit Hilfe eines Werkzeugwechselsystems wäre es zudem möglich verschiedene Sensoren direkt am Roboter zu betreiben, da so nicht die Drucker um- bzw. nachgerüstet werden müssen, sondern das der Roboter die Kontrollen übernimmt und hierfür pro Drucker den passenden Sensor einsetzen kann. Hierfür ist aber auch wieder eine sehr umfangreiche Integration notwendig. Dafür müssten die Datenschnittstellen der 3D-Drucker, als auch der Roboter weitestgehend zugänglich sein. Nur so ließe sich eine tiefgehende Integration umsetzen.

# Anhang

# A. Appendix

## A.1. Verwendete Hardware und Software

Im folgenden Abschnitt werden die verwendeten Hardwarekomponenten kurz aufgeführt und erläutert.

### A.1.1. 3D-Drucker Prusa MK3S+

Was als Hobbyprojekt von Josef Prusa begann, entwickelte sich mit der Zeit zu einer der erfolgreichsten 3D-Druckerhersteller. Die Firma Prusa hat mit ihren Druckermodellen eine sehr große Community, die gerade zu Beginn wichtige Rückmeldung zur Weiterentwicklung der Drucker gegeben haben.

Über die Jahre wurde die MK Serie immer weiter verbessert. In dieser Arbeit kamen mehrere Prusa MK3S+ Drucker zum Einsatz. Mit einem Bauvolumen von 25x21x21 cm können selbst große Objekte hergestellt werden. Die X-Achse ist an der Z-Achse aufgehängt, während sich die Y-Achse orthogonal dazu bewegt. Dadurch ist es möglich den Bauraum abzudecken und gleichzeitig viel Steifigkeit in den einzelnen Achsen zu haben, damit das Druckergebnis dadurch nicht verschlechtert wird.

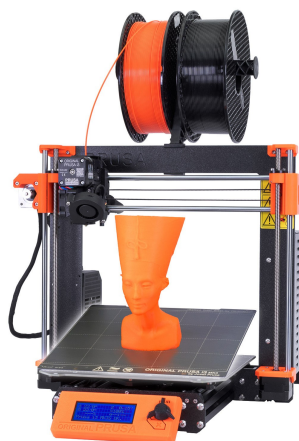


Abbildung A.1.: Prusa MK3S+ [163]

### A.1.2. 3D-Drucker Raise3D N2 Plus

Das Modell N2 Plus vom Druckerhersteller Raise3D wurde initial 2015 per Kickstarter Kampagne finanziert und wurde in den Jahren darauf offiziell vertrieben. Der Drucker zeichnet sich vor allem durch den großen Bauraum von 30,5 x 30,5 x 61,0 cm (Länge x Breite x Höhe) aus, sowie durch den Dual-Extruder Ansatz. Hiermit ist der Drucker in der Lage zwei Filamentarten gleichzeitig zu verarbeiten. Im Gegensatz zu vielen anderen Druckern verfährt hier die XY-Achse mit dem Druckkopf nicht nach oben, sondern ist fest montiert und die Z-Achse verfährt nach unten und ermöglicht so das Drucken von bis zu 61cm hohen Objekten.

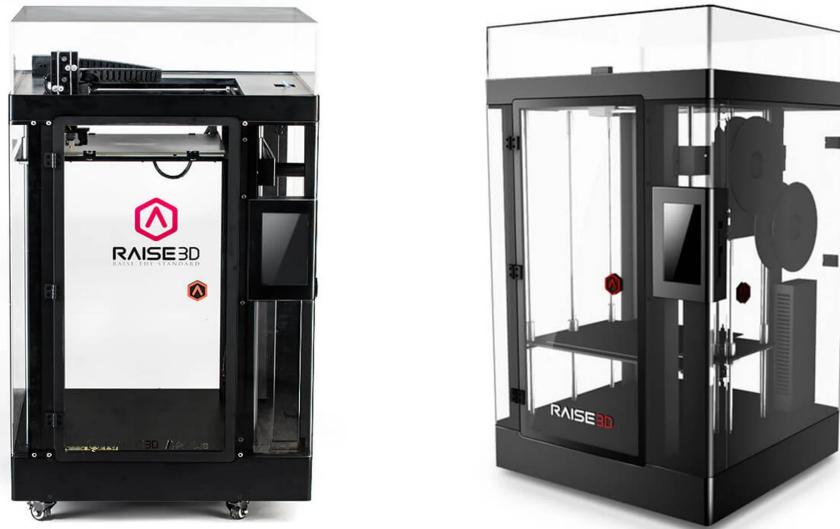


Abbildung A.2.: Raise3D N2 Plus [164]

### A.1.3. Intel RealSense

Die Kameraserie RealSense der Firma Intel hat sich nach der Microsoft Kinect Kamera zum Quasi-Standard entwickelt. Durch die Kombination aus Bauform, technischen Eigenschaften, sowie Preis und Verfügbarkeit sind die RealSense Kameras aus der Robotik nicht mehr wegzudenken. Neben der Möglichkeit ein 2D Farbbild über diverse Softwareschnittstellen auszulesen, werden auch verschiedene Stereoformate (3D) bereitgestellt. Neben der Hardware hat aber auch die Software einige interessante Features. Vom Konfigurationsprogramm zum Einstellen aller möglichen Parameter bis hin zum eigenen ROS-Treiber lässt Intel hier keine Wünsche offen.

Im Rahmen der Arbeit kamen zwei verschiedene Typen zum Einsatz. Ein etwas älteres Modell (RealSense R200), sowie ein neueres (RealSense D415).

### Intel RealSense R200

Die Intel RealSense R200 eignet sich vor allem für enge Bauräume, da sie mit einer sehr geringen Tiefe von nur 7 mm sehr gut untergebracht werden kann.

- Sensor-Technologie: IR-Stereokamera
- Auflösung Farbkamera: 1920x1080
- Auflösung IR-Kamera / Tiefenbild: 640x480
- Größe: 20x130x7mm

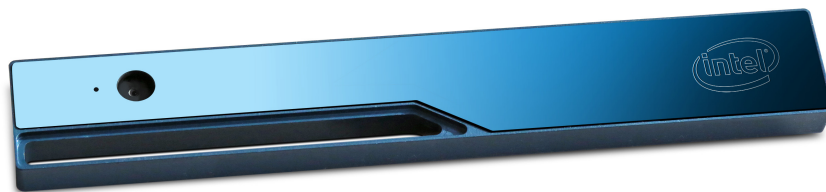


Abbildung A.3.: Intel RealSense R200 [165]

### Intel RealSense D435

Auch die Intel RealSense D4e5 besticht durch ihre Baugröße, kann im Vergleich zur R200 auch mit einer besseren Auflösung des Tiefenbildes glänzen. Gerade für die Erkennung von Objekten in den 3D-Daten ist diese Eigenschaft entscheidend.



Abbildung A.4.: Intel RealSense D435 [166]

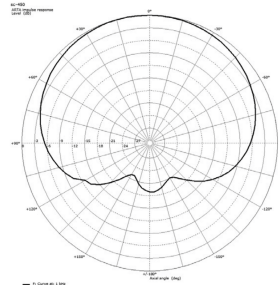
- Sensor-Technologie: IR-Stereokamera
- Auflösung Farbkamera: 1920x1080
- Auflösung IR-Kamera / Tiefenbild: 1280x720
- Größe: 99x20x23 mm

### A.1.4. the t.bone SC 450

Bei dem verwendeten Mikrofon handelt es sich um ein the t.bone SC 450. Das Studio Großmembran Mikrofon hat eine nierenförmige Richtcharakteristik A.5b und ist ausgelegt für den Frequenzbereich von 20 bis 18.000 Hertz. Es kann einfach per USB an einen Computer angeschlossen werden und dann ohne spezielle Treiber betrieben werden.



(a) the t.bone SC-450 Mikrofon [167]



(b) Nieren-Charakteristik des t.bone SC-450 Mikrofons [168]

### A.1.5. Universal Robots Roboter

Im Rahmen dieser Arbeit wurden verschiedene Robotervarianten des Herstellers Universal Robots eingesetzt. Alle eingesetzten Modelle ähneln sich von der Kinematik grundsätzlich, sprich haben sechs Freiheitsgrade und unterscheiden sich nur in ihrer Reichweite und Traglast. Hierbei gibt die Modellnummer immer die maximale Tragkraft des Roboters am Endeffektor an. Ein UR5 ist somit in der Lage am Ende seines Endeffektors bei maximal ausgestrecktem Arm eine Last von 5 kg zu handhaben. Der Namenszusatz „e“, wie z.B. beim UR16e, gibt an, dass der Roboter einen integrierten Kraftmomentensensor im Endeffektor verbaut hat.

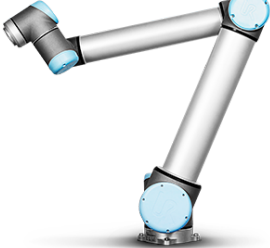
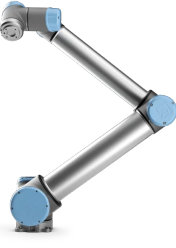
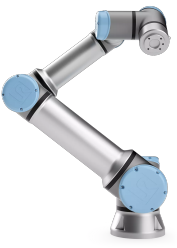
	<b>UR5</b>	<b>UR10</b>	<b>UR16e</b>
			
<b>Tragkraft</b>	5kg	10kg	16kg
<b>Reichweite</b>	850 mm	1300 mm	900 mm
<b>Freiheitsgrade</b>	6 rotierende Gelenke		
<b>Eigengewicht</b>	20,6 kg	33,5 kg	33,1 kg

Tabelle A.1.: Übersicht über grundlegende technische Daten der verschiedenen Robotermodelle von Universal Robots [169], Bilder aus Produktkatalog von [www.universal-robots.com](http://www.universal-robots.com)

Neben der hohen Verfügbarkeit der Roboter im Labor des FZI Forschungszentrum Informatik zeichnet sich das Robotersystem durch seine umfangreiche Integration in das Robot Operating System (ROS, siehe A.1.6) aus. Die Roboter von UR können einfach und sehr flexibel mit Hilfe von ROS programmiert werden, ohne dass eine proprietäre Software notwendig ist. Je nach Anwendungsfall wurde der Roboter dann entweder kartesisch, Punkt-basiert oder mit Hilfe von geplanten Pfaden bewegt.

### **A.1.6. Robot Operating System (ROS)**

Das Robot Operating System (ROS) ist eine Open-Source Software zur Programmierung und Steuerung von Robotern aller Art [52, 170]. Es ist eine Middleware, die jedwede Robotikhardware abstrahiert und mit Hilfe von wenigen generischen Schnittstellen steuerbar macht. Durch eine weite Verbreitung in der Robotik-Community wird heutzutage fast jede Hardware im Bereich der Robotik (Roboterarme, Greifsysteme, etc.) und weitere Komponenten (Sensoren, Schnittstellen, etc.) unterstützt. Neben der Schnittstelle zur Hardware bietet ROS auch umfangreiche Schnittstellen zur Integration von Algorithmen, z.B. für Bahnplanung, Objekterkennung und mehr.

Im Jahr 2007 haben Mitarbeiter eines Robotikinstituts der Stanford University mit der Entwicklung von ROS begonnen. Ihr Ziel war es, ein einheitliches Softwareframework für die Robotik-Welt zu entwickeln, damit nicht jede Forschungsgruppe die state-of-the-art Algorithmen und Funktionalitäten immer wieder aufs Neue implementieren müssen. Ab 2009 wurde das Framework dann von der Willow Garage weiterbetreut. Seit 2012 unterstützt und koordiniert die Open Source Robotics Foundation (OSRF) die Entwicklungen um ROS.

Seit 2013 unterstützt zudem das ROS Industrial Consortium die Entwicklung von Paketen für Anwendungen in der Industrierobotik. Heutzutage werden einige ROS-Pakete von Roboterherstellern selbst entwickelt und gepflegt, um der ROS-Community den bestmöglichen Einsatz ihrer Produkte zu ermöglichen.

## **A.2. Design der Greiferhardware**

Um eine für diesen Anwendungsfall passende Variante der Greiferfinger zu kommen wurde mit Hilfe der Methode "Morphologischer Kasten" für jeden Teilaspekt mehrere Optionen aufgezeigt und diese anschließend in sinnvoller Kombination zusammengefügt. So ergab sich eine Auswahl an potenziellen Kandidaten für die spätere Entnahme.



# A. Appendix

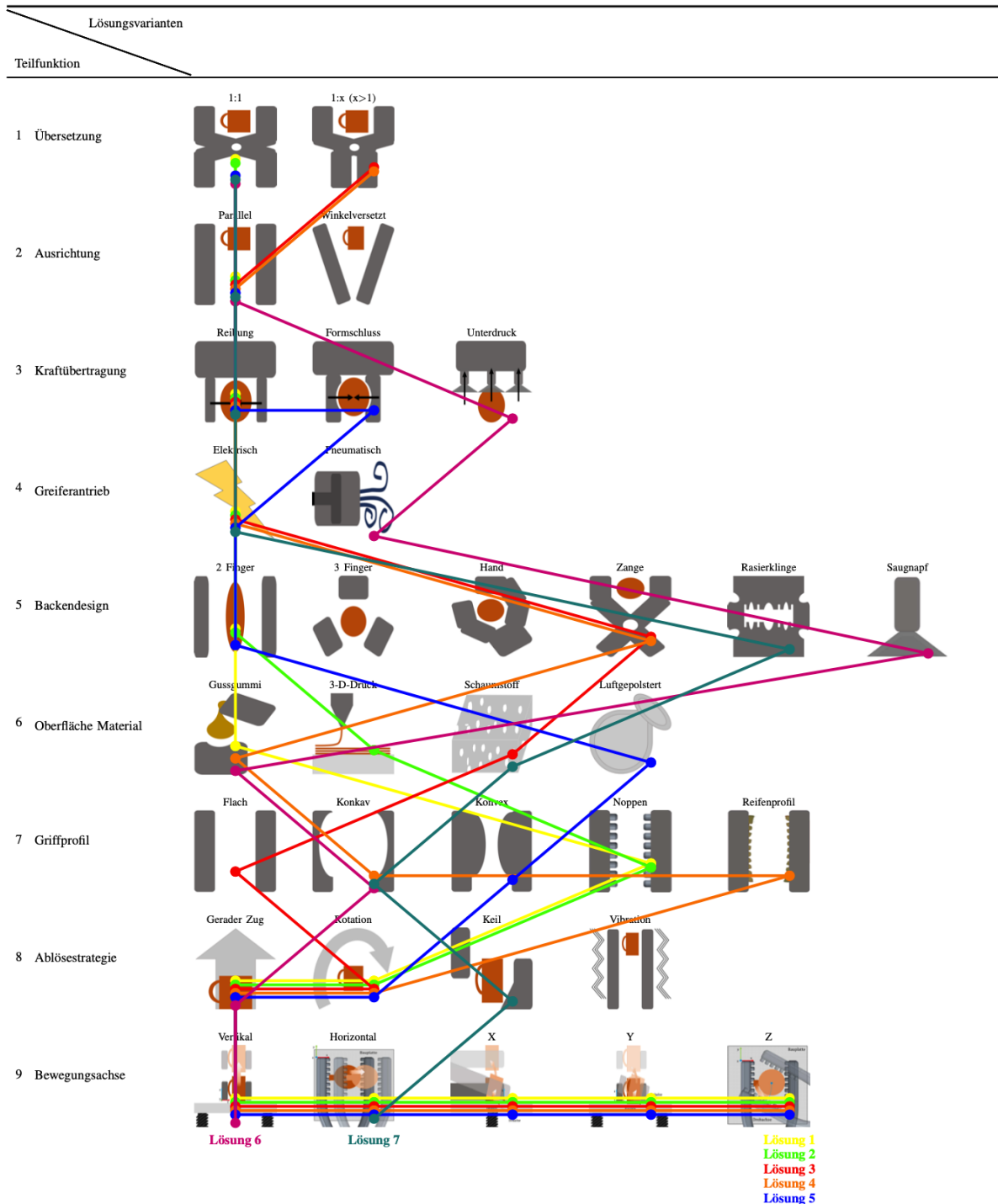


Abbildung A.6.: Morphologischer Kasten mit der Übersicht der verschiedenen Greifaspekte, sowie deren Ausprägung

## A. Appendix

# Glossar

**Additive Fertigung** beschreibt die Fertigung von digitalen Geometrien durch das sequentielle Hinzufügen von Material bis zur vollständigen Geometrie..

**Build Plate** Die Build Plate oder auch Druckplatte oder Druckbett ist essenzieller Teil des Druckprozesses. Auf ihr wird die erste Druckschicht platziert und legt quasi den Grundstein für den nachfolgenden Druck. Damit eine möglichst hohe Haftung zwischen Platte und frisch gedrucktem Filament zustande kommt, ist die Oberfläche der Build Plate üblicherweise rau und beheizt, damit es zu wenig Verformung des Objektes während des Drucks kommt..

**Endeffektor** ist das Werkzeug am Ende eines robotischen Armes mit dem Objekte gegriffen oder manipuliert werden.

**G-Code** Der G-Code ist Maschinencode, der vom Drucker ausgeführt wird, um ein Objekt zu drucken. Der G-Code besteht hauptsächlich aus einfachen Anweisungen, wie die Achsen sich zueinander bewegen müssen und wie viel Filament dabei extrudiert werden muss. Somit ist es möglich das ursprüngliche CAD-Modell auszudrucken..

**Region of Interest** Die Region of Interest beschreibt einen (Teil-) Ausschnitt aus einer Menge, die von besonderer Interesse ist. Bei einem Bild ist dies zum Beispiel ein Bildausschnitt, auf den ein besonderes Augenmerk gelegt wird..

**RGBD-Kamera** ist eine Kombination aus zwei Sensoren in einem Gehäuse: Durch die gleichzeitige Auswertung der Daten einer Farbkamera (RGB: Red Green Blue) und einer Tiefenbildkamera (D: Depth) können eingefärbte 2,5 dimensionale Abbildungen einer Szene erstellt werden.

**Roboter Konfiguration** beschreibt den Zustand aller beweglichen Achsen eines Roboters und somit seine *Körperhaltung*..

**Slicer** Der Slicer zerlegt das vorhandene CAD-Modell in einzelne, horizontale Schichten, sodass es für den Drucker druckbar wird. Das Ergebnis ist der G-Code, welcher anschließend vom Drucker ausgeführt werden kann..

**TensorFlow** TensorFlow ist ein im Rahmen eines Projektes von Google entwickeltes Framework für maschinelles Lernen verschiedenster Art. Mit diesem können Netzwerkarchitekturen erstellt, trainiert und eingesetzt werden. Neben PyTorch ist TensorFlow eines der beiden am häufigsten eingesetzten Frameworks für maschinelles Lernen.

# Akronyme

**AE** Akustische Emission.

**AED** Accoustic Emission Detection.

**AM** Additive Manufacturing, 3D-Druck.

**CAD** Computer Aided Design.

**CNN** Convolutional Neural Network.

**DOF** Degree of freedom.

**FDM** Fused Deposition Modeling.

**FFF** Fused Filament Fabrication.

**FPS** Frames per Second.

**GMM** Gaussian Mixture Models.

**GPU** Graphics Processing Unit.

**HMM** Hidden Markov Model.

**HSMM** Hidden Semi Markov Model.

**LSTM** Long Short-Term Memory.

**MFCC** Mel-Frequenz-Cepstral Koeffizient.

**MLP** Multi-Layer Perceptron.

**NN** Neuronales Netzwerk.

**OVA** One vs. All.

**OVO** One vs. One.

## *Akronyme*

**PBF** Powder Bed Fusion.

**PCA** Principal Component Analysis.

**PCL** Point Cloud Library.

**ROI** Region of Interest.

**ROS** Robot Operating System.

**SLA** Stereolithografie.

**SLS** Selective Laser Sintering.

**STL** STL-Dateiformat zum standardisiertem Austausch von 3D-Daten..

**SVM** Support Vector Machine.

**TCP** Tool Center Point.

# Abbildungsverzeichnis

1.1. Übersicht über die Prozesskette in der additiven Fertigung, sowie deren Arbeitsschritte und Möglichkeit zur Automatisierung. . . . .	3
2.1. Schematische Darstellung des Druckprozesses bei Fused Filament Fabrication. 1) gedrucktes Objekt, 2) Extruder / Druckkopf, 3) Filamentrolle, 4) Druckbett.	10
2.2. Schematische Darstellung des Druckprozesses beim Powder Bed Fusion Druckprozess. 1) Gedrucktes Objekt, 2) Laserquelle, 3) Druckraum gefüllt mit Pulver	11
2.3. Prozessüberwachung durch die Anordnung mehrerer Kameras (grünes Boards auf weißen Säulen) direkt vor dem Drucker auf Höhe des Extruders. So kann der komplette Entstehungsprozess aus verschiedenen Perspektiven beobachtet werden. Quelle [60]. . . . .	14
2.4. Grafische Zusammenfassung der Arbeit von Holzmond <i>et al.</i> Ein Zwei-Kamera-Setup wurde in einem FFF-Drucker installiert, um den Druck zu verifizieren, während das Objekt gedruckt wurde. Mit digitaler 3D-Bildkorrelation (3D-DIC) und einem Vergleich mit dem Computermodell konnten sie Fehler während des Druckes erkennen. Quelle [61]. . . . .	15
2.5. Bildbasierte Fehlererkennung mit zwei an den Extruder montierten Digitalmikroskope. Die Durchflussrate des Filaments wird angepasst, wenn ein Fehler erkannt wurde. Bild (a) zeigen die Durchflussrate während des Prozesses, das Bild der Mikroskope vor (b) und nach (c) der Regleranpassung. Diese Anpassung wurde durch ein k-NN gelernt und übertraf Alternativen wie LDA, SVM und NBC. Quelle [73]. . . . .	17
2.6. Übersicht über die drei Komplexitätsklassen, die gedruckt wurden. Untersucht wurden letztendlich allerdings nur die Schichten oberhalb der zusätzlichen Geometrien, um eine Vergleichbarkeit herzustellen. Quelle [94]. . . . .	22
2.7. Der Creality CR-30 hat ein integriertes Förderband, welches als Build Plate dient. Somit ist es möglich Objekte beliebige Länge zu drucken. Gleichzeitig werden abgeschlossene Drucke automatisch vom Druckbett entfernt, da sie nach Erreichen des Ende des Druckbetts sich dort automatisch ablösen. Quelle [99]. . . . .	24
3.1. In diesem Abschnitt der Arbeit liegt der Fokus auf der Überwachung des Druckprozesses live während des Prozesses (schwarzer Kasten). Hierfür werden verschiedene Verfahren untersucht und angewandt, um Fehler im Prozess frühzeitig zu klassifizieren. . . . .	28

## Abbildungsverzeichnis

3.2.	Übersicht über die einzelnen Vorverarbeitungsschritte um den passenden Bildausschnitt im Bild der Kamera zu finden. 1) Umwandlung zu Graustufen-Bild, 2) Anwendung Sobel-Filter, 3) Suche nach Nozzle im Bild mit Detailbild von Nozzle, 4) Ausschneiden der Region of Interest unterhalb der Nozzle-Position.	30
3.3.	Fehlerbild von einem Layershift. Dieser führt zu einer Verschiebung der einzelnen Schichten ab einem bestimmten Zeitpunkt. Quelle [192]. . . . .	30
3.4.	Verschiedene Fehlerfälle visuell dargestellt. Quelle [112]. . . . .	32
3.5.	Schematische Darstellung der Position der einzelnen Druckerkomponenten, sowie der Kamera auf Höhe des Extruders. Quelle basierend auf [115]. . . . .	36
3.6.	Bild des Raise Druckers mit angebauten Kameras. Die Kameraposition ist diagonal zum Druckbett ausgerichtet und ermöglicht so eine Sicht sowohl auf die X- als auch auf die Y-Achsenbewegung des Druckers. Quelle [192]. . . . .	37
3.7.	Die Verteilung der Extrudererkennung in der XY-Ebene für die Druckplatte des ideaMaker-Druckers. Die Farben zeigen an, an welcher Stelle der Extruder zuverlässig erkannt werden konnte. In der Regel werden Objekte in der Mitte der Druckplatte gedruckt, so dass Druckvorgänge problemlos überwacht werden können. (Grün = zuverlässige Erkennung, Gelb = schwankende Erkennung, Rot = keine Erkennung). Quelle [192]. . . . .	38
3.8.	Druckfortschritt in Prozent für die beide Fehler können während des Druckvorgangs erkannt werden. Bei der Schichtverschiebung dauert es im Durchschnitt länger, bis sie erkannt wird, da der Algorithmus mehrere verschobene Schichten benötigt, um einen Fehler zu erkennen. Es gibt keine Erkennung vor 40 %, da beide Fehler erst nach 39 % des Druckvorgangs erkannt wurden. Quelle [192].	39
3.9.	Klassifizierungsergebnisse während des Druckprozesses für die Fehlerfälle Layer Shift und Nozzle Clogging / Extruder verstopft. Quelle [192]. . . . .	39
3.10.	Verschiedene Aufnahmen von gedruckten Objekten mit und ohne aufgetretenem Fehler. . . . .	40
3.11.	Das Konzept der in-situ Prozessüberwachung bei einem Fused Filament Fabrication (FFF) Drucker. Hierbei wird ein Convolutional Neural Network (CNN) Modell verwendet, um Fehler im Druckprozess zu klassifizieren. Quelle [186].	41
3.12.	Möglichkeiten der binären Klassifikation der Klasse Gut sowie den Fehlerklassen Warping und Layer Shift. Quelle [186]. . . . .	43
3.13.	Ablauf des vorgestellten Entwicklungskonzepts für das modulare System angefangen bei der Auswahl der passenden Fehlerklassen und möglichen Testobjekten über die Erzeugung von Trainingsdaten bis hin zur Implementierung der einzelnen Module sowie anschließenden Evaluierung. Quelle [186]. . . . .	44
3.14.	Auswahl der relevanten Fehlerklassen am Beispiel einer Pyramide. Quelle [186].	44
3.15.	Reduktion der Dimensionalität der Dateninstanzen anhand 5.000 Bildinstanzen für die fünf Fehlerklassen: 0 = Gut, 1 = Warping, 2 = Layer Shift, 3 = Spaghetti und 4 = Clogged Nozzle. Quelle [186]. . . . .	47



3.16. Bild des Testaufbaus mit Roboterarm links im Bild und 3D-Drucker rechts. Die Kameraposition ist diagonal zum Druckbett ausgerichtet und ermöglicht so eine Sicht sowohl auf die X- als auch auf die Y-Achsenbewegung des Druckers. Sie ist um 180 Grad gedreht montiert. Zum einen, damit sie eine optimale Perspektive auf Höhe des Druckbetts einnehmen kann, zum anderen, um bei allen weiteren Prozessschritten die Bewegungen des Roboters nicht zu beeinträchtigen. Quelle [186]. . . . .	49
3.17. Originalbild und gedrehtes Bild, welches später als Grundlage für die weiteren Vorverarbeitungsschritte dient. Die Region of Interest ist als roter Kasten eingezeichnet. Quelle [186]. . . . .	50
3.18. Ausgewählte ROI nach den verschiedenen Vorverarbeitungsschritten. Quelle [186]. . . . .	50
3.19. Histogramm eines Grauwertbildes einmal ohne und einmal mit angewandter Gamma-Transformation. Quelle [186]. . . . .	51
3.20. Initiale Netzarchitektur des CNNs. Ausgabeschicht Binär-Klassifikation: 1 Neuron, Ausgabeschicht, Mehrklassen-Klassifikation: 5 Neuronen, Loss-Funktion: Binary Crossentropy / Sparse Categorical Crossentropy, Optimierer: Adam. Quelle [186]. . . . .	52
3.21. Aufteilung der Trainingsdaten für den One vs. One- und den One vs. All-Ansatz. Quelle [186]. . . . .	53
3.22. Netzstruktur 4CNN-1Dense. Das Netz wird über den Adam-Optimierer und die Binary Crossentropy Loss-Funktion trainiert. Die Ausgabeschicht besteht aus einem Neuron. Quelle [186]. . . . .	55
3.23. Zusätzlich verwendete Objekte mit Fehler im Datensatz. Verschiedene Geometrien, die im bisherigen Datensatz noch nicht vertreten waren. Quelle [186]. . . . .	57
3.24. Grad-CAM Visualisierung verschiedener Fehlerklassen. Die Relevanz des entsprechenden Bereichs ist durch die „Wärme“ in der Heatmap dargestellt. Je heller der Farbton eines Bereichs, desto relevanter ist dieser für die Klassifikation. Quelle [186]. . . . .	59
3.25. Erweiterung der Trainingsdaten durch Data Augmentation wie z.B. durch Spiegelung oder Helligkeitsvariation. Quelle [186]. . . . .	60
3.26. Darstellung der Ergebnisse der Rastersuche am Beispiel Layer Shift. Quelle [186]. . . . .	65
3.27. Grafische Darstellung der Einflüsse der verschiedenen Methoden auf das erzielte Ergebnis der Klassifizierungen. Quelle [186]. . . . .	67
3.28. Gesamtübersicht über alle fünf Modelle. Beim aktuellen Eingabebild handelt es sich um den Fehler Layer Shift, der auch vom Modell passend klassifiziert wird. Alle anderen vier Modelle klassifizieren die Bildeingabe nicht als für sich relevant. Quelle [186]. . . . .	68
3.29. Konfusionsmatrix für die Fehlerklassifikation. Hierbei wird deutlich, dass die Fehler deutlich voneinander unterschieden werden können und es nur zwischen den beiden Fehlerklassen Gut und Warping zu Verwechslungen kommt. . . . .	70

## Abbildungsverzeichnis

3.30. Objekte in den beiden verwendeten Ähnlichkeitsstufen. Zur Evaluierung der Generalisierungsfähigkeit werden weitere Geometrien zu den Datensätzen hinzugefügt. Hierbei kommen sowohl weitere einfache Geometrien (Vase, Runddach, Gengon), aber auch erstmals komplexe Geometrien (Schildkröte, Calicat, Cyndaquil) zum Einsatz. Quelle [186]. . . . .	70
3.31. Konfusionsmatrix für den modularen Ansatz bei Testung mit neuen, einfachen Objekten aus der 1. Ähnlichkeitsstufe, siehe Abbildung 3.30. . . . .	71
3.32. Konfusionsmatrix für die komplexen Modelle aus der 2. Ähnlichkeitsstufe. . . . .	72
3.33. Eingesetzte Machine Learning Pipeline zur Umsetzung der neuronalen Netze im Rahmen der akustischen Fehlererkennung. Quelle basierend auf [190]. . . . .	74
3.34. Bild eines beschädigten Objekts, bei dem die Höhe der Düse während des Druckvorgangs nicht korrekt war und zu tief über das Objekt fuhr. Dadurch ergaben sich mehrere Unsauberkeiten, die akustisch detektierbar sind (rote Kreise). Quelle [190]. . . . .	74
3.35. Grinding Filament. Tritt entweder auf, wenn die Düse verstopft ist oder wenn der Durchmesser des Filaments falsch ist. Quelle [112]. . . . .	75
3.36. Verschiedene Methoden der Datenaugmentation zur Erhöhung der Varianz der aufgezeichneten Daten. Drei Methoden wurden auf die Originaldaten angewandt: Zeitdehnung (Timestretch), Tonhöhenverschiebung (Pitchshift) und Verstärkung (Amplifying). Quelle [190]. . . . .	76
3.37. Audiosignale im gegebenen Zeitfenster für mehrere Ereignisse wie das Lüftergeräusch (fan noise) (a), das Druckgeräusch (printing) (b), die Bewegung der Z-Achse (movement z-axis) (c) oder die falsche Extruderhöhe (nozzle height incorrect) (d). Quelle [190]. . . . .	78
3.38. Verlust von Training und Validierung über 25 Epochen. Quelle [190]. . . . .	78
3.39. Versuchsaufbau Druckbett: Das Mikrofon befindet sich während des Druckvorgangs in der Nähe des Extruders. Aufgrund der Nierencharakteristik des Mikrofons werden die meisten Umgebungsgeräusche nicht aufgenommen und der Fokus liegt auf dem Drucker selbst. Quelle [190]. . . . .	79
3.40. Konfusionsmatrix des LSTM auf ungesehene Daten. . . . .	80
3.41. Mel-Spektrogramme von Beispielen der Fehlerklassen Lüftergeräusch, Druck, Bewegung Z-Achse und falsche Extruderhöhe. Quelle [190]. . . . .	81
3.42. Mel-Spektrogramme von Beispielen der Fehlerklassen Lüftergeräusch und falsche Extruderhöhe. Die Energien im Hochfrequenzbereich scheinen fast gleich zu sein, aber in den Energien im Niederfrequenzbereich sind kleine Unterschiede zu erkennen (vgl. grüne Kästen). Quelle [190]. . . . .	81
3.43. Mehrere Testobjekte in Würfelform mit unterschiedlichen Druckeinstellungen wurden gedruckt, um die für das CNN benötigten Daten zu erzeugen. Quelle [138]. . . . .	83
3.44. Komplexitätsklassen der gedruckten Objekte. Von links nach rechts nimmt der Anteil der porösen Struktur zu, beginnend mit Klasse 1 ohne jegliche Porosität bis hin zu Klasse 3 mit einer hohen Wahrscheinlichkeit von Porosität. Der graue Teil des Bauteils ist die zusätzliche Struktur die zum Würfel hinzugefügt wurde, um die Auswirkung komplexer Geometrien zu untersuchen. Quelle [94].	83

## Abbildungsverzeichnis

3.45. Übersicht über die einzelnen Verarbeitungsschritte: Von den akustisch aufgezeichneten Daten jeder Schicht wird ein Offset subtrahiert, das Downsampling und die Normalisierung durchgeführt. Nach der Umwandlung der Daten in Mel-Spektrogramme werden sie als .npys gespeichert. Für die Trainingsphase werden die .npys erneut geladen, in der Größe verändert und dem CNN zugeführt. . . . .	84
3.46. Ein mit 4 MHz und $n_{fft} = 2048$ abgetastetes Spektrogramm. Spektralenergien im unteren Frequenzbereich ( $< 500\text{kHz}$ , grün hervorgehoben) sind deutlich sichtbar. Quelle [190]. . . . .	85
3.47. Mel-Spektrogramm, abgetastet mit 176400 Hz, Achse ist logarithmisch skaliert. Gelb hervorgehobene Spektralmuster im oberen Frequenzbereich ( $> 10\text{ kHz}$ , grün hervorgehobener Bereich). Der Rest ist größtenteils konstant. Quelle [190].	86
3.48. Die Architektur des evolutionären neuronalen Netzes. Es hat insgesamt 1,8 Millionen trainierbare Parameter und besteht aus mehreren Pooling- und Faltungsschichten, die das Klassifizierungsergebnis verbessern sollen. Quelle [190]. . . . .	87
3.49. Zuordnung der Porosität zur Klassenbezeichnung. Die geschätzte Porosität (in %) befindet sich auf der Y-Achse, die Objekte geordnet nach zunehmender Porosität auf der X-Achse. Quelle [94]. . . . .	89
3.50. Wahrscheinlichkeitsdichtefunktionen bezüglich der geschätzten Porositätsprozente. Die rot hervorgehobenen Linien stellen die strengen Klassengrenzen dar. Quelle [94]. . . . .	90
4.1. In diesem Abschnitt der Arbeit liegt der Fokus auf der Entnahme des gedruckten Objekts, nachdem der Druck abgeschlossen wurde (schwarzer Kasten). Hierfür werden verschiedene Verfahren untersucht und angewandt, um das Objekt schadfrei und zuverlässig zu entnehmen. . . . .	94
4.2. Der Greiferoffset beschreibt die räumliche Differenz zwischen der Energieaufnahme und der Krafteinleitung. Quelle [187]. . . . .	95
4.3. Diese Grafik zeigt den Greifbereich von 68 und 200 mm, was zu einer 60%igen und 97%igen Greifobjektdeckung der analysierten Daten führt. Für die Approximierung wurde eine Kurve in die gemessenen Datenpunkte gelegt. Quelle [187]. . . . .	96
4.4. Analyse der Dimensionen verschiedener Objekte in X-,Y- und Z-Achse. Es wird ersichtlich, dass ein Greifer mit 140 mm Öffnungsbreite 90% aller analysierten Objekte handhaben kann. Quelle [187]. . . . .	97
4.5. Entworfenen und bewerteten Greiffinger mit unterschiedlichen Greifbereichen sowie unterschiedlichen flexiblen Greifflächen. Quelle [187]. . . . .	98
4.6. Schematische Darstellung der vier Grundelemente zur Erzeugung der Objekt-ablösestrategien: Ausführen einer Translation entlang der Z-Achse (a), einer Rotation um eine der Achsen X (b), Y (c) oder Z (d). Quelle [187]. . . . .	101

## Abbildungsverzeichnis

4.7. Die Bewertung der Ablösekraft erfolgte anhand der Position und der Kontaktfläche zwischen dem Objekt und der Druckplatte, um das Entlastungsverhalten bei zukünftigen Ablösungen abzuschätzen (wie bei der Trendlinie Position 3 beschrieben). Hierfür wurden die gedruckten Objekte mit Hilfe einer Waage von der Druckplatte abgezogen und die dabei auftretende Kraft gemessen. Quelle [187]. . . . .	102
4.8. Aufbau zur Simulation der Adhäsionskräfte zwischen dem gedruckten Objekt und der ferromagnetischen Oberfläche. Die auftretenden Kräfte werden durch das Hinzufügen von Magneten zum 3D-gedruckten Objekt simuliert, um die vorgeschlagenen Greiferfinger und Ablösestrategien mit kurzen Zykluszeiten bewerten zu können. Quelle [187]. . . . .	104
4.9. Eine Nahaufnahme vor und nach der Bewertung des gegossenen, entgrateten Greiferfingers, der nach mehr als 2000 Benutzungszyklen wenig Verschleiß zeigt. Quelle [187]. . . . .	105
4.10. Beispiel aus der Slicing-Software mit 3 stl-Dateien, die auf der Druckplatte platziert werden und die Grundlage für den generierten G-Code bilden. Quelle [189]. . . . .	106
4.11. Je nachdem welche Zentren für die Bestimmung der linearen Regression verwendet werden, unterscheidet sich die spätere Perspektive der Kamera am Roboterarm. Quelle [189]. . . . .	109
4.12. Beispiel eines Druckobjekts in der XZ-Ebene: Keine Möglichkeit, den Greifer in Richtung $d_{Z_{mass}}$ zu bewegen, da die Greiferbacken zu breit für die berechnete Position sind. Quelle [189]. . . . .	111
4.13. Vergleich der Greifpunkte auf $Z_{center}$ und $Z_{mass}$ : Der Punkt auf $Z_{center}$ hat weniger Kontaktpunkte und ist daher weniger stabil. Quelle [189]. . . . .	112
4.14. Die Punktwolke wird verarbeitet und an Hand von Hypothesen die darin enthaltene Anzahl von Objekten bestimmt. Quelle [189]. . . . .	116
4.15. Die Punktwolke des Objekts wird in mehreren Schritten vorverarbeitet, um die bestmögliche Eingabe für den Greifbarkeitsalgorithmus zu erhalten. Hierbei wird final der Ausschnitt gesucht, in dem der Griffpunkt liegt, um hierfür die Griffbreite zu bestimmen. Quelle [189]. . . . .	117
4.16. Bildsequenz der verschiedenen Zwischenschritte des vorgestellten Ansatzes zur automatischen Entfernung gedruckter Objekte. Quelle [189]. . . . .	119
5.1. In diesem Abschnitt der Arbeit liegt der Fokus auf der Nachbearbeitung gefertigter Objekte (schwarzer Kasten). Prozesstechnisch notwendige Stützstrukturen müssen nach dem Druck entfernt werden, um ein funktionales Bauteil zu bekommen. . . . .	120
5.2. Der Roboterarm führt die berechnete Trajektorie aus, um die Support Struktur vom eigentlichen Bauteil zu trennen. Hierbei steht das gedruckte Objekt auf der Druckplatte, die wiederum in eine vorher kalibrierte Position gebracht wurde. Quelle [185]. . . . .	121

## Abbildungsverzeichnis

5.3.	Architekturdiagramm des vorgestellten Ansatzes. Die Eingabedaten werden segmentiert und voxelisiert und je nach Anwendungsfall wird die Trajektorie mit integrierter Kollisionsprüfung generiert. Die berechnete Trajektorie kann dann von einem Motion Stack wie der FZI Motion Pipeline oder MoveIt![155] ausgeführt werden. Basierend auf Quelle [185]. . . . .	123
5.4.	Ergebnis der G-Code-Voxelisierung mit dem gewünschten Objekt und der Stützstruktur. Dies ist die Grundlage für die Pfadplanung mit OMPL sowie für die Kollisionsprüfung. Quelle [185]. . . . .	124
5.5.	Beispiel der Voxelsätze, die durch Gleichungen 5.8 und 5.9 aus den Robotermodell STL- und G-Code-Dateien erzeugt wurden. Quelle [185]. . . . .	126
5.6.	Beispiel für Voxelcluster von Stützkontaktregionen in verschiedenen Farben. Quelle [185]. . . . .	127
5.7.	Das Beispielobjekt als STL-Modell (a), gedruckt mit Support Struktur (b), nachbearbeitet durch den Roboter (c) und einen Menschen (d) und mit einem möglichen Bearbeitungsfehler (e). Quelle [185]. . . . .	128
5.8.	Simulieren Sie die Entfernung der Stützstruktur im Inneren eines zylindrischen Objekts. Der untere Teil der Stützstruktur wurde bereits entfernt und die Voxel-darstellung entsprechend aktualisiert. Der obere Teil der Stützstruktur wurde bisher noch nicht vom Werkzeug angefahren und ist daher noch in den Daten repräsentiert. Quelle [185]. . . . .	129
5.9.	Das Robotermodell bei der Nachbearbeitung und Entfernung mehrerer Säulen der Stützstruktur aus dem Druck. Die ersten Säulen sind bereits entfernt worden. Quelle [185]. . . . .	130
5.10.	Indem die beiden verwendeten Eingangsdaten CAD-Datei und G-Code entsprechend angepasst werden, ist es auch problemlos möglich mit dem Roboter zu Fräsen. . . . .	130
5.11.	Überblick über den vorgeschlagenen Nachbearbeitungsablauf. Nachdem das Teilemodell im CAD konstruiert und mit einem Slicer in G-Code-Maschinenanweisungen übertragen wurde, werden zusätzliche Stützstrukturen identifiziert und Trajektorien geplant und gespeichert. Nach der Fertigung des Teils wird es mit einer Tiefenkamera erfasst und vermessen, bevor die Support Struktur mit einer festen Frässpindel entfernt wird. Basierend auf Quelle [193]. . . . .	131
5.12.	Segmentierung einer einzelnen Extrusionsschicht. Die dunkleren Linien sind als Stütze klassifizierte Extrusionen, eine morphologische Schließung dieser ergibt die farbigen Regionen und die roten Grenzen werden über alle Schichten hinweg verbunden, um solide Stützmodelle zu erhalten. Die verschiedenen Farben geben an, dass die entsprechenden Teile zu einer Region gehören. Quelle [193]. . . . .	132
5.13.	Beispiel für eine verbindende Bewegung zwischen Zwei Punkten auf einer Objektoberfläche. Der grüne Start- und der rote Endpunkt werden entlang der Werkzeugachse auf die safe box projiziert. Die Bewegung zwischen ihnen wird auf der Oberfläche der safe box ausgeführt. Quelle [193]. . . . .	133

## Abbildungsverzeichnis

5.14. Das Hardware-Setup, mit dem das vorgeschlagene Verfahren zum Entfernen von Stützen evaluiert wurde. Der UR16e-Roboterarm auf der linken Seite hält das Werkstück vor der Tiefenkamera, bevor er die Fräsbahn vor der unteren Frässpindel startet. Quelle [193]. . . . .	135
5.15. Die Testobjekte, die für die Auswertung der Werkzeugwegausführung verwendet werden. Quelle [193]. . . . .	136
5.16. Das Testobjekt zur Auswertung der Fräsgenauigkeit. Das Fräsen und Messen erfolgt entlang der hervorgehobenen Achsen A, B und C. Quelle [193]. . . . .	137
5.17. Mittlere Fehler und Standardabweichungen der Maße entlang der Achsen A, B und C, siehe Abbildung 5.16 nach dem Druck, nach dem Fräsen der ersten Seite und nach dem Fräsen der zweiten Seite. Quelle [193]. . . . .	138
5.18. Prozessablauf für zwei Testobjekte. Die bei der Auswertung der Entfernung der Support Struktur verwendeten Teile. a/d) gedrucktes Objekt, b/e) geplante Fräspfade entlang der Oberfläche, c/f) Objekt nach Fräsdurchgang. Sowohl das konkave, als auch das konvexe Testobjekt können beide vom vorgestellten Ansatz bearbeitet werden. Quelle [193]. . . . .	139
A.1. Prusa MK3S+ [163] . . . . .	147
A.2. Raise3D N2 Plus [164] . . . . .	148
A.3. Intel RealSense R200 [165] . . . . .	149
A.4. Intel RealSense D435 [166] . . . . .	149
A.6. Morphologischer Kasten mit der Übersicht der verschiedenen Greifaspekte, sowie deren Ausprägung . . . . .	153

# Tabellenverzeichnis

3.1. Übersicht der verglichenen Ansätze. . . . .	54
3.2. Klassifizierungsergebnisse des 4CNN-1Dense Netzes . . . . .	55
3.3. Vergleich der konventionellen Netzstrukturen untereinander. . . . .	56
3.4. Ergebnisse mit erweitertem Datensatz . . . . .	56
3.5. Accuracy nach Erweitern des Datensatzes durch Data Augmentation. . . . .	61
3.6. Klassifizierungsergebnisse nach Kombination von Dropout und Data Augmentation. . . . .	61
3.7. Ergebnisse mit Anwendung von Early Stopping . . . . .	62
3.8. Übersicht der Erkennungsgenauigkeit, sowie der Falsch-Negativ-Rate, bei den verschiedenen, angewandten Schritten. . . . .	63
3.9. Parameter zur Suchraumeingrenzung beim Hyperparameter Tuning am Beispiel Layer Shift . . . . .	64
3.10. Ergebnisse der Klassifizierung nach erfolgter Rastersuche . . . . .	65
3.11. Ergebnisse der Klassifizierung nach erfolgter Zufallssuche . . . . .	66
3.12. Fehlerklassifikation mit erweitertem Datensatz. . . . .	69
3.13. Generalisierungsfähigkeit . . . . .	71
3.14. Verwendete Klassen für das Training des neuronalen Netzes. . . . .	76
3.15. 5-fache Cross-Validation. Bei jeder Iteration werden 5 Foldings erstellt, von denen ein Fold als Validierungsset fungiert, die anderen vier gehören zum Trainingsset. . . . .	88
3.16. Experiment 1: Ergebnismetriken einer 5-fachen Cross-Validation für verschiedene Abtastraten und ein Mel-Spektrogramm $256 \times 256$ . F1-Score ist für die Vorhersage nach Training und Validierung. . . . .	91
3.17. Experiment 2: Resultierende Metriken einer 5-fachen Cross-Validation für verschiedene Abtastraten und Mel-Spektrogramm der Größe $512 \times 512$ . F1-Score ist für die Vorhersage nach Training und Validierung. . . . .	91
3.18. 5-fache Cross-Validation für verschiedene Abtastraten und Mel-Spektrogramm der Größe $256 \times 256$ . . . . .	92
3.19. Konfusionsmatrix für $F_S = 176.400 Hz$ . . . . .	92
3.20. Ausführungszeiten für ausgewählte Experimente. Die Trainingszeit gilt für eine 5-fache Cross-Validation, d.h. 5 Trainings + Validierung. Die Vorhersage ist für eine Probe à 0.9 s. Es wurden zwei verschiedene Eingabeformate evaluiert. Die Mel-Spektrogramme und unveränderte Spektrogramme. Durch die lange Trainingsdauer bei gleichzeitig langer Inferenzzeit wurden Spektrogramme als Eingabe nicht weiter verfolgt, da sie im Vergleich zu den Mel-Spektrogrammen deutlich aufwändiger und zeitintensiver sind. . . . .	93

## *Tabellenverzeichnis*

4.1. Unterschiedliche Clustering-Algorithmen ohne (obere Zeile) und mit (untere Zeile) Mindestabstand zwischen den Objekten in der G-Code-Datei. Quelle [189]. . . . .	108
4.2. Testdateien mit mehreren Objekten und unterschiedlichen Mengen an Objekten. Die Tests 1 bis 4 waren erfolgreich, während Test 5 fehlschlug, da der Abstand zwischen den Objekten zu gering war. Quelle [189]. . . . .	114
5.1. Bewertungsergebnisse der Entfernung . . . . .	138
A.1. Übersicht über grundlegende technische Daten der verschiedenen Robotermodele von Universal Robots [169], Bilder aus Produktkatalog von <a href="http://www.universal-robots.com">www.universal-robots.com</a> . . . . .	151



# Literaturverzeichnis

- [1] Nadav Noor, Assaf Shapira, Reuven Edri, Idan Gal, Lior Wertheim, and Tal Dvir. 3d printing of personalized thick and perfusable cardiac patches and hearts. *Advanced Science*, page 1900344, 2019.
- [2] Eckart Uhlmann, Robert Kersting, Tiago Borsoi Klein, Marcio Fernando Cruz, and Anderson Vicente Borille. Additive manufacturing of titanium alloy for aircraft components. *Procedia Cirp*, 35:55–60, 2015.
- [3] David E Cooper, Mark Stanford, Kevin A Kibble, and Gregory J Gibbons. Additive manufacturing for product improvement at red bull technology. *Materials & Design*, 41:226–230, 2012.
- [4] Rupp gebäuedruck. <https://rupp-gebaeuedruck.de>. Zugriff am 21.04.2024.
- [5] Futurecraft 4d: Adidas präsentiert neuen schuh mit 4d-zwischensohle. <https://3druck.com/kunst/futurecraft-4d-adidas-praesentiert-neuen-schuh-mit-4d-protect\discretionary{\char\hyphenchar\font}{}{}zwischensohle-1081681/>. Zugriff am 21.04.2024.
- [6] Thomas Campbell, Christopher Williams, Olga Ivanova, and Banning Garrett. Could 3D printing change the world. *Technologies, Potential, and Implications of Additive Manufacturing*, Atlantic Council, Washington, DC, 2011.
- [7] Barry Berman. 3-d printing: The new industrial revolution. *Business horizons*, 55(2):155–162, 2012.
- [8] LA Hockaday, KH Kang, NW Colangelo, PYC Cheung, B Duan, E Malone, J Wu, LN Girardi, LJ Bonassar, H Lipson, et al. Rapid 3d printing of anatomically accurate and mechanically heterogeneous aortic valve hydrogel scaffolds. *Biofabrication*, 4(3):035005, 2012.
- [9] Yi Wei Daniel Tay, Biranchi Panda, Suvash Chandra Paul, Nisar Ahamed Noor Mohamed, Ming Jen Tan, and Kah Fai Leong. 3D printing trends in building and construction industry: a review. *Virtual and Physical Prototyping*, 12(3):261–276, 2017.
- [10] Samuel H Huang, Peng Liu, Abhiram Mokasdar, and Liang Hou. Additive manufacturing and its societal impact: a literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1191–1203, 2013.

- [11] fizik antares versus evo 00 adaptive. [https://www.fizik.com/eu\\_en/antares-versus-evo-00-adaptive.html](https://www.fizik.com/eu_en/antares-versus-evo-00-adaptive.html). Zugriff am 21.04.2024.
- [12] You mawo. <https://www.youmawo.com>. Zugriff am 21.04.2024.
- [13] Thierry Rayna and Ludmila Striukova. From rapid prototyping to home fabrication: How 3D printing is changing business model innovation. *Technological Forecasting and Social Change*, 102:214–224, 2016.
- [14] Simon Ford and Mélanie Despeisse. Additive manufacturing and sustainability: an exploratory study of the advantages and challenges. *Journal of cleaner Production*, 137:1573–1587, 2016.
- [15] Sofiane Guessasma, Weihong Zhang, Jihong Zhu, Sofiane Belhabib, and Hedi Nouri. Challenges of additive manufacturing technologies from an optimisation perspective. *International Journal for Simulation and Multidisciplinary Design Optimization*, 6:A9, 2015.
- [16] Iso/ts 15066:2016 - robots and robotic devices - collaborative robots. Standard, International Organization for Standardization, Geneva, CH, February 2016.
- [17] The mainstreaming of additive manufacturing. <https://www.mckinsey.com/capabilities/operations/our-insights/the-mainstreaming-of-additive-manufacturing>. Zugriff am 05.03.2024.
- [18] Linn Danielsen Evjemo, Signe Moe, Jan Tommy Gravdahl, Olivier Roulet-Dubonnet, Lars Tore Gellein, et al. Additive manufacturing by robot manipulator: An overview of the state-of-the-art and proof-of-concept results. In *Emerging Technologies and Factory Automation (ETFA), 2017 22nd IEEE International Conference on*, pages 1–8. IEEE, 2017.
- [19] Formlabs. Form cell. 2017 (zuletzt besucht 11. Januar, 2023).
- [20] Nayanee Gupta, Christopher Weber, and Sherrica Newsome. Additive manufacturing: status and opportunities. *Science and Technology Policy Institute, Washington*, 2012.
- [21] Matthew S Thompson. Current status and future roles of additives in 3d printing—a perspective. *Journal of Vinyl and Additive Technology*, 28(1):3–16, 2022.
- [22] Rasheedat Modupe Mahamood, Tien C Jen, Stephen A Akinlabi, Sunir Hassan, Kamar O Abdulrahman, and Esther T Akinlabi. Role of additive manufacturing in the era of industry 4.0. In *Additive manufacturing*, pages 107–126. Elsevier, 2021.
- [23] Prahar M Bhatt, Rishi K Malhan, Aniruddha V Shembekar, Yeo Jung Yoon, and Satyandra K Gupta. Expanding capabilities of additive manufacturing through use of robotics technologies: A survey. *Additive manufacturing*, 31:100933, 2020.

- [24] Abduladim Salem Bala, Md Saidin bin Wahab, and Mazatusziha binti Ahmad. Elements and materials improve the fdm products: A review. In *Advanced Engineering Forum*, volume 16, pages 33–51. Trans Tech Publ, 2016.
- [25] Siranee Nuchitprasitchai, Michael C. Roggemann, and Joshua M. Pearce. Three hundred and sixty degree real-time monitoring of 3-d printing using computer analysis of two camera views. *Journal of Manufacturing and Materials Processing*, 1(1), 2017.
- [26] Matthias Faes, Wim Abbeloos, Frederik Vogeler, Hans Valkenaers, Kurt Coppens, Toon Goedemé, and Eleonora Ferraris. Process monitoring of extrusion based 3d printing via laser scanning. *arXiv preprint arXiv:1612.02219*, 2016.
- [27] Felix Baumann and Dieter Roller. Vision based error detection for 3D printing processes. *MATEC Web of Conferences*, 59:06003, 2016.
- [28] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, January 2016.
- [29] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert Gao, and Dazhong Wu. Deep Learning for Smart Manufacturing: Methods and Applications. *Journal of Manufacturing Systems*, 48:144–156, January 2018.
- [30] Zhicheng Zhang, Ismail Fidan, and Michael Allen. Detection of Material Extrusion In-Process Failures via Deep Learning. *Inventions*, 5(3):25, July 2020.
- [31] Digital Metal® launches a no-hand production concept for 3D metal printing. 11 October 2018, (Zugriff am 13.01.2023).
- [32] Figure 4 - 3D Systems. 2018 (Zugriff am 13.01.2023).
- [33] Baohua Zhang, Yuanxin Xie, Jun Zhou, Kai Wang, and Zhen Zhang. State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review. *Computers and Electronics in Agriculture*, 177:105694, 2020.
- [34] Atef A Ata. Optimal trajectory planning of manipulators: a review. *Journal of Engineering Science and technology*, 2(1):32–54, 2007.
- [35] Ron Alterovitz, Sven Koenig, and Maxim Likhachev. Robot planning in the real world: research challenges and opportunities. *Ai Magazine*, 37(2):76–84, 2016.
- [36] MG Mohanan and Ambuja Salgoankar. A survey of robotic motion planning in dynamic environments. *Robotics and Autonomous Systems*, 100:171–185, 2018.
- [37] Yonghua Chen and Fenghua Dong. Robot machining: recent development and future research issues. *The International Journal of Advanced Manufacturing Technology*, 66(9):1489–1497, Jun 2013.
- [38] Wei Ji and Lihui Wang. Industrial robotic machining: a review. *The International Journal of Advanced Manufacturing Technology*, Apr 2019.

- [39] Lejun Cen and Shreyes N Melkote. Effect of robot dynamics on the machining forces in robotic milling. *Procedia Manufacturing*, 10:486–496, 2017.
- [40] Charles W Hull. Apparatus for production of three-dimensional objects by stereolithography, März 11 1986. US Patent 4,575,330.
- [41] RepRap. Online, Zugriff am 13.01.2023.
- [42] Rhys Jones, Patrick Haufe, Edward Sells, Pejman Iravani, Vik Olliver, Chris Palmer, and Adrian Bowyer. Reprap – the replicating rapid prototyper. *Robotica*, 29(1):177–191, 2011.
- [43] A.Bowyer. 3d printing and humanity’s first imperfect replicator. *3D Printing and Additive Manufacturing*, 1(1):4–5, 2014.
- [44] Stephen Mellor, Liang Hao, and David Zhang. Additive manufacturing: A framework for implementation. *International journal of production economics*, 149:194–201, 2014.
- [45] Iñigo Flores Ituarte, Eric Coatanea, Mika Salmi, Jukka Tuomi, and Jouni Partanen. Additive manufacturing in production: a study case applying technical requirements. *Physics Procedia*, 78:357–366, 2015.
- [46] A comprehensive list of all 3d printing technologies (2023). <https://manufactur3dmag.com/a-comprehensive-list-of-all-3d-printing-technologies/>. Zugriff am 14.04.2024.
- [47] Apparatus and method for creating three-dimensional objects. <https://patentimages.storage.googleapis.com/21/01/d3/69165ba25d15e0/US5121329.pdf>. Zugriff am 14.04.2024.
- [48] Andreas Gebhardt. Generative fertigungsverfahren: Additive manufacturing und 3d drucken für prototyping-tooling-produktion. 2013.
- [49] DIN Richtlinie. Roboter und robotikgeräte –kollaborierende roboter (iso/ts 15066:2016), 2014.
- [50] DIN Richtlinie. Industrieroboter - sicherheitsanforderungen - teil 1: Roboter (iso 10218-1:2011), 2011.
- [51] DIN Richtlinie. Industrieroboter - sicherheitsanforderungen - teil 2: Robotersysteme und integration (iso 10218-2:2011), 2011.
- [52] Morgan Quigley, Josh Faust, Tully Foote, and Jeremy Leibs. Ros: an open-source robot operating system.
- [53] Ros metrics. <https://docs.ros.org/en/rolling/The-ROS2-Project/Metrics.html>. Zugriff am 21.04.2024.
- [54] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics and Automation Magazine*, 19(4):72–82, December 2012.

- [55] Wang, TM., Xi, JT., and Jin. A model research for prototype warp deformation in the fdm process. April 2007.
- [56] Alfredo Liverani, Francesco Leali, and Marcello Pellicciari. Real-time 3d features reconstruction through monocular vision. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 4(2):103–112, May 2010.
- [57] Stefan Kleszczynski, Joschka zur Jacobsmühlen, Jan T. Sehart, and Gerd Witt. *Error Detection in Laser Beam Melting Systems by High Resolution Imaging*. 08 2012.
- [58] J.Rowe R.Dinwiddie, L.Love. Real-time process monitoring and temperature mapping of a 3d polymer printing process. May 2013.
- [59] Harald Krauss, Thomas Zeugner, and Michael F. Zaeh. Layerwise monitoring of the selective laser melting process by thermography. *Physics Procedia*, 56:64 – 71, 2014. 8th International Conference on Laser Assisted Net Shape Engineering LANE 2014.
- [60] Jeremy Straub. Initial Work on the Characterization of Additive Manufacturing (3D Printing) Using Software Image Analysis. *Machines*, 3(2):55–71, April 2015.
- [61] Oliver Holzmond and Xiaodong Li. In situ real time defect detection of 3D printed parts. *Additive Manufacturing*, 17:135–142, October 2017.
- [62] Siranee Nuchitprasitchai, Michael Roggemann, and Joshua M. Pearce. Factors effecting real-time optical monitoring of fused filament 3D printing. *Progress in Additive Manufacturing*, 2(3):133–149, September 2017.
- [63] M. Moretti, A. Rossi, and N. Senin. In-process monitoring of part geometry in fused filament fabrication using computer vision and digital twins. *Additive Manufacturing*, page 101609, September 2020.
- [64] Jie Liu, Youmin Hu, Bo Wu, and Yan Wang. An improved fault diagnosis approach for FDM process with acoustic emission. *Journal of Manufacturing Processes*, 35:570–579, October 2018.
- [65] Haixi Wu, Yan Wang, and Zhonghua Yu. In situ monitoring of FDM machine condition via acoustic emission. *The International Journal of Advanced Manufacturing Technology*, September 2015.
- [66] Yedige Tlegenov, Geok Soon Hong, and Wen Feng Lu. Nozzle condition monitoring in 3D printing. *Robotics and Computer-Integrated Manufacturing*, 54:45–55, December 2018.
- [67] Yedige Tlegenov, Wen Feng Lu, and Geok Soon Hong. A dynamic model for current-based nozzle condition monitoring in fused deposition modelling. *Progress in Additive Manufacturing*, 4(3):211–223, September 2019.

- [68] Prahalad K. Rao, Jia (Peter) Liu, David Roberson, Zhenyu (James) Kong, and Christopher Williams. Online Real-Time Quality Monitoring in Additive Manufacturing Processes Using Heterogeneous Sensors. *Journal of Manufacturing Science and Engineering*, 137(6):061007, December 2015.
- [69] Michele Moretti, Federico Bianchi, and Nicola Senin. Towards the development of a smart fused filament fabrication system using multi-sensor data fusion for in-process monitoring. *Rapid Prototyping Journal*, 26(7):1249–1261, June 2020.
- [70] Gabriel Pieter Greeff and Meinhard Schilling. Closed loop control of slippage during filament transport in molten material extrusion. *Additive Manufacturing*, 14:31–38, March 2017.
- [71] Sayyeda Saadia Razvi, Shaw Feng, Anantha Narayanan, Yung-Tsun Tina Lee, and Paul Witherell. A Review of Machine Learning Applications in Additive Manufacturing. In *Volume 1: 39th Computers and Information in Engineering Conference*, page V001T02A040, Anaheim, California, USA, August 2019. American Society of Mechanical Engineers.
- [72] Xinbo Qi, Guofeng Chen, Yong Li, Xuan Cheng, and Changpeng Li. Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives. *Engineering*, 5(4):721–729, August 2019.
- [73] Chenang Liu, Andrew Chung Chee Law, David Roberson, and Zhenyu (James) Kong. Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication. *Journal of Manufacturing Systems*, 51:75–86, April 2019.
- [74] Aditya Saluja, Jiarui Xie, and Kazem Fayazbakhsh. A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks. *Journal of Manufacturing Processes*, 58:407–415, October 2020.
- [75] Chenang Liu, David Roberson, and Zhenyu Kong. Textural Analysis-based Online Closed-Loop Quality Control for Additive Manufacturing Processes. page 7.
- [76] Yuanbin Wang, Jiakang Huang, Yuan Wang, Sihang Feng, Tao Peng, Huayong Yang, and Jun Zou. A CNN-based Adaptive Surface Monitoring System for Fused Deposition Modeling. *IEEE/ASME Transactions on Mechatronics*, pages 1–1, 2020.
- [77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [78] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

- [79] Yaser Banadaki, Nariman Razaviarab, Hadi Fekrmandi, and Safura Sharifi. Toward Enabling a Reliable Quality Monitoring System for Additive Manufacturing Process using Deep Convolutional Neural Networks. *arXiv:2003.08749 [cond-mat, stat]*, March 2020. arXiv: 2003.08749.
- [80] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [81] Zeqing Jin, Zhizhou Zhang, and Grace X. Gu. Automated real-time detection and prediction of interlayer imperfections in additive manufacturing processes using artificial intelligence. *Advanced Intelligent Systems*, 2(1):1900130, January 2020.
- [82] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and Avoiding Negative Transfer. *arXiv:1811.09751 [cs, stat]*, October 2019. arXiv: 1811.09751.
- [83] Chloé Clavel, Thibaut Ehrette, and Gaël Richard. Events detection for an audio-based surveillance system. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1306–1309. IEEE, 2005.
- [84] Xiaodan Zhuang, Xi Zhou, Mark A Hasegawa-Johnson, and Thomas S Huang. Real-world acoustic event detection. *Pattern Recognition Letters*, 31(12):1543–1551, 2010.
- [85] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.
- [86] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018. ACM, 2015.
- [87] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM, 2014.
- [88] Venkatesh Boddapati, Andrej Petef, Jim Rasmusson, and Lars Lundberg. Classifying environmental sounds using image recognition networks. *Procedia computer science*, 112:2048–2056, 2017.
- [89] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [90] Sharath Adavanne, Giambattista Parascandolo, Pasi Pertilä, Toni Heittola, and Tuomas Virtanen. Sound event detection in multichannel audio using spatial and harmonic features. *arXiv preprint arXiv:1706.02293*, 2017.
- [91] Annamaria Mesaros, Toni Heittola, Emmanouil Benetos, Peter Foster, Mathieu Lagrange, Tuomas Virtanen, and Mark D Plumbley. Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 26(2):379–393, 2018.

- [92] Haixi Wu, Zhonghua Yu, and Yan Wang. Real-time fdm machine condition monitoring and diagnosis based on acoustic emission and hidden semi-markov model. *The International Journal of Advanced Manufacturing Technology*, 90(5-8):2027–2036, 2017.
- [93] N Eschner, L Weiser, B Häfner, and G Lanza. Development of an acoustic process monitoring system for selective laser melting (slm). 2018.
- [94] N Eschner, L Weiser, B Häfner, and G Lanza. Classification of specimen density in laser powder bed fusion (l-pbf) using in-process structure-borne acoustic process emissions. *Additive Manufacturing*, 34:101324, 2020.
- [95] D Kouprianoff, I Yadroitsava, A Du Plessis, N Luwes, and I Yadroitsev. Monitoring of laser powder bed fusion by acoustic emission: Investigation of single tracks and layers. *Frontiers in Mechanical Engineering*, 7:678076, 2021.
- [96] Austin Weber. Additive Manufacturing: The Quest for Automation, 2019. [Online]. Available: <https://www.assemblymag.com/articles/95074-additive-manufacturing-the-quest-for-automation>.
- [97] Keppner, Kahlenborn, Richter, Jetzke, Lessmann, and Bovenschulte. *Die Zukunft im Blick: 3D-Druck*. 2018.
- [98] The Factory of The Future: 3D Demonstrators | Stratasys. 2018 (Zugriff am 13.01.2023).
- [99] Creality cr-30: Der 3dprintmill, infinite z-belt 3d drucker. <https://www.creality3dofficial.com/files/goods/CR-30-2.jpg>. Zugriff am 28.04.2023.
- [100] Valeria Villani, Fabio Pini, Francesco Leali, Cristian Secchi, and Cesare Fantuzzi. Survey on human-robot interaction for robot programming in industrial applications. *IFAC-PapersOnLine*, 51(11):66–71, 2018.
- [101] Rüdiger Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2-3):109–116, 2004.
- [102] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Robot programming by demonstration. *Springer handbook of robotics*, pages 1371–1394, 2008.
- [103] Barak A Pearlmutter. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2):263–269, 1989.
- [104] Yanlong Huang, Dieter Büchler, Okan Koç, Bernhard Schölkopf, and Jan Peters. Jointly learning trajectory generation and hitting point prediction in robot table tennis. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 650–655. IEEE, 2016.
- [105] Pedro Neto, Dário Pereira, J Norberto Pires, and A Paulo Moreira. Real-time and continuous hand gesture spotting: An approach based on artificial neural networks. In *2013 IEEE International Conference on Robotics and Automation*, pages 178–183. IEEE, 2013.



- [106] Francisco A Candelas, Santiago T Puente, Fernando Torres, Francisco G Ortiz, Pablo Gil, and Jorge Pomares. A virtual laboratory for teaching robotics. *complexity*, 1:10–11, 2003.
- [107] Chih-Hsing Chu, Yu-Wang Liu, Pai-Chia Li, Liao-Chuan Huang, and Yuan-Ping Luh. Programming by demonstration in augmented reality for the motion planning of a three-axis cnc dispenser. *International Journal of Precision Engineering and Manufacturing-Green Technology*, pages 1–9, 2019.
- [108] Arne Muxfeldt, Jan Niklas Haus, Jingyuan Cheng, and Daniel Kubus. Exploring tactile surface sensors as a gesture input device for intuitive robot programming. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2016.
- [109] A. Karim and A. Verl. Challenges and obstacles in robot-machining. In *IEEE ISR 2013*, pages 1–4, Oct 2013.
- [110] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [111] E. R. Davies. *Computer and machine vision: theory, algorithms, practicalities*. Elsevier, 2012.
- [112] Simplify3D. Print quality troubleshooting guide. <https://www.simplify3d.com/support/print-quality-troubleshooting/>, 2019. [Zugriff am 13.01.2023].
- [113] Intel. Intel realsense r200 datasheet. Online, Zugriff am 13.01.2023.
- [114] Intel. Intel realsense d400 datasheet. Online, Zugriff am 13.01.2023.
- [115] Schematische darstellung des raise3d n2 plus drucker. [https://www.printer3d.one/wp-content/uploads/Raise3D-raise-N2\\_N2\\_Plus-Quick\\_Start\\_Guide-04.png](https://www.printer3d.one/wp-content/uploads/Raise3D-raise-N2_N2_Plus-Quick_Start_Guide-04.png). Zugriff am 28.04.2023.
- [116] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- [117] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [118] Roland Schwaiger and Joachim Steinwendner. *Neuronale Netze programmieren mit Python*. Rheinwerk Computing, 2019.
- [119] Siranee Nuchitprasitchai, Michael Roggemann, and Joshua M Pearce. Factors effecting real-time optical monitoring of fused filament 3d printing. *Progress in Additive Manufacturing*, 2(3):133–149, 2017.
- [120] Ugandhar Delli and Shing Chang. Automated process monitoring in 3d printing using supervised machine learning. *Procedia Manufacturing*, 26:865–870, 2018.

- [121] Peter Lipton. Inference to the best explanation. *A Companion to the Philosophy of Science*, pages 184–193, 2017.
- [122] François Chollet and Joseph J Allaire. *Deep Learning mit R und Keras: Das Praxis-Handbuch von den Entwicklern von Keras und RStudio*. MITP-Verlags GmbH & Co. KG, 2018.
- [123] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [124] Aurélien Géron. *Hands-on machine learning with scikit-learn and tensorflow: Concepts, Tools, and Techniques to build intelligent systems*, 2017.
- [125] Sungheon Park and Nojun Kwak. Analysis on the dropout effect in convolutional neural networks. In *Computer Vision—ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*, pages 189–204. Springer, 2017.
- [126] Haibing Wu and Xiaodong Gu. Towards dropout training for convolutional neural networks. *Neural Networks*, 71:1–10, 2015.
- [127] Nurshazlyn Mohd Aszemi and PDD Dominic. Hyperparameter optimization in convolutional neural network using genetic algorithms. *International Journal of Advanced Computer Science and Applications*, 10(6), 2019.
- [128] Erik Bochinski, Tobias Senst, and Thomas Sikora. Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In *2017 IEEE international conference on image processing (ICIP)*, pages 3924–3928. IEEE, 2017.
- [129] K Krishnakumari, E Sivasankar, and Sam Radhakrishnan. Hyperparameter tuning in convolutional neural networks for domain adaptation in sentiment classification (htcnn-dasc). *Soft Computing*, 24:3511–3527, 2020.
- [130] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.
- [131] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.
- [132] Wentai Zhang, Brandon Abranovic, Jacob Hanson-Regalado, Can Koz, Bhavya Duvvuri, Kenji Shimada, Jack Beuth, and Levent Burak Kara. Flaw detection in metal additive manufacturing using deep learned acoustic features. 2020.
- [133] Marco Grasso and Bianca Maria Colosimo. Process defects and in situ monitoring methods in metal powder bed fusion: a review. *Measurement Science and Technology*, 28(4):044005, 2017.

- [134] Wenjia Wang, Jinqiang Ning, and Steven Y Liang. Prediction of lack-of-fusion porosity in laser powder-bed fusion considering boundary conditions and sensitivity to laser power absorption. *The International Journal of Advanced Manufacturing Technology*, 112(1):61–70, 2021.
- [135] Jan Niclas Eschner. *Entwicklung einer akustischen Prozessüberwachung zur Porenbestimmung im Laserstrahlschmelzen*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2021.
- [136] Niclas Eschner. Dataset: In-process monitoring of laser powder bed fusion using structure-borne acoustic process emissions to predict porosity. *Karlsruhe Institute of Technology (KIT)*, 2021.
- [137] Haijun Gong. Generation and detection of defects in metallic parts fabricated by selective laser melting and electron beam melting and their effects on mechanical properties. 2013.
- [138] N Eschner, L Weiser, B Häfner, and G Lanza. Development of an acoustic process monitoring system for selective laser melting (slm). In *Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium, Austin, TX, USA*, pages 13–15, 2018.
- [139] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer, 2015.
- [140] Emre Çakir and Tuomas Virtanen. End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2018.
- [141] Robert Müller, Fabian Ritz, Steffen Illium, and Claudia Linnhoff-Popien. Acoustic anomaly detection for machine sounds based on image transfer learning. *arXiv preprint arXiv:2006.03429*, 2020.
- [142] Arun Solanki and Sachin Pandey. Music instrument recognition using deep convolutional neural networks. *International Journal of Information Technology*, pages 1–10, 2019.
- [143] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [144] Thingiverse - Digital Designs for Physical Objects. 2018 (Zugriff am 13.01.2023).
- [145] Admesh. <https://github.com/admesh/admesh>. Zugriff am 21.04.2024.
- [146] Monika Teliskova, Jozef Torek, Tomas Cmorej, Marek Kocisko, and Jaroslav Petrus. Adjustments of reprop type printer workbench. In *Industrial Engineering and Applications (ICIEA), 2017 4th International Conference on*, pages 15–19. IEEE, 2017.

- [147] Thomas Kramer Frederick, Thomas R Kramer, Frederick M Proctor, and Elena Messina. The nist rs274ngc interpreter-version 3. 2000.
- [148] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.
- [149] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [150] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.
- [151] Radu Bogdan Rusu and Steve Cousins. PCL API Documentation. [Online]. Available: <http://pointclouds.org/documentation/index.html>.
- [152] Zengxi Pan, Joseph Polden, Nathan Larkin, Stephen Van Duin, and John Norrish. Recent progress on programming methods for industrial robots. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8. VDE, 2010.
- [153] Ioan A Sucas, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.
- [154] Andreas Hermann, Sebastian Klemm, Zhixing Xue, Arne Roennau, and Rüdiger Dillmann. Gpu-based real-time collision detection for motion execution in mobile manipulation planning. In *2013 16th International Conference on Advanced Robotics (ICAR)*, pages 1–7. IEEE, 2013.
- [155] Sachin Chitta, Ioan Sucas, and Steve Cousins. Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.
- [156] P. Beeson and B. Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 928–935, Nov 2015.
- [157] Patrick Min. binvox. <http://www.patrickmin.com/binvox>, 2004 - 2019. Zugriff am 13.01.2023.
- [158] Fakir S. Nooruddin and Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003.
- [159] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.
- [160] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, Oct 1994.

- [161] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.
- [162] David T Coleman, Ioan A. Suçan, Sachin Chitta, and Nikolaus Correll. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. *Journal of Software Engineering for Robotics*, 2014.
- [163] Prusa mk3s. <https://cdn.prusa3d.com/content/images/product/original/3325.jpg>. Zugriff am 28.04.2023.
- [164] Anatol Locker. Raise3d n2 plus: Review of specs. <https://all3dp.com/1/raise3d-n2-plus-review-3d-printer/>. Zugriff am 27.04.2023.
- [165] Intel realsense r200. <https://reconstructme.net/wp-content/uploads/2015/11/r200.jpg>. Zugriff am 28.04.2023.
- [166] Intel realsense d435. [https://www.intelrealsense.com/wp-content/uploads/2019/02/stereo\\_DT\\_d435\\_front-cropla.png](https://www.intelrealsense.com/wp-content/uploads/2019/02/stereo_DT_d435_front-cropla.png). Zugriff am 28.04.2023.
- [167] the t.bone sc-450 microphone. [https://www.delamar.de/wp-content/uploads/2014/07/t.bone\\_sc450\\_test-750x422.jpg](https://www.delamar.de/wp-content/uploads/2014/07/t.bone_sc450_test-750x422.jpg). Zugriff am 27.04.2023.
- [168] the t.bone sc-450 microphone - datenblatt. [https://images.static-thomann.de/pics/atg/atgdata/document/specs/c\\_152310\\_datasheet\\_v4\\_de\\_online.pdf](https://images.static-thomann.de/pics/atg/atgdata/document/specs/c_152310_datasheet_v4_de_online.pdf). Zugriff am 01.05.2023.
- [169] Robot arm, technical specification - Übersicht über technische daten der verschiedenen robotertypen. [https://www.universal-robots.com/media/1827367/05\\_2023\\_collective\\_data-sheet.pdf](https://www.universal-robots.com/media/1827367/05_2023_collective_data-sheet.pdf). Zugriff am 26.02.2024.
- [170] Robot operating system (ros). [https://de.wikipedia.org/wiki/Robot\\_Operating\\_System](https://de.wikipedia.org/wiki/Robot_Operating_System). Zugriff am 26.02.2024.

# Eigene Veröffentlichungen

Dieses Verzeichnis listet alle Publikationen, bei denen der Autor dieser Dissertation entweder der Erstautor ist, oder als Co-Autor maßgeblich zu der Veröffentlichung beigetragen hat (in Form von Problemstellung, -lösung, Diskussion oder experimenteller Evaluation).

- [171] P. Becker, C. Eichmann, A. Roennau, and R. Dillmann. Automation of post-processing in additive manufacturing with industrial robots. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1578–1583, 2020.
- [172] P. Becker, N. Spielbauer, A. Roennau, and R. Dillmann. Real-time in-situ process error detection in additive manufacturing. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 426–427, 2020.
- [173] Pascal Becker. Automation and error detection in am. *TCT Magazine - 3D Printing & Additive Manufacturing Intelligence*, Volume 29 Issue 6, 2021.
- [174] Pascal Becker, Joshua Gebert, Arne Roennau, Florian Finsterwalder, and Ruediger Dillmann. Online error detection in additive manufacturing: A review. In *2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 167–175, 2021.
- [175] Pascal Becker, Etienne Henger, Arne Roennau, and Ruediger Dillmann. Flexible object handling in additive manufacturing with service robotics. In *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 121–128, 2019. Best Oral Presentation Award.
- [176] Pascal Becker, Anastasiia Maklashevskikh, Arne Rönnaun, and Rüdiger Dillmann. Multiple object detection and segmentation for automated removal in additive manufacturing with service robots. In *Intelligent Autonomous Systems 16: Proceedings of the 16th International Conference IAS-16*, pages 299–313. Springer, 2022.
- [177] Pascal Becker, Christian Roth, Arne Roennau, and Ruediger Dillmann. Acoustic anomaly detection in additive manufacturing with long short-term memory neural networks. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 921–926, 2020.
- [178] Pascal Becker, Christian Roth, Arne Rönnaun, and Rüdiger Dillmann. Porosity detection in powder bed fusion additive manufacturing with convolutional neural networks. In *2022 IEEE 9th International Conference on Industrial Engineering and Applications (ICIEA)*. Best Oral Presentation Award.

- [179] Georg Heppner, Felix Mauch, Stefan Scherzinger, David Timmermann, Pascal Becker, Stefan Ulbrich, Arne Rönnau, Peter Heiligensetzer, and Fabian Fürst. Fla<sup>2</sup>ir—flexible automotive assembly with industrial co-workers. *Bringing Innovative Robotic Technologies from Research Labs to Industrial End-users: The Experience of the European Robotics Challenges*, pages 97–126, 2020.
- [180] Stefan Scherzinger, Pascal Becker, Arne Roennau, and Rüdiger Dillmann. A practical dmps implementation for skill creation and teleoperation with assistive manipulators, 2022.
- [181] Stefan Scherzinger, Pascal Becker, Arne Rönnau, and Rüdiger Dillmann. Motion macro programming on assistive robotic manipulators: Three skill types for everyday tasks. In *20th International Conference on Ubiquitous Robots (UR 2023)*, 2023.
- [182] Stefan Scherzinger, Jakob Weinland, Robert Wilbrandt, Pascal Becker, Arne Roennau, and Rüdiger Dillmann. A walking space robot for on-orbit satellite servicing: The recobot. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 2231–2237, 2022.
- [183] J Camilo Vasquez Tieck, Pascal Becker, Jacques Kaiser, Igor Peric, Mahmoud Akl, Daniel Reichard, Arne Roennau, and Rüdiger Dillmann. Learning target reaching motions with a robotic arm using brain-inspired dopamine modulated stdp. In *2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing (ICCI\* CC)*, pages 54–61. IEEE, 2019.

# Studentische Arbeiten

Dieses Verzeichnis listet studentische Arbeiten, die durch den Autor dieser Dissertation im Rahmen seiner Forschung ausgeschrieben und betreut wurden. Dies beinhaltet die maßgebliche Vorgabe der Problemstellung, Diskussion der Arbeit, sowie Randvorgaben zur Lösung, Visualisierung und experimentellen Evaluation.

- [184] Iain Ahrens. Conception and Development of G-code Manipulator to Improve Grasp Stability of 3D Printed Models. Bachelorarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2019. Betreuung durch Pascal Becker.
- [185] Christian Eichmann. From STL to Robot Path: Generating Post-Processing Trajectories for 3D Printed Objects. Masterarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2019. Betreuung durch Pascal Becker.
- [186] Joshua Gebert. Bildbasierte Fehlerklassifikation in der in-situ Prozessüberwachung mit tiefen neuronalen Netzwerken. Masterarbeit, Hochschule Karlsruhe - University of Applied Science, Karlsruhe, Germany, 2021. Betreuung durch Pascal Becker.
- [187] Etienne Henger. Automatisierung des Entnahmeprozesses für ein FDM 3-D-gedrucktes Bauteil. Bachelorarbeit, Hochschule Karlsruhe - University of Applied Science, Karlsruhe, Germany, 2018. Betreuung durch Pascal Becker.
- [188] Jonas Keller. In-situ Überwachung durch bildbasiertes maschinelles Lernen in der additiven Fertigung. Bachelorarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2020. Betreuung durch Pascal Becker.
- [189] Anastasiia Maklashevskikh. Objekterkennung und -segmentierung für die automatisierte Entnahme mehrerer Objekte in der additiven Fertigung. Bachelorarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2020. Betreuung durch Pascal Becker.
- [190] Christian Roth. Akustische Fehlererkennung im FDM 3D-Druck mithilfe von maschinellem Lernen. Bachelorarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2019. Betreuung durch Pascal Becker.
- [191] Qyiao Shi. 3D-Print Object Grasping with Schunk SVH Hand. Masterarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2019. Betreuung durch Pascal Becker.
- [192] Niklas Spielbauer. Optische Prozessüberwachung bei Schmelzsichtdruckverfahren. Bachelorarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2018. Betreuung durch Pascal Becker.



- [193] Robert Wilbrandt. Milling Strategy for Support Structure Removal in Additive Manufacturing. Masterarbeit, KIT Karlsruher Institut für Technologie, Karlsruhe, Germany, 2022. Betreuung durch Pascal Becker.