Machine Learning for Camera-Based Monitoring of Laser Welding Processes

Zur Erlangung des akademischen Grades einer

DOKTORIN DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)
angenommene

DISSERTATION

von

Julia Hartung, M.Sc.
geb. in Schramberg

Tag der mündlichen Prüfung:	20. September 2023
Hauptreferent:	Prof. Dr.-Ing. Michael Heizmann, KIT
Korreferent:	Prof. Dr.-Ing. Thomas Längle, KIT

Preface

I want to express my sincere gratitude to all those who contributed to completing this doctoral thesis during my work at Trumpf Laser GmbH in Schramberg in collaboration with the Institute of Industrial Information Technology of the Karlsruhe Institute of Technology.

First and foremost, I extend my deepest appreciation to my supervisor, Prof. Dr. Michael Heizmann. His guidance, unwavering support, and invaluable insights have been instrumental in shaping this research and my academic journey.

I am equally grateful to Dr. Andreas Jahn for his exceptional mentorship. His stimulating discussions, brilliant ideas, and consistent encouragement have played a pivotal role in the development of this thesis. I extend my thanks to all my colleagues at TRUMPF Laser GmbH, who have contributed to this thesis through their insightful discussions, shared knowledge, and collaborative spirit. I am also grateful to the students who supported my thesis with their dedicated work.

Furthermore, I thank Prof. Dr. Thomas Längle for serving as the co-referent for this thesis.

On a personal note, I extend my heartfelt thanks to my family. Their unwavering support, belief in my abilities, and encouragement have made my pursuit of this academic endeavor possible. To my friends, I am grateful for providing a needed balance while writing this work. Their companionship and camaraderie have been a source of strength. Lastly, but by no means least, I would like to thank Fabian Ziegler. His support, understanding, and patience have been a constant source of motivation, and I am grateful for his presence in my life.

Abstract

The increasing use of automated laser welding processes causes high demands on process monitoring. The aim is to ensure a high joining quality and to detect faults in the earliest stage possible. By using machine learning methods, more cost-effective and, in the optimal case, already installed sensors can be used to monitor the entire process.

This work demonstrates methods that use a camera mounted on the focussing optics coaxial with the laser beam to perform pre-, in-, and post-process monitoring of welding processes. The work uses the joining process of copper wires to produce formed coil windings to illustrate the methods. Due to the geometry of the joining parts and the material properties of copper, the application presents challenges in detecting the component position and in the actual joining process. The pre-process monitoring includes optimizing component position detection by a deep convolutional neural network (CNN). In addition, a shape check of the detected parts contributes to monitoring pre-processing steps and preventing welding defects. In-process monitoring focuses on the detection of spatter in the camera image, as this serves as an indicator of an unstable process. Machine learning algorithms perform semantic segmentation, differentiating between plume, process light, and material ejections without hardware modification. Finally, different approaches are shown for post-process quality assessment. Besides extracting information about the size and shape of the weld surface from the camera image, a CNN-based algorithm reconstructs the weld's height information. Considering the height map, rule-based algorithms evaluate the quality of the welds. This procedure enables conclusions about individual defective contacts and the possibility of reworking the faulty welds.

All algorithms consider the integrability into industrial processes. These challenges include a small database, limited industrial manufacturing inference hardware, and user acceptance.

Zusammenfassung

Der zunehmende Einsatz automatisierter Laserschweißprozesse stellt hohe Anforderungen an die Prozessüberwachung. Ziel ist es, eine hohe Fügequalität und eine frühestmögliche Fehlererkennung zu gewährleisten. Durch die Verwendung von Methoden des maschinellen Lernens können kostengünstigere und im Optimalfall bereits vorhandene Sensoren zur Überwachung des gesamten Prozesses eingesetzt werden.

In dieser Arbeit werden Methoden aufgezeigt, die mit einer an der Fokussieroptik koaxial zum Laserstrahl integrierten Kamera eine Prozessüberwachung vor, während und nach dem Schweißprozess vornehmen. Zur Veranschaulichung der Methoden wird der Kontaktierungsprozess von Kupferdrähten zur Herstellung von Formspulenwicklungen verwendet. Die vorherige Prozessüberwachung umfasst eine durch ein faltendes neuronales Netz optimierte Bauteillagedetektion. Durch eine Formprüfung der detektierten Fügekomponenten können zudem vorverarbeitende Schritte überwacht und die Schweißung fehlerhafter Bauteile vermieden werden. Die prozessbegleitende Überwachung konzentriert sich auf die Erkennung von Spritzern, da diese als Indikator für einen instabilen Prozess dienen. Algorithmen des maschinellen Lernens führen eine semantische Segmentierung durch, die eine klare Unterscheidung zwischen Rauch, Prozesslicht und Materialauswurf ermöglicht. Die Qualitätsbewertung nach dem Prozess beinhaltet die Extraktion von Informationen über Größe und Form der Anbindungsfläche aus dem Kamerabild. Zudem wird ein Verfahren vorgeschlagen, welches anhand eines Kamerabildes mit Methoden des maschinellen Lernens die Höhendaten berechnet. Anhand der Höhenkarte wird eine regelbasierte Qualitätsbewertung der Schweißnähte durchgeführt.

Bei allen Algorithmen wird die Integrierbarkeit in industrielle Prozesse berücksichtigt. Hierzu zählen unter anderem eine geringe Datengrundlage, eine begrenzte Inferenzhardware aus der industriellen Fertigung und die Akzeptanz beim Anwender.

Contents

Nomenclature

Common Abbreviations

Abbreviation	Description
e. g.	For example
i. e.	In other words
1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
Acc	Accuracy
AdaGrad	Adaptive Gradient Algorithm
ADAM	Adaptive Moment Estimation Algorithm
AG	Attention Gate
AI	Artificial Intelligence
AL	Active Learning
BS	Batch Size
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planing
CCD	Charged-Coupled Device
CE	Cross Entropy
CGAN	Conditional Generative Adversarial Network
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network
Conf	Confidence
Conv	Convolution
CPU	Central Processing Unit
Cu	Cuprum (Copper)
Cu-ETP	Electrolytic Tough Pitch Copper

Abbreviation	Description
Cu-FRHC	Fire Refined Tough Pitch High Conductivity Copper
Cu-OF	Oxygen Free Copper
DCGAN	Deep Convolutional Generative Adversarial Network
DIN EN	German Institute for Standardization, European Standard (ger. Deutsches Institut für Normung, Europäische Norm)
DL	Deep Learning
DSC	Dice Similarity Coefficient
DW	Defective Weld
ECE	Expected Calibration Error
ELU	Exponential Linear Unit
FCNN	Fully Connected Neural Network
FD-OCT	Fourier Domain Optical Coherence Tomography
FL	Focal Loss
FN	False Negatives
FP	False Positives
FPS	Frames Per Second
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GT	Ground Truth
GW	Good Weld
IoU	Intersection over Union
IP	International Protection
IR	Infrared
JC	Jaccard Coefficient
LED	Light-Emitting Diode
lReLU	Leaky Rectified Linear Unit
LWM	Laser Welding Monitor
MAE	Mean Absolute Error
MCD	Monte Carlo Dropout
ML	Machine Learning
MSE	Mean Squared Error
NFL	No-Free-Lunch Theorem
NIR	Near Infrared
OCT	Optical Coherence Tomography

Abbreviation	Description
OEM	Original Equipment Manufacturer
ONNX	Open Neural Network Exchange
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RMSProp	Root Mean Square Propagation
ROI	Region Of Interest
SAE	Stacked Autoencoder
SD-OCT	Spectral Domain Optical Coherence Tomography
SD	Standard Deviation
SDU-Net	Stacked Dilated U-Net
SFS	Shape From Shading
SGD	Stochastic Gradient Descent
SSD	Solid State Drive
SSI	Structural Similarity Index
TD-OCT	Time Domain Optical Coherence Tomography
TN	True Negatives
ToF	Time of Flight
TP	True Positives
UV	Ultraviolet
VDE	Association for Electrical, Electronic and Information Technologies (ger. Verband der Elektrotechnik Elektronik und Informationstechnik e.V.)
VIS	Visible spectrum
wCE	Weighted Cross Entropy
WGAN	Wasserstein Generative Adversarial Network

Symbols

Latin Letters

Symbol	Description
A	Absorption ratio
$b, \boldsymbol{b}$	Bias of a neuron, vector of the bias values

Symbol	Description
c	Number of classes / number of channels
c_{WD}	Weight clipping factor of the WGAN
d	Distance
E	Energy
f	Frequency
$\boldsymbol{h}$	Hidden layer of a neural network
H	Structural element of a morphological filter
$\mathcal{H}$	Confidence metric based on entropy
J	Cost function
$\tilde{J}$	Regularized cost function
k	Kernel size of a convolutional layer
$\boldsymbol{K}$	Kernel matrix
$\boldsymbol{l}$	Latent space of a neural network
$\mathcal{L}$	Loss function
p_{data}	Data generating distribution
P	Laser power
P_A	Absorption power
P_R	Reflexion power
P_T	Transmission power
P_V	Heat dissipation power
r	Dilation rate of a convolutional layer
R	Reflexion ratio
s	Strides of the convolution along the height and width
t	Time
$\mathcal{U}$	Model uncertainty
v	Velocity
$w, \boldsymbol{w}, \boldsymbol{W}$	Model parameters, vector/matrix of model parameters
$x, \boldsymbol{x}, \boldsymbol{X}$	Input value, vector/matrix of input values
$\mathbb{X}, \mathbb{Y}$	Set of input/ target value samples
$y, \boldsymbol{y}, \boldsymbol{Y}$	Target value, vector/matrix of target values
$\hat{y}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{Y}}$	Predicted value, vector/matrix of predicted values
$\boldsymbol{z}$	Noise vector
$\varnothing$	Diameter

Greek Letters

Symbol	Description
α	Weighting parameter
α_a	Angle
θ	Trainable parameter of the neural network
ϵ	Learning rate
η	Smoothing factor
γ	Focusing parameter of the focal loss
λ	Wavelength
μ	Mean
ω	Penalty term
ϕ	Non-linear transformation
σ	Standard deviation
τ	Training iteration of the neural network

Superscripts

Index	Description
$(\bullet)^*$	Value that minimizes a function
$(\bullet)^l$	Layer of a neural network
$(\bullet)^\top$	Transposed

Subscripts

Index	Description
$(\bullet)_\mathrm{i}$	Element or vector i of a data set
$(\bullet)_\mathrm{j}$	Element j of vector
$(\bullet)_\mathrm{j,k}$	Element j, k of a matrix

Mathematical Operators

Operator	Description
$*$	Convolution operation
$\ominus$	Erosion operation
$\oplus$	Dilation operation
$\circ$	Opening operation
$log()$	Natural logarithm
$\nabla_x y$	Gradient of y with respect to x
$\exp()$	Exponential function with base e
$f(x;\theta)$	Function with input x, parameterized by θ
$f(x)$	Simplified representation of $f(x;\theta)$
$\hat{f}(x)$	Function approximation
$D(x;\theta)$	Discriminator with input x, parameterized by θ
$G(x;\theta)$	Generator with input x, parameterized by θ
$P(x)$	Probability distribution over a discrete variable
$p(x)$	Probability density over a continuous variable or a variable whose type was not specified
$\mathbb{E}_{x \sim P}[f(x)]$	Expectation for $f(x)$ with respect to $P(x)$

1 Introduction

Machine learning (ML) is a subfield of artificial intelligence (AI) in which systems learn from data and recognize patterns and relationships without being explicitly programmed. AI generally includes methods that can perform tasks that usually require human intelligence. The scientific discipline of AI traces back to the 1950s when Turing proved that a computing machine could perform cognitive processes [161]. The term "artificial intelligence" itself was introduced by John McCarthy at a conference on the campus of Dartmouth College in 1956. In the following years, several breakthroughs attracted media attention, for example, in 1996 when the world chess champion was defeated by the chess computer "Deep Blue". However, there was a lack of data and computing power for a long time, which is why there was no general technological breakthrough. This changed around 2011 when highly efficient processors and graphics cards significantly accelerated the calculation of algorithms. Technological leaps in hardware and software paved the way for AI to enter everyday life and enabled ordinary consumers to access the programs. Examples of typical applications include machines that respond meaningfully to natural language, recognize faces and objects or make custom-fit suggestions, e. g., about music tracks, videos, or products.

In the context of Industry 4.0, which describes the digital transformation of production, ML applications are also becoming the focus of industrial manufacturing [65]. A large number of studies show the economic potential of ML algorithms. In 2018, a study by the McKinsey Global Institute estimated that the entire field of AI will trigger a global annual growth spurt in the gross domestic product of 1.2% on average by 2030. This increase would exceed the growth spurts of the steam engine and industrial robots [105]. Recent studies by McKinsey [106] and the industry association Bitkom [12] also show that this trend will continue.

Nevertheless, the proportion of companies using AI in some form is rising very slowly. In 2021 the share increased from 8% to 9%, based on a study by Bitkom [12]. According to the German Federal Ministry of Economics and Technology, the level of digital readiness is generally weak in many companies. Besides the fact that most companies are not yet using any AI applications, many are just starting to think fundamentally about what digital products or manufacturing methods might look like. According to a survey by the German Association of Human Resources Managers, only 40% of the companies surveyed have even adopted a digital strategy by 2022 [43].

The biggest obstacles to use AI in companies are a lack of human ressources and the availability of too few data. Financial lack is also a factor slowing down the use of AI systems. Many companies hesitate to invest in new technologies and business models because the outcome is only vaguely defined at the beginning. Developing AI algorithms requires, in most cases, investment in new hardware and data generation. This hurdle is often still too large. In addition, employees' lack of acceptance and trust in AI are mentioned as obstacles [12, 124]. Thus, there are still few AI solutions in the industry that can be used immediately and are ready for the market. However, especially for small and medium-sized companies, getting started to develop their own AI algorithms is a major challenge [43].

Due to the great potential offered by AI applications, it must be ensured to take advantage of opportunities. Germany is already dropping behind in AI in an international comparison [12]. In everyday life, the strength of algorithms is evident in many applications, and it is no longer possible to imagine life without them. They are taking up more and more space in almost all areas of life. This change will also have an impact on industrial manufacturing. Since AI is a key technology, it will lead to competitive advantages or a downturn. The potential to make value-added processes more flexible and efficient with the help of AI applications is enormous [43].

1.1 Industrial Use of Machine Learning

A trend is that the average size of artificial neural networks is increasing massively. Since introducing hidden units around 1960, it has doubled approximately every 2.4 years [56]. Hidden units are elements of a neural network that belong to neither the input nor the output layer but are intermediate layers. These units compute the input from the previous layer and pass the results to the next layer until the final output layer is reached. This growth is accelerated by faster computers, better GPUs, software infrastructures that enable distributed computing, and the availability of larger data sets. Figure 1.1 shows the evolution using popular network architectures as examples over the years.

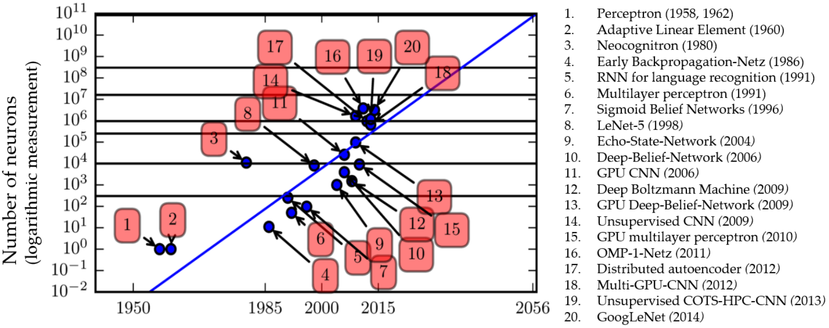


Figure 1.1 Growing size of neural networks over time. 1.[135, 136], 2.[169], 3.[46], 4.[137], 5. [131], 6. [10], 7. [140], 8.[91], 9.[74], 10. [64], 11. [18], 12. [139], 13. [129], 14.[75], 15. [21], 16. [24], 17.[90], 18. [86], 19. [25], 20. [158] (Figure from [56])

Not only has the size of networks increased over the years but so has the amount of training data. In the age of big data and the increasing digitization of society, much more training data is available. Even since 2016, the rule of thumb is that a supervised deep learning algorithm would generally perform acceptably with about five thousand labeled training data per label and outperform human performance with a training set of at least ten million data [56].

However, in industrial manufacturing, especially directly at production plants, neither powerful hardware for machine learning nor large labeled data sets are often available. In many cases, collecting and labeling the data is a significant obstacle that is not overcome for reasons of time and capacity. In addition, the data sets are often unbalanced. Initiating a faulty process run with scrap production just to train the model is rarely economically justified. Also, the uncertainty of whether and to what extent this effort is worthwhile plays a crucial decision factor. These are a few reasons deep learning algorithms have been less successful in this area than in other areas.

1.1.1 Requirements in Industrial Manufacturing

Industrial applications, primarily industrial manufacturing, often place different requirements on the algorithms compared to, e. g., face or speech recognition. For many applications, too much generalization is not necessary and sometimes even a hindrance. In addition, the data variance is much lower due to defined constraints. If significant deviations occur in the data, this usually indicates a fault in the application. While the algorithm must report this as an error, it does not need to be able to process the data holistically. Unlike facial recognition, which has to work in different environments, light conditions, and recording angles, the environment within a manufacturing process is usually well defined. Therefore, there is no need to follow the general trend toward larger models and more data in the manufacturing environment. Instead, small, lean network architectures are more effective in some cases.

According to surveys, many users see the problem of ML in industrial manufacturing not in the models or model architectures but primarily in the data. Because, in practice, companies would have to deal with entirely new data for almost every application [124]. This statement also shows the different requirements for the algorithm. Instead of a large, generalizing algorithm that covers many use cases, it can be valuable to train algorithms for smaller, defined tasks. For example, in the case of an original equipment manufacturer (OEM) supplier, many customers do not want their data to flow into algorithms also used by other customers. Especially not if components are visible and thus perhaps features that distinguish them from the competitors. Therefore, depending on the use case, focusing on individual algorithms for each customer rather than using one large model makes sense. Matching this observation, the research advisory board of the Industry 4.0 platform and the German

Academy of Science and Engineering have also identified the development of machine learning algorithms with excellent performance on small data sets and easy transferability as a research and development need for the successful implementation of Industry 4.0 in 2022 [65]. This aspect will be discussed in more detail in Chapter 3 of this work.

Another common requirement is real-time capability on industrial hardware since the processes are critical in cycle time. Network architectures that are too large require much time on the one hand and computationally powerful inference hardware on the other. This again presents a hindrance to the use of ML algorithms. This topic will be discussed in more detail in Chapter 4.

In the best case, using suitable ML algorithms can save time and more expensive sensor technology. Often, with simpler hardware and suitable algorithms, the same, or at least approximately the same, features can be detected as with complex sensor technology. Because neural networks capture multilayered data correlations, they can often extract features from the data that are not directly visible to humans. For example, Chapter 5 shows how a camera and a neural network that computes the height data can replace a height scanner.

1.1.2 User Acceptance

Besides the lack of specialists and limited availability of data, insufficient trust in ML systems also plays a decisive role, as mentioned at the beginning. According to a study by Bitkom, around half of the companies interviewed are concerned about the poor traceability of the results and possible application errors of ML algorithms. In addition, significant risk factors are that errors in programming and learning databases are challenging to detect [12]. These aspects are relevant in the area of in-house developments but also for purchased ML solutions. In order to be able to sell products that contain ML algorithms, customer acceptance must also be kept in mind.

The uncertainty concerning trust in the decision-making process of ML-driven systems is partly due to the often inconsistent quality of the training data [43]. This offset partly arises from the knowledge loss between application and AI experts. The AI experts know the required format of the labeled data for training the algorithms and can estimate on which data basis the network can learn useful features. However, they do not know the actual application in detail and, thus, cannot always correctly assign errors to classes, for example. On the other hand, the application experts and users of the ML algorithm do not know the training data in detail and therefore have little confidence in the database on which the ML algorithm is based. This gap between AI experts and application experts still leads to great uncertainty. The topic is discussed in Chapter 3.6, and a possible solution approach is given.

The danger to people and property posed by the algorithm's decision must also be considered. The European Commission drafted the world's first legal framework for AI in 2021 [40]. These regulations are intended to enable transparency and minimum requirements and prevent misunderstandings. AI approaches can be divided into different risk classes. Even well-known applications from everyday life can be classified like this. For example, an incorrect recognition of voice input for a music request has minor consequences, while an AI-based control of an autonomously driving vehicle poses much more significant risks. Therefore, especially at the beginning of the industrial use of AI algorithms, selecting applications with minimal risk is recommended. Examples of this are applications that only contribute to decision support. This means the data is processed by an algorithm and then evaluated by a human, who ultimately makes the decision. For algorithms whose results flow directly into an automated production process, monitoring or controlling the output by humans or, e. g., by knowledge-based systems, is appropriate. This aspect is also discussed in more detail in the following chapters.

1.2 Laser Welding and its Process Monitoring

The laser market was estimated at USD 16 705.2 million in 2021 and is expected to continue growing [113]. Besides the communications segment, material processing using lasers has a significant role in this market. Although lasers were initially used mainly for cutting applications, a considerable and growing proportion of lasers are now used for joining materials [112]. The laser beam has already replaced mechanical manufacturing processes in many areas when welding metallic and nonmetallic workpieces. Digitization in the context of Industry 4.0 requires tools that work quickly, directly, and flexibly and thus can be automated. All this applies to the laser beam. As a result, the increasing trend toward automation and continuous progress has significantly driven the use of laser welding technology.

Furthermore, improved productivity and cycle time reduction are increasingly crucial in today's industrial manufacturing. For example, in the automotive industry, where the total length of welded seams can add up to more than 50 m per car, it is essential to minimize processing time through high welding speeds, and automation [112]. Another strong trend is the increasing individualization of products. Marketing departments always want to offer potential customers a product tailored to their wishes or surprise them with special editions. At the same time, production planners groan when they constantly have to produce new variants and small batch sizes. Laser light brings freedoms to this area that mechanical processes cannot offer. As a laser welding process handles new shapes and contours through program changes, it offers the potential to change over processes at high frequencies. The laser beam also enables precise work due to the accurate and precise energy input. Furthermore, only minimal structural changes occur to the surrounding material due to the small heat-affected zone. This way, even the finest structures can be implemented with high process reliability and reproducibility.

Laser welding is used in many fields, ranging from high-precision micro welding of medical devices to fully automated laser welding in the automotive industry [112]. However, laser welding is a complex process. Many parameters influence the quality of the weld. Thus, monitoring the welding quality and checking for welding defects is necessary. Firstly, checking the components to be welded before welding is helpful, as this can avoid expensive follow-up costs in case of doubt. In addition, the weld seam must be checked after welding to ensure high product quality. Some weld defects are difficult or impossible to detect in the solidified weld after welding, which also argues for monitoring during the process. Especially with the trend towards short cycle times and complete automation, process monitoring systems for laser material processing are becoming increasingly important.

1.3 Contribution and Organisation of the Work

When the first laser was developed in the 1960s, it was described as a tool looking for an application [160]. In the meantime, it has become clear how versatile the laser can be and how, among other things, it has found its perfect place as a flexible tool in a world shaped by digitalization. The situation is similar today with AI algorithms. Again, there is a powerful tool whose use in many areas is already apparent but still waiting for proper implementation and large-scale use in industrial manufacturing, especially in quality monitoring.

Image processing is already an essential part of automation technology. The more progress is made in Industry 4.0, which is accompanied by automation and modularity, flexibility, and individualization, the more image processing is needed. However, this technology also reaches its limits, especially when products are individualized and processes frequently change over. As explained in the previous chapter, this is often the case with laser welding. For such applications, image processing solutions are often not adaptive enough. In contrast, deep learning algorithms offer new possibilities for such problems. Data-driven development, which is no longer strictly oriented to defined algorithm sequences, allows rapid adaptation and greater flexibility.

The work aims to apply deep learning methods in the quality assurance of laser welding processes where other algorithms reach their limits. It uses the example of the joining process of copper wires for the production of formed coil windings, the so-called hairpin welding. The focus of the analysis is on the data of a camera sensor, which is mounted on-axis on the focusing optics. In temporal terms, the work considers pre-process, in-process, and post-process monitoring. Observing all process stages enables the earliest possible fault detection and more stable overall monitoring. These temporal phases also determine the structure of the work.

The following section gives a brief overview of each chapter's content and scientific work.

Fundamentals Chapter 2 introduces basic concepts that are fundamental for the methods of this work. First, the principles of deep learning and the structure and evaluation metrics of deep learning models are explained. Second, the basics of laser welding and the process risks are summarized. Finally, a short introduction to the contacting process of copper wires to produce shaped coil windings concludes the chapter. This application is used to illustrate the proposed methods throughout the work.

Pre-Process Chapter 3 describes an extension of the upstream steps before the actual laser welding process. These steps include the detection of the component position using a camera image. In the image, the exact position and orientation of the component are captured and passed on to the laser control system so that welding is always performed at the correct place. Extending the algorithm by pre-processing the camera image with semantic segmentation by a deep convolutional neural network highlights the pixels belonging to the component. This additional step makes the algorithm more robust, and it can be adapted more quickly to changing processes. Accurate detection of the joining parts also enables verification of their shape and size, which prevents the welding of defective parts and allows monitoring of the pre-processing steps. This chapter discusses the selection of a suitable model architecture for semantic segmentation and a single- and multi-stage approach for joining part recognition depending on the database. Furthermore, the machine learning process integration into industrial manufacturing and the automation and support of the model generation will be addressed. An important aspect is, among other things, a method for optimizing and accelerating the data labeling process. There is also still a need for research and development in hybrid solution approaches and the verification and validation of the systems [65]. These points are also addressed in this chapter.

In-Process The fourth chapter deals with in-process monitoring during laser welding. Due to the laser-material interactions occurring during welding, energy is emitted in various forms. From the emissions of welding, process signals can be measured with the help of suitable sensors, leading to the detection of welding defects during the process. The focus of this chapter is on the use of a camera as a sensor for in-process monitoring. The analysis refers to the occurrence of spatter in the process, which is considered an indicator of process instability. Machine learning algorithms perform semantic segmentation of the images, which allows clear differentiation between plumes, process lights, and material ejection without hardware modification. It also examines spatter size and velocity to derive the predictive power of different camera acquisition frequencies and the associated monitoring of a spattering tendency. Some results of the camera-based spatter monitoring using deep learning methods have been published in *Applied Science* (Hartung et al. [184]).

Post-Process The post-process inspection presented in Chapter 5 aims to evaluate the weld quality after solidifying based on a camera image. The extraction of information about the size and shape of the weld allows a conclusion about the quality. The chapter demonstrates a deep-learning-based approach to reliably highlight the weld with pixel accuracy in the image. In addition, this chapter presents a reconstruction algorithm that computes height information based on a single camera image. The reconstruction uses machine learning methods. In both approaches, a knowledge and rule-based algorithm follows the machine learning algorithm to evaluate the quality. The different techniques are explained in detail in Chapter 5. The results are evaluated and compared. The 3D reconstruction procedure and comparisons with state-of-the-art have been published in *Sensors* (Hartung et al. [185]). Furthermore, the evaluation of the calculated height data in terms of quality assurance compared to measured height data and a purely image-based approach has been presented at the *Forum Bildverarbeitung 2022* (Hartung et al. [181]) and a further comparison was published in *tm - Technisches Messen* (Hartung et al. [182]).

Conclusion and Outlook The final Chapter 6 summarises the main results and findings of the work. Furthermore, possible future research in this field is proposed.

2 Fundamentals

The following chapter discusses basic concepts fundamental to the methods proposed in this work and contributes to a better understanding. The first part explains machine learning, how it works, and the setup of the used network architectures. Afterward, the chapter shows the basics of laser welding and the design of the welding station before it connects to process instabilities and process monitoring.

2.1 Deep Learning

Deep Learning (DL) is a special information processing method and an ML subfield. The fundamental difference between ML and traditional programming is that a program does not have to be created step by step based on input data and rules. Instead, an algorithm defines the regulations based on the data itself.

Mitchell [111] describes the ML task as a computer program that learns from experience E concerning a class of tasks T and a performance measure P if its performance on the tasks in T, as measured by P, improves with experience E. In other words, ML is the study of computer algorithms that allow computer programs to improve automatically through experience. However, there is no strict and single definition for either task, performance evaluation, or experience. The definition of each of these values can be very different.

In general, the task T describes what the network should learn. The ML algorithm is represented by the function $y = f(\boldsymbol{x})$, where the input features $\boldsymbol{x}$ are mapped to the desired target value y. In the case of classification, the target value is defined by $y = \{1, ..., c\}$, where c is the number of classes. However, the target value can also represent the prediction of an expected value (regression), a density estimation, or other quantities. It is essential for the algorithm that the target information is implicitly derivable from the input data and that similar data also have similar results. The input is usually represented as a vector $\boldsymbol{x} \in \mathbb{R}^n$, where each element x_i represents a feature. Depending on the database and ML model, pre-processing of the input data is required and, in some cases, an additional reduction of the data complexity. The algorithm learns and optimizes the relationship between the features $\boldsymbol{x}$ and the target value y to improve the performance score P. The choice of evaluation metric depends on the task T. For example, the classification uses the correct classification rate (accuracy) or the error rate for evaluation. With each performance score P, the model develops further experience E that contributes to the success of the task. Algorithms can be divided into supervised and unsupervised ML methods. In supervised methods, labeling informations are available for each input, and the algorithm learns the relationship between the target value and the input features. Unsupervised methods are applied to data without labeling information. The goal is, for example, to learn the probability distribution from which the input data set was generated based on the input features (density estimation, noise reduction) or to summarize the data based on structures (clustering). In addition, there are intermediate stages between supervised and unsupervised learning, e. g., so-called semi-supervised learning, in which some target values contain labeling information while others do not. Another area of ML methods is reinforcement learning. These algorithms operate not only on a database but learn the optimal behavior in an environment based on a defined feedback loop. The algorithm uses the feedback to extend its experience to learn the optimal way to achieve a given goal.

To form complex structures, DL algorithms combine simple concepts and functions. The great advantage here is that unstructured data can also be processed. While the performance of simple ML algorithms depends very much on the representation (i. e., the presentation or preparation) of the output data, DL algorithms extract the relevant features within chained functions.

2.1.1 Deep Learning Model

Deep feedforward networks are described by $y = f(\boldsymbol{x}; \boldsymbol{\theta})$. The parameters $\boldsymbol{\theta}$ are defined and adjusted to find the best function approximation

f^*. A linear model cannot always capture the interaction between variables because the algorithm can only linearly map the input to the output. Thus, the model must be extended to represent nonlinear functions of $\boldsymbol{x}$. The linear functions are therefore applied to transformed input data $\phi(\boldsymbol{x})$, where ϕ is a nonlinear transformation. ϕ can be considered as a set of features to describe $\boldsymbol{x}$.

2.1.1.1 Activation Function

Most neural networks use an affine transformation driven by learned parameters. An invariant nonlinear function, the so-called activation function, follows this transformation.

The transformation of the output unit is often defined differently from the hidden layers. Standard functions are sigmoid and softmax activation. The sigmoid output unit for a network with hidden layer $\boldsymbol{h} = f(\boldsymbol{x}; \boldsymbol{\theta})$, a bias factor b and the assignment parameter to the output $\boldsymbol{w}$ is defined by $\hat{\boldsymbol{y}} = \phi_{\text{sigmoid}}(\boldsymbol{w}^\top \boldsymbol{h} + b)$, where

$$\phi_{\text{sigmoid}}(z) = \frac{1}{1 + \exp(-z)}. \tag{2.1}$$

It uses a linear shift to calculate $z = \boldsymbol{w}^\top \boldsymbol{h} + b$. The sigmoid function is often used to determine the parameter for ϕ of a Bernoulli distribution since its range of values is $[0, 1]$. This function is used, e. g., in a two-class problem in which the neural network predicts the probability $P(y = 1|\boldsymbol{x})$. Similar to the sigmoid function is the hyperbolic tangent function

$$\phi_{\text{tanh}}(z) = 2\phi_{\text{sigmoid}}(2z) - 1. \tag{2.2}$$

This represents a shifted and stretched version of the sigmoid, covering the range $[-1, 1]$. Conversely, the softmax function is suitable as the output of a classifier for a classification problem with c classes. It provides a probability distribution over a discrete variable with c possible values. Thus, the function represents a kind of generalization of the sigmoid function. In addition to the condition that each element $\hat{y}_i$ must be in the range $[0, 1]$, the sum of the entire vector must add up to 1. Thus, the approach used in the Bernoulli distribution is applied to the Multinoulli distribution. First, a linear layer predicts the non-normalized log-probabilities with $\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{h} + \boldsymbol{b}$, where $z_i = \log \tilde{P}(y = i|\boldsymbol{x})$. Then $\boldsymbol{z}$ is exponentiated and normalized to obtain $\hat{y}$, giving

$$\phi_{\text{softmax}}(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}). \tag{2.3}$$

The most commonly used activation function for the hidden layers is the rectified linear unit (ReLU) [1, 116]

$$\phi_{\text{relu}}(z) = \max\{0, z\}. \tag{2.4}$$

Negative values always result in zero for the ReLU, while it results in a linear mapping for the other values. An extension of the function is the leaky ReLU [99], which is defined by $f_{\text{lrelu}}(z) = \max\{\alpha \cdot z, z\}$ with $\alpha \in [0, 1]$. The function has a slight slope for $z < 0$ and thus does not drop these values. However, this feature no longer guarantees a noise-robust deactivation state. Therefore, Clevert et al. [23] proposes an approach with negative values to allow mean activations close to 0 but saturates to a negative value for smaller arguments. The exponential linear unit (ELU) with $\alpha > 0$ is

$$\phi_{\text{elu}}(z) = \begin{cases} z & \text{if } z > 0, \\ \alpha(\exp(z) - 1) & \text{otherwise.} \end{cases} \tag{2.5}$$

2.1.1.2 Cost function

Depending on the task T, different cost functions are needed. In supervised learning the functions are evaluated using a defined cost function $J(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta})$, which measures the deviation between the target value y and the result $\hat{y} = f(\boldsymbol{x}; \boldsymbol{\theta}, w)$ of a data set $\mathbb{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ with n samples and the corresponding target values $\mathbb{Y} = \{y_1, ..., y_n\}$. The fitting of the parameters $\boldsymbol{\theta}$ is done to find the best function approximation f^*, with $\boldsymbol{\theta}^* = \arg\min_\theta J(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta})$. The cost functions are divided into regression models and classification models. While the regression models predict continuous values, the classification models predict an output from a set of finite categorical values. In the following, only the functions used in this work will be discussed.

A commonly used cost function from the category of regression models is the mean square error, also known as L_2 loss. This function calculates the square of the difference between the actual and predicted values and is defined as

$$J_{\mathrm{MSE}}(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x_i}; \boldsymbol{\theta}))^2, \tag{2.6}$$

for n data samples. Larger deviations are more significant with the help of squaring.

A typical cost function for classification tasks is the cross entropy

$$J_{\mathrm{CE}}(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\log\bigl(\hat{y}_{ij}\bigr), \tag{2.7}$$

where

$$y_{ij} = \begin{cases} 1 & \text{if } i_{th} \text{ element is in class } j, \\ 0 & \text{otherwise.} \end{cases} .$$

Assuming that the target vector $\boldsymbol{y}_i$ is one-hot coded, the output of classification tasks is the probability $p \in [0, 1]$ for containing the respective class of the total number of classes c. Thus, cross entropy accounts for both model uncertainty and incorrect predictions. However, an imbalance between the classes introduces bias into the process. Since the result improves enormously when the model predicts the more often represented class with higher confidence, it adjusts the parameters $\boldsymbol{\theta}$ to this class. This cost function is also used in semantic segmentation, where $\boldsymbol{Y}_j$ and $\hat{\boldsymbol{Y}}_j$ are matrices with a class probability value per pixel. In this case, the problem of unequal class ratios often occurs since, usually, a background class takes up the largest part of the image. To overcome this bias caused by unequal class ratios, weighted cross entropy defined by

$$J_{\mathrm{wCE}}(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} \alpha_j y_{ij}\log\bigl(\hat{y}_{ij}\bigr) \tag{2.8}$$

can be used. The function is defined analogously to the cross entropy, with the extension that the class weighting can be adjusted by the weighting factor $\boldsymbol{\alpha}$. The weighting can be defined by the inverse class frequency or treated as a hyperparameter.

Lin et al. [96] propose a further optimization of the cost function. In addition to class weighting, they extend the cross entropy with the modulation factor $(1 - \hat{y})^{\gamma}, \mathrm{with} \gamma \geq 1$, via which a focus is placed on more complex samples. As a result, the α-balanced focal loss is defined by

$$J_{\mathrm{FL}}(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \left(\alpha (1 - \hat{y}_{ij})^{\gamma} y_{ij} \log(\hat{y}_{ij}) \right) . \tag{2.9}$$

If an example is classified correctly, which results in $\hat{y}_{ij} \to 1$, it follows that $(1 - \hat{y}_{ij})^{\gamma} \to 0$ (down weighting). If an example is classified incorrectly, $\hat{y}_{ij} \to 0$, the value is close to 1, and the result is unaffected. The parameter γ controls the strength of the weighting. For $\gamma = 0$, $J_{\mathrm{FL}} = J_{\mathrm{wCE}}$ is valid. Lin et al. [96] recommend slightly reducing α when increasing γ.

Another cost function, proposed by Milletari et al. [109], which is often used in semantic segmentation is the Dice loss [73, 179]. The training uses a probabilistic version of the Dice similarity coefficient (DSC), which approximates it. The Dice loss is defined as:

$$\mathcal{L}_{\mathrm{Dice}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{c} \sum_{j=1}^{c} \frac{2 \sum_{k=1}^{m} y_{kj} \hat{y}_{kj} + \eta}{\sum_{k=1}^{m} (y_{kj} + \hat{y}_{j,k}) + \eta} \tag{2.10}$$

for a one-hot encoded prediction, where m is the number of pixels and c is the number of classes. η represents a smoothing factor that prevents the denominator from being zero in the case of $y_{j,k} = \hat{y}_{j,k} = 0$. This results in the cost function

$$J_{\mathrm{Dice}}(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{Dice}}(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i), \tag{2.11}$$

for n data sampels.

All cost functions are calculated on the training data and validation data during training and are assumed to have the same effect on the test data. The purpose of the optimization is $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\mathbb{X}^{\mathrm{test}}, \mathbb{Y}^{\mathrm{test}}; \boldsymbol{\theta})$.

2.1.1.3 Optimization

The gradient descent method generally solves the optimization problem and finds $\boldsymbol{\theta}^*$. The method traces the negative gradient in the given step size ϵ (learning rate). Batch gradient descent determines the error for all examples of the training data set and only then updates the model. The so-called stochastic gradient descent (SGD) updates the parameters for each training example individually. Thus, the entire data set does not need to be present in memory. Since the gradient is an expected value that can be estimated approximately with a small set of samples, it is not necessary to use all training data for the calculation. The so-called mini-batch gradient descent is the preferred method because it combines the concepts of batch gradient descent and SGD. In this method, only small subsets (mini-batch) randomly drawn from the training data are used to calculate the gradient:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \epsilon \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k). \tag{2.12}$$

The definition of the hyperparameter ϵ is usually not straightforward because the different model parameters react differently to the change. Another reason is that by randomly drawing training samples, the gradient estimator for SGD provides a noise source that does not disappear even at the minimum. As a result, the learning rate ϵ is usually linearly reduced until the defined iteration τ is reached:

$$\epsilon_k = (1-\alpha)\epsilon_0 + \alpha \cdot \epsilon_\tau, \text{ with } \alpha = \frac{k}{\tau}, \tag{2.13}$$

where k indicates the current iteration. After iteration τ is reached, the learning rate usually remains constant. Based on the problematic definition of the learning rate, there are other extensions to the algorithm. The adaptive gradient algorithm (AdaGrad) adjusts the learning rate for all model parameters individually by scaling proportionally to the square root of the sum of all previous squared values of the gradient [35]. This means that it accelerates the updating process for parameters with weak gradients and slows down the updating of the weights with large gradients. The root mean square propagation algorithm (RMSProp) [63] modifies AdaGrad by replacing the gradient accumulation with an exponentially weighted average of recent results. The history of the distant past is thus discarded, allowing for rapid convergence after detecting a convex trough. However, using the moving average introduces the new hyperparameter ρ, specifying the moving average's length scale. The adaptive moment estimation algorithm (ADAM) [83] further optimizes the algorithm. It extends the RMSProp algorithm to include a momentum directly as an estimate of the first-order gradient moment. Momentum accelerates learning, especially in the presence of solid curvature. In addition, ADAM uses bias corrections for the estimates of the first and second-order moments to account for their initialization at the origin. The algorithm updates the exponential moving averages of the gradient and the squared gradient, requiring two hyperparameters $\beta_1, \beta_2 \in [0, 1)$ that control the exponential decay rates of the respective moving averages:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \frac{\epsilon_k}{\sqrt{\hat{v}_{k+1}} + 1e^{-8}} \hat{m}_{k+1}, \tag{2.14}$$

where

$$\begin{aligned}
\hat{m}_{k+1} &= \frac{m_{k+1}}{1 - \beta_1^k}, \\
\hat{v}_{k+1} &= \frac{v_{k+1}}{1 - \beta_2^k}, \\
m_{k+1} &= \beta_1 m_k + (1 - \beta_1) \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k), \\
v_{k+1} &= \beta_2 v_k + (1 - \beta_2) \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k)^2.
\end{aligned}$$

Kingma and Ba [83] propose the values 0.9 and 0.999 for the hyperparameters β_1 and β_2, respectively. The moving averages are initialized as (vectors of) 0, resulting in moment estimates biased toward zero, especially during the first time steps and especially when the decay rates are small. Therefore, bias-corrected estimates of the values are used, leading to $\hat{m}_{k+1}$ and $\hat{v}_{k+1}$.

The gradient calculation is performed using the backpropagation [137] method. In forward propagation, the data $\boldsymbol{x}$ are given to the network, which computes the output $\hat{y}$. In backpropagation, the data from the cost function flows backward through the network to compute the gradient.

2.1.1.4 Convolutional Neural Network

Convolutional neural networks (CNNs) are a special form of neural network for processing data with a grid-like topology, such as time series data (1D grid) or image data (2D grid). In a fully connected neural network (FCNN), each neuron is connected to each neuron in the previous layer. In contrast, CNN uses the mathematical operation of convolution instead of general matrix multiplication in at least one layer. For example, the discrete convolution from signals s_1 and s_2, assuming an integer time index t, can be defined by

$$s(t) = (s_1 * s_2)(t) = \sum_{a=-\infty}^{\infty} s_1(a)s_2(t-a). \tag{2.15}$$

Transferred to the ML context, the inputs of the convolution are usually a multidimensional array of data, and a kernel, which is a multidimensional array of parameters. It is usually assumed that the functions are zero everywhere except for the limited set of points defined in the array. This assumption allows infinite summation to be implemented as a summation over a finite number of array elements. The discrete convolution is then performed over multiple dimensions simultaneously. For a 2D input image $\boldsymbol{X}$ and a 2D kernel $\boldsymbol{K}$ this results in:

$$S(i,j) = (\boldsymbol{X} * \boldsymbol{K})(i,j) = \sum_{m}\sum_{n} \boldsymbol{X}(m,n)\boldsymbol{K}(i-m,j-n), \tag{2.16}$$

respectively since it is a commutative operation

$$S(i,j) = (\boldsymbol{X} * \boldsymbol{K})(i,j) = \sum_{m}\sum_{n} \boldsymbol{X}(i-m,j-n)\boldsymbol{K}(m,n). \tag{2.17}$$

The kernel is chosen to be smaller than the input in the convolution operation of a neural network. Thus, iterating over the kernel dimensions is more efficient since there is less variation in the range of valid values for m and n.

In matrix multiplication of fully connected layers, a separate parameter is used to describe the interaction between each input and output unit.

A convolution layer, on the other hand, computes only the input units in the kernel area for each output unit. Moreover, each kernel element is used at each input position (parameter sharing). Thus, the output can be computed in fewer operations, and fewer parameters must be stored. Since not all units are connected, this is called sparse connectivity. All units that affect an output unit are called the receptive field of the output unit. Due to the concatenated operations, the receptive field of the units in the deeper layers is larger than the receptive field of the units in the flat layers. To obtain a larger receptive field in a flatter architecture, architectural features such as dilated convolution, in which the step size of the kernel is increased, can be used.

An activation layer and a pooling layer follow several convolution operations. After non-linear activation, the pooling layer aggregates the results by replacing outputs close to each other with a pooled statistical value. Max pooling [180] is most commonly used, where the largest value of the results is retained. Alternatives are the average value or a weighted mean. Depending on the range of k results to be aggregated, the size of the outputs reduces by a factor k. The reduced input size leads to higher efficiency and lower memory requirements. In addition, pooling helps to ensure that the representation is invariant to more minor shifts in the input.

2.1.2 Generalization and Regularization

Unlike a classical optimization problem where the parameters are fitted precisely to the data, the aim in ML is a parameter optimization for $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\mathbb{X}^{\text{test}}, \mathbb{Y}^{\text{test}})$. This means it pursues the goal of a small test error on new, previously unseen data, also called a generalization error. The model has to find the balance between a small training error, which avoids underfitting the data, and keeping the distance between the training error and the test error small and thus not overfitting to the data.

The No-Free-Lunch Theorem (NFL) states that within certain constraints in the space of all possible problems, each optimization method performs on average as well as any other [171]. However, this result holds only when considering any problem involving all possible data-generated distributions. Thus, the NFL implies that ML algorithms must be trimmed to perform well on a given task. Moreover, by making assumptions about the probability distributions of the actual applications' data, learning algorithms can be developed that perform well for just those distributions. To affect the performance of a model on the test data set, the training and test data must not be arbitrary. Ideally, they are identically distributed, i. e., from an identical probability distribution. Then, for a randomly selected model with fixed weights $\boldsymbol{w}$, the expected training error would equal the expected test error. Since the weights are determined in an ML algorithm to reduce the error in the training data set, the expected test error will always be greater or equal to the expected value for the training error. To achieve a good result, it is essential that the training data set can represent the test data set as well as possible.

The **capacity** of the model determines whether it tends to underfit or overfit. Models with a large capacity can handle more features of the input data. To a certain degree, it makes sense to increase the capacity because the model can better represent the features of the training data. However, overfitting to the training data can also occur if too many training data-specific features are stored. As a result, the generalization performance of the model deteriorates. Depending on the task's complexity, the algorithm's capacity must be adjusted accordingly. The amount of training data also has an impact. If fewer data are available, there is a higher risk that the algorithm stores irrelevant data features of these few training data. If there is a higher variance in the data set, the algorithm must already generalize better on the data, and overfitting is prevented.

Regularization is a fundamental part of ML. It describes any change to the learning algorithm that aims to reduce the generalization error but not the training error. Thus, methods of generalization counteract overfitting. There are many regularization methods, of which the appropriate one must be selected depending on the task to be solved and the available database. There is no general best form of regularization.

As mentioned earlier, the size of the data set plays an essential role in generalization performance. Artificial **data augmentation** may be helpful if only a few training data are available. The easiest way to extend a data set is to use and modify the existing data. It is crucial to preserve the mapping from input $\boldsymbol{x}$ to output y. For example, images can easily be simulated over many variation factors. Shifting the image's content a few pixels in any direction can increase generalizability. Other methods include rotating, scaling, or flipping images. A certain amount of distortion can also be helpful in some use cases. It is essential not to use transformations that change the assigned class value. Also, data expansion is only beneficial if the data is still within the natural distribution.

It can also be helpful not to optimize the training algorithm over too many iterations to prevent an algorithm from tending to overfitting. It is often observed that only the training error steadily decreases after a certain time while the test error increases. At this point, the algorithm loses generalization power and learns too specific features of the training data set. A validation data set with data not used for parameter fitting during training can be used to store the parameters of the models at the time of the lowest validation error. If this shows no improvement over a specified number of iterations, the last saved parameter set is used. This procedure is referred to as **early stopping**. It is a weak form of regularization because the training procedures, objective function, or admissible parameter space do not need to be adjusted.

A stronger type of regularization is achieved by adding a **penalty term** ω to the cost function:

$$\tilde{J}(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) = J(\mathbb{X}, \mathbb{Y}; \boldsymbol{\theta}) + \alpha\omega(\boldsymbol{\theta}), \tag{2.18}$$

where $\alpha \in [0, \infty)$ defines the weighting of the penalty term. This constrains the capacity of the model. Common penalty terms are the L_1 or the L_2 regularization of the network weights. These cause the weights to approximate the origin and tend to have smaller values.

Other possibilities for regularization represent **ensemble methods** or **dropout**. Bootstrap aggregation (bagging) [16] is an ensemble method combining multiple models. The models are trained separately and used together to predict the test data. This procedure works because different models generally do not make the same errors. While bagging allows using the same model type, training algorithm, and objective function multiple times, there are other ensemble methods where the model types are fundamentally different. It uses k different data sets defined from a training data set by drag and drop. Dropout [152] attempts to represent the functionality of bagging in a less computationally intensive method. Dropout uses a base network architecture and removes various non-output units from the architecture. The number of units to be removed is set as a parameter. The units are then selected randomly during the execution of each mini-batch. Unit removal can be seen as masking noise on the hidden units. Unlike bagging, the individual models are not trained to converge, but usually only a few steps. However, since the models share common parameters, the subnetworks converge to an appropriate parameter setting. The advantages of the dropout are the low computational effort and that it only insignificantly restricts the type of model or training procedure. The disadvantages are the significantly larger model size required due to the capacity reduction caused by the dropout and a larger number of iterations of the training algorithm. In addition, dropout is usually only effective if many training examples are available.

2.1.3 Evaluation Metrics

In supervised learning, the result of the model can be compared with the actual label. Consequently, the deviation between the predicted and target values for regression models gives the model quality. For a test data set $\mathbb{X}^{\text{test}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ with n data samples and the associated labels $\mathbb{Y}^{\text{test}} = \{y_1, ..., y_n\}$ where $\hat{y}_i = f(\boldsymbol{x}_i; \boldsymbol{\theta})$ represents the predicted result of the i-th sample, the mean squared error (MSE) is defined by

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2. \tag{2.19}$$

Another metric that provides the deviation of the calculated value from the actual result is the mean absolute error (MAE), defined by

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|. \tag{2.20}$$

Both metrics can be applied to a one-hot encoded multi-class problem or semantic segmentation, where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are vectors or matrices.

Standard classification measures are accuracy, precision, recall, and the F_β score. Accuracy Acc is a commonly used classification metric that gives the percentage of correctly matched samples. It is suitable for both binary and multiclass classification problems. The result options are true positive (TP), false positive (FP), true negative (TN), and false negative (FN):

$$\mathrm{Acc} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{2.21}$$

The precision p describes the percentage of correctly predicted error-free samples concerning the total of all positive labeled results:

$$p = \frac{TP}{TP + FP}. \tag{2.22}$$

Finally, the recall r indicates proportionally how many of the predicted positive samples are correct:

$$r = \frac{TP}{TP + FN}. \tag{2.23}$$

The F_1 score considers both recall and precision. If the weighting of the class under consideration is not equal, the more general F_β score can be used, which includes a class weighting β:

$$\mathrm{F}_\beta = (1 + \beta^2)\left(\frac{p \cdot r}{\beta \cdot p + r}\right). \tag{2.24}$$

When evaluating a semantic segmentation, i. e., the pixel-precise classification, each image pixel's result must be considered. Therefore, the Jaccard coefficient, also called Intersection over Union (IoU), represents a suitable metric and is defined by

$$\mathrm{JC} = \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{\boldsymbol{y}}_i \cap \boldsymbol{y}_i|}{|\hat{\boldsymbol{y}}_i \cup \boldsymbol{y}_i|}. \tag{2.25}$$

The closer the Jaccard coefficient is to 1, the greater the similarity of the sets. Its minimum value is 0. Depending on the task and the input data, it may be helpful to calculate the Jaccard coefficient based on a defined class only.

2.1.4 Model Interpretability

Depending on the model type and the activation and the loss functions used, the result includes an evaluation of the model confidence in addition to the class assignment. For example, in a regression task with one-hot encoding, a probability $p \in [0, 1]$ indicates how confident a sample, or in semantic segmentation, a pixel, is assigned to a class. For better visualization of the results, this work uses a jet colormap representation of the predictions, whose progression is shown in Figure 2.1. Blue indicates that the pixel does not belong to the class ($p = 0$), and red indicates a class assignment with high confidence ($p = 1$).

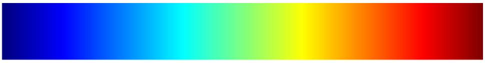


Figure 2.1 Jet colormap.

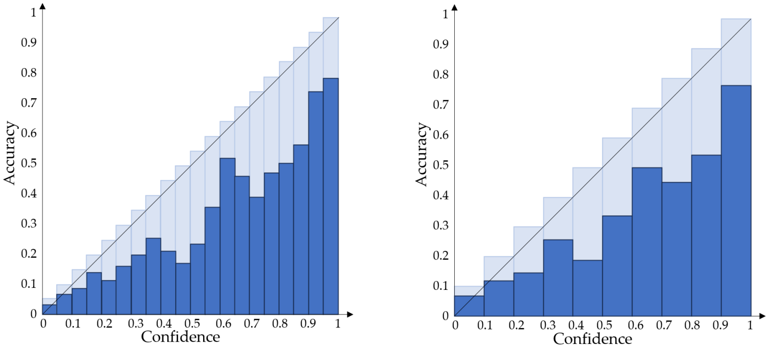
The models must be well-calibrated for the interpretation of the prediction results as model confidence to be valid [107, 164], i. e., the prediction value should correspond to the prediction probability. Therefore, the expected calibration error (ECE) is used as a summary statistic for the calibration [88, 114, 164]. It is defined by

$$\mathrm{ECE} = \mathbb{E}[|P(\hat{y} = y|\hat{p} = p) - p|], \tag{2.26}$$

where $\hat{y}$ is the predicted label, y is the true label, $\hat{p}$ is the model probability for its prediction, and $P(\hat{y} = y|\hat{p} = p)$ is the data distribution's probability for a correct prediction that the model prediction $\hat{p} = p$. To quantify the continuous values, the results can be divided into M bins. Assuming D_m to be indices of samples with the prediction results are in the range $(\frac{m-1}{M}, \frac{m}{M}]$, the ECE is

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|D_m|}{n} |\mathrm{Acc}(D_m) - \mathrm{Conf}(D_m)|, \tag{2.27}$$

where n is the total sample number. $\mathrm{Acc}(D_m) = \frac{1}{|Dm|} \sum_{j \in D_m} \mathbf{1}(\hat{y_j} = y_j)$ and $\mathrm{Conf}(D_m) = \frac{1}{|D_m|} \sum_{j \in D_m} p(\hat{y_j} = y_j|\boldsymbol{x}_j; \boldsymbol{\theta})$ are accuracy and confidence averaged over the samples in the bin. The ECE is the summation of the weighted average of the differences between the average accuracy and the confidence over bins. A reliability diagram represents the metric by plotting the confidence against the accuracy per bin, as shown in the diagrams in Figure 2.2. If the confidence value matches the accuracy for each bin, which would result in a diagonal plot, the calibration is perfect. Using a higher number M of bins better approximates the diagonal line. The examples in Fiugre 2.2 show poorly calibrated results, where the light blue bars would correspond to a good calibration. In semantic segmentation, each pixel is considered a separate sample. The prediction probability can also be used to estimate the model uncertainty for a well-calibrated model. For example, it can detect out-of-distribution data, whose results are usually predicted with high uncertainty.



(a) Reliability diagram with $M = 20$ bins. **(b)** Reliability diagram with $M = 10$ bins.

Figure 2.2 Reliability diagrams. The diagonal black line and the light blue bars show the perfect calibration. The diagonal line is more approximated using a higher number M of bins. Both examples show poorly calibrated results.

For a well-calibrated model, entropy [146] is the most commonly used information measure for the detection of out-of-distribution data [100, 145]. Adapted from the segment-level confidence metric of Mehrtash et al. [107], the following metric is obtained for evaluating the segmentation quality of a foreground class without the presence of ground truth. This metric is based on the pixel-level class prediction $\hat{\boldsymbol{y}}_{ij}$, which gives the pixel-level probability for class j out of a total of c classes. It calculates the average pixel-wise entropy values for that class. This gives for a sample $\hat{\boldsymbol{y}}_i$ and a predefined class j:

$$\mathcal{H}(\hat{\boldsymbol{y}}_j) = -\frac{1}{|\hat{\boldsymbol{y}}_j|}\sum_{k=1}^{m}[p(\hat{y}_{j_k})\log\big(p(\hat{y}_{j_k})\big) + (1-p(\hat{y}_{j_k}))\log\big(1-p(\hat{y}_{j_k})\big)], \tag{2.28}$$

where m is the number of pixels of the sample. For the calculation, a binary classification is assumed. This means that the probability that a sample belongs to class j is $p(\hat{y}_k = j)$ and that it belongs to another class is $1 - p(\hat{y}_k = j)$. One-hot encoding achieves this binarity.

The metric can also be extended over all classes, which is for a given sample $\hat{\boldsymbol{y}}_i$:

$$\mathcal{H}(\hat{\boldsymbol{y}}) = -\frac{1}{|\hat{\boldsymbol{y}}|}\sum_{j=1}^{c}\sum_{k=1}^{m} p(\hat{y}_{j,k})\log\big(p(\hat{y}_{j,k})\big). \tag{2.29}$$

Both formulas result in the same outcome in a two-class problem with one foreground and a background class. A limitation to the validity of the metric is the requirement of good model calibration.

Despite the knowledge about the certainty of model predictions, the problem remains in the context of model interpretability that modern ML methods are usually models that humans cannot fully understand [52]. Because the models are based on data, they lack transparency, which makes it challenging to interpret and explain the procedure. This aspect complicates their use in many fields. However, knowledge-based systems have already been established, especially in the production environment. These models are based on existing expert knowledge and a series of equations and logical rules. Therefore, humans can understand the functionality and explain how the model works. Unfortunately, this traceability of the results is not given to artificial neural networks. The terms **hybrid machine learning** or **hybrid AI** describes the combination of rule-based knowledge systems (symbolic AI) with machine learning models (sub-symbolic AI) [102]. This approach combines the ("unconscious") processing of perceptual data with ("conscious") logical reasoning. There are different methods for combining expert knowledge with data-based models. For example, knowledge-based systems can be supplemented by a data-based model as soon as these systems reach their limits in the application [52]. In this way, the system can still be explained for straightforward examples, and the data-based approach, which is difficult to interpret, is only used when there would be no solution otherwise. The hybrid model achieves higher accuracy through this combination than the single solutions. However, a series connection of knowledge-based and data-based models is also conceivable. Often, the symbolic AI serves as a data provider for the subsymbolic AI, which processes the pre-processed data. But a result validation of the data-based model via a knowledge-based model is also possible [52]. Thus, the result does not have to be blindly trusted since this is verified by expert knowledge and rule-based approaches. Verification is otherwise only possible with cross-validation and sufficient tests to rule out malfunctions. Therefore, using so-called hybrid AI can increase interpretability and strengthen confidence in the algorithm.

2.1.5 Network Architectures

Neural networks can be divided into groups depending on the structure of the network architecture. This includes the overall network structure, with the definition of the connection layers, the type of connection, the depth of the network, and the training procedure. Sometimes the different network architectures use slightly different approaches to accomplish the same task, but often they are designed for various problems. The chapter presents three structures used in this work.

2.1.5.1 Autoencoder

An example of a group of neural network models from unsupervised learning is autoencoders [8, 15]. The goal of this network structure is to reconstruct the input $\boldsymbol{x}$ at the output $\boldsymbol{y} = \hat{\boldsymbol{x}}$ to learn meaningful features between the layers in a lower dimensional space. The network effectively consists of two parts: the encoder with $E(\boldsymbol{x}) = \boldsymbol{l}$ and the decoder $D(\boldsymbol{l}) = \boldsymbol{y}$. Figure 2.3 symbolically shows the layout of an autoencoder. The network architecture is always symmetric and usually reduces the features until the middle layer, the latent space $\boldsymbol{l}$. Autoencoders are often used for dimension reduction or feature learning. Another typical application of the autoencoder is also anomaly detection. If the autoencoder is trained only on good data, the reconstruction will not consider deviations from the input image. This results in a significant error between the original input $\boldsymbol{x}$ and the network output $\boldsymbol{y}$. Based on this difference, the anomaly can be detected. An important aspect is that the latent space $\boldsymbol{l}$ represents all relevant features of the input image. Otherwise, the model cannot give suitable results.

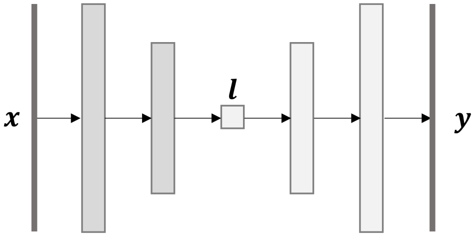


Figure 2.3 The architecture of an autoencoder. The input $\boldsymbol{x}$ is translated into an internal representation $\boldsymbol{l}$ and assigned to an output $\boldsymbol{y} = \hat{\boldsymbol{x}}$. The feature maps can be fully connected or connected via convolution layers.

2.1.5.2 U-Net

A network architecture structurally similar to the autoencoder is the U-Net presented by Ronneberger et al. [134]. The U-Net architecture also has an encoder path to extract features from the input image. The feature vector is then expanded in the decoder path. However, in this model architecture, the feature maps generated during the downsampling of the input image are reused. These so-called skip connections, shown as blue arrows in Figure 2.4, are located between the encoder and decoder paths and help to ensure that no critical information is lost. The feature maps from the encoder path are copied to the decoder path and concatenated with the corresponding feature maps. The architecture does not use fully connected layers but only convolution operations. In the original version of Ronneberger et al. [134], each operation consists of two convolutions, a max pooling in the encoder and an analog transposed convolution in the decoder path. This network architecture is used in most cases to create segmentation maps of images.

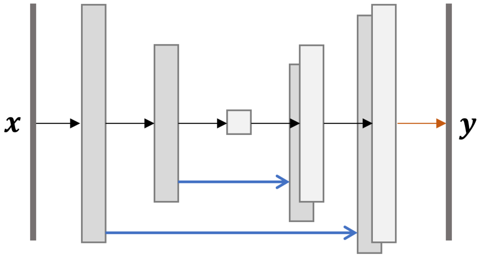


Figure 2.4 The boxes denote the feature maps, while the black arrows between the boxes represent the operations from Figure 2.5, where the first block does not contain max pooling. The orange arrow indicates a convolution with kernel 1×1 and a following sigmoid or softmax activation to map the feature vector to the desired number of classes. The blue arrows represent the skip connections.

Due to the tremendous success of the U-Net architecture, several modifications and optimizations exist. Often the structure and arrangement of the convolution operations are adapted. Others add functions in the skip connections, for example. Wang et al. [166] propose using dilated convolutional layers with an increased step size in the kernel instead of the two standard ones. ones. As described in 2.1.1.4, the receptive field can be increased by dilated convolutions. Each encoding and decoding operation uses a standard convolution followed by multiple dilated convolutions concatenated as input to the next operation. This architecture provides a larger receptive field despite a less partial network architecture, which helps capture the context of the image.

Oktay et al. [122] propose extending the U-Net architecture using attention gates. The soft attention modules for CNN proposed by Jetley et al. [76] enhance the feature maps of relevant image regions while weakening the influence of unimportant image regions. The output of the attention gate represents a weighted feature map $\hat{\boldsymbol{h}}_i^l = \boldsymbol{h}_i^l \cdot \alpha_i^l$, where $\boldsymbol{h}_i^l$ is the feature map of the previous layer $l \in \{1, ..., L\}$ and $\alpha_i^l \in [0, 1]$ is the weighting factor. The attention gates are inserted into the skip connections when integrated into the U-Net architecture. This actively suppresses the activation of irrelevant features, thereby pushing back the number of redundant features transmitted. Multidimensional attention coefficients can also be used for multiple semantic classes. By integrating

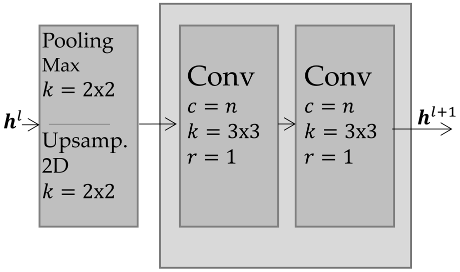


Figure 2.5 Operations in each down- or upsampling step. An activation with ReLU follows each convolution layer (Conv). The parameter n denotes the number of feature maps, while k is the kernel size.

the attention gates into the skip connection, the factor α is calculated considering the feature maps of the skip connection $\boldsymbol{h}^l$ and the feature maps of the next deeper layers of the decoder path $\boldsymbol{g}$. The additive attention is formulated as follows,

$$\begin{aligned} q_{att}^l &= \psi^\top(\phi_R(W_h^\top \boldsymbol{h}_i^l + W_g^\top \boldsymbol{g}_i)), \\ \alpha_i^l &= \phi_S(q_{att}^l(\boldsymbol{h}_i^l, \boldsymbol{g}_i; \Theta_{att})), \end{aligned} \tag{2.30}$$

where ϕ_S represents a sigmoid activation and ϕ_R corresponds to a ReLU activation function. The set of parameters Θ_{att} contains transformations computed using channel-wise $1 \times 1 \times 1$ convolutions for the input tensors. The linear transformations are $W_h \in \mathbb{R}^{F_h \times F_{\text{int}}}$, $W_g \in \mathbb{R}^{F_g \times F_{\text{int}}}$ and $\psi \in \mathbb{R}^{F_{\text{int}} \times 1}$, where F_x corresponds to the number of feature maps in layer h, g, and the intermediate space int. So the feature maps are mapped linearly and then summed element by element. Afterward, a ReLU activation occurs, and the vector transformed using a linear transfomation to the correct dimension. Due to the sigmoid activation at the end, a multiplication factor in the range $[0, 1]$ is achieved.

2.1.5.3 Generative Adversarial Network

Another common used network architecture is the generative adversarial network (GAN) [57]. The particular property of this network architecture is that it consists of two adversarial parts, which optimize each other.

The generator G is trained to produce images that could come from the set of training images $\mathbb{X}^{\text{train}} = \{\mathbf{x}_1, ...\boldsymbol{x}_n\}$. This requires the network to learn the distribution of the generator p_g over the data from $\mathbb{X}$. Since the generator is trained with noise, a prior probability distribution on input noise variables must be learned in advance $p_{\boldsymbol{z}}(\boldsymbol{z})$, which is mapped to the data space with $G(\boldsymbol{z};\boldsymbol{\theta}_g)$. The discriminator $D(\boldsymbol{x};\boldsymbol{\theta}_d)$ attempts to distinguish real images from generated images and returns the probability of being a real image as output. Figure 2.6 illustrates the structure of a GAN with its two parts. For example, an autoencoder or a U-Net can be used as a generator network. The discriminator D is trained to maximize the probability of correctly recognizing both the training sample and the samples from G. At the same time, D is trained to keep the part of the correctly recognized samples low. Therefore the value function

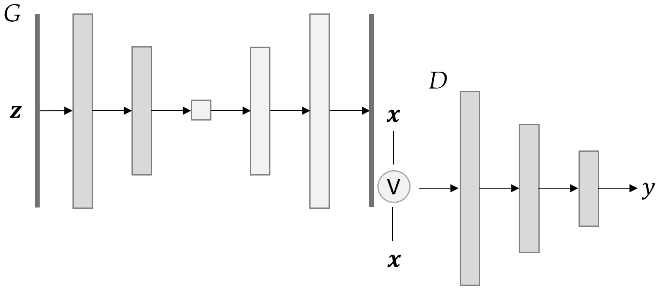


Figure 2.6 The architecture of a GAN. The generator G produces the sample $\boldsymbol{x} = G(\boldsymbol{z};\boldsymbol{\theta}_g)$. The discriminator network D tries to distinguish between the samples $\boldsymbol{x}$ drawn from the training data and the samples drawn from the generator network. It outputs a probability value $\boldsymbol{y} = D(\boldsymbol{x};\boldsymbol{\theta}_d)$ indicating the probability that the sample is an element from the available training data.

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.31}$$

minimizes the part $\log(1 - D(G(\boldsymbol{z}))$. By this min-max play of the two networks, realistic-looking images are produced by the generator after optimization. Mirza and Osindero [110] extended the approach of the GAN by adding additional information in the generator and the discriminator. The so-called conditional generative adversarial networks

(CGAN) receive the additional information $\boldsymbol{v}$, which can be a class label or other information besides the noise term $\boldsymbol{z}$. The value function is adapted as follows

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{v})] \\ &+ \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{v})))].\end{aligned} \tag{2.32}$$

To map the images produced by the generator closer to the ground truth, Isola et al. [72] suggest the regularization by adding L_1 or L_2 distance measurement to the function. By the weighting factor α, the ratio of the functions can be determined, and the following formula results:

$$\mathcal{L} = \min_G \max_D V(D,G) + \alpha \mathcal{L}_{\mathrm{L}_1|\mathrm{L}_2}(G). \tag{2.33}$$

2.2 Laser Welding

The joining process using laser technology is becoming increasingly prevalent in industry due to the possibility of automation, process time reduction, and suitability for individualized products.

2.2.1 Laser Technoloy

The word **laser** used in everyday language is just an acronym and stands for "**L**ight **A**mplification by **S**timulated **E**mission of **R**adiation".

The process of light generation can be explained with the help of Bohr's atomic model. According to Bohr, the electrons orbit the atomic nucleus, which consists of protons and neutrons, on circular paths with a fixed radius and defined energy levels [14]. Thereby, the inner orbits, which are closer to the nucleus, have a lower energy level than the outer orbits. It is necessary to add energy to an electron to move it to a more distant orbit (absorption, figure 2.7 (left)). The amount of energy E is proportional to the frequency f of the proton. This dependence is described by $E = h \cdot f$, where $h = 6.626 \cdot 10^{-34}$ J s is Planck's quantum. In other words, when the light of frequency $f_{E_1 E_2}$ is supplied to an atom, the electron can transition to a higher energy state E_2 if Bohr's condition $E_2 - E_1 = h \cdot f\text{ť}_{E_1E_2}$ is satisfied. In this case, a light quantum of energy from the proton $hf_{E_1E_2}$ is taken from the supplied light. Due to the additional energy, the electron is in an unstable, so-called excited state. After a short time, the electron falls back to the lower energy level. During this transition, the absorbed energy is released in the form of a photon (spontaneous emission, Figure 2.7 (center)), and light is emitted. The released energy has the same frequency $f_{E_1E_2}$ as the previously supplied light and is emitted in a spatial direction. In addition to the process of spontaneous emission, Einstein [38] also postulated induced or stimulated emission in 1916 (Figure 2.7 (right)). In this case, the return of an atom from an excited state does not occur spontaneously but by the external action of a light wave, which also satisfies the Bohr frequency condition. Due to the induced emission, the released proton is emitted in the same propagation direction as the incident proton. Both protons have the same frequency and phase (i. e., are coherent with each other), which amplifies the incident light wave. This amplification effect is the basis of laser technology.

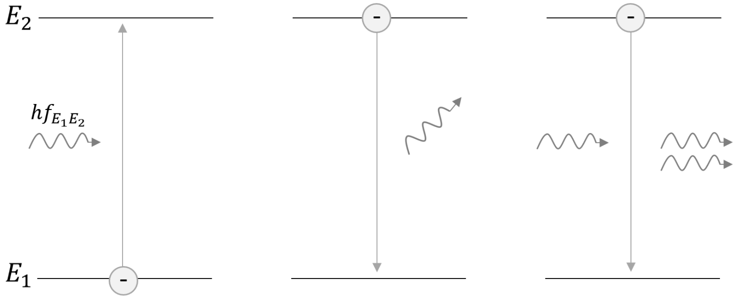


Figure 2.7 The energy level change of an electron. Absorption (left), spontaneous emission (center), stimulated emission (right). Based on Eichler and Eichler [37].

A laser device consists of three components: the laser medium, the pump, and the resonator. The laser medium can consist of different aggregation states, such as gases like carbon dioxide, solids like crystals and glasses, or liquid substances. The most crucial step is the excitation of the active material, called pumping, which leads to light amplification. Depending on the type and excitation of the laser material, laser devices are divided into the following types: optically pumped lasers (excitation by light), electron beam pumped lasers (excitation by electron or other particle beams), gas discharge lasers (excitation of gases by electrical energy input), chemical lasers (excitation by a chemical reaction), and injection or diode lasers (excitation by the passage of current in a semiconductor). In the most simple case, the resonator consists of two mirrors arranged parallel to each other and enclosing the laser medium. These mirrors cause the released protons to be reflected and move through the laser medium, triggering stimulated emissions and releasing new protons. With the proper spacing between the mirrors, the waves of released light overlap, and the light waves are optimally amplified. One of the mirrors is partially transparent, called a decoupling mirror. The light emitted from there is the so-called laser beam. Maiman [101] developed the first prototype of a laser device in 1960.

Compared to radiation from conventional light sources, amplified laser light is characterized by narrow spectral linewidth, high beam power, and strong focusing. It also exhibits a high degree of local and temporal coherence. The radiation can be generated in the wavelength range from below 0.01 µm to above 1000 µm, covering the spectral ranges of soft x-rays, ultraviolet, visible and infrared light, and millimeter waves.

2.2.2 Laser Welding Process

Laser welding uses a laser device as the energy source. Before the laser beam can be used, the distance between the beam source and the workpiece must be bridged. The laser beam is guided through a fiber optic cable. When the beam hits the inside of the fiber optic cables, it is deflected by total internal reflection. The angle of divergence during decoupling corresponds to the angle during coupling. The outcoupled laser beam is then aligned in parallel by a collimator and focused by an optical system. The schematic setup is shown in Figure 2.8.

At the focal point of the focused laser beam, the energy is so high that the material starts melting. The subsequent solidification of the melt in the joining zone joins the components together. The use of filler materials is usually not necessary.

A portion of the power P of the laser beam is reflected by the material surface (P_R). The difference $P - P_R$ penetrates the piece of work. The

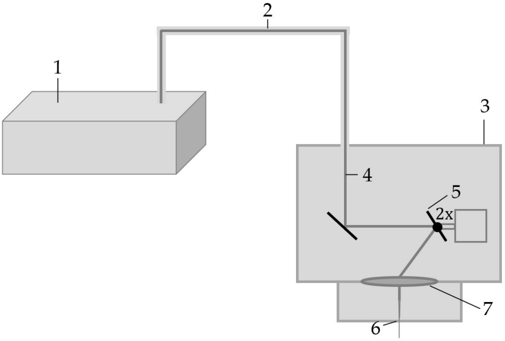
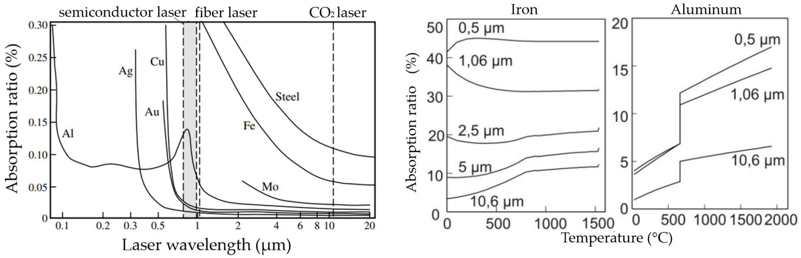


Figure 2.8 The connection from the laser device to the processing field with programmable focusing optics. (1) laser device, (2) fiber optic cable, (3) optics, (4) laser beam, (5) scanner mirrors moved by motors (2×), (6) focused laser beam, (7) plan field lens.

material absorbs this energy mostly completely (P_A). Depending on the material properties, a part of the laser beam is transmitted (P_T). The principle of conservation of energy follows the detailed energy balance with $P = P_R + P_A + P_T$. The reflectance $R = P_R/P$ and the absorption rate $A = P_A/P$ are essential for applying the laser welding process [69]. The values depend on the wavelength of the laser beam. For example, the absorption coefficient of different materials and different wavelengths at room temperature is plotted in Figure 2.9(a). Copper and precious metals such as gold and silver show a substantial decrease in absorption in the visible range. At the same time, aluminum has a sparse absorption rate in the entire wavelength range considered. Moreover, depending on the material, the absorption rate also depends on temperature, as shown in Figure 2.9(b). In aluminum, the absorption rate always increases with temperature, regardless of the wavelength, which changes the material's behavior during the melting process. This behavior does not occur with iron. There, the course depends mainly on the wavelength. Consequently, when choosing a laser for material processing, the beam wavelength and the corresponding absorption of the material must be considered.

Furthermore, not the total power released P_A can be used as processing power. This value is reduced by the heat dissipation P_V, which gives the amount of heat flowing into the workpiece per unit of time at the



(a) Absorption ratio of the laser beam in metals depending on laser wavelength [117].

(b) Absorption ratio depending on laser wavelength and temperature for iron and aluminium [69].

Figure 2.9 Absorption ration of laser in metals at vertical beam incidence depending on wavelength and material (a) and on temperature (b).

boundary surface of the processed volume. P_V thus depends on the difference between the temperature prevailing at the welding spot and at the rest of the workpiece, its thermophysical material values, and its geometry [69]. Laser welding of components made of metallic materials is distinct into two types of welding: **heat conduction welding** and **deep penetration welding**. Deep penetration welding is more relevant in manufacturing technology than heat conduction welding because it enables a higher process efficiency and significantly higher welding speeds. Energy can be applied to the joining zone in a very targeted manner, keeping heat conduction losses to the surrounding material low. Deep penetration welding produces vapor capillaries, shown in Figure 2.10. These capillaries are tubular cavities filled with metal vapor, at the exit of which a metal flare is formed. The molten metal flows around the capillary and re-solidifies behind it to form the weld. The laser beam is reflected several times on the inner walls of the vapor capillary, which increases the absorption of the laser beam's introduced energy. As a result, the melting volume increase. The depth of the melting zone is usually larger than the width, hence the term deep welding. In heat conduction welding, there are no vapor capillary and no multiple reflections. It is mainly used for valuable objects with low material thickness. The transition from heat conduction welding to deep penetration welding occurs abruptly when a threshold value of the beam parameter quotient is reached. This quotient is calculated as the incident laser power P related to the focal spot diameter d. The threshold value assumes higher values for higher thermal conductivities and decreasing absorption rates of the material. A higher feed rate also increases the threshold value.

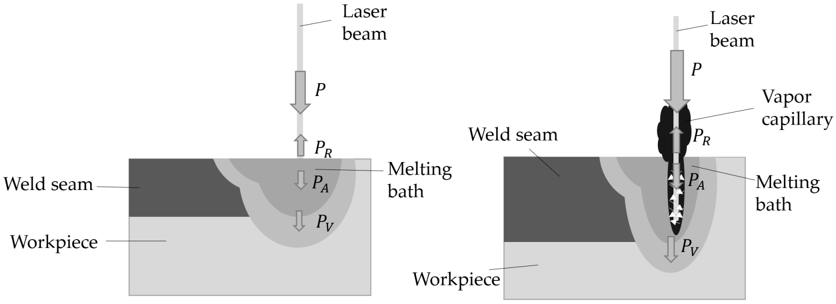


Figure 2.10 Schematic drawings of heat conduction welding (left) and deep penetration welding (right).

Laser welding offers several advantages compared to other welding processes. High welding speeds of over 1000 mm/s and small beam diameters of less than 50 µm are possible. Another advantage over other processes like arc or oxyacetylene welding is a lower thermal load on the component and the possibility of producing thin seams. In addition, the use of movable mirrors that deflect the laser beam, as shown in Figure 2.8, allows complex geometries to be processed fast and automatically in high quality [68].

2.2.3 Process Instabilities

The process window represents the parameter range in which a stable welding process with desired results can be achieved. Outside the process window, there are losses in process efficiency and quality. Furthermore, instabilities cannot always be robustly remedied due to the multiple interactions of the phenomena involved. In addition to the mechanisms primarily inherent in the process, influences such as surface condition or aspects of seam preparation like the size of the joint gap and joint offset play an essential role [22]. Stability in the capillary is a prerequisite for a calm melt pool and thus good results [84].

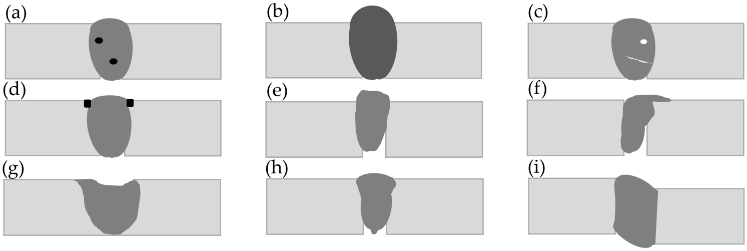


Figure 2.11 Weld seam defects (following DIN EN ISO 6520-1). (a) solid inclusions, (b) mixed joint due to foreign material, (c) cracks and/or cavity, (d) undercut, (e) excess weld metal, (f) root overlap, (g) incompletely filled groove, (h) lack of fusion and penetration, (i) missalignment.

Figure 2.11, based on DIN EN ISO 6520-1, schematically shows a few possible welding errors. If foreign substances are deposited in the weld metal, this is called a solid inclusion (Fig. 2.11(a)). These foreign substances can be slag, flux, or oxide residues, but also foreign metals. As a result, they lead to a reduction in strength behavior by reducing the weld cross-section. Furthermore, if the foreign material is also melted during the melting process, it can mix with the material of the workpiece. This results in weld seams with undesirable material properties (Fig. 2.11(b)). Cracks rarely occur during laser welding because the heat input into the material is low, and thus less stress is created. Nevertheless, they can occur. Far more common are pores, spatter, and holes (Fig. 2.11(c)). They often occur in combination due to the same or at least similar causes. For example, a very liquid melt can cause material ejection due to the pressure in the vapor capillary. Contaminated surfaces, residues of coatings, or different melting temperatures of the materials to be welded can also lead to an unstable welding process. On the one hand, fires can occur on the material (Fig. 2.11(d)) or, due to different material properties and absorption behavior, an unstable capillary can result, causing spatter and pores. Consequential defects of an unstable capillary can be seam overheight (Fig. 2.11(e)) or seam leakage (Fig. 2.11(f)). If the critical gap size of the workpiece is exceeded, the material melting is no longer sufficient to produce a stable joint. The consequences are, for example, seam collapses, critical errors, or no joint at all (Fig. 2.11(g),(h)). Furthermore, no stable joint is created in the case of an offset, where the welded parts are not in the required identical parallel plane. In this case, there is also the problem that the material is no longer in the correct focus position of the laser (Fig. 2.11(i)).

2.2.4 Process Monitoring

A calm and stable weld pool must be ensured to avoid errors. This stability can be controlled by process parameters such as wavelength, laser power, speed, focus position, and pre-processing of the workpiece. Furthermore, there is the possibility of using different welding techniques. For example, moving the laser spot quickly and simultaneously during forward motion (wobbling) can create stable dynamics in the weld pool. This welding technique can improve the process and weld quality. Another technique is welding with inner and outer fiber cores of different intensities. The inner fiber core produces the desired weld depth with high intensity, while the outer fiber ring stabilizes the weld pool with lower intensity. Especially for materials such as copper or aluminum, which have a low absorption level at room temperature that increases massively in the liquid keyhole, it makes sense to use such techniques [44, 144]. Nevertheless, errors can only be partially avoided. Process monitoring is essential, especially in manufacturing operations that are highly automated [36, 126]. Stavridis et al. [155] and Sun and Kannatey-Asibu Jr. [156] give an overview of different quality assessment methods in laser welding. Quality monitoring can be divided into pre-process, in-process, and post-process phases. Table 2.1, partly adapted from the paper of Stavridis et al. [155], gives an overview of the different phases and the monitorable quality criteria. In addition, standard technologies used for the respective process monitoring are listed.

The sensors used for quality monitoring can be mounted on- or off-axis to the system. With the on-axis setup, the welding process is observed via the light path of the optics. This setup has the advantage that no additional external installation is required at the welding station. In addition, the sensor’s field of view is always in the welding position, even if the mirrors deflect the laser beam. Sensor outputs are attached

Table 2.1 Quality citeria used for inspection.

Quality assessment stage	Principal quality criteria	Technology
Pre-process	Seam tracking, clamping, gap, part geometry	Camera, ToF, OCT
In-process	Weld defects, melt pool dimensions, weld position, spatter	Camera, photodiode (VIS, UV, IR), OCT, acustic, x-ray radiography
Post-process	Weld geometry, visible defects	Camera, ultrasound, ToF, OCT

to the optics for this purpose. The signals are routed to the correct output using the wavelength via semi-transparent mirrors within the optics. A schematic diagram for two sensors is shown in Figure 2.12. The monitoring methods considered in this work cover the different stages of process monitoring. A more detailed overview of each method's state-of-the-art is given in the corresponding Chapters 3, 4, and 5.

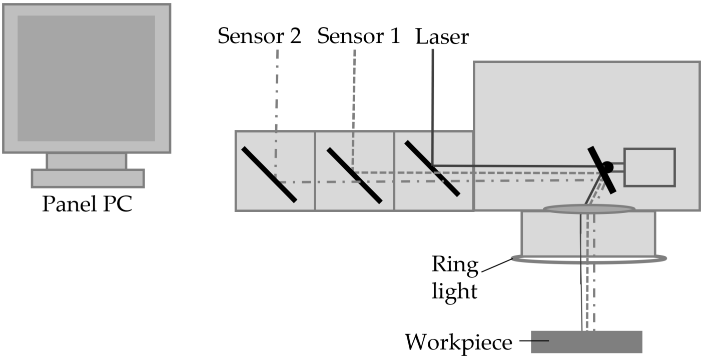


Figure 2.12 On-axis sensor attachment for laser welding.

Used setup When setting up the monitoring sensors, the welding process should be restricted as little as possible. This applies to the modification of the welding station and the influence on the process times. Other aspects that must be considered are the additional calibration effort and costs incurred by the monitoring sensors. As indicated in Table 2.1, different sensor systems can often be used to achieve the same purpose. In this work, the focus is on using a camera as a surveillance sensor. A non-modified standard setup of a welding station is assumed.

Most experiments use a 1.5 Mpixel intensity monochrome **camera** with a CMOS sensor. The full image resolution is 1440×1080 pixels [9]. The camera is attached to the optics at the sensor output. For this reason, the imaging ratio varies depending on the focus distance to the component. In addition, the magnification can be changed by installing lenses between the camera and the optics. For pre- and post-observation with the camera, additional illumination is required. For this purpose, an LED ring light is attached to the optics so that it does not interfere with the design of the welding station or the welding process itself. The light thus shines from above and is reflected by the component. A red light of wavelength 625 nm is used.

In the post-process inspection in Chapter 5, additional height data are used. These are acquired using the principle of **optical coherence tomography** (OCT). OCT was first introduced in 1991 [67] and uses a technique known as low-coherence interferometry. The system design is similar to the measurement system of a Michelson interferometer [108]. The difference is that OCT uses a light source with a well-defined and relatively short coherence length [34]. A beam splitter divides the light wave into two parts. One part is directed onto the workpiece and reflected there. The other part of the light, transmitted by the beam splitter, falls on a mirror in a reference arm and is reflected there (reference beam). The sample beam and the reference beam meet again and interfere exactly when the difference in the paths traveled by the two beams is less than the coherence length. The interference signal is recorded with a detector and then evaluated. By moving the mirror in the reference arm, interference signals are recorded from different depths of the sample, as far as reflecting structures are present. Moving the mirror in the reference arm while simultaneously measuring the interference signal thus enables axial scanning of the sample. The path length differences over the speed of light can also be expressed as time-of-flight differences. Therefore this OCT method is called time-domain OCT (TD-OCT). In the spectral-domain OCT (SD-OCT), the simple detector of the TD-OCT system is replaced by a spectrometer. This modification eliminates the need to move the reference arm mechanically, increasing axial resolution. The Fourier transform of the spectrum provides a back reflection profile as a function of depth. Analysis of the depth information is derived from the different interference profiles resulting from the different path lengths of the reference and sample arms. This concept is referred to as Fourier-domain OCT (FD-OCT). This work uses an FD-OCT with a sampling rate of 70 000 scans per second, which is connected to the optics coaxially with the laser beam. Using this system, the relative height values of the component are acquired. The height information is sampled in increments of 11.7 µm with a measurement range of approximately 12 mm. The lateral resolution varies depending on the distance of the optics to the component and is thus dependent on the focal length.

The **computing hardware** is an industrial panel PC with the specification shown in Table 2.2. The touchscreen panel PC has a robust case and is thus ideal for use in industrial environments. They fulfill the IP65 protection class according to DIN EN 60529 (VDE 0470-1) at the front, which means the case is dust-tight and protected against jets of water from any angle. The rear side meets IP20, meaning the case is protected against foreign objects with diameter $\geq$ 12.5 mm. Cooling is passive, i. e. without a fan.

Table 2.2 Configuration computing hardware.

Item	Specification
Operation system	Linux
Memory/ Storange	DDR4L 8 GB RAM and 32 GB SSD
Processor	Intel Core i5-7300U dual-core 2.6 GHz
Safety class	IP65 (rear IP20)

2.3 Application Hairpin Welding

To illustrate the proposed methods, the process of joining copper wires to produce formed coil windings, often described as hairpin welding, is used. This application is well suited because it presents challenges in the various steps of the welding process.

Hairpin Welding The increasingly important application with high-quality requirements for laser welding comes from electromobility. Electromobility will become more and more prevalent in individual transportation in the future. This is why vehicles' designs and various components are constantly refined and optimized. Manufacturing the winding is a key technology in this context, so innovations are essential for this area [142]. Furthermore, distributed windings have become common in the automotive industry for producing high power density drives for battery electric vehicles [55].

The conventional copper windings in the stator of an electric motor are replaced by preformed plug-in coils inserted into the stator slots of the laminated core and connected to each other. The use of open plug-in coils (Figure 2.14(a)) has become prevalent and is still being further developed in research [53, 79]. The plug-in coils consist of U-shaped, enameled copper flat wires. The geometry of a bent wire, shown in Figure 2.14(a), resembles a typical hairpin, which is why they are often called hairpins. The process is divided into four steps in a highly simplified representation, shown in Figure 2.13 following the illustration in Glässel et al. [55]. The first step is to cut the insulated copper wire to the desired length. Then the ends of the copper pieces are stripped, and the wire is bent into a U-shape (1). Next, the preformed hairpins are inserted into the stator slots of the sheet metal core (2). Here the ends are bent apart and twisted together in pairs (3). This is followed by the final step, welding of the copper wires (4), whose quality monitoring is the focus of the work.

For welding, the ends must be stripped in a previous step. In Figure 2.14(a), the stripped end is shown in light gray. Figure 2.14(b) shows the structure of the welded hairpin elements symbolically.

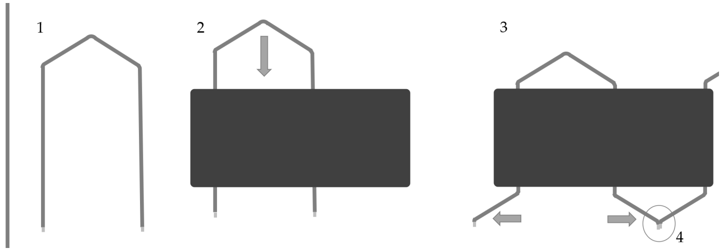
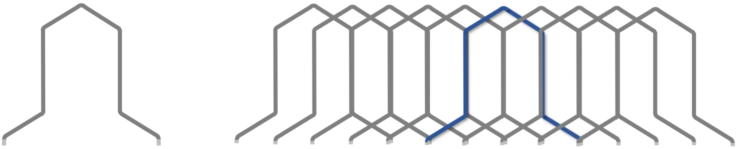


Figure 2.13 Process of hairpin stator production.



(a) Front view of a open formcoil.

(b) Structure of welded hairpins. One pin is highlighted in blue color for better visualization.

Figure 2.14 Geometry of a bent hairpin and the welding arrangement of the wires.

As with conventional stators, the stack of sheets continues to consist of many electrical sheet layers insulated from each other. The overall structure of the stator also remains essentially unchanged. Depending on the motor design, between 160 and 220 pairs of copper bars are connected in the laminations of a stator [54, 71, 128, 184]. Figure 2.15 shows the construction of a stator in a schematic front view.

The plug-in coil design saves space and increases the efficiency of an electric motor. In addition, the technology enables a high level of automation in production [53, 80]. However, since even one defective contact point leads to machine failure, it is also highly relevant to achieve the required contact point properties in a reproducible manner and to check each contact point for a defect [54, 79, 104, 162]. The high number of contact points is a significant challenge. Due to different evaluation criteria such as process times, long-term stability, reproducibility, electrical or mechanical connection properties, and automation capability, laser welding is well suited for the joining process.

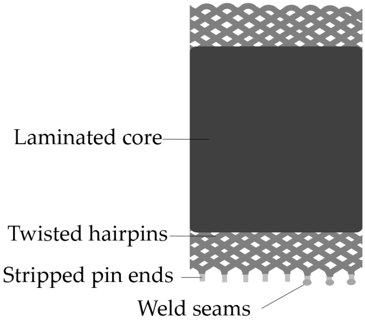


Figure 2.15 Schematic front view of a stator. The illustration shows some weld seams on the stripped pin ends. In a finished stator, all pin ends are welded.

According to DIN EN 13602, the copper grades Cu-ETP, CU-FRHC, and Cu-OF with a conductivity of at least 58 m mm^2/Ω are used as conductor materials [53]. As mentioned in Chapter 2.2.3, various influencing factors can lead to errors in the welding process. As also mentioned in Chapter 2.2.3 and Chapter 2.2.4, copper has challenging properties for laser welding. Among other things, the rapidly increasing absorption ratio in the liquid well is a major challenge. Figure 2.16 shows exemplary welding results of varying quality produced by different sources of defects. Several factors can lead to errors, such as an offset of the wires to be welded, incorrect laser power, or welding without first stripping the wire ends.

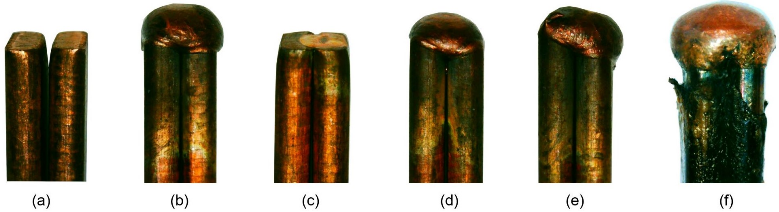


Figure 2.16 Various results of welding copper wires. (a) no weld, (b) good weld, (c) wires are not in the focus of the laser, (d) weld with too low power, (e) misaligned wires, (f) insulated copper rods.

Different properties and measured variables can be used to evaluate the quality of the weld seam [104, 162]. Influencing factors before, during, and after welding play an essential role, and there are various approaches to improve or monitor quality. Some approaches will be discussed in more detail in the following chapters.

3 Pre-Process Monitoring

3.1 Introduction

Laser welding is established in industry to produce permanent joints between metal structures. The process is used in a wide range of different applications and is often automated. By deflecting the laser beam with adjustable mirrors in the optics, complex geometries can be welded without requiring manual intervention during the process. In addition, laser optics are often attached to robots that guide the laser beam along a defined trajectory. Based on three-dimensional models from a computer-aided design (CAD) system, the trajectory is programmed using a specialized computer-aided manufacturing (CAM) system and a computer-aided process planning (CAPP) system [31]. Finally, mathematical coordinate transformation methods transform the welding positions from the model space to the working space.

However, the absolute definition of the welding coordinates requires that the part is always in exactly the same position in the defined working space. This condition can only be achieved with precisely aligned and accurate clamping devices. But especially in small batch production, which often processes different components, it is expensive and time-consuming to create exact clamping fixtures that maintain positions precisely. In addition, the precise clamping of the joining partners leads to increased setup time. Therefore, to increase the efficiency of this process, the degree of automation in determining the welding coordinates must be further increased [172]. Also, in the hairpin welding application presented in Chapter 2.3, the position of the copper wire pair is not always exact due to the pre-processing steps, the fixture that clamps the copper wires, and the position of the entire stator. Therefore, in a purely coordinate-based weld, defective parts may result due to misaligned weld positions.

To realize the process with a high degree of automation and process reliability, the component, or more precisely, the welding position, must be detected automatically. Using a camera sensor and image processing algorithms, the component's position can be detected, and the information forwarded to the laser control system. The challenges for computer vision are low contrast, reflections and surface defects such as scratches or other disturbing elements [29]. Although the images within a production line are similar, there are deviations caused by pre-processing steps or surface texture. In addition, different positions on the part result in different orientations and various areas around the weld. Moreover, the light from the illumination is reflected differently, resulting in shaded areas.

Another important aspect is the pre-processing monitoring of the components to avoid errors and dangerous situations during welding. For example, faulty pre-processing steps can lead to deviating component geometries, so no proper welded joint can be produced. Using hairpin welding as an example, errors such as a gap or offset of the copper wires or a missing wire are possible. If such deviations are detected before welding, they can be rectified directly. This saves expensive follow-up costs. Furthermore, steering the laser beam into a pre-defined position can cause serious damage if the component is missing or misaligned.

This chapter proposes an approach that uses semantic segmentation to reinforce the features of the part's geometry. This enables the calculation of the welding position by shifting and rotating the model coordinates to the position in the workspace. In addition, the presence of the components, as well as their geometry and size, are monitored. Thereby no definition of a region of interest (ROI) is necessary. An ML algorithm processes the entire camera image acquired with the setup presented in Chapter 2.2.4. The result is a semantic segmentation of the image that returns the defined classes as one-hot encoded matrices. Further pre-process monitoring and weld position detection algorithms can be performed downstream on a false color image representing the individual detected classes.

3.2 State-of-the-Art

In order to detect the welding position in a camera image, individual algorithms must be developed considering the relevant features. Depending on the component geometry and material properties, there are different challenges in detecting the welding position.

Dmitry et al. [31] develop an algorithm to reliably detect the gap between two sheets with a camera for butt welding. This algorithm will adjust the weld position to the detected gap. With LED illumination on two sides, the component is illuminated to create a shadow in the gap. Segmentation is then performed based on the contrasts of the object according to brightness. The extreme values of the pixel distribution of the whole image concerning the brightness define the threshold values for the analysis. Finally, they perform morphological closure, a combination of erosion and dilation, for the segmentation. However, the segmentation still does not allow precise detection because the image may contain different elements in the threshold selection range. Therefore, the detected segments are reselected using a set of rules that include, for example, the parallelism of the segments or a constraint on the width.

Kong et al. [85] and Dinham and Fang [29] focus on recognizing a weld seam on sheet metal to calculate the seam shape for further welds. This application faces similar challenges regarding shape recognition on images with reflections and weak contrasts. Kong et al. [85] develop an algorithm to detect the initial position of the seam using corner detection. First, they pre-process the images, including smoothing, sharpening, and region segmentation. Then, they use the Harris operator to detect the corners of the geometry by sudden changes in image brightness to separate the weld from the background. Dinham and Fang [29] use the Hough transformation to detect the outer boundary of the weld so that they can remove the background. Then, other algorithms are applied to the filtered images to detect the weld reliably. These include Sobel edge detection, matching neighboring pixels to remove small areas, and smoothing algorithms.

Using a pre-defined ROI makes the recognition task more trivial since irrelevant objects in the background can be ignored from the beginning. Dinham et al. [30] and Ryberg et al. [138] use a ROI in the center of the image. As a result, many interfering signals are already ignored, and the relevant features can be found more easily based on the pixel intensities and their value changes.

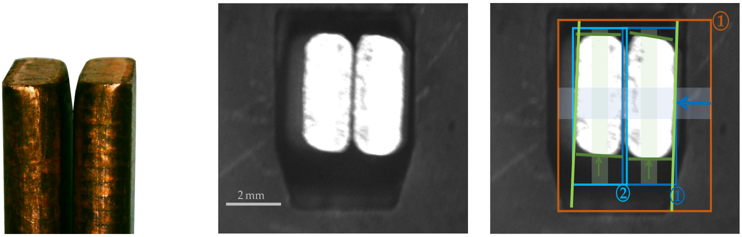
Depending on the application, the signals from other sensors can be used to identify the components better. An example is using a height scanner with the help of OCT. Baader et al. [6] show how the position of a component can be determined using the height signal. In addition, geometric deviations between the two joining components, such as weld gaps, different heights, or a lateral offset, are detected. This way, the components' geometry deviations can be detected in pre-process monitoring in addition to the welding positions.

3.3 Experimental Setup and Data Basis

Images are recorded with a monochrome camera for the pre-processing steps of welding position detection and monitoring of the previously performed process steps. The setup from Chapter 2.2.4 with a camera mounted on the optics is used for data acquisition. The images are captured with different cameras. On the one hand, the camera's sensor technology varies with CCD or CMOS sensors. On the other hand, cameras with different resolutions and varying imaging scales and magnifications are used. The proposed algorithm works independently of the exact camera model, and different approaches are proposed for widely varying image resolutions. To illustrate the proposed algorithm, process data from hairpin welding with a resolution of 656×494 pixels and 720×540 pixels are used. The data used for the evaluation and comparison of network architectures are not allowed to be shown in this work. Therefore, similar images are shown as examples. The inference of the algorithm is computed on the hardware also shown in Chapter 2.2.4.

3.4 Component Detection

The material properties and the geometry determine how well the component can be detected within the camera image. The following section thoroughly examines images of copper wires from a hairpin welding process. In the first use case shown in Figure 3.1, the stripped wire ends to be welded are cut straight in the pre-processing step. The smooth surface shown in Figure 3.1(a) is clamped in a fixture parallel to the optics and the illumination for the welding process. Since the plane copper surface reflects the illumination light directly into the camera, the image clearly shows the pin areas contrasting with the background (Figure 3.1(b)). The component surface is identified using a thresholding method based on pixel intensities. Figure 3.1(c) shows the component boundaries detected by the detection algorithm with green lines.



(a) Straight cropped copper wires.

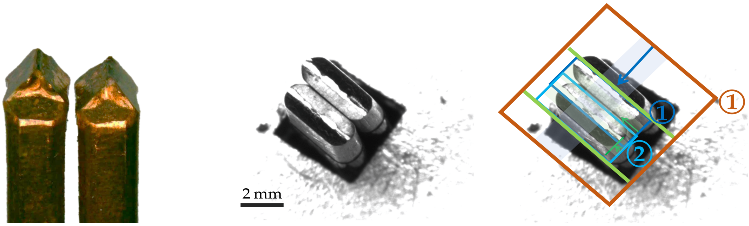
(b) Image of the wires recorded through the optics.

(c) Detected pin surface (within the green lines).

Figure 3.1 Straight cropped pair of hairpins (a) on an image recorded through the optics (b) and the result of the image processing algorithm (c).

The algorithm uses a user-defined ROI (orange rectangle (1)) and a pre-defined search direction (horizontal blue arrow) together with a threshold value to find the border of the pins. Therefore it evaluates the pixel intensity in the search direction and recognizes the threshold crossings of the pixel values (light green lines). For more robust detection, the intensity is not determined by a single pixel value but by an averaged value over an area (blue-shaded area). The width of the area is defined by the user. Then, considering further parameters like the number of pin pairs and the width range of the pins, the algorithm establishes small ROIs for the individual components (blue rectangles (1) and (2)). Within these ROIs, the pixel intensity values are evaluated in an orthogonally oriented direction to the first search direction (vertical green arrows). According to this, the upper and lower edge of the pins is defined by a threshold crossing of pixel value changes in the green-shaded areas (green lines). Finally, the weld position is calculated based on the defined component boundaries.

Monitoring of the component's presence and its geometry is performed considering the detected pin edges. For this purpose, limit values can be defined for the pin sizes, as well as the lateral and radial offset. Measured in terms of electrical resistance, lateral and radial misalignment of the pins have a more significant effect on the welded joint than axial misalignment [170]. While axial misalignment cannot be recognized in the camera image, lateral and radial misalignment certainly can.



(a) Squeezed copper wires.

(b) Image of the hairpins recorded through the optics.

(c) Image processing with pre-defined ROI and threshold method.

Figure 3.2 Squeezed pair of hairpins (a) on an image recorded through the optics (b) and the result of the image processing algorithm (c).

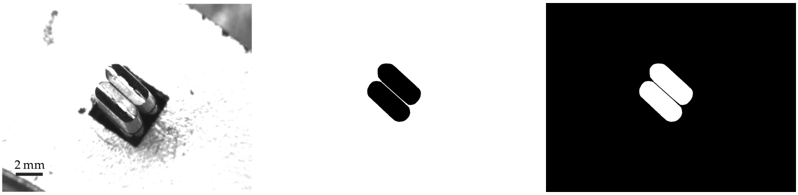
Changing the pre-processing step from straight cutting to crimping the wire results in a different surface structure, which is no longer aligned parallel to the optics. Triangular structures are created as shown in Figure 3.2(a). The method saves time and expensive tools compared to the straight cut-off. The resulting surface structure does not have a negative effect on the welding result, but it affects the previous image processing. Also, wear of the cutting tool can have similar consequences. Blunter cut edges lead to irregularities in the wire surface, which become visible in the camera image. Figure 3.2(b) shows that light from some slanted surfaces is reflected into the camera, while other pin surfaces are shaded. False edges are now detected by the algorithm just shown, which approximates thresholds within a defined ROI. At the first approximation in the orange ROI, a false pin beginning is recognized. In the middle of the pin, there is a significant pixel intensity change because the light is reflected in different directions. This change is detected and defined as the beginning of the pin due to the largest threshold crossing (light green line). Also, the second pin edge is wrongly defined. The transition from the lower pin end to the stator opening is recognized as the edge of the pin contact surface. The outer pin edges are difficult to detect using thresholding methods because the images show many shades and structures. In addition, the structures are different from image to image, which makes a fixed rule-based approach challenging. Specifying a smaller ROI or narrower ranges for the pin width only helps to make the algorithm more stable to a limited extent.

The hairpin surface structure can vary significantly due to the pre-processing steps. Even sophisticated pre-processing steps would have to be repeatedly adapted to the degree of tool wear or other influencing factors. However, image-processing ML algorithms have proven their generalization power and are robust against more minor variances. With the help of a CNN, a wide range of image features can be processed and evaluated without having to define fixed rules.

3.4.1 Model Architecture

In order to find the best solution for the problem, the design of the solution approach must first be determined. Both the suitability and the effort for data generation have to be considered.

By comparing the type of algorithms, their advantages and disadvantages become apparent. **Classification** offers the least labeling effort, as only one label needs to be assigned per image. Nevertheless, it is unsuitable because it is not possible to identify the component position. **Object detection** provides the coordinates of the objects. However, the exact position is relevant for calculating the weld seam coordinates and the corresponding translation and rotation. To achieve the required accuracy, **keypoint detection** can be used. The algorithm detects the component's position, size, and orientation by placing three or four points at the edges. Labeling is relatively fast because only points with a predefined size need to be placed on the image. However, the plausibility prediction reaches its limits, and a pre-verification of the component's geometry is not possible. **Semantic segmentation** using pixel-precise labeling is associated with a high effort but is best suited for the application. Pixel-precise labeling can be used to determine the exact position of the components. Based on this exact definition, the weld seam coordinates and a plausibility check are calculated. The fact that less training data is needed is another advantage of semantic segmentation. Due to the pixel-based loss function, within whose calculation the value of each pixel is included, each pixel can be considered an individual training instance [159].



(a) Image of hairpins recorded through the optics.

(b) Binary mask of the background class.

(c) Binary mask of the foreground class.

Figure 3.3 Image with its one-hot encoded mask. The pixels relevant for the respective class are shown in white, while the other pixels are shown in black.

The semantic segmentation needs pixel-accurate labels for the training process. These so-called masks have the same resolution as the camera images and contain a class assignment for each pixel. The masks are defined using one-hot encoding for each class. One-hot encoding converts a categorical variable into a binary representation, ensuring that each category is considered equally. Therefore a separate mask is defined for each class, containing the values "0" for no class assignment and "1" for a class assignment. The labels must be precise and accurate, as the model is only as good as the quality of the training data. At a minimum, the model is trained with one background and one foreground class showing the component. Figure 3.3 shows an example of a camera image and the corresponding one-hot encoded mask. In a two-class problem, the background and foreground classes are complimentary.

Architecture Definition The most popular model architecture for semantic segmentation is the U-Net architecture, according to Ronneberger et al. [134]. This model architecture requires little training data and is well-suited for data augmentation. Ronneberger et al. [134] have conducted experiments using only 30 images for training. This aspect is essential for an industrial manufacturing application where data availability is often a problem, as explained in Chapter 1.1. Therefore, the definition of the architecture is based on the original definition of the U-Net of Ronneberger et al. [134] (vanilla U-Net), and the depth and number of filters are adopted.

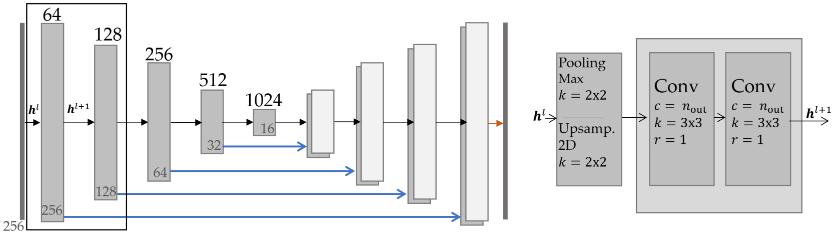
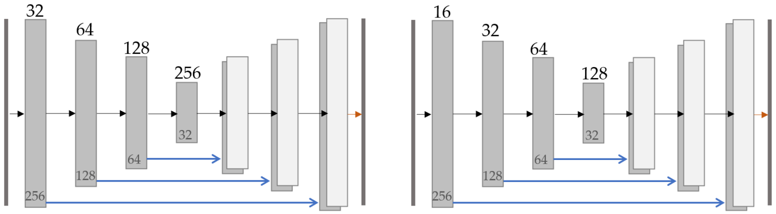


Figure 3.4 Vanilla U-Net. The boxes represent the feature map, with the x- and y-resolution at the bottom of the box and the number of channels at the top of the box. The black arrows between the boxes represent the encoder-/ decoder operations shown next to the architecture, where the first block does not contain max pooling. The parameter n_{out} indicates the number of feature maps resulting from the operation, k the kernel size, and r the dilation rate. The orange arrow represents a convolution with kernel 1×1 to map the features vector to the desired number of classes. The blue arrows represent the skip connections.

Figure 3.4 shows the structure of the architecture. The number inside the box represents the x- and y-resolution of the feature map, where 256 represents a resolution of 256×256 pixels. Five encoder operations with $n_{\text{out}} = \{64, 128, 256, 512, 1024\}$ and the corresponding decoder operations are used. Each operation contains two standard convolutional layers with kernel size $k = 3 \times 3$ and zero padding. Between the operations, a max pooling algorithm with kernel size $k = 2 \times 2$ is performed to reduce the dimensionality of the feature maps. A corresponding upsampling takes place in the decoder path. The steps of each block are shown in Figure 3.4 on the right. An activation with ReLU follows each convolution layer (Conv). After the last decoder operation in the expansive path, a 1×1 convolution and a softmax activation are performed, which maps the feature vectors to the number of classes to be learned. The number is $c = 2$ in the presented example. One class is the background, and the other is the foreground class containing the hairpin structure.

Since model capacity is critical in overfitting and regularization, the evaluation considers smaller variants of the U-Net architecture. Memory size and inference time are also strongly dependent on model size. The input size of 256×256 pixels is used for the evaluation. This dimension is smaller than in the original work, which uses a size of 572×572. Also, the images and the recognition tasks are less complex than in the original work. In most cases, only one foreground class of a component that is in the focus of the image is to be detected. Therefore, the architecture uses fewer filters and a smaller model depth to reduce capacity. Figure 3.5 shows the number of filtering and pooling operations of the used variants.



(a) Small vanilla U-Net.

(b) U-Net.

Figure 3.5 U-Net modifications with lower model capacity. The structure of the encoder and decoder operations is analogous to Figure 3.4. Only the depth and the number of channels $c = n_{\text{out}}$ varies.

Dilated convolutions can be used to increase the receptive field [174]. For example, Devalla et al. [26] use extended convolutional layers within the U-Net architecture to obtain a larger receptive field and capture more contextual information. They increase the dilation rate in the convolutional layers of the deeper encoder-/ decoder operations. The risk is that small objects may not be detected due to the increased dilation rate [61]. Wang et al. [166] present an architecture that modifies the U-Net architecture by merging convolution layers with different dilation rates. They modify the encoder and decoder operation by replacing the two standard convolutions with one standard convolution followed by four dilated convolutions. The output of each convolution is then concatenated, preserving the information of each convolution. The number of filters c per convolutional layer is reduced by the factors $\{2, 4, 8, 16, 16\}$ while the dilation rate r increases with $r = \{1, 3, 6, 9, 12\}$. As a result, the network has fewer parameters but can capture more receptive image information. Furthermore, the model can capture objects of different sizes. Figure 3.6 shows the architecture of the encoder and decoder operation. Each convolution is followed by an activation with ELU, while the last decoder operation in the expansive path is followed by a 1×1 convolution and a softmax activation.

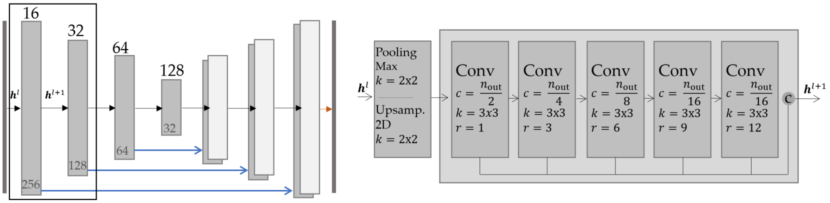


Figure 3.6 SDU-Net architecture with the adapted encoder and decoder operation. n_{out} represents the number of channels after concatenating the convolutions' outputs. The dilation rate r increases while the kernel size k remains unchanged.

Oktay et al. [122] propose extending the U-Net architecture by attention gates (AGs). Models trained with AGs implicitly learn to suppress irrelevant regions in an input image while highlighting salient features useful for a given task. Thus, by integrating AGs within the skip connection, only relevant features are transferred to the expansive path. Since increasing the receptive field is also very promising, the SDU-Net architecture is used and extended by AGs within the skip connection. The AG calculates a weighting for the copied feature maps of the encoder path based on the output feature maps of the previous operation in the upsampling path. First, the resolution and the number of channels of the feature maps are adjusted. Then both feature maps g^l and h^l are summed by element. This process causes aligned weights to become larger while unaligned weights become relatively smaller. Next, an activation with ReLU of the resulting vector and a 1×1 convolution is performed, reducing the dimensions. Finally, the vector passes through a sigmoid layer that scales the vector in the range $[0, 1]$ and generates the attention coefficients α, with coefficients closer to one indicating more relevant features. Figure 3.7 shows the procedure on the right side. The rest of the network definition is adopted from the SDU-Net.

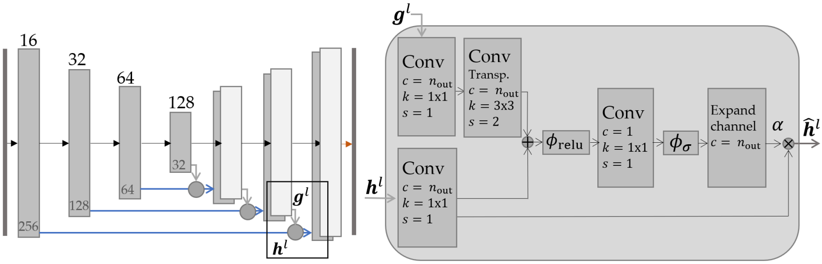


Figure 3.7 AttSDU-Net. Extension of the SDU-Net with AGs in the skip connections. The block next to the model architecture shows the operations within an AG. The other structure and the definition of the encoder and decoder operations are analogous to the SDU-Net (Figure 3.6).

The number of parameters for each architecture is specified in Table 3.1. The ability of a network to learn specific features increases with the number of parameters. Usually, more parameters require more training images and more training iterations. In addition, the model size influences its memory size and inference time.

Table 3.1 Number of parameters of the network architectures.

Model Architecture	Number of Parameters
Vanilla U-Net	31 030 658
Small vanilla U-Net	1 925 058
U-Net	183 922
SDU-Net	162 457
AttSDU-Net	238 844

For training, the model uses the categorical focal loss with $\alpha = 0.25$ and $\gamma = 2$. In addition, it uses an ADAM optimizer with the parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, which control the length of the moving averages. The learning rate is reduced during the training after three epochs without any improvement and starts at $\epsilon = 0.001$. All models are trained with data augmentation and early stopping based on the loss value of the validation data set for better regularization. The image data and the labeled one-hot encoded masks are modified by rotation, horizontal and vertical flip, shift, zoom, and shear. The missing pixels at the edge resulting from the data transformation are supplemented with a nearest-neighbor algorithm. However, the hyperparameters are low enough that the generated data corresponds to the realistic population. The data is scaled to a value range of $[0, 1]$.

Architecture Evaluation The following shows a comparison of the presented network architectures. The performance is compared by running training procedures with various training-validation-test splits and comparing the results. The basis is a data set $\mathbb{X}$ with $n = 900$ samples. Since the number of acquired training images is relevant for the model architecture selection, the set of $\mathbb{X}^{\text{train}}$ is varied from $n = \{5, 10, 25, 50, 100, 200\}$. In contrast, the set of validation images $\mathbb{X}^{\text{val}}$ remains stable at $n = 500$ and the set of test images $\mathbb{X}^{\text{test}}$ at $n = 200$ for better comparability. Ten random training, validation, and test data splits were performed for each size n of the training data set. On each split, five independent networks were trained from scratch. Detailed curves of the training sessions per network model, broken down by accuracy, can be found in Appendix A. The training sessions were performed with batch size $BS = 2$ and 100 steps per epoch.

Figure 3.8 shows the accuracy history for different numbers of training samples. The different training-validation-test splits are each summarized in a graph, showing the mean of the training results for 50 training procedures as a line and the variance as a shaded area. The diagrams show that the variance of the training procedures is widely spread. This variance arises because, for single training sessions, the model achieves an accuracy of only about 80% on both $\mathbb{X}^{\text{train}}$ and $\mathbb{X}^{\text{val}}$. Considering the data basis, this result suggests that the entire image is predicted as background, and the model does not learn relevant class features for the foreground class. The evaluations of the IoU on the foreground class of the training data, shown in Figure 3.9, confirm this assumption. Each diagram shows the results for 50 individual training processes. For some processes, the IoU does not improve and remains at 0, meaning no

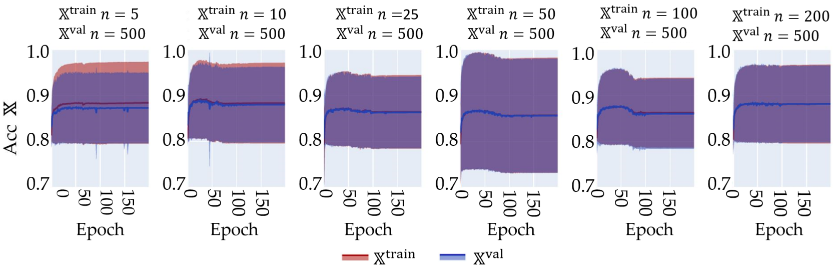


Figure 3.8 Accuracy of the vanilla U-Net for a different number n of $\mathbb{X}^{\text{train}}$. The red lines show the averaged results of the 50 training sessions on the data set $\mathbb{X}^{\text{train}}$ and the blue lines for $\mathbb{X}^{\text{val}}$. The variance is presented within the shaded area. The x-axis represents the progression of epochs during the training process. Each epoch includes 100 steps with $BS = 2$. The evaluation was performed after each trained epoch with 100 steps.

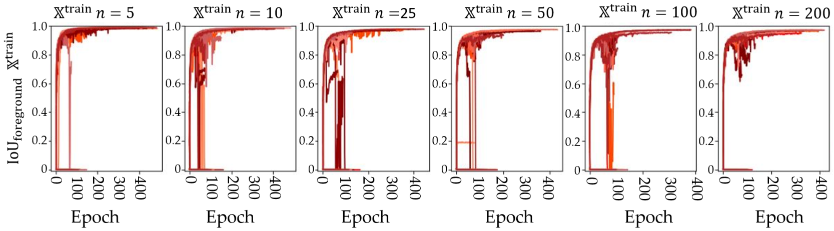


Figure 3.9 IoU of the foreground class of the vanilla U-Net for a different number n of $\mathbb{X}^{\text{train}}$. The histories of a total of 50 different training sessions on ten various training-validation-test splits are shown in each case.

pixel is correctly assigned to the foreground class. Due to early stopping, these training progressions are stopped after a few epochs. The diagrams of the individual training sessions for different training-validation-test splits are shown in the Appendix A in Figure A.2.

In comparison, Figure 3.10 shows the diagrams of the training history of the minor U-Net variant (Figure 3.5(b)). The plots show that all 50 models achieve reasonable results in each case, regardless of the split between training, validation, and testing data or the training session. As a result, the variance of the outcomes is significantly lower. This finding suggests that the vanilla U-Net has a too large capacity for the amount and complexity of the data sets. Figure 3.10 also demonstrates that a

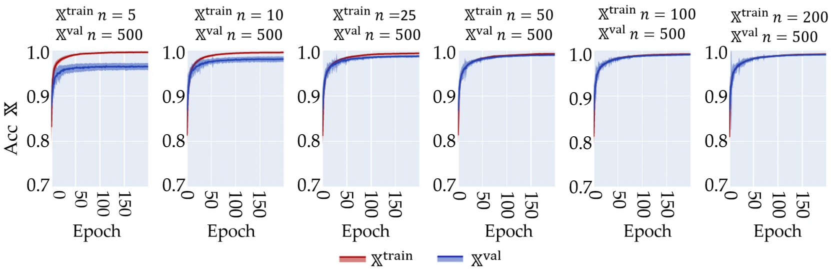


Figure 3.10 Accuracy of the minor U-Net for a different number n of $\mathbb{X}^{\text{train}}$. The red lines show the averaged results of the 50 training sessions on the data set $\mathbb{X}^{\text{train}}$ and the blue lines for $\mathbb{X}^{\text{val}}$. The variance is presented within the shaded area. The x-axis represents the progression of epochs during the training process. Each epoch includes 100 steps with $BS = 2$. The evaluation was performed after each trained epoch with 100 steps.

larger amount n of $\mathbb{X}^{\text{train}}$ gives a better result for $\mathbb{X}^{\text{val}}$. These models show better generalization because the larger training data set covers a more considerable variance of data features.

In semantic segmentation, class imbalance often occurs between the background and foreground classes. Consequently, the accuracy of the sample prediction is usually high. This is also visible in comparing the results in Figure 3.8 and Figure 3.9. While the foreground class IoU stagnates at 0 for some models, the accuracy value for these models is still approximately 80%. This results from the large proportion of the background class. The class contains many pixels, so minor deviations do not affect the results significantly. However, most of the class is outside the region of interest, i. e., outside the image area showing the component. For this reason, Figure 3.11 considers the IoU of the foreground class, i. e., the detected component. This class has fewer pixels associated with it, so a deviation of one pixel is more noticeable. Figure 3.11 shows the model results on $\mathbb{X}^{\text{test}}$ for different numbers n of $\mathbb{X}^{\text{train}}$ in the form of boxplots. Each boxplot represents the results from models of 50 individual training sessions. The diagram confirms that increasing $\mathbb{X}^{\text{train}}$ leads to a significant improvement, especially between $n = 5$ and $n = 10$. After that, the outliers due to the random train-test splits are also relatively small, which shows that the models generalize well. The difference between $n = 50$ and $n = 200$ is tiny for the far-increased labeling effort.

This slight difference shows that $n = 50$ randomly chosen images cover the relevant features of the existing data set. By comparing the results considering the different architectures, it becomes clear that the enlarged receptive field of the SDU-Net architecture provides an added value. Also, the extension by the AGs still creates a slight improvement, which is marginal concerning the enlargement of the network architecture by factor 1.5. The vanilla U-Net variants show that the version with more feature maps has a slightly better average value but contains more outliers. The diagrams do not include the largest architecture of the vanilla U-Net due to its low performance.

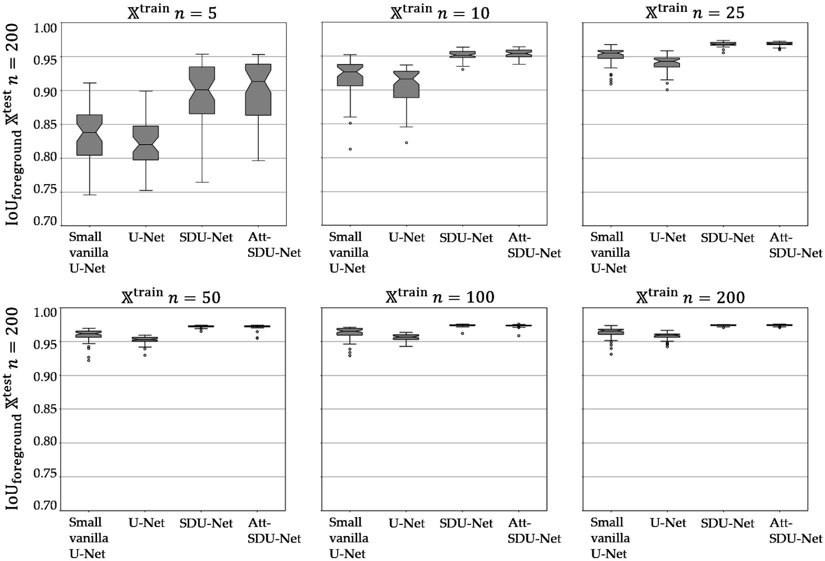


Figure 3.11 Comparison of the IoU of the foreground class for the different model architectures. Results on $\mathbb{X}^{\text{test}}$ with $n = 200$ are shown. Training was performed on data sets $\mathbb{X}^{\text{train}}$ of different sizes $n = \{5, 10, 25, 50, 100, 200\}$. For each number of training data, there are ten different data splits, on each of which five models are trained from scratch. The accuracy curve of the models is shown in Appendix A.

The overall evaluation suggests that the SDU-Net is the most suitable architecture for the application regarding the number of model parameters and model results.

Further evaluations of the SDU-Net architecture with respect to the loss function were performed on different data sets. A different number of classes and the proportion of assigned pixels per class were considered. The categorical focal loss was the most suitable compared to the dice loss, the cross entropy, and the weighted cross entropy. One advantage of the focal loss function is that it can handle multiple and unequally distributed classes without modification. Unlike the weighted cross entropy, the function is not parameterized separately for each data set and each recognition task. Moreover, the weighting factor γ gives more weight to individual divergent samples in the training data, which usually leads to a better model generalization. Appendix B shows the analyses and training histories. Also, in terms of model calibration, focal loss results in a better-calibrated model than dice loss. A good calibration allows conclusions about the model uncertainty. This aspect is elaborated further in Chapter 3.4.3. Furthermore, a comparison of the loss functions concerning model calibration is presented in Appendix C.

Based on these results, the SDU-Net architecture and optimization with ADAM and focal loss are used for feature extraction to identify the components in the camera image. The following sections explain the detailed application and the evaluation of the model. Due to the small model size, both training and inference time are reasonable. An execution using the ONNX runtime and the integrated GPU from the Intel i5-7300U CPU from the hardware presented in Chapter 2.2.4 achieves an inference time of 16 ms. This duration is comparable to other pre-processing image operations and is within the required time of the production cycle.

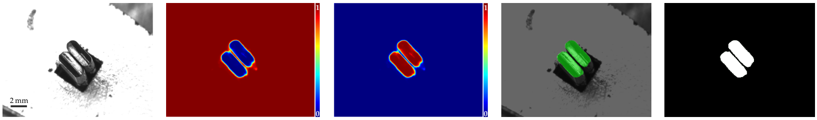
3.4.2 Algorithm

This chapter shows the integration of semantic segmentation with the SDU-Net model architecture to the component position detection procedure. The one-hot encoded model predictions are visualized per layer using the jet colormap, with high confidence class alignment ($f(\boldsymbol{x};\boldsymbol{\theta}) = 1$) in red and no class alignment ($f(\boldsymbol{x};\boldsymbol{\theta}) = 0$) in blue. Depending on the camera resolution and the size of the part to be detected, two strategies are suggested to realize the component position detection. With low image resolutions of the camera, usually only one weld position, e. g. one pair of hairpins, is captured per image to achieve sufficient accuracy.

The entire image is processed to highlight the relevant pixels in this case. Due to technical progress with improving camera and data transmission technologies, the trend is increasingly moving toward using higher resolution cameras. These cameras enable the captured image area to be observed with higher accuracy and a smaller pixel pitch. Alternatively, a minor zoom can be used to view a larger section of the part with the same resolution as before. This changeover allows capturing multiple weld positions, i. e., multiple pairs of hairpins in one image. The accuracy for detecting the individual pins is high enough due to the higher overall resolution. However, since there is often an uninteresting sub-region between the welds, an approach is proposed that crops out the relevant regions of the camera image and processes these regions separately to highlight the relevant pixels.

The following section first presents the method to detect the parts on the entire image before the two-stage training follows. Finally, the chapter shows how the semantic segmentation result is used to detect the weld position independently of the detecting procedure.

Single-Stage-Training A model based on the SDU-Net architecture is used to highlight the pixels of the pin surface. Figure 3.12(b) shows the one-hot prediction for the background class, while Figure 3.12(c) shows the prediction for the hairpin class.

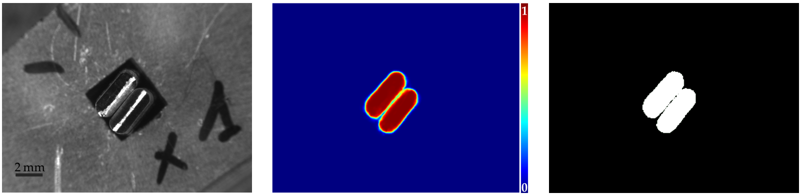


(a) Camera image. (b) Background class. (c) Foreground class. (d) Overlaid image with binary mask. (e) Binary foreground mask.

Figure 3.12 Model prediction results of a trained SDU-Net architecture on a test image. (b) shows the jet-colormap representation of the background class and (c) of the foreground class. (d) and (e) visualize the binary prediction of the foreground class with a threshold value of 0.5 overlaid in green on the camera image and as a binary image.

The class results are complementary for a two-class problem due to softmax activation. For the component detection use case, only the one-hot result of the foreground class is relevant, since it highlights the pixels with the relevant features. Therefore, in the following, the representation of the background class is often omitted, and the result of the foreground class is called "mask", "foreground mask" or "foreground". For creating the binary mask, the prediction result of the foreground class is binarized with a threshold of 0.5. The image shows the pixels associated with the class in white, while the other pixels are black. The $\mathrm{IoU}_{\mathrm{GT}}$ to the hand-labeled ground truth is calculated by the binarized prediction in each case.

In the example in Figure 3.12 the similarity of the background class is $\mathrm{IoU}_{\mathrm{GT}} \approx 0.995$, while the foreground class has an $\mathrm{IoU}_{\mathrm{GT}} \approx 0.939$. The $\mathrm{IoU}_{\mathrm{GT}}$ of the foreground class is more meaningful for the detection quality because each pixel deviation is more considered due to the smaller pixel ratio.

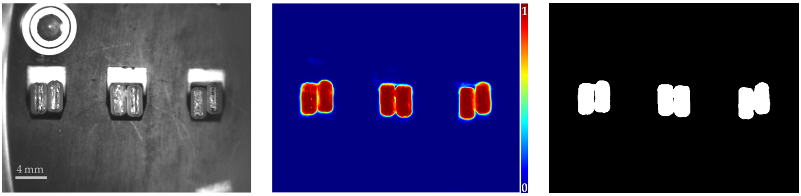


(a) Camera image. (b) Foreground mask. (c) Binary foreground mask.

Figure 3.13 The result of the same model from Figure 3.12 on another test image. $\mathrm{IoU}_{\mathrm{GT}} \approx 0.918$ for the foreground class.

Figure 3.13 shows the result of the same model on another test image from $\mathbb{X}^{\mathrm{test}}$. The camera images (Figures 3.12(a) and 3.13(a)) show the variation of the data set at different positions of the stator. Besides the brightness and the reflections, the orientation of the copper wires varies. The procedure works for different components and a different number of components. It is only essential that relevant features are recognizable in the image. For example, Figure 3.14 shows a result of a model trained to detect three pairs of copper wires in the image. This example illustrates that multiple components of a class can be recognized and highlighted.

When using a high-resolution camera, much information is lost by reducing the image resolution to the input dimension of 256×256 in the used model architecture. Therefore, the input dimension of the CNN



(a) Image with three pairs of hairpins.

(b) Foreground mask.

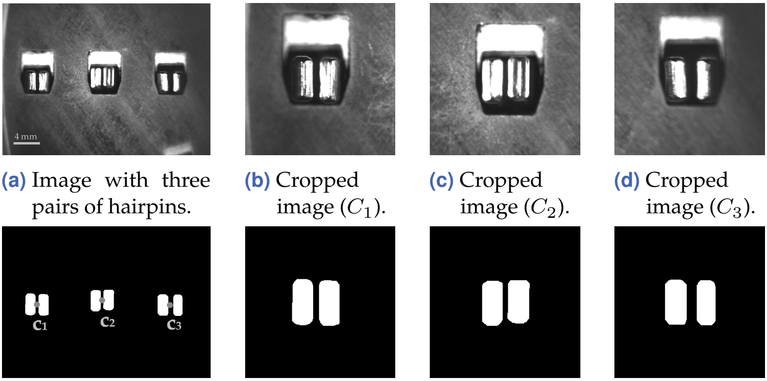
(c) Binary foreground mask. $\mathrm{IoU}_{\mathrm{GT}} \approx 0.964$.

Figure 3.14 Model prediction result of a trained SDU-Net architecture on a test pattern with three pairs of hairpins. Since all components have similar features, they can be captured within the same foreground class.

must be increased to counteract the loss of accuracy. As a result, more features are processed in the network, meaning the architecture must also become deeper in order to be able to process all features. This would result in a significant increase in the number of model parameters. Furthermore, the area between the welds is not relevant for evaluation. Depending on the distance between the recorded welds on the component, this area can take up a large part of the image. The relevant regions showing the weld area have a high similarity. Therefore, their features can be learned from a model regardless of their positions in the image. Consequently, it is more efficient to crop the relevant regions from the image than to downsample the entire camera image to the input dimension. In a two-step process, ROIs of 256×256 pixels around the relevant weld areas are cropped in the first step. The second step is training a model to detect the hairpin regions in the cropped areas.

Multi-Stage-Training A definition of ROIs at fixed coordinates would restrict the flexibility of the ML approach for component position detection. Furthermore, an additional effort for the end user by the two-step approach should be avoided. Therefore, the information about the relevant image regions is extracted from the pixel-precise class assignment, which is needed for the model training. This manually created mask already contains information about the relevant foreground class regions. Thus, in the first step, the user labels all the component pixels in the entire camera image to obtain a mask, as shown in Figure 3.15(e).

Then, based on the pixel-by-pixel class mapping, the algorithm identifies the relevant image regions using a contour search in the binary mask according to Suzuki and Abe [157]. Finally, it crops the areas in both the camera image and the masks depending on the centroids C of the detected contours. The cutouts, which are shown in Figure 3.15 as an example, are then used for the training process to detect the component pixelwise.



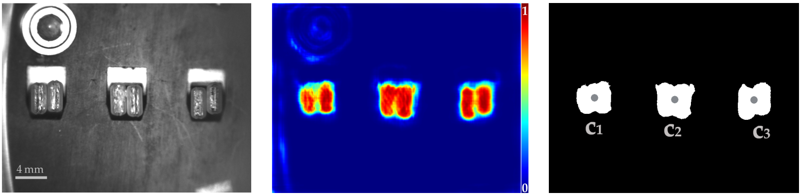
(a) Image with three pairs of hairpins. (b) Cropped image (C_1). (c) Cropped image (C_2). (d) Cropped image (C_3).

(e) Binary Mask with contour centers. (f) Cropped mask (C_1). (g) Cropped mask (C_2). (h) Cropped mask (C_3).

Figure 3.15 Two-stage training. (a) and (e) show the camera image with 720 × 540 pixels and the corresponding mask of the foreground class with the contour centers C. (b-d) show the cropped sections of the camera image around the contour centers and (f-h) the areas of the corresponding foreground masks with 256 × 256 pixels each. The mask of the background is cropped analogously.

This approach has several advantages. First, the model is further regularized by cutting out the relevant image regions. Thus, using a ROI eliminates irrelevant background features that need not be considered in further algorithms and cannot affect model performance. Second, this step increases the size of the data set used for training. In the example with three pairs of hairpins, three input images are available for training instead of one.

When predicting unknown images, the algorithm cannot define the ROI based on the contours of a human-labeled mask. Nevertheless, the regions must be defined at the correct coordinates to crop the image for the model prediction. For this purpose, a separate model is trained on the entire image and mask with an input dimension of 256×256 pixels. Since this model is only used to extract the relevant image regions, abbreviated training can be performed, which is not optimized to obtain the best result with a minimal loss value. The loss of information due to the low input resolution is also not disturbing for this use case. For example, Figure 3.16(c) shows the result of a model trained for 1000 steps. The figure shows that there is still uncertainty in the boundary areas of the pin pairs, and the contours are not well defined. However, these results can be used to find the relevant image areas for the second model since there are evident detections in the foreground class.



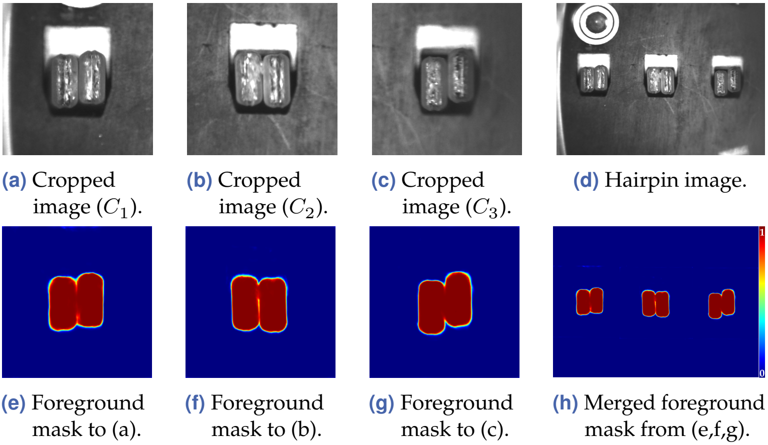
(a) Image with three pairs of hairpins.

(b) Foreground mask.

(c) Binary foreground mask with contour centers C.

Figure 3.16 Result of a model trained for ten epochs with 100 steps each and $BS = 2$. The relevant image areas are defined based on the centroids of the found contours in the binary mask (c).

Figure 3.17 shows the cutout areas of the test image at centroids C_1, C_2, and C_3. In addition, it shows the subsequent prediction of the areas with a second model trained on the cutouts. Since this model is used to determine the welding position, it is trained to the optimal result that minimizes the loss function. Depending on the improvement of the loss value, the training reduces the learning rate, and uses early stopping to regularize the process. The result shows an accurate prediction of the foreground class and well-defined edges. For the subsequent step, in which the welding positions are determined, the individual areas are combined using the coordinates that were previously used to cut them out. The pixels in the image regions that were not predicted by a model are set to zero. The result highlights the relevant hairpin pixels for the entire image. Figure 3.17(h) shows the merged image for the example.



(a) Cropped image (C_1).

(b) Cropped image (C_2).

(c) Cropped image (C_3).

(d) Hairpin image.

(e) Foreground mask to (a).

(f) Foreground mask to (b).

(g) Foreground mask to (c).

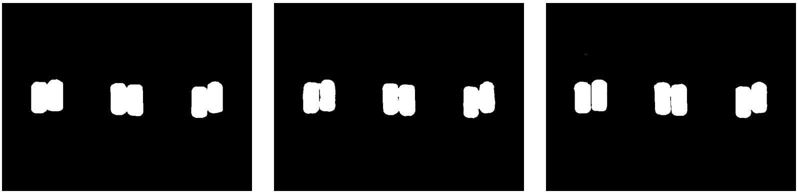
(h) Merged foreground mask from (e,f,g).

Figure 3.17 Foreground class prediction from the cropped image regions with 256×256 pixels. $\mathrm{IoU}_{\mathrm{GT}} \approx 0.9805$.

This method is beneficial for higher-resolution images. However, the inference time increases with the number of components to be detected. The inference time is quadrupled in the example with a prediction of three components compared to the one-step approach. Besides the increasing inference time, two models must be trained in parallel. One model is needed to identify the relevant regions for the cropped images. The second model then predicts the exact component pixels within the regions. Depending on the size of the regions, the cropped images do not need to be scaled down for model input. This prevents a loss of information.

Comparison Comparing the two-stage training with the result of the one-stage training on the image with the three hairpins shows an increase in accuracy. Figure 3.18 shows the human-labeled binary mask of the foreground class (a), the result of the one-stage training (b), and the result of the two-stage training (c). By cropping the image regions, the IoU compared to the ground truth improves to $\mathrm{IoU}_{GT} \approx 0.981$. Compared to the one-stage training, this is an improvement by about 0.02. This improvement is because the entire image in one-stage training is reduced from 720×540 to 256×256 pixels, resulting in a loss of information.

Additionally, the predicted mask is scaled back to the original image resolution, distorting the results. The direct comparison of the resulting binary masks of the one- and the two-stage training yields an IoU of 0.966 of the two binarized predictions.



(a) Human-labeled ground truth.

(b) Binary mask one-stage prediction. $\mathrm{IoU}_{\mathrm{GT}} \approx 0.964$

(c) Binary mask two-stage prediction. $\mathrm{IoU}_{\mathrm{GT}} \approx 0.981$

Figure 3.18 Comparison of the binarized foreground class result of the one-stage prediction (b) and the two-stage prediction (c).

The two-stage procedure is especially beneficial for images with higher resolution and more recorded components within one image. For example, the zoom factor can be reduced by using a high-resolution camera. Thus nine or twelve hairpin pairs can be captured in one camera image. The algorithm can then determine the weld position for each pin pair with sufficient accuracy. Despite the multiple inference time, this saves time compared to capturing each welding position separately. In both cases, the algorithm processes one image per position. However, the time for illumination and image acquisition is saved. Therefore, the two-step process can further increase the efficiency of the system by using a single camera image for multiple position determinations.

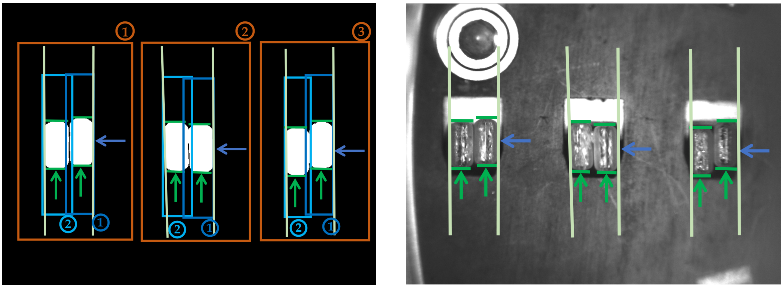
In the one- and two-stage procedure the algorithm uses a CNN to create a false color representation of the image that highlights the relevant regions. This representation image is then used for further processing. The detection of a single foreground class results in a binary image representing the component with the value one and the background with zero. For further classes, e. g., component and fixture, the representation of these classes must be supplemented by other pixel values in the image. The generated image is then further processed to calculate the coordinates of the weld position.

3.4.3 Result Validation

Semantic segmentation highlights the pixels with features relevant to the foreground classes and filters out the background and distracting features from the image. In addition, it harmonizes structures on the component surface. Thus, the ML algorithm can be considered a preprocessing image filter that eliminates the interfering image features and makes post-processing algorithms easier to implement and less error-prone.

Knowledge-Based Model The downstream component position detection is performed analogously to the procedure described on page 53 and Figure 3.1(c). It uses the knowledge-based algorithm but executes it on the false color representation instead of the camera image. In the example shown in Figure 3.19, this is a binary image. Compared to the usage on page 53, the algorithm is easy to parameterize. The ROIs (orange areas) can be large, and the hairpin width range (width of the blue boxes) can be generously restricted since no interfering elements need to be eliminated. An example of an interfering part on the stator can be seen at the top left of the camera image, which is no longer present in the semantic segmentation result. Also, the thresholds for the gray value change and the ranges within which the gray values are averaged do not need to be adapted to the data set. These are clearly defined in the false color representation resulting from the semantic segmentation algorithm. Alternatively, a simplified algorithm with fewer parameters can be applied to the false color representation.

The knowledge-based algorithm also measures the component size and detects a lateral offset or a gap between the copper rods. If the semantic segmentation algorithm assigns the wrong pixels to the component class, the algorithm indicates this as an error because the rule-based checks fail. In these cases, the part must be inspected manually before welding. The manual inspection involves checking whether the hardware or the image processing caused the error. This safeguarding prevents dangerous situations and faulty welds due to incorrect predictions. It also detects if the situation in the process changes significantly. Substantial deviations in the image data can, for example, be caused by a change in the lighting situation or a pre-processing step. In this



(a) Component position detection on the binary mask.

(b) Transfered boudaries of the detected edges to the camera image.

Figure 3.19 Component position detection with threshold-based post-processing on the false color representation. The orange rectangles represent three manually set ROIs. Within these, boundary detection is performed in the direction of the blue arrow. The algorithm detects the change in the grayscale value and defines the blue ROIs based on a shift that exceeds the threshold. These ROIs have a predefined width range in which a gray value change is searched for in the direction of the green arrows. The exceeding of a defined threshold defines the green boundaries. Figure (b) shows the transfer of the detected edges to the camera image.

case, the model may have to be retrained to make it more robust to this situation. However, deviating data can also be caused by wear of the cutting tool or an error in the upstream process chain, which must be corrected to ensure consistently good product quality. If the algorithm detects an increasing number of false detections, this is an indication that something has changed in the process. In this way, the CNN and the hybrid approach help to monitor the upstream process and to detect errors at an early stage.

In addition, this combination of ML and rule-based algorithms considering expert knowledge increases users' acceptance. While rule-based algorithms are already widely used in industrial environments, the hurdle to trusting ML algorithms is even greater. This behavior is mainly because the model's prediction cannot be completely understood, so wrong predictions cannot be easily verified. Research is still very active in uncertainty quantification, but incorrect model predictions are not always detected directly. With the hybrid approach, additional monitoring is provided by checking the results downstream, and the model does not have to be blindly trusted.

Uncertainty Estimation However, deviation situations in the process can also be detected based on the properties of the model prediction. Since the model predicts the probability per class, a conclusion can be made about the model's certainty and reliability based on these values. This correlation is only valid for well-calibrated models. A commonly used information measure is the entropy [146], which is often used to calculate the model uncertainty [100, 107, 145]. The entropy can be used as a measure of uncertainty since its value is maximal when the model assigns the same probability to each considered class. On the other hand, it is minimal when the model is sure of its decision. Through this analysis, the system detects out-of-distribution samples and can generate a warning or abort at the welding station.

Table 3.2 The table shows the ECE and entropy metrics $\mathcal{H}$ for the foreground class using different examples. The corresponding images, as well as the reliability and the class probability diagram are shown in Figure 3.20, 3.21, and in the Appendix C in Figure C.2.

	ECE	$\mathcal{H}(\hat{\boldsymbol{y}}_j)$	Image
Two pins	1.127	0.042	Figure 3.20 upper row
Two pins	1.388	0.047	Figure C.2 (Appendix)
One pin	2.913	0.051	Figure 3.20 lower row
Welded pins	-	0.105	Figure 3.21

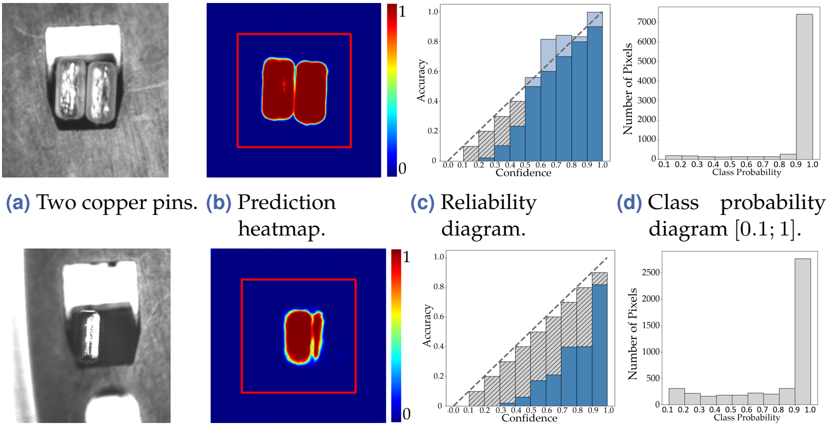
Following the segment confidence metric $\mathcal{H}$ of Mehrtash et al. [107], the entropy of the foreground class is used to determine the uncertainty of the predictions. The model calibration is derived using the expected calibration error (ECE), which evaluates the deviation between the estimated likelihood of a model and its actual likelihood by taking a weighted average over the absolute difference between accuracy and confidence. The smaller the ECE, the better the model is calibrated. For better visualization of the metrics, this chapter uses the class probability diagram to illustrate the uncertainty and the reliability diagram to show confidence versus accuracy. The prediction results are divided into $M = 10$ bins for a better overview.

Analogous to Mehrtash et al. [107], the evaluation uses a slightly smaller region around the segmented element of the foreground classes to compute the metrics and the diagrams. The background class, which takes up a large portion of the image, is usually predicted with high confidence and smooths the results for the foreground class. Additionally, the class probability diagram is scaled in the range $p(\hat{y}_{ij} = y_{ij}|\boldsymbol{x}_i, \boldsymbol{\theta}) \in [0.1, 1]$ to emphasize the remaining ranges better, regardless of the amount of background class pixels.

This section illustrates the results based on the class prediction of the cropped areas of the two-stage approach. The model is trained only on good examples, i. e., images containing two hairpins with no offset. Table 3.2 shows the results of the ECE and the uncertainty measure $\mathcal{H}$ for four different examples of $\mathbb{X}^{\text{test}}$. The first two examples are data samples that are similar to the training data set $\mathbb{X}^{\text{train}}$. They show two correctly pre-processed copper wire surfaces visible in the image. Figure 3.20(c) shows that the model is calibrated quite well. Furthermore, both the heatmap of the foreground class prediction and the class probability diagram show that the pixel-wise model prediction is confident. These aspects are also evident in the values of ECE and $\mathcal{H}$, which are lower than for the third and fourth examples.

In the third example, one hairpin is missing. In this case, the downstream rule-based verification of the hybrid AI approach detects an error since only one pin surface is detected. However, the slightly increased value of $\mathcal{H}$ shows a scatter of the class assignment and, thus, a higher uncertainty. Such an image was not included in the training data set. A mask was created to calculate the ECE and reliability diagram that only highlights the existing pin. Since the existing pin matches the features of the training data, it is detected with high confidence, as the heatmap in Figure 3.20(f) shows. Furthermore, only minor uncertainties around the pin's border area are visible. The model predicts a background class for most of the area where the second pin should be located. Thus the downstream rule-based approach is quite reasonable.

The fourth example consists of an already welded pin pair. No welded pin pairs were included in the training because the model is supposed to detect the pin surface still to be welded. Thus, this example represents an out-of-distribution sample. Since there is no area to cover in this example, no ECE is calculated. The model could not detect a pin surface,



(a) Two copper pins. **(b)** Prediction heatmap. **(c)** Reliability diagram. **(d)** Class probability diagram $[0.1;1]$.

(e) One missing copper pin. **(f)** Prediction heatmap. **(g)** Reliability diagram. **(h)** Class probability diagram $[0.1;1]$.

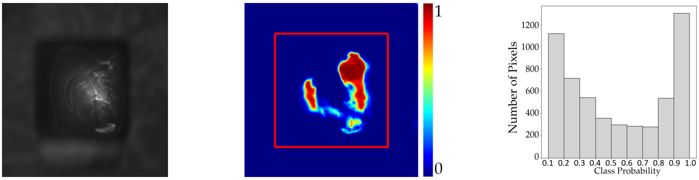
Figure 3.20 Calibration and out-of-distribution detection. The model is trained on good examples containing two unwelded copper pins in the image. Figures (b) and (f) show the predictions of the foreground class of the samples (a) and (e). In addition, the reliability diagram and the class probability diagram of the foreground class are given. For better illustration, the class probability diagram is limited to the range $[0.1;1]$ because of the many background pixels predicted with high certainty. The bottom line represents an out-of-distribution sample.

resulting in many uncertain artifacts in the prediction. This is shown in the class probability diagram and an increasing value of $\mathcal{H}$.

Figures 3.20 and 3.21 show the reliability diagram and the class probability diagram for the different examples besides the camera image and the prediction of the foreground class. The representation of the second example of a good pin pair is shown in Appendix C.

3.5 Protective Device Contamination

Depending on the application, a protective device shields the component from spatter and other deposits during laser welding. For this purpose, a so-called welding mask is attached around the welding position. The protective device has an opening in the center through which the component is processed with the laser beam. The device covers the rest of



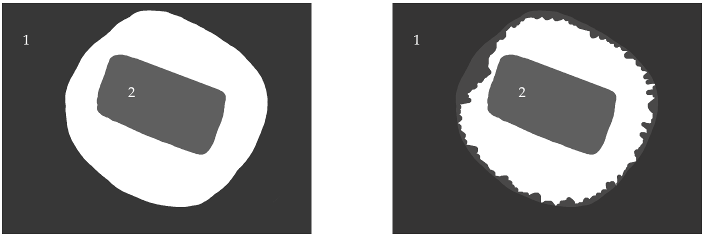
(a) Welded pin pair.

(b) Prediction heatmap.

(c) Class probability diagram $[0.1; 1]$.

Figure 3.21 Figure (b) shows the predictions of the foreground class of the already welded pin pair from the image in (a). Figure (c) shows the class probability restricted to the range $[0.1; 1]$.

the components. As a result, spatter and deposits caused by settling metal vapors are intercepted by this device and do not settle on the component. This prevents undesirable connections caused by spatters and other damage to the part.



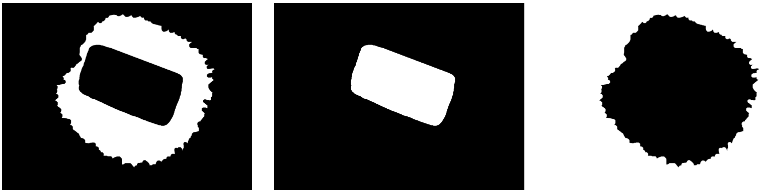
(a) Protective device in a good condition.

(b) Protective device with deposits.

Figure 3.22 Schematic drawing of a coaxial view of a welding position (2), in which the remaining component around the welding position is protected by a protective device (1).

The protective device becomes increasingly dirty due to material deposits. Especially the deposits, which settle at the edge of the opening, can lead to problems in the welding process. In spatter-prone processes, this happens more quickly. Especially areas already narrowed by previous deposits are susceptible to further deposits. As a result, these areas often quickly grow into the welding position, making the welding process impossible. Therefore, the device must be cleaned or replaced frequently. This is usually done at regular intervals, regardless of how badly the mask has been affected. Depending on the position and quantity of the deposits, the time interval until the device must be cleaned can vary greatly. Figure 3.22 symbolically shows the view of a welding position covered with a welding mask. In Figure 3.22(a) the welding mask is shown in a clean condition, while in Figure 3.22(b) the mask already has deposits on the edge. This results in the uneven structures shown in the schematic drawing.

Therefore, an approach is proposed that uses the setup from Section 2.2.4 with a camera mounted coaxially to the laser beam to monitor the degree of contamination of the protective device. Due to the on-axis arrangement of the camera, the component, the welding mask, and possible deposits are captured. By detecting the welding position using a CNN, presented in Section 3.4.2, the area (2) from Figure 3.22 can be detected with pixel accuracy. Extending the presented approach to a three-class problem also allows the model to detect the welding mask region (1) accurately. The protective device is mounted closer to the laser optics than the component. Since the camera is focused on the welding position, the protective device is too close and not in the camera's focus. As a result, it is slightly blurred in the image, which further simplifies the distinction. An example of a one-hot encoded mask for a semantic segmentation network for the three-class problem is shown in Figure 3.23. In addition to the background and the welding position class, a third class is created, representing the protective device.



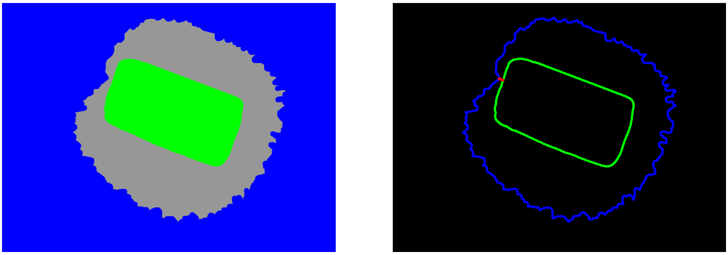
(a) Background mask. (b) Weld position mask. (c) Protection device mask.

Figure 3.23 One-hot-encoded mask for a semantic segmentation network.

Since the image data is analogous to the data from the component position detection, the network selection can refer to the results of the network architecture evaluation from Section 3.4.1. An SDU-Net architecture with the input dimension 256×256 and filter sizes $n_{\text{out}} =$

$\{16, 32, 64, 128\}$ is used. The filter size of the last 1×1 convolutional layer is increased from two to three to get three output layers. For training, the model also uses the categorical focal loss with $\alpha = 0.25$ and $\gamma = 2$ as in the previous use case. This loss function can handle unequal class ratios since it gives a high weight to the wrong results. In addition, the model uses the ADAM optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate starts at $\epsilon = 0.001$ and is reduced during the training after three epochs without improvement.

The informations about the welding position and the protective cover area must be combined to monitor the protective mask's condition and determine when it needs to be cleaned. For this purpose, the two one-step coded binary results of the welding position class and the protective device class are used. First, their contours are determined by a contour search, according to Suzuki and Abe [157]. Finally, an algorithm determines the smallest distance by comparing the distances between the individual points of the contours. The procedure is shown symbolically in Figure 3.24. For better illustration, the one-hot encoded result classes are shown in one image in different colors.



(a) Representation of the one-hot encoded masks of the welding position (green), the protective device (blue), and the background (gray) in one image.

(b) Distance determination based on the contours of the one-hot encoded mask. The smallest distance is marked in red.

Figure 3.24 The distance between the welding position and the protective device is determined based on the semantic segmentation class results.

If the alignment between the welding position and the protective device is performed each time before the welding process, the cyclical cleaning of the protective device is not necessary. Nevertheless, faulty and dangerous welds caused by excessive contamination are prevented. In combination with the component position detection, the adjustment can be performed without additional computing time. Since the network architecture of the SDU-Net has been retained, the inference time is constant. The subsequent contour search and the distance calculation can be performed parallel to the welding process and trigger the cleaning process afterward. However, the labeling effort for training the model is increased since the protective device must also be marked with pixel accuracy in addition to the welding position.

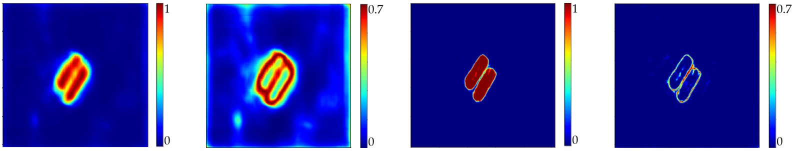
3.6 Labeling Process

Building a semantic segmentation model involves an increased labeling effort due to the accurate annotation of each pixel. Process knowledge is required to annotate the data correctly. Even if the problem of defining the component position may seem trivial at the beginning, the component boundary cannot always be precisely defined by someone unfamiliar with the process. Using the example of squeezed hairpins, burrs or slightly beveled surfaces can occur at the copper pin edge, which cannot always be detected in the camera image without process knowledge. The process expert is aware of the previous production steps and can assign the image data better than, for example, an external creator of data labels or an AI expert.

Similar to Gorriz et al. [58] describing the problems with medical image interpretation, the time and associated costs of data labeling are also problematic in industrial manufacturing. In addition, the advantage that laser welding offers for small quantities of parts should not be negated by time-consuming and cost-intensive data labeling, which is required for finding the welding position. Section 3.4.1 shows that for part position detection in the welding process, only a small amount of training data is required due to the similarity of the database. Nevertheless, it must be ensured that the existing data variance is covered.

Reduction of the Required Training Images Active learning (AL) is a well-established approach to reducing labeling efforts by iteratively selecting a subset of informative examples from an extensive collection of unlabeled images. Pool-based sampling essentially aims to query only the data sets from a large pool of unlabeled samples for labeling that are more likely to result in more accurate models when used in place of other data. Through the selection process, only highly informative data are used in training. As a result, the time and financial costs associated with labeling are reduced [100]. Several metrics can be used to determine a proper labeling order of the data. In the following, the entropy of the softmax probability, Monte Carlo Dropout (MCD), and the structure similarity index (SSI) are examined to determine the labeling order.

As mentioned earlier, **entropy** is a commonly used information measure in uncertainty evaluation, which is also often used in the literature for AL [100, 145]. The data whose posterior probability distribution yields the highest entropy are estimated to have the most significant positive impact on the model's performance. The entropy or uncertainty measure $\mathcal{H}$ considers all classes for estimating the labeling order. Figure 3.25 shows the uncertainty per pixel based on the test image in Figure 3.13(a). Summing up the pixel values gives the uncertainty score per data sample.



(a) Softmax probability of the foreground class.

(b) Entropy of the softmax probability. $\mathcal{H} = 0.1166$.

(c) Softmax probability of the foreground class.

(d) Entropy of the softmax probability. $\mathcal{H} = 0.0025$.

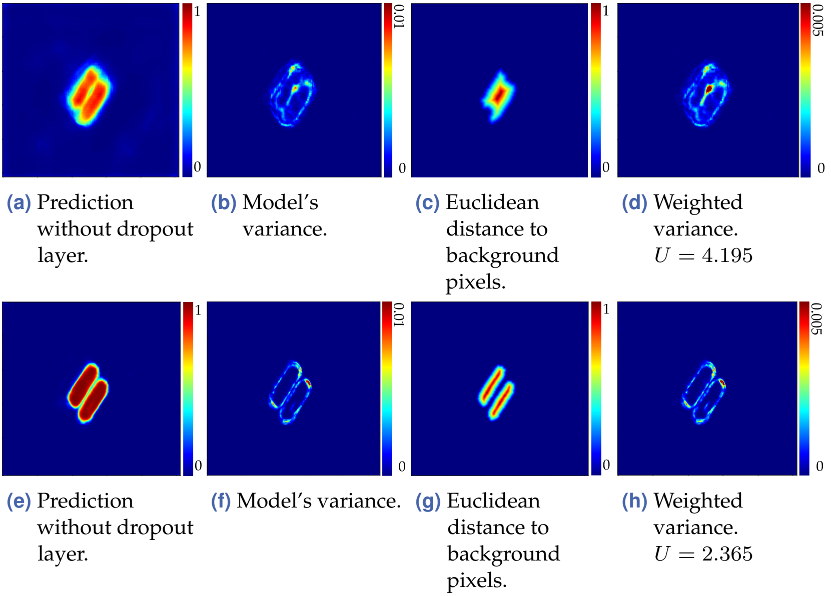
Figure 3.25 Figures (a) and (c) show the softmax activation of an uncertain and a certain model, while (b) and (d) show the pixel-wise entropy of the softmax probability. The evaluation is done for the test image in Figure 3.13(a).

The softmax function approximates the relative likelihood between classes, which may not always equate to an overall measure of model uncertainty [48]. In an example of Kendall et al. [82], the bias of the result due to the relative representation of the class likelihood is illustrated. Furthermore, there is a risk that the prediction probability is still high even if the class assignment is wrong [48]. Therefore, this metric would not consider these incorrect predictions. For a two-class problem, as is common in component recognition, the entropy result is more meaningful and less error-prone than for multi-class problems. Similarly, deeper and wider models tend to have poorer calibration than small models [164]. A well-calibrated model is required for the metric to be valid.

Unlike the highly constrained inference time at the plant, more time-consuming uncertainty estimates can be used in the labeling process. Another way to determine epistemic uncertainty is, for example, using ensemble training or **MCD** [42, 58, 100]. Gal and Ghahramani [48] show how dropout training in deep neural networks can be represented as approximate Bayesian inference in deep Gaussian processes. Derived from this insight, they offer a method for establishing uncertainty in the model by creating a MCD ensemble [47]. As explained in Section 2.1.2, ensemble methods are suitable for the regularization of models. Because different models usually make various errors, they perform better in an overall prediction. However, considering the errors and focusing on recurrent errors, it can be seen that the data samples representing the corresponding features are underrepresented in the training data set $\mathbb{X}^{\text{train}}$. Instead of multiple models, the MCD uses one model with dropout layers. The results are samples of the posterior distribution of models by randomly disabling network activations according to a Bernoulli distribution with a base probability d_p per layer during inference. Since the distribution is not tractable, this must be approximated, which can be done with variational inference [59].

Kendall et al. [82] propose an optimal value for the dropout probability of $d_p = 0.5$. Inspired by their findings on probabilistic network architectures variants, the dropout layers are added to the central layers of the SDU-Net architecture. The detailed definition of the used SDU-Net architecture with dropout layers is shown in Appendix D. The procedure follows that of DeVries and Taylor [27], using $d_p = 0.5$ and $T = 20$ model predictions to determine the prediction variance. Figure 3.26 shows a result of a more uncertain prediction and a more confident one.

The pixel-wise uncertainty must be converted into a numerical score to estimate the predictive reliability of the sample. Then, the uncertainty map is summed up to obtain a higher score for the ambiguous segmentations. Since the differences mainly occur at single pixels of the contour



(a) Prediction without dropout layer.

(b) Model's variance.

(c) Euclidean distance to background pixels.

(d) Weighted variance. $U = 4.195$

(e) Prediction without dropout layer.

(f) Model's variance.

(g) Euclidean distance to background pixels.

(h) Weighted variance. $U = 2.365$

Figure 3.26 Figures (a) and (e) show the softmax activation of an uncertain and a certain model without using dropout layers in the prediction. (b) and (f) show the pixel-wise model variance. (c) and (g) show the per-pixel distance to the nearest background pixel, while (d) and (h) show the weighted variance. The evaluation uses $d_p = 0.5$ and $T = 20$ model predictions for the test image in Figure 3.13(a).

edge, where they do not significantly affect the result, the evaluation performs value weighting based on pixel position, similar to Gorriz et al. [58]. The Euclidean distance to the nearest background pixel is used as the weighting factor for the variance of each foreground pixel. Uncertainties located at the edge of the contour are, therefore, less significant than uncertainties within the contour. However, since the variance of the model prediction is more informative, it should not be possible to weight it to 0 due to the class assignment of the pixels. Therefore, an average of the variance and the variance weighted by the Euclidean distance is used as the weighted value.

The disadvantage of MCD or ensemble training is that it requires more training effort. Ensemble training requires several models to be trained in parallel. In contrast, MCD needs only one model. However, as described in Section 2.1.2, the model must be larger due to the capacity reduction caused by the dropout and requires more training iterations for a good result. In addition, the most significant uncertainties usually occur at the edge of the object contour, which must be considered in the evaluation [58].

Regardless of the model uncertainty, the coverage of the total variance of the data pool by the training set $\mathbb{X}^{\text{train}}$ can be considered a relevant factor for selecting the label order. To generalize well on the entire data set, the model must learn all relevant features in training. Using the **structural similarity index (SSI)** [167], the similarity between images can be determined. The luminance, contrast, and structure of the images are considered with

$$l(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2\mu_{\boldsymbol{x}_i}\mu_{\boldsymbol{x}_j} + c_1}{\mu^2_{\boldsymbol{x}_i}\mu^2_{\boldsymbol{x}_j} + c_1}, \tag{3.1}$$

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2\sigma_{\boldsymbol{x}_i}\sigma_{\boldsymbol{x}_j} + c_2}{\sigma^2_{\boldsymbol{x}_i}\sigma^2_{\boldsymbol{x}_j} + c_2}, \tag{3.2}$$

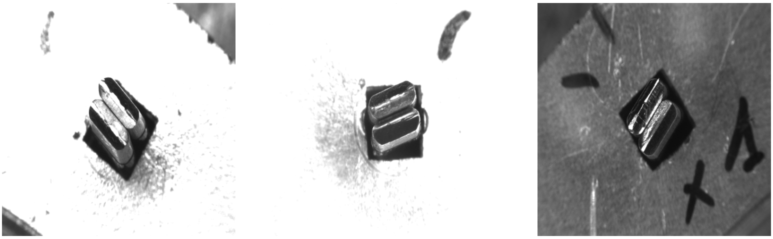
$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2\sigma_{\boldsymbol{x}_i\boldsymbol{x}_j} + c_3}{\sigma_{\boldsymbol{x}_i}\sigma_{\boldsymbol{x}_j} + c_3}, \tag{3.3}$$

where $\mu_{\boldsymbol{x}_i}, \mu_{\boldsymbol{x}_j}$ are the mean, $\sigma_{\boldsymbol{x}_i}, \sigma_{\boldsymbol{x}_j}$ the standard deviation and $\sigma_{\boldsymbol{x}_i\boldsymbol{x}_j}$ is the covariance of two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The constants c_1 and c_2 are defined by $c_1 = (K_1L)^2$ and $c_2 = (K_2L)^2$ where L is the dynamic range of the pixel value and K_1 and K_2 are defined parameters. A combination of these comparisons defines the SSI with a weighing $\alpha > 0, \beta > 0, \gamma > 0$ as

$$\text{SSI} = [l(x,y)]^\alpha * [c(x,y)]^\beta * [s(x,y)]^\gamma. \tag{3.4}$$

The weighting is set to $\alpha = \beta = \gamma = 1$, the constant $c_3 = \frac{c_2}{2}$ and $K_1 = 0.01$ and $K_2 = 0.03$, analog the definition from Wang et al. [167].

Comparing the fundamental image similarities and using very different data sets for training can represent more variance in the training data set. Using this metric, the remaining data in the data pool are each compared to the training data in training set $\mathbb{X}^{\text{train}}$. Figure 3.27 gives an example. The subfigures (b) and (c) are compared with the image in (a).



(a) Hairpin image. **(b)** SSI $= 0.679$. **(c)** SSI $= 0.346$.

Figure 3.27 Comparison of the SSI. The subfigures (b) and (c) are compared with the image in (a)

The advantage of this method is that the actual model architecture does not need to be modified, and computing the metric requires little computational power. However, the disadvantage is that the computation is independent of the already-created model. Therefore, no conclusions can be drawn about the actual model performance. Thus, this data-centric approach focuses on the database, independent of the actual model performance.

The following diagram in Figure 3.28 compares the training process with the different methods. Images are added to the training step by step. The training starts with the same sample each time and adds another sample after each epoch until a total number of $n = 50$ training samples is reached. Each epoch is trained with 200 steps and batch size $BS = 2$. In addition, data augmentation with translation, shear, zoom rotation, and flipping is applied to the training images to prevent overfitting. The order of the samples to be added is determined based on the different metrics. The validation data set remains unchanged and contains $n = 500$ samples. The graphs show the average and the standard deviation of 20 training processes. The data set from Section 3.4.1 or Appendix A with 50 training images and the splits with index 3 and 7 are used.

The accuracy of $\mathbb{X}^{\text{val}}$ varies and increases differently for the different methods. MCD shows the primary disadvantage of the larger recommended model size since the dropout reduces the model capacity. In addition, a larger number of training data and training iterations are usually required. The results in Figure 3.28 show these disadvantages in a slower increasing accuracy. Due to the dropout layers, the model

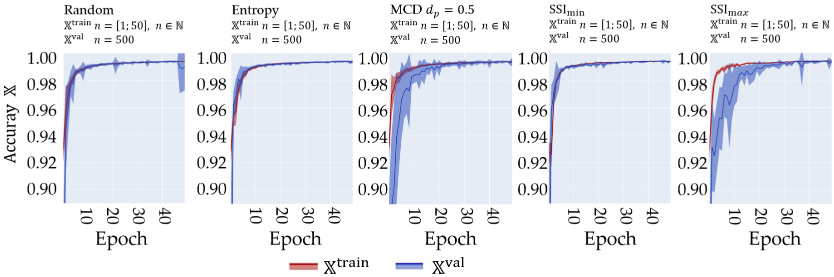


Figure 3.28 Comparison of the effects of the order of labeling the training images. Each epoch is trained with 200 steps, and after each epoch, a new image is added to the training data set. The image is selected based on the corresponding metrics. The graphs show the mean and standard deviations of 20 training sessions.

requires more epochs to produce good results on $\mathbb{X}^{\text{val}}$. Plotting the score SSI_{max} in Figure 3.28 shows that the model performs worse than the other methods due to successive labeling of very similar samples. Since only very identical images are available in training at the beginning, the model tends to memorize these features and generalize worse to the variance of the entire data set. For this reason, the model performs lower on average across all 500 samples from $\mathbb{X}^{\text{val}}$ and requires more epochs to learn the relevant features. This order could theoretically occur in random selection. The results of softmax entropy and SSI_{min} both perform well. In both cases, the variance of $\mathbb{X}^{\text{val}}$ is already covered with a small data pool $\mathbb{X}^{\text{train}}$. Moreover, the metrics can be used to find a measure to determine the added value of labeling new data. If the remaining unlabeled images in the data pool $\mathbb{X}^{\text{val}}$ all have very high similarity to the patterns in $\mathbb{X}^{\text{train}}$ and thus have high SSI, their features do not contribute much to model performance. The same conclusion can be drawn if the current model can predict all remaining images with low entropy. Also, in this case, the training data set sufficiently captures the image features.

Label Creation Effort A second aspect of AL is reducing the human effort required to create a label. For example, this effort can be measured by the number of clicks during the annotation process [100, 145]. Gorriz et al. [58] show how a good training data selection can drastically reduce labeling effort. Moreover, they suggest the data sample with the highest prediction uncertainty as the label order. In their cost-effective active learning approach, they also propose automatically generating labels for the samples with the highest prediction probability, i. e., those with an entropy smaller than a defined threshold. These labels are generated without human intervention. Analogous to the approach of Wang et al. [165], who propose the automatic labeling for a classification problem, they use predictions of the actual model as labels for a segmentation problem. This procedure offers the advantage of no human labeling effort. Since the samples are usually very similar to the labeled samples, this can be seen as a form of data augmentation that makes feature learning in CNN more robust. The approach is based on the curriculum learning principle, in which a model learns by gradually moving from simple to more complex patterns during training. As a result, the diversity of the training patterns increases [11].

This approach entails the risk of wrongly annotated pixels and, thus, the acquisition of wrong mask features. This risk is primarily because of the transfer from the classification to the semantic segmentation. In the example of burr formation or flattened hairpin surface mentioned earlier, an automated labeling process could set the boundary incorrectly for pseudo-labeled samples. However, the entropy of this entire sample would still be low. This risk can be reduced by adding a human feedback process. Nevertheless, early prediction with the current model should be possible to reduce human labeling effort. In particular, this works very well and saves a lot of effort for samples similar to the already labeled samples from $\mathbb{X}^{\text{train}}$. The optimal label order of the data plays an essential role because it influences the quality of the early predictions. However, human feedback is incorporated before the images are included in the training process. The human annotator can correct the prediction previously or add it directly to the training data set if the prediction is already exact. In addition to human intervention in the automatic generation of labels, algorithmic monitoring of the generated labels is helpful. For example, artifacts are detected, i. e., locally individual pixels assigned to a different class than the surrounding pixels. The algorithm displays a warning before adding a sample to the training pool. Similarly, the evaluation suggests performing further checks, for example, a plausibility check of the shape, size, or number of labeled surfaces. These checks are helpful for both the labels from early predictions and human-labeled data to ensure a correctly labeled training data set of good quality.

3.7 Evaluation and Discussion

The evaluation compares the weld positions found by the algorithm directly on the camera image and on the false color representation resulting from the CNN, which marks the component-relevant pixels. The CNN is used as a pre-processing step. After that, the actual threshold-based algorithm detects the weld position. Thus, only the image pre-processing needs to be adapted for the algorithm to compare the resulting positions.

Comparison with Knowledge-Based System For the evaluation, images are taken on the system before the laser welding process. Welding positions are then determined using the different approaches, compared with each other, and their deviation is calculated. Both approaches have slightly different requirements for image properties. For example, the ML approach requires image data containing structures to assign the image features to the classes. On the other hand, these structures disturb the rule-based approach, and overexposed images, which contain fewer textures, can be processed better. Considering this fact, images with optimized exposure time and gain are acquired for both algorithms.

In the first approach, without ML, the images are preprocessed with a Gaussian filter to smooth the image content. Then the filtered image is processed with a threshold-based algorithm that identifies the pin edges and derives the corresponding weld position. The second approach uses ML-based semantic segmentation to create a false color representation highlighting the relevant pixels. This representation image serves as input for the threshold-based algorithm to determine the weld position.

The welding position and the component's rotation are derived based on the detected edges. Since welding uses defined geometries, component rotation is also crucial for a good result. Three examples in Figure 3.29 show definitions of the welding position. The light blue dot

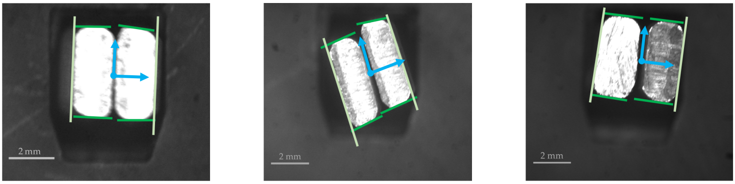


Figure 3.29 The pictures show examples of calculated welding positions with different orientations. The light blue dot and the arrows indicate the coordinates and orientation.

marks the calculated welding position, and the arrows symbolize the orientation. Based on this coordinate system, the translation, and rotation of the weld geometry can be calculated in the laser control. In addition, welding geometries or the laser power can be adjusted based on the detected gap or offset. If the values, such as the size of the joining parts, gap, offset, or rotation, exceed the predefined limit values, an error is reported instead of the welding position.

Comparing the algorithms can be automated by matching the welding positions found using the same input image. Only if the results differ strongly a manual inspection of the samples must be done to determine the cause of the deviation. The evaluation has considered different processes using a variety of data. Since most processes involve customer data, not all results can be published.

The following result was obtained in comparing an evaluation of a test data set $\mathbb{X}^{\text{test}}$ of $n = 9510$ data samples. Performing the algorithm directly on the camera image classifies 79 data samples as erroneous. Hardware-related error classes are a gap or offset of the pins, an incorrect pin size due to the pre-processing steps, or a missing pin. However, due to an incorrectly detected edge, the image processing results can also incorrectly calculate the pin size, as shown in Figure 3.2(c). By using the false color representation, which is created based on the result of the CNN, the algorithm only classifies 19 samples as erroneous. The remaining 60 positions, which were previously defectively classified, meet the quality criteria and can be welded. During the manual inspection of the samples, this result can be confirmed. The remaining defect cases are due to an error in the pre-processing steps or an offset or gap between the pins due to defective clamping. However, the remaining samples meet the quality criteria.

Table 3.3 Averaged absolute deviations of the welding positions with and without ML algorithm.

	Average	Standard deviation
$\lvert\Delta x\rvert$	0.016 mm	0.017 mm
$\lvert\Delta y\rvert$	0.017 mm	0.019 mm
$\lvert\Delta\alpha_a\rvert$	0.510°	0.614°

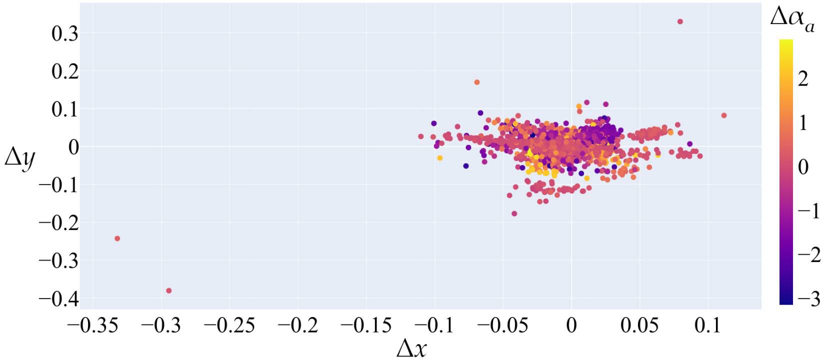
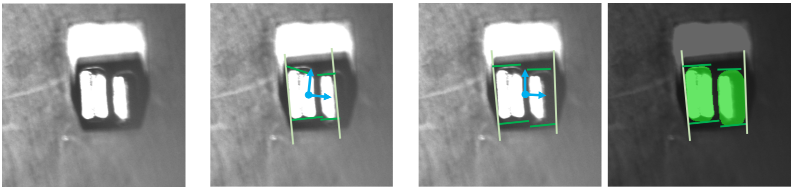


Figure 3.30 The diagram compares the offset of the detected weld positions on the camera images and the AI-based false color representations. The axes show the deviation of the weld position in x and y, while the colors represent the deviation of the angle α_a of the defined coordinate system.

The averaged and standard deviations of the absolute distance of the detected weld positions in the x-direction, y-direction, and orientation angle of the 9431 data samples are listed in Table 3.3. The distance values are also shown in the diagram in Figure 3.30. This evaluation considers only samples for which both algorithms did not detect any error. Again, the more significant deviation is due to incorrectly detected boundaries on the camera image without using semantic segmentation. Figure 3.31 shows an example. The right side of the second pin is shaded in the camera image due to its triangular cut structure. Directly on the camera image, the algorithm detects the boundary of the pin area in the center of the pin (Figure 3.31(b)). In contrast, the ML algorithm correctly highlights the entire pin surface, shown in Figure 3.31(d) in an overlaid representation. The boundaries are thus placed on the edge of the pin surface based on the false color representation. In addition, the length is detected shortened in (b) due to the lighting situation. However, if the size and the offset derived from the edges found are within the defined limits in both algorithms, this leads to different definitions of the weld position and the angle.



(a) Hairpin image with shadings.

(b) Detected edges on the camera image.

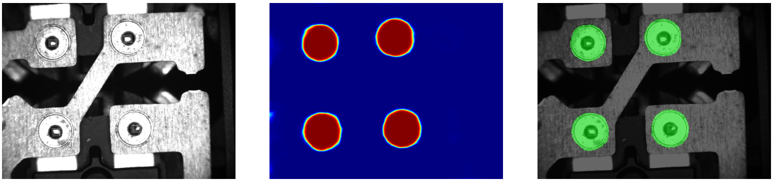
(c) Detected edges on the false color representation.

(d) Overlaid representation of the ML result.

Figure 3.31 Fiugre (a) shows the camera image of a pin pair where the right side is shaded. Figure (b) shows the boundaries found directly on the camera image. (d) represents the semantic segmentation result with the associated class pixels overlaid in green. Based on this, the edges marked in (c) are detected.

Extension to other Data Sets The great advantage of this approach is that it works for different image data of various components without the need to develop complex algorithms manually. The manual effort is limited to the labeling of training data. As shown in Section 3.4.1, the amount of 25-50 data samples is sufficient.

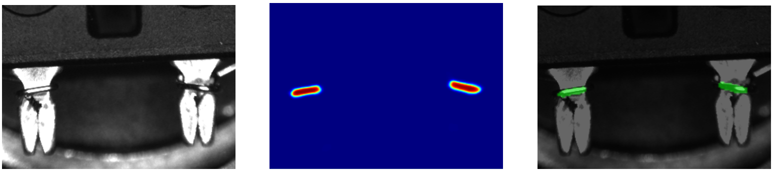
The following example in Figure 3.32 shows circular shapes on a component that are to be detected by an algorithm. The component surface has strong structures, and the shading within these circles is partially different. Since the ML algorithm considers multiple features and contexts, recognizing the relevant surfaces still works. Thus a binary mask with the relevant geometries can be generated, which is used for further processing. Therefore, the subsequent algorithms no longer have to deal with challenges such as surface structures or interfering elements.



(a) Image example part. (b) Foreground class. (c) Overlaid image.

Figure 3.32 Example of a component on which the circular shape is to be detected. Figure (c) shows an overlaid representation of the camera image and the binary mask of the predicted component class (green).

Figure 3.33 shows another example. Two wires are to be detected in the camera image. Due to the reflective material and the curved shape, they reflect the light in different directions, resulting in shading with different gray values. Similar to Figure 3.31, the shaded areas can be reliably detected.



(a) Image example part. (b) Foreground class. (c) Overlaid image.

Figure 3.33 Example of a component where two wires are to be detected. These wires have different reflections and, therefore, various grayscale values. Figure (c) shows an overlaid representation of the camera image and the binary mask of the predicted component class (green).

The algorithm achieves similarly good results for problems such as the bloom effect that can occur in camera images. Due to the often highly reflective metallic surfaces used in laser welding, cameras with a CCD sensor often produce local overexposures which affect also the surrounding pixels. It is not easy to distinguish between component pixels and pixels illuminated by bloom based on the actual pixel value. The image is bright in the blooming and the area to be detected. However, if the effect is still detectable in the image context, a distinction can be made using semantic segmentation.

Extending the algorithm with semantic segmentation, images with complex structures can also be processed. This eliminates the need to develop complex customer-specific image processing algorithms, which saves time when setting up the system. In addition, in some cases, more cost-effective and faster pre-processing steps can be used in the production process before welding. For example, the algorithm can handle different surface textures, and the pre-processing method does not have to create smooth surfaces reflecting light directly into the optics. This possibility of replacing pre-processing steps was illustrated in this chapter using the example of the hairpin from electromobility. At the same time, the approach provides a way to reliably monitor part geometry and part positions and observe protective device contamination in pre-process monitoring without needing additional hardware. In particular, the lateral and radial misalignment of weld components, which has a negative influence on the welded joint, are monitored by the algorithm. The model generalizes well enough to handle reflections and minor changes. However, for more significant changes, the algorithm no longer works reliably. Thus, undesirable process deviations are detected, and the results of the previous process step are monitored. The hybrid approach of combining ML and rule-based algorithms increases user acceptance. Rule-based algorithms, already widely used in the industrial environment, monitor the result of the ML algorithm and can provide a warning in the event of significant deviations.

The network architecture is defined as very small with few parameters, so the network learns good results quickly. Extensions of the algorithm by adding information about the component geometry are conceivable. For example, size information from the downstream rule-based algorithm can be incorporated into the loss function during training. This can further speed up and improve the training process since the algorithm is constrained to the correct part size and shape. In addition, information from the CAD system about the part could also be extracted and used during training. However, when enriching the algorithm with specific information, care must be taken to ensure the model is not susceptible to false detections of the correct shape and size. This could lead, in turn, to erroneous results that are no longer recognized by the downstream algorithm. Further extensions of the algorithm are conceivable, for example, concerning process-specific data augmentation. Since the algorithm is used in laser welding processes, the images often have similar characteristics. For example, previous welds contaminate fixtures with spatter and dirt. These deposits are often reflected in the camera image. The algorithm can be made more robust by artificially enhancing the training data with artifacts such as spatters and other reflective clutter in the training process.

Implementability in Industrial Manufacturing An essential factor for realizing the application of ML algorithms in the welding position detection procedure is the fast adaptation to new processes. Therefore, the focus is on a low set-up effort, as well as a fast and easy algorithm adaptability. Furthermore, the system and, consequently, the algorithm should be configurable without knowledge of AI, computer science, or programming.

This chapter shows that the same model architecture and training method can realize the processing of data from different production lines. Thus, the optimized model architecture and two predefined training procedures achieve a sufficiently automated backend for generating a ML model. As shown in Section 3.4.1, a small network architecture based on the SDU-Net structure has prevailed over the other architectures. For training, the categorical focal loss has proven its suitability. This model structure works for different types of image data and single- or multi-class approaches and does not need to be customized for individual problems. The method of single or multi-stage training from Section 3.4.2 is specified depending on the image resolution and the number of components to be detected.

However, the process cannot be fully automated. Since semantic segmentation is a supervised process, labeled data is required. With the AL methods, shown in Section 3.6, the number of images to be labeled can be reduced. Moreover, early predictions by the current model reduce the data labeling effort. Since the data within a production line is highly similar, the prediction of unseen data is often very good, and only a few pixels need to be adjusted. Otherwise, pixel-precise labeling is very time-consuming. Moreover, the annotator's attention and concentration decrease if too much time is spent on the labeling process. With the help of an overlay presentation of the image and the annotation, which shows different classes in different colors, the labeling process can be supported and simplified for the user.

The predefined framework, including model architecture and training procedures, and the support in the labeling process make it possible for anyone to teach the ML model. This setting removes a significant barrier to use ML in industrial manufacturing. On the one hand, the labeling effort is reduced, and on the other hand, it can be carried out directly by the process expert in the plant without having to rely on external help. This possibility not only facilitates and speeds up the process but also increases the acceptance of the ML solution. Furthermore, the independent training and the insights into the training process through the early predictions increase the trust in the ML algorithm. Users are not only confronted with a black box model that is entirely new to them and to which they tend to be critical according to different surveys [12, 65, 124]. A model can also be adapted quickly and without much effort if it does not yet recognize specific patterns well.

4 In-Process Monitoring

4.1 Introduction

The previous chapter shows how robust component position detection and monitoring of the joining component geometry before the actual welding process avoids welding defects. Nevertheless, laser welding is a very complex process with many influencing factors. Therefore, poor welding results can occur due to process instabilities, regardless of the pre-processing steps. These instabilities in the process can be attributed to various causes and appear more frequently depending on the welding task and material properties.

In hairpin welding, the laser melts copper wires, which are joined to form a coil winding. As mentioned in Section 2.2.4, copper has challenging conditions for laser welding due to its material properties. The absorption rate of infrared laser light at room temperature is low on the copper surface. For laser light with a wavelength of $\lambda \approx 1000\,\mathrm{nm}$, the absorption ratio is about 5%. Therefore, for deep penetration welding, a high beam parameter quotient must be realized by a high incident laser power P and a small beam waist diameter [53]. The absorption ratio of copper materials exhibits substantial variations of up to 10% depending on surface roughness and oxidation [39]. In addition, it changes depending on the workpiece temperature and increases abruptly during the phase transition from solid to liquid. For example, in deep penetration welding, this abrupt increase happens during the formation of a vapor capillary as the process melts the material. In addition, the material's thermal conductivity decreases as the workpiece temperature rises, which can lead to harmful interactions in conjunction with increased absorption. At a melt temperature of 1600 °C, copper has a comparatively low viscosity of $2.10\,\mathrm{N\,s/m^2}$. The currents generated in the melt pool by the continuous melting of the material are thus only slightly counteracted. As a result, the movements extend to the entire melt pool. According to Fabbro et al. [41], the turbulent melt pool at the back of the keyhole causes weld spatter. Fujinaga et al. [45] also conclude that an unstable keyhole results in ejections and pores. Despite the challenging conditions, different approaches can achieve higher stability in the welding process of copper. The following section mentions a few possibilities.

For example, the absorption ratio of copper increases in the solid state at short wavelengths. Figure 4.1 shows the absorption at room temperature and perpendicular beam incidence on a highly polished surface for various materials [32]. The graph shows that the degree of absorption for copper increases sharply at a wavelength of $\lambda \approx 600\,\text{nm}$. Therefore, different approaches use blue ($\lambda \approx 450\,\text{nm}$) or green ($\lambda \approx 515\,\text{nm}$) laser light to weld copper [32, 45]. Due to the higher absorption, the effects of surface conditions or oxidation fluctuations are smaller. In addition, low intensity of the laser beam is necessary since more laser power can be used to generate melt [32].

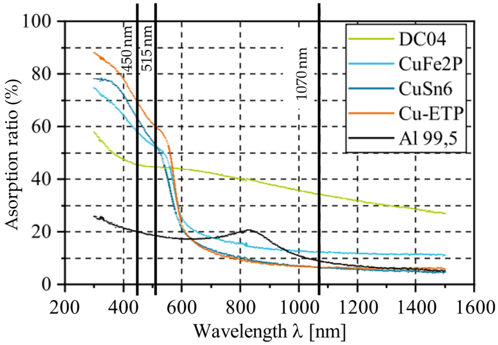


Figure 4.1 Absorption rates of copper (Cu-ETP), copper alloys (CuFe2P, CuSn6), aluminum (Al), and steel (DC04) surfaces at room temperature over the wavelength. The diagram is adopted from the publication of Helm et al. [62].

The disadvantage of commercially available laser sources in shorter wavelength is the restricted laser power of $P \approx 1\,\text{kW}$, which limits the weld penetration depth [13, 33]. To achieve greater weld depth, pulsed systems [2], or combinations of multiple continuous wave lasers in a single fiber can be used [33]. Future developments of devices with green laser powers up to several kW offer high potential for a wide range of copper applications. Up to now, however, NIR high-power lasers with several kW are used for welding copper hairpins due to the high melting volume and short process times [13]. In addition, many industrial companies already work with infrared lasers for various welding tasks. Therefore, despite the above factors, these systems are also used for various copper welding applications [95].

Another approach to achieve a better and more reproducible result in the welding process is using two superimposed laser beams. The so-called 2-in-1 fiber consists of an outer part, which has a ring profile, and an inner part, which consists of a single-core fiber. Different laser powers in the core and ring can stabilize the keyhole while welding. This results in less spatter and pores in the process [13]. In addition, rapid and uniform oscillation of the laser spot during forward motion (wobbling) can create stable dynamics in the weld pool [32].

Besides the challenges posed by material properties, external influencing factors can lead to an unstable welding process and spatter formation. These include, for example, surface contamination, gaps between the joining components, misalignment, or an excessively oxidized surface. Correct adjustment of laser welding parameters such as laser power, speed, and focus size is critical in copper welding. Further, the process must not drift due to temperature fluctuations or other reasons. If this still happens, early detection and readjustment is important. The presence of spatter on the component can indicate an unstable situation in the welding process. This correlation allows conclusions about the quality of individual welds and the event of defects based on spatter occurrence [53]. Spatter formation is particularly problematic in electronics manufacturing. On the one hand, material loss in the weld seam can result in unstable connections or lossy connections with increased contact resistance. On the other hand, deposited spatter on the surrounding components or conductive paths can generate short circuits [32, 53].

As mentioned in Section 2.2.4, monitoring is essential for the welding process. Since blowouts and spatter indicate an unstable weld pool and allow conclusions about possible increased contact resistance, unstable connections, and short-circuit risks, this is used as a primary monitoring metric. In addition, an important requirement for a monitoring system is the fast execution time, which is a prerequisite for using a method in large-scale production. Welding an entire engine's contacts takes just slightly more than a minute, and quality monitoring should not slow down the process [78, 150].

Some parts of the chapter are based on results from the article "Camera-Based In-Process Quality Measurement of Hairpin Welding", published by the author in the journal *Applied Science* (Hartung et al. [184]).

4.2 State-of-the-Art

Various sensors can be used for in-process monitoring in laser welding. As mentioned in the introduction, the focus of this chapter is on the detection of spatter and melt blowouts in the process.

Glässel [53] performs spatter detection based on the mass difference before and after welding in an experimental setup to optimize the welding process. However, this procedure is subject to various inaccuracies. For example, spatters that reattach to the workpiece cannot be detected based on the mass difference. The use of contamination masks, similar to those shown in Section 3.5, counteracts this source of error. Glässel [53] also neglects the influence of evaporation of the material due to its low mass loss, which results in inaccuracies in the process. Furthermore, the procedure is not suitable for series production.

Various works show that the acoustic signal can indicate defects in the welding process. For example, Zeng et al. [175] detect a gap by the intensity of low frequencies in the acoustic signal. Lee et al. [92] show that the acoustic emissions correlate with the process parameters, and the features allow inference of possible welding defects such as cracks. Schmidt et al. [143] conclude spatter using the acoustic signal. For the analysis, they use a neural network because they have difficulties in extracting the relevant information with conventional methods due to the noise from the robot and the cross-jet. Generally, acoustic monitoring systems based on airborne sound are often limited by the noisy environment of a factory floor and process noise [143, 147]. Therefore, acoustic solutions often require mechanical contact with the workpiece, which makes them challenging to implement for mass production.

Therefore, often optical monitoring is used. Photodiode sensors are commonly used in the industrial context because of their simple design and low cost. The signals of different wavebands provide information on various components of the welding process. Diodes in the ultraviolet (UV) (200-400 nm) and in the visible spectrum (VIS) (400-700 nm) are used for example, to monitor the plasma of the welding [81, 120, 123], while photodiodes in the infrared spectrum (IR) (1100-1700 nm) are used to monitor the emissions from the melt pool. The near infrared range (NIR) (700-1100 nm) measures the backscattered laser radiation, which provides information about the surface geometry of the area where the laser beam interacts with the material [81, 120, 121, 155]. One disadvantage of the VIS and IR sensors is that they are also susceptible to the vapor plume, which can distort the results. The back reflection signals, in contrast, can be difficult to interpret during deep penetration welding since complex geometry occurs in the molten pool, which does not always allow conclusions about the welding result. Therefore, this parameter is more suitable for heat conduction welding. Another disadvantage of the signals is a complex and abstract interpretation of the results. Imaging evaluations allow a better understanding of the process [81, 120].

This is one of the reasons why many applications use high-speed cameras to monitor the process. An additional bandpass filter is often used to filter the irrelevant information from the image to focus on specific components [50, 163, 177]. Zhang et al. [177] and Gao et al. [50] use high-speed cameras with a frame rate of up to 2 kHz for data acquisition. In addition, they attach a bandpass filter with a transmission band ranging from 350 nm to 650 nm, respectively 350 nm to 750 nm, in front of the camera. The camera is positioned laterally to the process. Gao et al. [50] then propose an approach in which they extract relevant features of the plume and spatter from the image data using image processing technologies such as threshold-based binarization and morphological operations. These extracted features are the inputs to a backpropagation neural network that predicts the weld quality. Volpp [163] also uses a high-speed camera mounted laterally to the weld with an acquisition frequency of 6 kHz for data acquisition. In addition, they use illumination with a pulsed laser of wavelength 808 nm in combination with a notch filter on the camera so that it captures only the reflected light from the illumination laser. A threshold-based binarization algorithm highlights the spatters in the captured images. In addition, they blacken the lower part of the image to eliminate the process light. Using the pre-processed images, they calculate the spatter size and the spatter distribution. They also compare the frames with the previously acquired frame and track the spatter based on size and distribution. This object tracking allows them to measure the total number of spatters.

Nicolosi et al. [118] propose using the Eye-RIS smart camera system [133] based on a focal plane sensor-processor platform with CMOS sensor to capture image data for welding process monitoring. The Eye-RIS system can pre-process images directly on the sensor [118, 119]. Additionally, they use a bandpass filter for image acquisition coaxially to the laser beam, so the sensor acquires a spectral range of 820 nm to 980 nm. Then they detect the spatters using a morphological filter and a defined mask that filters out the plume. Also, Lahdenoja et al. [89] propose using a focal plane processor system for image acquisition. They use the smart camera prototype system KOVA1 from Kovilta, which is attached off-axis to the welding process. The sensor continuously adjusts the integration time per pixel based on the average intensity in the neighborhood of each pixel through an adaptive image acquisition process. This process shortens the illumination time for pixels in very bright regions while lengthening it for low-intensity image parts. As a result, the area of process light is not as over-illuminated as in typical images. In addition to adaptive integration, they use optical filtering with a passband of 700 nm to 950 nm to filter out the light from the welding laser and the plasma. Then they perform a simple segmentation by edge detection, obtaining a black-and-white image in which the spatter pixels are highlighted. Finally, they analyze the spatters' number, size, and speed based on the frame rates.

An event-based camera is also suitable for monitoring spatter occurrence in the process. This event-based sensor captures pixel-brightness changes and outputs a stream of events that encode the brightness changes' time, location, and sign. This procedure has the advantage of a high temporal resolution and a high pixel bandwidth [49, 130]. Because the recording frequency is not limited by defined frames, far more spatters can be recorded in the context of events.

The algorithm proposed in this work is intended to work based on the existing hardware. This has the advantage that the process setup of the laser system for spatter monitoring does not have to be changed. Therefore, a camera with CMOS technology mounted coaxially to the laser beam on the focusing optics is used for the monitoring. In addition, no upstream bandpass filters or external illumination are adapted. Instead of pre-filtering the images, an AI algorithm detects the spatters and separates them from the process light.

In-process quality monitoring depends on fast response times and robust response. Depending on the sampling rate, a large amount of data is recorded during the process. Uploading the data to a cloud system would require a large amount of network bandwidth and constantly consume storage and computing resources of the cloud servers [98]. In addition, a reliable internet connection to the systems is required [115]. This is another hurdle for introducing new, ML-based monitoring systems in industrial manufacturing and creates integration efforts and uncertainties in the usage. Computing the algorithms close to the sensors, so-called edge computing, also offers advantages in terms of data protection and scalability [19, 149]. However, the prerequisite is to use compact and resource-efficient DL models designed for edge devices. The hardware used in the production lines usually fulfills protection classes against contamination, which can lead, for example, to it being fanless and only having passive cooling. As a result, this hardware often has less processing power. In order to avoid having to purchase new hardware for each line, this work uses the existing hardware that also performs the pre-processing monitoring. Therefore, a comparatively small and optimized model architecture must be chosen.

4.3 Experimental Setup and Data Basis

For in-process monitoring, recordings of various welding processes are used. In the configuration used, the camera is attached to the optics coaxially to the laser beam, as described in Section 2.2.4.

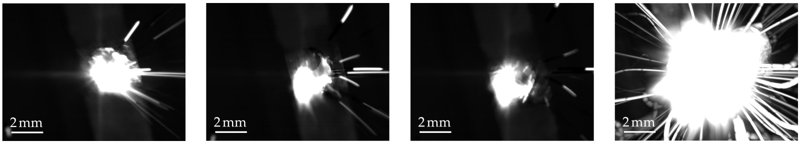
The first series of experiments includes data from welds of hairpin dummies. These dummies consist of copper workpieces cut from a copper sheet with a thickness of 4 mm. The size is derived from a real pair of hairpins with a 4×4 mm welding surface. These dummies are used to reduce external influencing factors such as faulty pre-processing steps, gaps, or an offset of the pins to make the results more stable. For welding, a TruDisk6001 and a PFO33-2 focusing optic are used. The laser power is varied between 2 kW and 6 kW for different welding results. In some cases, 2-in-1 fibers with different ring and core powers are used for low-spatter results. The welding speed is 200 mm/s. Data are acquired with a high-speed camera (IDT Os8 S3) with a maximum frame rate of 10 kHz. The camera is mounted in different orientations to the weld for the experiments. On the one hand, the camera is mounted laterally to the process to obtain a lateral view of the weld and the resulting spatters. On the other hand, the camera is mounted coaxially to the laser beam on the focusing optics as described in the setup in Section 2.2.4. The image resolution is 640×480 pixels.

In a second series of experiments, overlap welds of aluminum sheets with a thickness of 0.5 mm are performed. Two sheets are connected with a 22 mm long line weld. A TruDisk5000 and a PF033-2 are used for the welding, performed with a power of 450 W. For data acquisition, the process uses a VCXG-15M.I, which is attached to the focusing optics coaxially to the laser beam. The camera uses a framerate of ≈ 0.3 kHz and an image resolution of 720×540 pixels.

4.4 Spatter Detection Algorithm

Different methods can be used to identify spatter occurrence based on the camera images. A distinction can be made between evaluating each image of the sequence individually or the entire sequence. Furthermore, the evaluation methods themselves can also vary.

Figure 4.2(a) to (c) show images of a hairpin welding process which are recorded coaxially to the laser beam. Thus the images have a view of the welding area from above. While the process light is in the center of each image, the spatters are visible around it. The three images are successively taken in a sequence with a frame rate of 2 kHz. This sequence



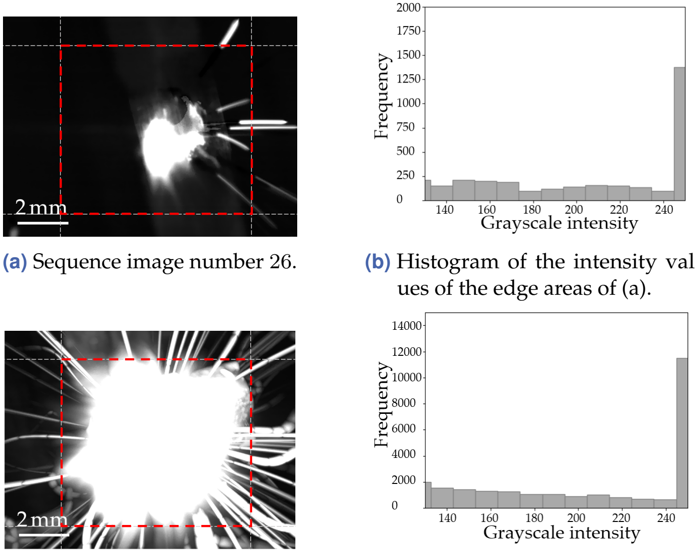
(a) Sequence image number 25. (b) Sequence image number 26. (c) Sequence image number 27. (d) Maximum value image.

Figure 4.2 Coaxially recorded series of images ((a) to (c)) and a maximum value image of the entire series (d). The representation in (d) shows the maximum value of the image series for each pixel.

shows that the spatters are sometimes captured in several consecutive images. For example, the spatter moving out of the top of the image is visible in all three frames. Each frame is taken with an exposure time of approximately 500 µs. Due to the exposure time in combination with the velocity of the spatters, they are mainly visible as lines.

In the welding process, the laser beam is deflected over the surface of the pin using flexible mirrors. Since the camera is mounted coaxially to the laser beam, the mirrors also position the image capture. Due to the different positions and the chromatic aberration, the process light is not always centered in the image for individual exposures. Figure 4.2(d) shows a summary of the entire image sequence ($\approx$ 250 images). For each pixel, the maximum value of the image series is taken. This results in a maximum value representation summarizing the welding process. The process light overlays the inner image area, but the spatters remain visible as lines at the edges.

Pixel Intensity The simplest inspection for spatter occurrence is the analysis of the gray value intensities of the pixels at the edge of the image. This observation works on the maximum value representation and the individual images. The approach is simple and requires only a small amount of computing time. However, it allows only an approximate estimation of the amount of spatter. If no spatter occurs during a welding process, the edge areas show only dark pixels in which, at most, a slight emission of the process lights can be seen. Therefore, the spatter density can be analyzed based on the number of pixels with intensity values above a certain threshold.



(a) Sequence image number 26.

(b) Histogram of the intensity values of the edge areas of (a).

(c) Maximum value image.

(d) Histogram of the intensity values of the edge areas of (c).

Figure 4.3 The red borders distinguish between the edge areas and the process illumination area. Figures (b) and (d) show the corresponding histograms of the intensity values for the edge areas.

Figure 4.3 illustrates the procedure. The red border is defined manually based on the average size of the process light in the maximum value representations of different welding processes. Based on the intensity threshold of 130, a distinction is made between a good process and a spatter-rich one. The spatter distribution could be estimated by observing the intensity values in a specific quadrant. However, it is not possible to make any statements about the spatter's size and exact position or direction. In addition, there is a risk that the plume emitted by the process light will also produce high-intensity values.

Morphological Filter The use of complexer image processing operations allows a more precise data analysis. For example, a separation between process light and spatter can be realized independent of fixed regions using morphological filters analogous to Gao et al. [50] or Nicolosi et al. [118]. First, the images are binarized above a certain threshold before the different areas are identified using the opening algorithm. The operation involves an erosion of the data set $\boldsymbol{x}$ followed by a dilation, both with the same structural element H:

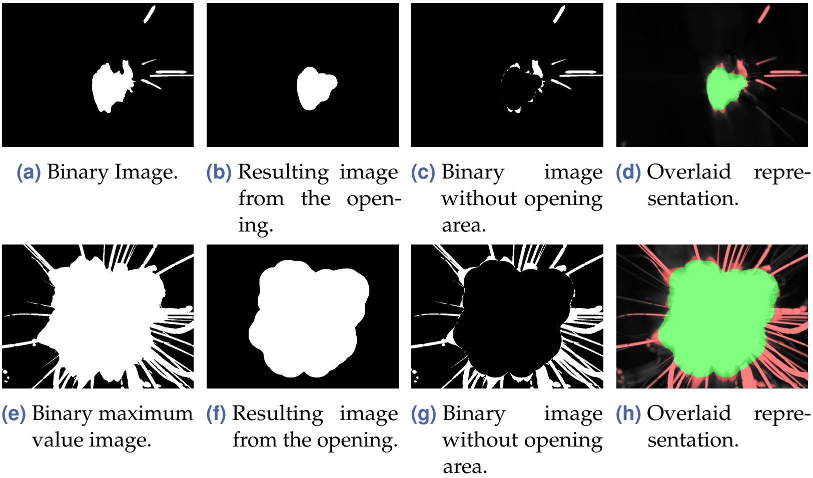
$$\boldsymbol{x} \circ H = (\boldsymbol{x} \ominus H) \oplus H. \tag{4.1}$$

In the first step, the erosion, the opening process eliminates all foreground structures smaller than the defined structural element. Then, dilation smoothes the remaining structures, restoring them to approximately their original size. The opening identifies the process light in the images using the structural element H. Next, the spatters are detected by subtracting the filtered image from the original image. The remaining image elements thus define the spatters.

For the images with a resolution of 480×640 pixels, the structural element is defined as a circle with the diameter $\varnothing = 45$ pixels for the single images and $\varnothing = 90$ pixels for the maximum value representations. The definition of H is based on the average size of the process light and the spatters estimated in the respective images. Because spatters usually represent smaller elements, they can be distinguished from the elements found by the filter. Figure 4.4 illustrates the process steps.

One disadvantage of this approach is that the size of the structure element H must be defined manually. In addition, the algorithm recognizes large spatters as process light, and therefore they are no longer included in the evaluation after substracting the process light. Also, the plume created by the welding process can degrade the result. By not using a band-pass filter, the image may have more spurious elements that complicate the use of the morphological filter. Figure 4.6, Figure 4.7, and Figure 4.8 show more examples of spatter detection using the opening algorithm. In these samples, the difficulties caused by plumes, for example, become visible.

Semantic Segmentation Higher process reliability can be realized by using a CNN to distinguish spatter from the process light. Even if the number of spatters and a rough estimation of their size can be measured



(a) Binary Image.

(b) Resulting image from the opening.

(c) Binary image without opening area.

(d) Overlaid representation.

(e) Binary maximum value image.

(f) Resulting image from the opening.

(g) Binary image without opening area.

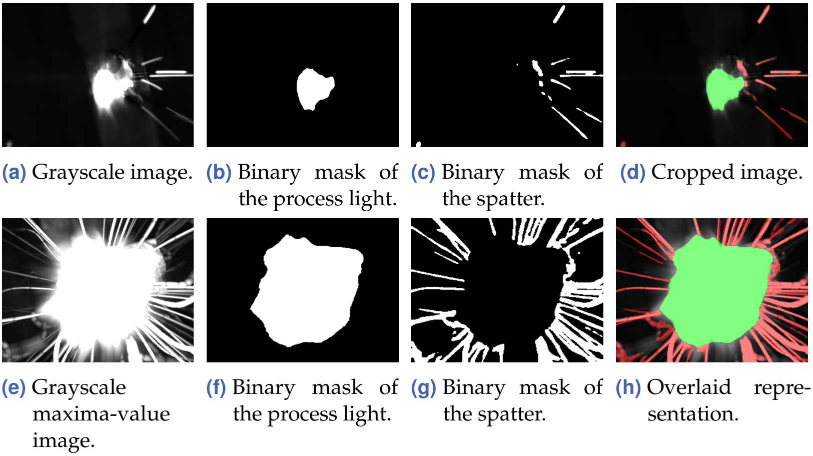
(h) Overlaid representation.

Figure 4.4 Morphological filter. The upper line shows the process for a sequence image and the lower line analogously for a maximum value image. First, a binary image (a) is filtered to identify the process light (b). Then, the spatter is identified by subtraction (c). Figure (d) shows an overlaid representation of the process light (green) and spatter (red) areas on the camera image.

by object detection, using a semantic segmentation algorithm is advantageous. For one thing, object detection works reliably only in the single image of the sequence, whereas semantic segmentation also works on the maximum value image. In addition, the pixel-accurate detection on the single images allows a conclusion about the flight direction and the speed of the spatters. Another advantage is that overfitting is counteracted by the pixel-precise loss function, which reduces the amount of training data needed.

Due to the similar requirements and a similar database as in the component position recognition, reference can be made to the model evaluation from Section 3.4.1. Furthermore, the image context captured by the receptive field is essential in spatter detection. Therefore, using the SDU-Net architecture is also advantageous in this use case.

Considering the fast calculation time required on the hardware at the welding station, the network architecture is further reduced. In this application, the exact position of the spatter in the camera image is less crucial than the detection of the component position. In addition, the images are not very complex, and only little information is lost through downscaling. Therefore, the input dimension is reduced, so the model has to process fewer features. The model input is a grayscale image of the size 128×128 pixels. The smaller resolution and lower complexity of the input images allow to reduce the depth of the network compared to the model presented in Section 3.4.1. Instead of four encoder operations, the architecture uses only three with the number of output filters $n_{\text{out}} = \{16, 32, 64\}$ and the corresponding decoder operations. The total number of the parameters is 39 145. Appendix E shows the detailed structure of the architecture.


(a) Grayscale image. (b) Binary mask of the process light. (c) Binary mask of the spatter. (d) Cropped image.

(e) Grayscale maxima-value image. (f) Binary mask of the process light. (g) Binary mask of the spatter. (h) Overlaid representation.

Figure 4.5 One-hot encoded results of a small SDU-Net with three classes: background, process light (b) and spatter (c). Also, an overlaid representation of the image with process light (green) and spatter (red) is shown in (d). In the upper row, the images are shown for a single frame, and in the lower row, analog for maximum value representation of the sequence.

The training uses the categorical focal loss with $\alpha = 0.25$ and $\gamma = 2$. Furthermore, it uses the ADAM optimizer with the parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for optimization. For better regularisation and the possibility of using a smaller training data set, the training uses data augmentation with rotation, shift, shear, zoom, and flip in different degrees of intensity. In addition, it reduces the learning rate, which starts at $\epsilon = 0.001$, based on the reduction of loss value.

Figure 4.5 shows the one-hot encoded results of the semantic segmentation network. The CNN is trained on a three-class problem, with the classes background, process light, and spatter. Compared to the network architecture with the reduced capacity, predictions with slightly deeper architecture and an input dimension of 256×256 are shown in Appendix F. Due to the reduction of the model capacity, the inference time per frame is less than 3 ms on the Intel i5-7300U using batch prediction.

Comparison Figure 4.6 compares the methods of morphological opening, SDU-Net, and a U-Net variant (defined in Figure 3.5(b)). The figure shows the results as an overlaid representation in the image and evaluates them using the IoU compared to the ground truth. For evaluation, the methods process the maximum value images and the single images of the process sequences. In addition, images from a camera placed laterally to the process were analyzed (Figure 4.6(e)). The results show several advantages of semantic segmentation with a CNN over distinguishing process lights from spatters with the morphological filter. First, the algorithm does not require the specification of a defined size for the process light. Because of the fixed definition of the size of the structural element in the morphological filter, it has to be adjusted individually for the data sets. Furthermore, large spatters, for example, are detected as process light, as in Figure 4.6(b). A second advantage is that the exhaust plume can be better distinguished from process light and spatter. The morphological filtering method requires binarization of the image before the regions are separated in the opening algorithm. Because the plume is usually also bright, this separation is no longer possible. This problem is visible in Figure 4.6(e).

Figure 4.6(c) shows a welded hairpin after the welding process in the cooling phase. The weld surface is shown in blue in the overlaid plot for the SDU-Net. Minor blue artifacts can also be seen in the U-Net result. This example illustrates another superiority of the CNN over the morphological operation. Regardless of the size of the area to be detected, additional classes can be added to the analysis. This division is not possible with the opening algorithm. The example (c)

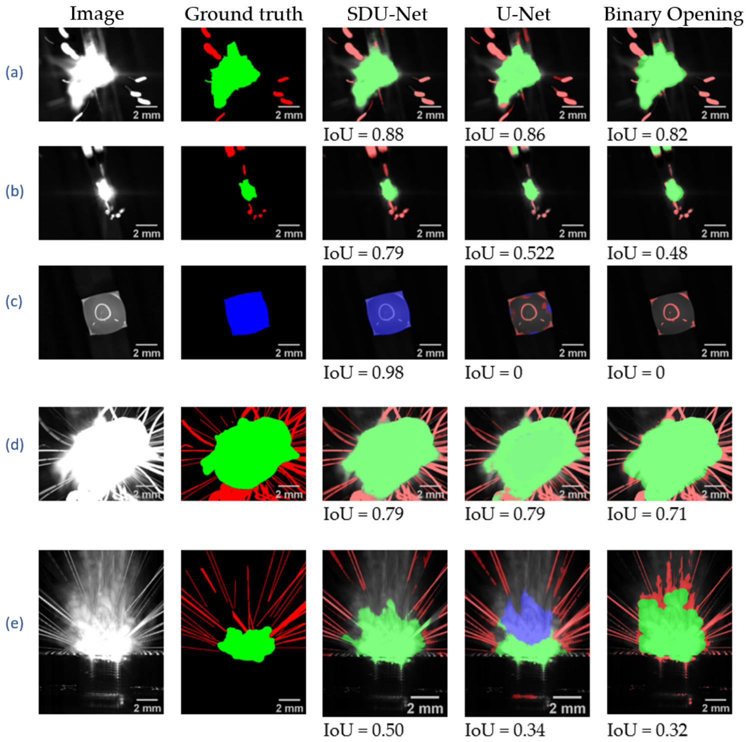
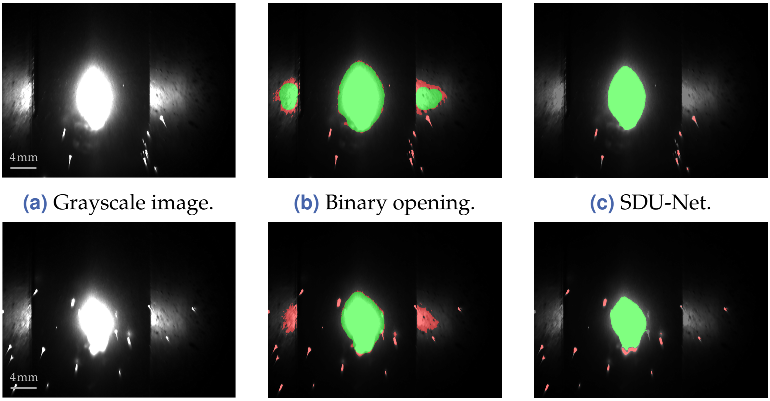


Figure 4.6 Comparison of spatter detection results on different input images. The first column shows the grayscale image, and the second is the hand-labeled ground truth mask. Then the segmentation results with a small SDU-Net, a U-Net variant, and the opening algorithm are demonstrated in an overlaid representation with the process light in green, the spatters in red and the cooling phase in blue. In addition, the average IoU of the foreground classes is given.

extended the three-class problem for CNN to four classes. In addition to the background, the process light, and the spatters, the model introduces another class for the cooling phase. In this phase, the pin is no longer being processed but has not completely solidified. For example, the solidification time provides information about the size of the bonding area. Stavridis et al. [155] shows a correlation between the solidification process and the weld quality. With the possibility of adding additional classes, a quality monitoring system can consider such behavior.

Figure 4.7 shows another comparison of the morphological opening with the prediction of the SDU-Net. Each line shows an image with an overlaid representation of the result of the morphologic opening and the SDU-Net. The images show an overlap weld of two plates, which are clamped together with a fixture. The illumination of the laser light during the process causes reflections on the fixture, which can be seen on the edges of the images. Since there are often interfering elements due to clamping devices in the production environment, it is essential that the algorithm can handle this. In binary opening, the areas are detected as process lights or spatters, depending on the size of the structural element. In the examples in Figure 4.7(b) and 4.7(e), a circular structure with the diameter $\varnothing = 45$ pixels was used as a structural element. The CNN, on the other hand, can separate the areas and detect spatters located in the area of reflection.

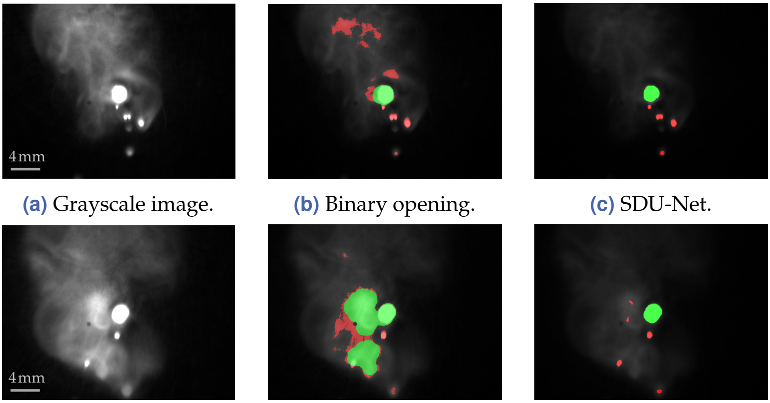


(a) Grayscale image. (b) Binary opening. (c) SDU-Net.

(d) Grayscale image. (e) Binary opening. (f) SDU-Net.

Figure 4.7 Comparison of spatter detection with the binary opening and the SDU-Net. The graphics show an overlaid representation on the camera image, in which the detected spatter is shown in red and the process light in green. In the pictures, bright areas are visible next to the process light due to reflections of the laser light on the fixture.

Another disturbing factor in the camera images is the plume. Since the first step is to investigate the possibilities without modifying the welding station, we do not use a band-pass filter. This way, other processes accessing the camera are not affected. But the smoke plume is more visible in the camera image without a band-pass filter. The plume appears more or less strongly, depending on the welding process. Due to the binarization before the morphological opening, much of the image information is lost. It is not possible to distinguish the plume based on the pixel value since it also appears bright in the image because of the reflections of the laser light. Therefore, the plume is often captured as a spatter or process light. With CNN, separation works better based on image features. Figure 4.8 compares the SDU-Net result and the morphological opening on images where plume is visible.

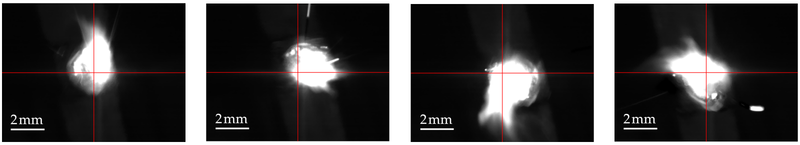


(a) Grayscale image. (b) Binary opening. (c) SDU-Net.

(d) Grayscale image. (e) Binary opening. (f) SDU-Net.

Figure 4.8 Comparison of spatter detection with the binary opening and the SDU-Net. The graphics show an overlaid representation on the camera image, in which the detected spatter is shown in red and the process light in green.

Masking the process light by predefined coordinates is also not recommended for the coaxially recorded data. The lens of the optics is optimized for laser light. Minor imaging errors occur since the camera captures the light in other wavelengths. Chromatic aberration results in the laser light not always being centered in the camera image. Instead, it moves within a certain range. Figure 4.9 shows successive pictures of a hairpin welding process. The laser beam is deflected slightly to melt the entire pin surface. Therefore, in the images, the process light moves away from the center of the camera image. Depending on the size of the working area and the associated deflection of the laser beam, this effect is more or less noticeable. When masking the process light on predefined corridors, the systematic deviation of the position in the camera image must be considered.



(a) Sequence image number 88. **(b)** Sequence image number 90. **(c)** Sequence image number 93. **(d)** Sequence image number 96.

Figure 4.9 Coaxially recorded series of images. There are small time intervals between the individual images. They show the effects of chromatic aberration.

The hand-labeled class assignments are often not accurate for every pixel. Especially in the maximum value representations, where the spatters are shown as lines, the labeling is often inaccurate. This makes it difficult to evaluate the model quality and the resulting prediction based on the IoU of the ground truth. Tabernik et al. [159] show, based on crack detection, that they obtain better and more exact results by labeling a larger area around the cracks than by annotating them exactly. This approach is transferable to spatter detection. The results could be confirmed in experiments but make an accurate semantic segmentation evaluation more complex. In addition, to monitor the laser welding process based on the images, the spatter must be sufficiently recorded in the database.

These aspects motivated, among other things, the evaluation of spatter detection on camera images using semantic segmentation with a CNN directly in the welding process. The results are compared with subsequently visible welding defects, intentionally generated defects during welding, and evaluations based on other sensor data.

Due to the maximum value representation of the frames, much information, such as the number of pixels per individual spatter or the temporal resolution, is lost. By evaluating the spatter percentage of the individual images, the spatter occurrence at different positions of the weld seam can be determined. In addition, the process light superimposes less spatter, which is especially important at lower frame rates. Since longer blind times lie between the exposures, the course of the spatter is not recorded continuously. In the maximum value representation, the spatters are therefore not always visible as a continuous line to the edge of the image. As a result, they can be masked by the process light of another frame, as it is shown in Figure 4.11. Based on the individual images, the fraction of the spatters is related to the area where the spatters can be detected, i. e., the image area not overlaid by the process light. This gives the spatter ratio S_{rating}, where A_{spatter} is the area where spatters are detected and $(A_{\text{image}} - A_{\text{light}})$ is the area in which the spatters could be detected since the process light does not mask it.

$$S_{\text{rating}} = \frac{A_{\text{spatter}}}{A_{\text{image}} - A_{\text{light}}}. \tag{4.2}$$

This relation helps to evaluate the spatter occurrence and the spatter size independent of the laser power and the superimposed radiation intensity [50].

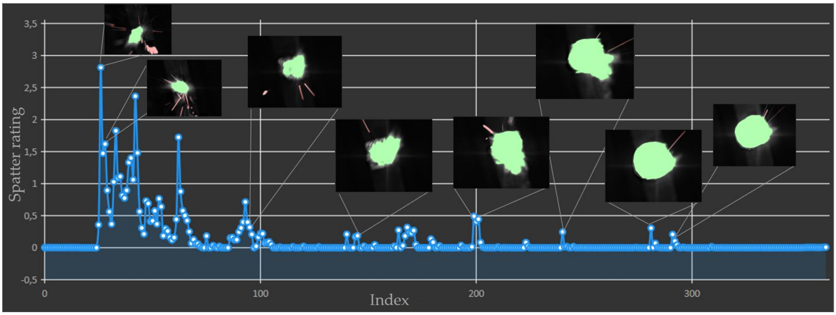


Figure 4.10 Spatter rating of the individual frames for a sequence of a hairpin welding process. Besides the spatter rating, the graphs show some overlaid image representations with the class assignment of the spatter (red) and process light (green) classes on the camera image. The class assignment is based on the prediction of the CNN.

Applying the spatter rating to a process results in a progression over the entire weld, as shown in Figure 4.10. The diagram shows an increased spatter occurrence at the beginning of the process. In contrast, the outliers near the end of the welding indicate smaller and faster spatter. This can be concluded from the thin lines in the camera images. A defined threshold value for the spatter rating can trigger a warning or a welding stop if this value is exceeded. Depending on the process behavior, the threshold value must be adjusted manually. The method also makes it possible to identify the image areas of spatter occurrence. For example, if the spatter is expected to move in a specific direction due to suction, the image analysis can verify this and monitor the other image areas.

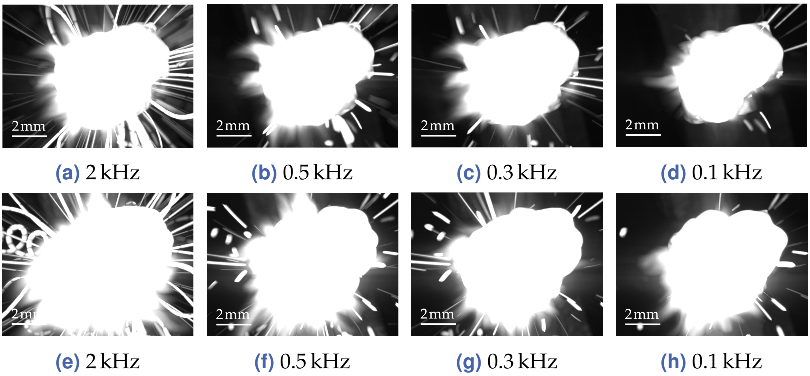
An analogous representation as for the spatter area can also be made for the size of the process light area or the cooling class. Thus, in some cases, burn-ins on the component are characterized by a larger process light area. Monitoring the size over time can therefore enable conclusions about defects. The duration of the cooling phase can also provide information about the size and stability of the weld area. By evaluating the corresponding pixels per image, this can be recorded.

4.5 Image Acquisition Frequency

High-speed cameras are usually more expensive and are not included in the standard setup of the welding station since the high frequencies are not required for pre- or post-process monitoring. Therefore, the following subchapter deals with detecting spatter using lower image acquisition frequencies.

The simulation of maximum value images at different acquisition frequencies is used to estimate whether spatter can be detected and how large the proportion of the detected spatters is. For example, Figure 4.11 shows the maximum value images of one hairpin dummy welding process per line. Through the deflection of the laser beam and the effect of the chromatic aberration, the process light is not always centered in the individual frames. As a result, the process light covers a larger area in the maximum value image. Another consequence is that slow spatters are captured in different images with a slight shift, resulting in circular patterns as in Figure 4.11(e) on the left. The exposure time of the images is about 500 µs.

The maximum value plots show that the process is continuously followed by recording with 2 kHz. This is shown by the fact that the spatters are represented as continuous lines. In the case of fast spatters, this rep-



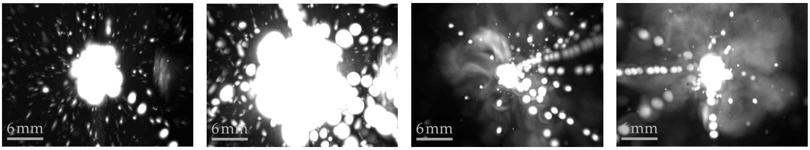
(a) 2 kHz (b) 0.5 kHz (c) 0.3 kHz (d) 0.1 kHz

(e) 2 kHz (f) 0.5 kHz (g) 0.3 kHz (h) 0.1 kHz

Figure 4.11 Maximum value representation of different acquisition frequencies with coaxial image acquisition. The data show two hairpin dummy welding processes (upper and lower row). The single frames were recorded with an exposure time of ≈ 500 µs.

resentation results from the long exposure time within which they cross the image area. In contrast, in the case of slow spatters, they are recorded in several successive images without a gap. The reduction of the acquisition frequency to 0.5 kHz shows that only a few spatters are captured in several images. In addition, the plot shows that not all spatters are recorded anymore. The amount of captured spatters is decreasing by further reducing the recording frequency. The fraction of detected fast spatters, represented as a bright line in the 2 kHz representations in Figure 4.11(a) and 4.11(e), decreases strongly with a lower frame rate and, thus, a higher blind time. However, the plots also show that individual spatters are still detected even at 0.1 kHz. Especially larger spatters are visible at lower frequencies.

With the default setup of a laser welding station from Chapter 2.2.4, the camera has a frame rate of 0.32 kHz . Due to the lower frame rate, the exposure time and the associated reduction of the blind time in the process become increasingly important. Extending the exposure time captures a larger part of the process and, thus, more spatters in the images. With this setting, a good balance must be found between blind time and overexposure due to the process light. The overexposure caused by the process light depends strongly on the laser power and the material properties. Figure 4.12 compares maximum value images of different weldings of aluminum and copper. When increasing the laser power, the exposure time should be shortened so that the process light does not outshine the image.



(a) Al 1 mm, butt weld, 1.1 kW, $t = 250$ µs **(b)** Al 1 mm, overlap weld, 2 kW, $t = 175$ µs **(c)** Cu 1 mm, overlap weld, 4 kW, $t = 50$ µs **(d)** Cu 2 mm, overlap weld, 7.5 kW, $t = 30$ µs

Figure 4.12 Maximum value representations from different processes captured with different exposure times (t). The images are from aluminum (Al) and copper (Cu) welding processes in butt weld joint and overlap welding with different laser powers.

Spatter Size and Velocity In high acquisition frequencies, single spatters are often captured in multiple images. However, one image of the respective spatter is sufficient to determine their existence. Thus, among other things, the speed of the spatter confines whether lower acquisition frequencies could be used for spatter detection. In addition, the size of the spatter is decisive for the resulting weld quality. Large spatters indicate a more significant material loss, usually resulting in an unstable weld seam.

Volpp [163] obtained an average spatter size of maximum 0.8 mm^2 depending on the welding process and laser power in their analysis of an aluminum alloy. However, most spatters are between 0.0001 mm^2 and 0.001 mm^2. With increasing laser power, the average size of the spatter also increases. The averaged spatter velocity is up to 10 m/s at a laser power of 800 W for a Gaussian beam profile, while it is only up to about 2 m/s at a laser power of 1 kW. In this case, only individual spatters exhibit a faster velocity of up to 10 m/s. The investigation of spatter behavior in laser welding of aluminum alloy with different laser sources by Cai and Xiao [17] yielded an average particle velocity of 0.75 m/s and an average spatter size of 0.2 mm^2 for a CO_2 laser. In contrast, the results for a fiber laser are 3 m/s and 0.13 mm^2. Volpp [163] and Cai and Xiao [17] work with images taken from the side of the process and thus determine the vertical velocity of the spatters in their calculation. A general conclusion of the analyses is that small spatters mostly have higher velocities than larger ones [17, 70, 163].

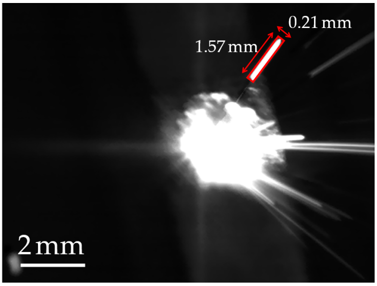


Figure 4.13 Values for estimating the spatter velocity based on the spatter size and length of the spatter tail. This calculation is limited to the horizontal velocity and to the assumption that the spatters are round.

The size of the spatters can be estimated based on the single-frame exposures. Also, the approximate horizontal speed of the motion flow can be calculated under the assumption that a spatter is round. The width of a spatter, in combination with the length of the spatter tails, gives the distance a spatter moves during the image acquisition. Since the exposure time per frame is known, the speed in the image plane can be calculated.

The scatterplot in Figure 4.14 shows the analysis of the spatter sizes and the horizontal velocities of various welding processes in cumulative representation. It shows results from images recorded with an acquisition frequency of 2 kHz and laser powers from 2 kW up to 6 kW. The velocity is calculated by

$$v_h = \frac{d}{t}, \tag{4.3}$$

where t is the exposure time, and d is the traveled distance, represented by the length of the spatter tail minus the spatter width. This results in $d = l - w$, where l is the length and w is the width of the spatter. Due to the view of the process from above, only the horizontal speed is considered in this calculation. However, this is the relevant speed because it determines whether or not a spatter is captured in the recordings. In the example of Figure 4.13, this gives $v_h = \frac{1.57\text{ mm}-0.21\text{ mm}}{0.498\text{ ms}} = 2.73\,\text{m/s}$.

The spatters that leave the camera's recording area are only partially captured. These are considered with a shortened tail length and distort the analysis slightly. Therefore, it can be assumed that the velocities are sometimes higher. In addition, slow spatters will be seen in several frames and detected more often by the single-frame analysis. The evaluation in Figure 4.14 uses object tracking, which compares the noticed spatters per frame with the spatters of the previous frame. Therefore, the algorithm evaluates the objects based on position, size, and flight direction, and only newly appearing spatters are considered. In the evaluation, it can be deduced that slow spatters occur more often, with a speed of less than 2 ms. In addition, the statement that large spatters are generally slower than small ones is also confirmed.

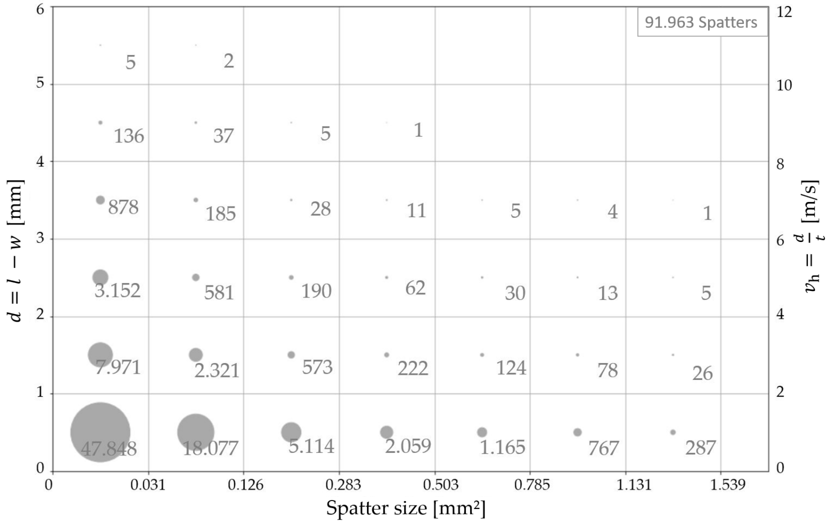


Figure 4.14 Cumulative scatterplot for size and velocity of spatters based on 136 weldings.

Depending on the imaging ratio, the camera image captures a different area. For example, observing a setup with a Baumer VCXG-15M.I, a TRUMPF PFO-33-2 focusing optic with a focal length of 255 mm and a magnification of $\beta = 0.4$, the camera captures an area of 31.7×23.7 mm. If the process is centered, there would be a distance of at least 11.58 mm to detect a spatter, assuming the process light does not mask it. With a detection frequency of 320 Hz and an exposure time of 250 µs, spatters can be expected to be detected at a horizontal speed of less than 4.12 m/s. The calculation considers the blind time between two exposures of 3.12 ms − 0.25 ms = 2.87 ms. Within this time, the image area of 11.58 mm must be crossed so that the spatter is not visible in one frame. This results in $\frac{11.58\text{ mm}}{2.87\text{ ms}} = 4.12$ m/s. The time will be slightly longer if the spatters do not take the shortest path out of the detection range. However, faster spatters can cross the area between two exposures without being visible in one image. Combining this finding with the results from Figure 4.14 enables the conclusion that 94% of the spatters would be captured at a frame rate of 320 Hz on at least one image. This is the fraction of spatters with a velocity of less than 4 m/s. If only the larger spatters above 0.78 mm^2 are considered, this makes 98%.

Comparision of Recording Frequencies To approximate the result with the frequency of the Baumer VCXG-15M.I, lower acquisition frequencies are simulated based on 2 kHz recordings. This way, the spatter evaluation can be compared based on different frequencies. Since frequency reduction can only be realized by omitting entire frames, the resulting comparison frequency is 286 Hz, approximating a realistic maximum frequency of the Baumer VCXG-15M.I camera. The comparison of the spatter rating with a recording frequency of 2 kHz (Figure 4.15 upper line) and 286 Hz (Figure 4.15 lower line) shows that powerful ejections are visible even at the low frequency.

While the high-frequency frames capture many spatters several times, they are contained in only one image at the lower frequency. Also, not all spatters are captured. Especially in the case of more significant defects, this is not very important because there are many spatters in these situations. Such a situation is visible, for example, in Figure 4.15 at the beginning of the process. An apparent spatter ratio is still visible even if not all spatters are detected. This is different for small and fast spatters that occur sporadically in the process. These are often not detected. However, large spatters, which usually have a greater influence on the

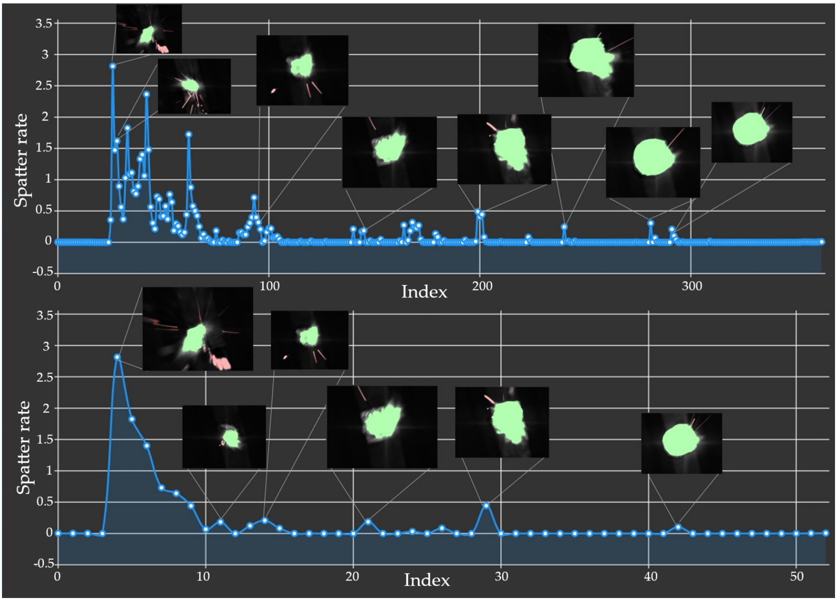


Figure 4.15 Spatter rating of a hairpin welding. The upper row shows the spatter rating for a recording frequency of 2 kHz while the lower row simulates a recording frequency of 286 Hz for the same process. The figure shows, for better illustration, some camera images with a superimposed representation of the semantic segmentation result (spatter in red, process light in green).

welding result, can be detected. Usually, these ejections move slowly, making them detectable in at least one image. They are particularly problematic in electronics manufacturing because the larger material deposits create connections with increased resistance, and the deposited ejections can cause short circuits in the component.

Further comparison of the spatter rating for welding processes of hairpin dummies are shown in Appendix G. The recordings show that more significant ejections and longer passages with recurring spatters are recognized, while smaller spatters remain partially undetected.

4.6 Evaluation and Discussion

This subchapter shows that process monitoring can be realized during laser welding with the standard setup with a camera mounted coaxially to the laser beam. It uses a camera with a maximal frame rate of 320 Hz to assess the stability in the welding process due to the occurrence of spatter. For this purpose, no external lighting or filters to limit the captured camera image to the relevant areas are used. By not modifying the laser welding station setup, the same design as for component position detection for pre-process monitoring can be used. The CMOS sensor in the Baumer VCXG-15M.I causes due to its relative response that preliminary information about the plasma (in the VIS range) is captured in the image data (Figure 4.16). This range is more comprehensive than, for example, in the work of Gao et al. [50] or Zhang et al. [177], who restrict to the lower range of 350 nm to 650 nm and 750 nm, respectively. Another example is the work of Nicolosi et al. [119] or Lahdenoja et al. [89], who use the upper range of $\approx$ 700 nm to $\approx$ 980 nm.

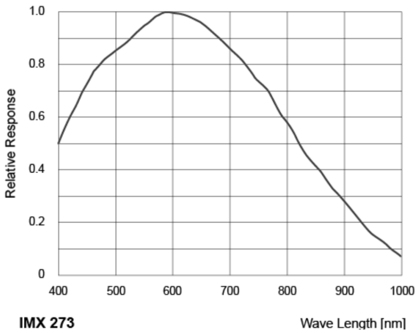


Figure 4.16 Relative response of the Sony IMX273 sensor of the Baumer VCXG-15M.I camera [9].

Using a CNN for segmentation, the spatter can be separated from the process light and the exhaust plume, even if there are more interfering elements in the image. The comparison of the morphological opening with the semantic segmentation using CNN shows the advantages. This approach realizes a more accurate and robust spatter detection.

In addition, the chapter shows investigations in terms of the recording hardware and compares different recording frequencies. It shows that lower frequencies result in less spatter being recorded in the image sequences. However, more significant weld defects, in particular, lead to an increased spatter volume, which can also be detected with lower recording frequencies. Also, large spatters, which usually have lower velocities, are captured with a high probability in the image sequences.

Comparison with Diode Signal The evaluation compares the spatter rating based on image data with the results of different diode signals. Signals in range $< 600\,\mathrm{nm}$ (plasma detector), 1064 nm (back reflection detector) and 1100-1800 nm (temperature detector) are used. Similar patterns can be detected as an overall result compared to the defined spatter rating.

The Precitec LWM system [125] is used to record the signals. The sampling rate of the diode signal is about 50 kHz, which is 158 times faster than the camera frequency. Figure 4.17 shows the results of an overlap weld of two copper sheets with a thickness of 2 mm. The exposure time of the captured images is 30 µs.

Figure 4.17 shows a weld with two larger defects. These defects are visible in a camera image of the weld taken after the process. In addition, the figure shows the spatter rating defined in this chapter and the three diode signals related to the welding result. The diagrams show that especially the more significant ejections are visible in all signals. All three photodiode signals, as well as the camera-based spatter rating, increase strongly. Compared to the other signals, the value of the back reflection (1064 nm) decreases to the normal level later after the defects. Moreover, the signal shows a sharp increase at the beginning of the weld when the laser beam is coupled. The figure also shows some captured camera images with a superimposed representation of the semantic segmentation result. These images also show strong ejections at the corresponding locations.

Figure 4.18 shows the result of a weld that was realized using the wobble technique, resulting in a good weld. It is essential to note in the comparison the scaling of the spatter rating. This shows that with a good weld, the metric's range is no longer in the interval 0-25, but

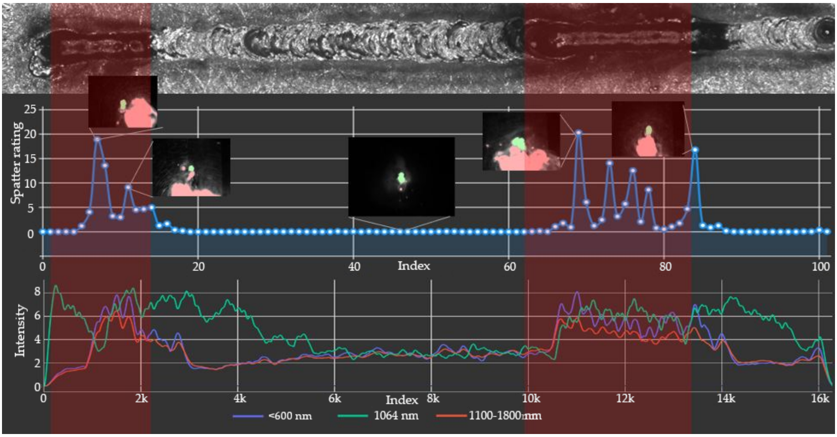


Figure 4.17 Comparison of diode signals and camera-based spatter rating of a defective overlap welding. For better illustration, the figure shows additional camera images with a superimposed representation of the semantic segmentation result (spatter in red, process light in green).

only up to a maximum of 0.07. In the image-based spatter rating, these small spatters can be partially captured. However, as mentioned before, it is assumed that not all spatters are included due to the blind times between frames. With a framerate of 320 Hz and an exposure time of 30 µs, this results in a time of about 3.1 ms between the frame captures. Assuming the minimum image area of 11.58 mm again would mean that spatters are no longer reliably detected with a horizontal speed greater than 3.8 m/s. Thus, the metric does not allow a reliable conclusion on the coverage of all spatters. Therefore, the threshold for detecting a defective spot should be higher than 0.07, as the detected small spatters do not allow meaningful conclusions. These small ejections are not perceived at the signal of the photodiodes because the signal constantly fluctuates slightly.

Figure 4.19 shows a weld without wobbling and without specially generated defect cases. However, the subsequently taken camera image shows that a more irregular weld is produced. In addition, the weld shows defective areas at the beginning due to increased ejections. The scaling of the spatter rate covers the range from 0 to 1.3. Comparing the

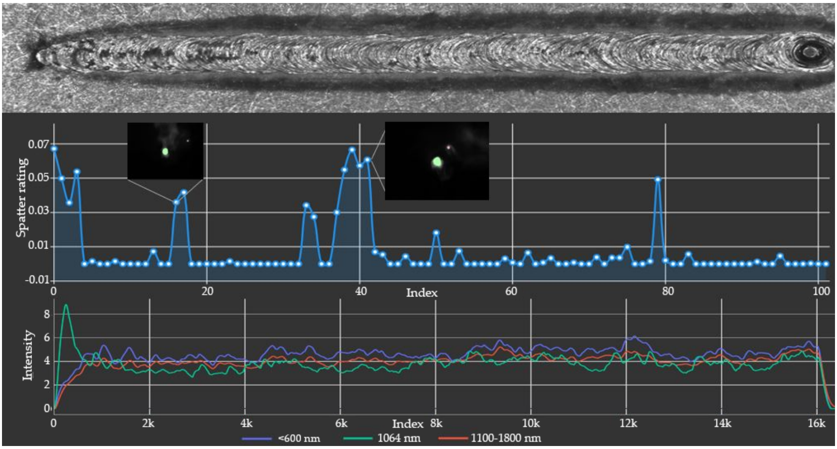


Figure 4.18 Comparison of diode signals and camera-based spatter rating of a good quality overlap welding.

metric with the photodiode signals shows that the smaller peaks of the spatter rate are not captured in the diode signals. Furthermore, the first peak in the process is seen only in the back reflection (1064 nm), while the temperature (1100 nm-1800 nm) and the plasma (< 600 nm) do not indicate this defect. However, the second more substantial peak in the spatter rating coincides with a small peak in the plasma signal.

The comparison with Figure 4.17 shows that the larger defects produce more distinct deflections in all signals. While the spatter rate there mostly has a value above 5, the intensity value of the plasma and the back reflection is greater than 6. In the good weld in Figure 4.18, the spatter rate does not exceed a value of 0.7, and the diode signals remain at a lower level. The intensity of the plasma (< 600 nm) does not exceed a value of 6, the intensity of the temperature (1100 nm-1800 nm) and the back reflection (1064 nm) remain in the whole course even under ≈ 5.

Comparing the temperature and plasma diode signals with the in-process camera images is consistent with the finding of Kaplan et al. [81] that the size and intensity of the plume strongly influence these signals. Increased occurrence of the plume often correlates with the formation of weld spatter. However, this correlation is not always given. For example,

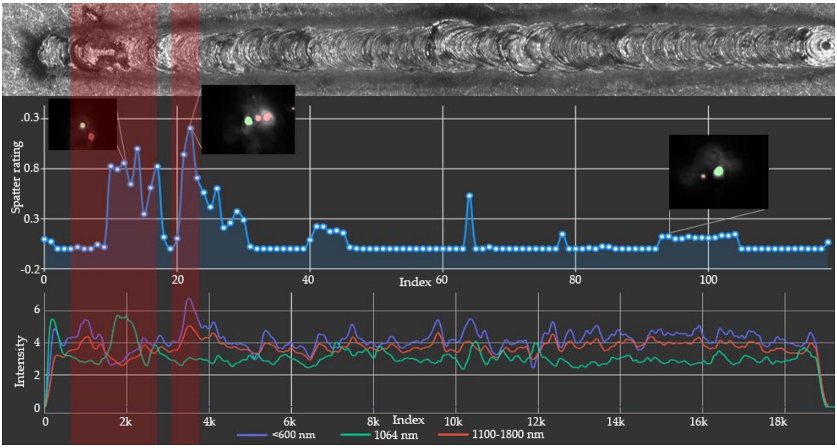


Figure 4.19 Comparison of diode signals and camera-based spatter rating of an overlap welding with smaller defects.

no plume can be seen when looking at the image data in the first red highlighted area in Figure 4.19. However, the pixels around the spatters show a plume in the second red highlighted region. In this case, the diode signals increase. Norman et al. [120] also found a correlation between the plasma sensor, the temperature sensor, and the image intensity. However, in contrast to the sensor signals, the image-based approach with CNN-based semantic segmentation offers the advantage that the algorithm can detect spatters independently of the smoke plume and the general pixel intensity in the camera image. Therefore, for example, the threshold value distinguishing between a good and a bad weld does not have to be adjusted precisely to the image intensity. However, photodiode monitoring, as well as camera-based monitoring, can result in spatter not being detected. Therefore, it also depends on how critical the process is and whether all spatter has to be recognized. For example, it is sufficient for some processes to detect larger ejections, while others require every ejection to be detected. In addition, the thresholds for a good weld must be set manually with both systems.

Volpp [163] divide the potential spatter into two categories. The first is eruptive spatter ejections, which occur in a stable process with large amplitudes and low frequencies in the keyhole. This results in an eruption of micro-ejections from the escaping vapor. The second category of spatters is these detachments that arise directly from the wall of the keyhole. These spatters result from high frequencies at small amplitudes and have a larger size and low velocity. In contrast to the eruptive spatter ejections, they indicate an unstable process, which should be detected in the in-process monitoring.

The evaluations have shown that larger and, thus usually, slower spatters can be detected in most cases with a sampling rate of 320 Hz. Additionally, defects leading to a high spatter volume are recognized. The approach describes a possibility of monitoring the welding process without additional hardware. Thereby the claim is not to realize a 100% monitoring of all spatters. Instead, the aim is to detect process drift, faulty pre-processing steps, material defects, or other significant deviations from the standard process that indicate an unstable process.

5 Post-Process Monitoring

5.1 Introduction

Chapter 3 and 4 explain monitoring methods that are used before and during the welding process. The methods aim to ensure a good welding result and thus relate to the weld quality after the process. However, since neither pre-process nor in-process monitoring can detect all defects, downstream monitoring is also essential. As the previous chapter shows, some defects are visible during and after welding. During the process, this can be determined using spatters as an indicator. Afterward, the defects are visible through surface characteristics and the geometry of the solidified weld. This is shown, for example, in the overlap weld in Figure 4.17. In this case, the in-process images show spatter, and the weld defects are also visible in the post-process image. Other defect cases, however, are only visible after the process. In hairpin welding, for example, this includes a defect from Figure 2.16 where the pin pair was welded with too little power. While there are no visible errors in the process, it results in a connection that is too small and unstable. In addition, the copper material tends to form pores during laser welding. This defect is attributed, among other things, to the low surface tension of molten copper. The pores have a negative effect on the electrical and mechanical properties similar to a seam connection that is too small [77]. They are often subsequently visible on the seam surface or result in a raised seam containing a hollow space.

Different sensor data can be used to evaluate the quality of the weld seam after the welding process [51, 104, 162, 178]. Various works show that the analysis of 3D data provides higher accuracy than the analysis of 2D camera images [28, 153, 154, 162]. The disadvantages of using 3D data are higher hardware costs, higher system complexity, and long process times. As in the previous chapters, this work focuses on influencing the setup and welding process as little as possible. Therefore, methods are presented to evaluate the quality accurately based on a camera image taken coaxially to the laser beam. Among other approaches, this work presents a method that calculates the height map from a camera image instead of capturing it with a 3D sensor. This method allows using the height data for the quality assessment without the disadvantages mentioned above.

The chapter is structured as follows: First, it compares the state-of-the-art quality assessment of hairpins after welding. In addition, it investigates and compares different methods for 3D reconstruction. After presenting the data basis and the experimental setup, the chapter presents the developed algorithms. These include approaches from machine learning, which use semantic segmentation and 3D reconstruction to extract quality relevant features. The final part of the chapter compares the quality assessment methods before summarizing the results and drawing a conclusion on the findings.

The method for subsequent quality assessment of welds using semantic segmentation has been published in "Camera-based spatter detection in laser welding with a deep learning approach" (Hartung et al. [183]) and was presented by the author at the conference *Forum Bildverarbeitung 2020* in Karlsruhe. Different 3D reconstruction methods are analyzed in the article "Analysis of AI-based single view 3D reconstruction methods for an industrial application", which was published by the author in the journal *Sensors* (Hartung et al. [185]). The comparison of different quality assurance methods was presented by the author at the conference *Forum Bildverarbeitung 2022* and is published in the proceedings in the article "Quality control of laser welds based on the weld surface and the weld profile" (Hartung et al. [181]). A further comparison was published by the author in *tm - Technisches Messen* in the article "Machine learning based geometry reconstruction for quality control of laser welding processes" (Hartung et al. [182]).

5.2 State-of-the-Art

There are a variety of systems for quality monitoring and control in laser welding. The use of machine learning methods is investigated and evaluated by Mayr et al. ([103], [104]), and Weigelt et al. [168]. They conclude that, unlike the development of many other ML applications, the amount of data samples in the industrial environment, especially in research, is limited. They also suggest that it is essential that the computation time does not extend the production time [168]. This requires the algorithms to be computed quickly. These two requirements both represent challenges that must be considered during algorithm development.

Mayr et al. [104] use images from three perspectives, front, top, and back, to evaluate the seam quality of hairpins. The different perspectives allow for obtaining more information about the seam connection. However, integration into a production line is more complex because attaching cameras in a side view to the welding station is often difficult. They use a neural network to obtain the weld quality. The network's resulting accuracy ranges from 61% to 92% [104]. Vater et al. [162] analyze and compare different CNN architectures to perform post-process quality control of hairpins. Besides 2D grayscale images, they use 3D scans as input to the CNN. Based on the 3D scans, the classification accuracy is higher than using the intensity images. This result supports the assumption that the height values contain relevant information for quality assessment. Ye et al. [173] and Stadter et al. [153] also use a height profile to determine weld quality in laser welding. Especially in hairpin welding, the height difference between the pair of hairpins before and after welding provides information about the volume of the molten material. This volume, together with the other measured parameters of the weld seam surface profile, shows crucial information about the welding quality of the hairpins [28, 60]. Will et al. [170] also deal with the evaluation of welds of hairpins in their work. They discuss the correlation between the electrical resistance of the weld and the offset of the copper pins. Their experiments conclude that the weld joint's height profile allows the determination of poor welds.

Due to the cost, higher system complexity, and acquisition time, it is advantageous to calculate the height profile using a method of 3D reconstruction instead of measuring it with a height scanner. Different methods can be used to calculate the height values. For example, Lei et al. [93] use shape from shading (SFS) to perform a 3D reconstruction of a weld seam. Based on the curvature features, the weld quality is evaluated. Especially for the classification of complex welds with complicated structures and features, the information content of the curvature feature is limited. Due to this, the method reaches its limits and cannot be used for more complex tasks. The SFS algorithm reconstructs a shape based on shading variations, assuming a single-point light source and Lambertian surface reflectance. Here, the brightness of an image pixel depends on the direction of the light source and the surface normal. Due to the height of the hairpin and the dome shape of the weld, a reconstruction from a single image with SFS is not possible. The incidence of light can only be realized on one side. The other side is accordingly in shadow [66]. This means that several images would be necessary for a complete height calculation. Rodríguez-Gonzálvez et al. [132] calculate a 3D reconstruction from several images taken during the data acquisition phase with different relative positions between the camera and the weld. This way, they calculate a 3D model of the weld based on the different relative positions. Using this model, they perform a quality assessment.

In addition, DL-based methods for 3D reconstruction are showing promising results in various research areas [20, 97, 141, 148, 151]. While classical methods deal with shape and image properties such as reflection, albedo, or light distributions, DL-based methods use complex network architectures to learn the correlations between 2D and 3D data. However, many approaches are challenging to integrate into existing industrial processes because they require multiple cameras, further sensor technologies, or new lighting equipment. Processes based on only one camera are necessary for easy integration in the laser welding station. Zhang et al. [176] propose reconstructing 3D surfaces of human faces from corresponding 2D images using a stacked autoencoder (SAE). Low-dimensional features of the 2D and 3D images are learned separately using autoencoders and connected by additional neural network layers. This results in a deep neural network that has a 2D image as input and 3D height information as output. Baby et al. [7] use a similar approach by implementing a CGAN [110] to reconstruct a depth map from a single image. The advantage of the generative adversarial network (GAN) [57] is that it is trained to produce realistic-looking images. The CGAN considers additional information besides the noise vector that is given as input to the network during image generation. This makes it suitable for creating height maps based on an intensity image. Also, Arslan and Seke [5] use a GAN structure to reconstruct 3D informations from 2D images of faces [4, 5]. They extend the GAN approach using Wasserstein distance [3] to achieve better predictions.

This work presents and compares different camera-based methods for post-process quality assessment. One approach considers the seam surface and geometry to conclude the quality. Another approach applies DL-based methods for 3D reconstruction to calculate the height maps based on an image. Afterward, the calculated height information is used by an algorithm to determine the weld quality.

5.3 Experimental Setup and Data Basis

The quality assessment is performed on a data set $\mathbb{X}$ with $n = 953$ samples of laser-welded pairs of copper pins. Different welding results are recorded to obtain a representative data set that includes error cases. The setup presented in Section 2.2.4 is used for data acquisition. Thus, the sensors are mounted on-axis on a programmable focusing optic, i. e., the component is observed via the light path of the optics coaxially to the laser beam. This has the advantage that no external installation is required at the welding station, which restricts the welding process as little as possible.

Height Data An OCT scanner from Lessmüller Lasertechnik is attached to the first sensor output. The sensor uses FD-OCT to capture the relative height information of the weld. It performs 1000 line scans analogous to those shown in Figure 5.1 to capture the entire weld area. The x-direction of the image represents the position of the scan point of the line scan, and the y-direction represents the actual height value. While the first row shows three line scans of a good weld, the bottom row shows the scans of a misaligned pin pair. These line scans show that the front left corner of the weld is raised, which can be attributed to different heights of the pins before welding. The individual line scans are combined to create an overall height map of the part, as shown in Figure 5.2(a).

The lateral resolution of the height map is 1000×1000 pixels, with a step size of 7.5 µm. The height information is recorded in increments

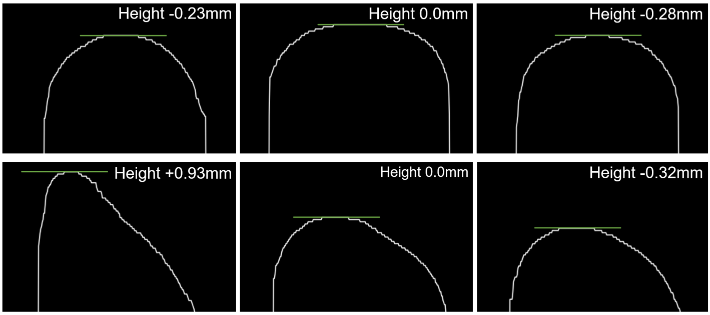
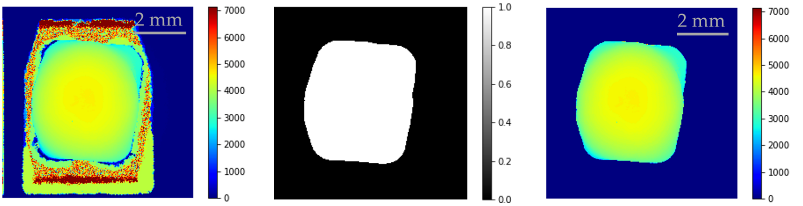


Figure 5.1 OCT line scans at different positions (left side, center, right side). The y-axis shows the height value, while the x-axis represents the scan position. Top row - similar height values indicate a good seam. Bottom row - the different heights indicate the fault case (misaligned pin pair).

of 11.7 µm, with the sensor covering a measurement range of approximately 12 mm. For further processing, the height values from the OCT scan are converted to a grayscale image, where each pixel of the image represents a scan point from the height map. The height informations are scaled to 256 gray values, resulting in increments of 46.8 µm. This loss of accuracy in the elevation data does not affect any downstream quality assessment based on the height data. The height difference of the hairpins is in the millimeter range, and the error cases show more significant height deviations than 46.8 µm. Such minor deviations are not relevant, so scaling to 256 values can be performed. Among others, Vater et al. [162] show that a meaningful quality assessment can be made with this simplification.

Since OCT is susceptible to artifacts and noise, undesirable interference occurs in the 3D images, which is why pre-processing of the data is necessary. For example, the opening in the stator surrounding the hairpin is outside the measurement range of the OCT scanner. Some component areas are still close enough to reflect a signal but are recorded with false height values based on the detected signal frequency. Other component areas are too far from the scanner, so the sensor only provides a noise signal. The hairpin surface is the focus of the parameterization of the reference arm, which means that the height values are correctly detected there. The other areas are cut out of the recording in a pre-processing



(a) Height map (in µm) of a welded pair of hairpins.

(b) Foreground mask weld surface.

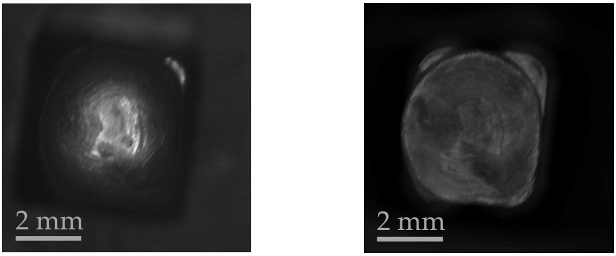
(c) Height map (a) windowed with the foreground mask (b).

Figure 5.2 The hairpin surface is detected by semantic segmentation in the height map (a). Subsequently, the height map is multiplied with the foreground mask (b) of the model prediction, resulting in the pre-processed height map (c).

step. For this purpose, a CNN has prevailed over edge-based algorithms, such as high-pass filtering. The high-pass filter can be used to detect the areas with a substantial pixel difference, i.e., the pin edges. Downstream steps, such as binarization of the high-pass signal and contiguous region detection, can be used to extract the pin region. This algorithm also works well in many cases, but sometimes, it removes incorrect areas, such as unwelded pin areas at the edge of the weld. The use case is similar to detecting the component positions in a camera image. The image shows height values instead of intensity values but still contains interfering elements and often unclear contour boundaries. Analogous to the procedure in Chapter 3, a semantic segmentation using an SDU-Net architecture performs detection of the pin surface. The mask of the foreground class cuts out the height map to the relevant area, as shown in Figure 5.2. Another pre-processing step eliminates artifacts on the hairpin surface by outlier detection. The artifacts are caused partly by lens contamination but can also be caused by measurement errors of the OCT. An artifact is defined by an outlier of a few pixel values that deviate from their local environment. It can be physically excluded that the welding process, such as spatters, causes such artifacts. A distance-based algorithm detects outliers. It compares the pixel values with the respective values of the neighboring pixels and replaces them with the neighborhood average if the deviation is too large.

Camera At the second sensor output, a camera records the intensity images of the welded hairpins. For this purpose, a Baumer VCXG-15M.I industrial camera based on CMOS technology is used, as shown in the setup in Section 2.2.4. The images have a resolution of 720×540 pixels, with a pixel pitch in x- and y-directions corresponding to 18 µm each. The very reflective surface of the copper material causes reflections and shading in the image. These can be reduced, for example, by using a dome or lateral illumination. With dome lighting, the light is transmitted into a dome-shaped reflector and diffusely scattered onto the object. The reflector is mounted at a similar height as the component. As a result, the component is illuminated evenly and without shadows. This makes the surface structures and properties of the seam more visible in the image. However, this type of lighting interferes with the welding process because the lighting surrounds the part. This means the illumination must be repositioned each time new joining partners are welded. In addition, the lighting was often contaminated by material deposits during welding if it was not removed for the welding process. The deposits also change the illumination situation for the data recorded subsequently. Furthermore, more contamination also means more frequent cleaning. The ring light from Section 2.2.4 is attached to the optics and thus has a certain distance to the component and the welding process. Not only is it less contaminated, but it also does not interfere with the change of joining partners since it is not in the processing field. To create a realistic situation for quality assessment based on the system's camera data, illumination with an LED ring light is used. The light is attached to the optics and illuminates the component from above. Figure 5.3 shows an example of the captured camera images with ring illumination compared to images of the same component with dome illumination. Appendix H shows another comparison between a ring light attached to the optic, dome illumination, and side illumination.

Mapping The different sensor data must match exactly to apply the 3D reconstruction algorithms, which will be presented in Section 5.4.3. These algorithms require identical image pairs in translation, rotation, and scaling. Since the sensors are not calibrated to each other during data acquisition, the data must be adjusted afterward.



(a) Ring illumination. **(b)** Dome illumination.

Figure 5.3 Camera image of a welded hairpin pair taken with a ring illuminator attached to the optics (a) and with dome illumination (b).

As described in the experimental setup, the acquired data have different sizes and scales. The performed mapping uses the image area corresponding to the OCT scan. Even if a larger area is visible in the camera image, the height information is only available for the area of the OCT scan. The corresponding area in the camera images is defined manually, and the images are cropped to this size.

Since the resolution of the intensity images is lower, this is adopted for both data sets. Therefore, the resolution of the height scans is reduced accordingly. This results in a loss of accuracy but is acceptable for the use case. The component's smooth surface structure does not show drastic changes within 7.5 µm. Therefore, scanning at a distance of 18 µm is sufficient.

The relevant criterion for mapping is the shape of the weld seam. This can be seen in both the camera image and the height profile. The weld area is detected on the intensity image and the height map due to interfering image elements using semantic segmentation. This results in a one-hot encoded output of the foreground class, which is converted to a binary mask that highlights the relevant surface. The centroid of the masks' area is then computed, and the masks are centered on this point. Finally, by exploiting the correlation of the polar coordinate images, the rotational offset of the images is calculated. The exact mapping algorithm is described in detail in Appendix I. In a productive setup, it is not necessary to map the data manually since a uniform sensor calibration is available. This pre-processing step is due to the experimental setup.

Data Basis Different welding results are recorded to obtain a representative data set that includes error cases. The causes of the errors and the formation of different error cases are presented in Section 2.3 and Figure 2.16. To reflect the situation in the industry with low data availability, 10% of the data are used for algorithm development. The other 90% are used for testing and evaluation. It results in a data set $\mathbb{X}^{\text{train}}$ with $n = 95$ and $\mathbb{X}^{\text{test}}$ with $n = 858$. The selection of the small data set is motivated by very little data available, especially from error cases. Developing a quality assurance algorithm should not result in many faulty materials (scrap) and a high time requirement for collecting the database.

5.4 Algorithm

Various algorithms for the weld inspection are analyzed to compare the quality assessment results. The height data acquired by OCT, intensity images acquired by a monochrome camera, and reconstructed height data are used to create feature vectors as input for a rule-based quality evaluation.

5.4.1 Height Profil

The OCT sensor measures the relative height differences within the weld seam. Good welding of a pin pair results in a round welding bead, which has its maximum in the center [28, 60, 94]. Therefore, the line scans should have a structure like in Figure 5.1(a) over the entire weld bead. Figure 5.1(b) shows the images at the same positions of a weld with misaligned pins for comparison. Analog Lessmueller Lasertechnik GmbH [94] and Baader et al. [6], the quality assessment algorithm compares multiple line scans with each other. It considers various criteria.

Analogous to Lessmueller Lasertechnik GmbH [94], the algorithm considers the difference between the maximum height values of the individual line scans and the pin center's height. This comparison detects the misalignment of the hairpins or misshapen welding beads. Figure 5.4 visualizes the height data and shows the procedure. Figure 5.4(a) shows a good weld, which results in a curve with its maximum in the center. The defects in Figure 5.4(b) and (c) are visible in the curve profiles in

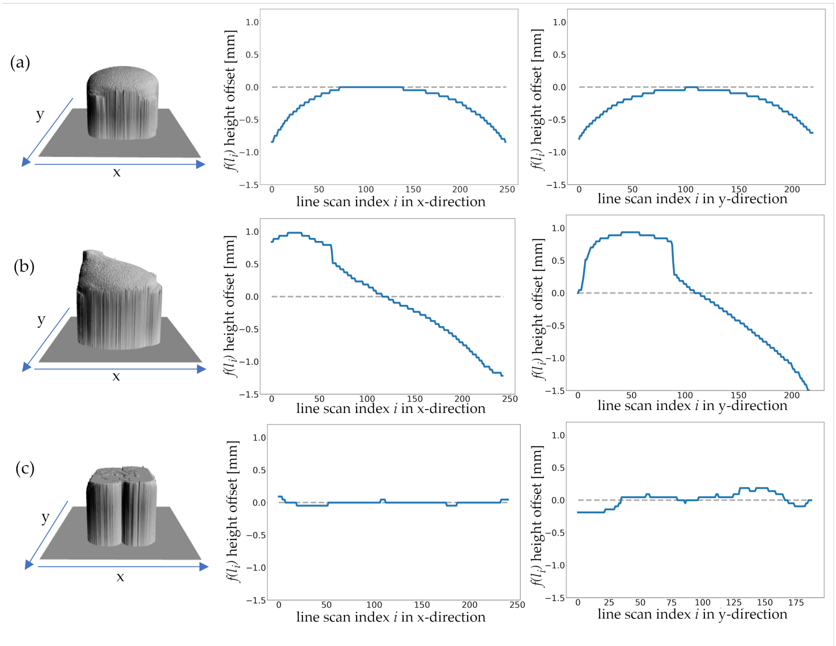


Figure 5.4 Difference of the maximum values of the line scans to the center. The maximum value of each line scan is determined. Then the difference to the center is calculated and the resulting values are plotted in a curve. Mathematically this means for each value $f(l_i) = -(h_c - \max l_i)$, where l_i is the line scan in x- or y-direction with index i and h_c is the height value in the center. Figure (a) shows a good weld, which results in a curve with its maximum in the center. Defective welds, such as misaligning pins (b) or pins that are not in the laser's focus (c), can be detected in the curve profile.

the x- and y-directions. While the misaligned pins (b) show an elevation on the left side instead of in the center, the almost non-welded pin pair (c) does not show any elevation. In addition to the curve profile, the algorithm evaluates the line scans' maximum and minimum distance to the height of the pin center. If the distance to the center is too small, the weld is not sufficiently stable. If, on the other hand, the minimum distance is too large, this provides information about pores or cracks in the weld. The algorithm also considers the width of the weld bead in the evaluation, as this allows conclusions concerning a radial or lateral offset between the pins.

5.4.2 Weld Area and Shape

The evaluation of the weld seam quality based on the camera image has been investigated in different works. A comparison of the quality evaluation using height maps and camera images shows that the camera images often also contain relevant information for the quality evaluation. The comparison investigates the detection of spatter deposits on the component. Furthermore, an evaluation of good and bad overlap welds is performed. Simple algorithms can be determined to detect characteristic defect properties, such as raised spatter deposits or a raised or collapsed seam surface on the height profile. For example, a threshold analysis can separate good from poor welding results. Using CNN for a pixel-wise classification, similarly good results could be obtained on the camera image. The detailed results of the analysis of the overlap welds are published in the conference proceedings of the *Forum Bildverarbeitung 2020* (Hartung et al. [183]).

As also mentioned in the introduction, it is not always possible to capture the height profile due to time constraints and the increasing cost and complexity of the system, including a height scanner. Therefore, this chapter shows an alternative approach to the one using the height data by deriving the quality-relevant properties of the weld from the grayscale image. Similar to the height values, the algorithm can also infer the width of the weld from the grayscale image. In addition, it can also detect the size of the weld surface in the 2D intensity images. This information provides information about the stability of the weld. For the detection of the seam area, threshold-based methods reach their limits due to the low-intensity differences and contrasts in the images. Another challenge comes from the reflective material properties of copper. The component reflects the light from the lighting above. Because the surface is not smooth, the light is reflected in different directions, and the camera captures different intensities. Figure 5.5 shows varying welding results with reflections and overexposed areas. These reflections further complicate the evaluation. However, CNN-based semantic segmentation detects the area well, even with a small network architecture. The model

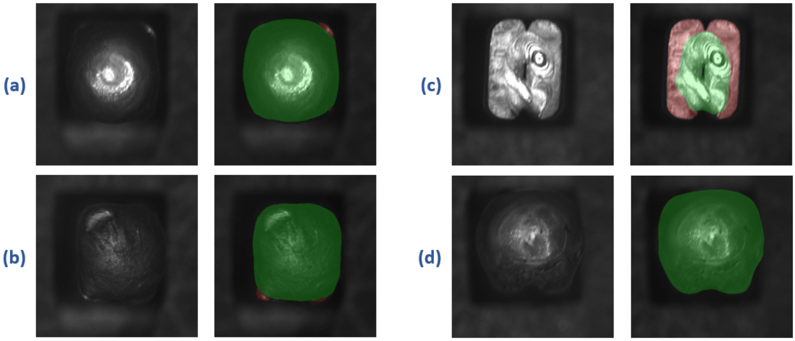


Figure 5.5 The detection of the surface of the weld and the unwelded pins are shown in a camera image. In each case, the right image shows the binary mask overlaid on the image (weld in green, unwelded pin surface in red). Different welding results are shown: (a) good weld, (b) misaligned pin pair, (c) pin not in the focus of the laser, (d) insulated copper rods.

is trained to a three-class problem to detect both the welded seam and the non-welded pin regions beside the background. Since the model deals with the same type of images as in Chapter 3, the SDU-Net architecture is used, which has prevailed in the evaluation from Section 3.4.1. Also, the same training configuration is used. The predicted masks are shown in an overlay representation in Figure 5.5.

Many defect cases are detected by evaluating the width of the weld and the size of the two classified areas. As a further evaluation, an algorithm analyses the contour shape of the weld seam. In good welds, the shape is approximately circular and has no solid corners and edges. However, if too little material is melted during welding, no round weld bead is formed, and the contour is slightly angular due to the initial pin shape. Other defects, such as copper pins that have not had their insulation stripped, also result in edges in the weld shape. Since the weld surface is a closed contour, Fourier descriptors can be used to characterize it. Analogous to Kuhl and Giardina [87], the algorithm computes the Fourier descriptors of the contours. Fourier descriptors are derived from the Fourier series for the cumulative angular function of the cross-section boundary. Thus, an evaluation of the Fourier series's harmonics considers the contour's complexity. In particular, in combination with the

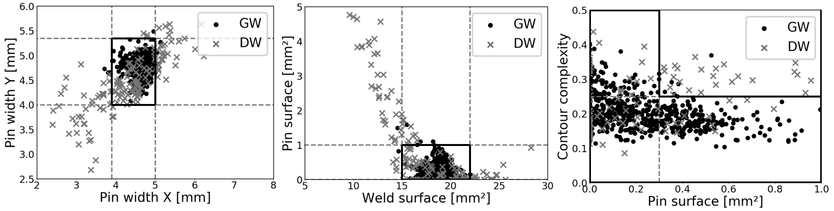


Figure 5.6 Quality-related features derived from the grayscale images. The correlation of the features derived from the 2D image with the seam quality based on the height profile is shown (GW -good weld, DW - defective weld). The gray lines symbolize the limit values for a GW and the thick black lines enclose the area in which the GW are located.

information about the size of the non-welded pin region, this contains information about insufficiently welded pin pairs. The relationship between the defined features and the evaluation result of the seam quality, based on the height profile, is shown in Figure 5.6.

Limiting the suitable ranges of the defined features makes it possible to detect insufficient welds. When comparing the pin widths in the x- and y-direction, for example, it becomes clear that many defective welds (DW) are outside a defined range. In contrast, good welds (GW) accumulate in this area. Also, the outliers can be identified as DW based on the total welded cases or the non-welded pin range. Thus, by setting a maximum limit for the non-welded pin area at 1 mm^2 and a maximum and minimum limit for the welded area 15 mm^2 and 22 mm^2 GW can be distinguished from DW. The contour complexity and the pin area are contrasted in the third diagram. Based on the correlation between the two features, a further separation between GW and DW is done. For example, if a larger portion of the non-welded pin area is visible, but the contour is still very circular, this indicates that only the outer corners of the pins are not welded. In this case, the weld can still be classified as GW. However, if the shape of the weld is more complex, this indicates that the pin area is visible because one of the pins was not fully welded. As a result, it is not possible to ensure a stable connection. Therefore, a high contour complexity combined with a high percentage of visible pin area can be considered another indicator for a DW.

5.4.3 3D Reconstruction

In the third approach, an AI-based single-view reconstruction method is used. This approach combines the advantages of the two methods just presented. First, it calculates the height profile from the captured camera image. For this purpose, only one camera image must be taken in the production line. Then the algorithm can replace the time-consuming OCT scan. Thus, further analysis can still be performed on the more informative height profile.

There are several approaches for image-based 3D reconstruction. The greatest challenge is using only a single camera image of the task. ML algorithms especially perform well in this application. Based on the state-of-the-art mentioned at the beginning of this chapter, the methodologies addressed there will be applied to the industrial data set of welded hairpins, among others.

SAE On the one hand, the approach of Zhang et al. [176] is applied using a stacked autoencoder (SAE). An SAE differs from a traditional AE in the way of training. It is characterized by training each layer separately and then stitching them together. An autoencoder learns to reconstruct an input image, compressing the input features into a low-dimensional latent space. Thus, the input also matches the expected output. Therefore, to use the model for 3D reconstruction, two separate models are first trained for feature extraction. One model learns the latent space of the intensity images, and the other learns that of the height maps. Then the encoder of the model trained on the intensity images is connected to the latent space and the decoder path of the second model. A fully connected layer is used for the connection of the latent spaces. The neurons in the fully connected layer are optimized for mapping to ensure a good connection. This results in a network with a 2D intensity image as input and a height map as output. The exact implementation of the network and the training procedure are explained in more detail in Appendix J.

GAN As a second method for 3D reconstruction, the CGAN is used following the approach of Baby et al. [7] and Arslan and Seke [5]. The generator of the CGAN creates a realistic height map y and receives an intensity image $\boldsymbol{x}$ as a constraint in addition to the noise vector $\boldsymbol{z}$. This results in the output $\boldsymbol{y}, G : \{\boldsymbol{x}, \boldsymbol{z}\} \rightarrow \boldsymbol{y}$. The construction of various GAN structures is possible with the choice of a generator network, a discriminator network, and a loss function. This work analyses four different structures. Table 5.1 shows the configurations.

The generator always consists of a U-Net modification. While the first three configurations follow the structure of Isola et al. [72], the fourth implementation follows the SDU-Net architecture defined in Section 3.4.1. Configuration I uses the PatchGAN analogously to the procedure of Isola et al. [72] as a discriminator. In contrast, configuration II, III, and IV use a deep convolutional GAN (DCGAN) [127] in combination with a conditional version of the loss function from the Wasserstein GAN (WGAN). The loss functions are regularized using L_1 or L_2 distance with the weighting factor $a = 100$. The training procedure of the networks uses the standard approach from Goodfellow et al. [57]. Appendix J shows the exact implementation of the network architectures of the different configurations and explains the training procedure.

Table 5.1 Configurations of GANs for 3D reconstruction.

Configuration	Generator	Discriminator	Loss function
I	U-Net	PatchGAN	CGAN + L_1
II	U-Net	DCGAN	WGAN + L_1
III	U-Net	DCGAN	WGAN + L_2
IV	SDU-Net	DCGAN	WGAN + L_2

U-Net The third model architecture used for the 3D reconstruction task is based on the U-Net architecture proposed by Ronneberger et al. [134]. The architecture has been introduced for the semantic segmentation task and has achieved outstanding results on different data sets. As well this work demonstrates its use for various semantic segmentation tasks in the previous chapters. The difference between the computation of height maps and semantic segmentation is that each pixel is assigned a corresponding height value as a label instead of a class assignment. This means that the output of the regression model is no longer converted to the probability of class assignment but to the corresponding height value. It is a more demanding task than semantic segmentation because the number of gray values extends the number of possible values. It includes up to 256 values depending on the height profile. In contrast, the number of classes in semantic segmentation is usually more limited than the number of height values. An advantage is that no manual labels have to be created since the height maps are available via a sensor recording.

To take advantage of the superior segmentation performance of the U-Net while overcoming drawbacks such as small receptive fields, the stacked variant of the extended convolution U-Net proposed by Wang et al. [166] is used. This architecture has repeatedly demonstrated its worth in previous chapters on similar image data. For the 3D reconstruction of welds, local areas such as spot heights or spatter are essential. However, larger areas, such as the shape of the weld bead, are also important. The different dilated convolutions capture both. The architecture of the SDU-Net has the same structure as in the semantic segmentation task. However, the training uses the MSE as loss function and an ADAM optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In addition, data augmentation with rotation, shift, shear, zoom, and flip is used for regularization. Since the model is tiny, with only 162 423 parameters, it can also be executed efficiently on the edge hardware directly on the plant.

Comparison Table 5.2 shows the results of the different models and corresponding configurations based on the mean absolute error (MAE), the standard deviation (SD), and the root mean squared error (RMSE). It also shows the number of parameters per network architecture. The number of parameters of the GAN configurations refers to the generator network since this is decisive in the inference, and the discriminator is only used for the training.

As a further evaluation, Appendix K shows reconstruction results of different examples. The jet colormap is used to visualize the height maps and the absolute errors compared to the ground truth.

The results show that the SDU-Net approach outperforms the other methods. In terms of model size and prediction result, measured by absolute error, the SDU-Net achieves better values than the GAN and SAE approaches.

One explanation why the SAE performs worse than in the example of Zhang et al. [176] can be traced back to the data used. While they use syntetic data, the hairpin images have more structures, reflections, and interfering elements. This complicates the reconstruction task. In addition, the autoencoder works as a loss compressor. Although it has many parameters, even the most parameters of the methods used, it loses much information per layer. In constrast, the U-Net architecture counteracts this problem with skip connections, among other things. The examination of the results in Appendix K shows that especially the edges of the contour are predicted to be very blurred with the SAE.

Table 5.2 Number of parameters and the mean MAE and RMSE of the 3D reconstruction algorithms. In addition of the MAE, its SD is calculated to indicate the dispersion in the test samples. The parameters of the GANs refer to the generator network.

Structure	Parameters	MAE (µm)	SD (µm)	RMSE (µm)
SAE	197 981 736	237.3	82.5	471.0
GAN I	54 419 713	197.7	85.9	473.9
GAN II	54 419 713	174.5	63.1	339.2
GAN III	54 419 713	142.2	57.2	303.0
GAN IV	162 423	130.0	70.1	341.8
U-Net	2 164 305	74.4	38.6	237.8
SDU-Net	162 423	**71.4**	**26.1**	**229.4**

The U-Net architectures use convolutional layers in addition to the skip connections, which effectively extracts the image features. Compared to the U-Net the extended SDU-Net is more efficient because it has a larger receptive field due to the stacked dilated convolutional layers. Therefore, it can better capture image pixel correlations and show higher robustness to local variations. This improvement in results is reflected in the application within the GAN structure and the end-to-end training of the architecture. Overall, the result is worse in the training process of the GAN. While the U-Net-based approach optimizes the parameters in the end-to-end training, the GAN trains two adversarial networks. The advantage of this training method is that it produces realistic-looking images. This can be seen, for example, in the clearly delineated edges. However, the model is forced to draw clear boundaries between the edges and the background from the beginning of the training. This can have a negative effect on the progress during training. The higher deviations at the edges are more significant when evaluated with MAE and RMSE. In addition, samples from underrepresented defect classes can be reconstructed worse by the GAN. This could be because the network is more oriented towards the frequently occurring samples showing good weld results.

Another reason the networks show larger deviations from the ground truth at the edges could be the manual mapping algorithm. This does not ensure that the edges are exactly matched. If slight deviations exist, this can lead to inaccuracies and inconsistencies in the neural networks.

The number of parameters affects the training time, the neural network's memory requirements, and the inference time. As discussed in the previous chapter, running on an edge device directly at the plant is preferable. This is possible due to the small and optimized model. Besides the good results, the number of parameters is another aspect that suggests using the SDU-Net architecture.

Quality Assessment The results from Table 5.2 were generated with a training-test split of 80 to 20. After reducing the training data $\mathbb{X}^{\text{train}}$ to 10%, $n = 95$ samples, an MAE of 93.5 µm and an SD of 68.7 µm could still be obtained with the SDU-Net on the test data set $\mathbb{X}^{\text{test}}$ with $n = 858$ samples. Because the data set of industrial manufacturing processes from one line is generally homogeneous, the learned features can be transferred well. Detailed evaluations of the reduction of the training data sets and the effects on the SDU-Net results are shown in Appendix L.

Since small deviations of a few micrometers are, in most cases, not relevant for quality evaluation, slight deviations in the reconstruction can be tolerated. More critical is the computation time for the algorithm and the effort to teach the algorithm. Using the small SDU-Net architecture, running on standard hardware directly at the plant is possible. Depending on the accuracy and the associated size of the input dimension, the execution time is 16 ms for 256×256 pixel resolution or 45 ms for 432×432.

To compare the model with both methods from Section 5.4.1 and 5.4.2, it is trained on the same training data. The SDU-Net is trained with $\mathbb{X}^{\text{train}}$ with $n = 95$ samples. The model uses an input resolution of 432×432. After calculating the height profile, the algorithm uses the same quality criteria as for the OCT scans in Section 5.4.1.

5.5 Result Comparison

The quality assessment of the $\mathbb{X}^{\text{test}}$ with $n = 858$ test samples is performed separately with each method to evaluate the different approaches. Ground truth is the division into GW and DW based on the features derived from the entire recorded height map using OCT. This method is often used as state-of-the-art in a quality check of hairpins [6, 94]. However, the disadvantages are primarily the execution time and the hardware costs. Therefore, the two methods that perform the quality assessment based on a camera image are compared with the result of the entire height profile recorded by OCT.

Section 5.4.2 and 5.4.3 evaluate the quality assessment based on the shape visible in the camera image (WS) and the AI-based 3D reconstruction (3D-R) data. When height data is used for quality assessment, only a few line scans are usually acquired due to time constraints. Therefore, another analyze uses an approach in which only six OCT scan lines (three in the x-direction and three in the y-direction) are considered in the evaluation (6L). One scan is in the center of the weld, and the other two are on each side. The feature vectors for the quality assessment are defined based on those of the entire height map. Table 5.3 presents the results using confusion matrices.

The ML-based 3D reconstruction using the camera images gives the best results of the three methods compared. 842 of the 858 test samples are classified in the same way as with the ground truth data, even if only the camera image was used as input. The discrepancies are due to borderline cases. As described in the previous section, the model is trained on $n = 95$ images and has an average deviation of 93.5 µm from the ground truth. Due to the rule-based partitioning into GW and DW,

Table 5.3 Confusion matrices to compare the results of the different methods. The results of the approaches: Weld shape extracted from the camera image (WS), ML-based 3D reconstruction (3D-R), and six line scans OCT (6L) are compared with the ground truth based on the features from the entire recorded height map (OCT).

	WS	WS
	GW	DW
OCT GW	679	20
OCT DW	25	134

	3D-R	3D-R
	GW	DW
OCT GW	694	5
OCT DW	11	148

	6L	6L
	GW	DW
OCT GW	688	11
OCT DW	20	139

in case of doubt, the deviation from one pixel value may yield a different result. One pixel value corresponds to a deviation of 46.8 µm in height and a difference of 18 µm in width. The borderline cases are welds where the width or the minimum height of the weld bead was barely reached with one method and just missed with the other.

When evaluating the results based on the camera images, it is noticeable that more pin pairs with height offset were detected as GW. This wrong classification can be attributed to the fact that the height offset is not considered in any of the used image-based classification features. The offset cannot be identified by the shape, size of the weld bead or the area of the unwelded pin surface. Therefore, this error case, unfortunately, often remains undetected. In contrast, samples that are incorrectly classified as DW can be attributed to tiny weld beads. If less material was melted during the process, the welds often have a rather rectangular shape due to the pin shape. In some cases, the height of the weld is sufficient to create a stable weld, although it still has an edged shape. Based on the camera image, these samples are classified as DW because they look very similar to unstable low-power welds. GWs with a round weld bead are reliably detected as GWs.

The evaluation with a few line scans also shows more deviating results than the evaluation with 3D reconstruction. In addition to borderline cases, these methods incorrectly classify pin pairs in which one of the pins was only partially connected or weld seams with a spatter as GW. Furthermore, insufficiently welded pin pairs (e. g., Figure 2.16(c, d)) were missed more often.

5.6 Evaluation and Discussion

This chapter showed different methods for the quality assessment of weld seams in hairpin welding. In addition to analyzing the acquired height profile, methods were presented that determined the quality based on a grayscale image. For the image-based evaluation, two different approaches are shown.

First, the approach in Section 5.4.2 uses features derived from the image, such as the width and shape of the weld, to perform a quality assessment. The most significant deficiencies were pin pairs, which have an axial offset between the pins. This misalignment is not captured in the image-based features and, thus, is not considered in the quality assessment. With this approach, the misalignment would have to be checked and corrected before welding, completely avoiding the faulty weld. In addition, Will et al. [170] show that radial and lateral misalignment have a more significant effect on weld quality than an axial misalignment measured by the electrical resistance of the weld joint. The significant advantage of using the image-based features is that no additional height scanner is needed. This reduces cost, setup effort, and acquisition time and allows quality analysis through a software update. The calculation of the binary mask following the approach shown in Section 5.4.2 only requires 16 ms on an i5-7300U CPU. It can be integrated into the process with the subsequent algorithmic evaluation without additional hardware requirements. The only additional effort is labeling the data, which is necessary to create the model. Thereby, the methods presented in Section 3.6 can be used to optimize the labeling process.

The second approach, shown in Section 5.4.3, performs an ML-based 3D reconstruction on a single grayscale image and then uses the computed height data for quality assessment. This approach achieves higher accuracy and correctly matches most test patterns, except for some borderline cases. The presented approach allows reconstruction based on a single grayscale image. When comparing the different approaches, the SDU-Net performes better than the GAN, the SAE, and the vanilla U-Net. The superiority of GANs and U-Nets over SAE is immediately evident in the results. Compared to training two adversarial networks in the GAN approach, the U-Net-based approach's end-to-end training results in higher accuracy. This could be because the GAN produces real-world images with sharp edges, even if it is not confident in the prediction. In addition, the SDU-Net is more efficient than the vanilla U-Net due to the stacked dilated convolutions. The calculation effort and prediction time are critical by running the 3D reconstruction algorithm for quality monitoring on an edge device directly on the plant. So for integrating the algorithm into the industrial manufacturing process, a small and optimized architecture is preferable. Using stacked dilated convolutions reduces the number of parameters since a less deep network is needed to cover the same receptive field. Depending on the size of the input dimension, the execution time of the SDU-Net is 45 ms for a resolution of 432×432 pixels. This method of single-image-based 3D reconstruction offers another possibility for quality assessment. In contrast to the feature-based evaluation of the camera image, a height scanner is required to train the AI model. However, this does not require manual data labeling with human effort for model training. After the model is trained, only one camera image is needed in the production system. In addition, the time for height scanning and manual pre-processing of height data for measurement errors can be saved.

Knowledge-Based Model The two-step approach with a downstream quality analysis based on expert knowledge and defined regularizations brings several advantages for model development. Using a semantic segmentation model makes it possible to work with a smaller data set than, for example, when directly classifying into GW and DW.

The result of a trained regression model on the image data that predicts the division in GW and DW is shown in Appendix M. This approach teaches a small CNN architecture on the two-class problem, resulting in a probability for a class label. The algorithm results in 62 false positive and 14 false negatives samples. These are 76 erroneous predictions out of 858 samples. In the training process, after a short time, only the training error improves, while the validation error increases. This behavior indicates overfitting to the training data and poor generalization performance of the model, despite the use of regularization methods.

The advantage of combining semantic segmentation and rule-based quality assessment is mainly due to three aspects: First, a pixel-wise loss function is used in semantic segmentation. This loss considers each pixel as an individual training pattern in its evaluation. Therefore, the effective number of training samples that optimizes the neural network increases [159].

Secondly, the algorithm does not need to separate GW and DW in the training process. It can consider all samples as equivalent training data. In most cases, there are only a few samples of defect cases since the production line should be set so that primarily good parts are produced. The failure cases often have to be intentionally provoked. The separation brings the disadvantage of training on an unbalanced data set, which can only be compensated for algorithmically to a limited extent.

The third aspect is that training can apply a stronger form of data augmentation in the semantic segmentation approaches. For example, the pixel-by-pixel mapping of the prediction mask must work for big and small zoom factors. In classification, too large a weld geometry means that the weld is defective, as in the defect case in Figure 2.16(f). Thus, a too large zoom factor could change the label from GW to DW. Due to the quality classification based on defined rules, not all error cases must be present in the training data.

Another advantage of the hybrid AI approach is that in the two-stage process, errors in the ML algorithm are detected by the downstream algorithm in many cases. The weld must be manually inspected if the rule-based algorithm reveals an error. For example, if the semantic segmentation assigns the wrong pixels to the weld surface or the pin surface, the rule-based algorithm will result in an error. The same applies to the reconstruction of erroneous height data, which lie outside of the defined good range. In these cases, errors of the ML are thus also detected.

Extension to other Data Sets Obviously, the methods are also applicable to other components, besides hairpins. For example, both the method of 3D reconstruction and the method of evaluation using semantic segmentation can be used to check welds of overlap welding. Figure 5.7 shows an example of the results of three overlap welds.

The figure shows in the first column an intensity image taken with a camera through the beam path of the focusing optics. Then the one-hot encoded semantic segmentation results with an SDU-Net architecture are shown. The results show the classes GW, DW, and spatter. A background

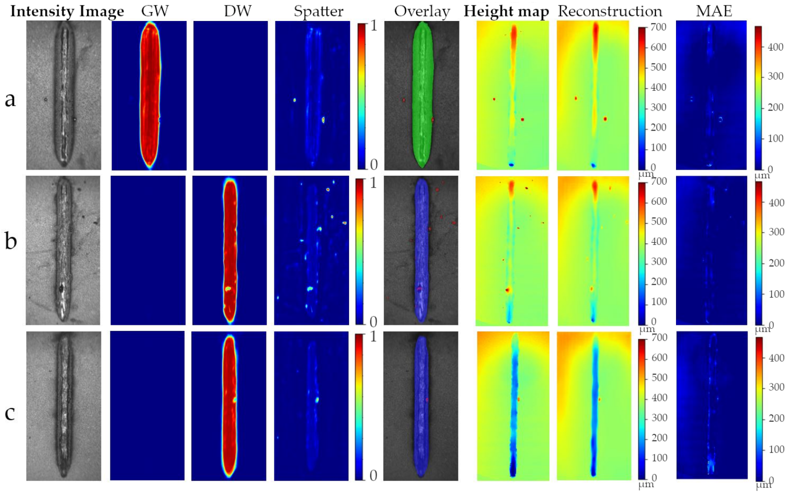


Figure 5.7 Semantic segmentation of overlapping welds: (a) good weld, (b) less power, (c) gap. The one-hot output layers are shown in the jet colormap and as an overlaid representation on the camera image (GW in green, DW in blue, spatter in red). In addition, the 3D measured values, reconstructed values, and the pixel-wise difference are also shown in a jet colormap representation.

class was also trained, which is not shown for space reasons. Then, in the fifth column, the figure shows an overlaid representation of the binarized class results with a threshold of 0.5 on the camera image. The sixth column shows the elevation profile recorded by OCT. For this, analogous to the procedure in Section 5.3, several line scans are performed, which are subsequently assembled into a height map. Due to the length of the weld seam and the chromatic aberration caused by the focusing optics, the height data is slightly distorted. These measurement errors were subsequently corrected but can still be seen slightly at the top of the false color representation. The seventh column shows a calculated height map with an SDU-Net based on the intensity image. These images also show slight distortion at the edge. This means the network learned this characteristic. The last column shows the difference between the ground truth of the height map and the calculated height map.

The examples show a linear weld joining two steel sheets. Figure 5.7(a) shows the result of a good weld with 6 kW. The second line is a weld with 4.5 kW, resulting in an insufficient joint. In Figure 5.7(c), a gap of 500 µm was created between the steel plates. This also creates an inadequate joint. The seam incidence is readily apparent in the elevation data plot. In both (b) and (c), the seam does not protrude beyond the measured height of the sheet. Especially (c) clearly shows that the seam's height is below the height of the sheet. The errors can also be detected by semantic segmentation in the camera image, where the seam pixels are classified as DW.

All three samples contain spatter that has been deposited on the sheets and the weld seams after the welding process. These were detected in the one-hot class of semantic segmentation but can also be detected in the 3D representation using a threshold-based method. This example also shows that semantic segmentation using CNN can distinguish a good seam area from a bad one. Due to the pixel-by-pixel class assignment, only individual defective seam areas are recognized as DW. This allows the defect to be localized.

6 Conclusion and Outlook

The work addresses the monitoring of laser welding processes. The aim is to obtain the optimal added value without modifying the welding station. Extensions of the hardware components result in hardware costs and often involve complex configurations and setup changes. On-axis mounting on the focusing optics of the laser welding station does not affect the welding process significantly, but the alternating effects of the laser beam and signal redirection must be considered in the signal. The work uses a grayscale camera mounted coaxially to the laser beam on the focussing optics. In addition, a ring illuminator in the wavelength range 625 nm and a computing unit with an Intel Core i5-7300U processor are used. Machine learning methods are used to extract the relevant features from the image data. This poses challenges regarding computation time on an edge device, the effort required for data labeling, and user acceptance. This work considers these aspects in developing the pre-, in-, and post-process monitoring.

The pre-processing in Chapter 3 includes calculating the exact welding position concerning the component position. Since the component is not exactly placed in many applications, its position is captured in the camera image, and the translation and rotation of the position are calculated. Due to the component geometry, reflective material properties, and different surface structures, detection in the grayscale image is often non-trivial. By extending the approach with ML-based semantic segmentation, the relevant pixels of the component surface are highlighted and can be easily processed in a downstream step. This highlighting also makes it possible to check the component geometry and the position of the joining partners in relation to each other. This way, it is possible to detect errors in pre-processing steps at an early stage.

The monitoring while welding primarily monitors the occurrence of spatters, as this serves as an indicator of an unstable process. Due to acquiring images during the welding process without upstream filters, the images contain more interfering elements due to plumes, process lights, and reflections of fixtures. The algorithms of Chapter 4, therefore, use semantic segmentation with a CNN, which infers the pixel-by-pixel class assignment based on various image contexts. This approach also offers the advantage of extending it to other classes, such as a cooling phase of the weld. Furthermore, the effect of a reduced acquisition frequency is investigated to estimate the captured spatter fraction without a high-speed camera.

Different methods are shown and compared in Chapter 5 for the quality monitoring of the solidified weld seam after the process. The monitoring by a camera is determined in one approach by the detected size and geometry of the weld seam. The different surfaces, e. g., good quality, defective seam surface, and unwelded component surface, are caught in the image with a pixel-by-pixel classification. Based on the number of assigned pixels and the contour shape, statements can be made about the seam quality. Alternatively, an approach for a single image-based reconstruction of height data is presented. For this purpose, an ML-based algorithm is trained with an image and assigned height data, which is subsequently used to calculate the height map based on an intensity image. Depending on the application, a more accurate definition of the weld quality is possible using the reconstruction approach compared to the pixel-wise class assignment.

When used in industrial manufacturing, ML algorithms must meet specific requirements. One crucial point is the database, which is often small and contains few error cases. In addition, the labeling should be done by application experts and costs time and resources, which is a hurdle for using ML models. To counteract this, ML methods can be used, which enable optimized training. Methods for the selection of relevant data, as well as the support in the labeling process by early model predictions, were shown in Section 3.6.

Another essential topic is edge computing in relation to ML applications. Among other things, due to data security, transmission delays, scalability, improved inference time, and low network loads, the computation of algorithms directly at the plant is often preferred. For this purpose, small network architectures optimized for the respective use cases are developed. These network architectures are limited to the welding station's defined task and the specific use case. They have a lower generalization performance but make use of their defined environment. Further optimization is conceivable, for example, by considering the component geometry information from the CAD system in the model training. Research is active in the field of learning with additional knowledge. The information could, for example, be integrated into the loss function of the network. This information could also be used to monitor the model prediction further. In addition, application-related data augmentation is also possible and can bring further advantages. For example, the images in the laser welding process often show similar structures, such as reflections. As a result, the model can be trained faster and more robustly, even with little training data, through adapted data augmentation. However, care must be taken to avoid introducing unnecessary variance into the training.

The algorithms presented are based on a hybrid approach combining ML algorithms with a knowledge-based system. For example, in post-process monitoring, not only a classification into good and poor is done. The ML model calculates the relevant features, which are then assigned to the respective class by a knowledge-based system. The network does not need to know the failure class, and the user can define it with understandable rules. In addition, this has the advantage that the downstream system also monitors the ML algorithm and generates an error in case of a wrong prediction. Section 3.4.3 presents additional methods for quantifying uncertainty in model prediction monitoring. The field of uncertainty quantification is still very active in research. Because the results of AI applications often cannot be tracked exactly, their use is often viewed critically. Therefore, monitoring the results better and detecting erroneous predictions is essential. Likewise, it is important to decide when a model is performing well. Often it isn't easy to estimate whether the entire relevant data variance is represented in the model. In Section 3.6, approaches are compared to support this process. However, further research is also open here, which estimates the model's quality.

The methods presented in pre-, in-, and post-process monitoring have been analyzed independently. However, their combination offers the potential for holistic monitoring and support of the welding process.

For example, the post-process evaluations, in combination with the preprocessing analysis, can provide information on an optimized welding strategy. Currently, the welding parameters and the welding geometry are already modified based on the position of the joining partners. Combining the information with the welding result allows the following components to be optimally welded by adjusting the welding parameters. The combination of information about spatter occurrence in the process and the position of the joining partner before the procedure also allow conclusions that help to parameterize the welding process better. In addition, the use of reinforcement learning offers great potential and many possibilities for optimizing the welding process.

As mentioned several times, this work relies on data from a single camera sensor. In general, the monitoring of the welding process can be improved and made more robust by extending the sensor technology. To realize 100% monitoring, using different sensors is an obvious solution. Evaluating various databases with sensor fusion can provide more comprehensive information. A variety of algorithms can be used for this purpose, including ML algorithms.

Appendix

A Evaluation Model Architectures

This chapter complements Section 3.4.1, which shows the analysis of different model architectures for the pre-process monitoring algorithm. This chapter extends the investigation with the training histories of training with varying splits of training and validation data. The analysis is based on a data set $\mathbb{X}$ with $n = 900$ samples of a hairpin welding process. It compares the training history for training sessions with different training data sizes. The size of the data set $\mathbb{X}^{\text{train}}$ is varied from $n = \{5, 10, 25, 50, 100, 200\}$, while the size of the set of validation data $\mathbb{X}^{\text{val}}$ remains stable at $n = 500$ and the size of the test data set $\mathbb{X}^{\text{test}}$ with $n = 200$ for better comparability. Ten training, validation, and test data splits were performed for any size n of the training data set, with five independent networks, each trained from scratch. Detailed results of the training sessions per model, broken down by accuracy, can be found in the following.

The training is performed with $BS = 2$ and 100 steps per epoch. For optimization, it uses the categorical focal loss with $\alpha = 0.25$ and $\gamma = 2$. In addition, it uses an ADAM optimizer with the hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, which control the length of the moving averages. The learning rate starts at $\epsilon = 0.001$ and is reduced during training after three epochs without improvement. Data augmentation and early stopping based on the validation data set are used for model regularization.

Results on the data set $\mathbb{X}^{\text{train}}$ are shown in red and $\mathbb{X}^{\text{val}}$ in blue. After every trained epoch, the evaluation is performed with 100 steps. Chapter 3.4.1 shows the definition of the model architecture in detail. In addition, the chapter processes and evaluates the results.

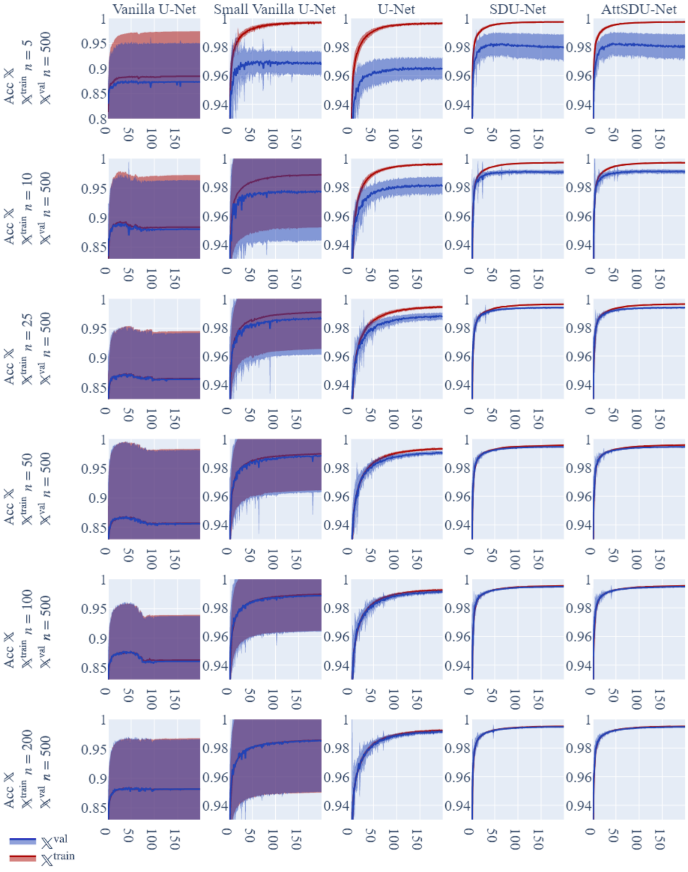


Figure A.1 Accuracy of all model achitectures. The red lines show the averaged results of the 50 training sessions on the data set $\mathbb{X}^{\text{train}}$ and the blue lines for $\mathbb{X}^{\text{val}}$. The variance is presented within the shaded area. The x-axis represents the progression of epochs during the training process. Each epoch includes 100 steps with $BS = 2$. The evaluation was performed after each trained epoch with 100 steps.

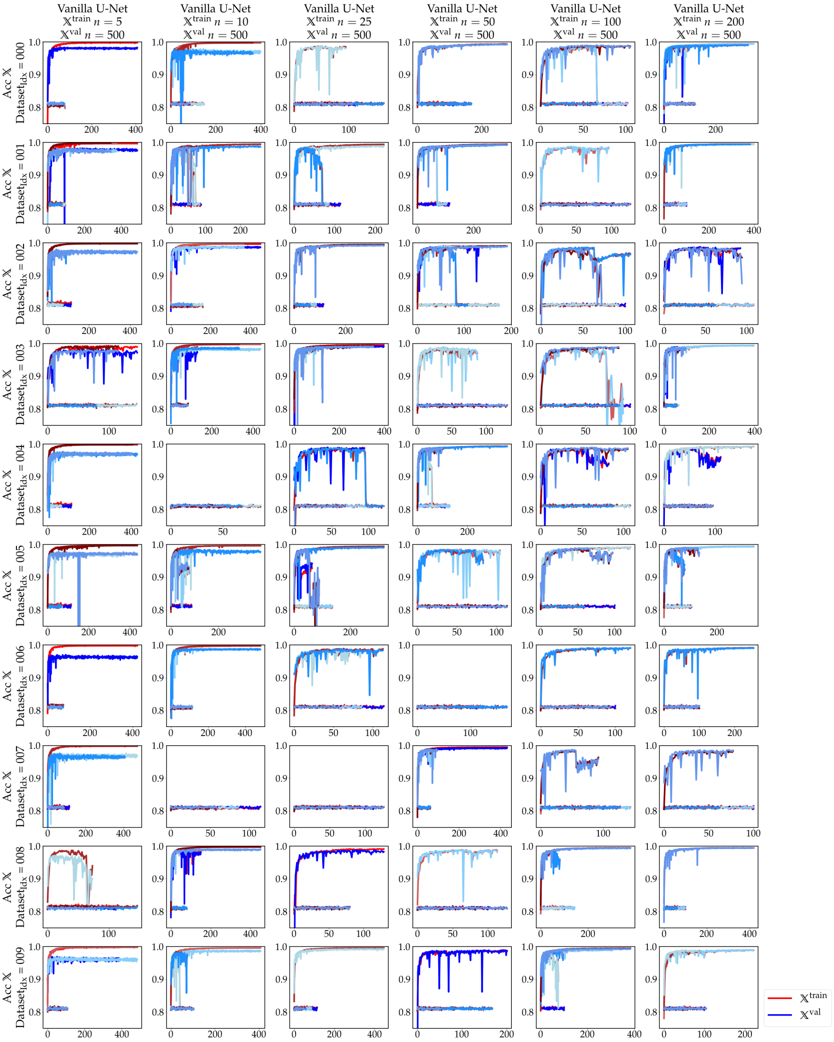


Figure A.2 Accuracy of the vanilla U-Net. In red are the results on the data set $\mathbb{X}^{\text{train}}$ and in blue for $\mathbb{X}^{\text{val}}$. The x-axis represents the progression of epochs during the training process. Five model were trained on each data split from scratch.

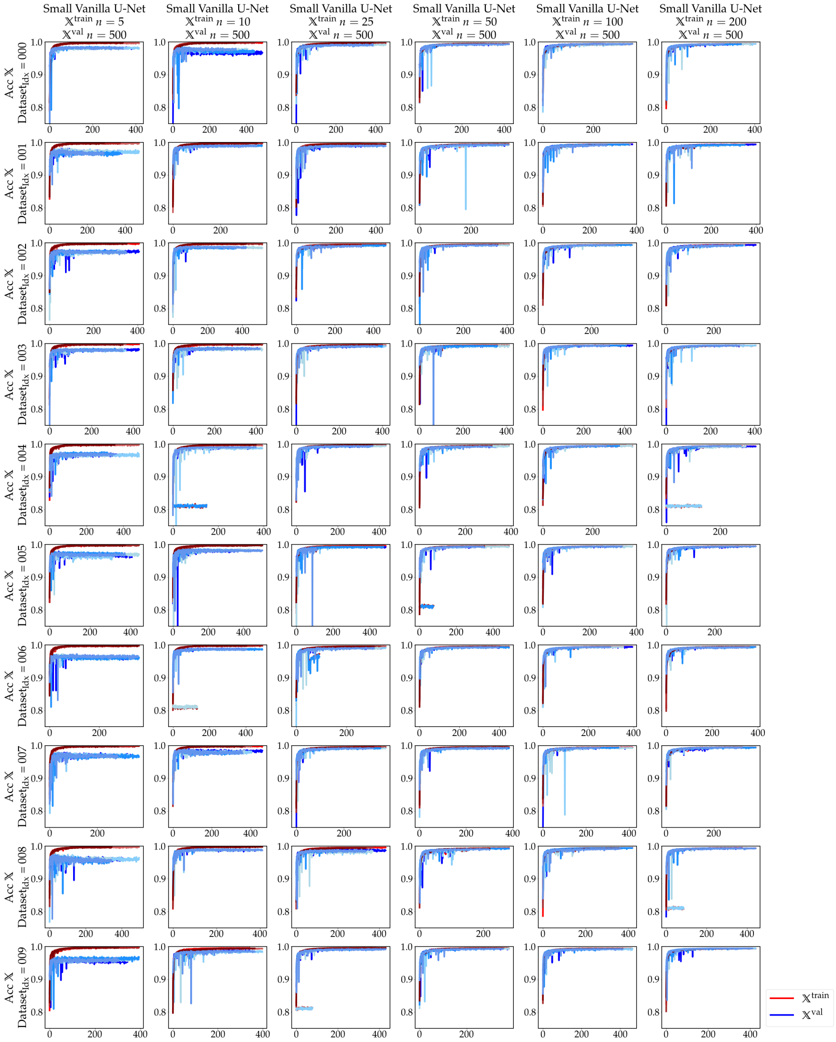


Figure A.3 Accuracy of the small vanilla U-Net. In red are the results on the data set $\mathbb{X}^{\text{train}}$ and in blue for $\mathbb{X}^{\text{val}}$. The x-axis represents the progression of epochs during the training process. Five model were trained on each data split from scratch.

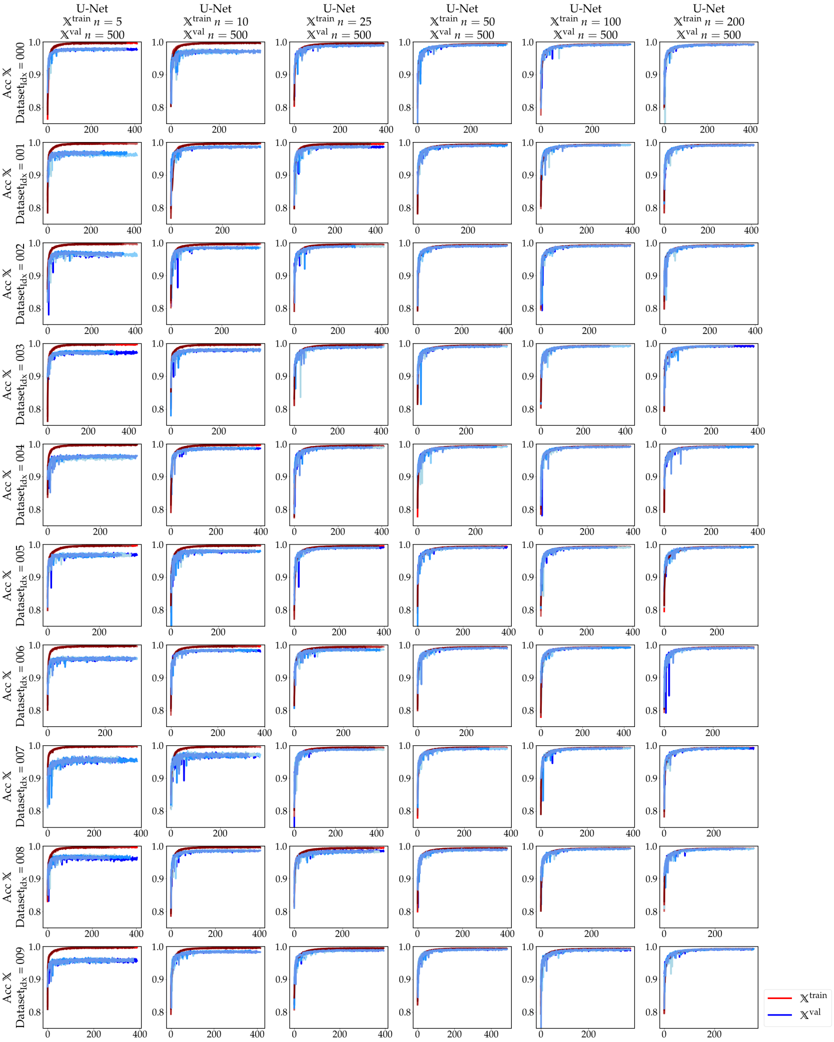


Figure A.4 Accuracy of the U-Net. In red are the results on the data set $\mathbb{X}^{\text{train}}$ and in blue for $\mathbb{X}^{\text{val}}$. The x-axis represents the progression of epochs during the training process. Five model were trained on each data split from scratch.

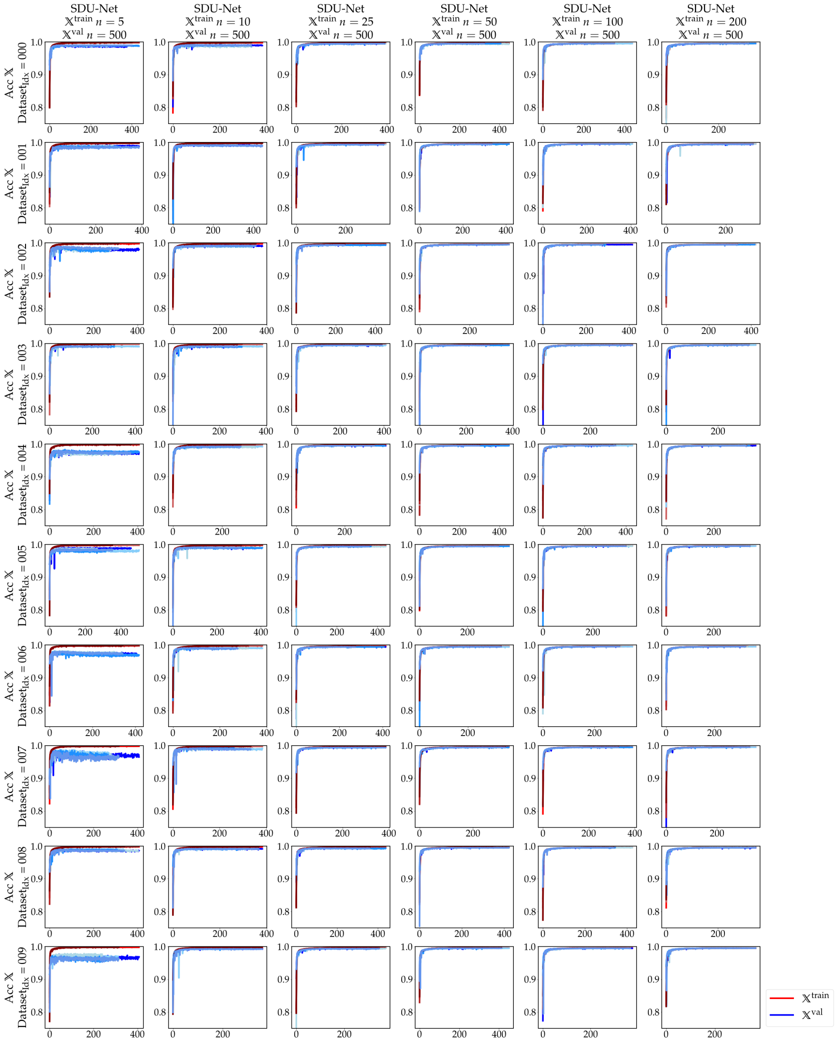


Figure A.5 Accuracy of the SDU-Net. In red are the results on the data set $\mathbb{X}^{\text{train}}$ and in blue for $\mathbb{X}^{\text{val}}$. The x-axis represents the progression of epochs during the training process. Five model were trained on each data split from scratch.

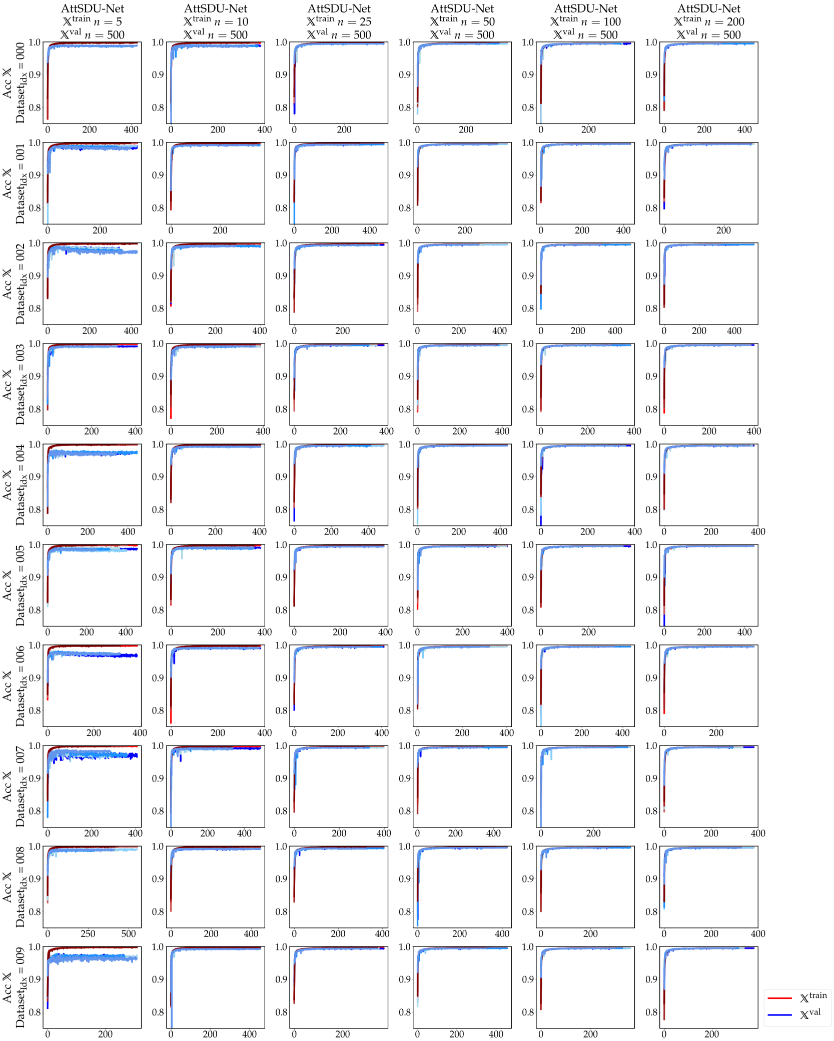


Figure A.6 Accuracy of the AttSDU-Net. In red are the results on the data set $\mathbb{X}^{\text{train}}$ and in blue for $\mathbb{X}^{\text{val}}$. The x-axis represents the progression of epochs during the training process. Five model were trained on each data split from scratch.

B Evaluation Loss Functions

Complementing the definition of the model architecture in Section 3.4.1, this chapter describes the selection of the optimal loss function for the model used in the pre-process monitoring. Therefore, it compares different loss functions commonly used for semantic segmentation. The SDU-Net defined in Chapter 2.1.5 is used as network architecture. The comparison considers problems with one and several foreground classes, whereas the data are always one-hot encoded. All training sessions use an ADAM optimizer with the hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In addition, data augmentation and early stopping based on the validation data set are applied.

The evaluation considers focal loss, dice loss, and cross entropy. In the case of focal loss, a parameterization with $\alpha = 0.25$ and $\gamma = 2$ is used. Only the foreground classes are considered in the calculation for the dice loss, analogous to the procedure of Zhang et al. [179]. That means the background class, which often takes the most significant part of the image, is not included in the calculation. Finally, the weighting of the weighted cross entropy loss is defined based on the pixel-wise class ratio of the data set.

Further Analysis of the Data Set from Appendix A The first section compares the different loss functions on the data set of Chapter 2.1.5 and Appendix A with $\mathbb{X}^{\text{train}}$ containing $n = \{50, 200\}$ samples and $\mathbb{X}^{\text{val}}$ containing $n = 500$ samples. In the case of the weighted cross entropy loss, the weighting factor is $[0.3, 0.7]$ for the background and foreground classes, respectively. Figure B.1 shows the training histories with the SDU-Net architecture using the different loss functions. The plots show the mean as a line and the variance in the shaded area for each of the 50 training sessions performed with ten different training-validation splits. The results show only a few differences in training with the different loss functions. Regardless of the loss function, the accuracy

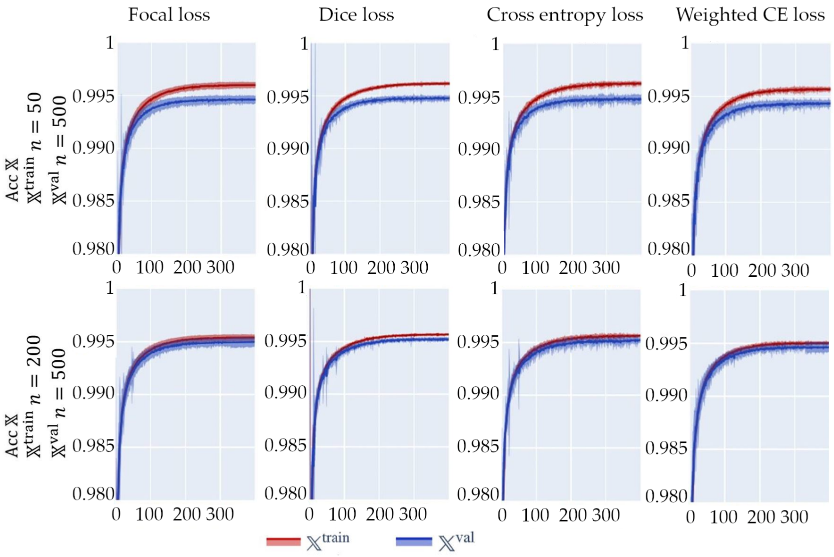


Figure B.1 Accuracy in the training process using different loss functions. The data set of appendix A with 10 training-validation splits for $\mathbb{X}^{\text{train}}$ with $n = \{50, 200\}$ data sampels is used and 5 trainings of scratch are performed. The comparison shows the mean and variance of the accuracy of the training sessions over 400 epochs, with 100 steps per epoch and $BS = 2$.

of the training data is higher for $n = 50$ training samples than for $n = 200$ samples. However, the model accuracy on the $n = 500$ validation data is comparable regardless of the number of training images, settling at $\text{acc} \approx 0.995$. In addition, the course of the training curves is also similar. When comparing the 50 training sessions from scratch, the variance is minimally lower when the dice loss is used instead of the other functions. The weighting factor has no significant effect on the cross-entropy loss since it already performs well without weighting. Also, the focusing factor γ of the focal loss does not provide much improvement compared to the cross entropy in this case.

Class Number and Pixel Proportions Furthermore, evaluations were performed on another data set, comparing loss functions with a focus on different class numbers and pixel proportions. The number of training data $\mathbb{X}^{\text{train}}$ is $n = 23$, while the validation data set $\mathbb{X}^{\text{val}}$ contains $n = 9$ samples. Figure B.2 shows an example image from the used data set. A camera image was deliberately used, which shows many detection possibilities in different sizes. There is no specific use case when performing the analysis.

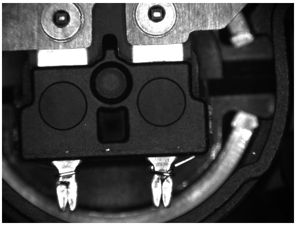
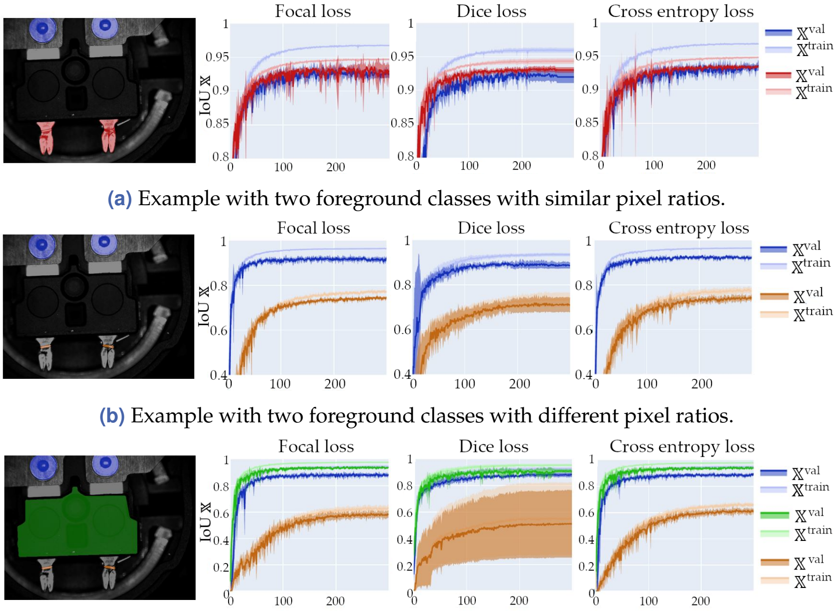


Figure B.2 Example image of the data set.

Figure B.3 shows each diagram's average training history and the variance for ten training sessions. Each session starts from scratch. The diagrams show the IoU of the foreground classes to be able to evaluate the model performance based on the defined classes and not only to get a holistic image evaluation. In this image evaluation, minor classes are considered weaker. To make the graphic easier to read, the values of the background class are not shown. Since the background class contains most pixels, it usually has a high IoU, while the values of the other classes have a more significant variance. In the images to the left of the diagrams, the part to be recognized is marked in the curve's color. Since the result of the validation data set is more meaningful, it is shown more prominently, while the result of the training data is shown only faintly in the background. The diagrams show the mean of the training sessions as a line and the variance as a shaded area.

Considering Figure B.3(a), no major differences regarding the loss function used can be detected. Both foreground classes to be detected have similar pixel ratios. Thus, the evaluation using IoU is also in a similar range for both classes. All three loss functions achieve a good result in all train sessions, which is shown by a low variance in the diagrams.



(a) Example with two foreground classes with similar pixel ratios.

(b) Example with two foreground classes with different pixel ratios.

(c) Example with three foreground classes with different pixel ratios.

Figure B.3 The plots show the IoU of the different foreground class layers over the course of training. The x-axis shows the number of epochs, while the y-axis shows the IoU. The model is trained with 100 steps per epoch and $BS = 2$. The validation also performs 100 steps per epoch. The mean and the variance of ten training sessions from scratch are shown.

Figure B.3(b) shows an example where a larger class (blue) and a small class (orange) are to be detected. The first thing to notice is that the IoU of the orange class is always lower than the blue class. This is because fewer pixels are assigned to the class, so a wrong assignment has a greater effect on the result. In addition, slight differences can be seen in the course of the loss functions. Especially the variance increases by using the dice loss for training. Furthermore, the result after 300 epochs is worse on average than using the focal loss or the cross entropy.

This behavior becomes even more evident when considering the results of Figure B.3. In this example, three classes with different-sized objects are detected. Especially with the small class (orange), the variance of the results increases when using the dice loss. This is due to single models, which give a bad result. In these cases, the small class is completely assigned to the background. The dice loss calculates the value for each class separately and offsets the values. Therefore, it is usually designed for unequal class ratios and does not need to be parameterized. However, since the dice loss is optimized for a binary answer, there is a higher risk that the optimal result is not found in the optimization. In most cases, however, good results can be achieved with all three loss functions.

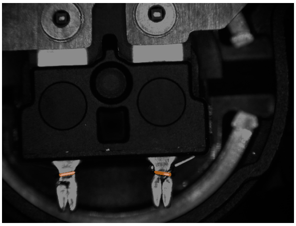


Figure B.4 Example with one foreground classes with a small pixel ratio.

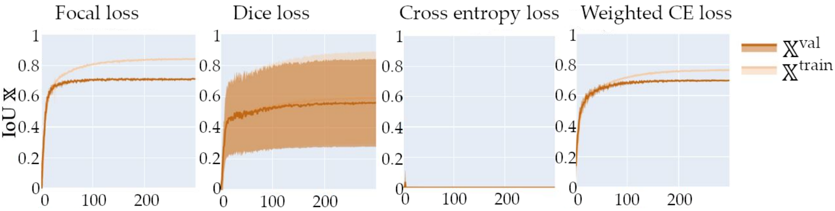


Figure B.5 The plots show the IoU of the different foreground class layers over the course of training. The x-axis shows the number of epochs, while the y-axis shows the IoU. The model is trained with 100 steps per epoch and a $BS = 2$. 100 steps per epoch are also used in the validation. The mean and the variance of ten training sessions from scratch are shown.

Teaching only a small class shows the same behavior. The relevant class is marked in the Figure B.4, while Figure B.5 shows the results. Ten training sessions from scratch were performed in each case, of which the mean and variance are shown. Even in this case, some models trained with the dice loss perform worse on the data sets. Even with a small class, the loss of dice does not always result in the correct allocation of pixels and assigns them the background class. This example also shows a disadvantage of entropy loss, which performs poorly on unequal class ratios. With entropy loss, the foreground class cannot be detected. This results in all pixels being assigned to the background. Due to the only small foreground class, the loss can still be optimized to a small value. The weighted entropy with the factor 0.1 for the background and 0.9 for the foreground class achieves similar results as the focal loss.

Conclusion Based on the results, focal loss is the most suitable loss function for different data sets. The advantage is that it does not have to be parameterized individually for a data set. In addition, the loss function can handle multiple classes, unequally distributed classes, and also individual deviating samples in the training data set. Due to the weighting factor γ, which focuses on uncertain samples, these are considered more strongly in the optimization.

C Model Calibration

Model calibration is essential to use the probability values per class in the result validation of the neural network used in preprocessing monitoring. Therefore, this Chapter complements Section 3.4.3 by discussing model calibration and uncertainty estimation in more detail.

A reliability diagram is used to evaluate the model calibration, which represents the individual bins of the ECE. The x-axis is the probability of prediction, and the y-axis is the proportion of actual assignment to the class. In a well-calibrated model, the columns should be on the diagonal.

The evaluation is based on a model trained on a data set of copper wires to be welded. This data set contained only good examples, i. e., two correct copper wires were always clamped in the fixture. Afterward, the evaluation uses a testing data set, including deliberate error cases. These are, for example, a missing copper wire or an already welded pair of pins.

Two models are trained for the detection of the component position. One model uses the dice loss, and the other the focal loss. Similar to the procedure from Chapter B, the dice loss is trained considering only the foreground class. The focal loss is parameterized with $\alpha = 0.25$ and $\gamma = 2$. Analogous to Mehrtash et al. [107], the evaluation calculates metrics and graphs on the area around the foreground segments. The background usually has the slightest uncertainty but takes many pixels. Therefore, the evaluation has limited the metric to an area surrounding the foreground segments marked in the graphs with a red rectangle.

The metrics shown in Table C.1 and the diagrams in Figure C.1, Figure C.1, Figure C.3 and Figure C.4 show that the models are better calibrated for a correct sample than for an unknown sample. This is shown by a lower ECE and in the reliability diagram because it is more aligned to the diagonal.

Figure C.1 and Figure C.2 show the results for examples with a correct initial situation, which are analogous to the training data. Figure C.3

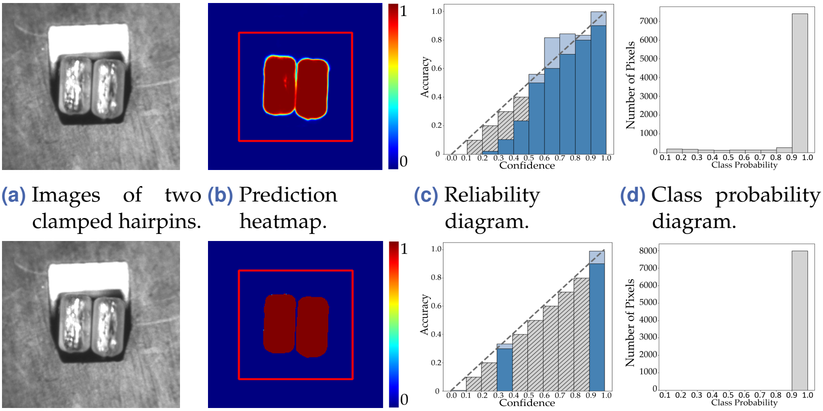
Table C.1 This table shows the ECE and entropy metrics $\mathcal{H}$ for the foreground class using different examples. The corresponding images, as well as the reliability and the class probability diagram, are shown in the linked figures.

Loss function	Example	Image	ECE	$\mathcal{H}(\hat{\boldsymbol{y}}_j)$
Focal	Two pins	C.1	1.127	0.042 217
Focal	Two pins	C.2	1.388	0.046 599
Focal	One pin	C.3	2.913	0.051 257
Focal	Welded pins	C.4	-	0.105 175
Dice	Two pins	C.1	1.599	0.000 094
Dice	Two pins	C.2	1.142	0.000 165
Dice	One pin	C.3	1.891	0.000 108
Dice	Welded pins	C.4	-	0.000 402

shows a faulty situation where one pin is missing. The data sample is labeled for evaluation with just one pin marked. However, such data samples were not present in training. Figure C.4 shows an already welded pair of pins. No mask has been assigned to this image because the model is not trained to detect welds. Therefore, there is no calculated ECE and no drawing of a confidence matrix for this example.

In contrast to dice loss, focal loss provides better calibrated results. The dice loss forces a binary result due to its evaluation using the dice coefficient. Thus, even in the case of an error, it obtains class probabilities with the values 0 and 1. Therefore, when using the dice loss, it is recommended to use an additional model calibration [107]. The focal loss, in contrast, is better calibrated. Thus, downstream methods like ensemble training, Monte-Carlo-Dropout, or temperature scaling are not mandatory [107, 164].

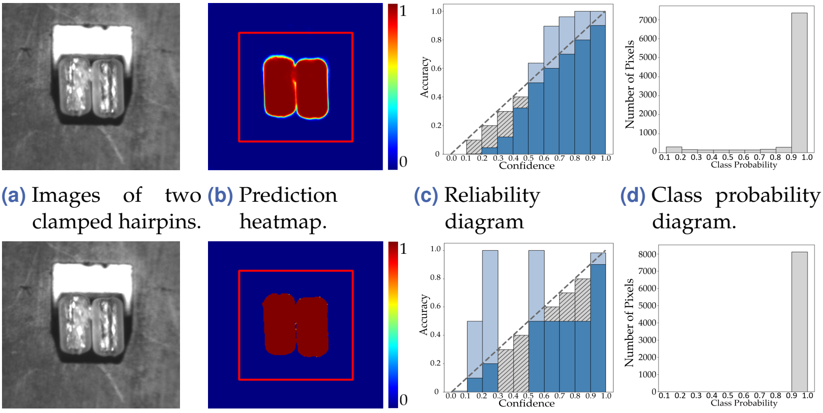
Also, the results show that using the metric $\mathcal{H}(\hat{\boldsymbol{y}}_j)$ considering the foreground class $\boldsymbol{y}_j$, without the presence of data labels, the uncertainty of the model can be predicted. The value for the welded pin pair is higher than the other values for both focal and dice loss. In case of a missing pin, the prediction doesn't work as well. Because the features of the remaining pin were included in the training data set, the model can predict this pin with high accuracy. Figure C.3 shows that the model is uncertain at the edge of the area where a second wire should be, but most of the area is assigned to the background with high accuracy.



(a) Images of two clamped hairpins. (b) Prediction heatmap. (c) Reliability diagram. (d) Class probability diagram.

(e) Images of two clamped hairpins. (f) Prediction heatmap. (g) Reliability diagram. (h) Class probability diagram.

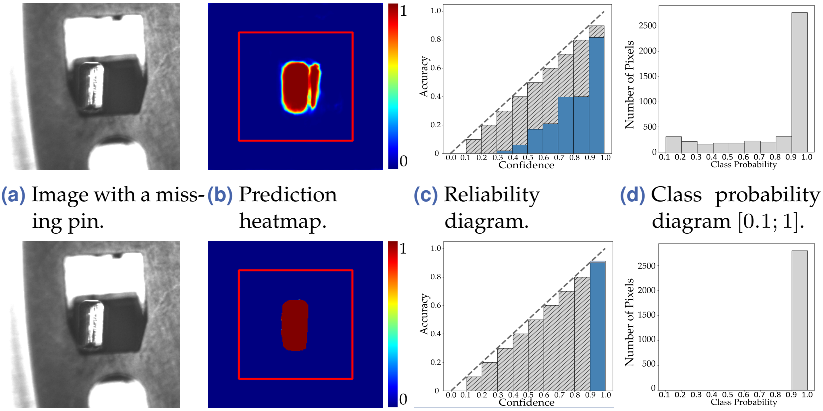
Figure C.1 Model prediction of a sample with two clamped pins for the foreground class. The first row shows the result of a model trained with focal loss, while the model in the second row was trained with dice loss.



(a) Images of two clamped hairpins. **(b)** Prediction heatmap. **(c)** Reliability diagram **(d)** Class probability diagram.

(e) Images of two clamped hairpins. **(f)** Prediction heatmap. **(g)** Reliability diagram **(h)** Class probability diagram.

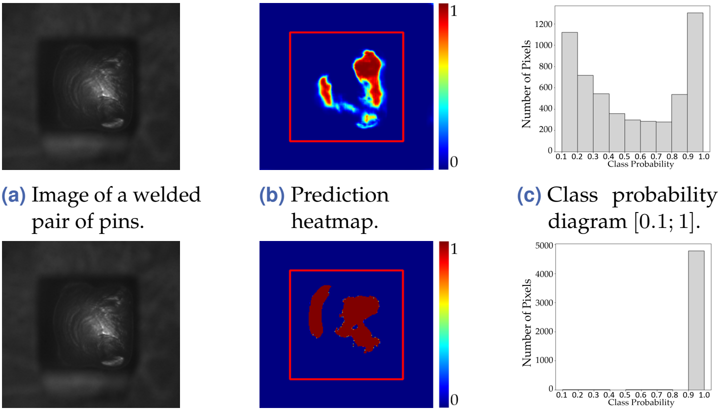
Figure C.2 Foreground class results of a sample with two clamped pins. The models are trained with focal loss (first row), and dice loss (second row).



(a) Image with a missing pin. **(b)** Prediction heatmap. **(c)** Reliability diagram. **(d)** Class probability diagram [0.1; 1].

(e) Image with a missing pin. **(f)** Prediction heatmap. **(g)** Reliability diagram. **(h)** lass probability diagram [0.1; 1].

Figure C.3 Foreground class results of a sample with two clamped pins. The models are trained with focal loss (first row), and dice loss (second row).



(a) Image of a welded pair of pins.

(b) Prediction heatmap.

(c) Class probability diagram $[0.1; 1]$.

(d) Image of a welded pair of pins.

(e) Prediction heatmap.

(f) Class probability diagram $[0.1; 1]$.

Figure C.4 Foreground class results of an out-of-distributaion sample. The models are trained with focal loss (first row), and dice loss (second row).

D SDU-Net with Dropout

Section 3.6 focuses on reducing effort in the labeling process, which includes defining a reasonable labeling order of the images. One investigated method is the so-called Monte-Carole dropout, whose usage is explained in more detail in Section 3.6. The MCD uses the dropout layer in training and inference time, which means that the network architecture has to be extended by dropout. The resulting architecture is described in more detail in this chapter.

The chapter defines an SDU-Net architecture with added dropout layers in the inner layers, following the approach of Kendall et al. [82]. Except for the added dropout layers, the definition of the SDU-Net remains unchanged from the definition in Section 3.4.1.

The first two operation blocks with filter size $n_{\text{out}} = \{16, 32\}$ remain unchanged, so there is no loss of information. Next, a dropout layer is added to each middle operation block. Analogous to the encoder path, a dropout layer is also added in the first two layers of the decoder path. The dropout is performed before the connection with the feature map from the encoder path via the skip connection. Figure D.1 represents this by a dark gray block in front of feature maps.

The dropout operation is performed after the pooling or upsampling operation but before the convolutions. The proportion of randomly selected units to be removed is defined by the parameter p_d. This value can be varied. For the example in Section 3.6, the network architecture is used in MCD to quantify uncertainty. There the parameter is defined with $p_d = 0.5$.

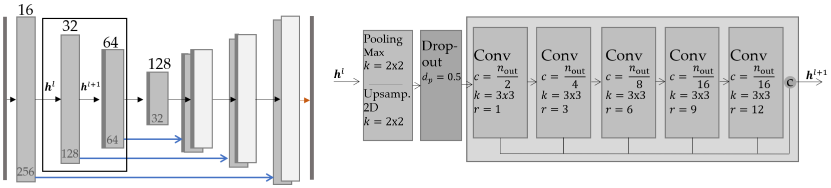

16
32
64
128
h^l
h^{l+1}
256
128
64
32
h^l
Pooling Max $k = 2x2$
Upsamp. 2D $k = 2x2$
Drop-out $d_p = 0.5$
Conv $c = \frac{n_{\text{out}}}{2}$ $k = 3x3$ $r = 1$
Conv $c = \frac{n_{\text{out}}}{4}$ $k = 3x3$ $r = 3$
Conv $c = \frac{n_{\text{out}}}{8}$ $k = 3x3$ $r = 6$
Conv $c = \frac{n_{\text{out}}}{16}$ $k = 3x3$ $r = 9$
Conv $c = \frac{n_{\text{out}}}{16}$ $k = 3x3$ $r = 12$
C
h^{l+1}

Figure D.1 SDU-Net architecutre with dropout layers. The representation is analogous to the description in Section 3.4.1.

E In-Process Model Architecture

Chapter 4 presents various methods for camera-based in-process monitoring. One of the methods, shown in Section 4.4, is a deep learning approach using a neural network. The network architecture is explained in more detail in this chapter.

To analyze the images directly in the welding process, an architecture optimized for this application is used. The in-process data is less complex, and the focus is on fast execution time on an edge device. Therefore, the acquired images are reduced to the dimensions 128×128 before the network processes them. The network architecture is based on the SDU-Net but with a reduced capacity. The architecture uses three encoder operations with the number of filters $n_{\text{out}} = \{16, 32, 64\}$ and the corresponding decoder operations in the upsample path. An activation with ELU follows each convolutional layer (Conv). The last decoder operation in the expansive path is followed by a 1×1 convolution and a softmax activation that maps the feature vectors to the number of classes to be learned. This architecture results in a total number of parameters of 39 145.

Figure E.1 illustrates the structure of the architecture. Chapter 4 uses the architecture to detect spatters in a camera image.

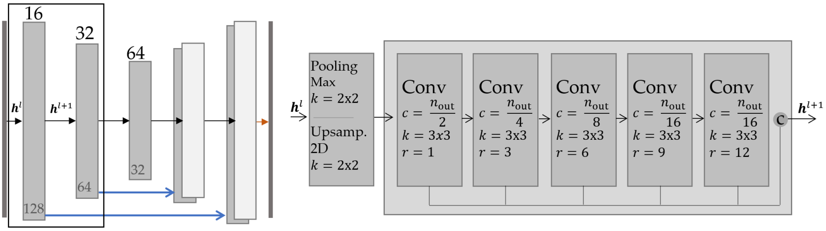

16
32
64
h^l
h^{l+1}
128
64
32
Pooling
Max
$k = 2x2$
Upsamp.
2D
$k = 2x2$
Conv $c = \frac{n_{out}}{2}$ $k = 3x3$ $r = 1$
Conv $c = \frac{n_{out}}{4}$ $k = 3x3$ $r = 3$
Conv $c = \frac{n_{out}}{8}$ $k = 3x3$ $r = 6$
Conv $c = \frac{n_{out}}{16}$ $k = 3x3$ $r = 9$
Conv $c = \frac{n_{out}}{16}$ $k = 3x3$ $r = 12$
C

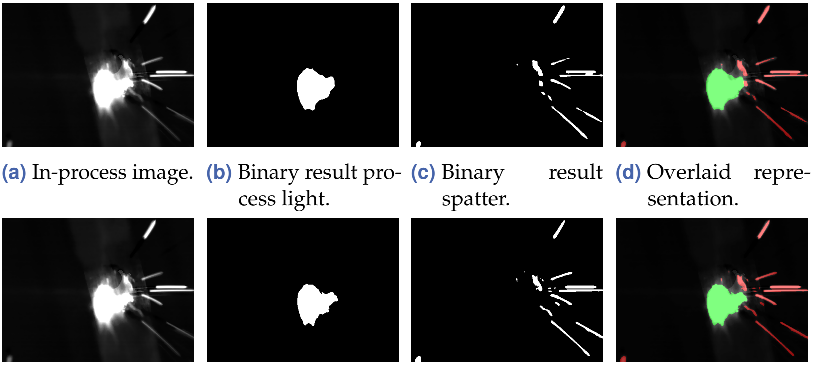
Figure E.1 Small SDU-Net architecture with the adapted encoder and decoder operation. The boxes represent the feature map, with the x- and y-resolution at the bottom of the box and the number of channels at the top of the box. The black arrows between the boxes represent the encoder-/ decoder operations shown next to the architecture. The parameter n_{out} represents the number of channels after concatenating the convolutions' outputs. While the orange arrow represents a convolution with kernel 1 × 1 and a softmax activation to map the feature vector to the desired number of classes, the blue arrows represent the skip connections.

F Spatter Detection Model Comparison

Section 4.4 uses a neural network architecture with extremely few parameters to achieve fast inference time per frame for in-process spatter detection. In this chapter, the architecture is compared to a larger model to show the loss of accuracy.

Figure F.1 compares the one-hot encoded model results with two different SDU-Net models with different capacities. The bottom row shows the prediction of a larger model with a parameter count of 162 457. The model uses the input dimension of 256×256 pixels. As the input in the higher resolution has more features, it uses four encoder operations with $n_{\text{out}} = \{16, 32, 64, 128\}$ and the corresponding decoder operations. The structure of the model architecture is explained in more detail in Section 3.4.1 and Figure 3.6. The top row shows the result of an SDU-Net with lower capacity in comparison. The architecture uses an input dimension of 128×128 and three encoder operations with $n_{\text{out}} = \{16, 32, 64\}$ and the corresponding decoder operations. This results in a total number of parameters of 39 145. A detailed description of the architecture definition is given in Appendix E. The original image resolution is 640×480, which is scaled down to the corresponding input dimensions. Figure F.1 uses a false color representation for the class assignment, where green represents the process light and red represents the pixels assigned to the spatter class. After the model predictions, the result of the CNN is scaled to the original image resolution and then binarized with a threshold of 0.5. The results show that the low model capacity with 39 145 parameters is sufficient to detect the process light and spatters. Since the images are not highly complex, fewer parameters are sufficient for the model to learn the relevant features. Small spatters of a few pixels can be lost by reducing the image dimensions to 128×128 input pixels. However, since

100% monitoring is not possible with this method, this loss is acceptable in return for the faster calculation time.



(a) In-process image. (b) Binary result process light. (c) Binary result spatter. (d) Overlaid representation.

(e) In-process image. (f) Binary result process light. (g) Binary result spatter. (h) Overlaid representation.

Figure F.1 The figure compares the predictions of two SDU-Net models with different capacities. The top row shows the binary one-hot results of the process light class (b) and the spatter class (c) of a lower capacity model with an input dimension of 128 × 128 (Appendix E). The bottom row shows the corresponding results for a larger capacity model (Section 3.4.1) with an input resolution of 256 × 256. Figures (d) and (h) show the results in an overlaid representation of the input image with the spatter class (red) and the process light (green).

G In-Process Acquisition Frequency

Further to Section 4.5, which discusses the influence of image acquisition frequency on spatter analysis, this chapter shows more examples of different welding processes. Figure G.1, G.2 and G.3 compare the captured spatter volume with an acquisition frequency of 2 kHz and 286 Hz. For the comparison, the low recording frequency was simulated. Besides the spatter rating resolved to single frames, the graphs show overlaid image representations with the predicted one-hot classes spatter (red) and process light (green) on the camera image. The graphs show that more significant ejections and extended areas with ejections are also visible at 286 Hz. However, the information about single, small, and especially fast spatters will be lost.

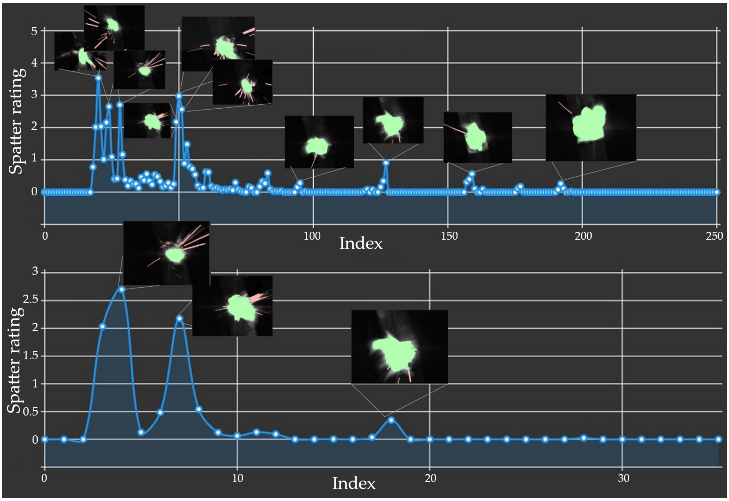


Figure G.1 The upper row shows the spatter rating for a recording frequency of 2 kHz while the lower row simulates a recording frequency of 286 Hz for the same process.

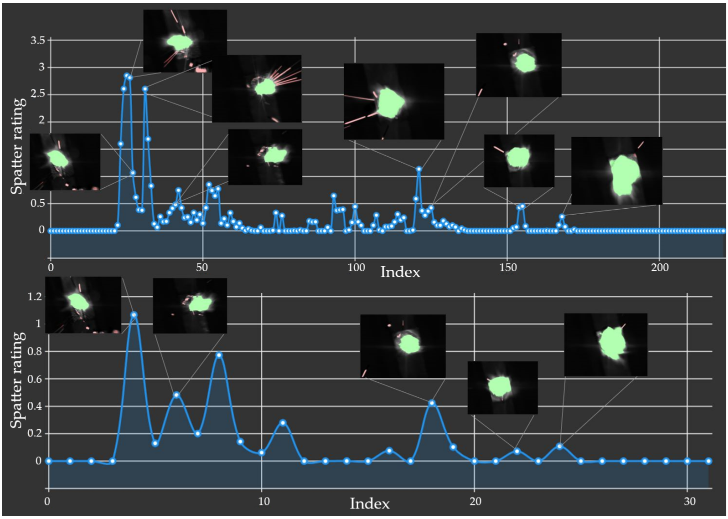


Figure G.2 The upper row shows the spatter rating for a recording frequency of 2 kHz while the lower row simulates a recording frequency of 286 Hz for the same process.

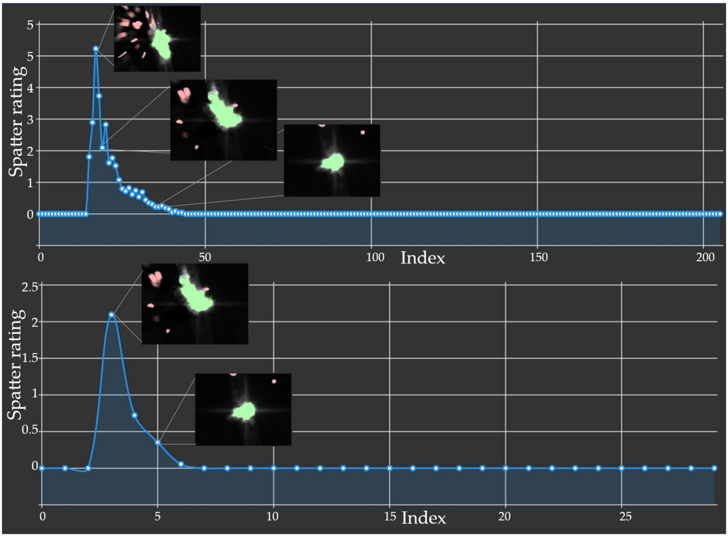


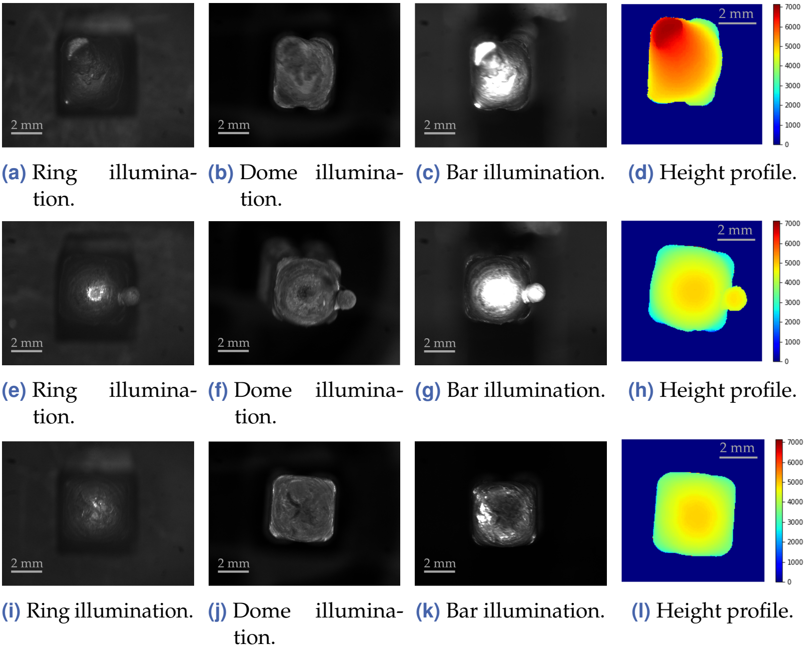
Figure G.3 The upper row shows the spatter rating for a recording frequency of 2 kHz while the lower row simulates a recording frequency of 286 Hz for the same process.

H Illumination Types

Different lighting allows different features of the component to be captured with the same sensor. The copper material of the hairpins is highly reflective. In this work, the standard setup for data acquisition is used to monitor the laser welding process. This uses a ring light attached to the focusing optics for illumination, which has little negative effect on the process. Extending the post-process data acquisition from Chapter 5 in Section 5.3, alternative illumination scenarios are shown in this chapter.

When lighting from above at a certain distance, the camera captures the parallel areas overexposed while other regions are shaded. To avoid overexposed areas, the exposure time must be reduced. Diffuse dome lighting can be used to enhance surface texture. A dome-shaped reflector is placed at the height of the component. The light is directed into the reflector, and from there, it is diffused onto the component from all directions. The bar lighting in the example in Figure H.1 was mounted from diagonally above slightly over the top of the weld. The lighting consists of four bar spotlights that illuminate the part from each side. However, the directional light causes reflections on the component. But, because it has been placed closer to the edge, it highlights the elevation in H.1(g), for example. The offset in Figure H.1(c) can also be imagined due to the shading of the right pin area.

Figure H.1 also shows the height profile of the pin pairs. This helps to identify better which structures are visible in the camera image. In particular, a height offset is difficult to see in the camera image taken from above.



(a) Ring illumination. (b) Dome illumination. (c) Bar illumination. (d) Height profile.

(e) Ring illumination. (f) Dome illumination. (g) Bar illumination. (h) Height profile.

(i) Ring illumination. (j) Dome illumination. (k) Bar illumination. (l) Height profile.

Figure H.1 Comparison of the different databases. The pictures show three different pairs of copper pins captured with different illuminations. Figure (d), (h) and (l) show the height profile in comparison. The upper line shows the images of two misaligned pins, the middle line shows a pin pair on which a spatter has settled, and the lower line shows a pin pair welded with less laser power. The data per line show the same component but is not precisely in the same orientation.

I Intensity Image and Height Data Mapping

In Section 5.3, prominent image features are used to determine the offset between the 2D intensity images and the 3D height maps. Since the pin's surface is in the image's foreground, it is used for mapping.

However, the corresponding pixel values in both images do not necessarily match. In addition, both the intensity image and the height map contain interfering structures and characteristics that make it difficult to assign the contents. Therefore, binary masks of the pin surface are created using semantic segmentation. For this purpose, two separate models are trained. One is trained on the intensity images, and a second model is trained on the height maps. The model uses an SDU-Net architecture trained with categorical focal loss and optimized with ADAM. The structure is analogous to the model defined in Section 3.4.1 in Figure 3.6. Only the size is adapted to the original image size of 432×432 pixels. As a result, the one-hot encoded class map of the foreground class is used for mapping. The masks represent the background class by 0 (black), while the pixels associated with the weld seam class are represented by 1 (white). Since the same seam surface is detected in both binary representations, the offset and rotation can be calculated using these binary masks. Subsequently, the calculated values are transferred to the intensity image and the height map.

The binary images are highly simplified and contain only two values for pin area and background, so a simplified procedure is used to determine the translation. First, the center of gravity of the pixels which are assigned to the pin class is determined. This point is moved to the center of the image, which removes the translation. In addition, this point represents the rotation center from which the rotation deviation is calculated. For this purpose, the images are translated into polar coordinates starting from the center point. Based on the distance and the angle of the surface to the center point, the offset of the two masks to each other can be determined. For this purpose, based on the representation in polar coordinates, the translation of the samples is determined by their correlation. From the translation of the polar coordinates in y-direction t_y, the rotation angle α_a can be concluded as follows

$$\alpha_a = \frac{t_y}{\mathrm{rows}} \cdot 360^\circ, \tag{I.1}$$

where rows is the image height. In a post-processing step, the translation is corrected using the correlation of the intensity image's rotated mask to the height map's original mask.

J Implementation Details 3D Reconstruction Models

Section 5.4.3 compares different ML methods used for single-view 3D reconstruction for post-process monitoring. The detailed implementation of these methods is explained in the following chapter.

J.1 Stacked Autoencoder

Two separately trained autoencoders are combined to adapt the autoencoder structure to the task of 3D reconstruction. First, two AEs are trained on the respective data to realize a low-dimensional feature extraction in the latent space. Then, the whole network to predict the height data based on the camera images is created by linking the encoder and decoder subspaces of the two networks. In addition, a mapping layer is added between the subspaces, which is post-trained on the mapping function.

The input and output resolution of the network is 256×256 pixels, where the images are scaled to a value range of $[0, 1]$. Larger resolutions are problematic because of the fully connected layers. Due to the many parameters, the network then reaches the memory limits of the GPU. The exact layer structure of the SAE is 256×256-1000-100 for the encoder and 100-2000-256×256 for the decoder in fully connected layers. A layer with 5000 neurons is used to map the subspaces, which connects the subspaces fully connected to each other. The layers were trained separately. While the inner layers have activation with ReLU, the activation function at the output of the last layer is a sigmoid function. The sigmoid function scales the output data in the range $[0, 1]$. Since this also corresponds to the training data range, this helps stabilize the learning process.

The training uses the MSE as the loss function. In addition it uses an ADAM optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of $\epsilon = 0.0001$. The first layer of the encoder and the decoder are trained for 4000 epochs. In contrast, the deeper layers are trained for 800 epochs since the number of parameters of the network to be optimized is smaller. The training is performed with 800 steps per epoch and batch size $BS = 1$.

J.2 Generative Adversarial Networks

The CGAN architecture is suitable for 3D reconstruction because an intensity image is given as an input condition for generating a realistic height map in addition to the noise vector. A GAN always consists of a generator, a discriminator, and a loss function. However, there are many configurations to define the components.

In this work, four different configurations are analyzed. Table J.1 shows an overview of the configurations. The input dimension of the generator and discriminator, as well as the output dimension of the generator is 256×256 pixels. Analogous to the work of Isola et al. [72], the images are scaled to a value range of $[-1, 1]$.

Table J.1 Configurations of GANs for 3D reconstruction.

Configuration	Generator	Discriminator	Loss function
I	U-Net	PatchGAN	CGAN + L_1
II	U-Net	DCGAN	WGAN + L_1
III	U-Net	DCGAN	WGAN + L_2
IV	SDU-Net	DCGAN	WGAN + L_2

The generator of the first three configurations uses the modified U-Net structure according to Isola et al. [72]. Figure J.1 shows the structure of the model. The boxes represent the initial dimensions of the successive convolutions. Each convolution is performed using a kernel with $k = 4 \times 4$ and a step size $s = 2$. In the encoder, the convolution decreases the dimension by a factor of 2, while the dimensions in the decoder are increased. Except for the first convolution layer, batch normalization is performed after each convolution. While the encoder uses activation with leaky ReLU with $\alpha = 0.3$, the decoder uses the ReLU function. After the last decoder layer, convolution with $k = 4 \times 4$ and $s = 2$ is performed with the number of filters $n = 1$ to obtain the output image. In this layer, the model uses the Tanh function for activation. In addition, the skip connections corresponding to the U-Net are included between the encoder and decoder layers. So the features are transferred from the encoder directly to the decoder. These are shown as blue arrows in the graphic.

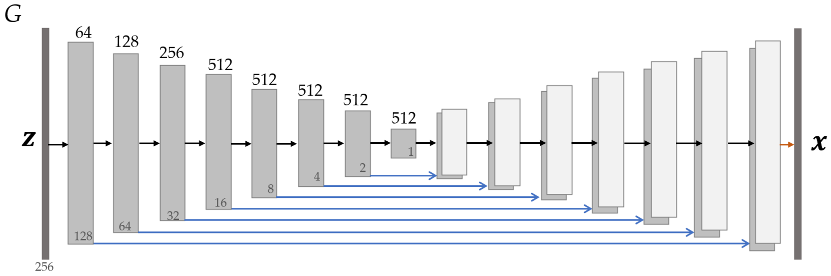


Figure J.1 The figure shows the architecture of the generator of the various GAN configurations. The boxes represent the output dimensions of the successive convolutions. At the bottom of the box, the x-y-resolution is shown, while the number of filters is denoted on top of the box. The blue arrows denote skip connections. The orange arrow represents a 4×4 convolution with $s = 2$, the number of filters $n = 1$, and a Tanh activation.

The configuration I uses the PatchGAN according to Isola et al. [72] as a discriminator. The PatchGAN has the characteristic that the input image is divided into smaller patches, for each of which a prediction is made in real and fake. In the architecture used, a patch prediction corresponds to an overlapping range of 70×70 patches in the input image. The Figure J.2 illustrates the layers of the discriminator network. The first three convolutions have a kernel $k = 4 \times 4$ and $s = 2$. In contrast, the last two convolutions are performed with $s = 1$, meaning the dimensions are no longer halved. Zero padding is applied before these convolutions. Batch normalization follows all convolutions except the first one. Each fold is followed by activation with the leakyReLU with $\alpha = 0.3$. The sigmoid activation normalizes the last convolution to the range $[0, 1]$.

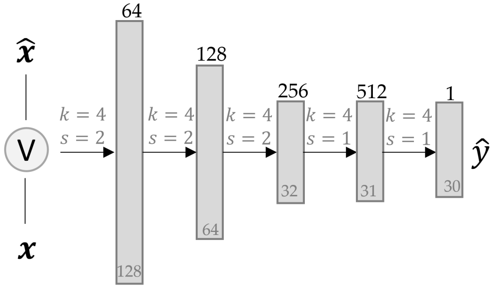


Figure J.2 The architecture of the discriminator of the PatchGAN. The arrows represent convolutional layers with corresponding kernel size k and step size s.

The II, III, and IV configurations use a deep generative convolutional networks (DCGAN) in combination with the Wasserstein distance [127]. The network is similar to the discriminator but without a sigmoid function in the last layer. This results in a scalar score as output instead of a probability score. Since the network thus cannot distinguish between real and fake using a threshold, it is referred to as a critic instead of a discriminator. The Wasserstein distance for the data distribution p and q is defined by:

$$W(p,q) = \underset{\gamma \in \pi(p,g)}{inf} \; (\mathbb{E}_{(x,y)\sim\gamma}[||x-y||], \tag{J.1}$$

where $\pi(p,g)$ is the set of all joint distributions $\gamma(x,y)$ whose marginals are p and q. As a cost function, the Wasserstein distance has the advantage over the Jensen-Shannon divergence of producing smoother gradients. For this reason, the gradient for the generator does not decrease as much when the generator is not yet performing well. As a result, the generator gets more information to improve its performance. This means that the WGAN learns whether or not the generator is already giving good results. To calculate the Wasserstein distance, the 1-Lipschitz constraint is used, which is achieved by clipping the weights with the hyperparameter c_{WD} to limit the maximum weight value. Figure J.3 shows the architecture structure. It consists of two convolution operations with $k = 5 \times 5$ and $s = 2$. Both convolutions are followed by a leaky ReLU with $\alpha = 0.3$ and a dropout layer with a dropout probability of $d_p = 0.3$. Finally, the output of the last layer is flattened and mapped to a one-dimensional output via a fully connected layer.

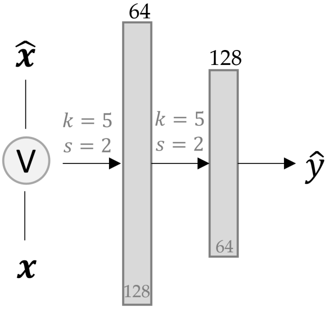


Figure J.3 The architecture of the critic of the DCGAN. The arrows represent convolutional layers with corresponding kernel size k and step size s. After the second convolutions the output mapped to a one-dimensional output via a fully connected layer.

Configuration IV is analogous to III except that a variant of the SDU-Net is used as the generator instead of the U-Net architecture from Figure J.1. Figure J.4 shows the structure of the architecture. The exact definition of the architecture is specified in Section 3.4.1. Unlike the generator from III, the model uses a kernel size of $k = 3 \times 3$, a step size $s = 1$, and adds pooling or upsampling layers with a kernel size of $k = 2 \times 2$. Each convolution is followed by activation with ELU. In contrast to the structure in Section 3.4.1, the last decoder operation in the expansive path is followed by a 1×1 convolution with the number of filters $n = 1$ and a Tanh activation, which maps the features to the output layer. The discriminator and loss function of configuration IV are similar to the definition in configuration III.

The standard approach from Goodfellow et al. [57] is used to train the networks. The training uses an ADAM optimizer with a learning rate $\epsilon = 0.0001$ and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. All networks are trained from scratch. A normal distribution with a mean of 0 and a standard deviation of 0.02 is used to initialize the weights. The networks are trained twice for 500 000 iterations with batch size $BS = 1$. The loss function uses the weighting $\alpha = 100$ and the L_1 or L_2 distance measure, depending on the configuration. The weight clipping factor of the WGAN is defined with $c_{WD} = 0.01$.

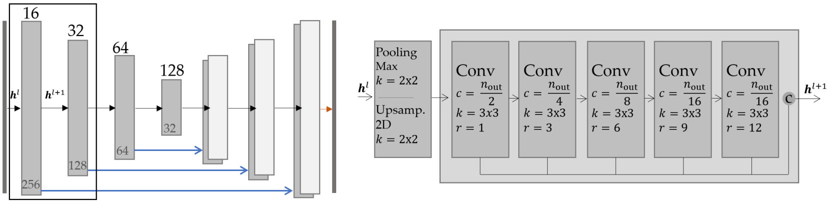


Figure J.4 The architecture of the generator of the GAN IV configuration. The boxes represent the output dimensions of the successive convolutions. While the x-y-resolution is provided at the bottom of the box, the number of filters is denoted on top of the box. The blue arrows denote skip connections. The orange arrow represents a 1 × 1 convolution with the number of filters $n = 1$ and a Tanh activation.

J.3 U-Net

To adapt the U-Net architecture from semantic segmentation to 3D reconstruction, a height value is learned instead of assigning a class value. In training, the focal loss is replaced by the MSE as loss function. The ADAM optimizer is still used with the momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate starts at $\epsilon = 0.001$ and is reduced during training. In addition, data augmentation with rotation, translation, mirroring, zoom, and shear is applied to the training data. The missing pixels at the edge resulting from the transformation are supplemented with a nearest-neighbor algorithm.

Two different architectural definitions are used. The first is based on the U-Net of Ronneberger et al. [134], while the second is based on the adapted SDU-Net of Wang et al. [166]. The increase of the receptive field due to the dilated convolutions also shows an advantage in 3D reconstruction. Both network architectures use a convolution with kernel size 1×1 in the last layer and sigmoid activation to scale the output to the range $[0, 1]$. Within the network structures, a ReLU activation is used.

Figure J.5 and J.6 represent the structure of the respective architecture. In the U-Net architecture from Figure J.5, batch normalization follows each convolutional layer to stabilize the model. In addition, in the encoder path, each max-pooling is followed by a dropout with a rate of $d_p = 0.1$. No dropout is applied in the decoder path. Upsampling is performed in this model by transposed convolution. In this type of enlargement, the kernel values used to increase the dimension are optimized in training.

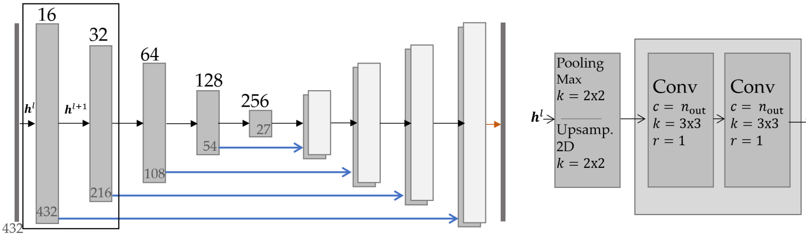


Figure J.5 The architecture of the U-Net used for 3D reconstruction. The boxes represent the feature map, with the x- and y-resolution at the bottom of the box and the number of channels at the top of the box. The black arrows between the boxes represent the encoder-/ decoder operations shown next to the architecture. The parameter n_{out} indicates the number of feature maps resulting from the operation, k the kernel size, and r the dilation rate. The orange arrow represents a convolution with kernel 1×1 and a sigmoid activation to map the features vector to the corresponding height value in the range $[0, 1]$. The blue arrows represent the skip connections.

The structure of the SDU-Net in Figure J.6 is based on the network archtiture from Section 3.4.1, which is also used for semantic segmentation.

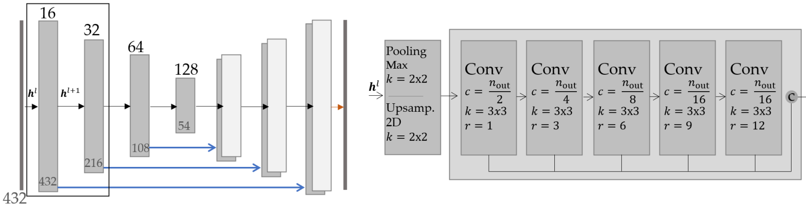


Figure J.6 The architecture of the SDU-Net used for 3D reconstruction. n_{out} represents the number of channels after concatenating the convolutions' outputs. The dilation rate r increases while the kernel size k remains unchanged.

K 3D Reconstruction Visualization

Continuing the analysis of the different ML 3D reconstruction algorithms presented in Section 5.4.3, this chapter visualizes reconstruction results for the different methods. Figure K.1 shows the reconstruction results with the different ML-based methods for various samples. Samples with different seam defects are selected to illustrate the model performances better. The examples which show defective welds are generally less represented in the training data set. The top row of each example shows the reconstruction results with the respective methods. The bottom row shows the ground truth meaured by OCT in the first column. Otherwise, it shows the absolute error of the reconstruction compared to the measured values per pixel. It must be noted that a different scaling is used for the false color representation of the absolute error values. This scaling seres as a better representation of the lower height values. The corresponding scalings are shown at the bottom of the figure.

The first example shows a good weld. This is the most common case in the training data, which is why most algorithms give quite a good result for this example. The second example shows a pin pair that was not in the laser's focus and, thus, does not have a sufficiently formed weld bead. The third example is a pin pair with an offset, visible by the elevation in the upper left corner. The pin pair in the fourth example has a lateral offset, while the hairpins in the fifth were welded without stripping. Finally, the last row shows a weld where too much laser power was used, making the weld misshapen and too wide. The generated images of the SAE show very blurred edges, indicating a loss of information in the model. Of the GAN methods, configuration I shows the largest deviation compared to the ground truth, while the other methods perform better. Especially in data that occur very little in training (second bottom row), there are more significant deviations. The U-Net approaches generally perform best. Even the SDU-Net, which is only trained on 10% instead of 80% of the data, shows small deviations.

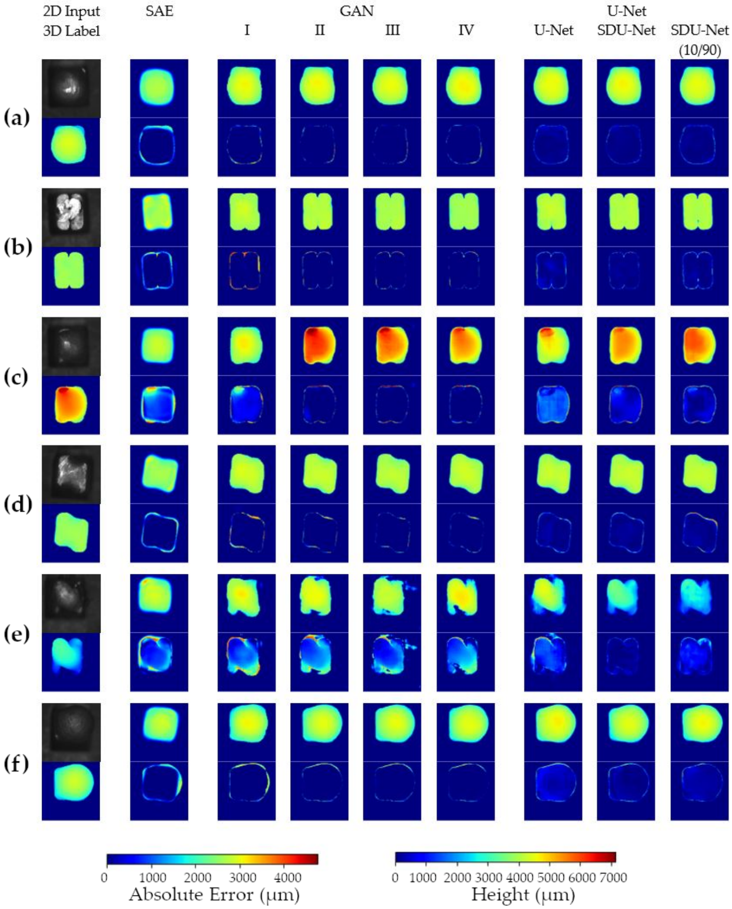


Figure K.1 3D reconstruction results of test images: (a) good weld, (b) pin pair not in the focus of the laser, (c) height offset of the copper rods, (d) lateral offset of the copper rods, (e) isolated copper rods, (f) too much power. The top row of each example shows the reconstruction results with the respective methods. The bottom row shows in the first column the ground truth measured by OCT and then the absolute error of the reconstruction to the measured values per pixel.

L 3D Reconstruction with Less Training Data

Complementing the analysis from Section 5.4.3, the following chapter examines the impact of the size of the training dataset on the ML approach using SDU-Net for 3D reconstruction.

Since applications in industrial manufacturing usually require working with a very small data set, further experiments are conducted with a reduced size of the training data. Based on the best performance and the smallest number of model parameters, the SDU-Net configuration is used for the experiments. Its performance is checked when the number of training samples is reduced to 60%, 40%, 20%, and 10% of the available data. The network and training parameters are defined similarly for the U-Net II configuration. The results were obtained by 20 different random training-test splits. Table L.1 shows the averaged results.

Table L.1 Mean MAE of the 3D reconstruction algorithm SDU-Net with a reduced number of training data. Different proportions of the data are used in the training part. In addition to the MAE, its standard deviation (SD) is calculated to indicate the dispersion in the test samples. The table shows the averaged values of 20 random train-test splits each.

n_{train} (%)	n_{test} (%)	n_{train}	n_{test}	MAE (µm)	MAE (%)	SD (µm)	SD (%)
80	20	762	191	78.8	1.126	37.2	0.532
60	40	572	381	81.1	1.158	48.1	0.687
40	60	381	572	81.0	1.157	47.1	0.673
20	80	191	762	86.9	1.242	52.2	0.745
10	90	95	858	93.5	1.336	68.7	0.981

The results show that the performance of SDU-Net decreases when the number of training data is reduced. For example, reducing the size of the training data set from 80% to 10% of the available data degrades the average MAE on the test set by 14.7 µm. However, all results are still better than the GANs and SAE, whose results are shown in Section 5.4.3 in Table 5.2. With the best GAN method, the evaluation achieves an average MAE of 142.2 µm on 762 training samples and 237.3 µm with the SAE.

Furthermore, the variance of the average MAE within each training-test proportion shown in Figure L.1 suggests that the composition of the training set has an impact on the model performance. Pins from different defect classes have different geometries and height profiles. If only the features of very similar parts are learned, the reconstruction of divergent geometries may become inaccurate. Therefore, random splits lead to a worse average result than a representative training data set. An unbalanced training data set can also explain poor performance with less training data. Using fewer data makes it more challenging to capture all of the variances. This increases the average MAE and standard deviation in tests since unknown geometries cannot be calculated accurately.

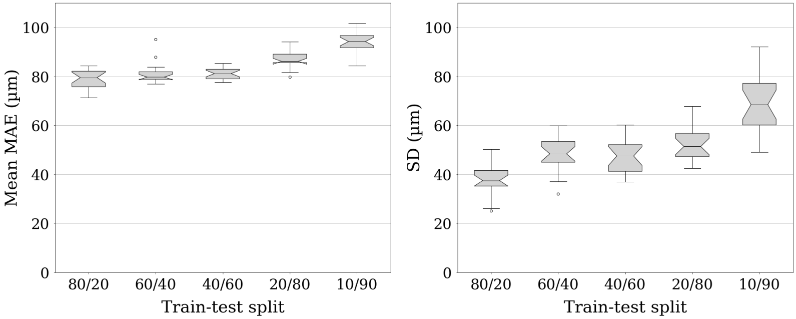


Figure L.1 Distribution of the mean MAE and SD of the 3D reconstruction algorithm for different train-test splits. Validation of the SDU-Net performance under different proportions of data in the training and testing data set for 20 random splits each.

M Post-Process Image Classification

Extending the hybrid approaches that combine AI and rule-based algorithms for postprocessing monitoring, the evaluation and discussion in Section 5.6 compares the methods with a regression model that directly predicts the division into GW and DW for the image data. The following chapter explains this approach in more detail.

This approach teaches a small CNN architecture on the two-class problem, resulting in a probability for a class label. The image-wise categorization into good and bad data reduces the labeling effort compared to semantic segmentation, where pixel-accurate labeling is necessary. As in the other approaches in Chapter 5, a training data set $\mathbb{X}^{\text{train}}$ with $n = 95$ is used. In addition, a weaker form of data augmentation with rotation, flip, and a slight shift is applied. Augmentations such as zoom or distortion could change the associated class score since, for example, very large welds should be classified as DW. The ratio of GW and DW is unbalanced in the training data set. There are 18 DW and 77 GW. Class weighting is used to handle the imbalance. Moreover, the training uses the focal loss focusing on the worse assigned samples by the parameter $\gamma = 2$. In addition, due to the low data availability, different regularization algorithms are used, such as dropout, spatial dropout, or L_2 regularization. The task can be extended to other classes, such as specific fault categories like misalignment, gap, or too much laser power. However, this requires that training data is also available for all defined classes.

Table M.1 shows the classification compared to the results of the semantic segmentation from Section 5.4.2 and the 3D reconstruction from Section 5.4.3 with confusion matrices.

The result of the regression model classification is the worst compared to the other methods. Early stopping based on the validation data set achieves a validation accuracy of 91.14% in the best case. This is 76 erroneous predictions out of 858 samples. Among them, there are 62

Table M.1 Confusion matrices to compare the results of the different methods. The results of the approaches: Weld shape extracted from the camera image (WS), AI-based 3D reconstruction (3D-R), and image classification (IC) are compared with ground truth based on the features from the entire height map (OCT).

	WS GW	WS DW
OCT GW	679	20
OCT DW	25	134

	3D-R GW	3D-R DW
OCT GW	694	5
OCT DW	11	148

	IC GW	IC DW
OCT GW	685	14
OCT DW	62	97

DW that were classified as GW (false positives). In the training process, after a short time, only the training error improves, while the validation error increases. This behavior indicates overfitting to the training data and poor generalization performance of the model, despite the use of regularization methods. By using additional training data, the result can be significantly improved. Vater et al. [162] achieve an accuracy of 99% in a classification of image data of hairpin welds. They use a training data set $\mathbb{X}^{\text{train}}$ with $n = 1827$ and $\mathbb{X}^{\text{test}}$ with $n = 457$ and four clearly defined classes. In the previously shown example, the performance can also be increased by adjusting the training-test split. Using $\mathbb{X}^{\text{train}}$ with $n = 500$ and $\mathbb{X}^{\text{test}}$ with $n = 453$ improve the accuracy from 91.14% to $\approx 95\%$.

Bibliography

[1] **Agarap, A. F.** *Deep Learning using Rectified Linear Units (ReLU)*. 2019. arXiv: 1803.08375 [cs.NE].

[2] **Alter, L., Heider, A., and Bergmann, J.-P.** *Investigations on copper welding using a frequency-doubled disk laser and high welding speeds*. In: *CIRP Conference on Photonic Technologies (LANE)*. Vol. 74. 2018, pp. 12–16.

[3] **Arjovsky, M., Chintala, S., and Bottou, L.** *Wasserstein Generative Adversarial Networks*. In: *International Conference on Machine Learning (ICML)*. Vol. 70. 2017, pp. 214–223.

[4] **Arslan, A. T. and Seke, E.** *Face Depth Estimation With Conditional Generative Adversarial Networks*. In: *IEEE Access* 7 (2019), pp. 23222–23231.

[5] **Arslan, A. T. and Seke, E.** *Training Wasserstein GANs for Estimating Depth Maps*. In: *IEEE International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 2019, pp. 1–4.

[6] **Baader, M., Mayr, A., Raffin, T., Selzam, J., Kühl, A., and Franke, J.** *Potentials of Optical Coherence Tomography for Process Monitoring in Laser Welding of Hairpin Windings*. In: *International Electric Drives Production Conference (EDPC)*. 2021, pp. 1–10.

[7] **Baby, A. T., Andrews, A., Dinesh, A., Joseph, A., and Anjusree, V.** *Face Depth Estimation and 3D Reconstruction*. In: *IEEE Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*. Vol. 7. IEEE, 2020, pp. 125–132.

[8] **Baldi, P.** *Autoencoders, Unsupervised Learning and Deep Architectures*. In: *International Conference on Machine Learning (ICML) Workshop on Unsupervised and Transfer Learning*. UTLW. JMLR, 2011, pp. 37–50.

[9] **Baumer GmbH**. *Industriekameras VCXG*. URL: https://www.baumer.com/de/de/p/36662 (visited on 02/26/2023).

[10] **Bengio, Y., Mori, R. D., Flammia, G., and Kompe, R.** *Phonetically motivated acoustic parameters for continuous speech recognition using artificial neural networks*. In: *Speech Communication* 11 (1991), pp. 261–271.

[11] **Bengio, Y., Louradour, J., Collobert, R., and Weston, J.** *Curriculum learning*. In: *International Conference on Machine Learning (ICML)*. 2009, pp. 41–48.

[12] **Bitkom e.V.** *KI gilt in der deutschen Wirtschaft als Zukunftstechnologie – wird aber selten genutzt*. 2022. URL: https://www.bitkom.org/Presse/Presseinformation/Kuenstliche-Intelligenz-2022 (visited on 12/15/2022).

[13] **Bocksrocker, O., Speker, N., Beranek, M., and Hesse, T.** *Reduction of spatters an pores in laser welding of copper hairpins using two superimposed laser beams*. In: *Lasers in Manufacturing Conference (LIM)*. 2019, pp. 1–8.

[14] **Bohr, N.** *On the Constitution of Atoms and Molecules*. In: *Philosophical Magazine* Volume 6 (1913), pp. 1–25.

[15] **Bourlard, H. and Kamp, Y.** *Auto-Association by Multilayer Perceptrons and Singular Value Decomposition*. In: *Biological Cybernetics* 59 (1988), pp. 291–294.

[16] **Breiman, L.** *Bagging predictors*. In: *Machine Learning* 24 (1994), pp. 123–140.

[17] **Cai, H. and Xiao, R.** *Comparison of spatter characteristics in fiber and CO2 laser beam welding of aluminum alloy*. In: *International Congress on Applications of Lasers and Electro-Optics (ICALEO)*. 2011, pp. 150–158.

[18] **Chellapilla, K., Puri, S., and Simard, P.** *High Performance Convolutional Neural Networks for Document Processing*. In: *International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006.

[19] **Chen, J. and Ran, X.** *Deep Learning With Edge Computing: A Review*. In: *Proceedings of the IEEE* 107.8 (2019), pp. 1655–1674.

[20] **Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S.** *3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction*. In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 628–644.

[21] **Cireşan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J.** *Deep, Big, Simple Neural Nets for Handwritten Digit Recognition*. In: *Neural Computation* 22.12 (2010), pp. 3207–3220.

[22] **Cleemann, L. and Beyer, E.** *Schweißen mit CO2-Hochleistungslasern*. In: *VDI-Handbuch*. Düsseldorf: DI-Verlag, 1987.

[23] **Clevert, D.-A., Unterthiner, T., and Hochreiter, S.** *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. In: *International Conference on Learning Representations (ICLR) (Poster)*. 2016.

[24] **Coates, A. and Ng, A.** *The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization*. In: *International Conference on Machine Learning (ICML)*. 2011, pp. 921–928.

[25] **Coates, A., Huval, B., Wang, T., Wu, D. J., Catanzaro, B., and Ng, A.** *Deep learning with COTS HPC systems*. In: *International Conference on Machine Learning (ICML)*. 2013, pp. 1337–1345.

[26] **Devalla, S. K., Renukanand, P. K., Sreedhar, B. K., Perera, S., Mari, J.-M., Chin, K. S., Tun, T. A., Strouthidis, N. G., Aung, T., Thiery, A. H., and Girard, M. J. A.** *DRUNET: A Dilated-Residual U-Net Deep Learning Network to Digitally Stain Optic Nerve Head Tissues in Optical Coherence Tomography Images*. In: *Biomed Opt Express* 9 (2018), pp. 3244–3265.

[27] **DeVries, T. and Taylor, G. W.** *Leveraging Uncertainty Estimates for Predicting Segmentation Quality*. In: *Conference on Medical Imaging with Deep Learning (MIDL)*. 2018.

[28] **Deyneka-Dupriez, N.** *Implementing OCT for industrial weld monitoring*. 2019. URL: https://www.lasersystemseurope.com/analysis-opinion/implementing-oct-industrial-weld-monitoring (visited on 01/16/2023).

[29] **Dinham, M. and Fang, G.** *Autonomous weld seam identification and localisation using eye-in-hand stereo vision for robotic arc welding*. In: *Robotics and Computer-Integrated Manufacturing* 29.5 (2013), pp. 288–301.

[30] **Dinham, M., Fang, G., and Zou, J. J.** *Experiments on Automatic Seam Detection for a MIG Welding Robot*. In: *International Conference on Artificial Intelligence and Computational Intelligence (AICI)*. 2011, pp. 390–397.

[31] **Dmitry, R., Alexander, L., and Valery, P.** *Development of Mechanisms for Automatic Correction of Industrial Complex Tools in the Preprocessing of Laser Welding for Small-Scale and Piece Production Using Computer Vision*. In: *Machines* 8.4 (2020), pp. 86.1–86.18.

[32] **Dold, E.-M., Willmes, A., Kaiser, E., Pricking S.and Killi, A., and Zaske, S.** *Qualitativ hochwertige Kupferschweißungen durch grüne Hochleistungsdauerstrichlaser*. In: *Fachzeitschrift für Metallurgie - Metall* 11 (2018), pp. 457–459.

[33] **Dold, E.-M., Kaiser, E., Klausmann, K., Pricking, S., Zaske, S., and Brockmann, R.** *High-performance welding of copper with green multi-kW continuous-wave disk lasers*. In: *High-Power Laser Materials Processing: Applications, Diagnostics, and Systems VIII*. Ed. by **Kaierle, S. and Heinemann, S. W.** Vol. 10911. International Society for Optics and Photonics. SPIE, 2019, pp. 28–33.

[34] **Dössel, O.** *Bildgebende Verfahren in der Medizin: Von der Technik zur medizinischen Anwendung*. Berlin Heidelberg: Springer, 2016.

[35] **Duchi, J., Hazan, E., and Singer, Y.** *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*. In: *Journal of Machine Learning Research* 12 (2011), pp. 2121–2159.

[36] **Dudeck, S. G.** *Kamerabasierte In-situ-Überwachung gepulster Laserschweißprozesse*. PhD thesis. Karlsruher Institut für Technologie, Institut für Industrielle Informationstechnik (IIIT), Karlsruhe, 2013. 260 pp.

[37] **Eichler, H. J. and Eichler, J.** *Absorption und Emission von Licht*. In: *Laser: Bauformen, Strahlführung, Anwendungen*. Berlin Heidelberg: Springer, 2010, pp. 29–53.

[38] **Einstein, A.** *Zur Quantentheorie der Strahlung*. In: *Physikalische Zeitschrift* 18 (1917), pp. 121–128.

[39] **Engler, S., Ramsayer, R., and Poprawe, R.** *Process Studies on Laser Welding of Copper with Brilliant Green and Infrared Lasers*. In: *Lasers in Manufacturing - International WLT Conference on Lasers in Manufacturing*. Vol. 12. 2011, pp. 339–346.

[40] **Europäische Kommission**. *Verordnung des Europäischen Parlaments und des Rates zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz (Gesetz über künstliche Intelligenz) und zur Änderung bestimmter Rechtsakte der Union*. 2022. URL: https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX%3A52021PC0206 (visited on 12/15/2022).

[41] **Fabbro, R., Slimani, S., Doudet, I., Frederic, C., and Briand, F.** *Experimental study of the dynamical coupling between the induced vapour plume and the melt pool for Nd–Yag CW laser welding*. In: *Journal of Physics D: Applied Physics* 39 (2006), pp. 394–400.

[42] **Feng, D., Wei, X., Rosenbaum, L., Maki, A., and Dietmayer, K.** *Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector*. In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019.

[43] **Fischer, S., Koch, T., Habenicht, T., and Becker, A.** *Künstliche Intelligenz (KI) in der Industrie – ein kurzer Überblick*. Ed. by **Bundesministerium für Wirtschaft und Klimaschutz (BMWK)**. 2022. URL: https://www.bmwk.de/Redaktion/DE/Publikationen/Industrie/ki-in-der-industrie.pdf?__blob=publicationFile&v=4 (visited on 12/15/2022).

[44] **Franco, D., Oliveira, J., Santos, T. G., and Miranda, R.** *Analysis of copper sheets welded by fiber laser with beam oscillation*. In: *Optics and Laser Technology* 133 (2021). Article id 106563.

[45] **Fujinaga, S., Takenaka, H., Narikiyo, T., Katayama, S., and Matsunawa, A.** *Direct observation of keyhole behaviour during pulse modulated high-power Nd:YAG laser irradiation*. In: *Journal of Physics D: Applied Physics* 33 (2000), pp. 492–497.

[46] **Fukushima, K.** *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*. In: *Biological Cybernetics* (1980), pp. 193–202.

[47] **Gal, Y. and Ghahramani, Z.** *Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference*. In: *International Conference on Learning Representations (ICLR) (Workshop)*. 2016.

[48] **Gal, Y. and Ghahramani, Z.** *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. In: *International Conference on Machine Learning*. Vol. 48. 2016, pp. 1050–1059.

[49] **Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D.** *Event-Based Vision: A Survey*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 154–180.

[50] **Gao, X., Sun, Y., and Katayama, S.** *Neural Network of Plume and Spatter for Monitoring High-Power Disk Laser Welding*. In: *International Journal of Precision Engineering and Manufacturing-Green Technology* 1 (2014), pp. 293–298.

[51] **Gao, X., Wen, Q., and Katayama, S.** *Analysis of high-power disk laser welding stability based on classification of plume and spatter characteristics*. In: *Nonferrous Metals Society of China*. Vol. 23. 12. 2013, pp. 3748–3757.

[52] **Gauger, I., Nagel, T., and Huber, M.** *Hybrides Maschinelles Lernen im Kontext der Produktion*. In: *Digitalisierung souverän gestalten II*. Ed. by **Hartmann, E. A.** Berlin: Springer Nature, 2022, pp. 64–79.

[53] **Glässel, T.** *Prozessketten zum Laserstrahlschweißen von flachleiterbasierten Formspulenwicklungen für automobile Traktionsantriebe*. PhD thesis. Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS), Erlangen, 2020.

[54] **Glässel, T., Seefried, J., and Franke, J.** *Challenges in the manufacturing of hairpin windings and application opportunities of infrared lasers for the contacting process*. In: *International Electric Drives Production Conference (EDPC)*. 2017, pp. 1–7.

[55] **Glässel, T., Seefried, J., Masuch, M., Riedel, A., Mayr, A., Kuehl, A., and Franke, J.** *Process Reliable Laser Welding of Hairpin Windings for Automotive Traction Drives*. In: *International Conference on Engineering, Science, and Industrial Applications (ICESI)*. 2019, pp. 1–6.

[56] **Goodfellow, I., Bengio, Y., and Courville, A.** *Deep Learning*. Massachusetts: MIT Press, 2016.

[57] **Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.** *Generative Adversarial Networks*. In: *Conference on Neural Information Processing Systems (NIPS)*. Vol. 27. 2014, pp. 2672–2680.

[58] **Gorriz, M., Carlier, A., Faure, E., and Giro-i-Nieto, X.** *Cost-Effective Active Learning for Melanoma Segmentation*. In: *Conference on Neural Information Processing Systems (NIPS) (Workshop)*. 2017.

[59] **Graves, A.** *Practical Variational Inference for Neural Networks*. In: *Conference on Neural Information Processing Systems (NIPS)*. Vol. 24. 2011, pp. 2348–2356.

[60] **Haid, E.** *100-Prozent-Kontrolle für das Laserschweißen von Hairpins*. 2022. URL: https://www.blechnet.com/100-prozent-kontrolle-fuer-das-laserschweissen-von-hairpins-a-ee2c4346f90e5b8abd5d1941b93ab363/1 (visited on 08/10/2022).

[61] **Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., and Hikosaka, S.** *Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery*. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1442–1450.

[62] **Helm, J., Schulz, A., Olowinsky, A., Dohrn, A., and Poprawe, R.** *Laser welding of laser-structured copper connectors for battery applications and power electronics*. In: *Welding in the World* 64 (2020), pp. 611–622.

[63] **Hinton, G., Srivastava, N., and Swersky, K.** *Lecture: Neural networks for machine learning*. 2012. URL: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (visited on 02/11/2023).

[64] **Hinton, G. E., Osindero, S., and Teh, Y. W.** *A Fast Learning Algorithm for Deep Belief Nets*. In: *Neural Computation* 18 (2006), pp. 1527–1554.

[65] **Hirsch-Kreinsen, H., Kubach, U., Stark, R., Wichert, G. von, Litsche, S., Sedlmeir, J., and Steglich, S.** *Themenfelder Industrie 4.0 – Forschungs- und Entwicklungsbedarfe zur erfolgreichen Umsetzung von Industrie 4.0*. Ed. by **Forschungsbeirat der Plattform Industrie 4.0, Deutsche Akademie der Technikwissenschaften (acatech)**. 2022. URL: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Themenfelder.pdf?__blob=publicationFile&v=1 (visited on 12/15/2022).

[66] **Horn, B. K. P. and Brooks, M. J.** *Shape from Shading*. Vol. 2. Cambridge: MIT Press, 1989.

[67] **Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., Hee, M. R., Flotte, T., Gregory, K., Puliafito, C. A., and Fujimoto, J. G.** *Optical Coherence Tomography*. In: *Science* 254.5035 (1991), pp. 1178–1181.

[68] **Hügel, H.** *Strahlwerkzeug Laser*. Wiesbaden: Teubner Studienbücher, 1992.

[69] **Hügel, H. and Graf, T.** *Grundlagen der Wechselwirkung Laserstrahl/Werkstück*. In: *Laser in der Fertigung: Strahlquellen, Systeme, Fertigungsverfahren*. Wiesbaden: Vieweg Teubner, 2009, pp. 114–173.

[70] **Hugger, F., Hofmann, K., Kohl, S., Dobler, M., and Schmidt, M.** *Spatter formation in laser beam welding using laser beam oscillation*. In: *Welding in the World* 59 (2014), pp. 165–172.

[71] **Ishigami, T., Tanaka, Y., and Homma, H.** *Motor Stator with Thick Rectangular Wire Lap Winding for HEVs*. In: *IEEE Transactions on Industry Applications* 51 (2014), pp. 1880–1885.

[72] **Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A.** *Image-to-Image Translation with Conditional Adversarial Networks*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1125–1134.

[73] **Jadon, S.** *A survey of loss functions for semantic segmentation*. In: *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020, pp. 1–7.

[74] **Jaeger, H. and Haas, H.** *Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication*. In: *Science* 304 (2004), pp. 78–80.

[75] **Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y.** *What is the best multi-stage architecture for object recognition?* In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 2146–2153.

[76] **Jetley, S., Lord, N. A., Lee, N., and Torr, P.** *Learn to Pay Attention*. In: *International Conference on Learning Representations (ICLR)*. 2018.

[77] **Kaiser, E., Huber, R., Stolzenburg, C., and Killi, A.** *Sputter-free and Uniform Laser Welding of Electric or Electronical Copper Contacts with a Green Laser*. In: *International Conference on Laser Assisted Net Shape Engineering (LANE)*. 2014.

[78] **Kaliudis, A.** *It's heading this way*. 2018. URL: https://www.trumpf.com/en_INT/presse/online-magazine/its-heading-this-way/ (visited on 02/16/2023).

[79] **Kampker, A., Kreisköther, K. D., Büning, M. K., and Treichel, P.** *Herausforderung Hairpintechnologie. Technologieschub für OEMs und Anlagenbauer*. In: *ATZelektronik Magazin* 5 (2018), pp. 62–67.

[80] **Kampker, A., Heimes, H. H., Kawollek, S., Treichel, P.-E., and Kraus, A.** *Produktionsprozess eines Hairpin-Stators*. Frankfurt: VDMA/PEM, 2019.

[81] **Kaplan, A. F. H., Norman, P., and Eriksson, I. A. G.** *Analysis of the Keyhole and Weld Pool Dynamics by Imaging Evaluation and Photodiode Monitoring*. In: *International Congress on Laser Advanced Materials Processing (LAMP)*. 2009.

[82] **Kendall, A., Badrinarayanan, V., and Cipolla, R.** *Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding*. In: *British Machine Vision Conference (BMVC)*. 2017, pp. 57.1–57.12.

[83] **Kingma, D. and Ba, J.** *Adam: A Method for Stochastic Optimization*. In: *International Conference on Learning Representations (ICLR) (Poster)*. 2014.

[84] **Klein, T., Vicanek, M., Kroos, J., Decker, I., and Simon, G.** *Oscillations of the keyhole in penetration laser beam welding*. In: *Journal of Physics D: Applied Physics* 27.10 (1994), pp. 2023–2030.

[85] **Kong, M., Shi, F. H., Chen, S. B., and Lin, T.** *Recognition of the Initial Position of Weld Based on the Corner Detection for Welding Robot in Global Environment*. In: *Robotic Welding, Intelligence and Automation*. Ed. by **Tarn, T.-J., Chen, S.-B., and Zhou, C.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 249–255.

[86] **Krizhevsky, A., Sutskever, I., and Hinton, G. E.** *ImageNet classification with deep convolutional neural networks*. In: *Communications of the ACM* 60 (2012), pp. 84–90.

[87] **Kuhl, F. P. and Giardina, C. R.** *Elliptic Fourier features of a closed contour*. In: *Computer Graphics and Image Processing* 18.3 (1982), pp. 236–258.

[88] **Kumar, A., Liang, P., and Ma, T.** *Verified Uncertainty Calibration*. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 32. 2019, pp. 1–12.

[89] **Lahdenoja, O., Säntti, T., Poikonen, J., Laiho, M., and Paasio, A.** *Characterizing Spatters in Laser Welding of Thick Steel Using Motion Flow Analysis*. In: *Scandinavian Conference on Image Analysis*. 2013.

[90] **Le, Q. V., Ranzato, M., Monga, R., Devin, M., Corrado, G. S., Chen, K., Dean, J., and Ng, A.** *Building high-level features using large scale unsupervised learning*. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 8595–8598.

[91] **Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P.** *Gradient-based learning applied to document recognition*. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[92] **Lee, S., Ahn, S., and Park, C.** *Analysis of Acoustic Emission Signals During Laser Spot Welding of SS304 Stainless Steel*. In: *Journal of Materials Engineering and Performance* 23 (2014), pp. 700–707.

[93] **Lei, Y., Li, E., Long, T., Fan, J., Mao, Y., Fang, Z., and Liang, Z.** *A welding quality detection method for arc welding robot based on 3D reconstruction with SFS algorithm*. In: *The International Journal of Advanced Manufacturing Technology* 94 (2018), pp. 1–12.

[94] **Lessmueller Lasertechnik GmbH**. *Hairpin welding*. 2003. URL: https://lessmueller.de/tasks/harpin-schweissen/?lang=en (visited on 01/22/2023).

[95] **Liebl, S., Wiedenmann, R., Ganser, A., Schmitz, P., and Zaeh, M.** *Laser Welding of Copper Using Multi Mode Fiber Lasers at Near Infrared Wavelength*. In: *International Conference on Laser Assisted Net Shape Engineering (LANE)*. Vol. 56. 2014, pp. 591–600.

[96] **Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P.** *Focal Loss for Dense Object Detection*. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2980–2988.

[97] **Liu, F., Shen, C., and Lin, G.** *Deep Convolutional Neural Fields for Depth Estimation from a Single Image*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 5162–5170.

[98] **Lu, S., Lu, J., An, K., Wang, X., and He, Q.** *Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review*. In: *IEEE Internet of Things Journal* (2023), pp. 1–24.

[99] **Maas, A., Hannun, A., and Ng, A.** *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. In: *International Conference on Machine Learning*. Vol. 28. 2013.

[100] **Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., and Rother, C.** *CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation*. In: *British Machine Vision Conference (BMVC)*. 2018.

[101] **Maiman, T. H.** *Stimulated Optical Radiation in Ruby*. In: *Nature* 187.4736 (1960), pp. 493–494.

[102] **Mainzer, K. and Kahle, R.** *Grenzen der KI – theoretisch, praktisch, ethisch*. Berlin: SpringerVerlag GmbH, 2022.

[103] **Mayr, A., Weigelt, M., Masuch, M., Meiners, M., Hüttel, F., and Franke, J.** *Application Scenarios of Artificial Intelligence in Electric Drives Production*. In: *Procedia Manufacturing* 24 (2018), pp. 40–47.

[104] **Mayr, A., Lutz, B., Weigelt, M., Gläßel, T., Kißkalt, D., Masuch, M., Riedel, A., and Franke, J.** *Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings*. In: *International Electric Drives Production Conference (EDPC)*. 2018, pp. 1–7.

[105] **McKinsey Global Institute**. *Notes from the AI frontier. Modeling the impact of AI on the world economy. Discussion Paper September 2018*. 2018. URL: https://www.mckinsey.com/~/media/mckinsey/featured%20insights/artificial%20intelligence/notes%20from%20the%20ai%20frontier%20applications%20and%20value%20of%20deep%20learning/notes-from-the-ai-frontier-insights-from-hundreds-of-use-cases-discussion-paper.pdf (visited on 12/15/2022).

[106] **McKinsey Global Institute**. *The state of AI in 2022—and a half decade in review*. 2022. URL: https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review (visited on 12/15/2022).

[107] **Mehrtash, A., Wells, W. M., Tempany, C. M., Abolmaesumi, P., and Kapur, T.** *Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation*. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 3868–3878.

[108] **Michelson, A. A. and Morley, E. W.** *On the relative motion of the Earth and the luminiferous ether*. In: *American Journal of Science (AJS)* s3-34 (1887), pp. 333–345.

[109] **Milletari, F., Navab, N., and Ahmadi, S.-A.** *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*. In: *International Conference on 3D Vision (3DV)*. 2016, pp. 565–571.

[110] **Mirza, M. and Osindero, S.** *Conditional Generative Adversarial Nets*. 2014. arXiv: 1411.1784 [cs.LG].

[111] **Mitchell, T. M.** *Machine Learning*. New York: McGraw-Hill, 1997.

[112] **Mordor Intelligence**. *Laser welding machines market*. 2022. URL: https://www.mordorintelligence.com/industry-reports/laser-welding-machines-market (visited on 12/15/2022).

[113] **Mordor Intelligence**. *Lasers market*. 2022. URL: https://www.mordorintelligence.com/industry-reports/lasers-market (visited on 12/15/2022).

[114] **Naeini, M. P., Cooper, G. F., and Hauskrecht, M.** *Obtaining Well Calibrated Probabilities Using Bayesian Binning*. In: *AAAI Conference on Artificial Intelligence*. Vol. 2015. 2015, pp. 2901–2907.

[115] **Nain, G., Pattanaik, K., and Sharma, G.** *Towards edge computing in intelligent manufacturing: Past, present and future*. In: *Journal of Manufacturing Systems* 62 (2022), pp. 588–611.

[116] **Nair, V. and Hinton, G. E.** *Rectified Linear Units Improve Restricted Boltzmann Machines*. In: *International Conference on Machine Learning (ICML)*. 2010, pp. 807–814.

[117] **Nakano, T.** *Selective Laser Melting*. In: *Multi-dimensional Additive Manufacturing*. 2020, pp. 3–26.

[118] **Nicolosi, L., Abt, F., Blug, A., Heider, A., Tetzlaff, R., and Höfler, H.** *A novel spatter detection algorithm based on typical cellular neural network operations for laser beam welding processes*. In: *Measurement Science and Technology* 23.015401 (2011).

[119] **Nicolosi, L., Tetzlaff, R., Abt, F., Heider, A., Blug, A., and Höfler, H.** *Novel algorithm for the real time multi-feature detection in laser beam welding*. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2012, pp. 181–184.

[120] **Norman, P., Eriksson, I., and Kaplan, A.** *Basic study of photodiode signals from laser welding emissions*. In: *Nordic Laser Materials Processing Conference*. 2009.

[121] **Norman, P., Engström, H., Gren, P., and Kaplan, A. F. H.** *Correlation between photodiode monitoring and high speed imaging of the dynamics causing laser welding defects*. In: *International Congress on Applications of Lasers and Electro-Optics (ICALEO)*. Vol. 27. 1708. 2008.

[122] **Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D.** *Attention U-Net: Learning Where to Look for the Pancreas*. In: *Conference on Medical Imaging with Deep Learning (MIDL)*. 2018.

[123] **Park, Y. W., Park, H., Rhee, S., and Kang, M.** *Real time estimation of CO2 laser weld quality for automotive industry*. In: *Optics and Laser Technology* 34.2 (2002), pp. 135–142.

[124] **Pohlink, C., Klug, A., Besier, J., Niestroj, J., Ofenloch-Wendel, N., and Börner, M.** *Bitkom: Maschinelles Lernen 2022 - Aktuelle Trends und deren Relevanz*. 2022. URL: https://www.bitkom.org/sites/main/files/2022-02/16.11.22_Maschinelles%20Lernen.pdf (visited on 12/15/2022).

[125] **Precitec KG**. *Laser Welding Monitor LWM*. 2003. URL: http://www.oco.ru/files/LWM_E.pdf (visited on 02/26/2023).

[126] **Purtonen, T., Kalliosaari, A., and Salminen, A.** *Monitoring and Adaptive Control of Laser Processes*. In: *Physics Procedia* 56 (2014), pp. 1218–1231.

[127] **Radford, A., Metz, L., and Chintala, S.** *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. In: *International Conference on Learning Representations (ICLR) (Poster)*. 2015.

[128] **Rahman, K. M., Jurkovic, S., Stancu, C., Morgante, J. C., and Savagian, P. J. M.** *Design and performance of electrical propulsion system of extended range electric vehicle (EREV) Chevrolet Voltec*. In: *IEEE Engergy Conversion Congress and Exposition (ECCE)*. IEEE, 2012, pp. 4152–4159.

[129] **Raina, R., Madhavan, A., and Ng, A.** *Large-scale deep unsupervised learning using graphics processors*. In: *International Conference on Machine Learning (ICML)*. 2009, pp. 873–880.

[130] **Robert Bosch GmbH**. *Vorrichtung und Verfahren zur Überwachung eines Laserbearbeitungsprozesses, Verwendung einer ereignisbasierten Kamera, Computerprogramm und Speichermedium*. Patent DE102019209376A1. Deutsches Patent- und Markenamt, 2020.

[131] **Robinson, T. and Fallside, F.** *A recurrent error propagation network speech recognition system*. In: *Computer Speech and Language* 5 (1991), pp. 259–274.

[132] **Rodríguez-Gonzálvez, P., Rodríguez-Martín, M., Ramos, L. F., and González-Aguilera, D.** *3D reconstruction methods and quality assessment for visual inspection of welds*. In: *Automation in Construction* 79 (2017), pp. 49–58.

[133] **Rodríguez-Vázquez, Á., Domínguez-Castro, R., Jiménez-Garrido, F., Morillas, S., Listán, J., Alba, L., Utrera, C., Espejo, S., and Romay, R.** *The Eye-RIS CMOS Vision System*. In: *Analog Circuit Design: Sensors, Actuators and Power Drivers; Integrated Power Amplifiers from Wireline to RF; Very High Frequency Front Ends*. Ed. by **Casier, H., Steyaert, M., and Van Roermund, A. H. M.** Dordrecht: Springer Netherlands, 2008, pp. 15–32.

[134] **Ronneberger, O., Fischer, P., and Brox, T.** *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015.

[135] **Rosenblatt, F.** *The perceptron: a probabilistic model for information storage and organization in the brain*. In: *Psychological review* 65 6 (1958), pp. 386–408.

[136] **Rosenblatt, F.** *Principles of Neurodynamics*. New York: Spartan, 1962.

[137] **Rumelhart, D. E., Hinton, G. E., and Williams, R. J.** *Learning representations by back-propagating errors*. In: *Nature* 323 (1986), pp. 533–536.

[138] **Ryberg, A., Ericsson, M., Christiansson, A.-K., Eriksson, K., Nilsson, J., and Larsson, M.** *Stereo vision for path correction in off-line programmed robot welding*. In: *IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2010, pp. 1700–1705.

[139] **Salakhutdinov, R. and Hinton, G. E.** *Deep Boltzmann Machines*. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 5. 2009.

[140] **Saul, L. K., Jaakkola, T., and Jordan, M. I.** *Mean Field Theory for Sigmoid Belief Networks*. In: *Journal of Artificial Intelligence Research (JAIR)* 4 (1996), pp. 61–76.

[141] **Saxena, A., Sun, M., and Ng, A.** *Make3D: Learning 3D Scene Structure from a Single Still Image*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (5 2009), pp. 824–840.

[142] **Schlick, T., Hertel, G., Hagemann, B., Maiser, E., and Kramer, M.** *Studie: Zukunftsfeld Elektromobilität. Chancen und Herausforderungen für den deutschen Maschinen- und Anlagenbau*. Ed. by **Roland Berger Strategy Consultants**. 2018.

[143] **Schmidt, L., Roemer, F., Böttger, D., Leinenbach, F., Strass, B., Wolter, B., Schricker, K., Seibold, M., Bergmann, J., and Del Galdo, G.** *Acoustic process monitoring in laser beam welding*. In: *CIRP Conference on Photonic Technologies (LANE)*. Vol. 94. 2020, pp. 763–768.

[144] **Schmidt, P. A. and Zäh, M. F.** *Laser beam welding of electrical contacts of lithium-ion batteries for electric- and hybrid-electric vehicles*. In: *Production Engineering* 9 (2015), pp. 593–599.

[145] **Settles, B.** *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin-Madison, 2009.

[146] **Shannon, C. E.** *A mathematical theory of communication*. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[147] **Shao, J. and Yan, Y.** *Review of techniques for on-line monitoring and inspection of laser welding*. In: *Journal of Physics: Conference Series* 15.1 (2005), p. 101.

[148] **Shi, B., Bai, S., Zhou, Z., and Bai, X.** *DeepPano: Deep Panoramic Representation for 3-D Shape Recognition*. In: *IEEE Signal Processing Letters* 22 (12 2015), pp. 2339–2343.

[149] **Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L.** *Edge Computing: Vision and Challenges*. In: *IEEE Internet of Things Journal* 3.5 (2016), pp. 637–646.

[150] **Sievi, P. and Käfer, S.** *Hairpins für E-Mobility schweißen*. 2021. URL: https://www.maschinenmarkt.vogel.de/hairpins-fuer-e-mobility-schweissen-a-1003621/ (visited on 02/16/2023).

[151] **Soltani, A. A., Huang, H., Wu, J., Kulkarni, T. D., and Tenenbaum, J. B.** *Synthesizing 3D Shapes via Modeling Multi-view Depth Maps and Silhouettes with Deep Generative Networks*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2511–2519.

[152] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.** *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[153] **Stadter, C., Schmoeller, M., Zeitler, M., Tueretkan, V., Munzert, U., and Zaeh, M. F.** *Process control and quality assurance in remote laser beam welding by optical coherence tomography*. In: *Journal of Laser Applications* 31.022408 (2019).

[154] **Stadter, C., Schmoeller, M., von Rhein, L., and Zaeh, M. F.** *Real-time prediction of quality characteristics in laser beam welding using optical coherence tomography and machine learning*. In: *Journal of Laser Applications* 32.022046 (2020).

[155] **Stavridis, J., Papacharalampopoulos, A., and Stavropoulos, P.** *Quality assessment in laser welding: a critical review*. In: *International Journal of Advanced Manufacturing Technology* 94 (2018), pp. 1–23.

[156] **Sun, A. and Kannatey-Asibu Jr., E.** *Sensor systems for real-time monitoring of laser weld quality*. In: *Journal of Laser Applications* 11 (1999), pp. 153–168.

[157] **Suzuki, S. and Abe, K.** *Topological structural analysis of digitized binary images by border following*. In: *Computer Vision, Graphics, and Image Processing* 30.1 (1985), pp. 32–46.

[158] **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.** *Going deeper with convolutions*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1–9.

[159] **Tabernik, D., Šela, S., Skvarč, J., and Skočaj, D.** *Segmentation-based deep-learning approach for surface-defect detection*. In: *Journal of Intelligent Manufacturing* 31 (3 2020), pp. 759–776.

[160] **The New York Times Archives**. *Developer of the Laser Calls It 'A Solution Seeking a Problem'; President of Korad Spends Spare Time Gardening and Fixing TV Sets*. 1964. URL: https://www.nytimes.com/1964/05/06/archives/developer-of-the-laser-calls-it-a-solution-seeking-a-problem.html (visited on 02/20/2023).

[161] **Turing, A. M.** *Computing machinery and intelligence*. In: *Mind* LIX.236 (1950), pp. 433–460.

[162] **Vater, J., Pollach, M., Lenz, C., Winkle, D., and Knoll, A.** *Quality Control and Fault Classification of Laser Welded Hairpins in Electrical Motors*. In: *European Signal Processing Conference (EUSIPCO)*. 2021, pp. 1377–1381.

[163] **Volpp, J.** *Keyhole stability during laser welding—Part II: process pores and spatters*. In: *Production Engineering* 11 (2016), pp. 9–18.

[164] **Wang, D., Gong, B., and Wang, L.** *On Calibrating Semantic Segmentation Models: Analyses and An Algorithm*. 2023. arXiv: 2212.12053 [cs.CV].

[165] **Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L.** *Cost-Effective Active Learning for Deep Image Classification*. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12 (2017), pp. 2591–2600.

[166] **Wang, S., Hu, S.-Y., Cheah, E., Wang, X., Wang, J., Chen, L., Baikpour, M., Ozturk, A., Li, Q., Chou, S.-H., Lehman, C. D., Kumar, V., and Samir, A.** *U-Net Using Stacked Dilated Convolutions for Medical Image Segmentation*. 2020. arXiv: 2004.03466 [eess.IV].

[167] **Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E.** *Image quality assessment: from error visibility to structural similarity*. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[168] **Weigelt, M., Mayr, A., Seefried, J., Heisler, P., and Franke, J.** *Conceptual design of an intelligent ultrasonic crimping process using machine learning algorithms*. In: *Procedia Manufacturing* 17 (2018), pp. 78–85.

[169] **Widrow, B. and Hoff, M. E.** *Adaptive Switching Circuits*. In: *IRE WESCON Convention Record, Part 4*. New York: IRE, 1960, pp. 96–104.

[170] **Will, T., Müller, J., Müller, R., Hölbling, C., Goth, C., and Schmidt, M.** *Prediction of electrical resistance of laser-welded copper pin-pairs with surface topographical information from inline post-process observation by optical coherence tomography*. In: *The International Journal of Advanced Manufacturing Technology* 125 (2023), pp. 1955–1963.

[171] **Wolpert, D. and Macready, W.** *No free lunch theorems for optimization*. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82.

[172] **Yang, J.-Z., Zhang, Y., Chen, J.-h., Wang, W.-q., and Liu, Y.** *Off-Line Programming System of Multi-Axis Platform for Dual Beam Laser Welding*. In: *International Conference on Applied Mechanics, Mechatronics and Intelligent Systems (AMMIS)*. 2016, pp. 85–92.

[173] **Ye, G., Guo, J., Sun, Z., Li, C., and Zhong, S.** *Weld bead recognition using laser vision with model-based classification*. In: *Robotics and Computer-Integrated Manufacturing* 52 (2018), pp. 9–16.

[174] **Yu, F. and Koltun, V.** *Multi-Scale Context Aggregation by Dilated Convolutions*. In: *International Conference on Learning Representations (ICLR)*. Vol. abs/1511.07122. 2015.

[175] **Zeng, H., Zhou, Z., Chen, Y., Luo, H., and Hu, L.** *Wavelet analysis of acoustic emission signals and quality control in laser welding*. In: *Journal of Laser Applications* 13 (2001), pp. 167–173.

[176] **Zhang, J., Li, K., Liang, Y., and Li, N.** *Learning 3D faces from 2D images via Stacked Contractive Autoencoder*. In: *Neurocomputing* 257 (2017), pp. 67–78.

[177] **Zhang, M., Chen, G., Zhou, Y., Li, S., and Deng, H.** *Observation of spatter formation mechanisms in high-power fiber laser welding of thick plate*. In: *Applied Surface Science* 280 (2013), pp. 868–875.

[178] **Zhang, X.-G., Xu, J.-J., and Ge, G.-Y.** *Defects recognition on X-ray images for weld inspection using SVM*. In: *International Conference on Machine Learning and Cybernetics*. Vol. 6. 2004, pp. 3721–3725.

[179] **Zhang, Y., Liu, S., Li, C., and Wang, J.** *Rethinking the Dice Loss for Deep Learning Lesion Segmentation in Medical Images*. In: *Journal of Shanghai Jiaotong University (Science)* 26 (2021), pp. 93–102.

[180] **Zhou, Y. and Chellappa, R.** *Computation of optical flow using a neural network*. In: *IEEE International Joint Conference on Neural Networks (IJCNN)*. Vol. 2. IEEE, 1988, pp. 71–78.

List of own Publications

[181] **Hartung, J., Jahn, A., and Heizmann, M.** *Quality control of laser welds based on the weld surface and the weld profile*. In: *Forum Bildverarbeitung 2022*. KIT Scientific Publishing, 2022.

[182] **Hartung, J., Jahn, A., and Heizmann, M.** *Machine learning based geometry reconstruction for quality control of laser welding processes*. In: *tm - Technisches Messen* (2023). https://doi.org/10.1515/teme-2023-0006.

[183] **Hartung, J., Jahn, A., Stambke, M., Wehner, O., Thieringer, R., and Heizmann, M.** *Camera-based spatter detection in laser welding with a deep learning approach*. In: *Forum Bildverarbeitung 2020*. KIT Scientific Publishing, 2020.

[184] **Hartung, J., Jahn, A., Bocksrocker, O., and Heizmann, M.** *Camera-Based In-Process Quality Measurement of Hairpin Welding*. In: *Applied Sciences* 11.21 (2021).

[185] **Hartung, J., Dold, P. M., Jahn, A., and Heizmann, M.** *Analysis of AI-Based Single-View 3D Reconstruction Methods for an Industrial Application*. In: *Sensors* 22.17 (2022).

List of Supervised Theses

[186] **Dold, P. M.** *Bildbasierte 3D-Rekonstruktion mittels Deep Learning zur Qualitätsprüfung von Hairpins*. Master thesis. Karlsruher Institut für Technologie (KIT), 2022.