Innovations in
Manufacturing Technology

The aim of this thesis was to develop and optimize deep learning models specifically designed for the identification of tool wear on microscopic images of cutting tools and cutting tool edges. Cutting tool wear has an impact on dimensional accuracy and surface quality of parts, ultimately affecting the costs associated with meeting part quality criteria.

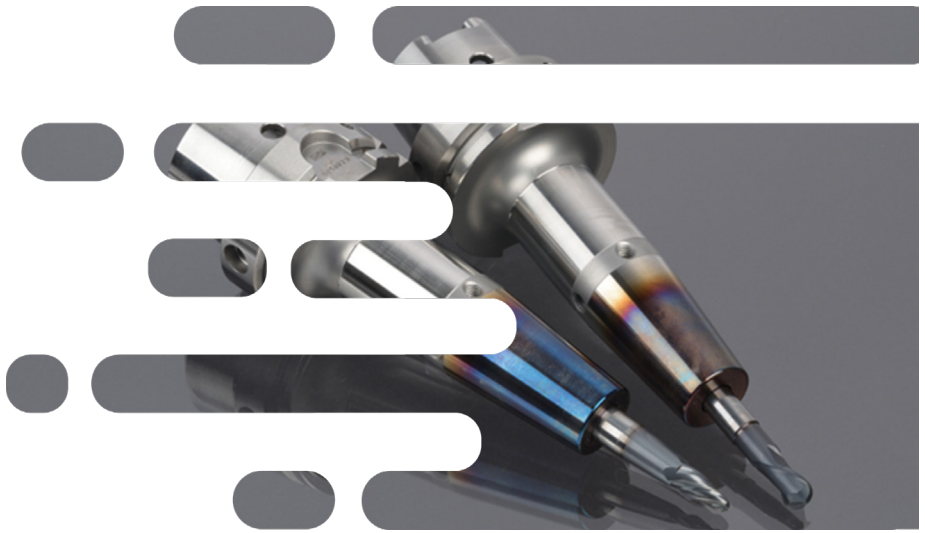
To accomplish this objective, the creation of a tool wear model based on empirical tool life trials was conducted. An outcome of the trials was the generation of a dataset of images, which were then utilized to develop a deep learning model capable of segmenting cutting tool flank wear. To ensure the effectiveness of the deep learning model, a screening analysis was conducted to investigate various dataset properties and model hyperparameters that could influence the quality of predictions. The screening analysis helped identify the key factors that significantly impacted the performance of the model. Building upon the insights gained from the screening analysis, the thesis proceeded with an in-depth investigation of the most influential factors. This investigation led to the development of a decision model that could guide the selection of dataset-specific hyperparameters for optimal performance. To validate the effectiveness of the model optimization strategy, a machine tool integrated measurement setup was employed, utilizing a microscope as well as a camera. These use cases provided a practical assessment of the developed deep learning model and its ability to identify and assess tool wear in a real-world manufacturing scenario.

By developing and refining deep learning models for tool wear identification on microscopic images, this thesis contributes to enhancing the understanding and management of tool wear in the manufacturing industry. The optimized models have the potential to facilitate timely maintenance interventions, minimize production errors, and reduce costs associated with part quality deviations. Moreover, the decision model for dataset-specific hyperparameter selection provides a valuable framework for researchers and practitioners working on similar image-based classification problems.

ISBN 978-3-98555-261-0

Automated Flank Wear Segmentation and Measurement with Deep Learning Image Processing
Carsten Holst

Innovations in
Manufacturing Technology



Automated Flank Wear Segmentation and Measurement with Deep Learning Image Processing

Carsten Holst
03/2025

MTI | RWTH AACHEN UNIVERSITY
Fraunhofer IPT
Apprimus

Automated Flank Wear Segmentation and Measurement with Deep Learning Image Processing

Automatisierte Segmentation und Messung von Freiflächenverschleiß mit Deep Learning Bildverarbeitung

Von der Fakultät für Maschinenwesen
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von

Carsten Holst

Berichter/in:

Univ.-Prof. Dr.-Ing. Thomas Bergs
Prof. Dr.-Ing. Martin Dix

Tag der mündlichen Prüfung: 16. Oktober 2024

INNOVATIONS IN
MANUFACTURING TECHNOLOGY

Carsten Holst

Automated Flank Wear Segmentation and Measurement with Deep Learning Image Processing

Herausgeber:
Prof. Dr.-Ing. T. Bergs

Band 3/2025

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über https://portal.dnb.de abrufbar.

Carsten Holst:

Automated Flank Wear Segmentation and Measurement with Deep Learning Image Processing

1. Auflage, 2025

Gedruckt auf holz- und säurefreiem Papier, 100% chlorfrei gebleicht.

Wissenschaftsverlag des Instituts für Industriekommunikation und Fachmedien
an der RWTH Aachen
Steinbachstr. 25, 52074 Aachen, Deutschland
Internet: www.apprimus-verlag.de, E-Mail: info@apprimus-verlag.de

Printed in Germany

ISBN 978-3-98555-261-0

D 82 (Diss. RWTH Aachen University, 2024)

Preamble

Vorwort

This dissertation was written while I was employed as a research assistent at the Fraunhofer-Institute for Production Technology IPT in Aachen.

I would like to express my gratitude to Professor Thomas Bergs, Chair of Manufacturing Technology at the Manufacturing Technology Institute (MTI) of RWTH Aachen University and Member of the Board of Directors of the Fraunhofer-Institute for Production Technology IPT, for his support and the sponsoring of my work.

For the critical review of this thesis, I thank Dr.-Ing. Martin Seimann, Dr.-Ing Thorsten Augspurger and Dr-Ing. Markus Landwehr.

I also want to thank Philipp Ganser M.Sc., the leader of my department high performance cutting at Fraunhofer IPT as well as his predecessor Dr.-Ing. Florian Degen MBA, who continuously supported me and my work and thus contributed significantly to my professional and personal development.

Furthermore, I would like to thank my colleagues at the machine tool laboratory and the Fraunhofer IPT in Aachen for their manifold support. I want to thank Alex, Daniel, David, Felix, Florian, Georg, Grigory, Ines, Jannik, Jonas, Josef, Kilian, Lionel, Markus, Martin, Marvin, Niklas, Pascal, Pascal, Philipp, Richard, Sascha, Sebastian, Semir, Stefan, Steffi, Sven, Timur, Tobias, Tommy, Victor, Vincent, and Willi for all the good times I had with them. Christian and Roman I want to thank you for your support which I really appreciated.

Further thanks go of course to my research assistants and students without whom such a work could not have been done. A special thank you to Alexander, Anna, Daniel, Eduardo, Felix, Pranjul, Patrick, Taha and Yuri.

I thank my family very much for their enormous support and belief in me. I thank my friends for discussions and support throughout the thesis and beyond. Finally, I would like to thank my wife Kira and my two children, who contributed greatly to the creation of this work.

Aachen, July 2024

Carsten Holst

Abstract

Zusammenfassung

The aim of this thesis was to develop and optimize deep learning models specifically designed for the identification of tool wear on microscopic images of cutting tools and cutting tool edges. Cutting tool wear has an impact on dimensional accuracy and surface quality of parts, ultimately affecting the costs associated with meeting part quality criteria.

To accomplish this objective, the creation of a tool wear model based on empirical tool life trials was conducted. An outcome of the trials was the generation of a dataset of images, which were then utilized to develop a deep learning model capable of segmenting cutting tool flank wear. To ensure the effectiveness of the deep learning model, a screening analysis was conducted to investigate various dataset properties and model hyperparameters that could influence the quality of predictions. The screening analysis helped identify the key factors that significantly impacted the performance of the model. Building upon the insights gained from the screening analysis, the thesis proceeded with an in-depth investigation of the most influential factors. This investigation led to the development of a decision model that could guide the selection of dataset-specific hyperparameters for optimal performance. To validate the effectiveness of the model optimization strategy, a machine tool integrated measurement setup was employed, utilizing a microscope as well as a camera. These use cases provided a practical assessment of the developed deep learning model and its ability to identify and assess tool wear in a real-world manufacturing scenario.

By developing and refining deep learning models for tool wear identification on microscopic images, this thesis contributes to enhancing the understanding and management of tool wear in the manufacturing industry. The optimized models have the potential to facilitate timely maintenance interventions, minimize production errors, and reduce costs associated with part quality deviations. Moreover, the decision model for dataset-specific hyperparameter selection provides a valuable framework for researchers and practitioners working on similar image-based classification problems.

Zusammenfassung

Abstract

Das Ziel dieser Arbeit war es, Deep-Learning-Modelle zu entwickeln und zu optimieren, die speziell für die Erkennung von Werkzeugverschleiß auf mikroskopischen Bildern von Zerspanungswerkzeugen und Schneidkanten konzipiert sind. Der Verschleiß von Zerspanungswerkzeugen beeinflusst die Maßgenauigkeit und die Oberflächenqualität von Bauteilen, was sich letztlich auf die Kosten auswirkt, die mit der Einhaltung der Qualitätskriterien für die Bauteile verbunden sind.

Um dieses Ziel zu erreichen, wurde die Erstellung eines Werkzeugverschleißmodells auf der Grundlage empirischer Standzeitversuche durchgeführt. Ein Ergebnis der Versuche war die Erstellung eines Satzes von Bildern, die dann zur Entwicklung eines Deep-Learning-Modells verwendet wurden, das in der Lage ist, den Verschleiß auf der Freifläche von Zerspanungswerkzeugen zu segmentieren. Um die Effektivität des Deep-Learning-Modells zu gewährleisten, wurde eine Screening-Analyse durchgeführt, um verschiedene Datensatzeigenschaften und Modellhyperparameter zu untersuchen, welche die Qualität der Vorhersagen beeinflussen könnten. Mit Hilfe der Screening-Analyse konnten die Schlüsselfaktoren identifiziert werden, welche die Leistung des Modells erheblich beeinflussten. Aufbauend auf den aus der Screening-Analyse gewonnenen Erkenntnissen wurde in dieser Arbeit eine eingehende Untersuchung der einflussreichsten Faktoren durchgeführt. Diese Untersuchung führte zur Entwicklung eines Entscheidungsmodells, welches die Auswahl von datensatzspezifischen Hyperparametern für eine optimale Leistung anleiten kann. Um die Effektivität der Modelloptimierungsstrategie zu validieren, wurde ein in eine Werkzeugmaschine integrierter Messaufbau unter Verwendung eines Mikroskops sowie einer Kamera eingesetzt. Diese Anwendungsfälle lieferte eine praktische Bewertung des entwickelten Deep-Learning-Modells und seiner Fähigkeit, Werkzeugverschleiß in einem realen Fertigungsszenario zu erkennen und zu bewerten.

Durch die Entwicklung und Verfeinerung von Deep-Learning-Modellen zur Identifizierung von Werkzeugverschleiß auf mikroskopischen Bildern trägt diese Arbeit dazu bei, das Verständnis und das Management von Werkzeugverschleiß in der Fertigungsindustrie zu verbessern. Die optimierten Modelle haben das Potenzial, rechtzeitige Wartungseingriffe zu erleichtern, Produktionsfehler zu minimieren und die mit Qualitätsabweichungen von Teilen verbundenen Kosten zu senken. Darüber hinaus bietet das Entscheidungsmodell für die datensatzspezifische Auswahl von Hyperparametern einen wertvollen Rahmen für Forscher und Praktiker, die an ähnlichen bildbasierten Klassifikationsproblemen arbeiten.

Content

Inhaltsverzeichnis

1 Introduction 1

2 Fundamentals and State of the Art 5

2.1 Tool Wear in Metal Cutting 5
2.1.1 Basic Concepts on the Cutting Part of Cutting Tools 5
2.1.2 Tribology in Metal Cutting 6
2.1.3 Chain of Effects in Cutting Tool Wear 7
2.1.4 Impact of Cutting Tool Wear 9
2.2 Quantification of Tool Wear 10
2.2.1 Terminology of Tool Life 10
2.2.2 Tool Life Testing in Metal Cutting 13
2.2.3 Direct Tool Wear Measurement 16
2.2.4 Computer Vision for Automated Tool Wear Detection 17
2.3 Image Processing with Deep Learning 18
2.3.1 Fundamentals of Machine Learning 19
2.3.2 Neural Network Training 21
2.3.3 Image Processing with Artificial Intelligence 23
2.3.4 Tool Wear Identification with Deep Learning 30
2.4 Interim Conclusion 36

3 Objectives and Approach 39

3.1 Objectives and Research Methodology 39
3.2 Procedure and Setup of the Thesis 40

4 Tool Wear Modelling and Segmentation 43

4.1 Surveys with Industry Professionals 43
4.2 Framework of Investigation 44
4.3 Process Specification 45
4.4 Empirical Investigation of Tool Wear 47
4.4.1 Design of Experiments 47
4.4.2 Analysis of Occurring Tool Wear 48
4.4.3 Tool Wear Model Creation 50
4.5 Model Design for Tool Wear Segmentation 54
4.5.1 Experimental Setup 54
4.5.2 Labeling Areas of Interest in the Image Data 55
4.5.3 Model Setup and Training 56
4.5.4 Evaluation of Model Performance 59
4.6 Interim Conclusion 63

5 Model Performance Optimization 65

5.1 Methodology for Model Performance Optimization 65
5.2 Prerequisites and Definitions 66
5.2.1 Model Hyperparameters 66
5.2.2 Dataset Properties 68
5.2.3 Model Evaluation Metrics 73
5.3 Screening Analysis 75
5.3.1 Preparation 75
5.3.2 Significance Analysis 77
5.3.3 Effect Size Analysis 80
5.3.4 Discussion of findings 81
5.4 Full Factorial Analysis 83
5.4.1 Preparations 83
5.4.2 Exploratory Analysis 86
5.4.3 Outlier Analysis 88
5.4.4 Interaction Analysis 89
5.4.5 Main Effect Analysis 90
5.4.6 Discussion of Findings 92
5.5 Decision Model 93
5.5.1 Modelling Approach 93
5.5.2 Methodology for Decision Model Creation 95
5.5.3 Regression Models 96
5.5.4 Target Value Optimization 99
5.5.5 Model Validation 102
5.5.6 Discussion of Findings 110
5.6 Interim Conclusion 111

6 Validation of AI-based Automated Tool Wear Measurement 113

6.1 Empirical Validation of the AI-based Measurement 113
6.1.1 Calculation of Width of Flank Wear Land VB 113
6.1.2 Fundamental Trial for Validation of the AI-based Wear Measurement 114
6.1.3 Turbine Blade Milling for Validation of the AI-based Wear Measurement 116
6.2 Economic Considerations 124
6.3 Interim Conclusion 125

7 Summary and Outlook 127

8 References 139

A Appendix 149

Formula Symbols and Abbreviations

Formelzeichen und Abkürzungsverzeichnis

Capital letters

A_α	mm²	Major flank face
A_γ	mm²	Major rake face
C	m/min	Theoretical intersection of x-axis and regression line
D	mm	Tool diameter
$Dice$		Sørensen-Dice coefficient
E		Error threshold
$F1$		Harmonic mean of the precision and recall
F		Measure of a test statistic with F-distribution
FN		False negative
FP		False positive
IoU		Intersect over union
K_a		x-dimension of receptive field in kernel
K_b		y-dimension of receptive field in kernel
L		Loss function
L_f	m	Tool life travel path
N		Number of images
OP_{TT}		Overfitting percentage training test
R^2		Coefficient of determination
SV_α	µm	Displacement of the cutting edge (direction flank)
SV_γ	µm	Displacement of the cutting edge (direction rake)
T	min	Tool life time
TN		True negative
TP		True positive
VB	µm	Width of flank wear land
VB_{avg}	µm	Width of flank wear land averaged over teeth
VB_{max}	µm	Maximum width of flank wear land

Small letters

a_e	mm	Width of cut
a_p	mm	Depth of cut
b		Bias term in neural networks
f_z	mm	Feed per tooth
k	log(1/min)	Tangent of angle between x-axis and regression line
m_t		Exponentially decaying average of past gradients
n		Number of samples
n_{rpm}	1/min	Rounds per minute of milling spindle
p		Probability threshold value for hypothesis rejection
r		Pearson's correlation coefficient
r_β	mm	Cutting edge radius
t		Measure of a test statistic with t-distribution
v_c	m/min	Cutting speed
$v_{c,max}$	m/min	Maximum cutting speed
$v_{c,th}$	m/min	Theoretical cutting speed
v_f	mm/s	Feed velocity
w		Weights in neural networks
z		Number of teeth
z_n		Measure of a test statistic with Z-distribution

Greek letters

α		Significance level in hypothesis testing
α_0	°	Tool orthogonal clearance angle
β_0	°	Wedge angle
β_{fn}	°	Tool axis inclination angle
γ_0	°	Tool orthogonal rake angle
$\Delta\theta$		Momentum parameter
η		Learning rate parameter
θ		Loss function parameter
λ_0	°	Helix angle of cutting edge

μ	Arithmetic mean
v_t	Exponentially decaying mean of squared gradients
σ	Standard deviation

Abbreviations

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence, Artificial Intelligence
AMS	Aerospace Material Specification
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
API	Application Programming Interface
BCE	Binary Cross Entropy
BEV	Battery Electric Vehicles
BIM	Basic Image Manipulation
BN	Batch Normalization
BUE	Build-up Edge
CAGR	Cumulative Annual Growth Rate
CMOS	Complementary metal–oxide–semiconductor
CNC	Computerized Numerical Control
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CV	Computer Vision
DIN	Deutsche Industrie Norm
DL	Deep Learning
DLA	Deep Learning Augmentation
DNN	Deep Neural Network
DOE	Design of Experiments
ELU	Exponential Linear Unit
FC	Fully Connected
FCN	Fully Convolutional Networks
FSIM	Feature-based Similarity Index
GAN	Generative Adversarial Network

GDP	Gross Domestic Product
GPU	Graphical Processing Unit
GSD	Generalized Subset Design
ISBI	International Symposium on Biomedical Imaging
ISO	International Organization for Standardization
ISSM	Information Statistic Similarity Measure
KPI	Key Performance Indicator
LPIPS	Learned Perceptual Image Patch Similarity
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NAMRC	North American Manufacturing Research Conference
NAS	Neural Architecture Search
NN	Neural Network
PSNR	Peak Signal-to-Noise Ratio
R-CNN	Region Based Convolutional Neural Network
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RMSE	Root Mean Square Error
SAM	Spectral Angle Mapper
SGD	Stochastic Gradient Descent
SRE	Signal to Reconstruction Error Ratio
SSIM	Structural Similarity Index
TCM	Tool Condition Monitoring
UIQ	Universal Image Quality Index
USB	Universal Serial Bus

1 Introduction

Einleitung

The global metal cutting tools market refers to the market for tools used in the manufacturing industry to cut metal. These tools are used in a variety of machining applications, including milling, turning, drilling, and grinding. The market had a size of approximately USD 76 billion in 2022 and is driven by factors such as the increasing demand for metal products and the growing need for precision and efficiency in manufacturing processes [FORT22]. About 65 % of metal cutting tools are made from cemented carbide due to its high hardness, wear resistance, and toughness [PERS19, KLOC18]. Cemented carbide is made of tungsten carbide and cobalt. The proportion of cobalt is in the range of 5 to 12 % [ISO513:2012].

There are several factors that might lead to an increase in the consumption of cobalt and tungsten, including economic growth, rising standards of living across the globe and technological advancements: Electric vehicles, smart devices and renewable energy systems require cobalt for their batteries. Especially, the trend of intelligent transportation systems with Battery Electric Vehicles (BEV), which are growing at a Cumulative Annual Growth Rate (CAGR) of 17 % [STAT 22a], leads to an increased demand for the ressources and therefore significant price increases for cutting tools may be expectable in the coming years. In 2022 prices for cutting tools increased between 5 - 25 % across major cutting tools manufacturers [SEIS22].

During manufacturing of aerospace engine components, high tool wear is generated. This is due to the use of hard to cut materials such as nickel-based alloys which are often required to produce aerospace engine components. Their outstanding properties make it difficult to machine and lead to severe milling tool wear, which can affect the quality of the product [MOHA20]. The decrease in product quality due to tool wear and the resulting machine downtime due to frequent tool changes are the main challenges in machining [BIND17]. Tool wear is thus responsible for high production costs and poor surface qualities, resulting in an increased need for optimization especially in workpiece surface generating finish milling operations. For the estimation of the tool life, empirical wear models may be applied, which require complex and cost-intensive experiments for the generation of the model data [ZHOU18].

Apart from the literature study, interviews with experts from the metal cutting industry were conducted to size the importance of the cutting tool wear problem. Especially in manufacturing of aerospace components, the tool costs currently amount approximately 8 % of the costs of goods sold, according engineers from the industry, see also Section 4.1, Surveys with Industry Professionals. In actual practice, there are three ways to cope with the cutting tool wear problem usually applied in series production, in research and development and in small batch production:

1. Fixed tool life from prior experiments with a safety margin to account for outliers
2. Creation of tool life models to allow a prediction of tool life across a range of cutting speed and/or other cutting parameters

3. Optical observation and assessment of the tool status by the machine tool operator

The first approach requires costly testing and is applicable for series production with highly repetitive processes and fixed process conditions. The safety margin applied to account for outliers, which are tools that fail earlier, results in a waste of numerous good tools. The second approach, tool life modelling, is complex and cost-intensive since many experiments are required for the generation of the model data. The last approach is common in single part or small batch production, it relies on experience of the machine tool operator and is thus not automated.

The above-mentioned data and argumentation provide motivation to increase cutting tool utilization. Making the best use of each individual tool regarding its tool life, could be achieved through intelligent assistance and automation solutions. This thesis aims to explore potential methods for addressing tool wastage in the future through the utilization of inline metrology, specifically, the capture of images within the machine tool, coupled with AI-based image processing. Specifically, an approach to segment flank wear on cutting tool edges with a U-Net model architecture is presented. Furthermore, an investigation of the influence of model hyperparameters and dataset properties on the neural network model's performance is conducted. Based on the findings, an approach to creating a decision model for hyperparameter optimization based on dataset properties is developed for these most influential factors. Finally, the approach is used to train a U-Net for a specific dataset made with an inline microscope that acquires cutting tool edge images within a machine tool.

1 Einleitung

Introduction

Der Weltmarkt für Zerspanungswerkzeuge bezieht sich auf den Markt für Werkzeuge, die in der Fertigungsindustrie zum Zerspanen von Metall verwendet werden. Diese Werkzeuge werden in einer Vielzahl von Bearbeitungsanwendungen eingesetzt, darunter Fräsen, Drehen, Bohren und Schleifen. Der Markt hat ein Volumen von ca. 76 Mrd. USD im Jahr 2022 und wird durch Faktoren wie die steigende Nachfrage nach Metallprodukten und den wachsenden Bedarf an Präzision und Effizienz in Fertigungsprozessen angetrieben [FORT22]. Etwa 65 % der Zerspanungswerkzeuge werden aufgrund ihrer hohen Härte, Verschleißfestigkeit und Zähigkeit aus Hartmetall hergestellt [PERS19, KLOC18]. Sinterkarbid wird aus Wolframkarbid und Kobalt hergestellt. Der Anteil an Kobalt macht zwischen 5 bis 12 % aus [ISO513:2012].

Es gibt mehrere Faktoren, die zu einem Anstieg des Verbrauchs von Kobalt und Wolfram führen könnten, darunter das Wirtschaftswachstum, der steigende Lebensstandard auf der ganzen Welt und der technische Fortschritt: Elektrofahrzeuge, intelligente Geräte und erneuerbare Energiesysteme benötigen Kobalt für ihre Batterien. Insbesondere der Trend zu intelligenten Verkehrssystemen mit batteriebetriebenen Elektrofahrzeugen (BEV), die mit einer jährlichen Wachstumsrate von 17 % wachsen [STAT 22b], führt zu einer erhöhten Nachfrage nach den Ressourcen, so dass in den kommenden Jahren mit einem erheblichen Preisanstieg für Zerspanungswerkzeuge zu rechnen ist. Im Jahr 2022 stiegen die Preise für Zerspanungswerkzeuge bei den wichtigsten Herstellern von Zerspanungswerkzeugen zwischen 5 und 25 % [SEIS22].

Bei der Herstellung von Triebwerkskomponenten für die Luft- und Raumfahrt kommt es zu einem hohen Werkzeugverschleiß. Dies ist auf die Verwendung von schwer zerspanbaren Werkstoffen wie Nickelbasislegierungen zurückzuführen, die in Triebwerkskomponenten für die Luft- und Raumfahrt eingesetzt werden. Ihre hervorragenden Eigenschaften erschweren die Bearbeitung und führen zu einem hohen Verschleiß der Fräswerkzeuge, was die Qualität des Produkts beeinträchtigen kann [MOHA20]. Die Abnahme der Produktqualität aufgrund von Werkzeugverschleiß und die daraus resultierenden Maschinenstillstandszeiten aufgrund häufiger Werkzeugwechsel sind die größten Herausforderungen bei der Bearbeitung [BIND17]. Der Werkzeugverschleiß ist somit für hohe Produktionskosten und schlechte Oberflächenqualitäten verantwortlich, was zu einem erhöhten Optimierungsbedarf insbesondere bei werkstückoberflächenerzeugenden Schlichtfräsoperationen führt. Zur Abschätzung der Werkzeugstandzeit können Verschleißmodelle eingesetzt werden, die komplexe und kostenintensive Experimente zur Generierung der Modelldaten erfordern [ZHOU18]. Neben der Literaturstudie wurden Interviews mit Personen aus der Zerspanungsindustrie geführt, um die Bedeutung des Verschleißproblems bei Zerspanungswerkzeugen zu ermitteln. Insbesondere bei der Herstellung von Bauteilen für die Luft- und Raumfahrt belaufen sich die Werkzeugkosten nach Angaben von acht Forschungs- und Entwicklungsingenieuren

aus der Branche derzeit auf etwa 8 % der Umsatzkosten des Produkts, siehe Abschnitt 4.1, Surveys with Industry Professionals. In der Praxis gibt es drei Möglichkeiten zur Bewältigung des Werkzeugverschleißproblems, die in der Regel in der Serienfertigung, in Forschung und Entwicklung und in der Kleinserienfertigung angewandt werden:

1. Festlegen der Standzeit aus einer vorherigen Prüfung mit einer Sicherheitsmarge, um Ausreißer zu berücksichtigen
2. Erstellung von Standzeitmodellen, die eine Vorhersage der Werkzeugstandzeit über einen Bereich von Schnittgeschwindigkeiten und/oder anderen Schnittparametern ermöglichen
3. Optische Beobachtung und Bewertung des Werkzeugstatus durch den Bediener der Werkzeugmaschine

Der erste Ansatz erfordert kostspielige Versuche und ist für die Serienproduktion mit sich stark wiederholenden Prozessen und festen Prozessbedingungen geeignet. Die Sicherheitsmarge, die zur Berücksichtigung von Ausreißern, d. h. Werkzeuge die früher versagen, angewandt wird, führt dazu, dass zahlreiche Werkzeuge verschwendet werden. Der zweite Ansatz, die Modellierung der Werkzeugstandzeit, ist komplex und kostenintensiv, da viele Versuche für die Generierung der Modelldaten erforderlich sind [ZHOU18]. Der letzte Ansatz ist in der Einzelteil- oder Kleinserienfertigung üblich, er beruht auf der Erfahrung des Werkzeugmaschinenbedieners und ist daher nicht automatisiert.

Die oben genannten Daten und Argumente motivieren dazu, die Ausnutzung der Zerspanungswerkzeuge zu erhöhen. Um jedes individuelle Werkzeug möglichst gut auszunutzen, könnten intelligente Assistenz- und Automatisierungslösungen genutzt werden. Ziel dieser Arbeit ist es, potenzielle Methoden zu erforschen, um den Werkzeugverschleiß in der Zukunft durch den Einsatz von Inline-Messtechnik, insbesondere durch die Erfassung von Bildern innerhalb der Werkzeugmaschine, in Verbindung mit KI-basierter Bildverarbeitung, zu reduzieren. Konkret wird ein Ansatz zur Segmentierung des Freiflächenverschleißes an Schneidkanten mit einer U-Netz-Modellarchitektur vorgestellt. Darüber hinaus wird eine Untersuchung des Einflusses der Modellhyperparameter und Datensatzeigenschaften auf die Leistung des neuronalen Netzmodells durchgeführt. Auf der Grundlage der Ergebnisse wird ein Ansatz zur Erstellung eines Entscheidungsmodells für die Hyperparameteroptimierung auf der Basis von Datensatzeigenschaften für diese einflussreichsten Faktoren entwickelt. Schließlich wird der Ansatz verwendet, um ein U-Netz für einen spezifischen Datensatz zu trainieren, der mit einem Inline-Mikroskop erstellt wurde, das Bilder von Schneidkanten in einer Werkzeugmaschine erfasst.

2 Fundamentals and State of the Art

Grundlagen und Stand der Technik

This chapter gives a brief overview about the following topics: Section 2.1, Tool Wear in Metal Cutting, Section 2.2, Quantification of Tool Wear, and Section 2.3, Image Processing with Deep Learning. The chapter provides necessary background information and highlights gaps in the field of automated cutting tool wear image analysis.

2.1 Tool Wear in Metal Cutting

Werkzeugverschleiß bei der Metallzerspanung

This section contains the fundamentals that are required to understand the degradation of cutting tools in the milling process. It gives a definition of the process itself and the cutting part. Further the loads and conditions that lead to tribological effects and finally cause cutting tool wear are described.

2.1.1 Basic Concepts on the Cutting Part of Cutting Tools

Grundbegriffe zur Schneide des Zerspanwerkzeugs

Milling according to DIN 8589-3 is a cutting manufacturing process with circular cutting movement of a tool to produce workpiece surfaces [DIN8589-3] . According to the classification of manufacturing processes according to DIN 8580, it belongs to the main group 3, cutting, with the group 3.2, cutting with geometrically determined cutting edge [DIN8580]. The latter means that in a milling process the number of cutting edges and the orientation and geometry of the tool are known, as opposed to e.g., the grinding process. The flute, or tooth, of a cutting tool exhibits several surfaces that interact with the workpiece material during the milling process. As an example, the ball end milling cutter is presented on the left in Figure 2-1.

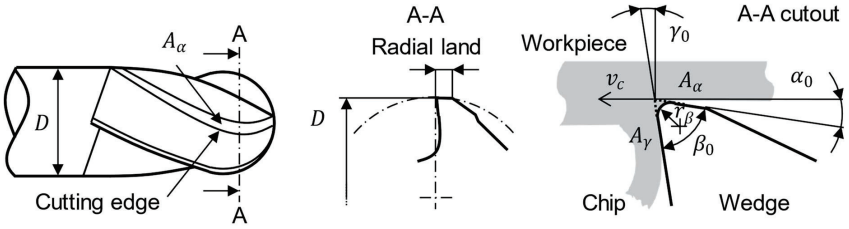


Figure 2-1: Schematic of ball end milling cutter, sectional view of a tooth and wedge of cutting tool with geometric characteristics as in [ISO8868-2, p. 6]
Schema eines Kugelkopffräsers, Schnittansicht einer Schneide und eines Schneidkeils mit geometrischen Abmaßen wie in [ISO8868-2, S. 6]

The functional part of the tool incorporating the cutting edge is called cutting part [CIRP04, p. 4]. The cutting edge is not perfectly sharp but has a radius, r_β, as shown

in Figure 2-1 on the right. The cutting edge and eventually a part of the wedge penetrate the workpiece material. The wedge is the portion of the cutting tool enclosed between the rake and the flank faces. Rake face, A_γ, is the tool surface over which the chip flows, the tool orthogonal rake angle, γ_0, is between rake face and the plane perpendicular to the cutting direction. Flank face, A_α, is the tool surface directed at the newly generated machined surface, the tool orthogonal clearance angle, α_0, is between the flank face and the cutting direction. The cutting direction is approximately the direction of cutting speed, v_c.

On the flank face there is radial land that is in contact with the workpiece surface during the cut, see Figure 2-1 [ISO3002-1]. Tool wear that occurs on the major (and minor) flank is called flank wear [CIRP04, p. 34]. Flank wear results in a loss of orthogonal clearance angle on the flank face of the tool. The interaction of tool and workpiece described above lead to different degradation effects acting on the tool. The tribology and the chain of effects resulting in cutting tool wear are described in the following subsections.

2.1.2 Tribology in Metal Cutting

Tribologie in der Zerspanung

Tribology is an interdisciplinary field that deals with the study of wear and friction problems. It is a relatively new field of science and technology, which was only established in the middle of the 20th century by Peter Jost. In the so-called Jost Report, tribology is defined as follows:

> “Tribology is the science and technology of interacting surfaces in relative motion and of related subjects and practices“ [CZIC10, p. 4]

The tools of tribology can be used to describe and optimize friction- and wear-related processes of technical systems. For a tribological analysis, the material properties and interactions of the structural elements involved as well as the load spectrum are of decisive importance, since even small deviations from the system variables result in different wear progressions [SOMM10, p. 3]. The structure of the tribological systems include the base body, counter body, intermediate materials and ambient medium [SOMM10, p. 4]. Figure 2-2 shows the schematic structure of the tribosystem.

In a machining process, the tool is considered as a basic body that is worn by a counter body, usually hard abrasive particles [KLOC18, p. 80]. The use of a cooling lubricant or minimum quantity lubrication can be regarded as an intermediate material [CZIC10, p. 565]. In interrupted cutting, which includes milling, the cutting part of the tool is subject to a strong mechanical alternating stress. This mechanical stress can be illustrated by the chip formation process. The tool cutting edge initially penetrates the material and deforms it both elastically and plastically. When the maximum allowable material dependent shear stress is exceeded, the material starts to flow and forms a chip due to the given cutting part geometry.

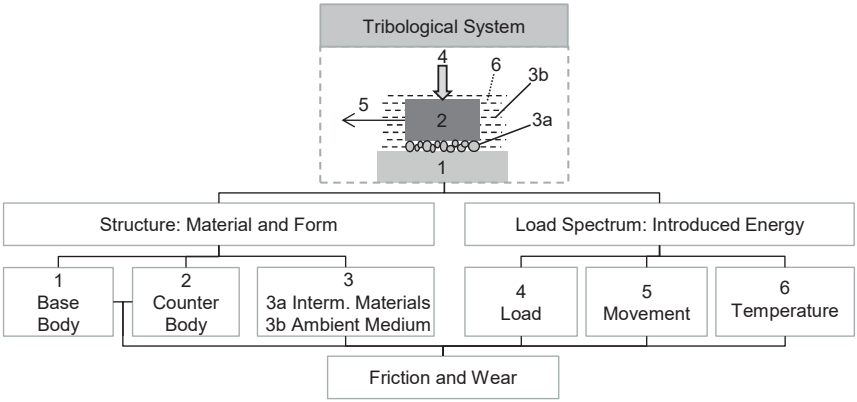


Figure 2-2: Structure of tribological system [SOMM10, p. 4]
Struktur eines tribologischen Systems [SOMM10, S. 4]

For plastic deformability, the amount and direction of the load play a decisive role in addition to the materials properties [PULS14]. This in turn means that the cutting speed, feed per tooth or cutting depth influence the amount of stress [KLOC18, p. 50]. The interrupted cut also leads to a thermal alternating stress on the tool. The tool cutting edge heats up to high temperatures during tool engagement and cools down again after exiting the workpiece. The temperature distribution depends on many factors in addition to the process parameters. For example, different cutting part geometries, material properties and cooling lubricants lead to different temperature developments [KLOC18, p. 78–83].

This means the tribology in metal cutting is a complex matter due to the high number of influencing factors. The load spectrum acting on the tool in metal cutting may be broken down into several key components, including the cutting force, cutting speed, cutting temperatur and duration of cut [CZIC10, p. 10]. It can even be further subdivided into mechanical, thermal and chemical loading as discussed by experts in the field [CIRP04, p. 32]. The following subsection elaborates on the intricate chain of effects in cutting tool wear, which originates from the comprehensive load spectrum described above.

2.1.3 Chain of Effects in Cutting Tool Wear

Wirkungskette beim Zerspanwerkzeugverschleiß

The chain of effects in metal cutting tool wear starts with the load spectrum which may be broken down to mechanical, thermal, and chemical loading (Figure 2-3). The cutting tool experiences localized loads at the cutting edge [KALP10, p. 574]. Particles and surfaces slide along between the workpiece surface and the tools flank face as described in the former subsection. Through the complex load spectrum of the cutting part, several wear mechanisms occur.

Wear mechanisms are physical and chemical interactions in the contact area of a tribological system. This triggers elementary processes, which in turn are responsible for material and shape changes of the contacting surfaces [CZIC10, p. 117]. Depending on the type and duration of load, different wear causes occur on the tool, since the cutting edges are exposed to deformation, separation, and friction processes during milling.

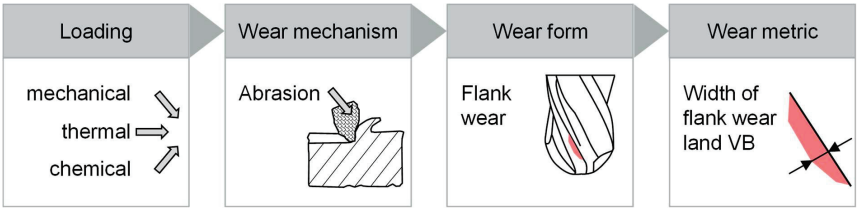


Figure 2-3: Chain of effects in metal cutting tool wear [CIRP04, p. 32–34]
Wirkungskette beim Zerspanwerkzeugverschleiß [CIRP04, S. 32-34]

In general, a distinction is made between the following mechanisms, see Annex A.1: Adhesion, abrasion, tribochemical reaction and surface disruption [KLOC18, p. 75]. Other authors seek a more granular distiction of wear mechanisms but oftentimes fail at distinguishing mechanisms and wear forms [SHAW05, p. 178]. At very high cutting temperatures, diffusion processes also occur, which reduce the wear resistance in particular and thus promote abrasion wear [KLOC18, p. 78].

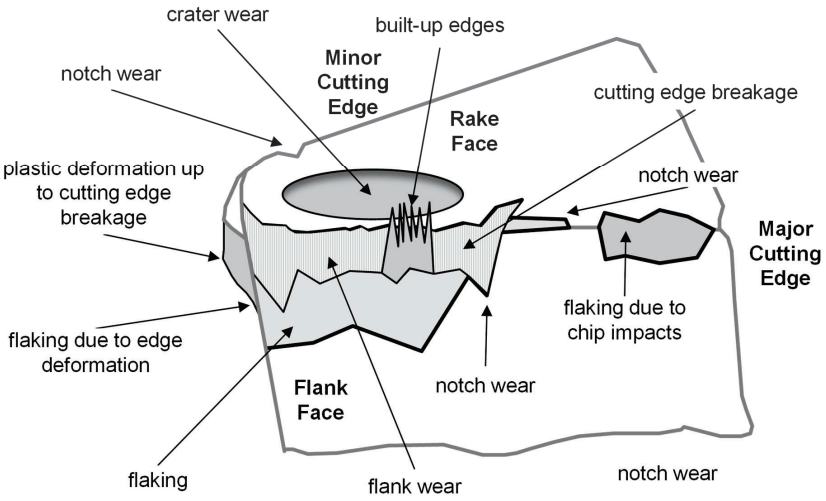


Figure 2-4: Characteristic wear forms on the cutting part in turning [KLOC18, p. 100]
Charakteristische Verschleißformen der Schneide beim Drehen [KLOC18, S. 100]

The chain of effects from loads to wear mechanisms leads to the wear forms. These phenomena must be distinguished from the causes. The tool is subject to various forms of wear, which vary in intensity depending on the type and duration of the load. Wear on the rake face is called crater wear. If wear occurs on the minor or major flank faces, this is referred to as flank wear [CIRP04, p. 32–36]. But differentiating wear forms with location of occurrence is not sufficient because of differences in tool deterioration phenomena, see Figure 2-4. Attempts have been made to classify not only flank and crater wear but also chipping, flaking, cracks and catastrophic failure [ISO8868-2, p. 8–14].

2.1.4 Impact of Cutting Tool Wear

Auswirkungen von Zerspanwerkzeugverschleiß

In a machining process, wear of the cutting tool causes a continuous change of the process state variables in the cutting zone like forces and temperatures acting on the workpiece and tool [KLOC18, p. 75]. Those continuously changing process conditions on their part influence the mechanisms that generate the tool wear and therefore affect its form and rate. Apart from workpiece and tool material properties and process conditions such as coolant and vibrations, the tool wear rate depends mainly on cutting speed [STEP16, p. 162]. Flank wear is the most influencal wear form in metal cutting [STEP16, p. 508] because the rubbing of wear land against the machined surface leads to an increase in temperature and forces which increase deflections and reduce dimensional accuracy [STEP16, p. 530]. Another effect is the displacement of the cutting edge that results in a reduced depth of cut and therefore potentially reduces cutting force and cutting temperature and hence tool wear [KLOC18, p. 4]. Flank wear also results in a loss of orthogonal clearance angle on the flank face of the tool leading to increased frictional resistance [SHAW05, p. 179]. The proportion of each of the effects described above are hard to quantify or to model for a specific operation. Generally, the tool wear and specifically the flank wear, progresses in a distinct shape over cutting time or other tool life parameters. After a break-in period, in which the flank wear rises rapidly, the curve enters the steady regime where its slope becomes constant. As soon as the first chipping of cutting-edge fragments occurs, flank wear accelerates until the cutting edge finally fails completely, that is cutting edge breakage, removal of one or more tool flutes or tool shaft shear off. The next chapter gives an overview regarding methods to quantify tool wear and to model tool wear curves for specific cutting operations.

The technical issues with tool wear described above lead to economic issues: Tool wear is a cost driver in the metal cutting industry. Besides costs for the cutting tools themselves, further costs appear - equipment downtime for tool changes, machining costs and nonproductive costs, see Annex A.2. Additionally, hard to quantify costs such as reworking of damaged surfaces, scrap parts and damages to the machine tool in the worst case [SHAW05, p. 170, STEP16, p. 529, BERG20]. Consequently, tools need to be monitored and exchanged on a regular basis, usually measured in time, tool travel path or parts produced, or at a defined tool wear state [EZUG99, WANG18].

To determine the useful tool life for a specific operation, a tool wear model is necessary. Cutting speed is the main driver for tool wear but also determines the productivity of a metal cutting process [TAYL06]. For this reason, an economical optimization of a cutting process with regards to unit costs is the search for the optimal cutting speed [STEP16, p. 751–771]. This Section 2.1, Tool Wear in Metal Cutting, gave an overview regarding the the basic concepts of the milling process and its cutting part. The former two are a tribological system that results in cutting tool wear. The complexity of the tribological system hampers the analytical or numerical calculation and prediction of cutting tool wear. The chain of effects in tool wear starts with loads acting on the tool, leading to wear mechanisms, that results in observable wear forms. The visible wear forms are quantifiable using cutting tool wear metrics, like *VB*, which may inform on the tool's condition in terms of a specific cutting operation. The subsection concludes with the impact of tool wear from a technological and economical viewpoint and the necessity of tool wear quantification for process optimization.

2.2 Quantification of Tool Wear

Quantifizierung des Werkzeugverschleißes

The quantification of tool wear is discussed in this section. Specifically, a standardized tool life testing procedure and the necessary direct measurement of cutting tool wear are elaborated. The direct and indirect approach to cutting tool wear measurement is contrasted. Furthermore, approaches for the automated processing of microscopic tool wear image data are presented.

2.2.1 Terminology of Tool Life

Terminologie der Werkzeugstandzeit

For a mutual understanding of words and their meaning in the context of a scientific subject, the terminology must be clear. There is a contradiction of terminology in one of the standards for cutting tool wear [DIN6583] in comparison to the more recent standards. In this work, the term "tool life criteria" stands for the criteria that can be used to describe the tools end of useful life. This terminology is consensus in most of the standards [CIRP04, ISO3685, ISO8868-2]. Examples for tool life criteria are tool wear, change of workpiece surface roughness or change of cutting forces. Apart from that, the framework of terminology of tool life is described using the conflicting norm [DIN6583]. Since there is no established translation of the german word "Standvermögen", the english term "cutting tool permanence" is used from now on. The following paragraphs contain a description of important terms of cutting procedure and their interconnections within the context of cutting tool wear testing and modelling. This includes explanations of the terms cutting tool permanence, cutting conditions, tool life criteria and tool life parameters. Figure 2-5 gives an impression of the relationship between the terms and their content technology-wise.

Cutting Tool Permanence

Standvermögen

Cutting tool permanence is the main term used to describe the performance of cutting and workpiece material during machining. It is defined according to DIN 6583 as the ability of a working pair, which is tool and workpiece, to withstand a specific machining process. The cutting tool permanence is an interplay of three assessment criteria: Cutting conditions, tool life criteria and tool life parameters. If two of these assessment criteria are constant, the third one can be determined through tool life testing [DIN6583]. The testing procedure is described in more detail in the next Subsection 2.2.2, Tool Life Testing in . The meaning and interaction of the assessment criteria with regards to the cutting ability and edge durability is explained in the following paragraphs.

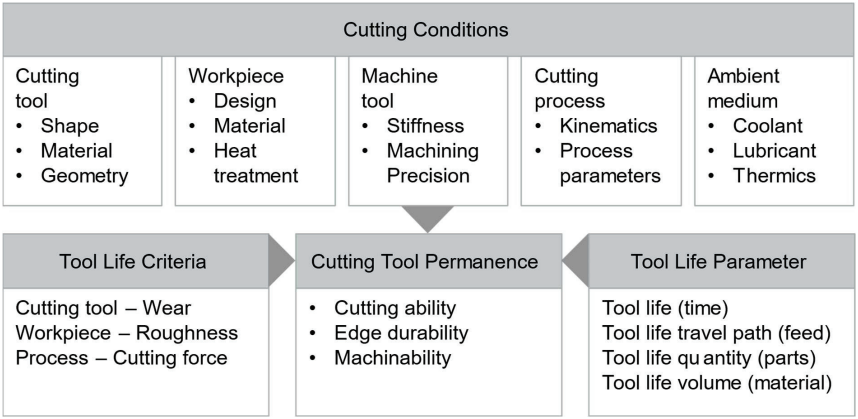


Figure 2-5: Terms of cutting procedure - Tool life terms [DIN6583]
Begriffe der Zerspantechnik – Standbegriffe [DIN6583]

Cutting Conditions

Standbedingungen

The cutting conditions in a cutting operation respectively in cutting tests consists of several components and their properties (Figure 2-5): Cutting tool, workpiece, machine tool, cutting process and ambient [DIN6583, p. 2].

The properties of the cutting tool that influence the permanence are among others, geometry of the cutter, geometry of the cutting part, the tools material and the tools coating. The temperature in the cutting zone has mostly the strongest influence on tool wear [AUGS18, KLOC18]. That is why the cutting speed, which is a proxy for cutting temperature, is often used as the independent variable for various approaches of modelling tool life [GRZE17, p. 224–227].

Tool Life Criteria

Standkriterien

Tool life criteria are threshold values for undesirable alterations of cutting tool, workpiece, or process state [DIN6583, p. 2]. To determine an end of life for cutting tools, either for the sake of process safety or part quality, the respective standards recommend the use of the type of deterioration that is believed to be most important to the end of useful tool life. Once the predominant wear form with high informative value is known, a wear metric can be derived. Depending on the operation type and its specifications, it may also be flank wear, crater wear, flaking or chipping. Based on occurrence in literature flank wear is popular, which may be measured with the tool-life criteria width of flank wear land *VB*. This most commonly used criterion can be further specified to uniform wear, i.e. *VB* averaged over all teeth, or localized wear, i.e. maximum *VB* on an individual tooth [ISO8868-2, p. 14, STEP16, p. 537]. In case of turning operations the standard breaks *VB* down into local zones along the cutting edge for a finer differentiation [ISO3685, p. 12]. Apart from a specific tool wear form and its respective wear metric, the cutting force or surface roughness could be used as a tool life criterion [DIN6583, p. 1]. Since the latter two are still more elaborate to quantify, the tool wear is the most common criterion for tool life determination.

Tool Life Parameter

Standgröße

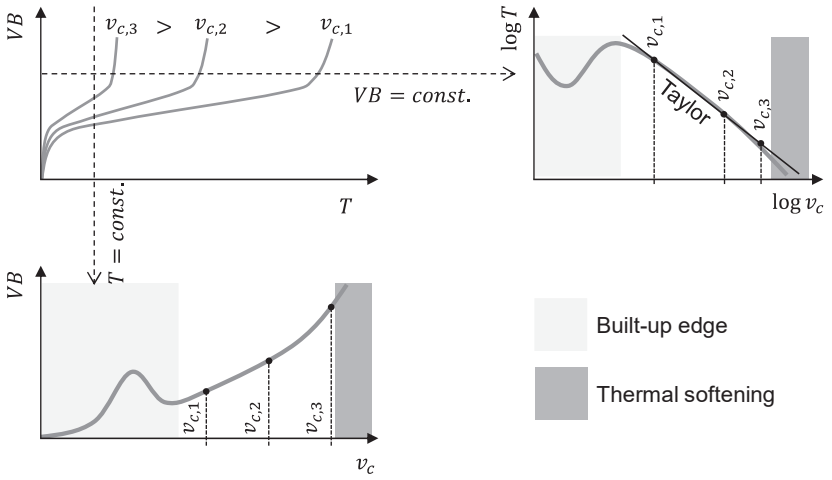


Figure 2-6: Qualitative plot of *VB* over tool life time, double-logarithmic *vT*-diagram and *VB* over cutting speed [TAYL06, KLOC18, p. 82]

Qualitative Grafik zu VB über die Schnittzeit, doppellogarithmisches vT Diagramm und VB über der Schnittgeschwindigkeit [TAYL06, KLOC18, p. 82]

For practical reasons in production environments, a tool life parameter is of interest to determine tool life time, travel path, volume or the quantity of produced parts until a certain tool life criterion is reached [CIRP04, p. 40, DIN6583, p. 2]. The tool life time is a common tool life parameter.

The spread of this parameter is explainable due to its practical use in manufacturing, especially turning, since TAYLOR introduced the systematic investigation and modelling of tool life using *vT*-curves [TAYL06]. The *vT*-diagram is a characteristic curve or diagram describing tool life time, T, as a function of cutting speed, v_c, in a log-log plot (Figure 2-6). Nevertheless, the diagrams may also be presented using other tool life parameters such as tool life travel path. It is also possible to derive a VB-v_c diagram using intersections of tool wear curves at constant cutting time, see Figure 2-6. The *vT*-diagram is valid in a range that is specific to the process investigated. This range is usually constrained by Built-up Edge (BUE) formation at low cutting velocities and capped by thermal softening of the work material towards higher cutting velocities.

2.2.2 Tool Life Testing in Metal Cutting

Standzeituntersuchungen in der Zerspanung

As noted in Section 9.1, tool life depends as much on part requirements as on the tool material and cutting conditions, making it difficult to develop general methods of predicting tool life [STEP16, p. 549]. The standard for tool life testing in milling for end milling cutters has been derived from the ISO 3685 for tool-life testing with single-point turning tools [ISO8868-2, p. 1, ISO3685]. The standard applies to end milling operations for high-speed steel. Since the general procedure is not different for carbide tools, this standard serves as a framework for tool life testing. The purpose of tool life testing can be manifold:

- To benchmark for example tool coatings, cutting fluids, tool materials, workpiece materials or tool geometries with regards to their effect on tool life.
- For investigations regarding favorable cutting parameters or cutting conditions for a specific operation
- To determine the useful end of life for a specific working pair consisting of workpiece material and cutting tool material.
- In scientific qualification of a cutting process there are four criteria for evaluation of machinability: tool wear, chip form, cutting force and surface quality.

ISO8868-2 covers five distinct types of tests, their purpose and number of investigated variables are shown in Table 2-1. Before carrying out the test runs for the tool life testing, a preliminary test is recommended to select a useful range of cutting speeds, feed values and time intervals between tool wear measurements. For the assessment of tool deterioration, e.g. the width of flank wear land measurement, the standard recommends the application of a toolmakers microscope and a mounting device for the cutting tool [ISO8868-2, p. 15].

Table 2-1: Types of tests for tool life testing in machining as in [ISO8868-2]
Testtypen für Standzeitversuche in der Zerspanung [ISO8868-2]

Type	Purpose	Vars.
A	Benchmarking of two or more process specifications	0
B	A characteristic *vT*-curve with variable cutting speed	1
C	Same as B but with variable feed, otherwise constant parameters	2
D	Same as C but with variable axial and radial depth of cut	3-4
E	Machining characteristics such as cutting forces, machined surface, and chip formation for one set of process parameters	1

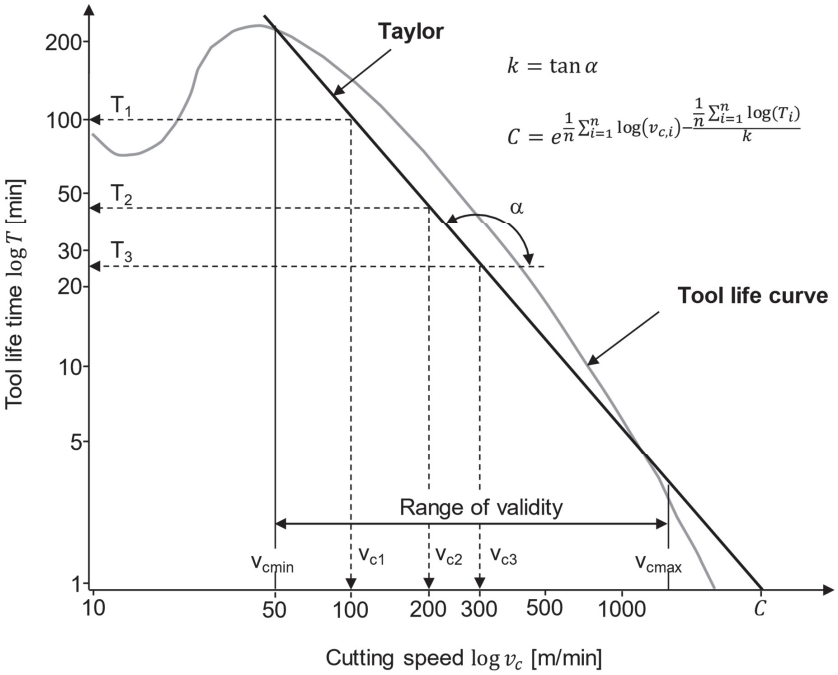


Figure 2-7: Log-log characteristic tool life diagram also called *vT*-diagram
Doppellogarithmisches Standzeitdiagramm, auch vT-Diagramm genannt

Once the cutting test have been conducted in accordance with, e.g., test type B requirements, the statistical evaluation of tool life data follows. Before constructing characteristic diagrams, the statistical significance of differences between the test

conditions must be proven. In the calculations the variable *x* represents a tool life parameter such as tool life time or tool travel path. First, the arithmetic mean, μ, and standard deviation, σ, of each test condition is determined with the number of test-runs within one test condition, *n*.

$$\bar{x}_{max/min} = \mu \pm t\frac{\sigma}{\sqrt{n-1}} \tag{1}$$

The boundaries of the confidence interval, $\bar{x}_{max}$ and $\bar{x}_{min}$, as in Equation (1), indicate where further tool life results would be located with an assumed probability. The students $|t_\alpha|$ value for determining significant difference is calculated using the following formula in Equation (2), where n_A and n_B are the number of test runs for two of the compared conditions *A* and *B*:

$$|t_\alpha| = \frac{(\bar{x}_A - \bar{x}_B)}{\sqrt{\frac{n_A \times \sigma_A^2 + n_B \times \sigma_B^2}{n_A + n_B - 2} \times \left(\frac{1}{n_A} + \frac{1}{n_B}\right)}} \tag{2}$$

If $|t_\alpha|$ is greater than $t(n_A + n_B - 2)$ at the chosen confidence level a significant difference exists between the two considered test runs *A* and *B*. After a significant difference between the runs of all test conditions is proven, the calculation of tool life curves for the characteristic diagram is performed (Figure 2-7), which is a natural log-log-chart originating from TAYLOR [TAYL06].

The regression analysis requires the number experimental observations, *n*, the independent variable for regression, *x*, which is usually the natural logarithm of cutting speed and the dependent variable, *y*, which is usually the natural logarithm of tool life time. The best straight line shall be fitted to the graph of *x* and *y*. The tangent of the angle between the x-axis and the regression line is *k*. The theoretical intersection of x-axis and the regression line is *C*. The calculation of the Taylor tool life parameters is complete. The Taylor equation may be used for determining cutting time at a specific cutting speed, as in Equation (3), or the cutting speed required for a targeted tool life time, as in Equation (4). Additional statistical indications for the goodness of fit can be calculated, such as dispersion, significance and confidence interval limits for the line or the individual constants [ISO3685].

$$T = e^{k \cdot \log(v_c/C)} \tag{3}$$

$$v_c = e^{\frac{\log(T_c)}{k} + \log(C)} \tag{4}$$

This subsection covered the techniques for tool life testing and modelling as suggested by the standards in the cutting tool wear domain. From an economic perspective, costs for tool life testing may exceed the value of the results [ISO3685]. This means there is a need for more efficient methods for tool life testing and the tool wear problem in metal cutting in general. A weak point is the complex test procedure and especially the direct microscopic wear measurement which will be covered in the next subsection.

Other attempts to tool life modelling apart from the Taylor-based methods presented here, may be found in textbooks and publications [SHAW05, p. 171–177] [GRZE17, SOMM10, HOLS21]. Some of them consider wear mechanisms and build formulas around physical equations for these mechanisms. Others rely on numerical simulations and feedback the information into tool life equations. A commonality of all approaches to tool life modelling is their requirement for fundamental material scientific experiments or empirical cutting experiments, mostly to a greater extent than the Taylor based approach.

2.2.3 Direct Tool Wear Measurement

Direkte Werkzeugverschleißmessung

In the realm of measurement techniques for cutting tool wear measurement, there are two basic categories: direct measurement using, for example optical devices, and indirect measurement, utilizing condition monitoring sensors.

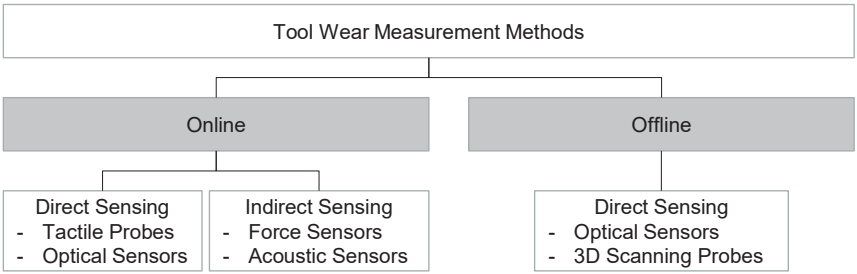


Figure 2-8: Examples of different tool wear measurement methods
Beispiele für unterschiedliche Verschleißmessmethoden

Direct measurement means directly observing and quantifying wear on cutting tool edges with optical sensors which tend to give higher accuracy compared to indirect methods [JEON88]. Direct measurements offer a direct view of the tools condition, allowing for the identification of various wear forms and their extent. On the downside, direct measurements are confined to line-of-sight observations. Cutting fluid or chip accumulation, ambient lighting conditions and vibrations can obstruct the optical view. Uncertainties arise from the measurement process and the interpretation of image pixel data by human operators who manually determine a metric, like the width of flank wear land *VB* [CIRP04]. Some direct sensing methods, such as 3D scanning, cannot yet be performed “Online”, i.e., parallel or in temporal proximity to the process, economically. In manufacturing companies, it is widespread practice to inspect cutting tools with a lense and let the operator assess the tool condition based on experience. Another possibility is the laser bridge to measure cutting edge set back. Tool life measurements and especially a high number of measurements that are necessary for testing are elaborate. Especially the requirement for manual, direct measurements of cutting tool wear may hinder an efficient creation of tool life or indirect monitoring models.

Indirect measurements, in contrast, are made by taking a substitutional value that has a particular relation to the actual measurand [JEON88]. Indirect measurements do not require direct access to the cutting zone. This enables measurements to be taken without interrupting the ongoing machining process. However, the measurements from sensors can be influenced by external factors, such as machine dynamics, making the interpretation of data more complex and potentially less reliable. Additionally, certain wear patterns might not be accurately represented through indirect measurements alone, introducing the risk of missing critical information. Finally, indirect measurement methods rely on frequent tuning by a direct measurement, especially when process variables change.

Figure 2-9: Direct offline measurement of cutting tool wear using a microscope
Direkte Offline-Messung von Zerspanwerkzeugverschleiß mit Mikroskop

2.2.4 Computer Vision for Automated Tool Wear Detection

Computer-Vision zur automatisierten Detektion von Werkzeugverschleiß

Since direct measurement of wear on cutting tool edges using optical sensors and human interpretation are elaborate, attempts have been made to automate the analysis of microscopic tool edge images with regards to tool wear using Computer Vision (CV) algorithms. Rule-based feature detectors for image processing, like sobel, canny and the active contour method [CANN86, KANO88, KASS88], are widely applied in literature to detect tool wear on cutting tool edges [D'AD17, ALEG09, MOLD17]. Another approach is the use of machine learning methods solely or in combination with feature detectors [D'AD13, DHAN15, XION10].

Common CV algorithms are transparent, compute efficient and optimized for specific tasks, while Deep Learning (DL) methods can be used for versatile environments, given the training data reflects the variance [MAHO20]. Some typical disturbances in an industrial environment with metal cutting processes are changing light exposure, different coating colors, changing orientation, blurry image acquisition conditions due to fluids or tool macro geometry, cold welded chips that disturb the view of the actual cutting edge and are difficult to differentiate from flank wear, dirt and changing tool geometries. A typical failure mode of CV for cutting tool wear detection is shown in Figure 2-10. This failure occures due to fixed thresholds that result in a high sensitivity towards variance in brightness for example. In case of a DL approach this weakness is adressed through providing training data that reflects real or artificial changes in brightness in the images.

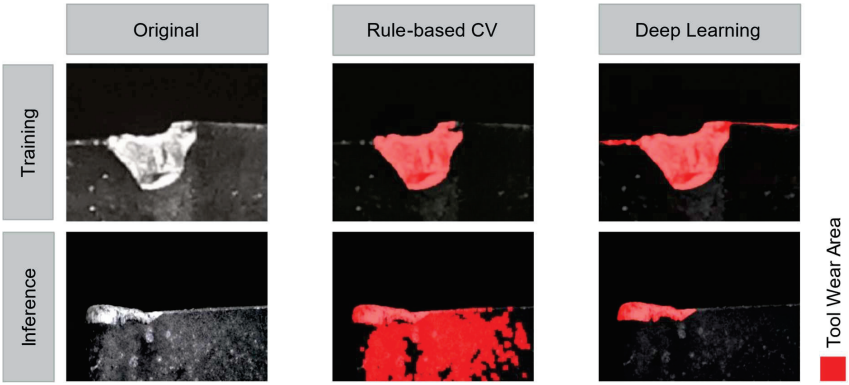


Figure 2-10: Rule-based CV fails as opposed to DL when applied to inference data
Regelbasierte CV versagt im Gegensatz zu DL angewandt auf Inferenzdaten

2.3 Image Processing with Deep Learning

Bildverarbeitung mit Deep Learning

Traditional CV methods have difficulties to fulfil the task of tool wear identification on microscopic images of cutting edges when there are small deviations compared to the image they were tuned on, see figure above. Although it is possible to use CV for this task in very restricted and homogeneous use cases, the spread in the industry has so far failed to materialize with this approach. Deep learning approaches promise to handle heterogeneous data better than classical CV algorithms [GREE16].

A literature review of the Artificial Intelligence (AI) topics relevant for this thesis is discussed in this section. The major topic of discussion is Fully Convolutional Networks (FCN), which are applied for semantic segmentation of images, and the appropriate background that led to their development. Firstly, an introduction to Artificial Neural Networks (ANN) is given in Subsection 2.3.1 followed by a description of the NN training process and the hyperparameter tuning process in Subsection 2.3.2. In the following Subsection 2.3.3, Convolutional Neural Networks (CNN) and Image Processing with Artificial Intelligence are discussed thoroughly. Subsequently, Tool Wear Identification with Deep Learning with a focus on FCNs is described in Subsection 2.3.4. FCNs are a type of network that is designed for dealing with image data and pixelwise classification specifically. The very network used for the task of semantic image segmentation in the cutting tool wear use case is an architecture which was initially designed for medical image processing. Specifically, for identification of pathological tissue for data from medical imaging techniques [RONN15]. This underscores the versatility of FCNs, bridging the gap between seemingly disparate domains and showcasing their adaptability in solving complex problems.

2.3.1 Fundamentals of Machine Learning

Grundlagen des maschinellen Lernens

Machine Learning (ML) is a branch of artificial intelligence that systematically applies algorithms to generate underlying relationships between data and information [AWAD15, p. 1]. General application areas of machine learning are, for example, speech recognition, next word prediction or weather forecasts [WITT19, p. 24–30]. In contrast to classical statistics which is based on probability theory and focusses on estimating parameters from a sample of data, machine learning takes a more flexible approach to modelling data and often involve more complex algorithms. ML is divided into three main categories depending on the type of learning: Supervised learning, unsupervised learning, and reinforcement learning. Within this thesis only the first one is used, since it requires high accuracy evaluation results [WANG20]. In supervised learning, the algorithm learns from training data, which is divided into input and output data. The inputs and outputs form pairs and must be determined before training. An algorithm finds rules to put these pairs into relationships. An example of such modeling is the regression method, in which a linear or nonlinear relationship between two variables can be represented as a function. With the help of regression, the parameters of this function are determined, and a model is created which allows predictions of the output variables [GÉRO19].

Artificial Neural Networks or simply Neural Networks are designed by taking motivation from their biological counterparts and thus are a crude approximation of neural systems existing in nature [ROSE58, MCCU43]. The perceptron, also called artificial neuron, is the simplest Neural Network (NN) existent, see Figure 2-11 left. It emerged as an artificial abstraction of a real neuron and takes one or more input numbers, denoted x_i, connected via weights, denoted w_i, to the neuron itself which contains a summation and an activation function f, generating an output, y. The bias term, b, shown in Figure 2-11 may shift the activation function left or right on the x-axis to allow offsetting it to positive or negative values.

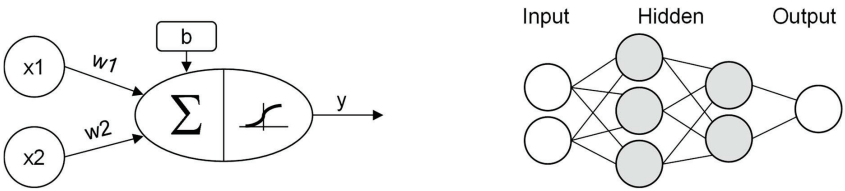


Figure 2-11: Perceptron (left) and Multilayer Perceptron (right)
Perzeptron (links) und Multilayerperzeptron (rechts)

The perceptron in the figure has an input layer and a hidden layer that functions as output layer at the same time. Equation (5) gives the mathematical expression of a perceptron [RUSS04, p. 896].

$$y = f\left(\sum_i w_i x_i + b\right) \tag{5}$$

The Multilayer Perceptron (MLP) is a wiring of several perceptrons, into a network with at least one hidden and one output layer, each having possibly several perceptrons in parallel, see Figure 2-11 right. A single perceptron can only learn simple tasks, but for problems that are more complex multilayer perceptrons are required. A NN with two or more hidden layer is called a Deep Neural Network (DNN), see Figure 2-11. For simplification, each circle shown below represents an artificial neuron with summation and activation function to process the weighted inputs shown as arrows.

Activation Functions
Aktivierungsfunktionen

Activation function layers are applied after hidden or outputs layers of a network. They calculate weighted sums of inputs and biases. In CNNs the most commonly used activation function following a hidden layer is the Rectified Linear Unit (ReLU) due to its performance improvements compared to nonlinear activations like Sigmoid, its fast computation and its property of preventing vanishing gradients problem which otherwise increases training time [NWAN18, VINO10].

For ReLU, all non-positive values are changed to 0 and the positive values scaled linearly. At activations below zero, the gradient will be 0, hence the weights will not get adjusted during descent. Therefore, neurons stop responding to variations in input, this is called the dying ReLU problem. Exponential Linear Unit (ELU) is an activation function that solve the dying ReLU problem through a factor, which is smoothly reached below activations of zero. ELU tends to produce more accurate results across different learning rates compared to ReLU [PEDA18]. The sigmoid activation function produces and output value that is scaled between (0,1). Due to this squashing behavior, it is commonly used for predicting probabilities for example in the output layer of a neural network to produce a classification, see Figure 2-12.

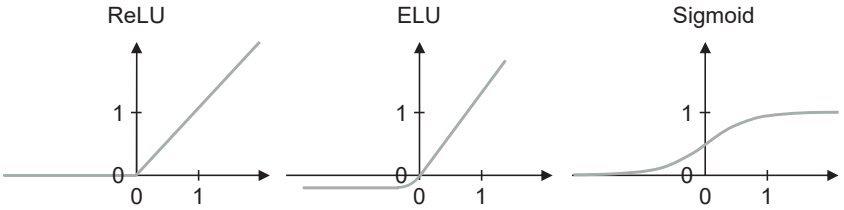


Figure 2-12: ReLU, ELU and Sigmoid activation functions
ReLU, ELU und Sigmoid Aktivierungsfunktionen

2.3.2 Neural Network Training

Training eines neuronalen Netzes

This section gives a brief introduction to basic concepts with regards to NN training and its related challenges.

Backpropagation

Fehlerrückführung

An efficient method for calculating gradients that are needed to carry out optimization of weights is backpropagation [RUME86]. The solution of the resulting optimization problem is a particular set of weights which minimise or maximise the objective function. The loss function should be continuously differentiable since this method requires the calculation of gradient of the objective function at every iteration step. The network weights are randomly initialised within a predefined range, for example between (-1,1). At this point, the network cannot make meaningful predictions for input because no functional mapping is present between the input and its labeled output. The weights are updated as the input and the respective labeled output class are fed into the network while training. In the following, the two passes (forward and backward) required for training a neural network are described. The process of feeding the output of one layer as the input to the next layer in a forward direction is referred to as the forward pass. To achieve the goal of minimising loss, the weights are modified by taking derivatives for all weight values present in the network starting from the output and moving backward. This process of updating the weights in the negative direction of the loss function gradient in a backward manner is known as backward pass.

In short: During the network training, with the backpropagation algorithm, one row of data is passed as input through the network [LECU88]. The produced output is compared to the true output yielding an error. The error is propagated back through the network, layer by layer, updating the weights to the amount they contribute to the error. This way all rows of the dataset train the network repeatedly to produce an abstract representation of the data through weights in the NN.

Optimization Function

Optimierungsfunktion

The optimisation function is used for minimising or maximising the objective function or the loss function. The most common optimisation function and the one used in this thesis is the Stochastic Gradient Descent (SGD). It calculates an approximation of the true gradient using only one or a subset of training dataset. This method minimises the loss function $L(\theta)$ parameterized by θ.

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta L(\theta) \tag{6}$$

To decrease the value of the loss function the parameters θ are updated in the non-positive gradient direction of the loss function $\nabla_\theta L(\theta)$. The step size is determined by

the learning rate η. The iteration of Equation **(6)** stops when a local or global minimum is reached. SGD is known as mini-batch SGD when using only a subset of training dataset.

Objective Function
Zielfunktion

Objective function also known as loss function L is calculated as the difference between the output image obtained after propagating it through the network and the user labelled output image. Two commonly used loss functions are:

The Quadratic Loss Function or the Mean Squared Error (MSE) loss function, see Equation (7), is one of the simplest loss functions used in training neural networks. With x_i being the neurons output, $\hat{x}_i$ the desired neurons output and N the number of training images.

$$L = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{7}$$

The Cross Entropy Loss Function, as shown in Equation (8) is commonly used in training convolutional neural networks [ZHAN18]. Loss is calculated during network training and validation and its interpretation shows how well the model performs for these two sets.

$$L = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i \ln(x_i) + (1 + \hat{x}_i)\ln(1 - x_i)) \tag{8}$$

Unlike accuracy, loss is not a percentage. It is a sum of the errors made for each example in training or validation sets. An accuracy metric is used to measure the algorithms performance in an interpretable way.

Hyperparameter Tuning
Hyperparameter-Abstimmung

The performance of neural networks is largely affected by the hyperparameters chosen. As explained previously in the paragraph **Backpropagation** neural network training is achieved by an optimization process. The hyperparameters include for example the above-mentioned **Optimization Function** and **Objective Function** (or Loss Function). The tuning of hyperparameters is a significant and complicated part of neural network training [YU20]. Generally, tuning is based on experience gained while training these networks (for example as rule of thumb the learning rate is set as 0.0001) rather than any established theory. The process is not straightforward because the space of possible hyperparameter settings is extremely large. Nevertheless, it is recommended to investigate hyperparameters systematically in a design of experiments and use inference data from the real world domain for model assessment if possible [D'AM20]. These measures aim to prevent so called underspecification, which expresses via

good model performance of ML pipelines on test sets but model failure in the real-world application, i.e., inference domain. The evidently present underspecification in various applications of deep learning image processing use cases implies, that small variations in hyperparameters or training setup (e.g. random seed) may produce models that succeed in training and testing but fail on slightly different real world data [D'AM20].

2.3.3 Image Processing with Artificial Intelligence

Bildverarbeitung mit KI

For image processing tasks the above-mentioned, so-called Fully Connected (FC) networks can be applied but they have a major downside concerning training time. A low-resolution input greyscale image of (32 x 32) pixels with 1024 neurons in the first layer and the same amount in two hidden layers leads to more than two million trainable parameters considering weights and biases. Scaling this up to relevant architectures with several layers and higher image resolution, the approach gets infeasible. Therefore, FC networks are not scalable for image processing tasks [RASC18].

Deep learning models for image processing tend to generalize well when enough data is present for the training process. Also, these models have proven to defeat traditional approaches in the image processing challenge ImageNet since 2012 [KRIZ17]. Augmentation methods allow to enlarge the database while bringing artificial variations into the dataset, such as orientation, light conditions, contrast, etc. making the models more robust to changes in the acquisition environment. Tool wear detection is a texture recognition task rather than an object recognition task. A recent study reveals, that CNNs trained on the ImageNet dataset, which contains e.g. cat pictures, are biased towards recognizing textures and not object shapes as previously thought [GEIR19]. This means CNNs probably could recognize texture variations coming from wear phenomena on metal cutting tools.

Convolutional Neural Networks

Faltendes neuronales Netzwerk

Image classification is discussed before as it is a precursor of image segmentation. Image classification aims to label entire images whereas image segmentation labels all pixels within an image. In 1998, while researching and developing neural networks LeCun et al. published a paper citing useful applications of Convolutional Neural Networks (CNNs) in document recognition [LECU98]. In 2012, CNNs regained spotlight as Krizhevsky et al. improved the CNN architecture, mainly through a much deeper architecture of layers. Over the last few years DL for CV tasks has garnered much popularity and new architectures are being published frequently [META22].

CNNs have applications in object detection, text recognition, pose estimation and many more [ALOY17]. They contain several different hidden layer types that bring a solution for the efficiency problem in image processing described earlier. The first CNN, as mentioned above, applied for digit recognition is called LeNet-5. It takes (32 x 32)

greyscale images, has roughly 340,000 connections but only 60,000 trainable parameters [LECU98]. An explanation how this is possible is briefly described in the following paragraphs. For a better overview, Figure 2-13 shows a CNN architecture conceptually.

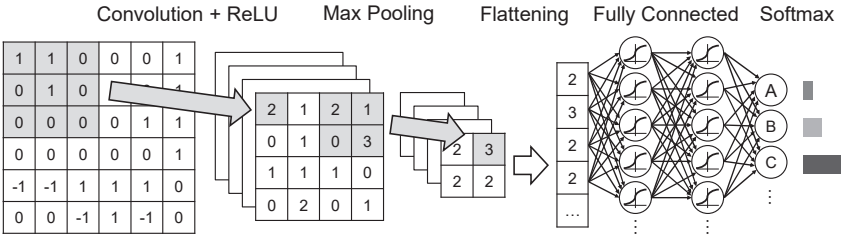


Figure 2-13: Schematic presentation of a simple CNN architecture
Schematische Darstellung einer einfachen CNN-Architektur

Convolution Layer

Faltungsschicht

This layer extracts features from an image, e.g., lines and dots, and compresses the image. Using a filter, also called kernel, which slides along the input image, a smaller representation of an image is created (Figure 2-14). An image is a matrix containing color information (numbers ranging 0-255) in the form of color codes and may be formalised as a matrix with heigth, width and depth (*h* x *w* x *d*). Convolutional layers generate a stack of feature maps from input images using kernels (or filters). For a kernel *K* with *a* rows, *b* columns and *d* depth, the notation can be formalised as (K_a x K_b x *d*). This kernel has a receptive field of (K_a K_b). The kernel produces a feature map by sliding over the image in a linear fashion. This process is repeated with different kernels to produce different feature maps over the same receptive field of the image.

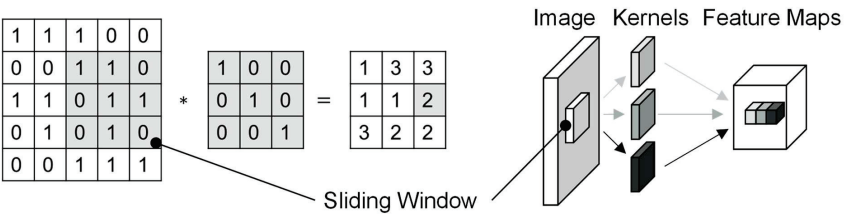


Figure 2-14: Convolution operation (left), convolutional layer schematic depiction with image, kernel and feature maps (right)
Faltungsoperation (links), Faltungsschicht schematische Darstellung mit Bild, Kernel und Feature Maps (rechts)

The sliding over images is referred to as the convolution operation and is mathematically defined as the sum of element-wise multiplication of a kernel and the original

image, holding the depth constant. The number of pixels by which the kernel slides after each convolution is called a stride. The stride *s* of a kernel is a hyperparameter which needs to be specified at training time. The output generated by a convolutional layer reduces its size due to the nature of convolution operation. Further, using a stride greater than one results in an output that has even lesser rows *h* and columns *w*. For example, applying a (3 x 3) filter with (1 x 1) stride, which means one pixel each step, on a (6 x 6) image yields an output, called feature map, of size (4 x 4) that means a complexity reduction of more than 50 %. Handcrafting of kernels is possible, exemplarily the diagonal line detection in Figure 2-14. CNNs learn filters during the training process automatically in context of the network task, respectively the training data. In a CNN there are several filters applied in each convolutional layer leading to several feature maps stacked upon each other. Due to the kernel sizes, closely located pixels relationship is preserved by the convolution operation, which is beneficial for image data where context of nearby pixels holds information about the content [CHOL18].

Pooling

Pooling

The two major reasons for applying pooling layers are decreasing the number of network weights and reducing overfitting to training data [SCHE10]. Pooling layer is calculated by taking a particular value of input within a kernel based on a specified metric (maximum, minimum, etc.). This results in an output size that is smaller than the input. The pooling method most used is max pooling (Figure 2-15). Other methods include average pooling and L2 norm pooling. The strides of the pooling kernel are equal to its length, e.g., a (2 x 2) kernel has a stride of *s* = 2. Thus, max pooling ends up reducing the spatial dimension of the input. The use of pooling layers has been found to be particularly effective in CNNs applied to image and speech recognition tasks, where reducing the input size while preserving key features is crucial for efficient processing.

Figure 2-15: Max pooling operation (left), average pooling operation (right)
Max pooling Operation (links), Average pooling Operation (rechts)

Regularization

Regulierung

Overfitting occurs when a ML model is too complex and learns to fit the noise in the training data, resulting in poor performance on new, unseen data. Regularization techniques aim to address this problem by reducing the complexity of the model, preventing

over-reliance on specific features, and encouraging the learning of more generalized representations of the input data.

Dropout is a popular regularization method that addresses this problem by randomly deleting neuron outputs during training [SRIV14]. This technique helps to prevent over-reliance on specific features and encourages the model to learn more generalized representations of the input data.

Another widely used regularization method is **Batch Normalization** (BN) [IOFF15]. This method involves normalizing and scaling neuron outputs to a mean of zero and a standard deviation of one. This normalization step helps to stabilize the learning process and reduce the internal covariate shift. BN has been shown to significantly improve the performance of deep learning models and is now a standard component of many state-of-the-art architectures.

Flattening and Fully Connected Layer
Abflachende und vollständig verbundene Schicht

The Flattening Layer is one such layer that is used to reshape the 3D arrays produced by the convolutional layers into 1D vectors. This layer is necessary because most FC layers in NNs require inputs in the form of 1D vectors.

The Fully Connected (FC) layer is another important layer in CNNs. This layer takes the flattened output from the previous layer and applies a set of weights and biases to learn non-linear combinations of the features. This allows the model to capture more complex relationships between the inputs and outputs and improve its ability to make accurate predictions. This FC layer can be seen as an MLP, see subsection 2.3.1 Fundamentals of Machine Learning, because it consists of one or more layers of fully connected neurons that learn to model the non-linear relationships between the input features and output classes.

In summary, the Flattening Layer is required to reshape 3D arrays into 1D vectors, te FC layer to learn non-linear combinations of the features coming from the convolutional layers, see Figure 2-13.

Fully Convolutional Neural Networks
Vollständig faltbare neuronale Netzwerke

With the popularity of DL in recent years, many semantic segmentation problems were addressed using architectures, like CNNs. This approach outperforms others in terms of accuracy and efficiency. Semantic segmentation means that, instead of classifying an image or an object in an image, each pixel in the image is labelled. This enables scene understanding for autonomous driving and analysis of biomedical images for identification of pathological structures [MENN18, LUND19].

The general architecture for semantic image segmentation tasks is an encoder, which is often a pre-trained CNN for image classification and a decoder network that classifies each pixel from the features learnt by the encoder. An example for this approach

to semantic segmentation is Fully Convolutional Networks (FCN) [LONG15]. In 2015, the concept of FCNs as a modification of CNNs for the application of pixel-wise semantic image segmentation is introduced. The Fully Connected (FC) layers of CNNs are replaced by convolutional layers in an FCN. This specific architecture of neural networks used for semantic image segmentation captures both the global and local information present in images. It does so through its two major parts: a downsampling path to capture contextual information and an upsampling path to recover spatial information. Vanilla CNNs, i.e., networks for image classification tasks, have a drawback of losing information about location of classes due to the FC layer at the end that outputs a single class. However, FCNs can predict locations per class. FCNs take, other than classical CNNs, arbitrary sized images since they do not make use of FC layers.

U-Net Architecture

U-Net-Architektur

U-Net is designed for biomedical image segmentation purposes. In 2015, the International Symposium on Biomedical Imaging (ISBI) cell tracking challenge in San Francisco was won by this network. U-Net is an FCN based architecture that learns to segment images in an end-to-end setting which means it takes in a raw image and puts out a segmentation map or mask [RONN15]. Note that it does not use any FC layer. As consequences, the number of parameters of the model is reduced and it can be trained with a comparably small dataset [RONN15].

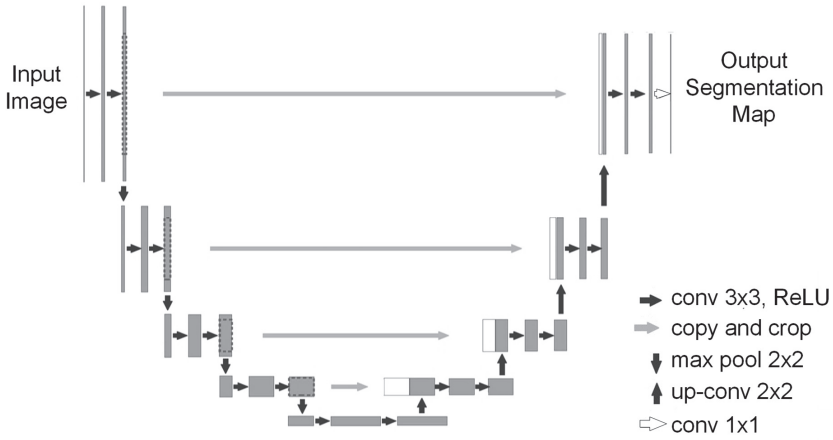


Figure 2-16: U-Net architecture [RONN15]
U-Net-Architektur [RONN15]

The network consists of symmetric encoder and decoder layers connected. Moreover, the U-Net architecture has shown to be highly flexible and easily customizable, allowing users to adapt the model to their specific research needs. Another important benefit

of the U-Net architecture is its efficient use of memory and computational resources, making it suitable for use on standard computing hardware [RONN15]. The input image is fed into the network at the beginning and the data propagated through the network layers along all possible paths resulting in a segmentation map. The layers are stacked such that they form a U-shape, hence the name U-net. This architecture combines lower and higher-level feature maps with skip connections, thus improving high resolution features with localization.

Each grey box corresponds to a multi-channel feature map. The size and number of featured channels are denoted in the Figure 2-16. Most of the operations are convolutions followed by a nonlinear activation function. In detail, there is a standard (3 x 3) convolution followed by a Rectified Linear Unit (ReLU). The next operation in the U-Net is a max pooling operation which reduces the size of the feature map as illustrated by a red downward arrow. The max pooling operation acts on each channel separately and propagates the maximum activation from each (2 x 2) window to the next feature map. All these sequence of convolutions and max pooling operations result in a spatial contraction where there is a gradual increase of “what is in the image” and at the same time decrease of “where is it in the image”. [BERG20]

CNNs for classification tasks end after the contraction and map all features to a single output vector. U-Net has an additional expansion path to create a high-resolution segmentation map. This expansion path consists of a sequence of upconvolutions and concatenation with the corresponding high-resolution features from the contracting path. This upconvolution uses a learned kernel to map each feature vector to the (2 x 2) pixel output window again followed by a nonlinear activation function that outputs the segmentation map. Accurate segmentation can be obtained with relatively small training data sets. The final resolution of the output map is increased by the deconvolutional layers. Many feature maps per convolutional layer are applied in the encoder and decoder layers, thus resulting in a transfer of contextual information to higher resolution layers.

Other notable architecture that could be used for tool wear detection in the future include Regions Based CNN (R-CNN) that has three output branches to split the tasks of localization, classification and segmentation [HE17]. Other architectures that may be used for the task aim at context awareness, such as DeepLabv3+ and Pyramid Scene Parsing Network [CHEN18, ZHAO16]. These networks did very well on the Cityscape dataset for autonomous driving, scoring above 80 measured in *mIoU* [CORD16]. Finally, there are transformer architectures originating from natural language processing. Researchers have proposed various transformer-based architectures for semantic image segmentation, which typically consist of an encoder-decoder structure. One transformer architectures that is adapted for segmentation tasks is TransUNet, which combines the U-Net architecture with transformers for improved medical image segmentation [CHEN21].

Performance Metrics for Semantic Image Segmentation Models

Leistungsmetriken für Modelle zur semantischen Bildsegmentation

The common metrics to evaluate the goodness of models for semantic image segmentation are the Sørensen-Dice Coefficient (*Dice*) and Intersect over Union (*IoU*). Considering every individual pixel in terms of a confusion matrix, the Dice cefficient [SØRE48], also called F1 Score, is defined as in Equation (9):

$$F1 = Dice = \frac{2TP}{2TP + FP + FN} \tag{9}$$

Where the variables are defined as True Positive, *TP*, False Positive, *FP* and False Negative, *FN*. In terms of the segmentation, it can also be defined as two times the intersect of two areas divided by the sum of both areas, see Figure 2-17. The Jaccard Index or *IoU*, is defined as the intersect of two areas divided by their union, as the name suggests, also see Figure 2-17 [JACC12]. In terms of a confusion matrix the *IoU* is defined as in Equation (10):

$$IoU = \frac{TP}{TP + FP + FN} \tag{10}$$

Figure 2-17: Graphical representation of IoU and Dice Coefficient

Grafische Darstellung des IoU- und Dice-Koeffizienten

There are examples for usage of both, *IoU* and *Dice*, in semantic image segmentation for pixel-wise classification. *IoU* is also commonly used in object detection besides other metrics. Object detection is a task that involves detecting and localizing objects within an image by classifying patches of the image into different object classes. *IoU* is often used in object detection as one of the key metrics to measure the overlap between the predicted and ground truth bounding boxes. As opposed to semantic image segmentation where each pixel is assigned to a class. The *Dice* coefficient, on the other hand, is typically used in the context of binary segmentation, where the task is to classify each pixel in the image as either foreground or background. The *Dice* coefficient provides a measure of the similarity between the predicted binary mask and the ground truth mask. While both metrics provide a measure of overlap, *Dice* coefficient is generally more favorable as it can handle cases where the intersection between the predicted and ground truth masks is very small.

Figure 2-18 shows the relation between *IoU* and *Dice*. Dice gives more favorable results, especially far away from the domain boundaries. It is important to note that there is currently no established standard for which evaluation metric to use, and the choice

of metric can vary between different research studies or applications. The decision whether to use either of both is up to the individual author.

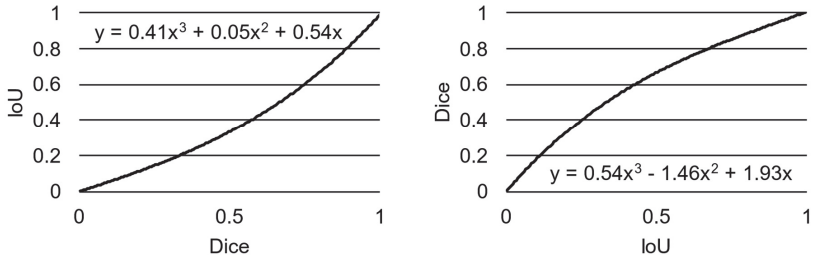


Figure 2-18: Mathematical relation between IoU and Dice Coefficient
Mathematische Beziehung zwischen IoU- und Dice-Koeffizient

2.3.4 Tool Wear Identification with Deep Learning

Erkennung von Werkzeugverschleiß mit KI

Recent advancements in Tool Condition Monitoring (TCM), specifically automated analysis or microscopic image of cutting tool wear, have been achieved using Deep Learning (DL), a subfield of Machine Learning (ML), which is an Artificial Intelligence (AI) technology [BERG20]. The approaches to tool wear identification mentioned below belong to the domain supervised ML, which describes the ability of a computer program to automatically find an optimal code that solves the problem, given multiple solved examples of that problem [GOOD14].

The first occurrence of DL in the cutting tool wear domain is an approach to tool wear form classification on indexable inserts with a VGG-16 architecture. The CNN yields 96 % precision rate in differentiating four wear forms and a mean absolute percentage error in wear measurement of less than 5 % using traditional image processing methods [WU19].

LUTZ put out the second occurrence of DL in the cutting tool wear image analysis uses a CNN with a small sliding window to perform a tool wear form classification task: Specifically, differentiate background, undamaged insert body and three different wear forms, namely flank wear, grooves and build up edge, on an indexable insert dataset. An overall accuracy of 91.5 % was reached with highest accuracy in the classes background and undamaged insert body (Figure 2-19). Further the hyperparameters number of CNN blocks, kernel size and number of neurons in the FC layer are investigated with regards to accuracy performance. Three optima are identified with the same accuracy of 91.5 % [LUTZ19]. In the paper's conclusion LUTZ states:

"*As this research focused only on the same insert, in the future, the process must be investigated and adopted for the analysis of multiple different inserts with different optical properties due to different coatings or other effects.*" [LUTZ19]

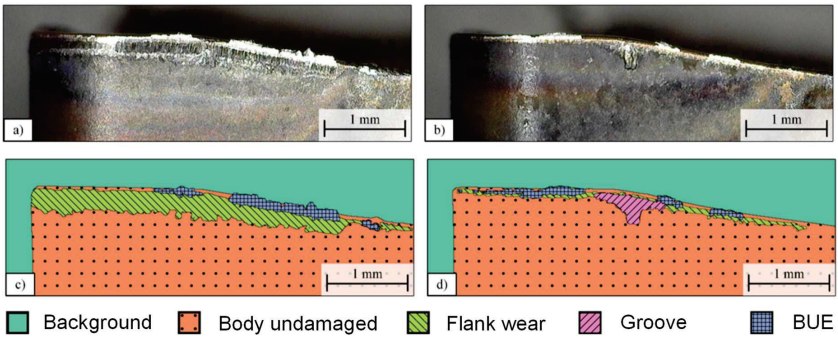


Figure 2-19: Images of indexable inserts (a,b) and respective mask (c,d) [LUTZ19]
Bilder der Wendeschneidplatten (a,b) und die jeweilige Maske (c,d) [LUTZ19]

BERGS published the next approach to semantic image segmentation which is conducted using pixel-wise classification of cutting tool wear on microscopic images [BERG20]. A U-Net architecture trained on a heterogeneous dataset consisting of 400 images with magnification levels between 20 and 200 is investigated. The dataset consists of eight different tool versions from three different tool types: Milling tools (end milling cutters and ball end milling cutters), indexable inserts and drilling tools. The authors use image augmentation, i.e., basic image manipulation like flipping and rotating, to increase the dataset size and introduce more variance into the dataset for better generalizability of the model. The mean IoU (*mIoU*) for held out test data is 0.73. The authors also made predictions on inference data, that is unknown tool versions that were not included in the train, validation, or test data. Those inference data images are acquired under artificially altered conditions, for example blurred due to incorrect focus or overexposed. The base images of the inference data set result in an *mIoU* of 0.48, whereas the whole inference dataset including disturbed images yields an *mIoU* of 0.37 (Figure 2-20).

The authors also investigate whether it makes sense to train one model with all images or if it makes sense to train the models tool type specific. For the more homogeneous individual datasets the test score is superior to the one-for-all model. The drilling tool dataset with a fixed magnification and therefore very homogeneous data gives an 20 % better *mIoU* of 0.87 as compared to the one-for-all model. Individual datasets including several magnifications leading to quite heterogeneous data has around 4 % worse *mIoU* scores [BERG20]. In the same paper, the authors propose a pipeline of tool type classification prior to segmentation by individual models. A first attempt to tool type classification is conducted using a CNN that yields an accuracy of 95.6 % on held-out test data. The authors suggest that a machine tool integrated inspection system for inline measurements would benefit from the proposed approach of automated tool wear identification. In the paper's conclusion BERGS states:

"*Data preparation requires more time and attention for segmentation networks compared to the classification network since the ground truth masks must be made manually with great care. […] Methods for model improvement using unlabelled data or artificial data generation using Generative Adversarial Networks (GAN) could help accelerating the data generation and increasing the database artificially.*" [BERG20]

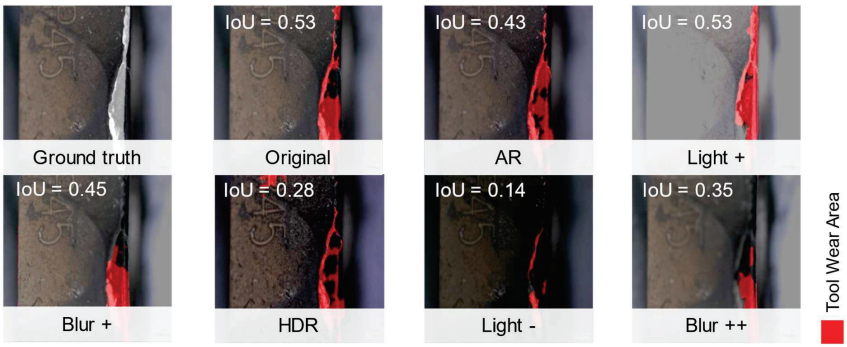


Figure 2-20: Different capturing settings with predicted mask and *IoU* coefficient of an indexable insert from the inference dataset, taken with different settings on a measuring microscope [BERG20]
Verschiedene Aufnahmeeinstellungen mit prädizierter Verschleißmaske und IoU-Koeffizienten von einer Schneidplatte des Inferenzdatensatzes, aufgenommen mit verschiedenen Einstellungen auf einem Standmessmikroskop [BERG20]

WALK conducts tool wear form classification with a CNN, specifically with a VGG-16 architecture [WALK20]. One network is trained to distinguish between indexable inserts with and without flank wear, another one is trained to distinguish between indexable inserts with and without chipping. A Matthews Correlation Coefficient (MCC) of 0.878 is reached for the flank wear CNN and 0.644 for the chipping CNN respectively. In the same paper the authors seek answer to the question whether a system can be designed for deep-learning-based computer vision to automatically determine the location and extent of wear phenomena on images from worn tools. They conclude that the approach of semantic image segmentation of cutting tool wear using a U-Net architecture may be used for the task. In a successive paper, the same authors plan to conduct this very task [WALK20]. In the paper's conclusion WALK states:

"*Another technical limitation is our (so far) limited consideration of only the flank of a worn cutting edge. […] However, domain experts confirmed the usefulness of an automatic characterization of the flank side. Thus, we believe this is a reasonable scope for now and leave this aspect for future work.*" [WALK20]

TREISS and two other authors from the paper mentioned above, published another work in the field of cutting tool wear image analysis. They used a Monte-Carlo based dropout

method to estimate the networks uncertainty regarding the tool wear detection (Figure 2-21). A technique that GAL introduced since they found that the probabability of the softmax outputs does not represent model uncertainty accurately [GAL15]. TREISS showed that prediction quality and model uncertainty have a linear relation with an R^2 = 0.72, allowing for an estimation of prediction goodness [TREI20]. Therefore, a possibly bad prediction of the network can be automatically identified and queued for relabeling. Further TREISS demonstrates that in an uncertainty-based human-in-the-loop system for tool wear image annotation, i.e., labeling, the performance of resulting models is higher compared to the random selection of images for relabelling. To prove the generalizability they conduct the same study on the publicly available Cityscape dataset [TREI20]. In the paper's conclusion TREISS states:

"*A human-in-the-loop system can be beneficial for all types of automation tasks, in which human experts display superior performance than automated systems, but in which the automated system is more cost efficient. An example for such a system would be an industrial quality control system.*" [TREI20]

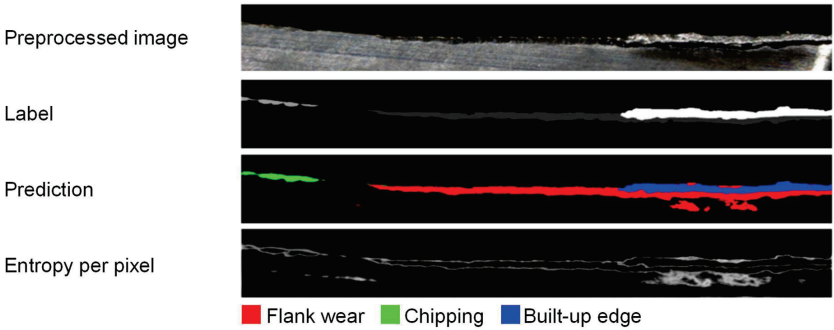


Figure 2-21: Image, Label, Prediction and Uncertainty map of an indexable insert [TREI20]
Bild, Label, Prädiktion und Unbestimmheitskarte einer Wendeschneidplatte [TREI20]

LUTZ compares the sliding window approach from the 2019 publication with several one-pass networks like U-Net using a homogeneous indexable insert dataset [LUTZ20]. The task is the segmentation of five different classes: Specifically, differentiate background, undamaged insert body and three different wear forms, namely flank wear, grooves and build up edge (Table 2-2). The benchmarking of the pixel-wise segmentation networks results in favor of LinkNet [CHAU17] which reaches an overall *mIoU* of 0.8 across all classes when trained with augmented data. The LinkNet also reaches the highest *mIoU* of 0.55 for the individual class flank wear compared to U-Net and three other networks. The U-Net architecture gives a maximum overall *mIoU* of 0.69 using the not augmented dataset in this investigation. The one-pass networks

are compared to the sliding window approach in combination with the Google Cloud AutoML framework were an *mIoU* of 0.94 was reached across all classes and the flank wear specific *mIoU* amounts to 0.76. The authors state that the approach is not yet tested for heterogeneous datasets, i.e., general usibility is not yet rated:

"In future work, a more diverse dataset containing multiple different types of tools should be investigated, thus allowing the approaches to be rated based on their general usability." [LUTZ20]

Table 2-2: Scores (*mIoU* and *IoU*) with and without augmentation and scores on individual classes for five network architectures
Werte (mIoU and *IoU) mit und ohne Augmentation sowie Werte für die einzelnen Klassen für fünf Netzarchitekturen*

	Aug. (*mIoU*)		Individual class scores (*IoU*)				
Architecture	**without**	**with**	**Backg.**	**Tool**	**Wear**	**Groove**	**BUE**
FCN	0.32	0.36	0.94	0.87	~0	~0	~0
U-Net	0.69	0.63	0.95	0.85	0.07	0.80	0.50
SegNet	0.47	0.31	0.78	0.76	~0	~0	~0
LinkNet	0.67	0.80	0.99	0.96	0.55	0.70	0.79
PSPNet	0.68	0.73	0.97	0.93	0.41	0.80	0.54

MIAO applies the U-Net apprach to tool wear area detection on indexable insert [MIAO21]. Figure 2-22 contains red boxes that show areas which are almost indistiguishable for humans but are successfully identified by the NN. The authors conduct three distinct investigations: Firstly, a method for using layer-wise objective functions is implemented, additionally to the overall objective function [LEE14]. This increases the prediction quality in terms of *Dice* by 2.5 % applied on their own dataset. Secondly, different loss functions are tested with the assumption of tackling data imbalance issues. The effect is not significant for the three compared loss functions Binary Cross Entropy (BCE) loss, *IoU* based loss and MCC based loss. Thirdly, Miao investigates the performance of a U-Net with attention gates, a method aiming to enhances important parts of the input data while diminishing less important parts [VASW17]. The U-Net with attention mechanism and layer-wise objective functions yields a *Dice* coefficient on the test data of 0.97 [MIAO21]. Finally, in a manual manner the width of flank wear land, *VB*, is extracted from a greyscale converted prediction mask of an indexable insert. A possible weakness in MIAOS results is that the dataset of 186 images stems from only 14 tools of the same tool version. In the paper's conclusion MIAO states:

"*However, this combination is a good one but may not be the best one. Many other good network structures and customized loss functions are worth trying.*" [MIAO21]

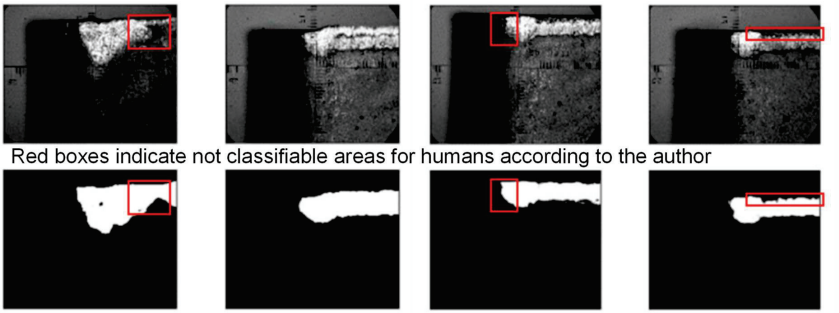


Figure 2-22: Prediction results by U-Net with layer wise objective functions and MCC loss on indexable insert images of the test dataset [MIAO21]
Prädiktionsergebnisse des U-Net mit schichtweisen Zielfunktionen und MCC-Verlustfunktion auf Bildern von Wendeschneidplatten der Testdaten [MIAO21]

BERGS proposed the synthesis of image data through Generative Adversarial Networks (GAN) for the cutting tool use case [BERG20]. It is conducted for the first time in the manufacturing domain for the use case of blanking tool wear identification [MOLI21]. Recently, LUTZ used a GAN approach to reduce the labelling effort for a homogeneous indexable insert dataset [LUTZ21]. A Deep Learning Augmentation (DLA) is performed with a network derived from the pix2pix architecture for image-to-image translation. A domain adaption is reached to convert data from one domain to another [ISOL16]. This method requires labelled data from the source domain, with more available data, and the target domain, with little available data. Using the GAN, a training dataset for the target domain is created from artificially created images and their masks stemming from the source domain (Figure 2-23).

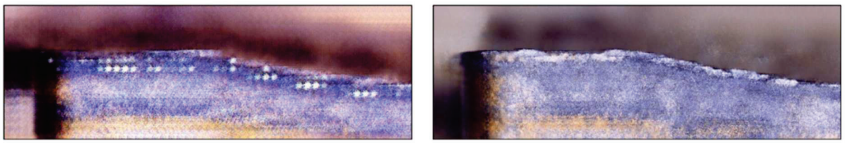


Figure 2-23: Images after GAN training with one (left) and ten (right) images [LUTZ21]
Bilder nach GAN-Training mit einem (l) und zehn (r) Bildern [LUTZ21]

The artificial dataset is further used to train a segmentation model for the target domain. With this approach an overall *mIoU* for the five considered classes of 0.72 could be reached with ten images instead of 32 images using the prior training procedure. The score for the isolated flank wear class is 0.65. For the indexable insert dataset the labelling effort is reduced by more than 68 %. In the paper's conclusion LUTZ states:

"*With the proposed algorithm, it is possible to reduce the labelling effort significantly, however, manual labelling is still required while some defect classes are hard to predict using the proposed approach.*" [LUTZ21]

On basis of the work and proposals in the previous paper by BERGS and some of the same authors [BERG20], HOLST published a method consisting of a pipeline of DL operations [HOLS22]. The pipeline consists of a CNN for tool detection and a U-Net for tool wear segmentation. It processes tool wear images collected with a digital microscope and is complemented by a rule-based approach to measuring wear along the cutting edge of machining tools. The end-to-end approach allows fully automated tool wear detection and measurement that can be used for inline measurements within CNC machine tools for machining applications. For the specific use case of ball end milling cutters, a dataset of 80 images is used for training and no augmentation methods are applied. The *Dice* for held-out test data amounts to 0.85. The *VB* measurements through a bounding box algorithm yield a linear fit with R^2 = 0.99 for inline measurements agains measurements with a stand microscope. *VB* values below 150 µm where not present in the dataset though [HOLS22]. In the paper's conclusion HOLST states:

"*Possible next steps in this research activity are: […] Study effects of network size, hyperparameters, activation functions and data augmentation on model performance.*" [HOLS22]

In summary this subsection treated the following research: Successful usage of a CNN for image classification in a tool wear identification use case [WU19]. The general methods and tools required for automated image analysis for cutting tool wear detection were established [BERG20, WALK20, LUTZ20]. TREISS and LUTZ established different approaches to reducing the labeling effort in their specific use case [TREI20, LUTZ21]. Methods to reduce required data, namely testing of loss functions and U-Net derivatives, were investigated by MIAO with a small dataset [MIAO21]. Finally, the need for an understanding of the effect of hyperparameters on model performance and possible optimizations was pointed out [HOLS22].

2.4 Interim Conclusion

Zwischenfazit

The Chapter 2 Fundamentals and State of the Art gave a brief overview about the following topics: Section 2.1, Tool Wear in Metal Cutting, Section 2.2, Quantification of Tool Wear, and Section 2.3, Image Processing with Deep Learning.

Section 2.1, Tool Wear in Metal Cutting, gave a brief introduction to the cutting part of the cutting tool as a complex tribological system. It also covered the impact of tool wear on the manufacturing process, resulting costs and hence the importance of knowledge about tool condition. Section 2.2, Quantification of Tool Wear, covers the terminology and methods required for an understanding of the tool life testing procedure. Tool life testing is elaborate and especially the quantification of cutting tool wear hinders an efficient creation of tool life models or indirect tool condition monitoring models. The section also covers direct measurement of cutting tool wear and attempts to computer

vision solutions for automated image analysis in the domain. Section 2.3, Image Processing with Deep Learning, treats image analysis with artificial intellicence algorithms, specifically DL for tool wear identification. This field of research dates to 2019 and is currently being addressed by several research groups with different foci, mainly due to their respective use cases within the niche of metal cutting tool wear. The table in Annex A.16 provides a summary of the research conducted to date in the field of automated image processing for cutting tool wear monitoring with deep learning.

The literature emphasises that there are many differences concerning the use case and methods. Concerning the use cases there are differences in the number of tool variants (mostly one tool variant) and tool types (mainly indexable inserts), the available labeled database, and the image size. Regarding the methods there is also different approaches to data augmentation, network type, training / validation / test dataset split, the used metric for determining the prediction goodness and the test scores itself. This heterogeneity makes comparison difficult, but nevertheless, or precisely because of this, recommendations and blind spots can be identified. Problems that have already been adressed by research include:

- The extensive **label effort** that consumes time of specialized experts may be reduced using TREISS' uncertainty-based human in the loop approach to labeling new data and LUTZ' GAN approach to creating synthetic data for new application domains [TREI20] [LUTZ21].
- The use of different **evaluation metrics** to determine the goodness of prediction (test score) of the models is a minor concern since the most used metrics $Dice$ and IoU can be calculated from each other.
- An investigation of the effect of **data augmentation**, specifically Basic Image Manipulation (BIM), on test data performance using different segmentation model architectures has been conducted for a dataset of indexable inserts.

A collection of problems in the domain of automated image analysis for cutting tool wear segmentation with deep learning algorithms that have not been addressed yet include the following:

1. **Datasets are not publicly available**; this makes identification of superior approaches to tool wear detection hardly possible.
2. **Failed predictions remain mostly unpublished**, though they could provide information on commonalities in failure modes.
3. **Knowledge about effects of train/val/test split, image size and number of images** on performance of a DNN for tool wear segmentation is not investigated yet.
4. **A set of metrics to characterize and compare datasets** has not been explored and established yet.
5. **Overfitting is not reported** in the form of a concise metric that allows comparison across datasets or even different use cases

6. **Underspecification** is likely due to lacking real world application assessment and a lacking systematical approach to hyperparameter optimization.

The progress and deficits in the research domain of automated image analysis for cutting tool wear segmentation with deep learning have been summarized in this section. In the next chapter conclusions are drawn from this information with regards to the content and structure of this thesis.

3 Objectives and Approach

Zielsetzung und Vorgehensweise

3.1 Objectives and Research Methodology

Zielsetzung und Forschungsmethodik

As seen in the previous chapters, tool wear and the quantification of tool wear in the machining process are relevant topics for manufacturers today. The two main technical-economic problems are:

- Cutting tool wear during machining reduces the efficiency of the process and increases the manufacturing costs due to its influence on dimensional accuracy and surface quality.

- The measurement process is manual, subjective, and elaborate which promotes arbitrary decisions, makes documentation laborious, hinders automation, promotes waste, and prevents further processing of data for sustainable problem mitigation.

The automation of cutting tool wear quantification is sought after by the manufacturing industry. Recent research in automated cutting tool wear identification with deep learning yields promising results regarding the semantic image segmentation for pixel-wise classification of flank wear and other wear forms on microscopic images of cutting tool edges (Section 2.3). This thesis aims at addressing blind spots elaborated in the chapter above in the field of image analysis for automated cutting tool wear segmentation with deep learning algorithms. Open questions regard the data assessment and preprocessing, others regard the optimization and qualification of semantic image segmentation models. In detail, there is no system of key metrics for characterizing an image dataset. This prevents targeted AI-modelling based on dataset characteristics and hinders comparing datasets. There was no attempt yet to investigate the influence of dataset and model properties on model performance. Especially, an assessment of the performance of tool wear segmentation models in the real-world domain, such as inline imaging within machine tools.

In view of the above, this dissertation aims to develop and optimize models for the identification of tool wear on images of cutting tools and cutting tool edges. The developed image processing and model optimization approach shall contribute to a more efficient and reliable process monitoring and shall increase the ability to obtain process understanding, paving the way for a better understanding of the tool wear phenomenon with regards to the process and its manifold variables. The objective pursued is summarized in the following:

Objective of the thesis

The objective of this work is the optimization of tool wear segmentation and measurement on microscopic images of cutting tool edges.

Based on the described problems and objective, the following research hypothesis was formulated:

Research hypothesis

The assessment of dataset properties enables a systematic, model-based hyperparameter optimization of the AI-model for cutting tool wear segmentation, reducing the need for iterative optimization cycles.

To verify the research hypothesis, the following research questions need to be investigated.

Research questions

1. How can image processing be applied to automatically segment tool wear on microscopic images of cutting tool edges?
2. What are the dataset and model properties with the highest impact on model performance for tool wear segmentation?
3. How can a systematic choice of hyperparameters with regards to dataset properties be employed to improve model performance for tool wear segmentation?
4. How can the optimized segmentation model be applied for an inline approach to cutting tool wear measurement within machine tools?

3.2 Procedure and Setup of the Thesis

Vorgehensweise und Aufbau der Dissertation

This dissertation is based on the research process of applied sciences according to Ulrich [ULRI01], see Figure 3-1. Based on practice-relevant problems a methodology was generated which allowed to automatically segment tool wear on microscopic images of cutting tool edges (Research Question 1). The effect of dataset and model properties on prediction quality of the segmentation model was investigated in statistically validated experiments (Research Question 2). The findings of the previous screening experiments were used to create a decision model. This model allowed the selection of model properties based on the dataset properties to achieve the highest possible prediction quality of the tool wear segmentation model (Research Question 3). Finally, a pre-optimized model was tested and evaluated in a validation experiment with an inline microscope and a camera inside machine tools (Research Question 4). The structure of this thesis and the research methodology is shown in the following figure.

The concepts relevant to a common understanding of the tool wear topic in metal cutting were covered in the introduction (Chapter 1), which also elaborated on the motivation of the thesis. Subsequently, an overview to the cutting tool wear topic was given (Chapter 2). The state of the art was presented, as well as methods that standards recommend for tool life modeling. Current topics of research with regards to image analysis of microscopic tool wear images were discussed, which are currently indispensable for tool wear quantification. Furthermore, deficits were highlighted,

regarding the current research in semantic image segmentation for cutting tool flank wear identification. On that basis the requirements for research and the objectives of the work were deduced (Chapter 3).

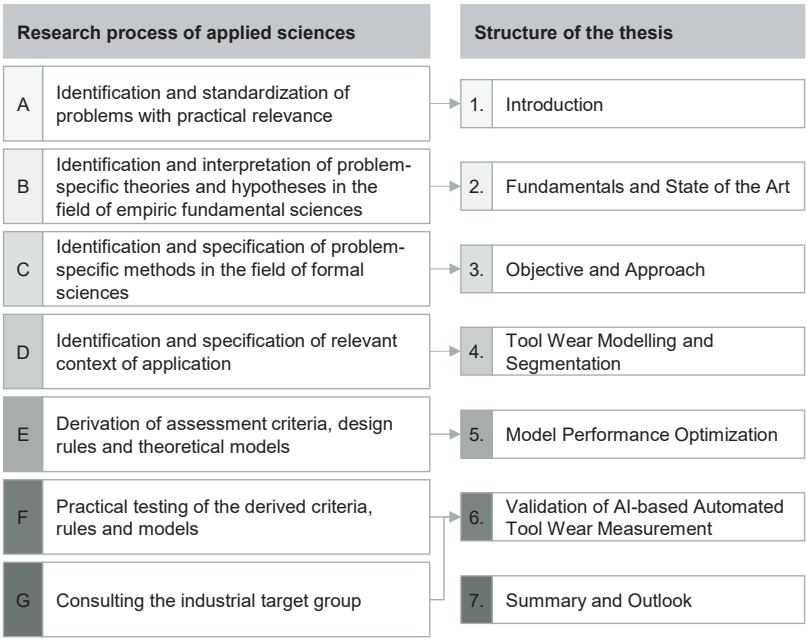


Figure 3-1: Structure of the thesis based on research methodology by ULRICH [ULRI01]
Gliederung der Arbeit auf Basis des Forschungsprozesses nach ULRICH [ULRI01]

A tool life model creation with methods recommended in the respective standards was made. The data created during this task served as a basis for further image processing investigations. A pipeline for the DL model creation in the use case of automated tool wear segmentation was presented (Chapter 4). Successively, dataset and model properties were investigated using a screening design of experiment, to determine the most important factors influencing prediction quality. Based on the results an in-depth investigation of the important factors was conducted and a decision model for the dataset-specific choice of hyperparameters was created (Chapter 5). Machine tool integrated measurement setups using a microscope and a camera serve as use cases for assessment of the model optimization strategy (Chapter 6). In the last chapter, the findings were summarized and critically reflected. Furthermore, an outlook was given on newly arising questions regarding the application of inline measurements of tool wear and other possibilities that arise through the automated tool wear segmentation and measurement aided by deep learning methods (Chapter 7).

4 Tool Wear Modelling and Segmentation

Modellierung und Segmentierung von Werkzeugverschleiß

This chapter covers the following topics:Section 4.1, Surveys with Industry Professionals, presents the results from surveys regarding the topic of tool wear. Section 4.2, Framework of Investigation, describes the exemplary use case of finishing hard-to-cut materials. Section 4.3, Process Specification, contains the materials and methods necessary to conduct a tool wear model creation as per standard, described in Section 4.4, Empirical Investigation of Tool Wear. Finally, in Section 4.5, Model Design for Tool Wear Segmentation, the first Research Question is answered:

RQ1: How can image processing be applied to automatically detect tool wear on microscopic images of cutting tool edges?

4.1 Surveys with Industry Professionals

Umfragen mit Experten aus der Industrie

There is a lack of published literature on the dealing with cutting tool wear in the manufacturing industry. This scarcity of information hampers the ability of professionals to make informed decisions and implement effective strategies. Therefore, addressing this gap becomes crucial for enhancing machining processes and optimizing tool performance.

As part of consortium projects at the Fraunhofer IPT, surveys were conducted with industry experts about cutting tool wear. The surveys aimed to utilize their experiences, identify challenges, and gather practical strategies. The surveys involved professionals from manufacturing and engineering: Twenty-two machine tool operators from tool and die making companies were surveyed in 2020. Eight research and development engineers from the aerospace industry, specifically turbomachinery component manufacturers, were interviewed in 2021. Finally, 34 employees from different industries were surveyed in 2023, including 20 % from tool and die making companies and 13 % from the aerospace industry. The surveys had different scopes, but there was an intersection in the questionnaires that is used as a basis in this thesis. The intersections of topics in the questionnaires included a prognosis of price progression for cutting tools, the safety margin that is applied on tool life and finally the fraction of tool cost relative to the cost of goods sold. The following paragraph summarizes the results of the surveys. Figure 4-1 shows selected results of the three surveys as barcharts.

The last two surveys asked for a forecast on the price development of cutting tools. In these surveys, 71 % and 76 % of the participants responded that they expected prices for cutting tools to rise. Data was collected on the following points in all three surveys. Accordingly, weighted average values are available for a total of 65 completed questionnaires. The average applied margin of safety is 23 % with a standard deviation of

7.5 %. This means that about 77 % of the tool life is utilized. The average share of the cost of cutting tools in the cost of goods sold is 10 % with a standard deviation of 2.8 %.

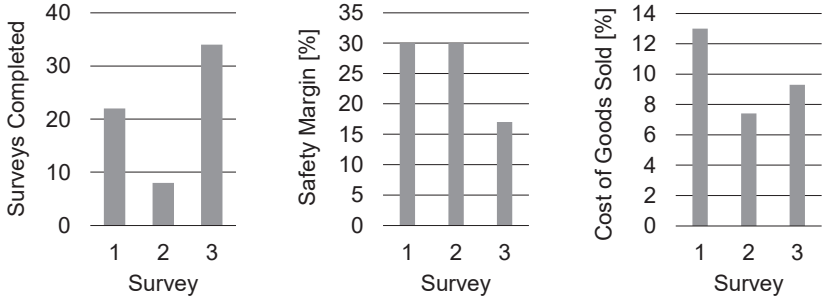


Figure 4-1: Results of three surveys focussing on cutting tool wear
Ergebnisse von drei Umfragen zum Zerspanungswerkzeugverschleiß

4.2 Framework of Investigation

Untersuchungsrahmen

The outstanding properties of alloy 2.4668, also known as Inconel 718™, (Figure 4-2) make it difficult to machine and lead to severe milling tool wear, which can affect the quality of the product.

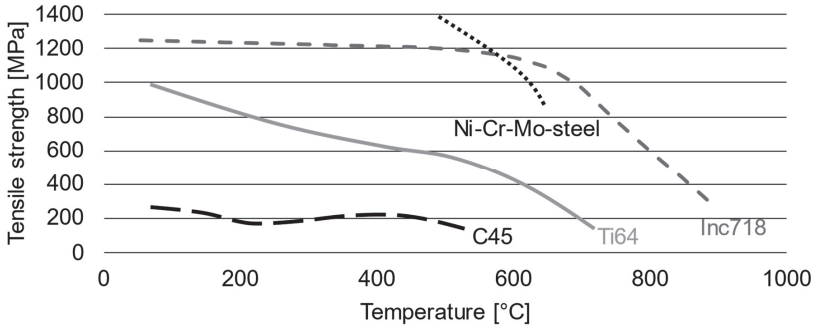


Figure 4-2: Effect of temperature on tensile strength for various materials according to SUN *and* MACHERAUCH [SUN10, p. 664, MACH11, p. 174]
Einfluss der Temperatur auf die Zugfestigkeit verschiedener Werkstoffe according to [SUN10, p. 664, MACH11, p. 174]

The decrease in product quality due to tool wear and the resulting machine downtime due to frequent tool changes are the main challenges in high performance machining [MOHA20]. Tool wear is thus responsible for high production costs and poor surface qualities, resulting in an increased need for optimization in finish milling operations.

For the estimation of the tool life, wear models are usually applied, which require complex and cost-intensive experiments for the generation of the model data [ZHOU18]. The framework of investigation in this work is the finish milling of alloy 2.4668 with ball end milling cutters. The relevance of the use case results from climate policy in aviation, aerospace-specific part quality requirements that suffer from tool wear and the technical predisposition of alloy 2.4668 with regards to tool wear, which is outlined in the next section in more detail.

4.3 Process Specification

Prozessspezifikationen

This section presents the process specifications for a classical tool wear model creation. Specifically, a Taylor model was created following the respective standard. This standard demands a complete description of the following prerequisites described individually in the following paragraphs.

Workpiece

Werkstück

The material alloy 2.4668 (according to EN 10027-2:1992-09) is a nickel-based alloy characterized by good corrosion resistance and outstanding high-temperature strength. Due to its material properties, Alloy 2.4668 is particularly suitable as a construction material for the aerospace industry, for example for the manufacture of compressor and turbine blades [KLOC18, p. 346].

The workpiece used in the fundamental trials has the dimensions 100 x 101 x 109 mm measured with a digital sliding caliper. The material is heat treated with the aerospace specification AMS 5663. The machinability of nickel-base alloys is investigated by EZUGWU based on various publications [EZUG99]. The material strength remains largely unchanged during machining due to the good high-temperature properties. Poor thermal conductivity of nickel-based alloys produces high temperature on the tool cutting edge and strong temperature gradients in the tool. The hard abrasive carbides in the superalloys, lead to strong abrasive wear on the tool.

Figure 4-3: Cutting edge radius and wedge angle measurement with Alicona Edge Master

Schneidkantenradius- and Keilwinkelmessung mit Alicona Edge Master

Cutting Tool

Zerspanwerkzeug

The finishing operations for turbine blades in turbomachinery manufacturing is conducted with barrel or ball end milling cutter. In this case a ball end milling cutter is applied for fundamental milling trials on a solid block workpiece. All cutting tools were inspected prior to the experiment regarding defects, see Figure 4-3.

Table 4-1: Cutting tool specification
Werkzeugspezifikation

Specification	Value
Tool Type / Coating	Ball end milling cutter / TiAlCN
Material	Tungsten carbide (10 % cobalt, grain size 0.8 µm)
Diameter D [mm]	12
Number of flutes z	2
Helix angle λ_0 [deg]	30
Rake angle γ_0 [deg]	-10
Clearance angle α_0 [deg]	0
Cutting edge radius r_ε [µm]	3

The detailed cutting tool specifications and geometric parameters can be found in Table 4-1. Cutting fluid was not used during the trials to realize a higher wear rate. The shaft of the ball end milling cutter was clamped 39 mm into a Regofix tool holder.

Machine Tool

Werkzeugmaschine

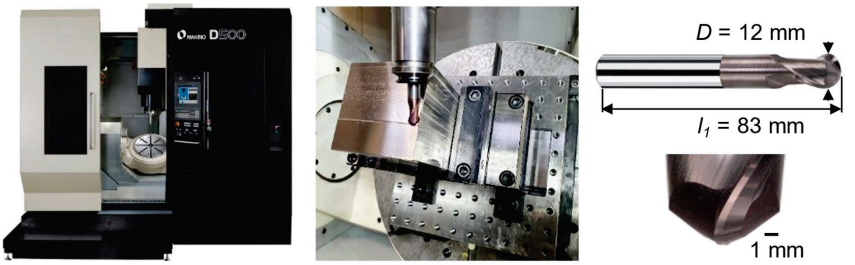


Figure 4-4: Machine tool (left), workpiece (middle) and cutting tool (right)
Werkzeugmaschine (links), Werkstück (mitte) und Zerspanwerkzeug (rechts)

The machine tool used for the finishing process in the proposed setup is a a Makino D500, see Figure 4-4. It is a 5-axis vertical machining center and the positioning accuracy of the linear axes is 2.5 µm and therefore in accordance with the standard

[ISO1701-1]. No vibrations were observed during the trials that could have manipulated the results of this experiment.

Process Parameters
Prozessparameter

For the cutting experiments a parameter set of (semi-)finishing conditions was chosen. The feed per tooth was set to f_z = 0.1 mm, the axial and radial depth of cut were a_p = 0.5 mm and a_e = 0.5 mm. The milling process was a fundamental milling experiment, compare Figure 4-4 and Figure 4-5, with a tool axis inclination of β_{fn} = 70° between axis of milling spindle and working plane.

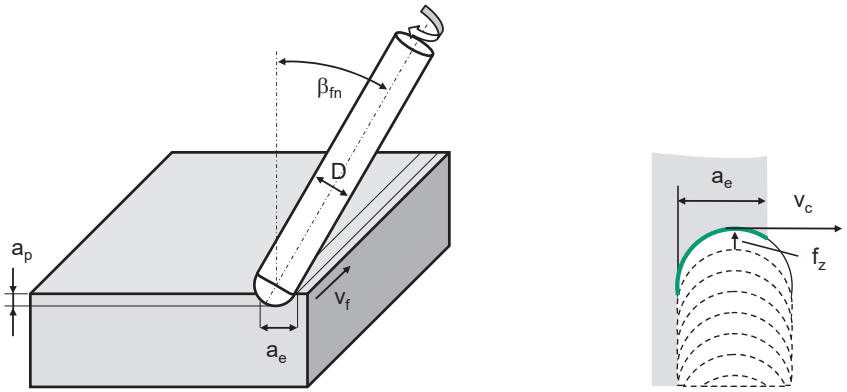


Figure 4-5: Schematic illustration of cutting experiment
Schematische Darstellung der Zerspanungsexperimente

4.4 Empirical Investigation of Tool Wear

Empirische Verschleißuntersuchung

The empirical investigation of tool wear was conducted in accordance with the respective standards described in Subsection 2.2.2. This section starts with a description of the experimental design in Subsection 4.4.1, Design of Experiments. Subsection 4.4.2, Analysis of Occurring Tool Wear, describes the measuring method and type of tool wear regarded in this investigation. Finally, in Subsection 4.4.3, Tool Wear Model Creation, the calculations to create the model are shown.

4.4.1 Design of Experiments

Versuchsplan

Cutting experiments were planned and evaluated according to guidelines in the relevant standards [ISO8868-2, ISO3685]. Specifically, a tool wear investigation of type B according to the classification in ISO8868-2 was chosen. This means one *vT*-curve was determined with the cutting speed as a variable for a particular combination of other cutting variables. In literature the investigated cutting speed range for alloy

2.4668 usually lies between 20 and 120 m/min [ROY18, WANG18, ZHU13]. For the investigation in this thesis the lowest test point was chosen at $v_{c,max}$ = 70 m/min. For each successive test point, the maximum cutting speed was increased in 10 m/min steps.

Since the minimum repetitions required for each experimental point is unknown previously, two repetitions were planned with further capacity to increase the number of repetitions. Later a third repetition was added to strengthen the statistical informative value of a tool wear model created with the data. Concluding, six test points with three repetitions each were investigated, resulting in a total of 24 cutting tests, respectively cutting tools, see Table 4-2.

Table 4-2: Experimental plan for type B tool wear model creation
Experimenteller Plan für die Verschleißmodellerstellung nach Typ B

Specification	Values					
Test point	1	2	3	4	5	6
$v_{c,max}$ [m/min]	70	80	90	100	110	120
n_{rpm} [1/min]	1936	2228	2494	2785	3051	3342
Experimental runs / Tools	4	4	4	4	4	4

4.4.2 Analysis of Occurring Tool Wear
Analyse des auftretenden Werkzeugverschleißes

The cutting tool wear was inspected using a Keyence VHX-6000 microscope and a DinoLite USB microscope. The visual tool wear inspection was conducted after one meter of tool travel path L_f. For measuring the maximum width of flank wear land metric, VB_{max}, of each cutting edge the Keyence software was used, see Figure 4-6.

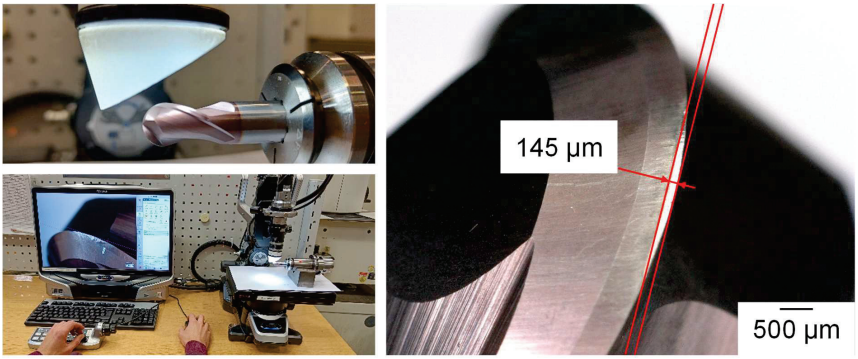


Figure 4-6: VB measurement using the Keyence VHX-6000 microscope
VB Messung mit dem Keyence VHX-6000 Mikroskop

A typical tool wear curve, as in Subsection 2.1.4 on page 9 was created from the successive tool wear measurements of a single tool. The wear follows the typical course of a third-degree polynomial, see Figure 4-7.

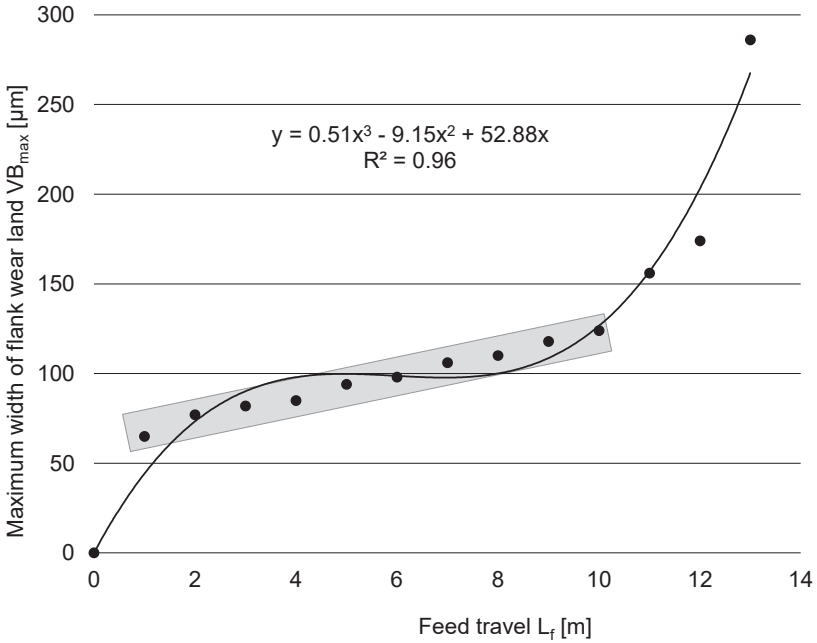


Figure 4-7: ***VB*$_{max}$ against feed travel for a ball end milling cutter at $v_{c,max}$ = 110 m/min**
VB$_{max}$ *über Vorschubweg für einen Kugelkopffräser bei* $v_{c,max}$ *= 110 m/min*

The *VB*$_{max}$ on the ordinate is the mean of the individual *VB*$_{max}$ measurements of both cutting edges. Cutting speed is defined as the instantaneous velocity of the primary motion of a selected point on the cutting edge relative to the workpiece [CIRP04]. The maximum cutting speed, $v_{c,max}$, in this experiment refers to the cutting speed at the top of the theoretical engagement line along the cutting edge. That means the theoretical cutting speed at the maximum diameter of the ball end milling cutter is higher. For example, the cutting speed of the experiment in the figure was $v_{c,max}$ = 110 m/min whereas the theoretical cutting speed was $v_{c,th}$ = 115 m/min. In the table above the respective milling spindle rotations per minute are documented.

In the Figure 4-8 the progression of wear along the feed travel is shown for one flute of the ball end milling cutter at 50x magnification. For the sake of readibility the *VB*$_{max}$ values are displayed with increased font. Also the measurements taken were amplified with red lines and arrows. The top left image shows the initial state of the cutter at 30x

magnification. The tool in the figure is the very tool that constitutes the wear curve in the figure above. Up until a VB_{max} value of 122 µm in Figure 4-8 a typical flank wear may be observed, as well as a mostly linear wear progression in Figure 4-7, compare gray, semi-transparent box. The transition from the linear wear progression to the progressive wear regime is found as a beginning cutting edge chipping in Figure 4-8 starting at 156 µm.

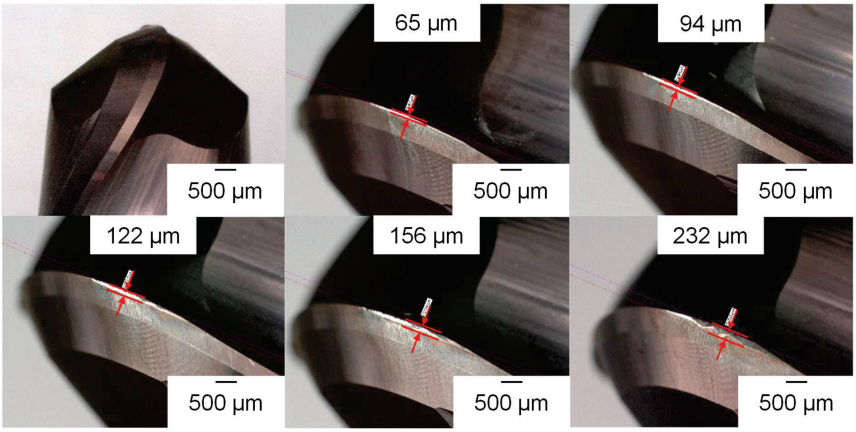


Figure 4-8: ***VB_{max}* of one flute *of the ball end milling cutter at $v_{c,max}$* = 110 m/min**
VB_{max} einer Schneide eines Kugelkopffräsers bei $v_{c,max}$ = 110 m/min

4.4.3 Tool Wear Model Creation

Erstellung des Werkzeugverschleißmodells

The tool wear model and necessary precalculations are shown in Subsection 2.2.2. A characteristic *vT*-curve with variable cutting speed for a particular set of process parameters is classified as Type B [ISO8868-2]. The precalculations are necessary to determine whether the means of two neighboring test points are sufficiently different for model creation.

The null hypothesis for this kind of hypothesis test states that the population means are equal. A significance level of α = 0.1 for a two-tailed data distribution indicates a 10 % risk of concluding that a difference exists when there is no actual difference between the means of two test points. Here, the tool life parameter tool travel path L_f of the four tools within one test point was used to calculate the test points mean value and standard deviation required for the hypothesis test. In the case of the first two test points the p-value of p = 0.047 or 4.7 % is lower than our chosen alpha (risk of 10 %), therefore the null hypothesis was rejected. This means a statistically significant difference between the two test points exists. Still the experimental point at cutting speed of 70 m/min was rejected from further calculations since it is obviously not following a possible linear relation, see Figure 4-10.

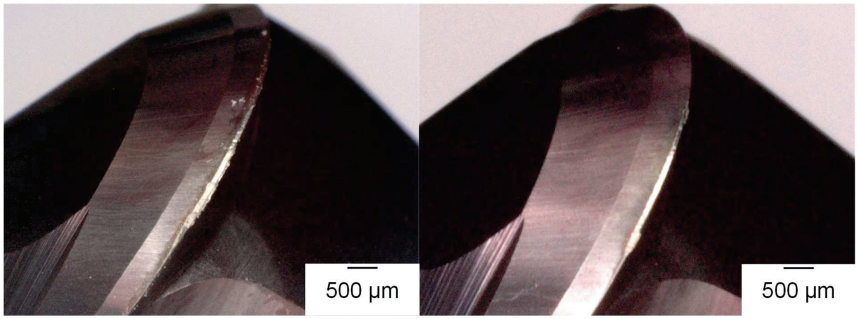


Figure 4-9: ***Two examples of BUE formation at $v_{c,max}$ = 70 m/min***
Zwei Beispiele für Aufbauschneidenbildung bei $v_{c,max}$ = 70 m/min

As shown conceptually in Figure 2-6 these tools were operated in the domain of possible build-up edge (BUE) formation. Figure 4-9 shows the BUE formation on two of the cutting edges at the respective cutting speed.

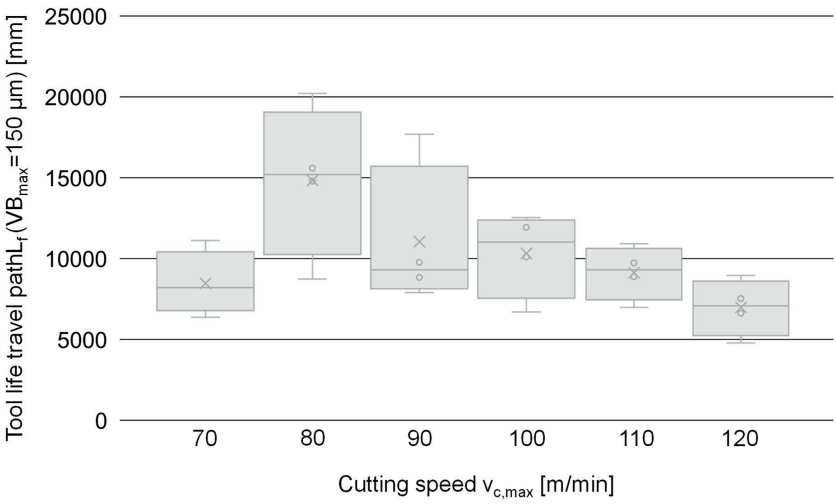


Figure 4-10: **Boxplots of four runs each with a travel path L_f (VB_{max} = 150 µm) at the six test points**
Boxplots der vier Durchläufe mit Vorschubweg L_f (VB_{max} = 150 µm) an jedem der sechs Testpunkte

For the data given a one-way analysis of variance (ANOVA) is means of choice instead of manually calculating each test point with a hypothesis test as described above. At the chosen alpha level of $\alpha = 0.1$ there is a significant main effect for cutting speed, the

test points differed significantly, $F(4,15) = 2.86$ with $F_{critical} = 2.36$, $p < 0.061$. The statistical requirements are met to build a model from the data underlying the consolidated data in Table 4-3.

Table 4-3: Experimental plan for type B tool wear model creation
Experimenteller Plan für die Verschleißmodellerstellung nach Typ B

Specification	Values					
Test point	1	2	3	4	5	6
$v_{c,max}$ [m/min]	70	80	90	100	110	120
Experimental runs / Tools	4	4	4	4	4	4
$\mu(L_f(VB_{max} = 150\ \mu m)$ [m])	8.474	14.827	11.045	10.314	9.125	6.974
$\sigma(L_f(VB_{max} = 150\ \mu m)$ [m])	1.701	4.080	3.887	2.273	1.439	1.514

Applying the calculations introduced in Subsection 2.2.2, Tool Life Testing in , a log-log characteristic tool life diagram, also called *vT*-diagram is created. Since the tool life parameter is tool life time T, instead of tool travel path L_f the values are converted according to Equation (11) below. The mean and standard deviation tool wear of the tools teeth of each test point are displayed in A.3 to A.8.

$$T = \frac{L_f}{v_f} = \frac{L_f\ \pi\ D}{f_z\ z\ v_{c,th}} \tag{11}$$

The Taylor equation, given in the chart and below, may be used for determining tool life time at a specific cutting speed, as in Equation (12), or vice versa, as in Equation (13). The parameters in the model are the slope, $k = -2.49$, and intersection of the x-axis, $C = 321$. The coefficient of determination of this model is $R^2 = 0.987$, the adjusted R^2 to penalize for required parameters calculates to 0.983. For example, if a cutting time of 15 minutes is desired for the modelled process, a cutting speed of 108 m/min should be set.

$$T = e^{k \cdot \log(v_c/C)} \tag{12}$$

$$v_c = e^{\frac{\log(T)}{k} + \log(C)} \tag{13}$$

The calculation of the model equation and confidence intervals shown in Figure 4-11 were performed according to the respective ISO standards [ISO3685, ISO8868-2]. To assess significance of the regression model an F-test was conducted. The F-value was calculated to $F(1,3) = 238$ with $F_{critical} = 5.54$ at an alpha level of $\alpha = 0.1$ which means the null hypothesis was rejected. The null hypothesis states that the variables, k and C, have no explanatory power. Using the p-value a risk of less than 0.06 % was calculated of being wrong about this statement, i.e., $p < 0.000592$.

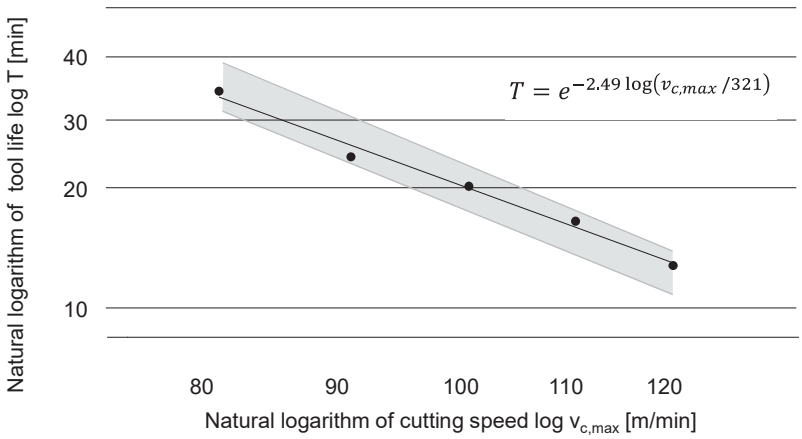


Figure 4-11: *vT*-diagram calculated from acquired experimental data
vT-Diagramm berechnet aus aufgenommenen experimentellen Daten

The experiments described in this subsection revealed weaknesses of the approach of classical cutting tool wear model creation:

- The parameter domain boundaries, in this case cutting speed, for the model creation is not exactly known prior to testing which causes additional effort.
- A high variance in tool life travel path occurs for some repetitive experiments, especially at cutting speeds of $v_{c,max}$ = 80 and 90 m/min, which may undermine the statistical prerequisites for a flawless model creation. The high variance is mostly due to a not well performing tool at 80 m/min and a very well performing tool at 90 m/min. No abnormalities were observed in these experiments that could explain these outliers.
- The required resolution of the parameter domain is not known, which can lead to additional effort without benefits to model usefulness.
- A variation in the process parameters or process conditions will most likely render the model useless, unless additional experiments are conducted to extend the valid range.

In general, the problems described above demonstrate the need for an individual observation of tool performance, which is currently labour-intensive and therefore not economically feasible in most of the cutting industry. To solve this problem, an automated analysis of microscopic tool wear images and an automated measurement process of cutting tool wear is required. An approach to tackle the former is covered in the following section.

4.5 Model Design for Tool Wear Segmentation

Modellerstellung zur Segmentation von Werkzeugverschleiß

This section gives an answer to Research Question 1: "*How can image processing be applied to automatically segment tool wear on microscopic images of cutting tool edges?*". Subsection 4.5.1 lists the most important hardware and software used for the experiments in this section. Subsection 4.5.2, Labeling Areas of Interest in the Image Data, briefly describes the workflow required for dataset creation for the creation of a segmentation model. In Subsection 4.5.3, Model Setup and Training, the model architecture and hyperparameters are shown. Subsection 4.5.4, Evaluation of Model Performance, describes the methods to assess segmentation model quality. The content of this section has partially been published by BERGS in the Proceedings of the North American Manufacturing Research Conference (NAMRC) [BERG20].

4.5.1 Experimental Setup

Experimenteller Aufbau

The workflow to arrive at a U-Net model for tool wear segmentation described in this Section is highly recursive, since each of the steps possibly influences the next step and, further down the line, also influences the model performance.

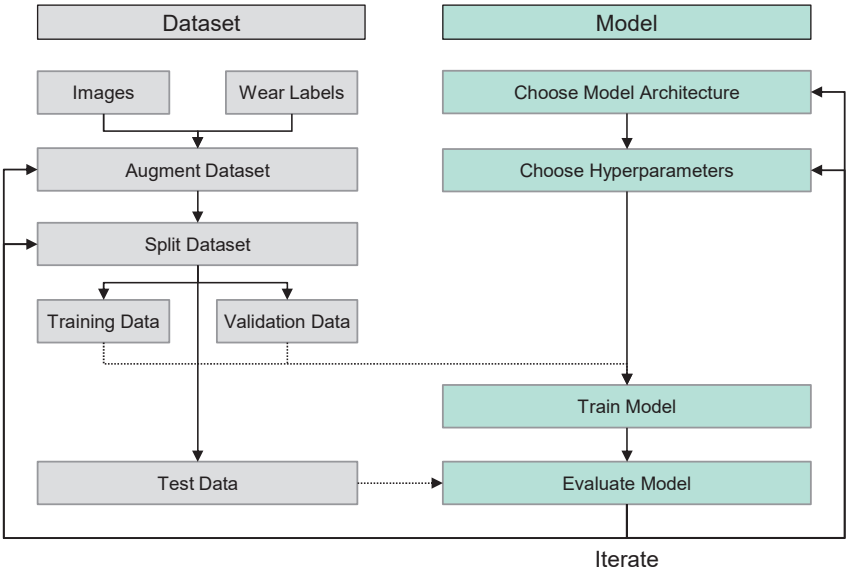


Figure 4-12: Overview of the iterative workflow for model optimization in ML
Überblick zum iterativen Arbeitsablauf bei der ML-Modelloptimierung

The model and its hyperparameters described in this chapter resulted from more than 40 heuristic iterations. Each iteration included changes of one or more hyperparameters which, according to human discretion, could possibly lead to an enhanced model performance. Figure 4-12 shows a schematic of the said workflow which ends once the operator is satisfied with the models' performance.

For NN training a Lenovo workstation type ThinkStation P920 with a NVIDIA Quadro P4000 Graphical Processing Unit (GPU) was used. The GPU accelerates NN training compared to Central Processing Units (CPU) mainly due to their higher memory bandwidth. GPUs can process large amounts of data in parallel. NVIDIA, a GPU manufacturer, provides an application-programming interface (API) named CUDA to enable using GPUs efficiently for general computation tasks like DL with support for several DL libraries.

Apart from the Python programming language and DL libraries, an open-source software called labelme applies for creating the image masks with information about the occurrence of wear on the tool [KENT21]. The created masks are called ground truth because they reflect the true answer about the wear localization. This information is required to train an AI model for the segmentation task. The following subsection describes the labeling process and its importance.

4.5.2 Labeling Areas of Interest in the Image Data

Markierung der relevanten Bereiche in den Bilddaten

Segmentation in image processing with deep learning is a supervised learning task. This means a training data set of images and label maps is required. The label maps are basically black and white images where everything is black except for the area of interest on the image, see Figure 4-13. The pair consisting of input image and the respective label map are used in the training process together with other pairs for network training. The label process is crucial since training data quality has several implications with regards to the training and evaluation process of image segmentation models. The more accurate the annotation, the better a network learns to differentiate the area of interest from background. On top of that the annotations are used for network evaluation on unsee test data. If the test data is poorly labeled a network that does accurate predictions will get bad mean scores of accuracies.

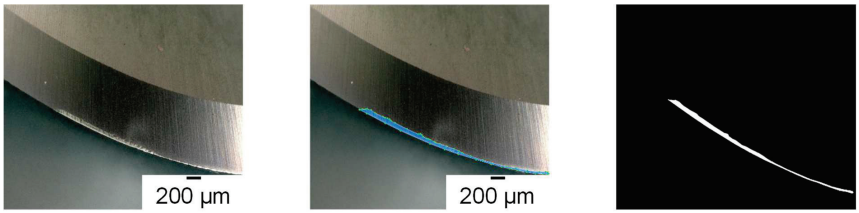


Figure 4-13: Original image (left), image with annotation (middle), label mask (right)
Originalbild (links), Bild mit Annotation (mitte), Labelmaske (rechts)

4.5.3 Model Setup and Training

Modellaufbau und -training

For a semantic segmentation task, there are more than 50 possible architectures available [ACTI19]. As described in Subsection 2.3.3, U-Net was derived from the FCN architecture and is the most influencial architecture with regards to its more than 40,000 citations. Apart from its popularity, the use case of tool wear detection resembles the medical image segmentation task U-Net was made for. This includes the comparably high image resolution and the small size of available data. The U-Net architecture consists of two main parts: a contracting path that captures context and a symmetric expanding path that enables precise localization. The contracting path is composed of convolutional layers followed by max pooling layers, while the expanding path consists of upsampling layers followed by convolutional layers.

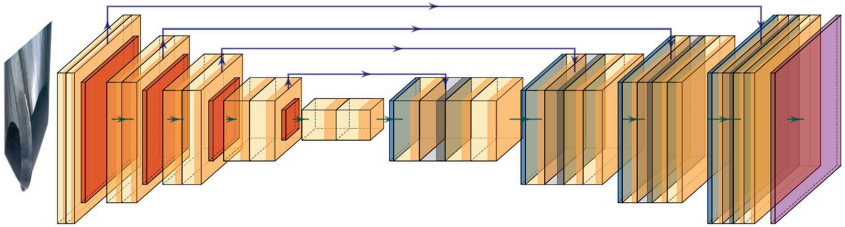


Figure 4-14: U-Net architecture with layers convolution (yellow), pooling (orange), upsampling (blue), softmax (violet) and skip connections (violet arrows)
U-Net Architektur mit Schichten Faltung (gelb), Pooling (orange), Hochtasten (blau), Softmax (lila) und Übersprungverbindungen (lila Pfeile)

The U-Net architecture also includes skip connections that allow information from the contracting path to be directly passed to the corresponding location in the expanding path, see Figure 4-14. These skip connections bring the following feature that likely lead to the popularity of the architecture: They help retain high-resolution features that would otherwise be lost during downsampling. The U-net architecture performs semantic segmentation since it classifies each pixel of the input image.

Table 4-4: Image data and network specifications
Spezifikationen der Bilddaten und des Netzwerkes

Parameter	Value
Image Size	512x512x3
Epochs	200
Trainable Parameters	1,941,105
Learning Rate	0.0001 (ADAM)
Train / val / test split	0.8 / 0.1 / 0.1

Parameters of image and training properties used in the U-Net training process for semantic segmentation are given in Table 4-4. The full architecture can be found in A.9, U-Net Layers and their corresponding output feature map and kernel size. A more in-depth introduction fo U-Net is given in Section 2.3, Image Processing with Deep Learning.

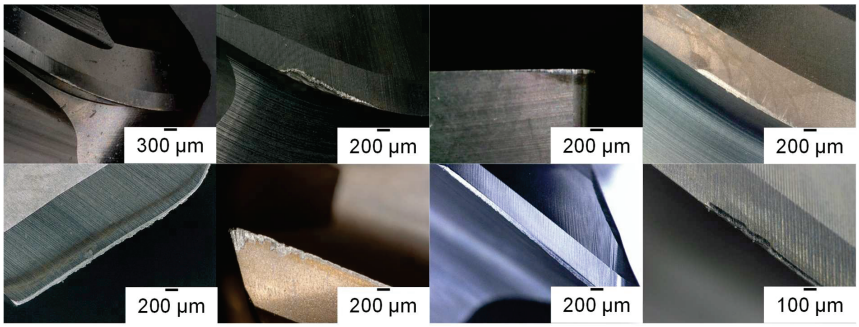


Figure 4-15: Samples of image data used for network training
Stichprobe der Bilddaten für das Netzwerktraining

The image size was chosen as a compromise between computational demand on the available hardware and the preservation of details required for the task of tool wear identification. The other parameters were chosen after a series of heuristic optimization trials. The image database consists of 400 microscopic images recorded with measuring microscopes. Magnifications levels of the images range from x20 to x200. Figure 4-15 shows a subset of images used for the network training. Table 4-5 contains metadata from the various tools' cutting operations.

Table 4-5: Metadata of the cutting tool images used for NN training
Metadaten zu den Zerspanungswerkzeugen für das NN-Training

Tool Type	Workpiece Material	Operation	Operation Type	Th. Cutting Speed [m/min]
Ball End Milling Cutter	2.4668	Turbine Blade Milling	Finishing	30-80
Ball End Milling Cutter	1.2379	Fundamental Cutting Test, Linear Cuts	Finishing	300-500
End Milling Cutter	3.7165	Fundamental Cutting Test, Linear Cuts	Semi-Finishing	150
End Milling Cutter	2.4668	Fundamental Cutting Test, Linear Cuts	Finishing	50-100
Insert	2.4668	Cylindrical turning	Roughing	160
Insert	various	Cylindrical turning	unknown	unknown
Drilling	unknown	unknown	unknown	unknown

The image dataset contains cutting edges of end milling cutters, ball end milling cutters, indexable inserts and drilling tools as shown in Figure 4-15. Using Basic Image Augmentation (BIM) methods, the dataset is increased to 3000 augmented variations of the original images.

This means from each original image seven to eight derivatives were created. In general there are BIM methods such as rotation, horizontal flip, vertical flip, zoom, translation, brightness adjustment, contrast adjustment, shear, color jittering such as hue, saturation and brightness, cropping and resizing. The augmentation setting implemented with the imaug library are shown in Table 4-6.

Table 4-6: BIM methods and values respectively value ranges
BIM-Methoden und Werte bzw. Wertebereiche

Method	Values
Flip	0.5
Multiply	4.9 – 1.1
Rotate	- 90° – + 90°
Blur Sigma	1
Contrast Normalization	0.9 – 1.1

Figure 4-16 shows an example of augmented images and respective annotations created in this process. In the specific case of the figure, the images belong to a dataset of microscopic drilling tool edge images. The augmented images in the figure are all based on the same tool, through the image manipulation the images appear to be quite different from each other. It is crucial for the training of the neural network, that the label masks belonging to the images experience the exact same augmentation to ensure that the model learns to locate the area of interest correctly.

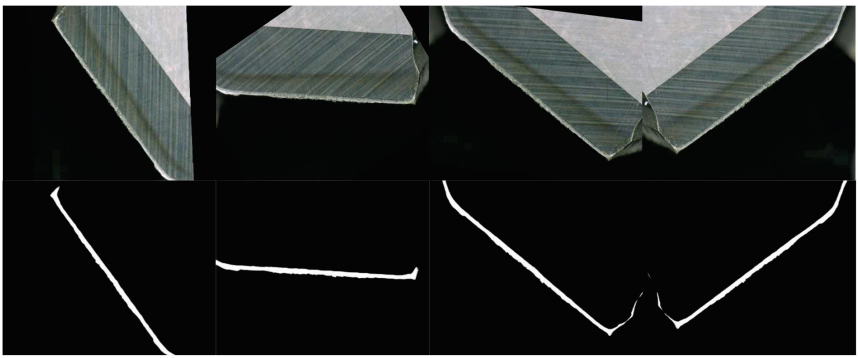


Figure 4-16: Augmented drilling tool cutting edge and respective labels
Augmentierte Bohrwerkzeugschneiden und die jeweiligen Label

4.5.4 Evaluation of Model Performance

Evaluierung der Modellperformanz

There are quantitative and qualitative methods to evaluate the performance of a model. Most common are KPIs to determine a mean accuracy of test results. Apart from that an expert can evaluate if she/he is satisified with the prediction quality for a specific use case. As introduced in Section 2.3 the Dice coefficient is a common quality metric for semantic image segmentation tasks. The mean Dice coefficient of the training dataset is $mDice_{train}$ = 0.96. The mean Dice coefficient of the validation dataset is $mDice_{val}$ = 0.88. The mean Dice coefficient of the test dataset is $mDice_{test}$ = 0.84.

There are five examples of test data given in Figure 4-18. For each example the figure contains the original image on the very left. A manually annotated ground truth, overlayed in white color, in the middle. The predicted tool wear, overlayed in red color, as well as the respective Dice coefficient is displayed on the right. The overall performance is satisfying since the tool wear is detected on the images. In the second row however, there is an example where a reflection of light is misclassified as tool wear. That means the model is prone to making errors when external disturbances are present.

Unknown and disturbed tool wear images, as in the inference dataset, yields a mean Dice of $mDice_{inf}$ = 0.54 with tendency of the network to misrecognize irrelevant edges and scratches as well as missing out bits of large worn areas. This means a generalization that allows inferring the model to make reliable predictions on unknown image data with disturbances could not be reached. An example of a misrecognition of an edge as tool wear is visible in Figure 4-18. In the second row and third column there is a white circle enhancing the said misrecognition.

A further approach to train models only on one individual tool type yield similar results compared to the model trained on a mixed dataset, see Table 4-7. The models for individual tool types have a very comparable or even higher accuracy in terms of training and test Dice coefficient. These models are trained with only one tool type and are therefore based on a fraction of the data in terms of number of images.

Table 4-7: Approaches to model creation with mixed and individual datasets

Ansätze zur Modellerstellung mit gemischtem und einzelnem Datensatz

Dataset Size	Dataset Size (BIM)	Tool Type	$mDice_{train}$	$mDice_{test}$
400	3000	Mixed dataset	0.96	0.84
100	750	Individual ball end milling cutter	0.95	0.82
100	750	Individual end milling cutter	0.96	0.83

100	750	Individual indexible insert	0.98	0.83

In conclusion this means for further work it is a viable approach to train a classifier model as proposed in the original paper to distinguish tool types and then pass over the segmentation task to a network specialized for this very tool type. The two general approaches of a general model and tool-type-specific models is shown conceptually in Figure 4-17.

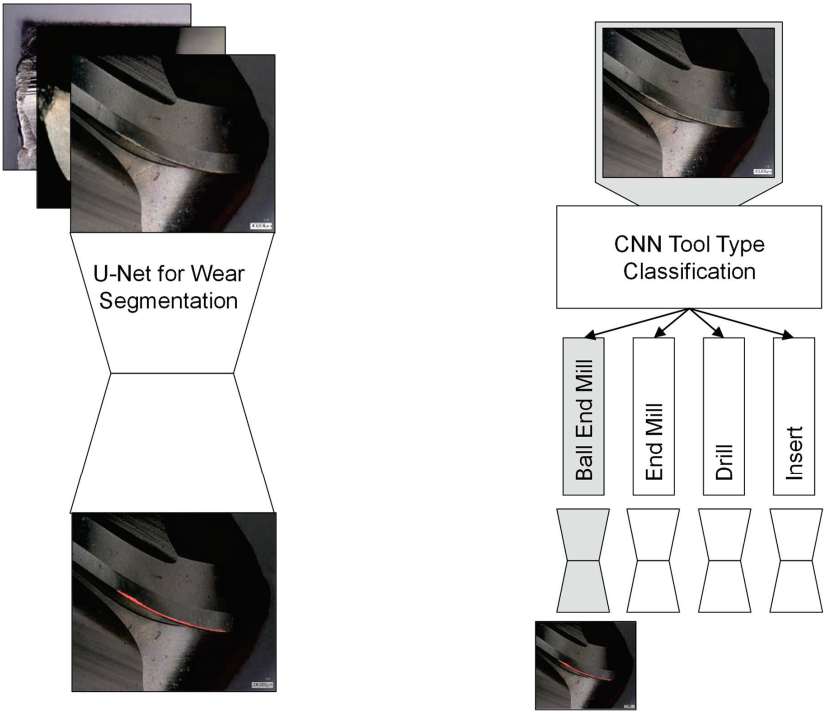


Figure 4-17: Visualization of an approach to train one general model for all tool type (left) or to train a classifier model and tool-type-specific segmentation models (right)
Visualisierung eines Ansatzes zum Trainieren eines Modells für alle Werkzeugtypen (links) oder zum Trainieren eines Klassifikatormodells und werkzeugtypspezifischer Segmentierungsmodelle (rechts)

A generalization that allows inferring the model trained with a mixed dataset to make reliable predictions on indexable inserts, although it was trained on ball end milling cutters and end milling cutters, could not be reached. This goal might be attainable with more training examples. Inference predictions with the tool-type-specific models did not give valuable results. Figure 4-19 shows inference data samples in the same

manner as Figure 4-18 for the test data. Both figures contain predictions made by the general model that was trained on the mixed dataset.

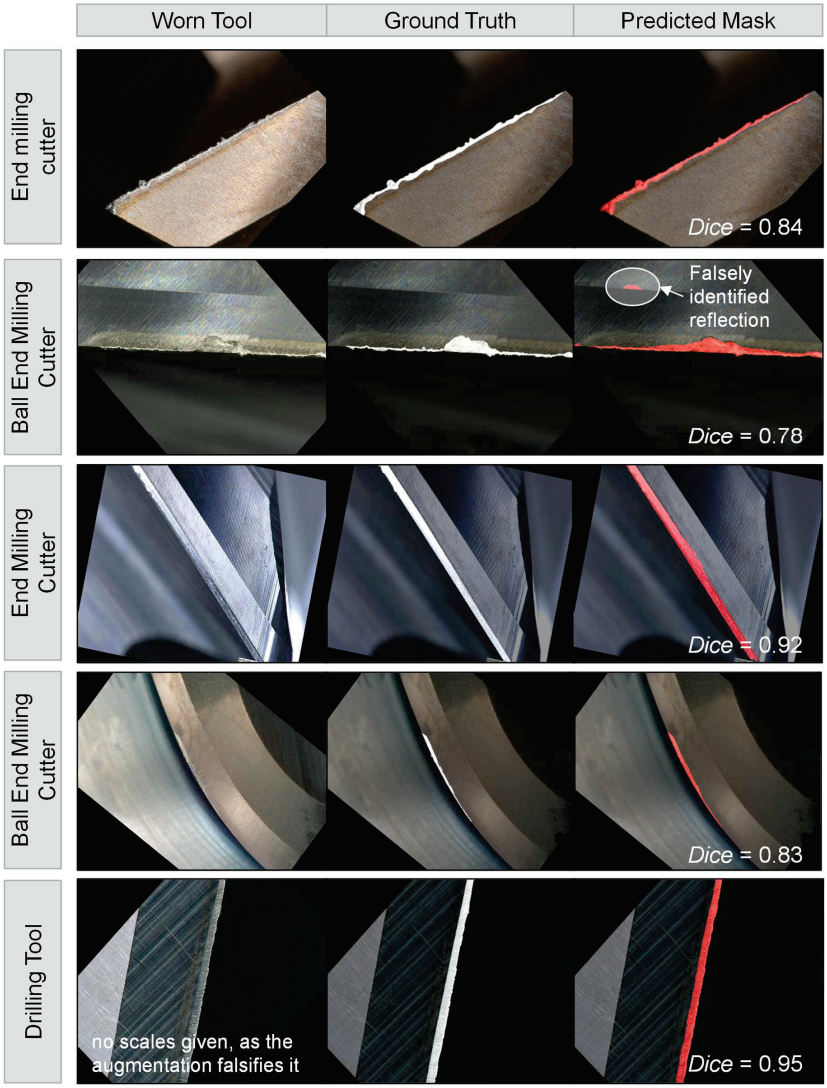


Figure 4-18: Examples of original, ground truth and wear detection on test data
Beispiele der Originale, Labelmasken und Detektionen von Verschleiß auf Testdaten

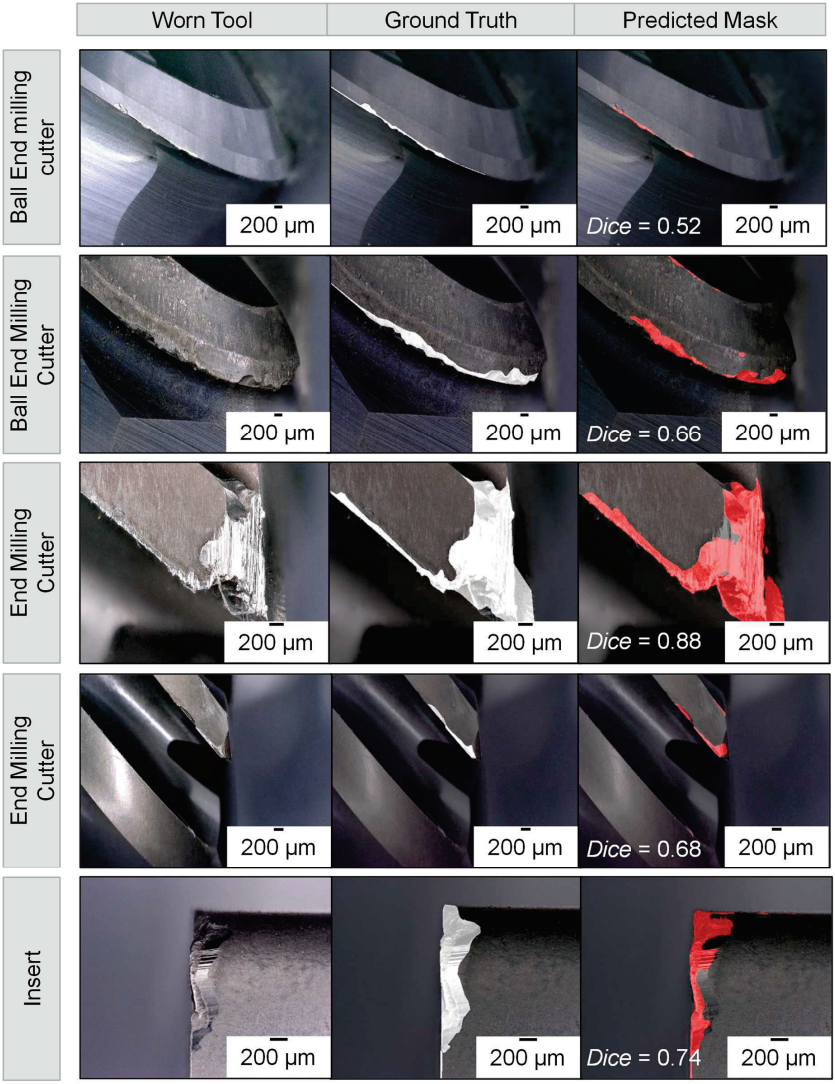


Figure 4-19: Examples of original, ground truth and wear detection on inference data
Beispiele der Originale, Labelmasken und Detektionen von Verschleiß auf Inferenzdaten

4.6 Interim Conclusion

Zwischenfazit

Chapter 4 Tool Wear Modelling and Segmentation gives an answer to Research Question 1: "*How can image processing be applied to automatically detect tool wear on microscopic images of cutting tool edges?*"

A Deep Learning approach using the U-Net architecture was used to perform semantic image segmentation for the sake of automated tool wear analysis of metal cutting tool edges. In more detail the following steps were necessary:

- Aggregation and selection of microscopic images with worn cutting tools
- Manual label process to create masks of the areas of interest (tool wear) on each of the images. The label masks serve as ground truth for network training
- BIM methods were applied to increase the dataset size and enhance the variance in the data base. GAN-based data synthesis was proposed for future investigations
- An iterative selection of dataset properties and model hyperparameters such as dataset split, network size, learning rate and dropout rate was conducted
- Conduction of model training process, i.e., backpropagation algorithm
- Evaluation of model performance with metrics that describe model accuracy and robustness regarding test dataset was performed

The tool wear segmentation approach using the U-Net architecture for semantic segmentation presented in this chapter achieved a mean Dice coefficient of $mDice_{test} = 0.82$ on test data. Training data consisted of 3000 augmented images originating from 400 raw images. Eight different cutting tool datasets with 50 images each and various levels of magnification made up the heterogeneous raw image dataset. On an inference dataset, which contains unknown images recorded with disturbances like increased or decreased brightness, the network yielded a Dice coefficient of $mDice_{inf} = 0.54$. Additionally, it was found that models trained on individual, homogeneous datasets tend to perform at least as well as larger mixed models on their held-out test data using the U-Net architecture.

The workflow described above is highly recursive since each of the steps possibly influences the next step and, further down the line, also influences the model performance. The model and its hyperparameters described in this chapter resulted from more than 40 heuristic iterations. Each iteration included changes of one or more hyperparameters or dataset properties which, according to human discretion, could possibly lead to an enhanced model performance. Figure 4-12 shows a schematic of the said workflow which ends once the operator is satisfied with the models' performance. The following chapter describes a systematic method to reduce the need for a heuristic search of a well performing image segmentation model.

5 Model Performance Optimization

Optimierung der Modellperformanz

This chapter describes a method to create a decision model for hyperparameter selection in deep learning semantic image segmentation based on the dataset properties. The approach for model performance optimization is described in Section 5.1, Methodology. Section 5.2, Prerequisites and Definitions, introduces basic terms in NN modelling to understand the investigated factors. The approach explained in Section 5.1 is split into the following three major steps, detailed in Section 5.3 to 5.5. Section 5.3 contains a screening analysis to grade possibly important factors, answering the second research question:

> RQ2: What are the dataset and model properties with the highest impact on model performance for tool wear segmentation?

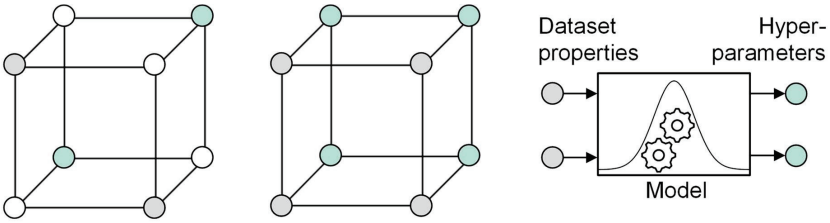
Section 5.4, Full Factorial Analysis, contains a consolidated analysis of possibly important factors with regards to model evaluation metrics. Finally, in Section 5.5, a decision model is created based on the the experimental data of the former section to answer the third research question:

> RQ3: How can a systematic choice of hyperparameters with regards to dataset properties be employed to improve model performance for tool wear segmentation?

5.1 Methodology for Model Performance Optimization

Methodik zur Optimierung der Modellperformanz

The deficits described in the final paragraph of the last chapter shall be solved using a systematic approach to hyperparameter selection based on dataset properties.



Step 1 **Screening Design**	**Step 2** **Full Factorial Design**	**Step 3** **Decision Model**
Identification of possibly important dataset properties and hyperparameters	Investigate important parameters and their influence on model evaluation metrics	Model for knowledge-based selection of NN hyper-parameters based on dataset properties

Figure 5-1: Outline of the approach to Model Performance Optimization
Übersicht zum Ansatz der Optimierung der Modellperformanz

As displayed in Figure 5-1 the approach is split into three distinct steps. In the figure, grey indicates dataset properties, whereas turquois indicates model hyperparameters. In step one a screening Design of Experiments (DOE) is made including a broad selection of hyperparameters and dataset properties. The respective experiments are conducted and analysed with regards to the model evaluation metrics. In step two the most important hyperparameters and dataset properties are selected and processed into a full factorial DOE. Again, the respective experiments are conducted and analysed. Finally, in step three the results from the full factorial DOE are used to create a decision model that enables the selection of favorable hyperparameters based on dataset properties. The approach is further compared to benchmark models to validate the findings. A reader familiar with the ML topic may skip Section 5.2, Prerequisites and Definitions, and head to Section 5.3, Screening Analysis.

5.2 Prerequisites and Definitions

Voraussetzungen und Definitionen

Subsections 5.2.1, Model Hyperparameters, and 5.2.2, Dataset Properties, introduce factors investigated in the following sections. Subsection 5.2.3, Model Evaluation Metrics, describes the assessment criteria applied for NN model optimization.

5.2.1 Model Hyperparameters

Modell-Hyperparameter

In this work hyperparameters are defined as any parameter in the NN configuration that is not directly learnable during the training process. This definition includes network architecture parameters such as number of network layers, i.e., network depth, and kernel size for convolutions and pooling, i.e., network width. Other hyperparameters are activation functions, learning rate, momentum, dropout rate and batch size (the number of training data samples that are used in one epoch). In this section some hyperparameters important for further procedure are briefly described.

Activation Functions

Aktivierungsfunktionen

These functions are covered in Subsection 2.3.1, Fundamentals of Machine Learning.

Learning Rate

Lernrate

A major difficulty in training neural networks is determining the appropriate hyperparameters like learning rate because large learning rates might overshoot the solution surface and low learning rates end up being too slow while converging on a solution. To improve performance, learning rate scheduling is used as an extension of the SGD algorithm [KIEF52]. The learning rate can be described as a decreasing function of the iteration number. Thus, first few iterations have larger learning rates causing larger change in parameters and as the iterations continue the learning rate decreases. An

overview of some gradient descent optimization algorithms is given in the paragraph below [RUDE17].

Momentum

Momentum

Momentum speeds up SGD in the relevant direction by adding a fraction α of the previous update to the current update. The Momentum update rule is given in Equation (14). With the following variables: momentum parameter, $\Delta\theta$, fractional hyperparameter, α_θ, learning rate, η, and finally the gradient direction of the loss function, $\nabla_\theta L(\theta)$. The loss function parameter θ calculates according to Equation (15).

$$\Delta\theta_t = \alpha_\theta \, \Delta\theta_{t-1} - \eta \nabla_\theta L(\theta) \tag{14}$$

$$\theta_t = \theta_{t-1} + \Delta\theta_t \tag{15}$$

RMSprop is an adaptive learning rate method which tackles the problem of accumulation of squared gradients [HINT19]. The learning rate of RMSprop is divided by an exponentially decaying average of squared gradients. Adadelta reduces the problem of decreasing learning rate [ZEIL12]. The range of accumulated squared gradients is restricted to a certain fixed size. The adaptive learning rate for each parameter is also determined by Adaptive Moment Estimation (ADAM) optimizer [KING14]. An exponentially decaying average of past squared gradients v_t are stored by Adadelta and RMSprop but ADAM also keeps an exponentially decaying average of past gradients m_t. The estimates of the mean and the uncentered variance of the gradients are given by vectors v_t and m_t respectively which are biased towards zero. These bias-corrected estimates $\hat{v}_t$ and $\hat{m}_t$ are calculated for the update rule where ϵ is a smoothing constant.

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{16}$$

Dropout Rate

Dropout-Rate

A major problem in neural network training is overfitting. This means loss in the generalizing capacity of the model i.e., the model learns to map the noise in training dataset too. Overfitting can be caused by training a high parameter network on a small training dataset [RUSS04, p. 909]. Dropout layers are used for preventing overfitting in neural network models. Different nodes and its connections are randomly dropped from the network during dropout. This results in the neuron units not being overfit to the same data and being able to adapt to rest of the training set [SRIV14]. Additionally, the dropout layers may be used for uncertainty estimation of the network's prediction through repeated inferences with random dropout. In Subsection 2.3.4, Tool Wear Identification with Deep Learning, the so-called Monte-Carlo based dropout is briefly described.

Network Size
Netzwerkgröße

The network size dimensions are depth and width. In this investigation the depth is constant due to the selection of vanilla U-Net as model architecture. The network width depends on the number of kernels, *d*, which produce the feature maps within the layer. The number of parameters within one convolutional layer calculates as in Equation (17). The parameters in all convolutional and FC layers combined yield the total number of network parameters, compare Subsection 2.3.3, Image Processing with Artificial Intelligence.

$$n_{conv} = \left(\left(h\ w\ d_{previous}\right) + 1\right) d_{current} \tag{17}$$

5.2.2 Dataset Properties
Datensatzeigenschaften

In this work, dataset properties are defined as parameters that describe the dataset. This definition includes the dataset size, the image size, the dataset split as well as the similarity of the images within the dataset. Dataset split could also be attributed to the hyperparameters and not dataset properties. For the sake of clarity this work assigns dataset split as a dataset property. Since data augmentation was covered in literature, this work abstains from including it into the scope of investigation. The following paragraphs include a brief description of the four factors assigned to dataset properties.

Dataset Size
Datensatzgröße

The number of raw images, without count of each images respective label mask, included in a dataset for network training is defined as the dataset size. The typical representation for dataset property dataset size is simply a scalar, such as 50 images.

Image Size
Bildgröße

The number of pixels along the xy-plane, i.e., image resolution, of the images included in a dataset for neural network training is defined as image size. A typical representation of this dataset property is (x-pixels, y-pixels). In an example this could be (512, 512) which means there are 512 pixels along the x and y dimension of an image. That means the image contains 262,144 pixels on the xy-plane, not considering possible color channels along the z-axis of the image matrix.

Dataset Split
Datensatz-Aufteilung

Seperating a dataset for machine learning model creation into training dataset, validation dataset and test dataset is defined as dataset split. It determines what fraction of the data is used in the training process and in the model assessment, i.e., testing,

process. It also determines what fraction of data is reserved for an unbiased evaluation of the current model during the training process, to prevent the model from optimizing the hyperparameters for a local minimum of the objective function.

In other words, the validation data split is required to prevent the neural network model from quickly overfitting, i.e., memorizing the training data. A typical representation of the dataset split is (training dataset, validation dataset, test dataset). In an example where 80 % of the data is used for the training dataset and 10 % for the other two the nomenclature for dataset plit is (0.8 / 0.1 / 0.1).

Dataset Similarity
Datensatz-Ähnlichkeit

The homogeneity, or respectively the heterogenity, of a dataset may be a good descriptor for the variance of the problem domain a neural network has to learn during training. In heuristic approaches to hyperparameter optimization, as mentioned in Section 4.6, it became apparent to the operators, that the perceptual similarity of images within a dataset influence the sensitivity of model performance dependant on hyperparameter choice.

To confirm or neglect this hypothesis the dataset similarity was included into to scope of this investigation. Since there are several methods available to quantify image similarity, a study was conducted to assess their performance and agreement. In the scope of this study two different groups of tasks were prepared: quantification of inner and outer similarity. Inner similarity describes the mean similarity of pairwise comparisons of all images within a specific dataset. Outer similarity describes the mean similarity of pairwise comparison of images between two different datasets. For the assessment of the image similarity algorithms, an operator rating of the similarity was produced additionally.

The aim of this investigation is to find an algorithmic metric that has the highest agreement with a human operator assessment of the similarity between two datasets. The human assesment is taken as a benchmark since human brains are wired to process visual information quickly and efficiently, and can easily recognize patterns, shapes, colors, and other features in images [THIB17]. A recent study suggest that humans are still able to outperform AI models in cases of categorization [VINT19]. Using a correlation analysis, specifically Pearson's *r* defined in Equation (18), the algorithm with the highest human operator agreement was identified for further analysis.

$$r = \frac{n(\sum x_i y_i) - \sum x_i \sum y_i}{\sqrt{(n\sum x_i^2 - (\sum x_i)^2)(n\sum y_i^2 - (\sum y_i)^2)}} \tag{18}$$

Where n is the number of samples and x_i and y_i are the individual values of the variables x and y. In detail, the inner similarity of six datasets with 30 images each was calculated for each unique pairwise combination of image for each algorithm, allowing

the calculation of a mean correlation coefficient with regards to the operator assessment, r_{inner}, see Figure 5-2. The complete data is in A.10 Inner similarity calculations.

For the outer similarity, see Figure 5-3, each of the unique combinations of pairs of two of the six datasets where calculated. Further the mean correlation coefficient with regards to the operator assessment was calculated, r_{outer}.

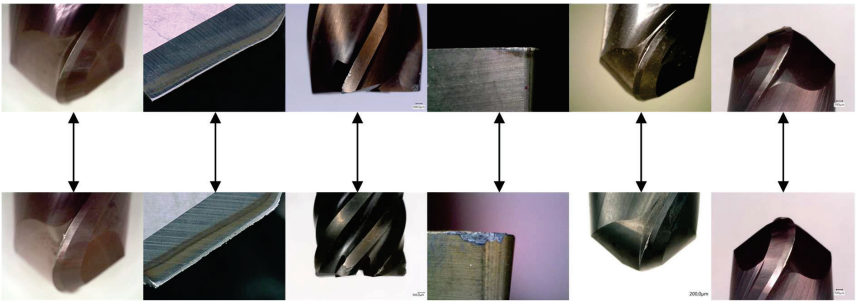


Figure 5-2: Exemplary data for display of inner similarity analysis
Beispielhafte Bilddaten zur Darstellung der Analyse der inneren Ähnlichkeit

The figure shows conceptually that for each pair of images between two datasets of 30 images the similarity metrics are determined. This procedure was conducted for each combination of datasets, which is not displayed in the figure for the sake of clarity. The complete data is in Annex A.11, Outer similarity calculations, and A.12, Outer similarity values at different dataset size levels.

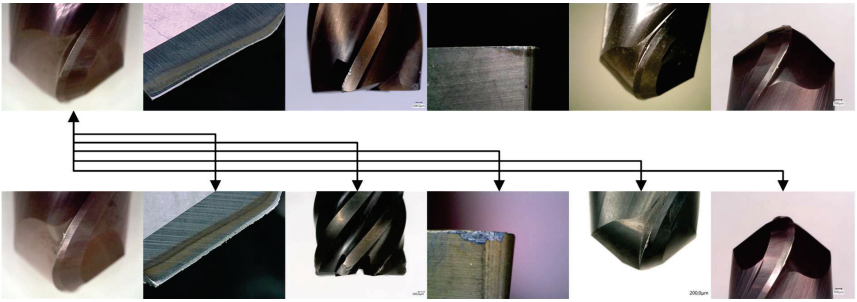


Figure 5-3: One example of the conduction of an outer similarity analysis
Ein Beispiel für die Durchführung der Analyse der äußeren Ähnlichkeit

The following algorithms were applied on a test dataset, including tool images and artificial images with adverse brightness levels, upfront the main investigation:

- Root Mean Square Error (RMSE) described by [MÜLL20]
- Peak Signal-to-Noise Ratio (PSNR) described by [MÜLL20]
- Structural Similarity Index (SSIM) described by [MÜLL20]

- Signal to Reconstruction Error ratio (SRE) described by [MÜLL20]
- Spectral Angle Mapper (SAM) described by [MÜLL20]
- Universal Image Quality index (UIQ) described by [MÜLL20]
- Information Statistic Similarity Measure (ISSM) described by [MÜLL20]
- Feature-based Similarity index (FSIM) described by [MÜLL20]
- Learned Perceptual Image Patch Similarity (LPIPS) described by [ZHAN18]

The FSIM was omitted since it failed at cases of strong dissimilarity between two images. UIQ, ISSM and LPIPS were excluded due to their computational requirements which would have led to unacceptable compute time when applied to the actual task. Besides the remaining algorithms to measure image similarity, the human operator estimation was added to the metrics to find the highest correlation between one of the above algorithms and the human assessment.

For the investigation regarding inner similarity, that is similarity within one dataset, the similarity metrics were calculated between each unique combination of two images within one dataset. The mean similarity for each dataset was calculated for each metric from these pairwise similarity values. Afterwards the correlation between each similarity metric was calculated yielding the following correlation matrix, see Figure 5-4.

	PSNR	SSIM	SRE	SAM	**Human**
RMSE	0.99	0.90	0.63	0.52	**0.89**
PSNR		0.88	0.61	0.51	**0.89**
SSIM			0.89	0.83	**0.91**
SRE				0.97	**0.68**
SAM					**0.66**

Figure 5-4: Mean Pearson's correlation matrix for metrics compared with human opinions on datasets for assessing inner similarity
Durchschnittliche Pearson-Korrelationsmatrix für Metriken im Vergleich zur menschlichen Meinung zur Beurteilung der inneren Ähnlichkeit.

Surprisingly, some of the different image similarity metrics have a very uneven correlation with each other for this task, especially the SRE and SAM. Whereas the RMSE, PSNR and SSIM correlate strongly. The same is true with regards to the correlation of the different metrics compared to the human operator.

The RMSE, PSNR and SSIM show a correlation of approximately 90 % with the human estimation of mean similarity within one dataset. For the investigation regarding outer similarity, that is similarity between two datasets, the similarity metrics were calculated between each unique combination of two images from two datasets.

The mean outer similarity for each dataset was calculated for each metric from these pairwise similarity values. Afterwards the correlation between each similarity metric was calculated yielding the following correlation matrix shown in the Figure 5-5.

	PSNR	SSIM	SRE	SAM	**Human**
RMSE	0.99	0.86	0.76	0.53	**0.36**
PSNR		0.82	0.68	0.45	**0.33**
SSIM			0.85	0.82	**0.58**
SRE				0.70	**0.48**
SAM					**0.44**

Figure 5-5: Mean Pearson's correlation matrix for metrics compared with human opinions on datasets for assessing outer similarity
Durchschnittliche Pearson-Korrelationsmatrix für Metriken im Vergleich zur menschlichen Meinung zur Beurteilung der äußeren Ähnlichkeit

The RMSE, PSNR and SSIM show a strong correlation among each other, as in the inner similarity investigation. The SRE shows mediocre to strong correlation with the other metrics. Between SRE and SSIM the correlation is at 0.85. The SAM has the worst correlation with the other metrics. The agreement with the human operator in terms of Pearson's correlation regarding the outer similarity is significantly lower, that is between 30 to 50 %, than for the task of inner similarity across all the image similarity metrics investigated in this study.

As described above, the Structural Similarity Index (SSIM) turned out to be the most suitable algorithm for the required tasks. The implementation of SSIM from the python library for image processing scikit-image was used [AVAN09, WANG04]. A typical representation of image similarity in terms of SSIM is a scalar between zero and one, where zero is not similar and one identical. The calculation of the SSIM can be conducted on various windows of images or on the complete images. In this case the image size as well as the window size is 512x512 and the SSIM is calculated using the following formula:

$$SSIM(w1, w2) = \frac{(2\mu_{w1}\mu_{w2} + c_1)(2\sigma_{w1w2} + c_2)}{(\mu_{w1}^2 + \mu_{w2}^2 + c_1)(\sigma_{w1}^2 + \sigma_{w2}^2 + c_2)} \quad (19)$$

Where μ_{w1} and μ_{w2} are the pixel sample means of the first and second image or window. The variance of these values is σ^2_{w1} and σ^2_{w2} respectively. The covariance of image one and two is denoted σ_{w1w2}. The variables c_1 and c_2 stabilize the division if the denominators are otherwise too small. As described above in more detail, for calculation of the similarity metric in the task of inner and outer similarity in this subsection, the similarity was calculated in terms of the mean SSIM of the pairwise unique combination of images.

5.2.3 Model Evaluation Metrics

Bewertungsmetriken für Modelle

Model evaluation metrics are required to assess a model's performance and to compare different models with each other. To capture the two most important model performance properties in machine learning, which are goodness of fit and generalization capability, two different metrics are required.

Accuracy Metric

Genauigkeitsmetrik

F1 or Dice coefficient is a widely used evaluation metric in machine learning that measures the performance of a classification model [DAVI06]. It is the harmonic mean of precision and recall, which are two important metrics used to evaluate a classifier's performance. Precision indicates the ability of a classifier not to label a sample as positive that was negative. In other words, it measures how accurate the classifier is when it identifies positive cases. On the other hand, recall evaluates the ability of the classifier to find all positive samples. It measures how well the classifier identifies all positive cases, including the ones that are missed or misclassified. The Dice coefficient is a specific way of calculating F1 that is commonly used in image segmentation tasks. It is calculated by taking two times the area of overlap between two images and dividing it by the total number of pixels in the two images. For more details on this method, please refer to Subsection 2.3.3.

To calculate the Dice coefficient, two masks are required: a manually labeled ground truth mask and the respective wear mask predicted by the model. This metric is reported as a mean of all images from a specific dataset, such as a test dataset $Dice_{test}$, as datasets typically contain many images. Overall, the Dice coefficient is an important metric that provides a reliable measure of how well a classification model performs. By calculating this metric, we can determine the accuracy of the model's predictions and compare different models to select the one that performs the best.

Overfitting Metric

Überanpassungsmetrik

Overfitting refers to a machine learning model that learns to model the training data too well. That is, noise is picked up and learned as important concept by the model, leading to poor performance when data is presented which is not part of the training dataset. Simply put, overfitting means good performance on the training data, poor generalization on other data. Currently, there is no standard metric to measure overfitting. A common definition of overfitting during NN training is the Epoch at which training and validation loss start to diverge, see Figure 5-6. Typically, the training and validation accuracy also starts to diverge at the same time, because the relation between loss and accuracy is inversly proportional.

Since loss functions and their scale are less intuitive, here, an overfitting metric is constructed using the accuracy values, which range between zero and one for the F1 score, respectively the Dice Coefficient. Furthermore, the overfitting metric constructed for the purpose of model evaluation and comparison is normalized, to obtain a percentual metric, see Equation (20). Additionally, using the Dice as a basis allows to construct a derative of this metric for test data as well.

$$OP_{TV} = \frac{Dice_{train} - Dice_{val}}{Dice_{train}} \tag{20}$$

In short, the OP_{TV} (Overftting Percentual Training Validation) describes the percentual difference between training and validation accuracy. In terms of Figure 5-, the relative distance between the green and blue curve. The notation is chosen to allow a consistent derivation of similar metrics, such as OP_{TT} (Overftting Percentual Training Test) or OP_{VT} (Overfitting Percetual Validation Test). In this thesis the Overfitting Percentual Training Test is chosen as an overfitting metric for the following reasons: It indicates cases with higher test accuracy than training accuracy with negative values, which allows filtering for model that are good by chance, due to unfavorable dataset split. This is especially important for small datasets. The OP_{TT} also covers cases where the training accuracy is very high, and the test accuracy is high in absolute terms. In such cases, it is hard to identify overfit networks solely based on test accuracy.

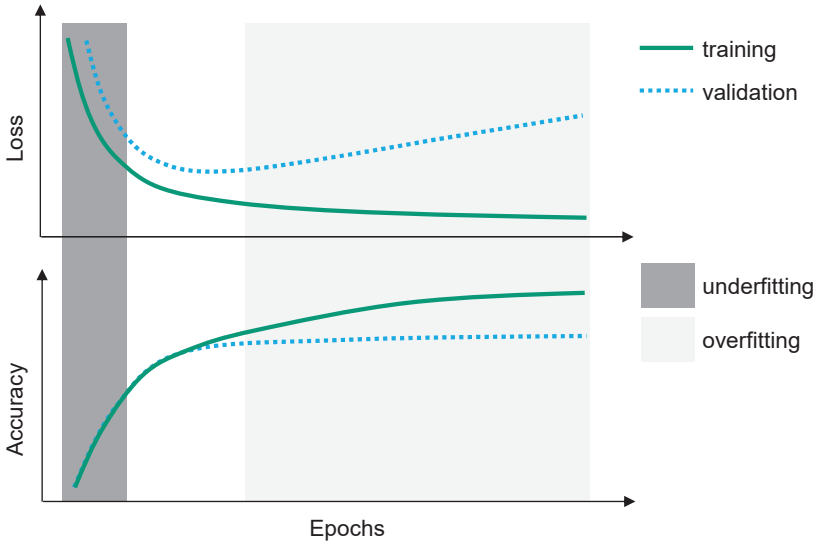


Figure 5-6: Loss and Accuracy during training of a NN
Verlust und Genauigkeit während des Trainings eines NN

5.3 Screening Analysis

Einflussgrößenanalyse

The above Section 5.2 Prerequisites and Definitions introduces the model training, the factors of interest and the metrics required to answer the second Research Questions:

> RQ2: What are the dataset and model properties with the highest impact on model performance for tool wear segmentation?

This section contains a description of the necessary steps to perform a Screening Analysis. It consists of Subsection 5.3.1, Preparation, followed by the Subsection 5.3.2, Significance Analysis, to determine which effects are significant, followed by Subsection 5.3.3, Effect Size Analysis, to determine the magnitude of said effects, concluding with Subsection 5.3.4, Discussion of findings.

5.3.1 Preparation

Vorbereitung

This subsection contains paragraphs describing the necessary steps to set up a screening analysis including the DOE, the sample size estimation, and the dataset itself.

Design of Experiments

Statistische Versuchsplanung

The Generalized Subset Design (GSD) is a generalization of traditional fractional factorial designs to problems where factors can have more than two levels [SURO17]. Previous reduced designs cannot provide this feature. Moreover, full multi-level factorial designs cover this as well, but they produce a non-economical number of experiments. Besides the arbitrary factor levels, the GSD can be used for problems with many factors to be investigated, because it allows a user-specified reduction factor. This allows the thinning out screening designs of experiment where the main goal is the separation of more and less important factors. The Table 5-1 shows all factors and their respective levels for the screening analysis. In a full factorial design with two levels, also denoted 2^k, this setup would yield 512 test points. This might seem feasable at first but considering required repetitions to account for the variance of identical experimental runs, the number of experiments quickly escalates.

Table 5-1: Screening DOE Factors and Levels

Faktoren und Faktorstufen des Versuchsplans zur Einflussgrößenanalyse

	Dataset Properties				Model Hyperparameters				
Factors / Levels	**Dataset Size**	**Image Size**	**Dataset Split**	**Data Similarity**	**Act. Function**	**Learning Rate**	**Dropout Rate**	**Network Size**	**Momentum**
0	50	256	0.8 / 0.1 / 0.1	0.67	ELU	0.0001	0.2	122k	0.8
1	400	512	0.9 / 0.05 / 0.05	0.77	ReLU	0.0005	0.6	486k	0.9

Moreover, the two levels would allow only linear terms if we were to create a model from the DOE. In summary, the GSD is suitable for a screening analysis with many factors and/or varying number of levels in the factors. Therefore, it is a suitable choice for the setup described above with a total of nine factors with two levels each.

Sample Size Estimation
Abschätzung der Stichprobengröße

To prepare the screening analysis a determination of the minimum sample size for each test point is required. From previous repeated model training it is known that the expected standard deviation, σ, of $Dice_{test}$ is 0.0575 from an experiment with ten repetitions, denoted N, which correspond to a normal distribution. The acceptable percentual error threshold, E, is set to 0.05, that is 5 % Dice. The z_N value of 1.96 results from the standard normal distribution for the targeted confidence level of 95 %.

$$n \geq \frac{N\, z^2\, \sigma^2}{z_N^2\, \sigma^2 + (N-1)\, E^2} = 4.69 \tag{21}$$

According to Equation (21), the number of experiments in each test point should be greater than 4.69, that is five [BUND21]. With the repetitions set to five and a reduction factor of six the number of experiments in the GSD DOE reaches 255.

Dataset
Datensatz

The image datasets for the analysis are constructed as described in the instructions specified in the DOE. The data consist of images and their respective label mask. Figure 5-7 below shows examples of original images next to an image with a manually created label mask overlay displayed in the figure white color.

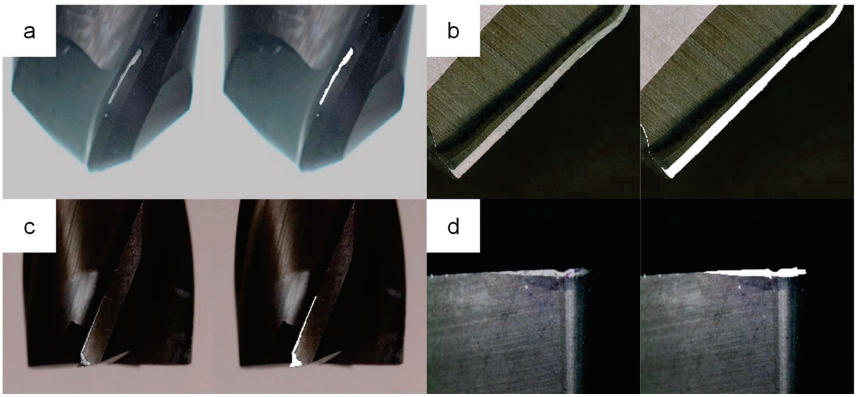


Figure 5-7: **Examples of image data used in the screening analysis: a) Ball end milling cutter, b) Drilling tool edge, c) End milling cutter, d) Indexable insert**
Beispiele von Bilddaten, die im der Einflussgrößenanalyse eingesetzt wurden: a) Kugelkopffräser, b) Bohrer, c) Schaftfräser, d) Wendeschneidplatte

Below is an example for a set of parameters in a test point within the screening DOE. Dataset properties: Assume the number of images to be 50, with an image size of 256x256 pixels, a dataset split of (0.8 / 0.1 / 0.1) and a dataset similarity of 0.67. Model hyperparameter: Assume the activation function to be ELU, the learning rate of 0.0001, the dropout rate of 0.2, the network size of 121.725 parameters and momentum of 0.8. Due to the necessary repetitions derived in the above paragraph this set of parameters is used to train five neural networks with random initial seed and a randomised dataset split with regards to the utilized data for each subset, i.e., training, validation, and test dataset.

5.3.2 Significance Analysis

Signifikanzanalyse

The evaluation of results from the screening experiment is divided into the paragraphs for dataset properties and model hyperparameters. For each set of parameters, the results are analysed with regards to accuracy and overfitting, which are covered in detail in Subsection 5.2.3, Model Evaluation Metrics. The analysis below is based on the Lenth's method for identifying active contrasts in sparse DOEs [LENT89]. A significant standardized effect calculated from the t-statistic gives confidence, that the observed difference between two groups, i.e., a factor and a target variable, is not due to chance. The implementation of Minitab is used to conduct the analysis. A description of the algorithm is given in A.13, Identification of the statistically significant effects in factorial experiment based on Lenth's Analysis.

Dataset Properties

Datensatzeigenschaften

The investigated factors are dataset size and split as well as image similarity and size. According to the screening experiments there is a significant standardized effect for all dataset properties with regards to the mean accuracy of the test dataset $mDice_{test}$.

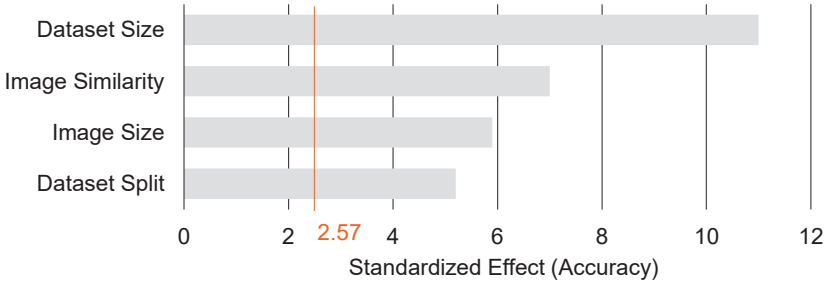


Figure 5-8: Standardized effect of dataset properties with regards to accuracy in terms of $mDice_{test}$ of the test dataset
Standardisierter Effekt der Datensatzeigenschaften mit Bezug auf Genauigkeit, gemessen in $mDice_{test}$ der Testdaten

The analysis ranks the dataset properties with regards to their importance in terms of a standardized effect based on a t-statistic as in Figure 5-8. Moreover, a significance threshold (orange) is calculated using a confidence level of 95 % with five degrees of freedom derived by the Lenth's method.

Figure 5-9 contains the analysis of the standardized effect of the dataset properties with regards to mean overfitting mOP_{TT}. This analysis yields a different factor importance than the one ranking factors with regards to accuracy. Only two factors have a significant standardized effect on overfitting. Compared to above, dataset split, and image similarity have switched ranks. The factors dataset size and dataset split have exceeded the significance threshold (orange).

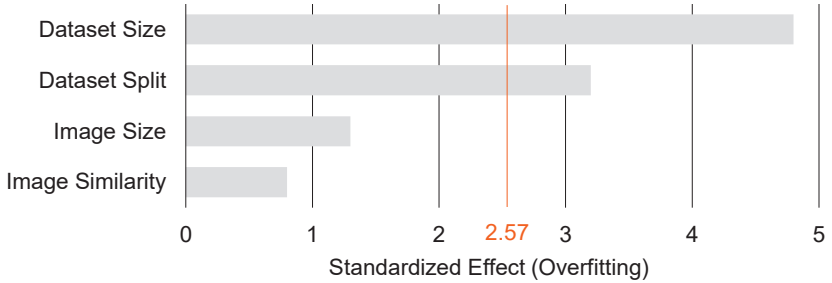


Figure 5-9: Standardized effect of dataset properties regarding mean overfitting in terms of mOP_{TT} of the test dataset
Standardisierter Effekt der Datensatzeigenschaften mit Bezug auf gemittelte Überanpassung, gemessen in mOP_{TT}, der Testdaten

Model Hyperparameters
Modell-Hyperparameter

The investigated factors are the NNs dropout rate, the network size, the learning rate, the activation function, as well as the momentum. A description of these model hyperparameters may be found in the Subsection 5.2.1, Model Hyperparameters. Additional information on the training process of the NN and effects of the hyperparameters are given in Subsection 2.3.2, Neural Network Training.

According to the screening experiments there is a significant standardized effect for all investigated model hyperparameters with regards to the mean accuracy of the unseen test dataset. The analysis ranks the model hyperparameters with regards to their importance in terms of a standardized effect based on a t-statistic as in Figure 5-10. Moreover, a significance threshold (orange) is calculated using a confidence level of 95 % with ten degrees of freedom derived by the Lenth's method. A description of the Lenth's method and specifically the algorithm may be found in the Annex A.13, Identification of the statistically significant effects in factorial experiment based on Lenth's Analysis.

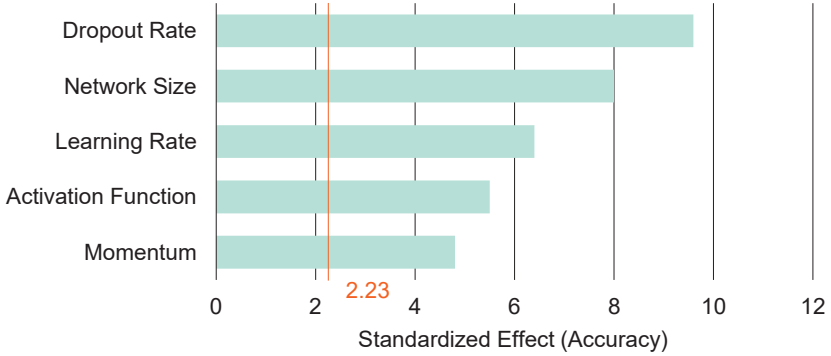


Figure 5-10: Standardized effect of hyperparameters with regards to accuracy in terms of $Dice_{test}$ of the test dataset
Standardisierter Effekt der Hyperparameter mit Bezug auf die Genauigkeit, gemessen in $mDice_{test}$, der Testdaten

Figure 5-11 contains the analysis of the standardized effect of hyperparameters with regards to mean overfitting, mOP_{TT}. This analysis of the overfitting yields a different factor importance than the one ranking factors with regards to accuracy. Only two factors have a significant standardized effect on overfitting. The factors dropout rate and and learning rate have exceeded the significance threshold (orange). Whereas the other factors network size, activation function and momentum are well below the threshold.

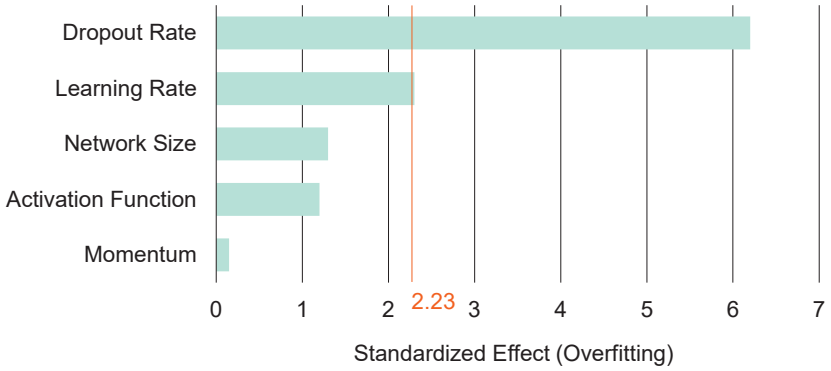


Figure 5-11: Standardized effect of dataset properties regarding overfitting in terms of mOP_{TT}, of the test dataset
Standardisierter Effekt der Datensatzeigenschaften mit Bezug auf Überanapssung, gemessen in mOP_{TT}, der Testdaten

5.3.3 Effect Size Analysis

Analyse der Effektstärke

A significant standardized effect calculated with Lenth's method gives confidence, that the observed difference between two groups is not due to chance. Therefore, it is a good metric to identify whether a factor is relevant.

Contrary to what the name suggests, the effect size of the factors is not known through Lenth's method. However, there is no information in small p-values about the effect size, to understand the magnitude of change that one variable imposes in the target variables [SULL12]. To evaluate the effect size of each relevant factor the Pearson's correlation coefficient, *r*, is used. Pearson's correlation measures the degree of linear relationship between to variables, in this case a factor and a target variable, like $mDice_{test}$. Figure 5-12 shows effect size over significance of the dataset properties and hyperparameters with regards to the target variable of accuracy, $mDice_{test}$.

Figure 5-13 shows effect size over significance of dataset properties and hyperparameters with regards to the target variable of overfitting, OP_{TT}. Factors that were identified as having an influence on the target variables due to chance have no black edge in the plot. The information of direction of the effect of factors on the target variables is contained within the figure. For example, dropout rate seems to have a negative correlation with accuracy. This means lower dropout values tend to cause higher accuracy, which is favorable. Since the analysis is based on a sparse DOE, a more detailed analysis of effect size and direction of factors with regards to the target variables is saved for the next Section 5.4, Full Factorial Analysis.

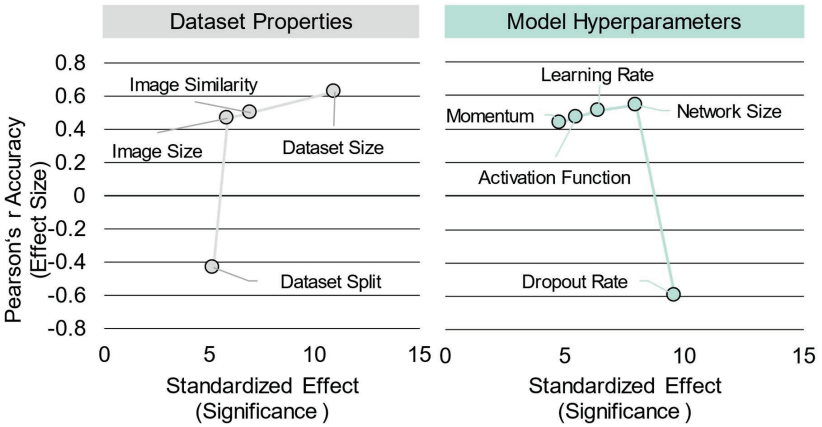


Figure 5-12: Pearson's *r* over Standardized effect of accuracy for Dataset Properties (left column) and Model Hyperparameters (right column)
Pearson's r über standardisiertem Effekt der Genauigkeit für die Datensatzeigenschaften (linke Spalte) und die Modell-Hyperparameter (rechte Spalte)

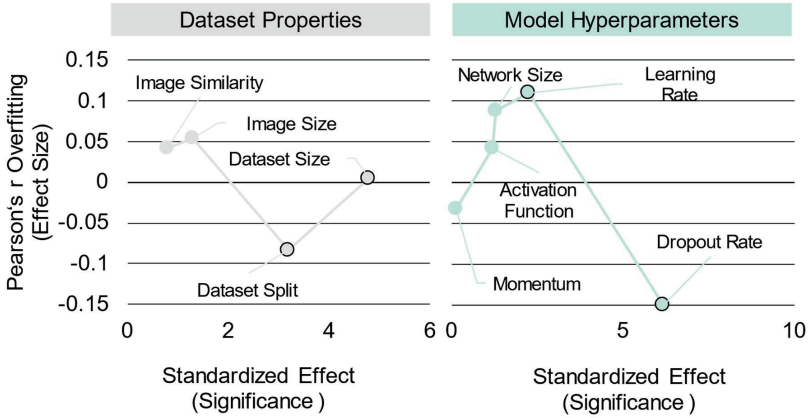


Figure 5-13: Pearson's *r* over Standardized effect of overfitting for Dataset Properties (left column) and Model Hyperparameters (right column)
Pearson's r über standardisiertem Effekt der Überanpassung für die Datensatzeigenschaften (linke Spalte) und die Modell-Hyperparameter (rechte Spalte)

The following Subsection 5.3.4 Discussion of findings contains the conclusions drawn from the successive significance and effect size analysis. That is the decision which factors will be considered in the dense DOE.

5.3.4 Discussion of findings

Diskussion der Ergebnisse

The analysis of dataset properties and model hyperparameters yields valuable results with respect to the relevance and effect size of factors for predicting the target variables accuracy and overfitting. In the first step in Subsection 5.3.2 a significance analysis was conducted to observe which factors are relevant with respect to accuracy and overfitting. The analysis shows that all considered factors are relevant with regards to accuracy. It also shows that dataset split, learning rate and dropout rate are also relevant with regards to overfitting. Dataset size has a high significance, but the effect size is negligible in comparison with dataset split. In general, the effect size of the factors with respect to accuracy are clearly higher than with respect to overfitting. To bring this into perspective, an ANOVA was performed for the accuracy in terms of $mDice_{test}$ and the overfitting in terms of mOP_{TT}.

The ANOVAs yield a linear regression. A metric to quantify the quality of a regression is the coefficient of determination, which describes the proportion in variance in one variable explained by another variable. The adjusted coefficient of determination penalizes the metric in terms of the number of parameters required for the fit. The adjusted coefficient of determination for predicting accuracy based on the factors is $R^2 = 0.88$.

The adjusted coefficient of determination for predicting overfitting based on the factors is $R^2 = 0.42$. The results show that overfitting is harder to predict from dataset properties and model hyperparameters than accuracy. Hence, for deciding on whether to drop factors from further analysis and model creation or not, an according weighting of the factors is appropriate.

$$\textit{Regression-weighted Effect} = r_{Dice_{test}} R^2_{Dice_{test}} + r_{OP_{TT}} R^2_{OP_{TT}} \quad (22)$$

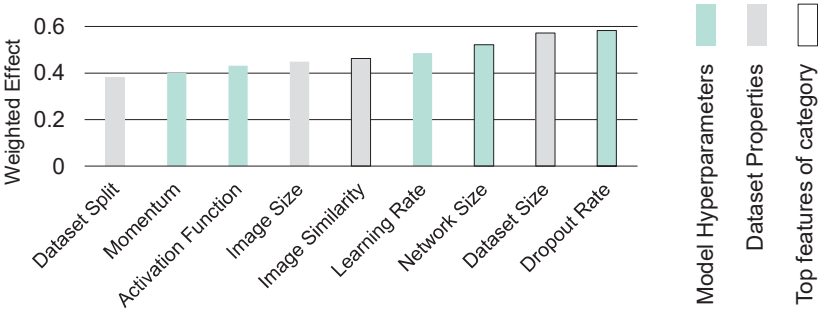


Figure 5-14: Regression-weighted effect for all investigated factors
Regressionsgewichteter Effekt für alle untersuchten Faktoren

The above Equation (22) defines the scoring function applied for each factor of dataset properties and model hyperparameters. Each factor's individual Pearson's correlation, *r*, for $mDice_{test}$ and mOP_{TT} is calculated and, in each case multiplied with the adjusted coeffcient of determination R^2 from the respective ANOVA regression, either for $mDice_{test}$ or mOP_{TT}.

Weighting the individual factors according to their effect size and importance with respecting the accuracy or overfitting regression score yields the ranking shown in Figure 5-14. The two highest ranked factors from each group (see Figure 5-14, framed bars) are chosen for further analysis. The analysis above answers the second Research Question: *"What are the dataset and model properties with the highest impact on model performance for tool wear segmentation?"*. Based on the screening analysis performed in this section and the successive discussion, the answer to this question is:

Dataset properties:

- Dataset size
- Image similarity

Model hyperparameters:

- Dropout rate
- Network size

5.4 Full Factorial Analysis

Vollfaktorielle Analyse

This section contains a description of the necessary steps to perform a Full Factor Analysis of the dataset and model properties with the highest impact on the model evaluation metrics. It consists of Subsection 5.4.1, Preparations describing the DOE, sample size estimation and the datasets. Followed by Subsection 5.4.2, Exploratory Analysis and Subsection, and Subsection 5.4.3, Outlier Analysis. In Subsection 5.4.4, Interaction Analysis, and Subsection 5.4.5, Main Effect Analysis, the factors interdependence and effects on model evaluation metrics are investigated. The section is closed with Subsection 5.4.6, Discussion of Findings.

5.4.1 Preparations

Vorbereitungen

Design of Experiments

Statistische Versuchsplanung

The Table 5-2 shows all factors and their respective levels for the full factorial analysis. In a full factorial design this setup yields 81 test points without repetitions. This seems feasable as the Generalized Subset Design (GSD) from the screenig analyis had a count of 255 in total. The three-level design, also denoted 3^k, allows modelling possible quadratic terms between each factor and the response or target variables.

In summary, the full factorial design is suitable for a model creation with four factors and three levels. The next paragraph elaborates on the required repetitions. The other parameters omitted from the full factorial DOE are fixed and set to the following values: Image size is fixed at (512 x 512 x 3), the training epochs are fixed at 300, the learning rate is set to 0.001 with momentum at 0.9, the train / val / test dataset split is set to (0.8 / 0.1 / 0.1). For the activation functions within the network are the ELUs are chosen since they produce smoother activation values than ReLU and are less prone to becoming inactive during training.

Table 5-2: Full Factorial DOE Factors and Levels

Faktoren und Faktorstufen des Versuchsplans zur Vollfaktoriellen Analyse

	Dataset Properties		Model Hyperparameters	
Factors / Levels	**Dataset Size**	**Data Similarity**	**Network Size**	**Dropout Rate**
0	50	0.67	120k	0.2
1	100	0.77	650k	0.4
2	400	0.87	2000k	0.6

Sample Size Estimation

Abschätzung der Stichprobengröße

To prepare the full factorial analysis a determination of the minimum sample size for each of the 81 test points is required. Findings from the previous screening analysis conducted with five experiments for each test point indicate that the required sample size can be reduced in the full factorial analysis.

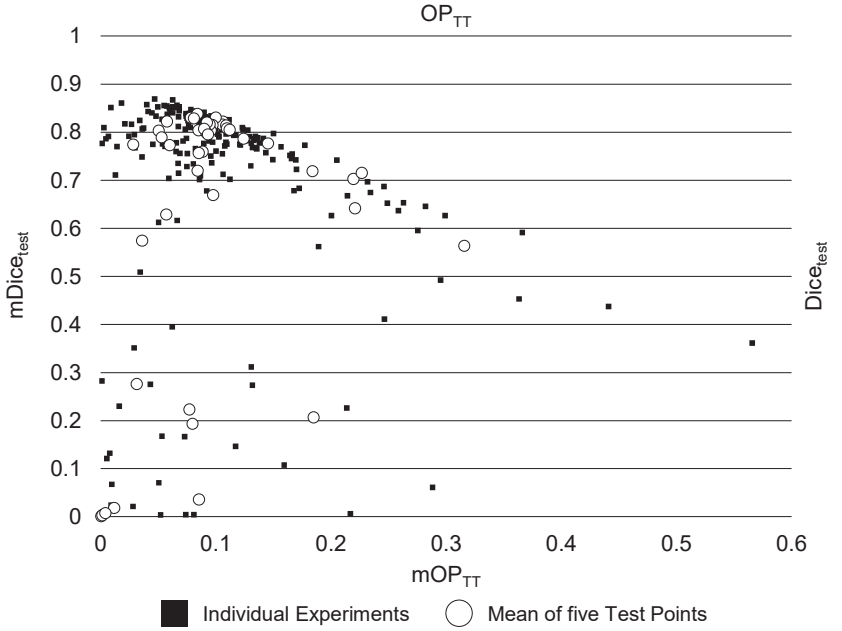


Figure 5-15: ***Dice$_{test}$* over *OP$_{TT}$* and *mDice$_{test}$* over *mOP$_{TT}$* of five identical experiments each**

Dice$_{test}$ über OP$_{TT}$ und mDice$_{test}$ über mOP$_{TT}$ von jeweils fünf identischen Experimenten

In Figure 5-15 it is possible to see a cluster of successful network trainings around *Dice$_{test}$* of 0.8 and *OP$_{TT}$* of 0.1. It appears that successful network trainings tend to scatter less with repeated identical experiments than unsuccessful trainings with high overfitting or low accuracy.

The representation of data in Figure 5-16 supports this thesis. When defining a sufficiently successful neural network training by a minimum mean accuracy *mDice$_{test}$* $\geq$ 0.7, compare grey area in the figure, the mean standard deviation is at $\mu(\sigma Dice_{test}) = 0.0366$, compare black cross in the figure.

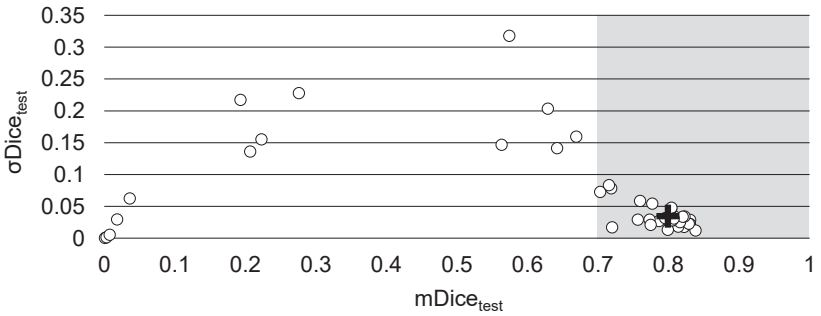


Figure 5-16: Standard deviation of *Dice$_{test}$* over *mDice$_{test}$* of screening analysis data
Standardabweichung von Dice$_{test}$ über mDice$_{test}$ für die Daten der Einflussgrößenanalyse

The sample size estimation for the full factorial analysis is set up according to Equation (23). The expected standard deviation from the screening experiment is $\sigma = 0.0366$, as stated above. The number of repetitions is five, denoted *N*. The acceptable percentual error threshold, *E*, remaining at 0.05, that is 5 % Dice. The z_N value of 1.96 results from the standard normal distribution for the targeted confidence level of 95 %.

$$n \geq \frac{N\, z^2\, \sigma^2}{z_N^2\, \sigma^2 + (N-1)\, E^2} = 1.70 \tag{23}$$

According to the formula, the number of experiments in each test point should be greater than 1.70. For a more conservative approach and to be more confident in possible outlier detection the sample size of three is chosen for each test point in the full factorial DOE. With the repetitions set to three and a reduction factor of zero the number of experiments in the full factorial DOE arrives at 243.

Dataset
Datensatz

The datasets are created from the instructions specified by the DOE. The data consist of images and their respective label mask. Below, in Figure 5-17, there are examples of original images row-wise grouped by image similarity. A human perceiving the top row as most homogeneous, the bottom row as most heterogeneous and the middle row as something in between, agrees with the image similarity metric SSIM applied in this thesis.

Due to the necessary repetitions derived in the above paragraph a random initial seed and a randomised dataset split is utilized in each experimental run for each subset of data, i.e., training, validation, and test dataset. This ensures that there is no favourable or unfavourable dataset split that effects the model evaluation metrics and dilutes the information of the observed results.

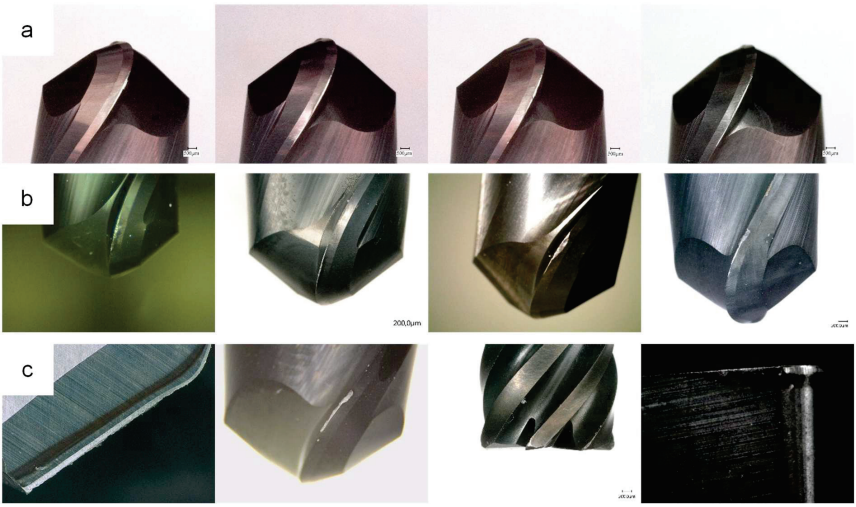


Figure 5-17: Examples of image data from three dataset with a mean inner similarity metric of a) SSIM = 0.87, b) SSIM = 0.77 and c) SSIM = 0.67
Beispielbilddaten von drei Datensätzen mit einer mittleren, inneren Ähnlichkeit von a) SSIM = 0.87, b) SSIM = 0.77 und c) SSIM = 0.67

5.4.2 Exploratory Analysis

Explorative Analyse

To get familiar with the data produced in the full factorial design, a visual data analysis was conducted. It aimed at identifying general trends of factors with respect to the model evaluation metrics accuracy and overfitting.

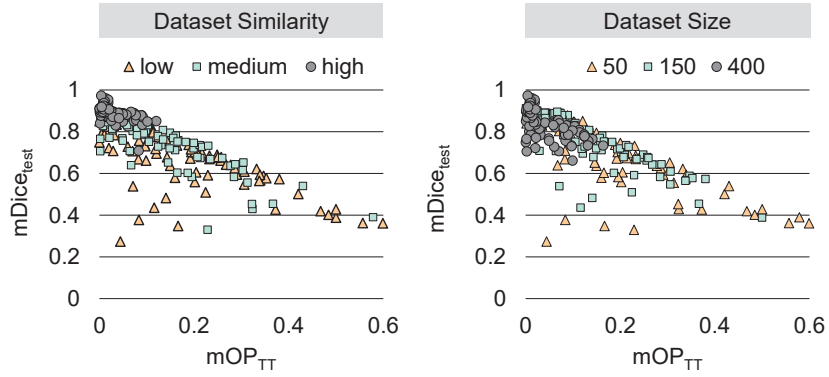


Figure 5-18: $Dice_{test}$ over OP_{TT} with color indication of dataset properties
$Dice_{test}$ *über* OP_{TT} *mit farblicher Indikation der Datensatzeigenschaften*

Figure 5-18 indicates where dataset property levels are grouped within the $Dice_{test}$ over OP_{TT} scatter plot. Color and symbol are assigned to each experiment's marker based on the factor level. The attribution of the color and symbol to the factor levels is given in the legend above each plot.

Some general trends are observable from this visualization: A higher dataset similarity tends to lead to better models in terms of accuracy and overfitting. The sample is observable for dataset size, where a larger dataset tends to lead to higher accuracy and less overfitting.

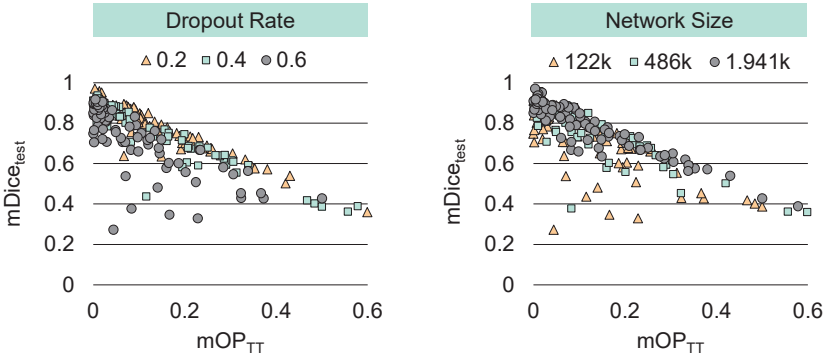


Figure 5-19: *$Dice_{test}$* over *OP_{TT}* with color indication of model hyperparameters
$Dice_{test}$ *über* OP_{TT} *mit farblicher Indikation der Modellhyperparameter*

Figure 5-19 indicates where model hyperparameter levels are grouped within the $Dice_{test}$ over OP_{TT} scatter plot. Color and symbol are assigned to each experiment's marker based on the factor level. The attribution of the color and symbol to the factor levels is given in the legend above each plot. A higher dropout rate tends to lead to better accuracy and especially less overfitting. Albeit some models with high dropout reach accuracies as low as $Dice_{test}$ = 0.3. An increase in network size tends to lead to higher accuracies.

Table 5-3: Best model in each dataset similarity group and its dataset properties and model hyperparameters
Das beste Modell in jeder Datensatzähnlichkeits-Gruppe und dessen Datensatzeigenschaften und Modell Hyperparameter

$Dice_{test}$	OP_{TT}	Dataset Similarity	Dataset Size	Dropout Rate	Network Size
0.956	0.011	High	400	0.2	1.941k
0.859	0.010	Medium	400	0.6	1.941k
0.810	0.110	Low	400	0.2	1.941k

The individual best models for each dataset similarity group created in the full factorial design, see Table 5-3, mostly support the tendencies found in the exploratory analysis above. These are: a high dataset size tends to produce the best models across the three similarity classes. Further, a low dropout rate and a high network size tend to produce the best models across the three similarity classes.

5.4.3 Outlier Analysis

Ausreißeranalyse

The outlier analysis aims at identifying data points that differ significantly from other observations. For reasons of rigourousity the outliers are split into two categories: technical outlier means system malfunction where the network trainings failed catastrophically and discretionary outliers where the training did not produce networks with sufficient model evaluation metrics.

Technical Outliers

Technische Ausreißer

Occasionally, a neural network training may fail catastrophically, e.g., $Dice_{test} \leq 0.01$, usually because the training process gets stuck in a local minimum of the loss landscape that cannot be escaped with the hyperparameter settings given. A detection of a failed training is possible by looking at the training curves of the neural network. Three experiments were made for each test point in the full factorial DOE, see Subsection 5.4.1 Preparations. For this reason it is also possible to calculate a standard deviation and identify outliers in Figure 5-20, where standard deviation of $Dice_{test}$, named $\sigma Dice_{test}$, is plotted as a function of $mDice_{test}$, which is the mean of $Dice_{test}$, in each test point.

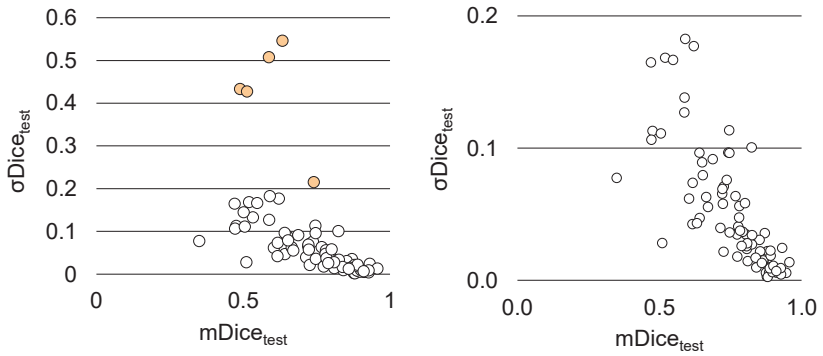


Figure 5-20: Standard deviation of $Dice_{test}$ over $mDice_{test}$ of full factorial data with outliers (orange) in test points (left). Outlier removal (right)
Standardabweichung von $Dice_{test}$ über *$mDice_{test}$ für die Daten der vollfaktoriellen Analyse mit Ausreißern (orange) in den Testpunkten (links). Entfernung der Ausreißer (rechts)*

The neural network trainings detected as outliers were repeated to assess the permanence of the outliers. By repeating and replacing the original experiment a reduced $\sigma Dice_{test}$ of the affected test points could be reached, compare left and right side of the figure.

The repetition was performed in five different test points, see left side in Figure 5-20. Each of these test points had one one experiment where the NN training resulted in a relatively low Dice coefficient, leading to the high $\sigma Dice_{test}$. In comparison to the screening analysis, no failure of technically successful NN training with $Dice_{test} \leq 0.1$ occurs, possibly due to the tendency of favorable hyperparameters in the dense DOE.

Discretionary Outliers
Diskretionäre Ausreißer

As in Subsection 5.4.1 Preparations a successful neural network training is defined as $mDice_{test} \geq 0.7$. Applying this filter to the data yields Figure 5-21 with $mDice_{test}$ over mOP_{TT}. From the original test points two thirds remain in the analysis, therefore 162 experiments, i.e., 54 test points, remain for the model creation from the full factorial DOE database.

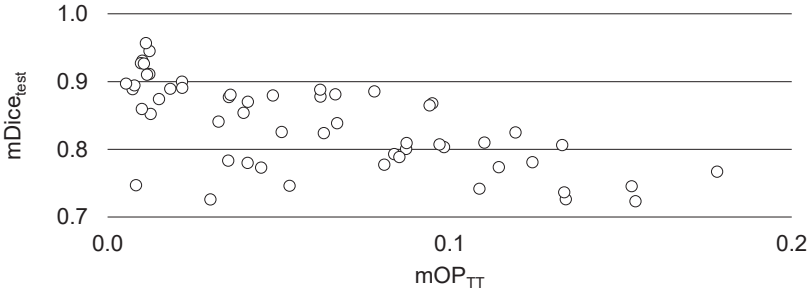


Figure 5-21: $mDice_{test}$ over mOP_{TT} of three identical experiments each
$mDice_{test}$ über mOP_{TT} von jeweils drei identischen Experimenten

5.4.4 Interaction Analysis
Wechselwirkungsanalyse

The exploratory and outlier analysis in the prior subsection gives an overview of general trends in the data from the full factorial analysis. This is a good start into understanding the data, but it omits interaction effects between the factors. Interaction occurs when the effect of one variable depends on the value of another variable. Possible interactions can be illustrated with the help of an interaction diagram where parallel lines indicate no interaction. The greater the difference in the slope between the lines, the greater the degree of interaction. For the planned target value optimization of hyperparameters based on dataset properties it is beneficial to find strong interactions between the variables of both groups.

If the main effects are of much greater magnitude, the overall optimization of the segmentation model is possible, but a nuanced optimization of hyperparameters based on dataset properties could be difficult. In the following chapter both will be attempted and reevaluated. Figure 5-22 serves an example for the negligible interaction effects: There is interaction between dataset size and dropout rate with regards to accuracy. A higher accuracy can be expected for larger datasets with small dropout rate. But for high dropout rates the effect of dataset size is inverted. In comparison there seems to be no interaction between dataset similarity and dropout rate, see right hand side of Figure 5-22. From the magnitude of effect, it is possible to estimate if an interaction effect could be meaningful, however it is not possible to tell whether the interaction is statistically significant. To determine the statistical significance of the interaction effects an ANOVA is performed. Unfortunately, the p-values and F-values of a general ANOVA yield that there are no significant two-way or three-way interactions between the factors data similarity, dataset size, dropout rate and network size regarding accuracy and overfitting. The complete F-statistic is given in A.14, ANOVA regards to accuracy.

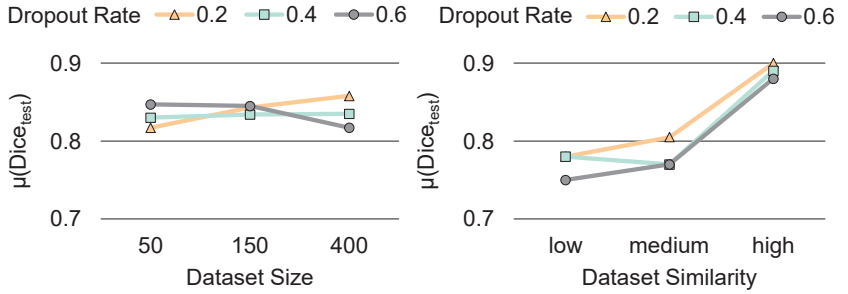


Figure 5-22: Interaction plot of dataset size with dropout rate (left) and dataset similarity with dropout rate (right) with regards to *Dice$_{test}$*
Wechselwirkungsdiagramm von der Datensatzgröße mit Dropout Rate (links) und der Datensatzähnlichkeit mit der Drop. Rate (rechts) in Bezug auf Dice$_{test}$

Nevertheless, in regression analysis interaction effects can still play a role since the isolated consideration of interaction effects may miss changes in prediction power of a full and reduced regression model.

5.4.5 Main Effect Analysis
Haupteffektanalyse

Since there is no significant interaction of factors, the main effects are not confounded and produce meaningful information [BRAM06]. Main effects are analyzed by main effect plots and the full ANOVA regression model to determine statistical significance.

In the main effect plots, dataset properties are indicated with black lines and model hyperparameters are indicated with turquoise lines, as in the rest of this document. Dataset similarity has the largest main effect regarding accuracy as compared to the

other factors, see Figure 5-23. It also has a non-linear relationship, more specifically a quadratic characterstic. This means the higher a similarity, the easier it is to reach a high accuracy. In general, there are weak linear relationships between the other factors and target variable accuracy. According to the isolated F-values there was a significant main effect for dataset similarity, $F(2, 10) = 133.38$, $p < .001$. There was also a significant main effect for dataset size, $F(2, 10) = 20.79$, $p < .001$. A more detailed analysis of the ANOVA is shown in Annex A.14, ANOVA regards to accuracy. This annex contains the complete F-Statistic.

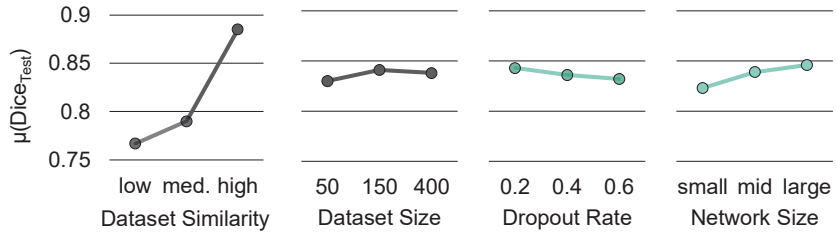


Figure 5-23: Main effect plots of the factors with regards to $Dice_{test}$
Haupteffektdiagramme der Faktoren mit Bezug auf $Dice_{test}$

In the main effect plot with overfitting, dataset similarity has the largest main effect as compared to the other factors, see Figure 5-24. In general, there are two non-linear, negative quadratic relationships between the dataset properties and target variable overfitting.

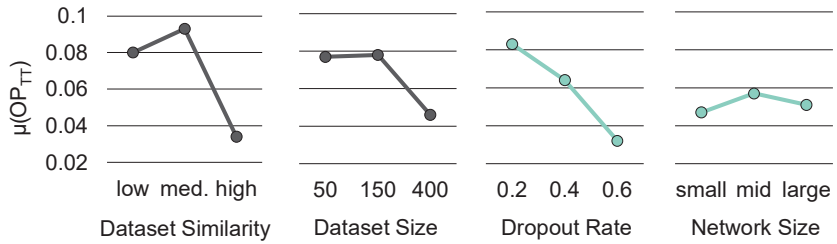


Figure 5-24: Main effect plots of the factors with regards to OP_{TT}
Haupteffektdiagramme der Faktoren mit Bezug auf OP_{TT}

Dropout has an almost linear relationship. Network size has a rather non-linear relationship with overfitting. Apart from network size, high values in the other factors promote low overfitting. According to the isolated F-values there was a significant main effect for dataset similarity, $F(2, 10) = 20.20$, $p < .001$. There was also a significant main effect for dataset size, $F(2, 10) = 10.76$, $p < .003$. Finally, there was a significant main effect for dropout value, $F(2, 10) = 9.79$, $p < .004$. The complete F-statistic is given in Annex A.15, ANOVA with regards to overfitting.

5.4.6 Discussion of Findings

Diskussion der Ergebnisse

The technical outlier analysis helped identify technically failed NN trainings without inspecting the training curve of each individual experiment. Through the analysis failed experiments were repeated and replaced by successful repetitions. Based on prior knowledge of acceptable DL segmentation models, a discretionary outlier threshold was set as priorily done in the screening analysis. Through this treatment the NN training settings that lead to unfavorable results with high variance are removed, see Figure 5-25, and thus uncertainty in the subsequent modelling process can be reduced.

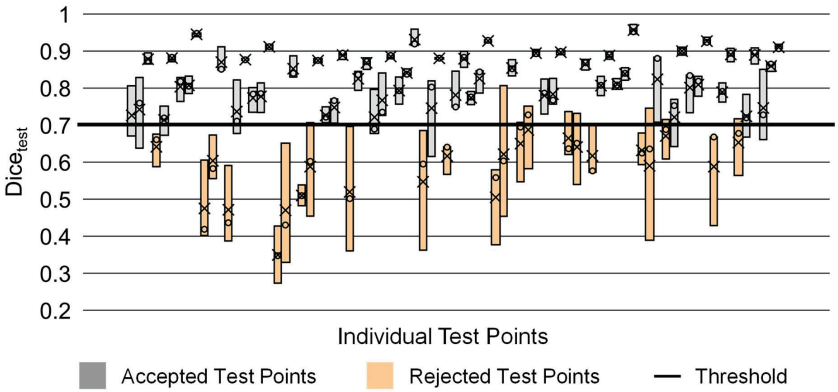


Figure 5-25: Individual test points boxplots with color-indication of mean values of accuracy below the discretionary threshold (orange)
Individuelle Boxplots der Testpunkte mit farblicher indikation der Mittelwerten der Genauigkeit, die sich unter der gewählten Grenze befinden (orange)

During interaction analysis, it is important to notice whether the main effects are confounded with interaction effects. When the targeted input variables and targeted output variables for target value optimization have little interaction, and the main effects of the variables are dominant, the model's behavior is largely determined by the main effects of each variable. This means that the target value optimization based on this data could become difficult, and it might be necessary to explore other variables to achieve the desired results.

The main effect analysis, on the other hand, shows that dataset similarity has the largest effect on accuracy and overfitting. It is important to note that overfitting occurs when a model becomes too complex and fits the training data too closely, resulting in poor performance on new data. In isolated F-tests, network size did not have a significant interaction or main effect with respect to the model evaluation metrics. This indicates that the network size may not be as important as other factors when it comes to optimizing the model's performance.

Based on the linear and non-linear nature of the main effects, it can be concluded that linear and quadratic terms must be considered in regression modelling. This is because the relationship between the factors and the target variables may not be linear, and it may be necessary to account for non-linear effects to accurately model the relationship. In hindsight, the non-linear main effects show that the full factorial 3^k DOE was a necessary choice to capture the curvature in the relationship between the factors and the target variables.

In general, datasets with high similarity and size tend to produce better models in terms of accuracy and overfitting. This is because larger datasets provide more data for the model to learn from, and similar datasets are more likely to have similar patterns that can be easily learned by the model. However, high dropout values can also lead to favorable overfitting metrics, but this comes at a cost of accuracy. Furthermore, large networks tend to produce more accurate models. Due to the flat main effect observable from the data, the influence of network size on overfitting is not clear with the usecase provided and may require further investigation.

5.5 Decision Model

Entscheidungsmodell

Based on the analysis above it is known how to set up a model with high accuracy and low overfitting. Still the third research question remains open:

> RQ3: How can a systematic choice of hyperparameters with regards to dataset properties be employed to improve model performance for tool wear segmentation?

Based on the findings of Subsection 5.5.1, Modelling Approach, a regression is conducted for modelling of the data. Subsection 5.5.2, Methodology, explains the required steps to answer research question 3. Subsection 5.5.3, Regression Models, contains the information on polynomial regression for accuracy and overfitting. Subsection 5.5.4, Target Value Optimization, explains the decision model and gives examples of its application.

5.5.1 Modelling Approach

Modellierungs-Ansatz

To find a suitable modelling approach for making a regression of accuracy and overfitting based on the model hyperparameters and dataset properties, a screening analysis of modelling approaches was conducted. Linear regression and a suite of ML algorithms were used to create regression models for accuracy in terms of $Dice_{test}$ and overfitting in terms of OP_{TT} based on the factors described above, see Figure 5-26 and Figure 5-27.

This procedure serves the purpose to identify a suitable regression modelling approach that shows good performance for both tasks, modelling of accuracy and overfitting, with a focus is on the accuracy as in prior investigations.

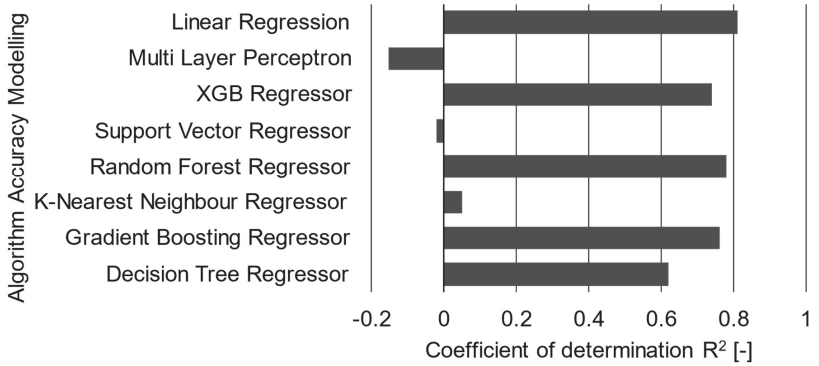


Figure 5-26: Results of an algorithm screening for modelling accuracy based on the data from the fullfactorial DOE
Ergebnisse eines Algorithmus-Screenings zur Modellierung der Genauigkeit auf Basis der Daten aus dem vollfaktoriellen DOE

The standard implementation of the model architectures and parameters from the Scikit-learn library were used for this investigation [PEDR11]. Additionally, a four-layered MLP was designed using the Tensorflow library [TENS22]. For each target variable, the $Dice_{test}$ coefficient, and the OP_{TT} a regression model was fitted with each of the algorithms. The coeffcient of determination R^2 was calculated for each algorithm to determine the relative performance between the approaches. Figure 5-26 shows the results for the regression of accuracy.

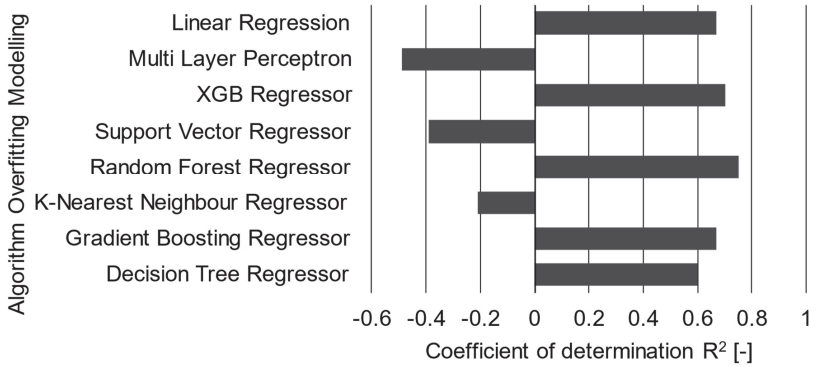


Figure 5-27: Results of an algorithm screening for modelling overfitting based on the data from the fullfactorial DOE
Ergebnisse eines Algorithmus-Screenings zur Modellierung der Überanapassung auf Basis der Daten aus dem vollfaktoriellen DOE

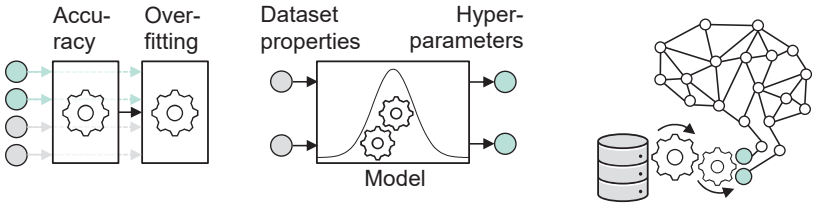
Linear regression seems to be most suitable for the task. Similar performance was achieved by the algorithms XBG Regressor (XGBR), Random Forest Regressor (RFR) and Gradient Boosting Regressor (GBR). Other models such as K-Nearest Neighbour Regressor (KNNR), Support Vector Regressor (SVR) and Multi-Layer Perceptron (MLP) perform very poorly and in two cases worse than just using the mean leading to negative R^2 values.

For fitting the overfitting metric OP_{TT} based on the factors, we get a similar result. Again, the MLP, SVR and KNNR fail at modelling the data. The other regression algorithms range from R^2 of 0.6 to 0.75, see Figure 5-27. Linear regression is the superior algorithm for modelling accuracy and ranks in the third place for modelling overfitting. Since modelling the accuracy is the most important task and for the sake of the decision model's transparency the linear regression approach is chosen for the furher procedure. In general, the simpler model should be chosen over more complex models for computationally efficiency, data requirements and interpretability.

5.5.2 Methodology for Decision Model Creation

Methode zur Erstellung eines Entscheidungsmodells

Based on the former investigation the linear regression approach is chosen for the decision model. The former investigation shows that the modelling of overfitting gives a worse fit compared to accuracy. To increase the prediction performance for modelling the overfitting a chained multi-output regression is conducted to model accuracy first and overfitting subsequently. Chained multi-output regression refers to using single-output regression models in a sequence of models. The first model uses the factors to predict a target variable. The factors and the target variable are used in a second step to predict a second target variable.



Step 1 Multioutput Regression	Step 2 Decision Model	Step 3 Optimized DNN Training
Regression modelling of accuracy and overfitting based on the previously identified variables	Model for knowledge-based selection of NN hyperparameters based on dataset properties	Optimized DNN training for high performance on the cutting tool wear image dataset of interest

Figure 5-28: Schematic representation of the approach in Section 5.5
Schematische Darstellung der Vorgehensweise in Abschnitt 5.5

The chained multi-output regression is used since model accuracy is more clearly dependant on the factors and since there is a linear relationship with $R^2 = 0.44$ between accuracy and overfitting this two-step approach is chosen, see Figure 5-29.

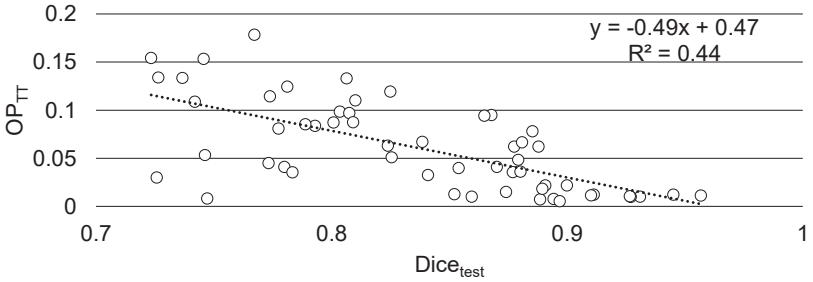


Figure 5-29: Scatter plot OP_{TT} over $Dice_{test}$ with linear regression
Streudiagramm OP_{TT} über $Dice_{test}$ mit linearer Regression

In a second step, after the multi-output regression model is complete, a minimization method is applied to the accuracy regression model. This is done to find optimal values of dropout rate and network size for a set of images with the given properties dataset similarity and dataset size. Afterwards the expected overfitting may be calculated from the expected accuracy, the expected dropout rate and network size and the given dataset similarity and dataset size. In the third and final step, the minimization method is used for target value optimization for the dataset used in Chapter 6, Validation of AI-based Automated Tool Wear Measurement. Figure 5-28 shows the methodology in a conceptual manner, where the color grey is used for dataset properties and the color turquois indicates model hyperparameters.

5.5.3 Regression Models
Regressionsmodelle

For regression modelling a feature selections approach was chosen that adds and removes polynomal features incrementally and evaluates the model using the adjusted coefficient of determination at each step. In the following paragraphs the result of the regressions is presented.

Accuracy Model
Genauigkeitsmodell

Based on the prior analysis linear and quadratic main effects as well as two-way interactions are considered for the accuracy regression model. For reasons of brevity, the following notation is chosen in this chapter for the display of target and predictor variables in the formulas: $f(x)$ is the target variable $Dice_{test}$, i.e., accuracy. The factors or independent variables are dataset similarity x_1, dataset size x_2, dropout rate x_3 and network size x_4. The data is fitted to the following equation. For reasons of brevity the

repeated patterns with the factors are replaced with three dots in the respective equation:

$$f(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_5 x_1^2 + \cdots + \beta_9 x_1 x_2 + \cdots \quad (24)$$

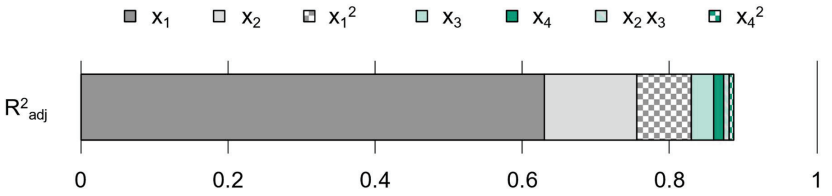


Figure 5-30: Adjusted coefficient of determination for the accuracy regression model with each terms' share of explained variance
Bereinigtes Bestimmtheitsmaß für das Genauigkeits-Regressionsmodell mit dem Anteil jedes Terms an der erklärten Varianz

In Equation (24) the zeroth term is the intercept, the first to fourth term are the linear main effects, the fifths to eigth term are the quadratic main effect and the ninth term onwards are the interaction effects. The model has an adjusted coefficient of determination of R^2_{adj} = 0.88. This means 88 % of the variance in the data is explained by the model, see Figure 5-30. The values of the regression model parameters are shown in Table 5-4.

Table 5-4: Parameters and values for accuracy regression model
Parameter und Werte des Genauigkeits-Regressionsmodells

Specification	**Values**							
Parameters	β_0	β_1	β_2	β_3	β_4	β_5	β_8	β_{12}
Independant Variables		x_1	x_2	x_3	x_4	x_1^2	x_4^2	x_2x_3
Values	2.591	-5.53	0.0934	-0.0163	0.1305	4.047	-0.0969	-0.0902

Overfitting Model
Überanpassungsmodell

Based on the prior analysis linear and quadratic main effects as well as two-way interactions are considered for the overfitting regression model. To prove the hypothesis from Subsection 5.5.2 Methodology about the regression quality increase by including $Dice_{test}$ as a predictor in the regression of OP_{TT}, two regression models are fitted. The first without and second below with $Dice_{test}$ as an independent variable. For reasons of brevity, the following notation is chosen: $f(x)$ is the target variable OP_{TT}, i.e., overfitting. The factors or independent variables are dataset similarity x_1, dataset size x_2,

dropout rate x_3 and network size x_4. Fitting the data to the model results in an R^2_{adj} = 0.75.

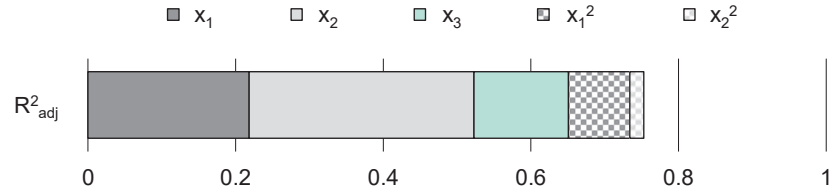


Figure 5-31: Adjusted coefficient of determination for the overfitting regression model with each terms' share of explained variance
Bereinigtes Bestimmtheitsmaß für das Überanpassungs-Regressionsmodell mit dem Anteil jedes Terms an der erklärten Varianz

This means 75 % of the variance in the data is explained by the model, see Figure 5-31. The values of the regression parameters are shown in Table 5-5.

Table 5-5: Parameters and values for overfitting regression model
Parameter und Werte des Überanpassungs-Regressionsmodells

Specification	Values					
Parameters	β_0	β_1	β_2	β_3	β_5	β_6
Independant Variables		x_1	x_2	x_3	x_1^2	x_2^2
Values	-1.231	3.95	0.0584	-0.1003	-2.806	-0.1039

As stated above the prediction of overfitting with knowledge about the $Dice_{test}$, for brevity named x_5, yields a regression with R^2_{adj} = 0.84. The other factors are named with the same pattern as above.

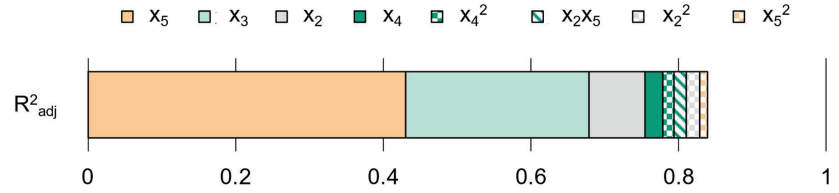


Figure 5-32: Adjusted coefficient of determination for the overfitting regression model with each terms' share of explained variance with prior knowledge about $Dice_{test}$
Bereinigtes Bestimmtheitsmaß für das Überanpassungs-Regressionsmodell mit dem Anteil jedes Terms an der erklärten Varianz mit vorheriger Kenntnis über $Dice_{test}$

Through the multi-output regression approach the predictive power of the overfitting regression model is increased by 9 % measured in adjusted coefficient of determination, see Figure 5-32. See Table 5-6 for the values of the model parameters, including $Dice_{test}$ to predict OP_{TT}.

Table 5-6: Parameters and values for overfitting regression model with $Dice_{test}$
Parameter und Werte des Überanpassungs-Regressionsmodells mit $Dice_{test}$

Specification	Values								
Parameters	β_0	β_2	β_3	β_4	β_5	β_7	β_9	β_{10}	β_{17}
Independant Variables		x_2	x_3	x_4	x_5	x_2^2	x_4^2	x_5^2	x_2x_5
Values	-0.393	-0.272	0.1269	0.1202	2.00	-0.1057	-0.0893	-1.703	0.439

As visible above the interaction terms in both, the accuracy model and the interaction model are small compared to the main effects. Target value optimization may be a challenging task when the input variables and output variables of a model have little interaction, and the main effects are dominant. This is because the model's predictions are dominated by the main effects of each input variable, rather than their interactions.

5.5.4 Target Value Optimization
Zielgrößenoptimierung

In regression an optimization is usually defined as a minimization problem. Since the goal is to maximize the accuracy, the equation is multiplied by minus one for optimization. The optimization method to find the roots of the multivariate function is the modified Powell method [MORÉ80]. Equation (25) is the function subject to minimization.

$$f_{opt}(x) = -(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_1^2 + \beta_8 x_8^2 + \beta_{12} x_2 x_3) \qquad (25)$$

With constant values for dataset similarity and dataset size, there are two variables, dropout rate, x_3, and network size, x_4, to be optimized for the optimization goal. As stated above the optimization goal is a low negative accuracy, i.e., a high accuracy. Due to the three-dimensional nature of this matter, a contourplot is chosen to present the result of the optimization for a specific set of dataset properties. Figure 5-33 shows an example of the contour plot. The figure shows that, according to the regression model, for the chosen dataset properties a dropout rate of 0.25 and a large network size should be chosen. Since the problem has not only three but rather five dimensions when considering the two dataset properties as input to the optimization, a collection of contourplots is shown in Figure 5-34 to visualize the optimization results in five dimensions. The x-ticks and y-ticks of each individual plot for dropout rate and network size are not given. Three and four levels of dataset similarity, respectively dataset size,

are given resulting in 12 individual contour plots of network size over dropout rate, see Figure 5-34.

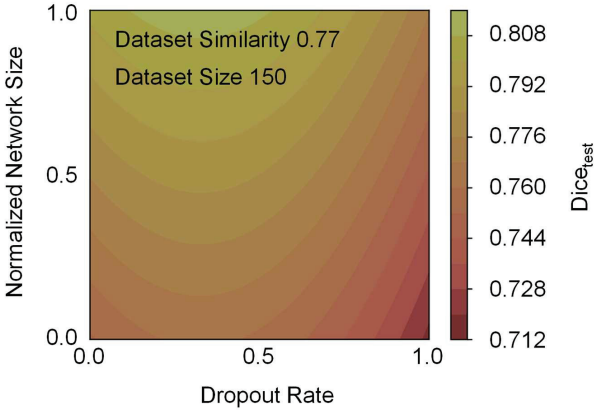


Figure 5-33: Contour plot of normalized network size over dropout rate with accuracy as colorscale for dataset similarity of 0.77 and dataset size of 150
Konturdiagramm der normalisierten Netzwerkgröße über der Dropout-Rate mit Genauigkeit als Skala für Ähnlichkeit von 0.77 und Datensatzgröße von 150

Where network size is the y-dimension and dropout rate are the x-dimension of each plot. Network accuracy is presented as a colorscale, where blue represents high accuracy values, green is medium accuracy values and red indicates low accuracy values.

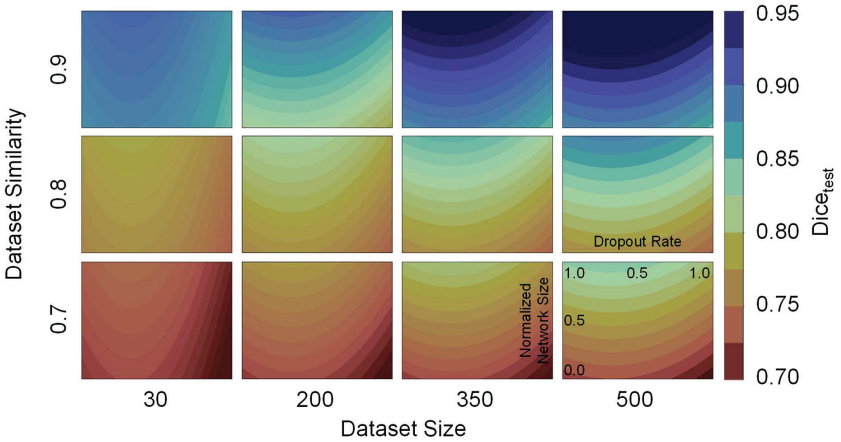


Figure 5-34: Collection of contour plot of normalized network size over dropout rate with accuracy as colorscale for the levels of dataset similarity and size
Konturdiagramm der normalisierten Netzwerkgröße über der Dropout-Rate mit Genauigkeit als Farbskala für die Stufen der Datensatzähnlichkeit und -größe

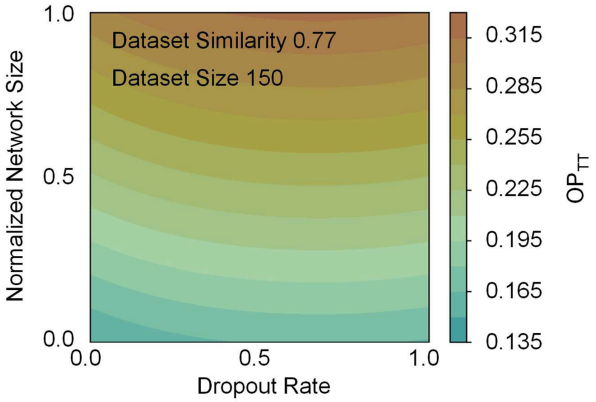


Figure 5-35: Contour plot of normalized network size over dropout rate with overfitting as colorscale for dataset similarity of 0.77 and dataset size of 150
Konturdiagramm der normalisierten Netzwerkgröße über der Dropout-Rate mit Überanpassung als Farbskala für Datensatzähnlichkeit von 0.77 und Datensatzgröße von 150

Since the OP_{TT} depends on the accuracy to a fair amount, compare Subsection 5.5.3, Regression Models, the check for overfitting tendency of the potentially optimized network is conducted in a subsequent step.

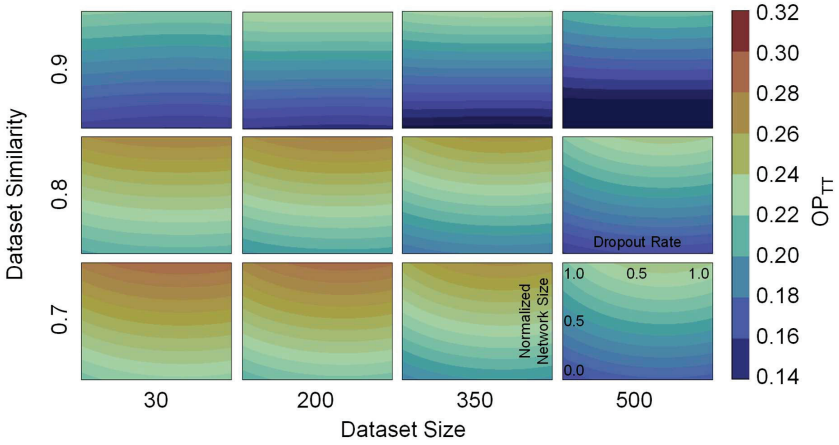


Figure 5-36: Collection of contour plot of normalized network size over dropout rate with overfitting as colorscale for the levels of dataset similarity and size
Konturdiagramm der normalisierten Netzwerkgröße über der Dropout-Rate mit Überanpassung als Farbskala für die Datensatzähnlichkeit und -größe

The Figure 5-35 shows that, according to the overfitting regression model, for the chosen dataset properties a small dropout rate and a small network size should be chosen. Figure 5-36 visualizes the optimization results in five dimensions. The x-ticks and y-ticks of each individual plot for dropout rate and network size are not given. Refer to the prior figure to see where high and low values of these two hyperparameters are in an individual contour plot. Three and four levels of dataset similarity, respectively dataset size, are given resulting in 12 individual contour plots of network size over dropout rate. Where network size is the y-dimension and dropout rate are the x-dimension of each individual plot, see lower right corner in the figure. Network overfitting in terms of OP_{TT}, is presented as a general colorscale, where dark blue represents low overfitting values, green stands for medium ovefitting values and red indicates relatively high overfitting values.

Dataset similarity has an indirect influence on overfitting in the regression model, due to the chained multi-output regression approach. Still the dataset similarity was chosen in the figure to complement Figure 5-34. An alternative to display the overfitting would be replacing dataset similarity with $Dice_{test}$ on the outer level y-axis.

5.5.5 Model Validation

Modellvalidierung

An approach to train models only on one individual tool type yielded similar results compared to the model trained on a mixed dataset in literature, see Annex A.14, Comparison of cutting tool wear segmentation models. It was found that models trained on individual, homogeneous datasets tend to perform as well as larger mixed models using the U-Net architecture [BERG20]. This finding paired with the data scarcity in a niche problem like cutting tool wear showed the need for specialized models with high accuracy that can be successfully trained with small datasets and possibly be selected via image classifiers or data similarity measures, to match a new image with the best fitting model. To validate the decision model four cases were constructed. A small dataset and a large mixed dataset which are used to train an optimized model and a non-optimized model according to the decision model. The optimized model is trained with network size and dropout rate that are expected to lead to a high Dice coefficient. The non-optimized model is trained with network size and dropout rate that are expected to lead to low Dice coefficients based on the findings from the decision model plots above in Figure 5-34. This investigation may also approve or reject the above statement from the paper cited.

Experimental Setup for Inline Measurements

Experimenteller Aufbau für Inline-Messungen

The setup of the experiment is almost identical to the setup described in Section 4.3, Process Specification. This includes the workpiece, the machine tool, and the cutting tool. Only one process parameter, namely cutting speed, distinguishes from the formerly conducted experiments. The cutting speed is chosen at a different value than

the ones represented in the original DOE. This way, the experiment conducted in scope of this chapter, serves as a validation for the tool wear model created in Section 4.4, Empirical Investigation of Tool Wear, as a byproduct. The manual measurements of cutting tool wear are conducted as displayed in Subsection 4.4.2, Analysis of Occurring Tool Wear, using a common measuring microscope and the provided analysis tools for human operators to extract wear metrics from the images.

The automated measurement uses the machine tool integrated measuring device prototype described in detail below to conduct an inline approach to cutting tool wear image acquisition developed at Fraunhofer IPT. The prototype consists of an off-the-shelf USB microscope and a housing that was custom made to protect the microscope. Power is supplied through a plastic hose that is partially fixed at the machine tool table. An angular slit allows removal of coolant from the tool using pressurized air. The housing is mounted close to the tool setting laser device on the backside of the machine tool table, see Figure 5-37.

With machine tool axes movement, the cutting tool is moved into a predetermined positions in the cameras focus point. Through incremental rotation of the spindle, it is possible to capture images of each cutting edge of the respective cutting tool.

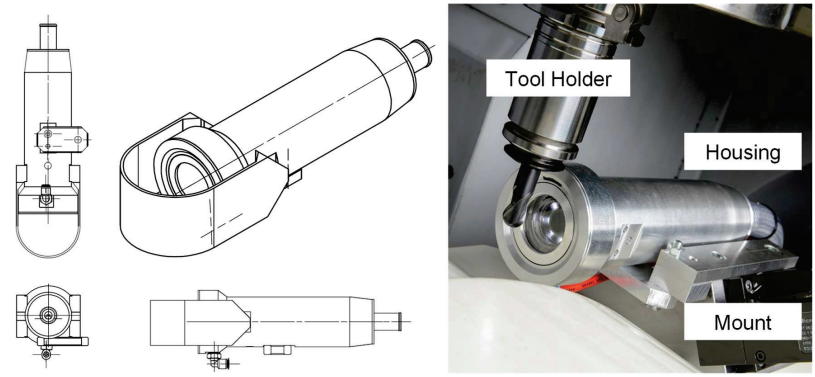


Figure 5-37: Design of the housing for the Dino-Lite AM73915MZTL (left) and the machine tool integrated prototype (right)
Design des Gehäuses für die Dino-Lite AM73915MZTL (links) und der maschinenintegrierte Prototyp (rechts)

The microscope used for the inline measurement approach is a Dino-Lite AM73915MZTL. The model offers uncompressed image quality and color reproduction in a compact metal housing. The mounted microscope inside the custom-made housing is shown in Figure 5-37 (right).

The specifications of the microscope Dino-Lite AM73915MZTL are given in Table 5-7. Figure 5-38 shows the location and setup during measurement within the Makino D500 machine tool in the validation trials.

Table 5-7: Specifications of the microscope Dino-Lite AM73915MZTL
Spezifikationen des Mikroskops Dino-Lite AM73915MZTL

Specification	Value
Sensor Type	Complementary metal–oxide–semiconductor (CMOS)
Resolution	5 MP (2592x1944)
Magnification	10x - 140x
Connector	USB 3.0
Illumination	8 LEDs (white)
Weight / Dimensions	110 g / 11.9 cm x 3.3 cm
Price Range	1.000 - 1.250 €

For the inline measurements a bright and homogenenous background is provided by a 3D printed shield made of thermoplastic filaments with a bright color. The magnification is set to 30x which translates to a working distance of 72.5 mm and a field of view in x and y direction of 13 respectively 9.7 mm. The depth of field in this setting is approximately 3.1 mm.

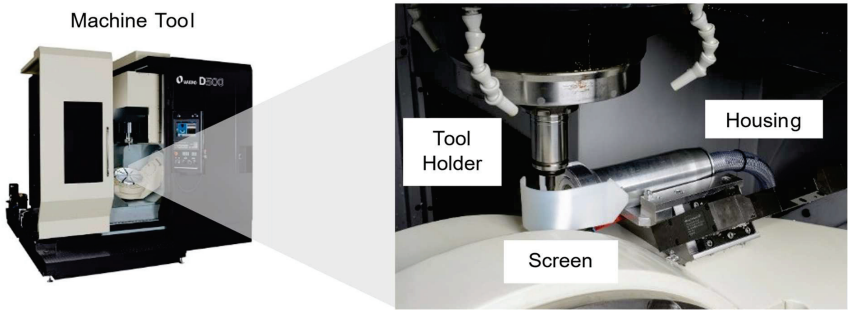


Figure 5-38: Machine tool Makino D500 and integration site with microscope
Werkzeugmaschine Makino D500 und Integrationsort mit Mikroskop

Small dataset
Kleiner Datensatz

In case of a relatively small dataset produced with the inline microscope, the decision model was used to generate favorable and unfavorable model hyperparameters to observe the effect on the model's accuracy and overfitting. Figure 5-39 below shows the non-optimized model training curves and the optimized training curves. The non-optimized training on the left shows a very volatile behavior throughout the whole training, a convergence does not take place albeit the high number of epochs. The optimized training on the right shows several short breakdowns during training.

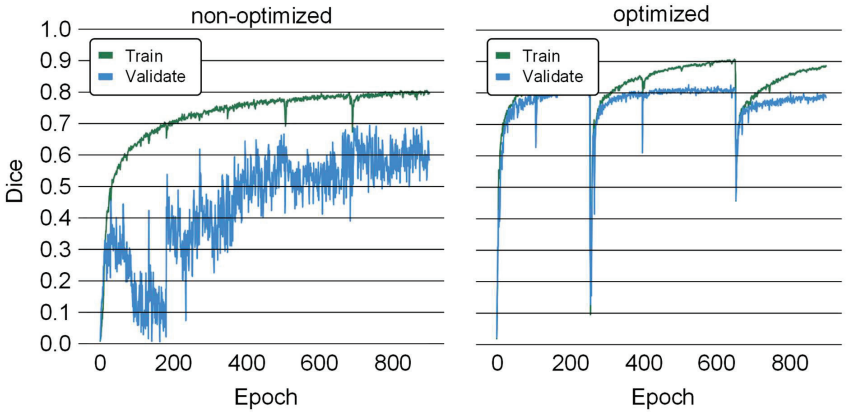


Figure 5-39: Dice coefficient over Epochs of the training based on the small dataset for the non-optimized (left) and the optimized (right) model
Dice-Koeffizient über Epochen des Trainings basierend auf dem kleinen Datensatz für das nicht optimierte (links) und das optimierte (rechts) Modell

The overfitting cycles start with a divergence of the training and validation curve. The model tends to memorize the training data rather than learning to identify relevant patterns in the data which leads to breakdowns. When held out test data are processed with the two models, the following performance is observed, see Table 5-8.

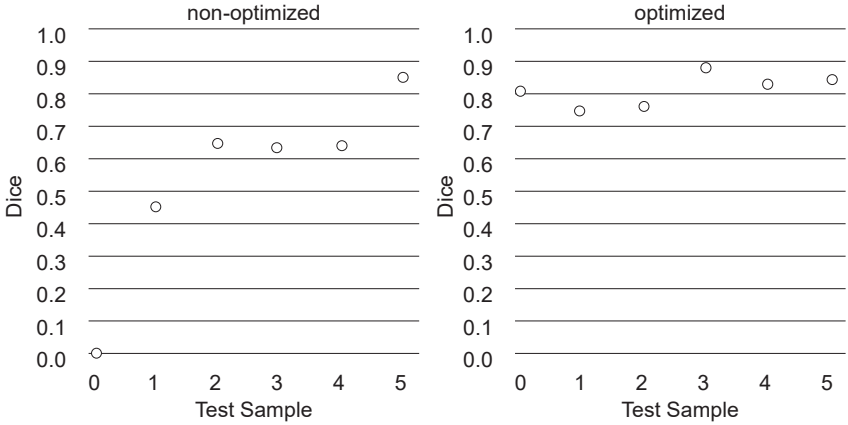


Figure 5-40: All test data predictions based on the small dataset for the non-optimized (left) and the optimized (right) model
Alle Prädiktionen von Testdaten basierend auf dem kleinen Datensatz für das nicht optimierte (links) und das optimierte (rechts) Modell

Table 5-8: Dataset properties, non-optimized and optimized model hyperparameters, as well as model performance metrics for the small dataset
Datensatzeigenschaften, nicht optimierte und optimierte Hyperparameters sowie die Performanzmetriken der Modelle für den kleinen Datensatz

Experiment	Dataset Properties		Model Hyperparameters		Performance Metrics	
	Dataset Size	Data Similarity	Network Size	Dropout Rate	$Dice_{test}$	OP_{TT}
non-opt.	46	0.92	122k	0.6	0.54	0.08
optimized	46	0.92	1.941k	0.2	0.85	0.12

The optimized model has a test and train accuracy of $mDice_{test}$ = 0.85 and $mDice_{train}$ = 0.97 respectively. The non-optimized model has an accuracy of $mDice_{test}$ = 0.54 and $mDice_{train}$ = 0.59 respectively. Figure 5-41 shows the Dice coefficient of the individual test data points. In the right column the predicted mask is displayed with a white transparent marker showing mispredicions in the tool wear area, more specifically the segmentation was incomplete in all three cases.

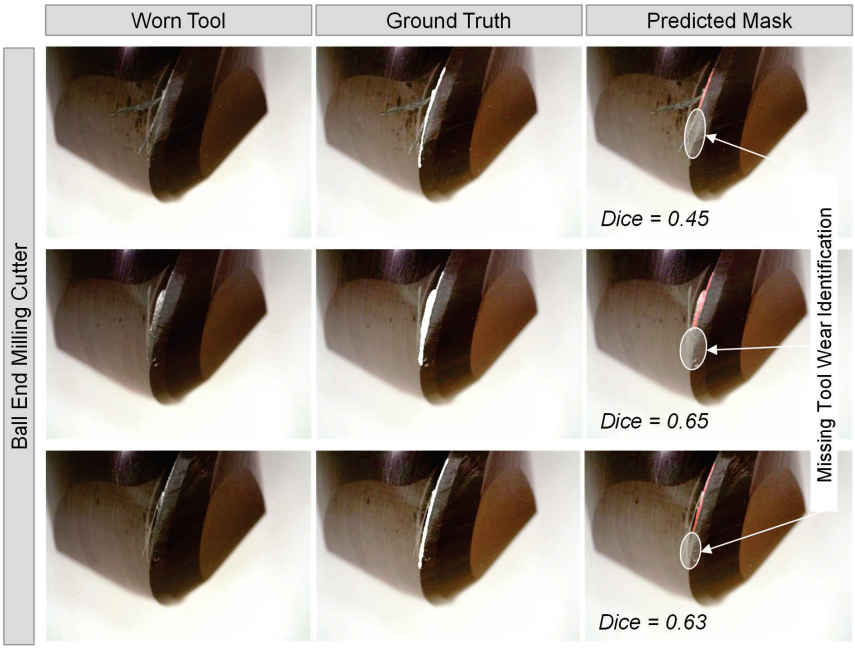


Figure 5-41: Test data predictions with non-optimized model for the small dataset
Prädiktionen auf Testdaten mit nicht optimiertem Modell auf dem kleinen Datensatz

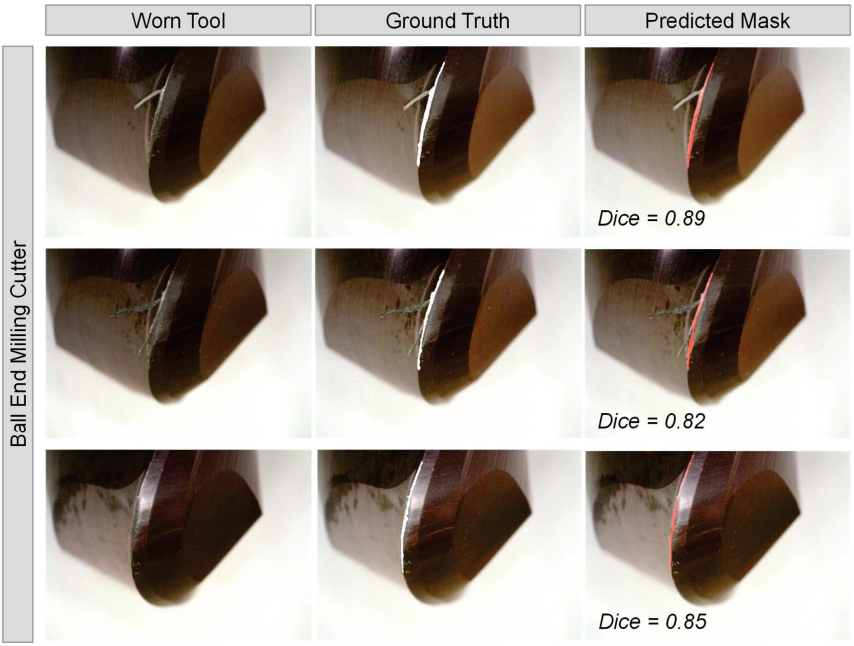


Figure 5-42: Test data predictions with optimized model for the small dataset
Prädiktionen auf Testdaten mit optimiertem Modell auf dem kleinen Datensatz

The overfitting metric for the non-optimized model is better than for the optimized model. This emphasises that this metric cannot be used alone. A bad training performance and a similarly bad test performance led to a favorable overfitting metric while the model is not useful. Figure 5-41 shows examples of test data for the above-mentioned models. The red outlines indicate locations where the model misses to predict tool wear, as opposed to the expected ground truth identified by a human operator.

Large dataset
Großer Datensatz

In case of a relatively large dataset the decision model was used to generate favorable and unfavorable model hyperparameters to observe the effect on the model's accuracy and overfitting. The non-optimized model stagnates at an unfavorable Dice coefficient while the optimized model reaches favorable values. In both cases the training and validation curves converge early on during the training, see Figure 5-44. The optimized model has a test and train accuracy of $mDice_{test}$ = 0.88 and a $mDice_{train}$ = 0.92 respectively. The non-optimized model has a test and train accuracy of $mDice_{test}$ = 0.78 and a $mDice_{train}$ = 0.75 respectively. Hence the overfitting metric for the non-optimized model is 0.5 % better than for the optimized model. Again, this emphasises again that this metric cannot be used alone.

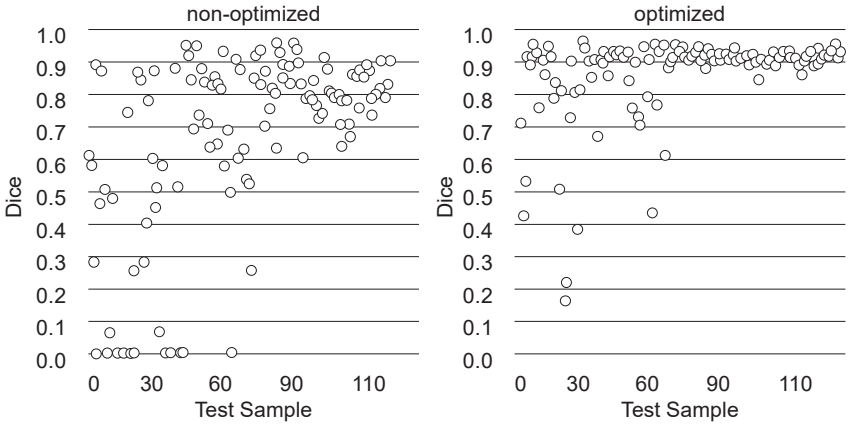


Figure 5-43: All test data predictions based on the large dataset for the non-optimized (left) and the optimized (right) model
Alle Prädiktionen von Testdaten basierend auf dem großen Datensatz für das nicht optimierte (links) und das optimierte (rechts) Modell

Figure 5-45 and Figure 5-46 show examples of test data for the non-optimized and optimized model. When held out test data is processed with the two models, the following performance is observed, see Figure 5-43 which shows the Dice coefficient of the individual test data points.

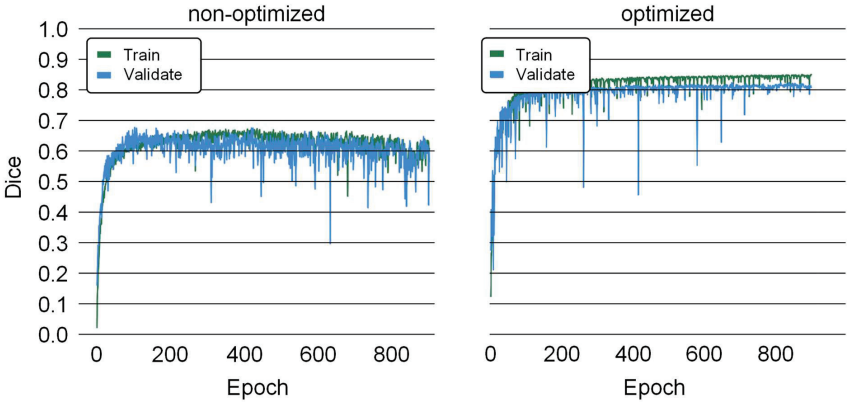


Figure 5-44: Dice coefficient over Epochs of the training based on the large dataset for the non-optimized (left) and the optimized (right) model
Dice-Koeffizient über Epochen des Trainings basierend auf dem großen Datensatz für das nicht optimierte (links) und das optimierte (rechts) Modell

Table 5-9: Dataset properties, non-optimized and optimized model hyperparameters, as well as model performance metrics for the large dataset
Datensatzeigenschaften, nicht optimierte und optimierte Hyperparameters sowie die Performanzmetriken der Modelle für den großen Datensatz

Experiment	Dataset Properties		Model Hyperparameters		Performance Metrics	
	Dataset Size	Data Similarity	Network Size	Dropout Rate	$Dice_{test}$	OP_{TT}
non-opt.	1200	0.78	122k	0.6	0.74	0.05
optimized	1200	0.78	1.941k	0.2	0.88	0.09

The below figure contains the original image, the ground truth image, and the prediction mask of the none-optimized model. In the righmost column the mispredictions may be observed. Specifically, the model segmented not the complete wear area on the cutting edges of these tools. This kind of error may lead to mismeasurements in a downstream measurement algorithm to obtain industry-specific wear metrics for the tool status assessment.

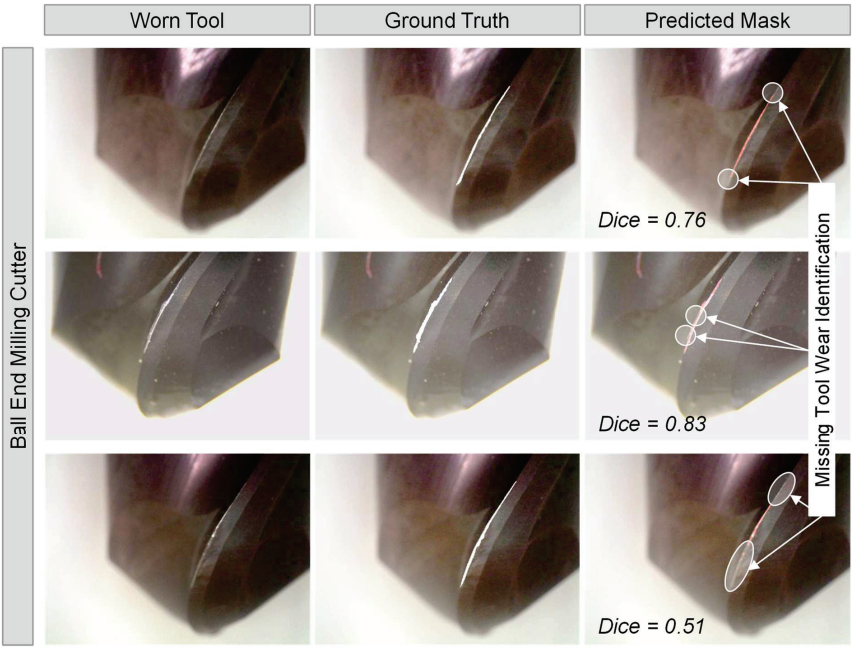


Figure 5-45: Test data predictions with non-optimized model for the large dataset
Prädiktionen auf Testdaten mit nicht optimiertem Modell auf dem großen Datensatz

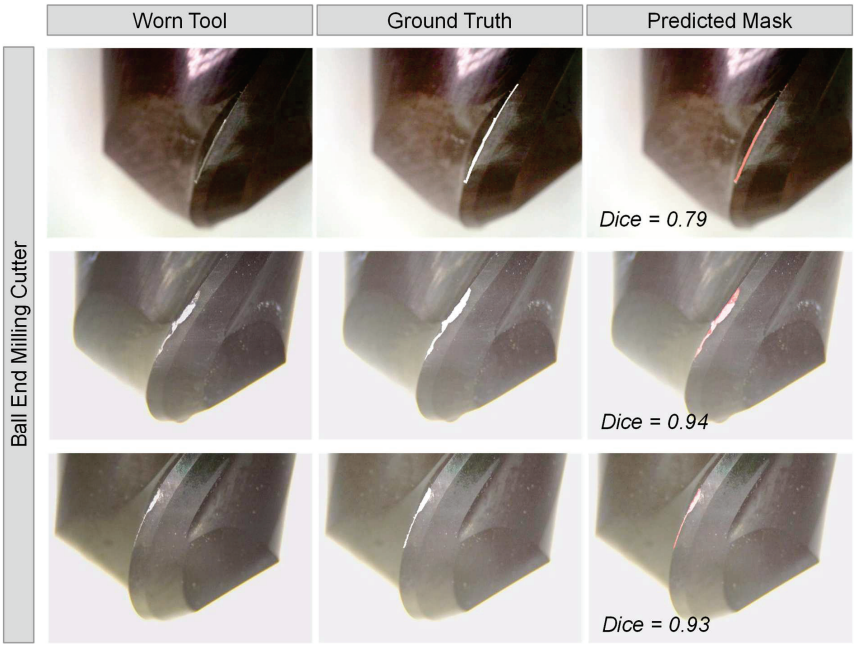


Figure 5-46: Test data predictions with optimized model for the large dataset
Prädiktionen auf Testdaten mit optimiertem Modell auf dem großen Datensatz

5.5.6 Discussion of Findings

Diskussion der Ergebnisse

As previously mentioned in the analysis of main effects in Section 5.4 Full Factorial Analysis, a high level of data similarity leads to more accurate networks across all other combinations of factors. Additionally, a larger dataset size also tends to result in higher accuracy for the networks. However, a deviation from this tendency can be observed for high dataset similarity values of 0.9 at a dataset size of 200. This effect can not be explained with certainty at this point. Possible reasons might be an insufficiency in the number of experiments conducted or an unfavorable combination of dataset and model hyperparameters leading to this deviation.

The tendency towards better accuracies with larger networks is observed across the entire domain, compare Figure 5-34. This means that regardless of the properties of the dataset, the overall trends dominate the segmentation model accuracy. This dominance of the overall tendency is also evident in the validation of the model, where both the small and large datasets yielded the same model hyperparameters from the decision model. While this indicates that a general optimization of the model performance was achieved, it also suggests that a dataset-specific optimization could not be shown.

Turning to overfitting, a dominant effect of network size is visible. Based on the regression model, smaller networks tend to overfit less than larger networks. This finding makes sense because with a high abundance of parameters, the model can learn the data domain with a high level of granularity, meaning it can learn the data by heart. However, a higher model accuracy comes with a higher tendency towards overfitting across the observed domain. Generally, a high model accuracy can be expected at dropout values from 0.2 to 0.4 and with high network sizes. At the same time, low overfitting values can be expected at lower network sizes, suggesting that there is a trade-off between accuracy and overfitting that needs to be considered when selecting the appropriate model hyperparameters.

5.6 Interim Conclusion

Zwischenfazit

Chapter 5 Model Performance Optimization gave an answer to Research Question 2: "*What are the dataset and model properties with the highest impact on model performance for tool wear segmentation?*" and Reasearch Question 3: *"How can a systematic choice of hyperparameters with regards to dataset properties be employed to improve model performance for tool wear segmentation?"*

Chapter 5, Model Performance Optimization, started with Section 5.1, Methodology, which cointains the description of the concept to create a decision model for hyperparameter selection in deep learning semantic image segmentation based on the dataset properties. Section 5.2 introduced necessary Prerequisites and Definitions to prepare for Section 5.3, Screening Analysis, where a fractional factorial design served for exploring the most influential model hyperparameters and dataset properties with regards to the model evaluation metrics. The identified most important factors are dataset size and image similarity in case of the dataset properties and dropout rate and network size in case of the model hyperparameters. Further in Section 5.4, Full Factorial Analysis, the factors identified as most important were investigated in a grid search design of experiment for the creation of a Decision Model in Section 5.5. The Decision Models predict model evaluation metrics, accuracy and overfitting, based on the factors described above. The model aimed for a target value optimization of favorable hyperparameters based on the properties of a given image dataset for cutting tool flank wear segmentation. The dataset similarities and dataset sizes linear and quadratic terms explain more than 80 % of the variance in the data. Therefore, dataset similarity and dataset size are good indicators to predict if a dataset yields an acceptable model. That means a general optimization of the hyperparameters is possible, but as expected from the weak interacting terms does not allow for a dataset property specific hyperparameter optimization. The model validation showed that for two different datasets in terms of dataset size and similarity the decision models yield the same model hyperparameters. The decision model could be used to narrow down the relevant range of a finer hyperparameter optimisation with, for example, the grid search method.

Annex A.14 provides a summary of the research conducted to date in the field of automated image processing for cutting tool wear monitoring with deep learning complemented by the results from this chapter. The table emphasises that the optimized models perform well and in line with results from research in this field. The utilization of a segmentation model, that was optimized using the decision model, in a machine tool integrated setup is described in the following chapter.

6 Validation of AI-based Automated Tool Wear Measurement

Validierung der KI-basierten automatisierten Werkzeugverschleißmessung

In this chapter, the approach to NN model optimization based on dataset properties will be applied in a real-world use case defined in the fourth Research Question:

> RQ4: How can the optimized segmentation model be applied for an inline approach to cutting tool wear measurement within machine tools?

To answer the research question, the following Section 6.1 Empirical Validation contains the algorithm to derive the wear metric *VB* from a segmented tool wear images. It also contains a fundamental experiment and an application-oriented experiments for the use of inline hardware and the image processing software in metal cutting. An accuracy assessment is given for the automated approaches in comparison with the manual approach to cutting tool flank wear measurement. Section 6.2, Economic Considerations, contains an approximation of the possible impact of an intelligent approach to tool wear measurements on the waste of cutting tools as well as a display of possible time savings through automated tool wear measurements in research and tool making facilities.

6.1 Empirical Validation of the AI-based Measurement

Empirische Evaluation der KI-basierten Messung

The empirical validation is conducted using a fundamental experiment and an application-oriented experiment. In both cases inline image acquisition and AI image processing is used with manual measurements as a benchmark.

6.1.1 Calculation of Width of Flank Wear Land VB

Berechnung der Verschleißmarkenbreite VB

This subsection introduces algorithms to calculate wear metrics from the optimized segmentation model. Specifically, an algorithm is introduced to arrive at the metric maximum width of flank wear land VB_{max} from a segmented wear area.

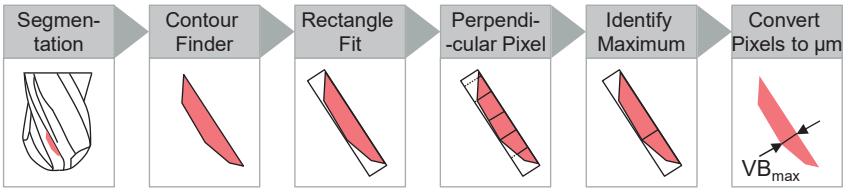


Figure 6-1: Conceptual visualization of an algorithm to calculate VB_{max} based on [HOLS22]
Konzeptionelle Visualisierung eines Algorithmus zur Berechnung von VB_{max} angelehnt an [HOLS22]

The width of flank wear land *VB* is the most common metric applied to quantify cutting tool wear in both literature and industrial practice. Below is a visual display of a possible algorithm to calculate the maximum width of flank wear land VB_{max} from a segmented wear area on the flank face of a cutting tool, specifically a ball end milling cutter. The introduced algorithm may be used to arrive at a localized *VB* value along the tool axis or the cutting edge for a more detailed resolution of wear information, see step four in the Figure 6-1 above.

6.1.2 Fundamental Trial for Validation of the AI-based Wear Measurement

Grundlagenuntersuchung zur Validierung der KI-basierten Verschleißmessung

The setup of the fundamental experiment is identical with the setup described in Section 4.3, Process Specification. This includes the workpiece, the machine tool, and the cutting tool. Only one process parameter, namely cutting speed, distinguishes from the formerly conducted experiments. The cutting speed is chosen at a different value than the ones represented in the original DOE. This way, the experiment conducted in scope of this chapter, serves as a validation for the tool wear model created in Section 4.4, Empirical Investigation of Tool Wear, as a byproduct. The manual measurements of cutting tool wear are conducted as displayed in Subsection 4.4.2, Analysis of Occurring Tool Wear, using a common measuring microscope and the provided analysis tools for human operators to extract wear metrics from the images. The automated measurement uses the machine tool integrated prototype described in detail in Subsection 5.5.5, Model Validation, to conduct an inline approach to cutting tool wear image acquisition developed at Fraunhofer IPT.

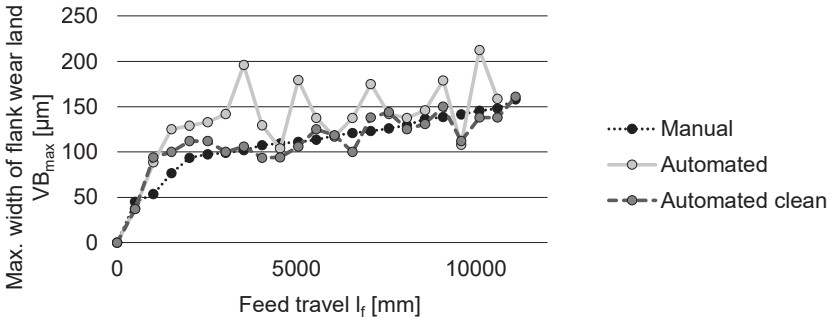


Figure 6-2: Cutting tool wear curve obtained with a measuring microscope manually (black line) and automatically obtained with an inline microscope Dino-Lite AM73915MZTL before (light grey) and after (dark gray) cleaning
Zerspanwerkzeugverschleißkurve manuell aufgenommen mit einem Standmessmikroskop sowie automatisiert erfasst mit einem inline Mikroskop Dino-Lite AM73915MZTL vor (hellgrau) und nach (dunkelgrau) der Reinigung

For the validation of the AI-based wear measurement in the fundamental setup, the images generated with the inline approach are labeled and trained with optimized parameters obtained from the Decision Model created in Section 5.5. The dataset consists of 46 individual images with a dataset similarity of 0.97. The obtained optimized hyperparameters are dropout rate of 0.2 and a network size of 1.941k. A $mDice_{test}$ accuracy of 0.85 was achieved using the settings proposed by the model.

The predictions of the optimized model are processed with the image processing pipeline from Subsection 6.1.1, Calculation of Width of Flank Wear Land VB, to calculate the VB_{max} values. Since there is no absolute true answer obtainable in the interpretation of cutting tool wear, the human expert knowledge presents the benchmark for this task. Figure 6-2 shows three tool wear curves obtained with a manual approach and with an automated approach to cutting tool wear measurement. The automated measurement yields a very unstable tool wear curve. After a manual cleaning process of the cutting-edge regarding build-up edges and cold-welded chips, the prediction is more stable and closer to the manual measurement which serves as the benchmark.

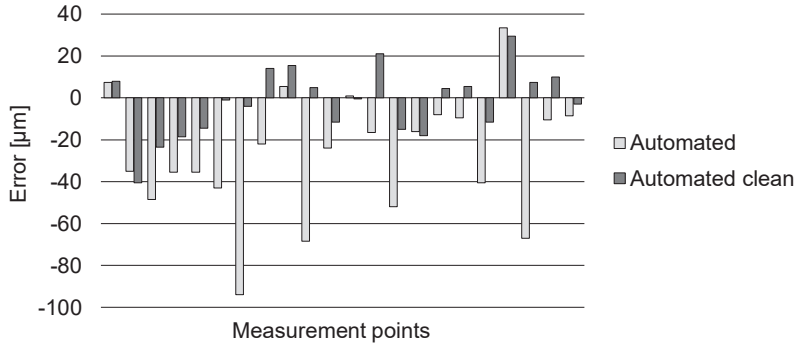


Figure 6-3: Error of automated flank wear measurement using the inline microscope Dino-Lite AM73915MZTL and the image processing pipeline compared to manual measurement using a measuring microscope
Fehler der automatisierten Freiflächenverschleißmessung bei Nutzung des Inline Mikroskops Dino-Lite AM73915MZTL und der Bildverarbeitungskette verglichen mit der manuellen Messung bei Nutzung des Standmessmikroskops

Figure 6-3 shows the error produced by the automated tool wear measurement compared to the manual measurement. Without cleaning the tools, errors up to 90 micrometers occur. After cleaning the tools, the maximum error is 40 micrometers. Taking the absolute values and calculating the mean error yields 30 micrometers in mean error for the automated measurement without cleaning and a mean error of twelve micrometers for the automated measurement after a manual cleaning step of the cutting edge.

At some points the curves slope in the Figure 6-2 above implies that the tool wear decreases locally. This behavior is not possible, i.e., cutting tool flank wear rises steadily, as its visible at the manual measurement. The variations in tool flank wear to positive and negative errors result from imprecisions and uncertainties of the image processing chain from segmentation and rule-based measurement routine.

6.1.3 Turbine Blade Milling for Validation of the AI-based Wear Measurement

Turbinenschaufelfräsen zur Validierung der KI-basierten Verschleißmessung

For the validation of the AI-based wear measurement in application a fundamentally different setup from the prior investigations was strived for. The commonality to the fundamental trials is the choice of workpiece material, which is the nickel-based alloy 2.4668 (according to EN 10027-2:1992-09). The differences are an off-the-shelf camera system for inline image acquisition, a simplified turbine blade geometry, see Figure 6-4, and the required cutting tools to produce the blade. The camera system represents the biggest change between the fundamental experimental setup and application-oriented setup. It is a wide-angle camera with the main intention to serve consumable images of the cutting process. For the trials in this subsection, the one of the camera lenses was modified to allow acquiring close-up images of cutting edges. This setup showcases the general applicability of the approach, utilizing inline hardware and AI-image processing in conjunction, for cutting tool wear monitoring in industrial practice.

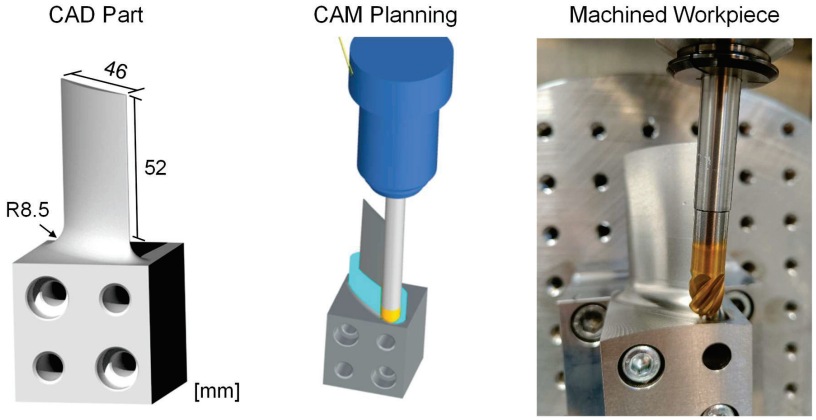


Figure 6-4: Process planning in CAM, clamped raw material and machined part
Prozessplanung in CAM, gespanntes Rohmaterial und gefertigtes Werkstück

The machine tool for conducting the metal cutting operations is a Mikron HPM800U-HD, see Figure 6-5. The cutting tools used in the machining process, as well as the applied process parameters in the roughing, semi-finishing and finishing operations to produce the simplified blade geometry may be found in Table 6-1.

Table 6-1: Process Specifications for Validation in Application
Prozessspezifikationen für die Validation in der Anwendung

Tool Specifications	Roughing	Semi-finishing	Finishing
Tool Type	End Milling Cutter	Ball End Milling Cutter	Ball End Milling Cutter
Diameter [mm]	16	16	12
Flutes	4	4	6
Corner Radius [mm]	2	8	6
Coating	AlTiN	AlTiN	AlCrN
Process Parameters			
Cutting speed [m/min]	30	35	40
Feed [mm]	0.2	0.2	0.15
Depth of cut [mm]	4	1	0.4
Width of cut [mm]	2	0.5	0.2

The cutting tool wear was inspected using a Keyence VHX-6000 microscope, see Subsection 4.4.2, Analysis of Occurring Tool Wear, and a Rotoclear C2 camera, see Figure 6-5. The specifications of the Rotoclear C2 camera are shown in Table 6-2. The unique value proposition of the inline camera is a rotating glass that prevents coolant fluid droplets from obstructing the view of the process and, in this case, the cutting tool.

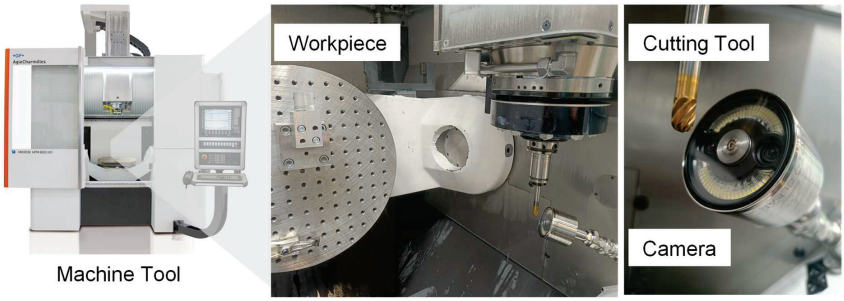


Figure 6-5: Machine tool Mikron HPM800U-HD and integration site with inline camera
Werkzeugmaschine Mikron HPM800U-HD und Integrationsort mit inline Kamera

Compared to the priorily applied Dinolite device, the inline camera has no magnifications. To apply the camera for this use case and generate images large enough to capture the tool wear, the focus points of the high-resolution lens on the right side of the camera head was manipulated to allow taking pictures at 2.5 cm proximity.

Table 6-2: Specifications of the camera Rotoclear C2
Spezifikationen der Kamera Rotoclear C2

Specification	Value
Resolution	1280x720, 1920x1080, 3840x2160
Magnification	1x
Connector	M12 x-codiert
Illumination	LEDs (white)
Weight / Dimensions	600 g / 7 cm x 5 cm
Price Range	3.000 - 4.000 €

This way, details of the cutting tool edges and the occuring tool wear can be captured. The visual tool wear inspection was conducted after each process step of roughing, semi-finishing and finishing withing a block of the blade, see Figure 6-6.

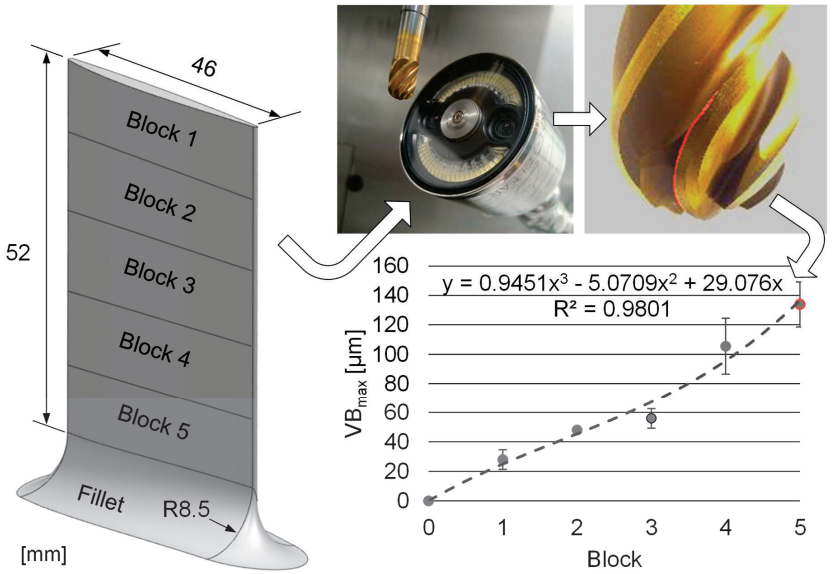


Figure 6-6: Schematic display of the machine tool integrated tool status inspection during turbine blade finish milling, dimensions measured in millimeters
Schematische Darstellung der in die Werkzeugmaschine integrierten Werkzeugstatuskontrolle beim Schlichtfräsen von Turbinenschaufeln, Abmaße sind in Millimetern gegeben

For the automated analysis of the cutting tool wear in this application-oriented validation experiments, several tests were conducted. Models were trained on the large dataset from Subsection 5.5.5, Model Validation, together with the newly acquired Rotoclear data. Additionally, a model was trained on all the newly acquired data combined. The large dataset with the newly acquired camera data yielded a $mDice_{test}$ = 0.84 with an mOP_{TT} = 0.09. The model trained with all the new camera data but without other existing data yielded a $mDice_{test}$ = 0.76 with an mOP_{TT} = 0.22. Compared to the prior model the model trained only on new data shows a significant deterioration. It uses 68 data points as compared to the prior model using 1268 data points. The training curves in Figure 6-7 have different characteristics. The model trained on the large dataset, including historical data acquired with other devices, such as a measuring microscope, has a stable training. The dataset using new inline camera data only has an unstable training process and a significantly higher overfitting tendency.

In addition to the two trainings described above, a model was trained with the newly acquired data of each process step individually, utilizing BIM to increase the dataset size. The results of the individual dataset models are given in Annex A.17, Comparison of segmentation models trained on different datasets in the application-oriented trial.

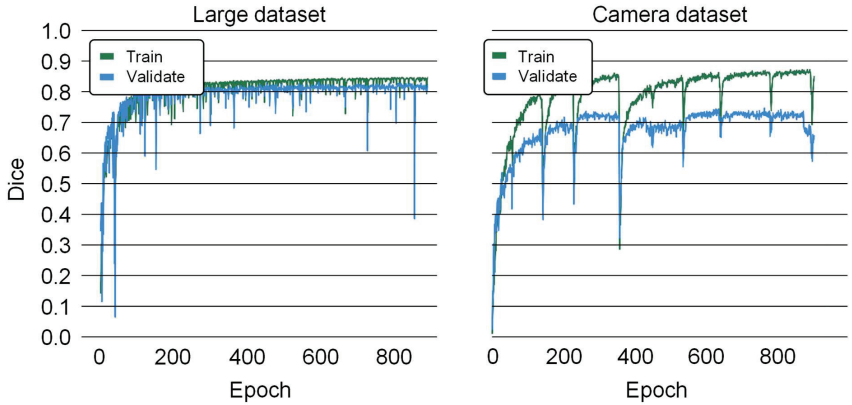


Figure 6-7: Dice coefficient over Epochs for the large and the camera dataset
Dice Koeffizient über Epochen für den großen und den Kameradatensatz

The Figure 6-8 contains test data of each of the models described above. The data is sorted by operation type, i.e., tool type for clarity. The top row contains test data of the three individual models, according to tool type. The white circles show flaws, were the models fail to predict the desired areas with flank wear or where flank wear is detected by the model mistakenly. Due to the large areas of flank wear on the roughing tools, the accuracies tend to be higher than for the semi-finishing and finishing tools, were a small error leads to a higher decrease in Dice coefficient.

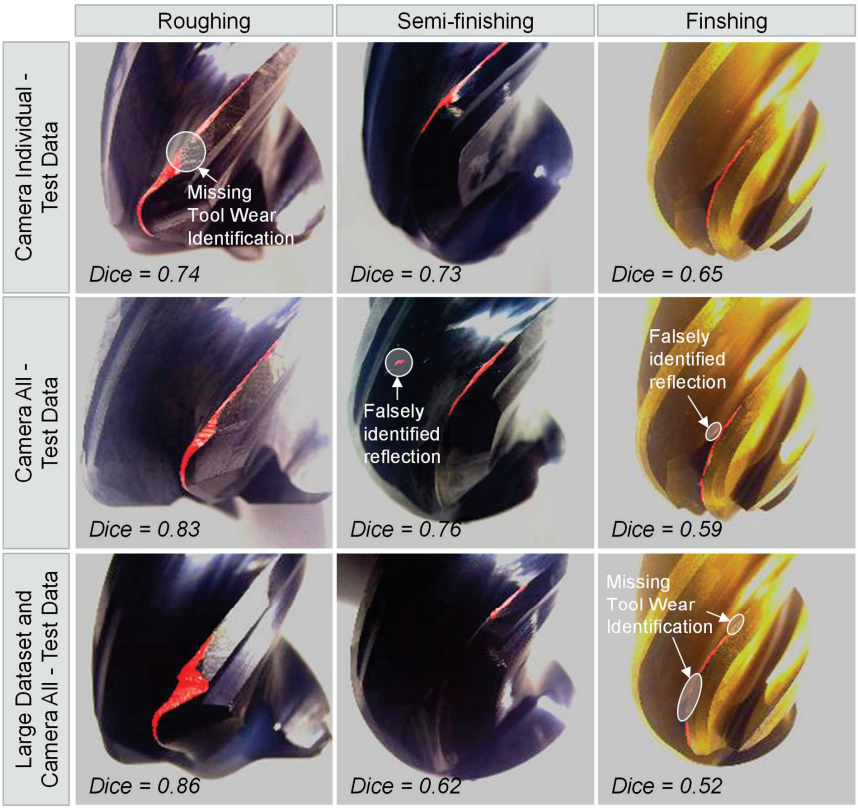


Figure 6-8: Test data of roughing, semi-finishing and finishing tools of models trained on individual camera tool datasets, all camera data and the existing data together with camera data
Testdaten von Schrupp-, Vorschlicht- und Schlichtwerkzeugen von Modellen, die mit einzelnen Kamera-Werkzeugdatensätzen, allen Kamera-Daten und den vorhandenen Daten zusammen mit Kamera-Daten trainiert wurden

Based on the segmented tool wear area, VB_{max} values were calculated for quantification of the tool wear extent. The flank wear characteristics exhibit arcs along the cutting edge due to the cutting tools engagement with the workpiece material, especially due to the hub and fillet feature at the blades transition to the clamping block. For this reason, the measurement routine introduced earlier in Subsection 6.1.1, Calculation of Width of Flank Wear Land VB, was adapted. The adaptation concerns the rectangle fit which can be conducted stepwise for an arbitrary number of pixels to account for the curvature of the wear area, which otherwise falsifies the automated VB_{max} measurement.

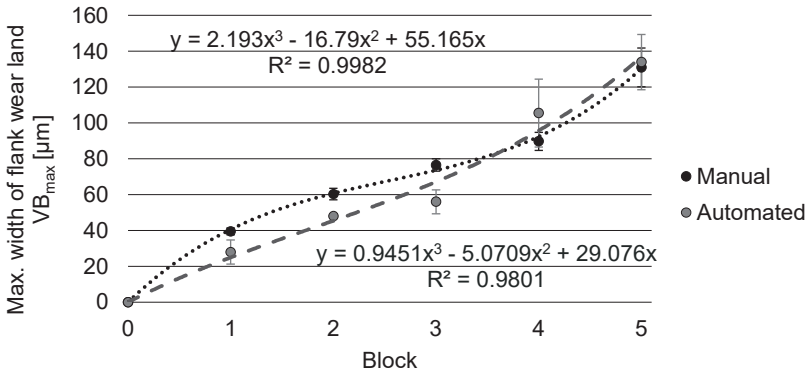


Figure 6-9: Finishing tool flank wear progression in terms of VB_{max} during turbine blade milling with standard deviation and polynomial regression fit
Verlauf des Freiflächenverschleißes des Schlichtwerkzeugs beim Fräsen der Turbinenschaufel mit Standardabweichung und polynomialer Regression

For the finishing tool the VB_{max} curve acquired using this adapted measurement algorithm is displayed in Figure 6-9. The relative error between the manual measurements using the measuring microscope and the automated measurement is up to 30 % in relative terms or 20 micrometers in absolute terms, see Figure 6-10.

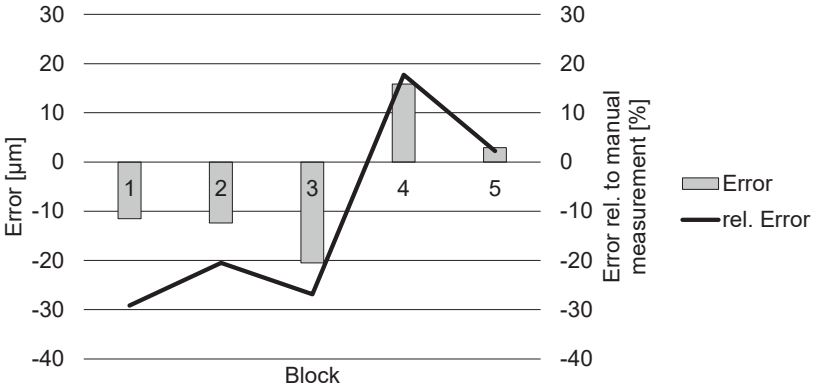


Figure 6-10: Absolute and relative error between automated and manual measurement for the finishing tools' VB_{max} value per block
Absolute und relative Abweichung zwischen automatischer und manueller Messung für den VB_{max}-Wert pro Block dss Schlichtwerkzeugs

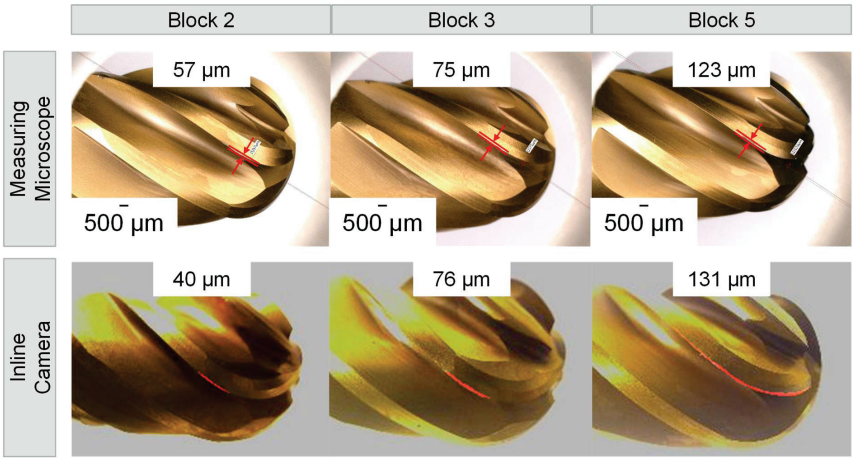


Figure 6-11: Finishing tool measuring microscope images (top) and inline camera images (bottom) with mean VB_{max} values of the respective process step
Schlichtwerkzeugbilder aufgenommen mit dem Standmessmikroskop (oben) und mit der inline Kamera (unten) mit dem mittleren VB_{max} des Prozessschritts

Figure 6-11 shows examples of images of the finishing tool acquired with the measuring microscope and the camera. The individual match between cutting edges is not possible due to the lacking quality in the image data. The measurement shown for the measuring microscope images is a singular measurement of the exact cutting tool displayed, the values shown for the camera are mean values of the cutting edges.

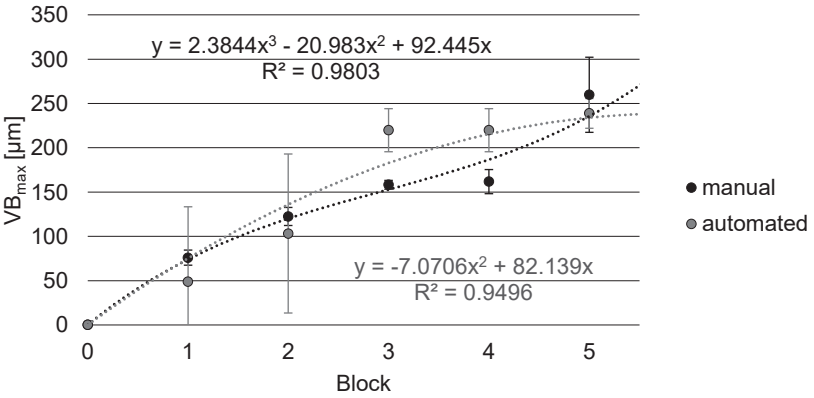


Figure 6-12: Semi-finishing tool flank wear progression in terms of VB_{max} during turbine blade milling with standard deviation and polynomial regression fit
Verlauf des Freiflächenverschleißes des Vorschlichtwerkzeugs beim Fräsen der Turbinenschaufel mit Standardabweichung und polynomialer Regression

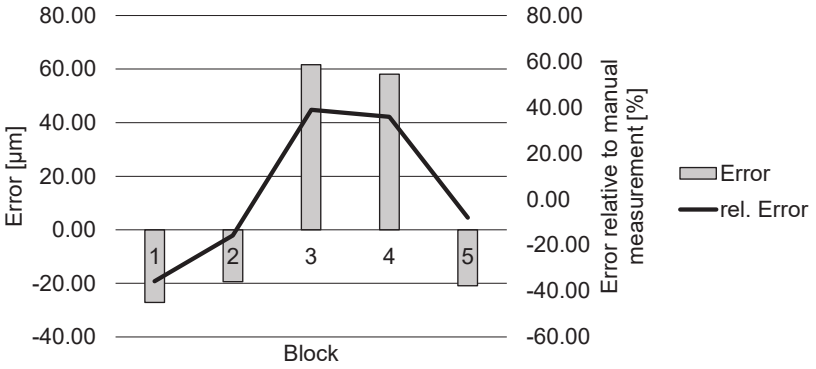


Figure 6-13: Absolute and relative error between automated and manual measurement for the semi-finishing tools' VB_{max} value per block
Absolute und relative Abweichung zwischen automatischer und manueller Messung für den VB_{max}-Wert pro Block des Vorschlichtwerkzeugs

For the semi-finishing tool, the VB_{max} curve acquired is displayed in Figure 6-12. The relative error between the manual measurements using the measuring microscope and the automated measurement is up to 40 % in relative terms or 60 micrometers in absolute terms, see Figure 6-13.

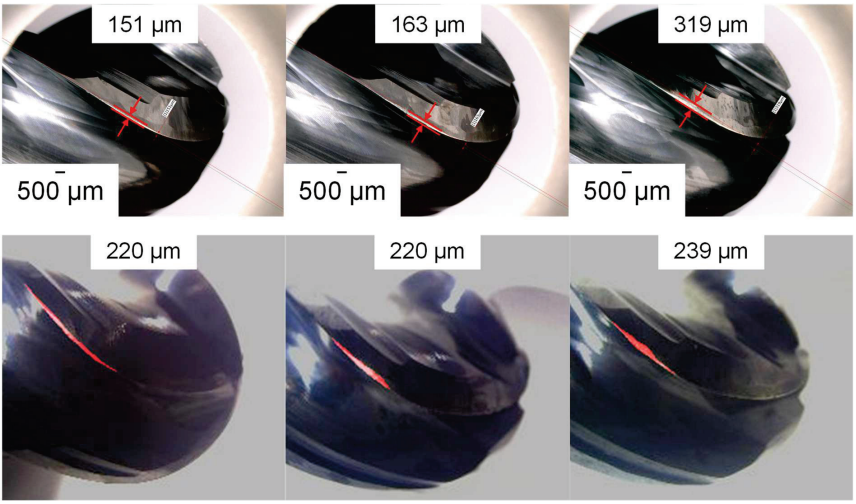


Figure 6-14: Semi-finishing tool microscope images (top) and inline camera images (bottom) with mean VB_{max} values of the respective process step
Vorschlichtwerkzeugbilder aufgenommen mit dem Standmessmikroskop (oben) und mit der inline Kamera (unten) mit dem mittleren VB_{max}

Figure 6-14 shows examples of images of the semi-finishing tool acquired with the measuring microscope and the camera. The individual match between cutting edges is not possible due to the lacking quality in the image data. The measurement shown for the measuring microscope images is a singular measurement of the exact cutting tool displayed, the values shown for the camera are mean values of the cutting edges.

For the roughing tool the same analysis was not conducted since the wear form deviated from the typical flank wear. Specifically, during the fourth block, one cutting edge experienced chipping, see Figure 6-8. Afterwards a new tool was used for the remaining operations. Due to these reasons, the display of a typical flank wear curve is not feasible.

6.2 Economic Considerations

Ökonomische Betrachtungen

The estimate of the participants of the surveys in Section 4.1, Surveys with Industry Professionals, for wasted tool potential is 15 - 30 % of the tools useful life. The worldwide cutting tool market has a size of roughly 73 billion Euro per year, with a european share of 12.4 billion Euro [STAT 22b]. Considering a 5 % reduction in wasted tool potential using digital technologies as presented in this thesis, roughly 3.5 billion Euro worldwide and 0.62 billion Euro in Europe could be saved annually.

Apart from that, in the research environment of public institutes or at cutting tool manufacturers, a lot of manual tool wear measurements are necessary to test tool performance, optimize tool geometries or to create tool life models as conducted in Section 4.4, Empirical Investigation of Tool Wear. The new approach to measuring key metrics of cutting tool wear, presented in Section 6.1, Empirical Validation, requires a critical assessment of the economic feasibility of the measurement process. To do so, a direct comparison of the manual and automated measurement process is conducted in this subsection. The tool wear curve acquisition described in the beginning of this chapter was also used for a comparison of the manual and automated cutting tool wear measurement process. In detail, nine of the twentythree measurement processes conducted with the manual approach were measured with a stopwatch to identify the time required for a typical manual tool wear measurement. The measurement process at the microscope itself accounted for more than 80 % of the total time. The remaining 20 % accounted to dismounting and mounting the cutting tool holder out of and into the machine tool as well as the transfer of the cutting tool between the machine and the measurement device. In the beginning the manual measurements were quite fast at roughly 150 seconds minimum. At a later stage of the cutting tool wear measurement the process took almost 250 seconds maximum, with the mean values at roughly 180 seconds.

The automated measurement, presented in Section 6.1 Empirical Validation, includes movement of the tool to the inline microscope and the rotation of the tool in front of the camera to capture all cutting edges. The process is described in more detail in the

following paragraph. Firstly, a movement of the spindle with the clamped tool in front of the camera is conducted. Specifically, the surface to be measured must be in the focus point of the microscopic camera. This process may be automated with a laser measurement bridge determining the actual radius of the current tool beforehand. When the tool is well positioned in the focus point, an intermittend rotation may be used to rotate the tool by an angle φ and halts for a unblurry image acquisition for a specified time t_φ. With $\varphi = 10°$, $t_\varphi = 1$ second and retraction movements of around 20 seconds, the time totals to roughly 60 seconds.

The the total time for the manual measurement ranges between 200 and 300 seconds. The mean time required for a manual measurement process is 250 seconds. In comparison the automated measurement takes 60 seconds on average. The variance in the time measures of the automated approach stems from the use of a stopwatch by a human operator.

As a side note, the tool wear model created in Section 4.4, Empirical Investigation of Tool Wear, predicts a cutting time of 12.88 minutes for the tool investigated in this chapter, while in the experiment a cutting time of 11.47 minutes was measured. This corresponds to an error of roughly 11 %.

6.3 Interim Conclusion

Zwischenfazit

In Chapter 6, Validation of AI-based Automated Tool Wear Measurement, the approach to NN model optimization based on dataset properties was applied in a real-world use case to answer the fourth Research Question: *“How can the optimized segmentation model be applied for an inline approach to cutting tool wear measurement within machine tools?”*

Section 6.1 Empirical Validation introduces an algorithm to calculate the flank wear on cutting tool edges based on the segmented flank wear area. The section demonstrates how an inline measurement of cutting tool edges is realized using a low-cost microscope and a custom-made housing to withstand the harsh environment in the machine tool. Additionally, the approach was demonstrated with in an application-oriented use case in a different setup. With usage of the machine tool axes and the measurement algorithm introduced in the prior section, a cutting tool flank wear curve was recorded in an automated manner. At the same time the tool status was manually measured with a measuring microscope to generate a benchmark measurement. For the direct comparison of the manual and automated approach the measurement accuracy and the required time were acquired. The accuracy and consistency of the manual measurement is higher than the automated measurement. The automated approach yields a mean error of 30 micrometers, without a manual cleaning step. With a manual cleaning step, which removes interfering bodies from the tool cutting edge, the mean error compared to the manual measurement can be reduced to twelve micrometers. In the application-oriented setup, an industrial camera for the working area of a machine tool

was used to assess flank wear on cutting tools during blade milling. The error between the manual measurement and the automated measurement was up to 30 % in relative terms or 20 micrometers in absolute terms. For the semi-finishing process the error was to 45 % in relative and 60 micrometers in absolute terms.

The chapter demonstrates how an optimized segmentation model can be applied in an industrial environment for the automated measurement of the cutting tools flank wear using inline hardware within a machine tool. In the next chapter, the findings are summarized and critically reflected. Furthermore, an outlook is given on newly arising questions regarding the application of inline measurements of tool wear and other possibilities that arise through the automated tool wear segmentation and measurement aided by deep learning methods.

7 Summary and Outlook

Zusammenfassung und Ausblick

Summary

Zusammenfassung

The motivation behind this research stemmed from the realization that the existing approach to addressing cutting tool wear in the metal cutting industry could benefit from enhancement. The following outlines the three methods presently employed in serial production, research and development, and small-batch production:

1. Fixed tool life from prior test with a safety margin to account for outliers
2. Creation of tool life models to allow a prediction of tool life across a range of cutting speed and/or other cutting parameters
3. Optical observation and assessment of the tool status by the machine tool operator

The current widely used first method of using fixed tool life expectations with large margins to account for the high variance in tool life, leads to avoidable costs. Additionally, the absence of knowledge regarding the tools status prevents intelligent cutting tool management and hinders intelligent automation of manufacturing processes regarding demand-driven tool changes and teaching of possible indirect tool wear estimation models relying on other machine tool integrated sensors.

The second approach was conducted within this thesis, see Section 4.4, revealed some weaknesses: The parameter domain boundaries, in this case cutting speed, for the model creation is not exactly known prior to testing which causes additional effort. Also, a high variance in tool life travel path occurs for repetitive experiments which may undermine the statistical prerequisites for a flawless model creation. Furthermore, the required resolution of the parameter domain is not known, which can lead to additional effort without benefits to models' usefulness. Finally, a variation in the process parameters or process conditions will render the model useless unless additional elaborate experiments are conducted to extend the model accordingly.

The third approach requires manual effort and relies on the experience and judgement of individual operators. Therefore, this approach is not scalable and the knowledge transfer to new operators is critical. A standardization may not be obtainable at all using this approach.

This work aimed to present an alternative way to deal with the tool wear problem in machining by introducing a method to automate segmentation of cutting tool flank wear with a subsequent automated measurement using an image processing chain. Applied with a machine tool integrated measuring device, the proposed approach yields a viable solution to the problems described above. In detail, the steps to achieve the above stated solution was split into four research questions:

Research Question 1: "How can image processing be applied to automatically detect tool wear on microscopic images of cutting tool edges?"

The segmentation of flank wear on microscopic images of cutting tools was achieved with a supervised DL approach, specifically the U-Net architecture. This NN architecture is used to perform semantic image segmentation. In more detail the following steps were conducted:

- Aggregation and selection of microscopic images of worn cutting tools
- Manual label process to create masks of the areas of interest (tool wear) on each of the images. The label masks serve as ground truth for network training.
- Image augmentation methods such as BIM or GAN-based data synthesis
- Selection of dataset properties and model hyperparameters such as: dataset split, image size, network size, learning rate, momentum, activation functions and dropout rate
- Start of training process, i.e., model parameter optimization
- Evaluation of model performance with metrics that describe model accuracy and robustness regarding test dataset

The tool wear segmentation approach using the U-Net architecture for semantic segmentation presented achieved a mean Dice coefficient of $mDice_{test}$ = 0.82 on test data. Training data consisted of 3000 augmented images originating from 400 raw images. The heterogeneous raw image dataset consisted of eight different cutting tool datasets with 50 images each and various levels of magnification. On an inference dataset, which contains unknown images recorded with disturbances like increased or decreased brightness, the network yielded a mean Dice coefficient of $mDice_{inf}$ = 0.54.

Research Question 2: “What are the dataset and model properties with the highest impact on model performance for tool wear segmentation?”

Based on a two-stage factorial DOE the model hyperparameters activation function, learning rate, dropout rate, network size and momentum as well as the dataset properties dataset size, image size, dataset split, and dataset similarity were investigated. According to the effect strength of the individual factors on the model evaluation metrics, especially test data accuracy, a ranking was produced which yields the following most influential parameters:

Dataset properties:

- Dataset size
- Image similarity

Model hyperparameters:

- Dropout rate
- Network size

Research Question 3: “How can a systematic choice of hyperparameters with regards to dataset properties be employed to improve model performance for tool wear segmentation?”

After the identification of the most influential factors, a full factorial DOE was conducted with these four remaining factors. Using an outlier analysis, a final database for model creation was filtered. The database was used for a regression of the model accuracy based on the four most influential factors from above. The regression model was further used for target value optimization to allow a selection of favorable hyperparameters based on the dataset properties of a given dataset for cutting tool flank wear segmentation. A general optimization of the hyperparameters was possible, but due too weak interaction terms, between dataset properties and model hyperparameters, did not allow for a dataset property specific hyperparameter optimization. The model validation showed that for two different datasets in terms of dataset size and similarity the decision model yielded the same optimized model hyperparameters.

During the regression modelling approach of the model accuracy, it was found that the dataset similarity and the dataset size explained more than 80 % of the variance in accuracy. That means datasets can be analyzed prior to modelling and an expected accuracy can be generated without training the model.

Research Question 4: “How can the optimized segmentation model be applied for an inline approach to cutting tool wear measurement within machine tools?”

With a low-cost microscope and a custom-made housing to make it withstand the harsh environment in the machine tool, an inline measurement of cutting tool edges is realized. Using the machine tool axes the cutting tool edges were recorded in an automated manner. At the same time the tool status was manually measured with a measuring microscope to generate a benchmark measurement. For the direct comparison of not only accuracy but also time requirements of the two approaches, a stopwatch was applied. The data collected in this trial was labeled and used to create an optimized model with the target value optimization from Research Question 3. An $mDice_{test}$ accuracy of 0.85 was achieved using the settings proposed by the decision model using only 46 example images. The automated measurement algorithm yields a mean error of twelve micrometers across the 23 measurements conducted for the acquired tool wear curve compared to the manual measurement with the measuring microscope. Furthermore, an industrial grade inline camera for the working area of the machine tool was applied for cutting tool measurements during a blade milling operation. The relative error between a manual measurement and the automated measurement with this method ranged from two to 30 % for a ball end milling cutter in finish milling and up to 45 % for a ball end milling cutter in semi-finishing.

Recapitulating the conclusion from Chapter 2 Fundamentals and State of the Art, the following problems in the domain of automated image analysis for cutting tool flank wear segmentation with deep learning algorithms have been addressed for the first time in this thesis, contributing to the novelty degree of this research:

1. **Knowledge about effects of model hyperparameters** on performance of a DNN for tool wear segmentation was obtained for the use case of cutting tool images.

2. **Metrics to characterize and compare datasets**, like dataset similarity, were tested and compared for the use case of cutting tool images. In this thesis the selected dataset similarity metric was used for a dataset property dependent selection of model hyperparameters. The use of metrics to characterize datasets may enable targeted model selection and / or combination, using e.g., federated learning, in the future.
3. **An approach to measuring overfitting** was proposed which can be constructed from different combinations of the training, test, validation datasets.
4. **Underspecification** of models was addressed using a systematical approach to hyperparameter optimization and testing of models with inline data.

Outlook

Ausblick

Still some open questions and topics to be researched remain which are assigned to the following system describing the building blocks of image acquisition and processing for surface inspections, see Figure 7-1.

With reference to the figure this thesis covered the levels ***Evaluation*** and ***Utilization*** in some detail. Whereas there are numerous possibilities to research further in these two levels: For example, a Neural Architecture Search (NAS) could be performed to identify network architectures that are superior to the U-Net in tasks with sparse data. Furthermore, image augmentation through Basic Image Manipulation (BIM) or Generative Adversarial Networks (GANs) and their effect on neural network performance in cutting tool segmentation could be investigated in the future. Other than that, some methods presented in Subsection 2.3.4, Tool Wear Identification with Deep Learning, such as layer-wise loss, have not been tested with a standardized dataset to evaluate the possible improvements in prediction quality.

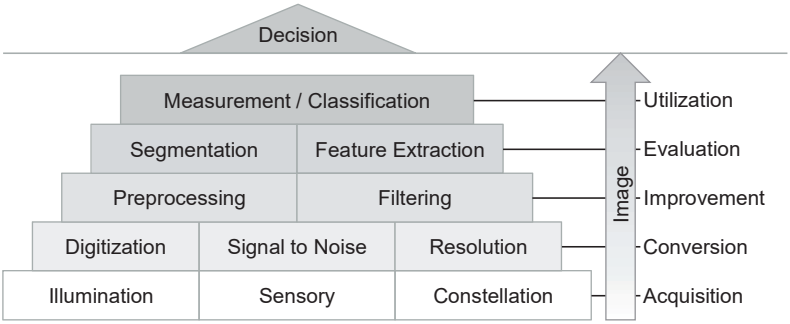


Figure 7-1: Pyramid of image acquisition and processing for surface inspection on the basis of LÄNGLE [LÄNG16]
Pyramide der Bildgewinnung und -verarbeitung zur Oberflächeninspektion in Anlehnung an LÄNGLE [LÄNG16]

Also, the measurement method that was introduced covers only the *VB* metric so far and could be extended for other wear form except for flank wear. Other than that, the levels ***Conversion*** and ***Improvement***, for example preprocessing steps such as sharpening with wavelet filter and pretraining the model weights, have not been covered in this thesis. The level ***Acquisition*** has only been addressed superficially, while significant advances in data quality and data improvement could be possible with better measurement devices and supporting illumination.

Especially, the pyramid level ***Decision*** allows for creative solutions in cyber physical production systems. Apart from obvious applications like demand-driven tool changes, the tool status information in conjunction with other sensor data may be used to continously capture knowledge from multiple machine tool integrated sensors and use them to approximate the tool status. In conjunction with analytical models, analogous to the procedure presented in Subsection 6.1.1, Calculation of Width of Flank Wear Land VB, further wear metrics, such as chipping (CH) or catastrophic failure (CF), can be determined using image processing for the evaluation of the tool condition and used for decision-making.

Apart from the above-mentioned possibilities of further research, a standardized set of labeled image datasets is necessary to compare approaches and possible improvements for different sizes and types of datasets in the domain of automated analysis of microscopic image data for cutting tool status identification. Furthermore, a method for classifying and rating datasets using KPIs could be helpful, including metrics like image quality, image similarity and dataset size.

7 Zusammenfassung und Ausblick

Summary and Outlook

Zusammenfassung

Summary

Ausgangspunkt dieser Arbeit war die Erkenntnis, dass die derzeitige Art und Weise des Umgangs mit dem Verschleiß von Zerspanungswerkzeugen in der metallverarbeitenden Industrie verbessert werden könnte. Im Folgenden werden die drei derzeit in der Serienproduktion, in der Forschung und Entwicklung und in der Kleinserienfertigung angewandten Methoden vorgestellt:

1. Festgelegte Werkzeugstandzeit aus einer vorherigen Prüfung mit einer Sicherheitsspanne, um Ausreißer zu berücksichtigen
2. Erstellung von Standzeitmodellen, die eine Vorhersage der Werkzeugstandzeit über einen Bereich von Schnittgeschwindigkeiten und/oder anderen Schnittparametern ermöglichen
3. Optische Beobachtung und Bewertung des Werkzeugstatus durch den Bediener der Werkzeugmaschine

Die derzeit weit verbreitete erste Methode, bei der feste Standzeiterwartungen mit großen Spielräumen verwendet werden, um der hohen Varianz der Werkzeugstandzeit Rechnung zu tragen, führt zu vermeidbaren wirtschaftlichen Kosten. Darüber hinaus verhindert die mangelnde Quantifizierung des Werkzeugstatus ein intelligentes Werkzeugmanagement und hemmt die intelligente Automatisierung von Fertigungsprozessen im Hinblick auf einen bedarfsgerechten Werkzeugwechsel und das Einlernen eines möglichen indirekten Modells zur Abschätzung des Werkzeugverschleißes, das auf anderen in die Werkzeugmaschine integrierten Sensoren beruht.

Die zweite der oben genannten Methoden, die im Rahmen dieser Arbeit demonstriert wurde (siehe Abschnitt 4.4), zeigte einige Schwächen auf: Die Grenzen des Parameterbereichs, in diesem Fall die Schnittgeschwindigkeit, für die Modellerstellung sind vor dem Test nicht genau bekannt, was zusätzlichen Aufwand verursacht. Außerdem tritt bei sich wiederholenden Versuchen eine hohe Streuung des Standwegs auf, was die statistischen Voraussetzungen für eine einwandfreie Modellerstellung verhindern kann. Darüber hinaus ist die erforderliche Auflösung des Parameterbereichs nicht bekannt, was zu zusätzlichem Aufwand führen kann, ohne dass dies der Nützlichkeit des Modells zugutekommt. Schließlich wird das Modell bei einer Änderung der Prozessparameter oder der Prozessbedingungen unbrauchbar, wenn nicht zusätzliche Experimente durchgeführt werden, um das Modell entsprechend zu erweitern.

Der dritte Ansatz erfordert manuellen Aufwand und stützt sich auf die Erfahrung und das Urteilsvermögen der einzelnen Bediener. Daher ist dieser Ansatz nicht skalierbar

und der Wissenstransfer an neue Bediener ist kritisch. Eine Standardisierung ist mit diesem Ansatz möglicherweise gar nicht möglich.

In dieser Arbeit wurde versucht, eine Lösung für alle drei oben genannten Fälle zu finden, indem eine Methode zur automatischen Segmentierung des Freiflächenverschleißes mit anschließender automatischer Messung mittels einer Bildverarbeitungskette eingeführt wurde. Angewandt auf ein in die Werkzeugmaschine integriertes Messgerät, bietet der vorgeschlagene Ansatz eine praktikable Lösung für die oben beschriebenen Probleme. Im Einzelnen wurden die Schritte zur Erreichung der oben genannten Lösung in vier Forschungsfragen unterteilt:

Forschungsfrage 1: "Wie kann die Bildverarbeitung zur automatischen Erkennung von Werkzeugverschleiß auf mikroskopischen Bildern von Schneidkanten von Zerspanungswerkzeugen eingesetzt werden?"

Die Segmentierung von Freiflächenverschleiß auf mikroskopischen Bildern von Zerspanungswerkzeugen wurde mit einem überwachten DL-Ansatz, insbesondere der U-Net-Architektur, erreicht. Diese NN-Architektur wird verwendet, um eine semantische Bildsegmentierung durchzuführen. Im Einzelnen sind die folgenden Schritte erforderlich:

- Aggregation und Auswahl von mikroskopischen Bildern mit verschlissenen Zerspanungswerkzeugen
- Manuelles Beschriftungsverfahren zur Erstellung von Masken der interessierenden Bereiche (Werkzeugverschleiß) auf jedem der Bilder. Die Beschriftungsmasken dienen als Grundlage für das Netzwerktraining.
- Methoden zur Bildanreicherung wie BIM oder GAN-basierte Datensynthese
- Auswahl von Datensatzeigenschaften und Modellhyperparametern wie: Datensatzaufteilung, Bildgröße, Netzwerkgröße, Lernrate, Momentum, Aktivierungsfunktionen und Dropout-Rate
- Beginn des Trainingsprozesses, d.h. Optimierung der Modellparameter
- Bewertung der Modellleistung mit Metriken, die die Genauigkeit und Robustheit des Modells in Bezug auf den Testdatensatz und, falls verfügbar, den Inferenzdatensatz beschreiben

Der vorgestellte Ansatz zur Segmentierung von Zerspanungswerkzeugverschleiß, der die U-Net-Architektur zur semantischen Segmentierung nutzt, erreichte bei Testdaten einen mittleren Dice-Koeffizienten von $mDice_{test}$ = 0.82. Die Trainingsdaten bestanden aus 3000 augmentierten Bildern, die aus 400 Rohbildern gewonnen wurden. Der heterogene Rohbilddatensatz bestand aus acht verschiedenen Zerspanungswerkzeugdatensätzen mit jeweils 50 Bildern und verschiedenen Vergrößerungsstufen. Bei einem Inferenzdatensatz, der unbekannte Bilder enthält, die mit Störungen wie erhöhter oder verringerter Helligkeit aufgenommen wurden, ergab das Netz einen mittleren Dice-Koeffizienten von $mDice_{inf}$ = 0,54.

Forschungsfrage 2: "Welches sind die Datensatz- und Modelleigenschaften mit dem größten Einfluss auf die Modellleistung bei der Segmentierung von Werkzeugverschleiß?"

Basierend auf einer zweistufigen faktoriellen DOE wurden die Modell-Hyperparameter Aktivierungsfunktion, Lernrate, Dropout-Rate, Netzwerkgröße und Momentum sowie die Datensatzeigenschaften Datensatzgröße, Bildgröße, Datensatzaufteilung und Datensatzähnlichkeit untersucht. Entsprechend der Effektstärke der einzelnen Faktoren auf die Modellbewertungsmetriken, insbesondere die Testdatengenauigkeit, wurde eine Rangfolge erstellt, die die folgenden einflussreichsten Parameter ergibt:

Datensatzeigenschaften:

- Größe des Datensatzes
- Bildähnlichkeit

Modell-Hyperparameter:

- Dropout-Rate
- Netzwerkgröße

Forschungsfrage 3: "Wie kann eine systematische Auswahl von Hyperparametern in Bezug auf die Eigenschaften des Datensatzes getroffen werden, um die Modellleistung zu verbessern?"

Nach der Identifizierung der einflussreichsten Faktoren wurde eine vollständige faktorielle DOE mit diesen vier verbleibenden Faktoren durchgeführt. Mit Hilfe einer Ausreißer-Analyse wurde eine endgültige Datenbank für die Modellerstellung gefiltert. Die Datenbank wurde für eine Regressionsmodellierung der Modellgenauigkeit auf der Grundlage der vier einflussreichsten Faktoren von oben verwendet. Das Regressionsmodell wurde ferner zur Zielwertoptimierung verwendet, um eine Auswahl günstiger Hyperparameter auf der Grundlage der Datensatzeigenschaften eines bestimmten Datensatzes für die Segmentierung des Freiflächenverschleißes von Zerspanungswerkzeugen zu ermöglichen. Eine allgemeine Optimierung der Hyperparameter war möglich, aber aufgrund zu schwacher Interaktionsterme zwischen Datensatzeigenschaften und Hyperparametern war eine datensatzeigenschaftsspezifische Hyperparameteroptimierung nicht möglich. Die Modellvalidierung zeigte, dass das Entscheidungsmodell für zwei unterschiedliche Datensätze in Bezug auf die Datensatzgröße und -ähnlichkeit die gleichen optimierten Modellhyperparameter ergaben.

Bei der Regressionsmodellierung der Genauigkeit wurde festgestellt, dass die Ähnlichkeit der Datensätze und die Größe der Datensätze mehr als 80 % der Varianz in der Genauigkeit erklären. Das bedeutet, dass die Datensätze vor der Modellierung analysiert werden können und eine erwartete Genauigkeit ohne Training des Modells generiert werden kann.

Forschungsfrage 4: "Wie kann das optimierte Segmentierungsmodell für einen Inline-Ansatz zur Messung des Zerspanungswerkzeugverschleißes in Werkzeugmaschinen angewendet werden?"

Mit einem kostengünstigen Mikroskop und einem speziell angefertigten Gehäuse, das den rauen Umgebungsbedingungen in der Werkzeugmaschine standhält, wurde eine Inline-Messung von Zerspanungswerkzeugschneiden realisiert. Über die Achsen der Werkzeugmaschine wurden die Schneidkanten automatisiert erfasst. Gleichzeitig wurde der Werkzeugzustand manuell mit einem Standmessmikroskop gemessen, um eine Benchmark-Messung zu erstellen. Für den direkten Vergleich nicht nur der Genauigkeit, sondern auch des Zeitbedarfs der beiden Ansätze, wurde eine Stoppuhr eingesetzt.

Die in diesem Versuch gesammelten Daten wurden gelabelt und zur Erstellung eines optimierten Modells mit der Zielwertoptimierung aus Forschungsfrage 3 verwendet. Mit den vom Entscheidungsmodell vorgeschlagenen Einstellungen und 46 Beispielbildern wurde eine $mDice_{test}$-Genauigkeit von 0,85 erreicht. Der automatische Messalgorithmus ergibt einen mittleren Fehler von zwölf Mikrometern bei den 23 durchgeführten Messungen für die erfasste Werkzeugverschleißkurve im Vergleich zur manuellen Messung mit dem Standmessmikroskop. Darüber hinaus wurde eine Industriekamera für den Arbeitsbereich der Werkzeugmaschine zur Messung des Zerspanungswerkzeugs während eines Fräsvorgangs für eine Turbinenschaufel eingesetzt. Der relative Fehler zwischen einer manuellen Messung und der automatischen Messung mit dieser Methode lag zwischen zwei und 30 % für einen Kugelkopffräser beim Schlichtfräsen und bis zu 45 % für einen Kugelkopffräser im Vorschlichten.

Rekapituliert man die Schlussfolgerung aus Kapitel 2 Grundlagen und Stand der Technik, so wurden die folgenden Probleme im Bereich der automatisierten Bildanalyse für die Segmentierung von Freiflächenverschleiß mit DL-Algorithmen in dieser Arbeit zum ersten Mal behandelt, was zum Neuheitsgrad dieser Forschung beiträgt:

- **Wissen über die Auswirkungen von Modellhyperparametern** auf die Leistung eines DNN zur Segmentierung von Werkzeugverschleiß wurden für den Anwendungsfall von Bilddaten von Zerspanungswerkzeugen gewonnen.
- **Metriken zur Charakterisierung und zum Vergleich von Datensätzen**, wie die Datensatzähnlichkeit, wurden für den Anwendungsfall von Zerspanungswerkzeugbildern getestet und verglichen. In dieser Arbeit wurde die gewählte Metrik der Datensatzähnlichkeit für eine datensatzeigenschaftsabhängige Auswahl von Modellhyperparametern verwendet. Die Verwendung von Metriken zur Charakterisierung von Datensätzen kann in Zukunft eine gezielte Modellauswahl und / oder -kombination, z.B. durch föderiertes Lernen, ermöglichen.
- **Es wurde ein Ansatz zur Messung der Überanpassung vorgeschlagen**, der aus verschiedenen Kombinationen von Trainings-, Test- und Validierungsdatensätzen erstellt werden kann.

- **Die Unterspezifikation von Modellen** wurde durch einen systematischen Ansatz zur Optimierung von Hyperparametern und Testen der Modelle mit Inline-Daten addressiert.

Ausblick
Outlook

Es verbleiben noch einige offene Fragen und zu erforschende Themen, die dem folgenden System zugeordnet sind, das die Bausteine der Bilderfassung und -verarbeitung für Oberflächeninspektionen beschreibt, siehe Abbildung 7-2.

Unter Bezugnahme auf die Abbildung wurden in dieser Arbeit die Ebenen ***Evaluation*** und ***Utilization*** ausführlich behandelt. Es gibt jedoch zahlreiche Möglichkeiten, auf diesen beiden Ebenen weiter zu forschen: Zum Beispiel könnte eine neuronale Architektursuche (NAS) durchgeführt werden, um Netzarchitekturen zu identifizieren, die dem U-Net bei Aufgaben mit spärlichen Daten überlegen sind.

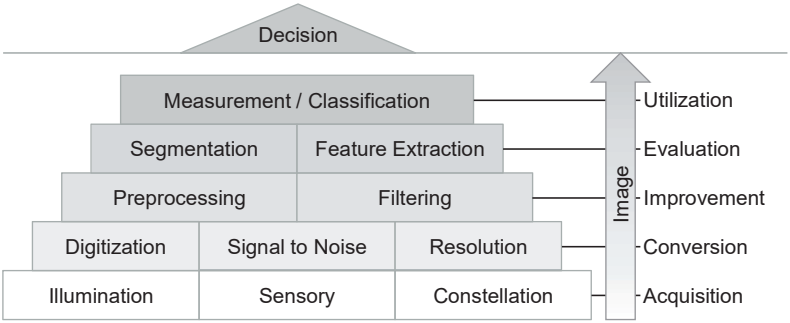


Figure 7-2: Pyramide der Bildgewinnung und -verarbeitung zur Oberflächeninspektion in Anlehnung an Fraunhofer-Allianz Vision [LÄNG16]
Pyramid of image acquisition and processing for surface inspection on the basis of Fraunhofer-Allianz Vision [LÄNG16]

Darüber hinaus könnten in Zukunft Bildaugmentationstechniken wie Basic Image Manipulation (BIM) oder Generative Adversarial Networks (GANs) und ihre Auswirkungen auf die Leistung neuronaler Netze bei der Segmentierung von Zespanungswerkzeugen untersucht werden. Darüber hinaus wurden einige der in Unterabschnitt 2.3.4, Tool Wear Identification with Deep Learning, vorgestellten Methoden, wie z. B. der schichtweise Loss, nicht mit einem standardisierten Datensatz getestet, um die möglichen Verbesserungen der Vorhersagequalität zu bewerten. Auch deckt die vorgestellte Messmethode bisher nur die VB-Metrik ab und könnte für andere Verschleißformen außer Freiflächenverschleiß erweitert werden. Darüber hinaus wurden die Ebenen ***Conversion*** und ***Improvement***, z. B. Vorverarbeitungsschritte wie das Schärfen mit Wavelet-Filter und das Vortrainieren der Gewichtungsfaktoren, wurden in dieser Arbeit nicht behandelt. Die Ebene ***Acquisition*** wurde nur oberflächlich behandelt, obwohl mit besseren Messgeräten und

unterstützender Beleuchtung erhebliche Fortschritte bei der Datenqualität und Datenverbesserung möglich wären.

Insbesondere die Pyramidenebene ***Decision*** ermöglicht kreative Lösungen in cyberphysikalischen Produktionssystemen. Neben offensichtlichen Anwendungen wie dem bedarfsgerechten Werkzeugwechsel kann die Werkzeugstatusinformation in Verbindung mit anderen Sensordaten genutzt werden, um das Wissen mehrerer in die Werkzeugmaschine integrierter Sensoren zu erfassen und zur Approximation des Werkzeugstatus zu nutzen. In Verbindung mit analytischen Modellen, analog zu der in Unterabschnitt 6.1.1, Calculation of Width of Flank Wear Land VB, vorgestellten Vorgehensweise, können weitere Verschleißmetriken, wie z.B. Ausbrüche (CH) oder katastrophale Ausfälle (CF), mithilfe der Bildverarbeitung für die Bewertung des Werkzeugzustands berechnet und für die Entscheidungsfindung genutzt werden.

Neben den oben genannten Möglichkeiten der weiteren Forschung ist ein standardisierter Satz von beschrifteten Bilddatensätzen notwendig, um Ansätze und Verbesserungsmöglichkeiten für unterschiedliche Größen und Arten von Datensätzen im Bereich der automatisierten Analyse von mikroskopischen Bilddaten zur Zustandserkennung von Zerspanungswerkzeugen zu vergleichen. Darüber hinaus könnte eine Methode zur Klassifizierung und Bewertung von Datensätzen anhand von KPIs hilfreich sein, einschließlich Metriken wie Bildqualität, Bildähnlichkeit und Datensatzgröße.

8 References

Literaturverzeichnis

[ACTI19] Active Neuron List of Deep Learning based Semantic Segmentation Models. URL: https://github.com/ActiveNeuron/List-of-Deep-Learning-based-Semantic-Segmentation-Models/blob/master/README.md

[ALEG09] Alegre, E.; Alaiz-Rodríguez, R.; Barreiro, J.; Ruiz, J. Use of contour signatures and classification methods to optimize the tool life in metal machining. In: Estonian Journal of Engineering, 15. Jg., 2009, Nr. 1, S. 3.

[ALOY17] Aloysius, N.; Geetha, M. A review on deep convolutional neural networks. 6th-8th April 2017, Melmaruvathur, India. In: 2017, S. 588–592.

[AUGS18] Augspurger, T. Thermal Analysis of the Milling Process. 1st ed. Aachen: Apprimus Wissenschaftsverlag, 2018 - ISBN: 978-3-86359-690-3.

[AVAN09] Avanaki, A. N. Exact global histogram specification optimized for structural similarity. In: Optical Review, 16. Jg., 2009, Nr. 6, S. 613–621.

[AWAD15] Awad, M.; Khanna, R. Efficient Learning Machines. In: 2015

[BERG20] Bergs, T.; Holst, C.; Gupta, P.; Augspurger, T. Digital image processing with deep learning for automated cutting tool wear detection. In: Procedia Manufacturing, 48. Jg., 2020, S. 947–958.

[BIND17] Binder, M. Mechanismenbasierte Verschleißsimulation zur integrierten Werkzeug- und Prozessauslegung. (Reihe: Technologie der Fertigungsverfahren, 2017, Band 16). 1. Auflage, 2017 - ISBN: 978-3-86359-525-8.

[BRAM06] Brambor, T.; Clark, W. R.; Golder, M. Understanding Interaction Models: Improving Empirical Analyses. In: Political Analysis, 14. Jg., 2006, Nr. 1, S. 63–82.

[BUND21] Bundesministerium des Innern, Bau und Heimat (BMI). *Leitfaden zur Personalbedarfsermittlung.* Berlin, 08 / 2021

[CANN86] Canny, J. A Computational Approach to Edge Detection. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8. Jg., 1986, Nr. 6, S. 679–698.

[CHAU17] Chaurasia, A.; Culurciello, E. LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. In: 2017, S. 1–4.

[CHEN18] Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, 07.02.2018

[CHEN21] Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Le Lu; Yuille, A. L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. In: 2021

[CHOL18] Chollet, F. Deep learning with Python. Shelter Island: Manning, 2018 - ISBN: 9781617294433.

[CIRP04] International Institution for Production Engineering Research (2004) Wörterbuch der Fertigungstechnik

[CORD16] Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): IEEE, 62016, S. 3213–3223.

[CZIC10] Czichos, H.; Habig, K.-H. Tribologie-Handbuch. Tribometrie, Tribomaterialien, Tribotechnik ; mit 123 Tabellen. (Reihe: Praxis. 3., überarb. und erw. Aufl. Wiesbaden: Vieweg + Teubner, 2010 - ISBN: 978-3-8348-0017-6.

[D'AD13] D'Addona, D. M.; Teti, R. Image Data Processing via Neural Networks for Tool Wear Prediction. In: Procedia CIRP, 12. Jg., 2013, S. 252–257.

[D'AD17] D'Addona, D. M.; Ullah, A. M. M. S.; Matarazzo, D. Tool-wear prediction and pattern-recognition using artificial neural network and DNA-based computing. In: Journal of Intelligent Manufacturing, 28. Jg., 2017, Nr. 6, S. 1285–1301.

[D'AM20] D'Amour, A.; Heller, K.; Moldovan, D.; Adlam, B.; Alipanahi, B.; Beutel, A.; Chen, C.; Deaton, J.; Eisenstein, J.; Hoffman, M. D.; Hormozdiari, F.; Houlsby, N.; Hou, S.; Jerfel, G.; Karthikesalingam, A.; Lucic, M.; Ma, Y.; McLean, C.; Mincu, D.; Mitani, A.; Montanari, A.; Nado, Z.; Natarajan, V.; Nielson, C.; Osborne, T. F.; Raman, R.; Ramasamy, K.; Sayres, R.; Schrouff, J.; Seneviratne, M.; Sequeira, S.; Suresh, H.; Veitch, V.; Vladymyrov, M.; Wang, X.; Webster, K.; Yadlowsky, S.; Yun, T.; Zhai, X.; Sculley, D. *Underspecification Presents Challenges for Credibility in Modern Machine Learning,* 06.11.2020

[DAVI06] Davis, J.; Goadrich, M. The relationship between Precision-Recall and ROC curves. In: 2006, S. 233–240.

[DHAN15] Dhanachandra, N.; Manglem, K.; Chanu, Y. J. Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. In: Procedia Computer Science, 54. Jg., 2015, S. 764–771.

[DIN6583] Deutsche Norm DIN. 6583 (1981) Begriffe der Zerspantechnik

[DIN8580] Deutsche Norm DIN. 8580 Fertigungsverfahren

[DIN8589-3] Deutsche Norm DIN. 8589-3 Fertigungsverfahren Spanen

[EZUG99] Ezugwu, E. O.; Wang, Z. M.; Machado, A. R. The machinability of nickel-based alloys: a review. In: Journal of Materials Processing Technology, 86. Jg., 1999, Nr. 1-3, S. 1–16.

[FORT22] Fortune Business Insights Metal Cutting Tools Market Size, Share & COVID-19 Impact Analysis. URL: https://www.fortunebusinessinsights.com/industry-reports/metal-cutting-tools-market-101751

[GAL15] Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, 06.06.2015

[GEIR19] Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: 2019

[GÉRO19] Géron, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. Concepts, tools, and techniques to build intelligent systems. (Reihe: Covid-19 collection. Second edition. Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly, September 2019 - ISBN: 9781492032649.

[GOOD14] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. In: arxiv.org, 2014

[GREE16] Greenspan, H.; van Ginneken, B.; Summers, R. M. Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. In: IEEE transactions on medical imaging, 35. Jg., 2016, Nr. 5, S. 1153–1159.

[GRZE17] Grzesik, W. Advanced machining processes of metallic materials. Theory, modelling and applications. Second edition. Amsterdam, Oxford, Cambridge, MA: Elsevier, 2017 - ISBN: 9780444637116.

[HE17] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. *Mask R-CNN,* 20.03.2017

[HINT19] Hinton, G. Neural Networks for Machine Learning. Lecture 6a Overview of mini-batch gradient descent. In: 2019

[HOLS21] Holst, C.; Königs, M.; Garcia, E. M.; Ganser, P.; Bergs, T. Spatially Resolved Tool Wear Prediction in Finish Milling. In: Procedia CIRP, 104. Jg., 2021, S. 85–90.

[HOLS22] Holst, C.; Yavuz, T. B.; Gupta, P.; Ganser, P.; Bergs, T. Deep learning and rule-based image processing pipeline for automated metal

cutting tool wear detection and measurement. In: IFAC-PapersOnLine, 55. Jg., 2022, Nr. 2, S. 534–539.

[IOFF15] Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 11.02.2015

[ISO1701-1] International Standard Organization. ISO1701 ISO 1701-1:2004

[ISO3002-1] International Standard Organization ISO. 3002-1:1982 (1.8.1982) Basic quantities in cutting and grinding

[ISO3685] International Standard ISO. 3685 Tool Life Testing with single point Turning

[ISO513:2012] International Organization for Standardization ISO. 513:2012 (11-2012) ISO 513:2012

[ISO8868-2] International Standard ISO. 8868-2 (1989) Tool Life Testing in Milling

[ISOL16] Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A. A. *Image-to-Image Translation with Conditional Adversarial Networks,* 21.11.2016

[JACC12] Jaccard, P. THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1. In: New Phytologist, 11. Jg., 1912, Nr. 2, S. 37–50.

[JEON88] Jeon, J. U.; Kim, S. W. Optical flank wear monitoring of cutting tools by image processing. In: Wear, 127. Jg., 1988, Nr. 2, S. 207–217.

[KALP10] Kalpakjian, S.; Schmid, S. R.; Musa, H. Manufacturing engineering and technology. 6. ed. in SI units. Singapore: Prentice Hall, 2010 - ISBN: 9810681445.

[KANO88] Kanopoulos, N.; Vasanthavada, N.; Baker, R. L. Design of an image edge detection filter using the Sobel operator. In: IEEE Journal of Solid-State Circuits, 23. Jg., 1988, Nr. 2, S. 358–367.

[KASS88] Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active contour models. In: International Journal of Computer Vision, 1. Jg., 1988, Nr. 4, S. 321–331.

[KENT21] Kentaro Wada; mpitid; Martijn Buijs; Zhang Ch. N.; なるみ; Bc. Martin Kubovčík; Alex Myczko; latentix; Lingjie Zhu; Naoya Yamaguchi; Shohei Fujii; iamgd67; IlyaOvodov; Akshar Patel; Christian Clauss; Eisoku Kuroiwa; Roger Iyengar; Sergei Shilin; Tanya Malygina; Kento Kawaharazuka; Jonne Engelberts; Aleksi J; AlexMa; Changwoo Song; Charlie; Daniel Rose; Douglas Livingstone; Doug; Erik; Henrik Toft wkentaro/labelme: v4.6.0: Zenodo, 2021

[KIEF52] Kiefer, J.; Wolfowitz, J. Stochastic Estimation of the Maximum of a Regression Function. In: The Annals of Mathematical Statistics, 23. Jg., 1952, Nr. 3, S. 462–466.

[KING14] Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization, 22.12.2014

[KLOC18] Klocke, F. Fertigungsverfahren 1. Zerspanung mit geometrisch bestimmter Schneide. 9. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018 - ISBN: 978-3-540-23458-6.

[KRIZ17] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. In: Communications of the ACM, 60. Jg., 2017, Nr. 6, S. 84–90.

[LÄNG16] Längle, T.; Heizmann, M. Leitfaden zur Inspektion und Charakterisierung von Oberflächen mit Bildverarbeitung. Sackewitz, M. Stuttgart, 2016

[LECU88] LeCun, Y. A Theoretical Framework for Back-Propagation. In: Proceedings of the 1988 Connectionist Model Summer School, 1988, S. 21–28.

[LECU98] Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, 86. Jg., 1998, Nr. 11, S. 2278–2324.

[LEE14] Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-Supervised Nets. In: 2014. Jg., 2014

[LENT89] Lenth, R. V. Quick and Easy Analysis of Unreplicated Factorials. In: Technometrics, 31. Jg., 1989, Nr. 4, S. 469–473.

[LONG15] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. Jg., 2015, S. 3431–3440.

[LUND19] Lundervold, A. S.; Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. In: Zeitschrift fur medizinische Physik, 29. Jg., 2019, Nr. 2, S. 102–127.

[LUTZ19] Lutz, B.; Kisskalt, D.; Regulin, D.; Reisch, R.; Schiffler, A.; Franke, J. Evaluation of Deep Learning for Semantic Image Segmentation in Tool Condition Monitoring. In: 2019. Jg., 2019, S. 2008–2013.

[LUTZ20] Lutz, B.; Reisch, R.; Kisskalt, D.; Avci, B.; Regulin, D.; Knoll, A.; Franke, J. Benchmark of Automated Machine Learning with State-of-the-Art Image Segmentation Algorithms for Tool Condition Monitoring. In: Procedia Manufacturing, 51. Jg., 2020, S. 215–221.

[LUTZ21] Lutz, B.; Kisskalt, D.; Regulin, D.; Aybar, B.; Franke, J. Automated Domain Adaptation in Tool Condition Monitoring using Generative Adversarial Networks. In: Procedia Manufacturing, 2021. Jg., 2021, Nr. 51, S. 1326–1331.

[MACH11] Macherauch, E.; Zoch, H.-W. Praktikum in Werkstoffkunde. 91 ausführliche Versuche aus wichtigen Gebieten der Werkstofftechnik. (Reihe: Studium Werkstofftechnik. 11., vollst. überarb. und erw. Aufl. Wiesbaden: Vieweg + Teubner, 2011 - ISBN: 383480343X.

[MAHO20] Mahony, N. O.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Velasco-Hernandez, G.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. In: 2194-5357, 943. Jg., 2020

[MCCU43] McCulloch, W. S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. In: The Bulletin of Mathematical Biophysics, 5. Jg., 1943, Nr. 4, S. 115–133.

[MENN18] Mennatullah Siam; Mostafa Gamal; Moemen Abdel-Razek; Senthil Yogamani; Martin Jagersand; Hong Zhang A Comparative Study of Real-Time Semantic Segmentation for Autonomous Driving. In: 2018

[META22] Meta Image Classification on ImageNet. URL: https://paperswithcode.com/sota/image-classification-on-imagenet [Stand: 05.01.2022]

[MIAO21] Miao, H.; Zhao, Z.; Sun, C.; Li, B.; Yan, R. A U-Net based approach for tool wear area detection and identification. In: IEEE Transactions on Instrumentation and Measurement, 2021, Nr. 70, S. 1–10.

[MOHA20] Mohanraj, T.; Shankar, S.; Rajasekar, R.; Sakthivel, N. R.; Pramanik, A. Tool condition monitoring techniques in milling process — a review. In: Journal of Materials Research and Technology, 9. Jg., 2020, Nr. 1, S. 1032–1042.

[MOLD17] Moldovan, O.; Dzitac, S.; Moga, I.; Vesselenyi, T.; Dzitac, I. Tool-Wear Analysis Using Image Processing of the Tool Flank. In: Symmetry, 9. Jg., 2017, Nr. 12, S. 296.

[MOLI21] Molitor, D. A.; Kubik, C.; Becker, M.; Hetfleisch, R. H.; Lyu, F.; Groche, P. Towards High-Performance Deep Learning Models in Tool Wear Classification with Generative Adversarial Networks. In: Journal of Materials Processing Technology, 2021, S. 117484.

MORÉ80 User Guide for Minpack-1. ANL (Series). Argonne National Laboratory Report ANL-80-74. Illinois, 1980

[MÜLL20] Müller, M. U.; Ekhtiari, N.; Almeida, R. M.; Rieke, C. Super-resolution of multispectral satellite image using convolutional neural networks. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, V-1-2020. Jg., 2020, S. 33–40.

[NWAN18] Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning, 08.11.2018

[PEDA18] Pedamonti, D. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. In: 2018

[PEDR11] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. In: Journal of Machine Learning Research, 12. Jg., 2011, S. 2825–2830.

[PERS19] Persistence Market Research Machine Tools Market. Global Industry Analysis 2014 - 2018 and Forecast 2019 - 2029. URL: https://www.persistencemarketresearch.com/market-research/machine-tools-market.asp

[PULS14] Puls, H.; Klocke, F.; Lung, D. Experimental investigation on friction under metal cutting conditions. In: Wear, 310. Jg., 2014, Nr. 1-2, S. 63–71.

[RASC18] Raschka, S.; Mirjalili, V. Python machine learning. Machine learning and deep learning with Python, scikit-learn, and TensorFlow. (Reihe: Expert insight. Second edition, fourth release,[fully revised and updated]. Birmingham, Mumbai: Packt Publishing, 04.09.2018 - ISBN: 9781787125933.

[RONN15] Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation, 18.05.2015

[ROSE58] ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. In: Psychological review, 65. Jg., 1958, Nr. 6, S. 386–408.

[ROY18] Roy, S.; Kumar, R.; Anurag; Panda, A.; Das, R. K. A Brief Review on Machining of Inconel 718. In: Materials Today: Proceedings, 5. Jg., 2018, Nr. 9, S. 18664–18673.

[RUDE17] Ruder, S. An overview of gradient descent optimization algorithms, 15.06.2017

[RUME86] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. In: Nature, 323. Jg., 1986, Nr. 6088, S. 533–536.

[RUSS04] Russell, S. J.; Norvig, P. Künstliche Intelligenz. Ein moderner Ansatz. (Reihe: Informatik. 2. Aufl. München: Pearson Studium, 2004 - ISBN: 978-3-8273-7089-1.

[SCHE10] Scherer, D.; Müller, A.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In: Hutchison, D. et al. (Hrsg.): Artificial Neural Networks – ICANN 2010. (Reihe: Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 92–101.

[SEIS22] SEISANZAI Tool manufacturers raise prices one after another. URL: https://seisanzai-japan.com/article/p3702/

[SHAW05] Shaw, M. C. Metal cutting principles. (Reihe: Oxford series on advanced manufacturing. 2nd ed. New York: Oxford University Press, 2005 - ISBN: 0195142063.

[SOMM10] Sommer, K.; Heinz, R.; Schöfer, J. Verschleiß metallischer Werkstoffe. Erscheinungsformen sicher beurteilen ; mit zahlreichen Tabellen. (Reihe: Praxis Werkzeugtechnik. 1. Aufl. Wiesbaden: Vieweg + Teubner, 2010 - ISBN: 9783835101265.

[SØRE48] Sørensen, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Its application to analyses of the vegetation on danish commons, V). 4. Kopenhagen: Biologiske Skrifter, 1948

[SRIV14] Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. In: J. Mach. Learn. Res., 2014. Jg., 2014, Nr. 15, S. 1929–1958.

[STAT 22a] Statista Electric Vehicles. Worldwide. URL: https://www.statista.com/outlook/mmo/electric-vehicles/worldwide

[STAT 22b] Statista European machine tool industry. statistics & facts. URL: https://www.statista.com/topics/4397/european-machine-tool-industry/

[STEP16] Stephenson, D. A.; Agapiou, J. S. Metal Cutting Theory and Practice. Third edition. Boca Raton, FL, London, New York: CRC Press, 2016 - ISBN: 9781498797672.

[SULL12] Sullivan, G. M.; Feinn, R. Using Effect Size-or Why the P Value Is Not Enough. In: Journal of graduate medical education, 4. Jg., 2012, Nr. 3, S. 279–282.

[SUN10] Sun, S.; Brandt, M.; Dargusch, M. S. Thermally enhanced machining of hard-to-machine materials—A review. In: International Journal of Machine Tools and Manufacture, 50. Jg., 2010, Nr. 8, S. 663–680.

[SURO17] Surowiec, I.; Vikström, L.; Hector, G.; Johansson, E.; Vikström, C.; Trygg, J. Generalized Subset Designs in Analytical Chemistry. In: Analytical chemistry, 89. Jg., 2017, Nr. 12, S. 6491–6497.

[TAYL06] Taylor F. W. On the art of cutting metals. In: Trans. Am. Soc. Mech. Eng., 1906. Jg., 1906, Nr. 28, S. 70–350.

[TENS22] TensorFlow Developers TensorFlow: Zenodo, 2022

[THIB17] Thibodeau, P. H.; Hendricks, R. K.; Boroditsky, L. How Linguistic Metaphor Scaffolds Reasoning. In: Trends in cognitive sciences, 21. Jg., 2017, Nr. 11, S. 852–863.

[TREI20] Treiss, A.; Walk, J.; Kühl, N. An Uncertainty-Based Human-in-the-Loop System for Industrial Tool Wear Analysis. In: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V, 12461. Jg., 2020, S. 85–100.

[ULRI01] Ulrich, H. Gesammelte Schriften. Die Unternehmung als produktives soziales System, 5). Bern: Haupt, 2001 - ISBN: 3-258-06291-9.

[VASW17] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. *Attention Is All You Need,* 12.06.2017

[VINO10] Vinod Nair; Geoffrey E. Hinton Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of ICML, 2010. Jg., 2010, Nr. 27, S. 807–814.

[VINT19] Vinther, J.; Parry, L. A. Bilateral Jaw Elements in Amiskwia sagittiformis Bridge the Morphological Gap between Gnathiferans and Chaetognaths. In: Current biology CB, 29. Jg., 2019, Nr. 5, 881-888.e1.

[WALK20] Walk, J.; Kühl, N.; Schäfer, J. Towards Leveraging End-of-Life Tools as an Asset: Value Co-Creation based on Deep Learning in the Machining Industry. In: Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020. Jg., 2020

[WANG04] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. In: IEEE Transactions on Image Processing, 13. Jg., 2004, Nr. 4, S. 600–612.

[WANG18] Wang, B.; Liu, Z. Influences of tool structure, tool material and tool wear on machined surface integrity during turning and milling of titanium and nickel alloys: a review. In: The International Journal of Advanced Manufacturing Technology, 98. Jg., 2018, Nr. 5-8, S. 1925–1975.

[WANG20] Wang, Z.; Wang, E.; Zhu, Y. Image segmentation evaluation: a survey of methods. In: Artificial Intelligence Review, 53. Jg., 2020, Nr. 8, S. 5637–5674.

[WITT19] Wittpahl, V. Künstliche Intelligenz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019 - ISBN: 978-3-662-58041-7.

[WU19] Wu, X.; Liu, Y.; Zhou, X.; Mou, A. Automatic Identification of Tool Wear Based on Convolutional Neural Network in Face Milling Process. In: Sensors (Basel, Switzerland), 19. Jg., 2019, Nr. 18

[XION10] Xiong, S. C.; Dong, L. P.; Wen, D. H. Tool Wear Image Segmentation Based on Markov Random Field Model. In: Advanced Materials Research, 102-104. Jg., 2010, S. 600–604.

[YU20] Yu, T.; Zhu, H. Hyper-Parameter Optimization: A Review of Algorithms and Applications, 12.03.2020

[ZEIL12] Zeiler, M. D. ADADELTA: An Adaptive Learning Rate Method, 22.12.2012

[ZHAN18] Zhang, Z.; Sabuncu, M. R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In: 2018

[ZHAO16] Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. *Pyramid Scene Parsing Network,* 04.12.2016

[ZHOU18] Zhou, Y.; Xue, W. Review of tool condition monitoring methods in milling processes. In: The International Journal of Advanced Manufacturing Technology, 96. Jg., 2018, Nr. 5-8, S. 2509–2523.

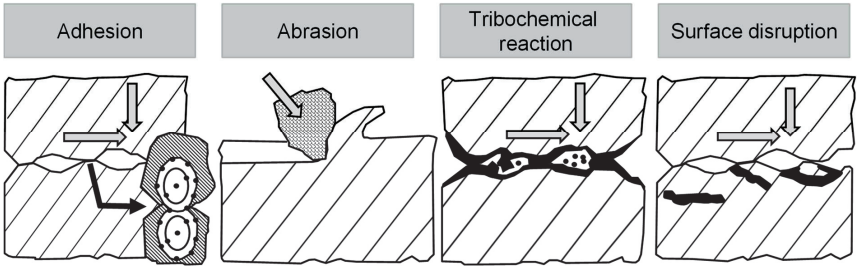
[ZHU13] Zhu, D.; Zhang, X.; Ding, H. Tool wear characteristics in machining of nickel-based superalloys. In: International Journal of Machine Tools and Manufacture, 64. Jg., 2013, S. 60–77.

A Appendix

Anhang

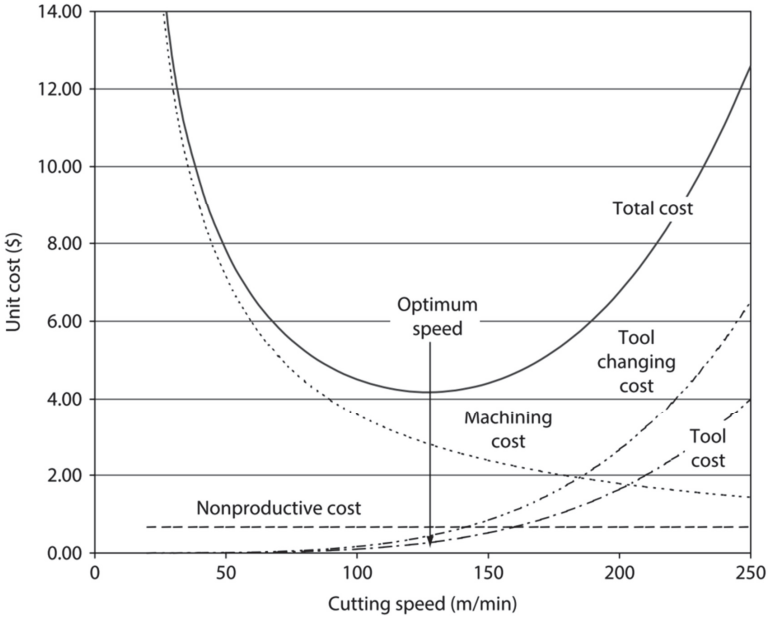
A.1 Mechanisms of wear [KLOC18, p. 75]

Verschleißmechanismen [KLOC18, S. 75]



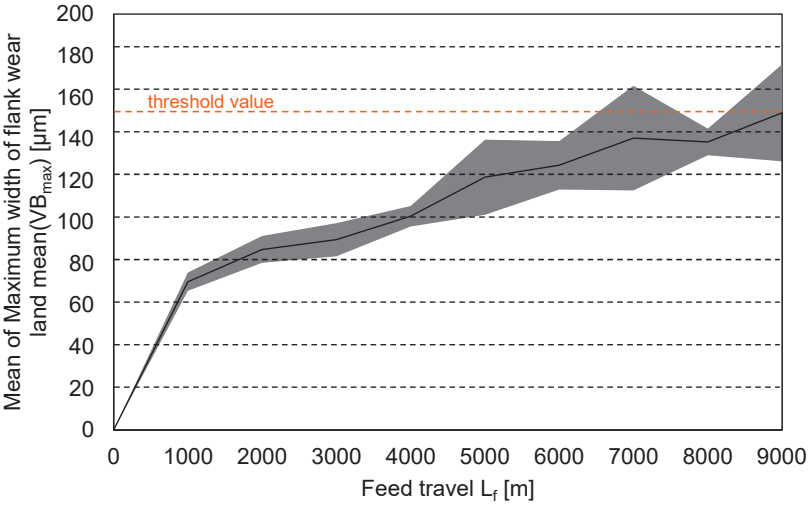
A.2 Various machining costs versus cutting speed [STEP16, p. 769]

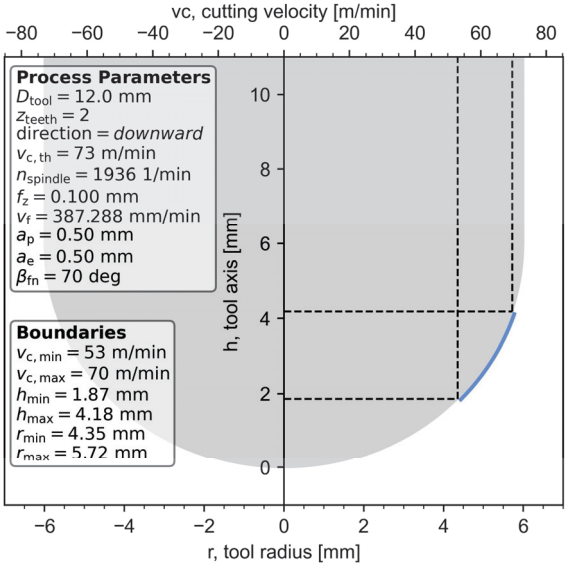
Diverse Fertigungskosten gegen Schnittgeschwindigkeit [STEP16. S. 769]



A.3 Mean VB_{max} at $v_{c,max}$ = 70 m/min with standard deviation

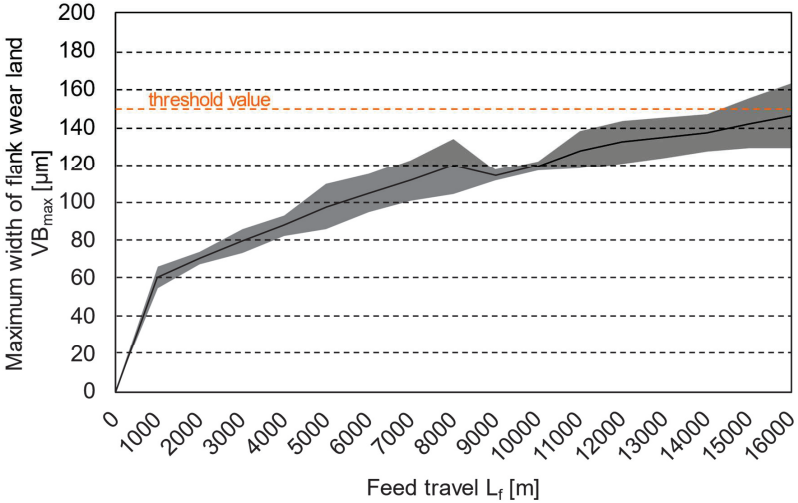
Mittlerer VB_{max} bei $v_{c,max}$ = 70 m/min mit Standardabweichung

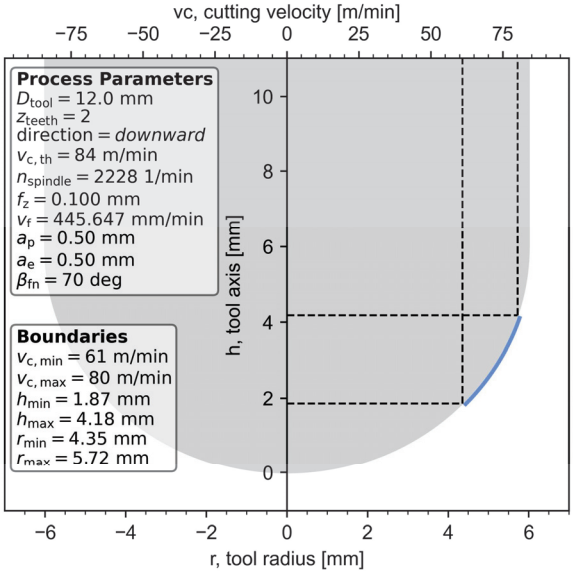




A.4 Mean VB_{max} at $v_{c,max}$ = 80 m/min with standard deviation

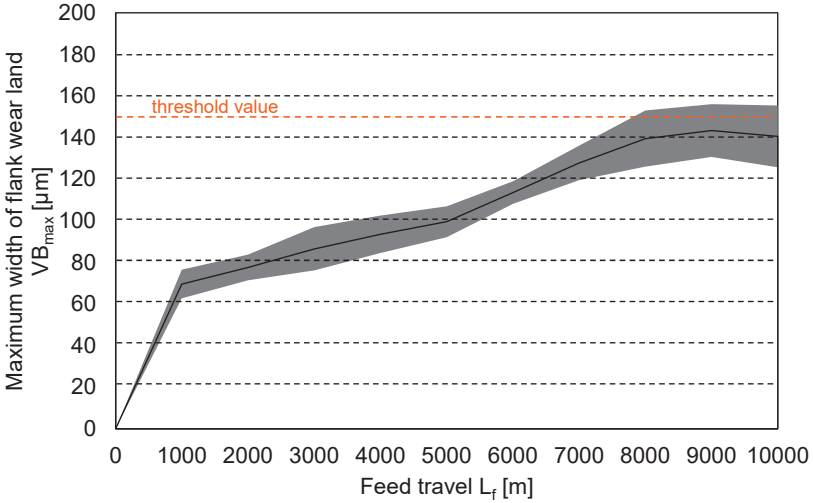
Mittlerer VB_{max} bei $v_{c,max}$ = 80 m/min mit Standardabweichung

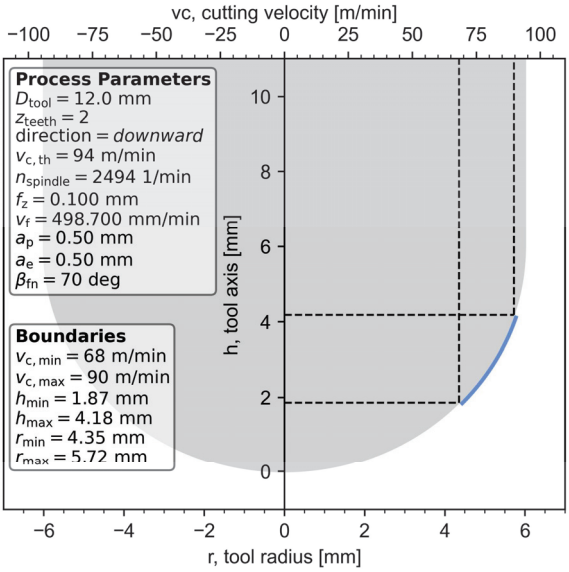




A.5 Mean VB_{max} at $v_{c,max}$ = 90 m/min with standard deviation

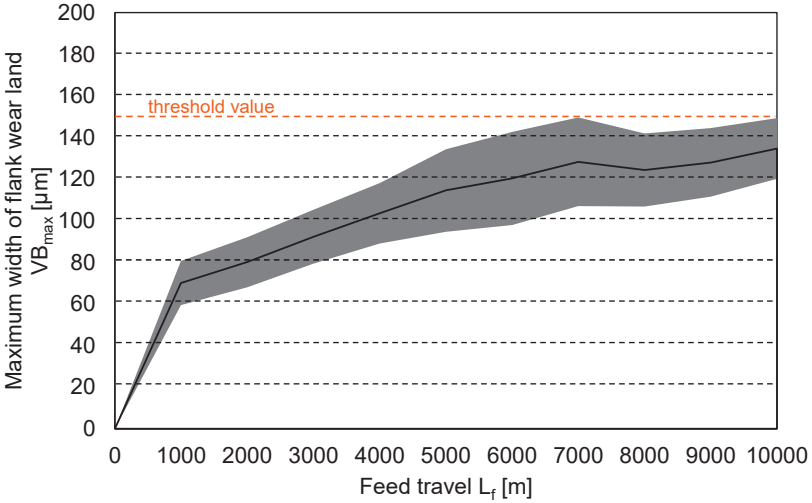
Mittlerer VB_{max} bei $v_{c,max}$ = 90 m/min mit Standardabweichung

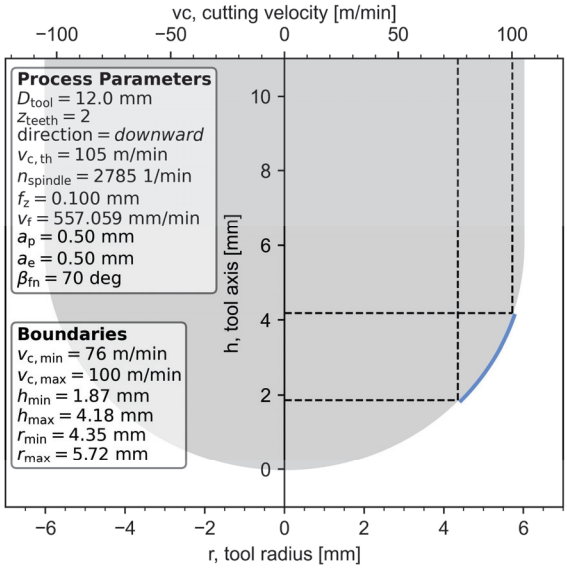




A.6 Mean VB_{max} at $v_{c,max}$ = 100 m/min with standard deviation

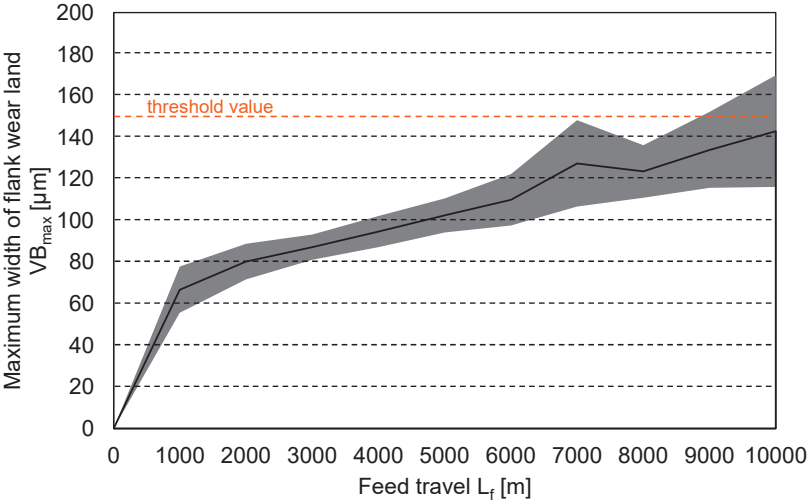
Mittlerer VB_{max} bei $v_{c,max}$ = 100 m/min mit Standardabweichung

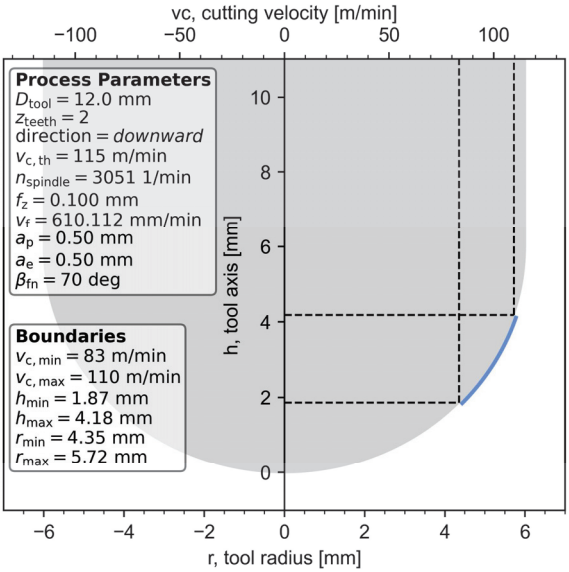




A.7 Mean VB_{max} at $v_{c,max}$ = 110 m/min with standard deviation

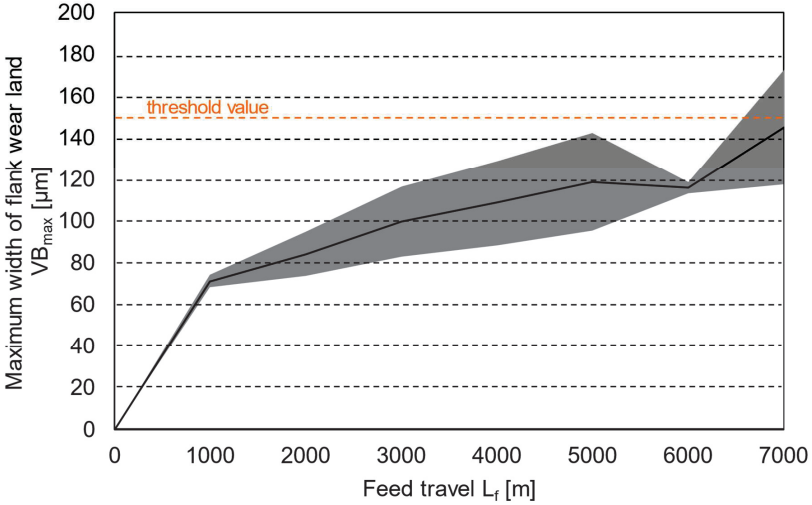
Mittlerer VB_{max} bei $v_{c,max}$ = 110 m/min mit Standardabweichung

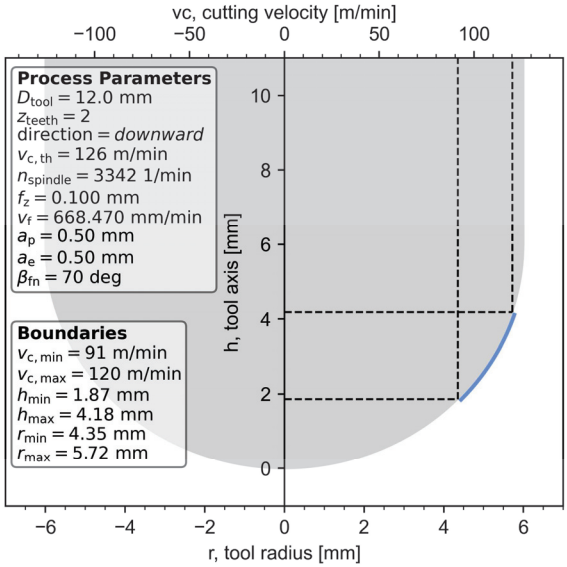




A.8 Mean VB_{max} at $v_{c,max}$ = 120 m/min with standard deviation

Mittlerer VB_{max} bei $v_{c,max}$ = 120 m/min mit Standardabweichung





A.9 U-Net Layers and their corresponding output feature map and kernel size

U-Net-Schichten und jeweilige output-feature-map- und kernelgröße

Number	Name	Output	Kernel	Parameters
1	Input	(n_{batch} x 3 x 512 x 512)	-	0
2	Normalization	(n_{batch} x 3 x 512 x 512)	-	0
3	Convolution (16)	(n_{batch} x 16 x 512 x 512)	(3,3)	448
4	Dropout	(n_{batch} x 16 x 512 x 512)	-	0
5	Convolution (16)	(n_{batch} x 16 x 512 x 512)	(3,3)	2320
6	Maxpooling	(n_{batch} x 16 x 256 x 256)	(2,2)	0
7	Convolution (32)	(n_{batch} x 32 x 256 x 256)	(3,3)	4640
8	Dropout	(n_{batch} x 32 x 256 x 256)	-	0
9	Convolution (32)	(n_{batch} x 32 x 256 x 256)	(3,3)	9248
10	Maxpooling	(n_{batch} x 32 x 128 x 128)	(2,2)	0
11	Convolution (64)	(n_{batch} x 64 x 128 x 128)	(3,3)	18496
12	Dropout	(n_{batch} x 64 x 128 x 128)	-	0
13	Convolution (64)	(n_{batch} x 64 x 128 x 128)	(3,3)	36928
14	Maxpooling	(n_{batch} x 64 x 64 x 64)	(2,2)	0
15	Convolution (128)	(n_{batch} x 128 x 64 x 64)	(3,3)	73856
16	Dropout	(n_{batch} x 128 x 64 x 64)	-	0
17	Convolution (128)	(n_{batch} x 128 x 64 x 64)	(3,3)	147584
18	Maxpooling	(n_{batch} x 128 x 32 x 32)	(2,2)	0
19	Convolution (256)	(n_{batch} x 256 x 32 x 32)	(3,3)	295168
20	Dropout	(n_{batch} x 256 x 32 x 32)	-	0
21	Convolution (256)	(n_{batch} x 256 x 32 x 32)	(3,3)	590080
22	Transposed Convolution	(n_{batch} x 256 x 64 x 64)	(2,2)	0
23	Concatenate	(n_{batch} x 384 x 64 x 64)	-	0
24	Convolution (128)	(n_{batch} x 128 x 64 x 64)	(3,3)	442496
25	Dropout	(n_{batch} x 128 x 64 x 64)	-	0
26	Convolution (128)	(n_{batch} x 128 x 64 x 64)	(3,3)	147584
27	Transposed Convolution	(n_{batch} x 128 x 128 x 128)	(2,2)	0
28	Concatenate	(n_{batch} x 192 x 128 x 128)	-	0
29	Convolution (64)	(n_{batch} x 64 x 128 x 128)	(3,3)	110656
30	Dropout	(n_{batch} x 64 x 128 x 128)	-	0
31	Convolution (64)	(n_{batch} x 64 x 128 x 128)	(3,3)	36928
32	Transposed Convolution	(n_{batch} x 64 x 256 x 256)	(2,2)	0
33	Concatenate	(n_{batch} x 96 x 256 x 256)	-	0
34	Convolution (32)	(n_{batch} x 32 x 256 x 256)	(3,3)	27680
35	Dropout	(n_{batch} x 32 x 256 x 256)	-	0
36	Convolution (32)	(n_{batch} x 32 x 256 x 256)	(3,3)	9248
37	Transposed Convolution	(n_{batch} x 32 x 512 x 512)	-	0
38	Concatenate	(n_{batch} x 64 x 512 x 512)	-	0
39	Convolution (16)	(n_{batch} x 16 x 512 x 512)	(3,3)	6928
40	Dropout	(n_{batch} x 16 x 512 x 512)	-	0
41	Convolution (16)	(n_{batch} x 16 x 512 x 512)	(3,3)	2320
42	Convolution (n_{class})	(n_{batch} x n_{class} x 512 x 512)	(1,1)	17

A.10 Inner similarity calculations

Berechnungswerte zur inneren Ähnlichkeit

dataset 1	dataset 2		rmse	psnr	ssim	sre	sam
dataset_inference_	dataset_inference_	count	465	465	465	465	465
ballend_inline_30	ballend_inline_30	mean	0.88972935	0.47771325	0.95399359	0.42467007	0.98617562
		std	0.05377662	0.15944848	0.03099421	0.18164391	0.00337348
		med	0.889328	0.43929792	0.96057582	0.39647165	0.98666074
		min	0.77645005	0.30192866	0.8692217	0.17174795	0.97690337
		max	1.00769244	1	1	1	0.9927525
		q25	0.84470635	0.37368031	0.93269104	0.31250348	0.98379074
		q50	0.889328	0.43929792	0.96057582	0.39647165	0.98666074
		q75	0.92113483	0.50349233	0.97729644	0.46679846	0.98847276
dataset_inference_	dataset_inference_	count	465	465	465	465	465
drill_edge_30	drill_edge_30	mean	0.87522871	0.44389428	0.93489268	0.21635292	0.925912
		std	0.04209193	0.15010977	0.02687879	0.21225305	0.02910328
		med	0.86997493	0.40823509	0.93393314	0.17214524	0.92792643
		min	0.77719706	0.30259231	0.86805663	-0.0436888	0.86044554
		max	1.00769244	1	1	1	0.97816814
		q25	0.85320017	0.38465857	0.91690778	0.13853895	0.90184228
		q50	0.86997493	0.40823509	0.93393314	0.17214524	0.92792643
		q75	0.88685667	0.43505921	0.9494546	0.20162181	0.95121279
dataset_inference_	dataset_inference_	count	465	465	465	465	465
endmill_30	endmill_30	mean	0.71914186	0.30724626	0.83061811	0.28216904	0.93823656
		std	0.12918548	0.2015265	0.08538391	0.22836524	0.04061002
		med	0.66047146	0.21854783	0.80147492	0.25216013	0.94918781
		min	0.50761129	0.14371794	0.68090896	0.03339864	0.7959486
		max	1.00769244	1	1	1	0.99285346
		q25	0.63749448	0.20540442	0.77435744	0.12353219	0.92104326
		q50	0.66047146	0.21854783	0.80147492	0.25216013	0.94918781
		q75	0.83566079	0.362601	0.90464162	0.31800154	0.96632932
dataset_inference_	dataset_inference_	count	465	465	465	465	465
inserts_30	inserts_30	mean	0.6369327	0.25819671	0.62718806	-0.1784416	0.7558003
		std	0.16996371	0.21115734	0.18754204	0.41289206	0.14741795
		med	0.63115351	0.20192075	0.59261603	-0.2475044	0.76316373
		min	0.36152167	0.09114657	0.15657886	-1.0402116	0.11464226
		max	1.00769244	1	1	1	0.99611757
		q25	0.48251108	0.13367236	0.47818474	-0.3916236	0.65940613
		q50	0.63115351	0.20192075	0.59261603	-0.2475044	0.76316373
		q75	0.75852776	0.28661678	0.77340371	-0.0944868	0.84557277
dataset_train_	dataset_train_	count	465	465	465	465	465
ballend_all_30	ballend_all_30	mean	0.61597521	0.24596286	0.77249871	0.24478577	0.96232917
		std	0.16219124	0.21430953	0.13102752	0.27807547	0.05203626
		med	0.57123097	0.17162878	0.74902417	0.27728016	0.98800801
		min	0.24368427	0.05678581	0.35781465	-0.3325887	0.61330966
		max	1.00769244	1	1	1	0.99614942
		q25	0.51587588	0.14713613	0.70978417	0.05448441	0.96833048
		q50	0.57123097	0.17162878	0.74902417	0.27728016	0.98800801
		q75	0.69260069	0.23846467	0.87052577	0.34584929	0.9916897
dataset_train_	dataset_train_	count	465	465	465	465	465
ballend_one_30	ballend_one_30	mean	0.8368735	0.39952784	0.9449169	0.44966828	0.97787681
		std	0.06568934	0.1672657	0.02589257	0.15833516	0.00411777
		med	0.8350611	0.36188724	0.94544891	0.41863475	0.97811204
		min	0.68880803	0.23601051	0.8732552	0.2306568	0.95761707
		max	1.00769244	1	1	1	0.98647753
		q25	0.79374622	0.31787509	0.92969422	0.37108616	0.97571231
		q50	0.8350611	0.36188724	0.94544891	0.41863475	0.97811204
		q75	0.87009097	0.408408	0.96029326	0.46969121	0.98089049

A.11 Outer similarity calculations

Berechnungswerte zur äußeren Ähnlichkeit

dataset 1	dataset 2	rmse	psnr	ssim	sre	sam	human
dataset_inference_ballend_inline_30	dataset_inference_drill_edge_30	0.54429796	0.16010599	0.64943729	0.09917383	0.98103188	0.1
dataset_inference_ballend_inline_30	dataset_inference_endmill_30	0.68746478	0.23774084	0.8137725	0.1791078	0.97639996	0.3
dataset_inference_ballend_inline_30	dataset_inference_inserts_30	0.54505959	0.16204716	0.61797453	0.10164974	0.91207687	0.1
dataset_inference_ballend_inline_30	dataset_train_ballend_all_30	0.58349122	0.18059629	0.75881589	0.12211057	0.98651265	0.5
dataset_inference_ballend_inline_30	dataset_train_ballend_one_30	0.53233962	0.15461337	0.71565997	0.09516597	0.98884637	0.85
dataset_inference_drill_edge_30	dataset_inference_endmill_30	0.46250768	0.12850224	0.57828532	-0.11460932	0.96400104	0.2
dataset_inference_drill_edge_30	dataset_inference_inserts_30	0.61275035	0.19431452	0.63587894	-0.04898652	0.88390709	0.3
dataset_inference_drill_edge_30	dataset_train_ballend_all_30	0.5101467	0.14972267	0.64370197	-0.092516	0.97112036	0.1
dataset_inference_drill_edge_30	dataset_train_ballend_one_30	0.51046466	0.14599658	0.64973342	-0.09613183	0.97145061	0.1
dataset_inference_endmill_30	dataset_inference_inserts_30	0.50396402	0.14431282	0.58205575	0.11914553	0.89915681	0.2
dataset_inference_endmill_30	dataset_train_ballend_all_30	0.54908026	0.16353541	0.70852753	0.13904188	0.96681516	0.3
dataset_inference_endmill_30	dataset_train_ballend_one_30	0.50811352	0.14443176	0.68957443	0.1194987	0.96885644	0.3
dataset_inference_inserts_30	dataset_train_ballend_all_30	0.42565226	0.121347	0.47026663	-0.33760427	0.82962615	0.1
dataset_inference_inserts_30	dataset_train_ballend_one_30	0.37797864	0.09991466	0.45181745	-0.35952952	0.83965475	0.1
dataset_train_ballend_all_30	dataset_train_ballend_one_30	0.55774398	0.17267876	0.73734092	0.1726931	0.96403425	0.5

A.12 Outer similarity values at different dataset size levels

Werte der äußeren Ähnlichkeit für die Stufen der Datensatzgröße

			SSIM	SSIM		
Training Datasets		**Size**	**mean**	**std**	**meanmean**	**meanstd**
dataset_train_ballend_all_50	dataset_train_ballend_all_50	50	0.7575	0.1322	0.77	0.131
dataset_train_ballend_all_100	dataset_train_ballend_all_100	100	0.7623	0.1445		
dataset_train_ballend_all_150	dataset_train_ballend_all_150	150	0.7824	0.1222		
dataset_train_ballend_all_400	dataset_train_ballend_all_400	400	0.7815	0.1246		
dataset_train_ballend_one_50	dataset_train_ballend_one_50	50	0.9339	0.0304	0.92	0.032
dataset_train_ballend_one_100	dataset_train_ballend_one_100	100	0.9182	0.0326		
dataset_train_ballend_one_150	dataset_train_ballend_one_150	150	0.9136	0.0326		
dataset_train_ballend_one_400	dataset_train_ballend_one_400	400	0.9121	0.0328		
dataset_train_mix_50	dataset_train_mix_50	50	0.6774	0.1529	0.67	0.170
dataset_train_mix_100	dataset_train_mix_100	100	0.658	0.1697		
dataset_train_mix_150	dataset_train_mix_150	150	0.667	0.171		
dataset_train_mix_400	dataset_train_mix_400	400	0.6727	0.1846		

A.13 Identification of the statistically significant effects in factorial experiment based on Lenth's Analysis

Identifizierung der statistisch signifikanten Effekte in einem faktoriellen Experiment auf der Grundlage der Lenth'schen Analyse

1. Calculation of the t-value for each factor:
   - Fit a regression model to the data and estimate the effect of each factor.
   - Calculate the t-value for each factor by dividing the estimated effect by its standard error.
2. Calculation of the median:
   - Take the absolute values of the t-values obtained in step 1.
   - Arrange them in ascending order and identify the middle value as the median.
3. Approximation of the standard error:
   - Multiply the median obtained in step 2 by 1.5 to approximate the standard error.
   - This approximation is often referred to as the "estimated standard error" or "median-based standard error."
4. Calculation of the threshold value:
   - Multiply the estimated standard error from step 3 by 2.5 to obtain a threshold value.
   - Exclude all factors with t-values above this threshold value to obtain a refined list of potentially significant effects.
5. Approximation of the Pseudo Standard Error (PSE):
   - Calculation of the median of the refined list obtained in step 4.
   - Multiply the median by 1.5 to approximate the PSE.
   - The PSE is used as a value for evaluating the statistical significance of individual effect estimates.
6. Calculate Degrees of Freedom (DF) and Lenth's Degrees of Freedom (DFL):
   - Calculate the total number of factors, denoted as n.
   - For each individual factor, calculate the number of levels (r) and compute n! / (r! * (n - r)!).
   - Sum all these values to obtain the Degrees of Freedom (DF).
   - Divide DF by 3 to obtain Lenth's Degrees of Freedom (DFL).
7. Find the critical t-value:
   - Determine the critical t-value corresponding to the desired confidence level and Lenth's Degrees of Freedom (DFL).
   - For example, to calculate the critical t-value for a 0.05 confidence interval with DFL = 5, you would use the t-distribution table or statistical software.
8. Convert PSE to critical value:
   - Multiply the PSE obtained in step 5 by the critical t-value from step 7.
   - This yields a critical value for determining statistically significant effects.

9. Compare t-values to the critical value:
    - Compare the t-values calculated in step 1 for each factor to the critical value obtained in step 8.
    - If the t-value for a factor is greater than the critical value, deem it statistically significant.
    - Factors with t-values below the critical value are considered not statistically significant.

A.14 ANOVA regards to accuracy

Varianzanalyse in Hinblick auf Genauigkeit

Analysis of Variance

Source	DF	cor SS	cor MS	F-Value	p-Value	F-test	Significance: F>=Fcrit		Random: p>pcrit	
Model	44	0,211780	0,004813	8,64	0.000	F(44 10)= 8.64	Fcrit(alpha=0.05)= 3.326	significant	pcrit= 0.05	**not random**
Linear	8	0,155514	0,019439	34,88	0.000	F(8 10)= 34.88	Fcrit(alpha=0.05)= 3.072	**significant**	pcrit= 0.05	**not random**
dataset_similarity_value	2	0,148668	0,074334	133,38	0.000	F(2 10)= 133.4	Fcrit(alpha=0.05)= 4.103	**significant**	pcrit= 0.05	**not random**
dataset_size_value	2	0,023174	0,011587	20,79	0.000	F(2 10)= 20.79	Fcrit(alpha=0.05)= 4.103	**significant**	pcrit= 0.05	**not random**
dropout_value	2	0,003391	0,001695	3,04	0.093	F(2 10)= 3.04	Fcrit(alpha=0.05)= 4.103	not significant	pcrit= 0.05	random
network_size_value	2	0,002086	0,001043	1,87	0.204	F(2 10)= 1.87	Fcrit(alpha=0.05)= 4.103	not significant	pcrit= 0.05	random
2-Factor-Interaction	20	0,007234	0,000362	0,65	0.804	F(20 10)= 0.65	Fcrit(alpha=0.05)= 2.774	not significant	pcrit= 0.05	random
dataset_similarity_value*dropout_value	4	0,000692	0,000173	0,31	0.865	F(4 10)= 0.31	Fcrit(alpha=0.05)= 3.478	not significant	pcrit= 0.05	random
dataset_similarity_value*network_size_value	4	0,002086	0,000521	0,94	0.485	F(4 10)= 0.94	Fcrit(alpha=0.05)= 3.478	not significant	pcrit= 0.05	random
dataset_size_value*dropout_value	4	0,001698	0,000424	0,76	0.573	F(4 10)= 0.76	Fcrit(alpha=0.05)= 3.478	not significant	pcrit= 0.05	random
dataset_size_value*network_size_value	4	0,001024	0,000256	0,46	0.764	F(4 10)= 0.46	Fcrit(alpha=0.05)= 3.478	not significant	pcrit= 0.05	random
dropout_value*network_size_value	4	0,000389	0,000097	0,17	0.946	F(4 10)= 0.17	Fcrit(alpha=0.05)= 3.478	not significant	pcrit= 0.05	random
3-Factor-Interactions	16	0,007562	0,000473	0,85	0.629					
dataset_similarity_value*dropout_value *network_size_value	8	0,002646	0,000331	0,59	0.764					
dataset_size_value*dropout_value *network_size_value	8	0,005370	0,000671	1,20	0.384					
Error	10	0,005573	0,000557							
Total	54	0,217353								

A.15 ANOVA with regards to overfitting

Varianzanalyse in Hinblick auf Überanpassung

Analysis of Variance									Significance: F>=Fcrit			Random: p>pcrit	
Source	DF	cor SS	cor MS	F-value	p-value								
Model	44	0,107483	0,002443	2,78	0.043	F(	44	10)=	2.78	Fcrit(alpha=0.05)=	3.326 not significant	pcrit=	0.05 **not random**
Linear	8	0,072814	0,009102	10,34	0.001	F(	8	10)=	10.34	Fcrit(alpha=0.05)=	3.072 **significant**	pcrit=	0.05 **not random**
dataset_similarity_value	2	0,035553	0,017777	20,20	0	F(	2	10)=	20.2	Fcrit(alpha=0.05)=	4.103 **significant**	pcrit=	0.05 **not random**
dataset_size_value	2	0,018940	0,009470	10,76	0.003	F(	2	10)=	10.76	Fcrit(alpha=0.05)=	4.103 **significant**	pcrit=	0.05 **not random**
dropout_value	2	0,017226	0,008613	9,79	0.004	F(	2	10)=	9.79	Fcrit(alpha=0.05)=	4.103 **significant**	pcrit=	0.05 **not random**
network_size_value	2	0,002786	0,001393	1,58	0.253	F(	2	10)=	1.58	Fcrit(alpha=0.05)=	4.103 not significant	pcrit=	0.05 random
2-Factor-Interaction	20	0,009091	0,000455	0,52	0.9	F(	20	10)=	0.52	Fcrit(alpha=0.05)=	2.774 not significant	pcrit=	0.05 random
dataset_similarity_value*dropout_value	4	0,003516	0,000879	1,00	0.452	F(	4	10)=	1	Fcrit(alpha=0.05)=	3.478 not significant	pcrit=	0.05 random
dataset_similarity_value*network_size_value	4	0,001966	0,000492	0,56	0.698	F(	4	10)=	0.56	Fcrit(alpha=0.05)=	3.478 not significant	pcrit=	0.05 random
dataset_size_value*dropout_value	4	0,003067	0,000767	0,87	0.514	F(	4	10)=	0.87	Fcrit(alpha=0.05)=	3.478 not significant	pcrit=	0.05 random
dataset_size_value*network_size_value	4	0,001917	0,000479	0,54	0.707	F(	4	10)=	0.54	Fcrit(alpha=0.05)=	3.478 not significant	pcrit=	0.05 random
dropout_value*network_size_value	4	0,001555	0,000389	0,44	0.776	F(	4	10)=	0.44	Fcrit(alpha=0.05)=	3.478 not significant	pcrit=	0.05 random
3-Factor-Interactions	16	0,008358	0,000522	0,59	0.83								
dataset_similarity_value*dropout_value *network_size_value	8	0,002858	0,000357	0,41	0.893								
dataset_size_value*dropout_value *network_size_value	8	0,004820	0,000603	0,68	0.698								
Error	10	0,008802	0,000880										
Total	54	0,116285											

A.16 Comparison of cutting tool wear segmentation models

Vergleich von Modellen zur Segmentation von Zerspanungswerkzeugverschleiß

Author	Usecase			Methods				
	Tool Versions	Number of Images	Image Size	Augmen -tation	Network Type	Train / Val / Test Split	Metric for Score	Test Score
[Lutz19]	1	100	48x48	-	CNN	0.8/0.1/0.1	Accuracy	0.91
[Bergs20]	8	400	512x512	BIM	FCN	0.8/0.1/0.1	mIoU	0.73
[Walk20]	1	648	640x420	-	CNN	-	MCC	0.88
[Treiss20]	1	213	1200x160	-	FCN	0.71/0.05/0.24	Dice	0.63
[Lutz20]	2	207	1280x1024	BIM	FCN	0.8/0.1/0.1	mIoU	0.69
[Miao21]	1	186	320x256	BIM	FCN	?/?/0.4	Dice	0.93
[Lutz21]	2	188	1280x1024	DLA	FCN	-	mIoU	0.72
[HOLS21]	2	80	512x512	-	FCN	0.9/0.05/0.05	Dice	0.85
[HOLS23]	1	46	512x512	-	FCN	0.8/0.1/0.1	Dice	0.85
[HOLS23]	4	1200	512x512	-	FCN	0.8/0.1/0.1	Dice	0.88

A.17 Comparison of segmentation models trained on different datasets in the application-oriented trial

Vergleich von Modellen zur Segmentation trainiert auf verschiedenen Datensätzen in der Anwendungsorientierten Erprobung