

FAU Studien aus dem Maschinenbau 379

**Tobias Sprügel**

Sphärische Detektorflächen als  
Unterstützung der Produktentwicklung  
zur Datenanalyse im Rahmen des  
Digital Engineering



Tobias Sprügel

Sphärische Detektorflächen als Unterstützung der  
Produktentwicklung zur Datenanalyse im Rahmen  
des Digital Engineering

# **FAU Studien aus dem Maschinenbau**

## **Band 379**

Herausgeber der Reihe:

Prof. Dr.-Ing. Jörg Franke

Prof. Dr.-Ing. Nico Hanenkamp

Prof. Dr.-Ing. habil. Tino Hausotte

Prof. Dr.-Ing. habil. Marion Merklein

Prof. Dr.-Ing. Michael Schmidt

Prof. Dr.-Ing. Sandro Wartzack

Tobias Sprügel

**Sphärische Detektorflächen  
als Unterstützung der Produktentwicklung  
zur Datenanalyse im Rahmen des  
Digital Engineering**

Dissertation aus dem Lehrstuhl für Konstruktionstechnik  
(KTmfk)  
Prof. Dr.-Ing. Sandro Wartzack

Erlangen  
FAU University Press  
2021

Bibliografische Information der Deutschen Nationalbibliothek:  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Bitte zitieren als

Sprügel, Tobias. 2021. *Sphärische Detektorflächen als Unterstützung der Produktentwicklung zur Datenanalyse im Rahmen des Digital Engineering*.

FAU Studien aus dem Maschinenbau Band 379.

Erlangen: FAU University Press. DOI: 10.25593/978-3-96147-476-9

Das Werk, einschließlich seiner Teile, ist urheberrechtlich geschützt.  
Die Rechte an allen Inhalten liegen bei ihren jeweiligen Autoren.  
Sie sind nutzbar unter der Creative-Commons-Lizenz BY-NC.

Der vollständige Inhalt des Buchs ist als PDF über den OPUS-Server der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:  
<https://opus4.kobv.de/opus4-fau/home>

Verlag und Auslieferung:

FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck: docupoint GmbH

ISBN: 978-3-96147-475-2 (Druckausgabe)

eISBN: 978-3-96147-476-9 (Online-Ausgabe)

ISSN: 2625-9974

DOI: 10.25593/978-3-96147-476-9

**Sphärische Detektorflächen als Unterstützung der  
Produktentwicklung zur Datenanalyse im Rahmen  
des Digital Engineering**

Der Technischen Fakultät  
der Friedrich-Alexander-Universität  
Erlangen-Nürnberg

zur  
Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von

Tobias C. Sprügel, M.Sc.

aus Nürnberg

Als Dissertation genehmigt  
von der Technischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen

Prüfung: 14.09.2021

Vorsitzender des

Promotionsorgans: Prof. Dr.-Ing. Knut Graichen

Gutachter:

Prof. Dr.-Ing. Sandro Wartzack  
Prof. Dr. Markus Zimmermann,  
TU München

## Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Konstruktionstechnik (KTmfk) der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). Die Grundlagen dieser Arbeit wurden im Rahmen der Forschungsprojekte BFS FORPRO<sup>2</sup> (Effiziente Produkt- und Prozessentwicklung durch wissenschaftsbasierte Simulation) und EU EFRE E|Asy-Opt (Kompetenz- und Analyseprojekt für die Prozess- und Produktionsoptimierung mittels Data Mining) erarbeitet.

Mein besonderer Dank gilt meinem Doktorvater Prof. Dr.-Ing. Sandro Wartack, Ordinarius des Lehrstuhls für Konstruktionstechnik (KTmfk), für die persönliche und wissenschaftliche Betreuung der Arbeit. Die mir ermöglichten Freiheiten, die Unterstützung und die zahlreichen fachlichen Diskussionen haben entscheidend zum Gelingen der Arbeit beigetragen.

Ebenso danke ich dem Zweitgutachter Herrn Prof. Dr. Markus Zimmermann, Ordinarius des Lehrstuhls für Produktentwicklung und Leichtbau an der Technischen Universität München für sein Interesse an meiner Arbeit.

Bei Prof. Dr. Björn Eskofier, Ordinarius des Lehrstuhls für Maschinelles Lernen und Datenanalytik der FAU bedanke ich mich für das „fachfremde“ Gutachten.

Darüber hinaus möchte ich mich bei Prof. Dr.-Ing. Sebastian Müller, Ordinarius des Lehrstuhls für Gießereitechnik der FAU für den Vorsitz während der mündlichen Promotionsprüfung bedanken.

Weiterhin gilt meine Dankbarkeit meinen Kollegen und Freunden am Lehrstuhl KTmfk, welche stets für eine fachliche oder auch private Diskussion zu diversen Themen bereit waren und mich so unterstützt haben. Namentlich erwähnen möchte ich hier insbesondere Prof. Dr.-Ing. Michael Walter, Dr.-Ing. Philipp Kestel, Sebastian Bickel und Dr.-Ing. Marius Fechter. Unterstützt wurde ich dabei auch durch zahlreiche studentische Arbeiten im Laufe meiner sechs Jahre am Lehrstuhl, besonders erwähnen möchte hier Sebastian Bickel, Tina Schröppel, Sebastian Schwarz, Alexander Grauf, Daniel Tiebl und Manuel Stadler.

Allen voran möchte ich mich bei meinen Eltern und Geschwistern bedanken, sowie bei meinem Großonkel - selbst Maschinenbauingenieur - der mich erst zum Maschinenbaustudium gebracht hat.

Am Ende möchte ich mich noch ganz herzlich bei meiner Frau Christin für ihre Geduld und ihr Verständnis während der Entstehungszeit dieser Arbeit bedanken.

Fürth, den 27.10.2021

Tobias C. Sprügel

# Inhaltsverzeichnis

Formelzeichen- und Abkürzungsverzeichnis .....	ix
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Problemstellung .....	2
1.2 Aufgabenstellung .....	4
<b>2 Stand der Forschung und Technik .....</b>	<b>7</b>
2.1 Maschinelles Lernen .....	8
2.1.1 Definition und Begriffserklärung .....	9
2.1.2 Regression, Klassifikation, Clustering etc. ....	10
2.2 Künstliche Neuronale Netze (KNN).....	17
2.2.1 Das biologische Vorbild .....	18
2.2.2 Funktionelle Beschreibung & Netzarchitektur .....	20
2.2.3 Lernvorgänge.....	26
2.2.4 Vor- und Nachteile .....	28
2.3 Deep Learning .....	30
2.3.1 Erklärung .....	32
2.3.2 Netzarchitektur CNN .....	34
2.3.3 Lernvorgänge (auch Transfer Learning).....	37
2.3.4 Vor- und Nachteile .....	39
2.4 Übergabe von Geometriedaten an Machine Learning Algorithmen ..	41
2.4.1 Volumetric Convolutional Neural Networks .....	41
2.4.2 Multiview Convolutional Neural Networks .....	43
2.4.3 Feature-Based Convolutional Neural Networks .....	44
2.4.4 Shape Google .....	45
2.4.5 Spektrale Convolutional Neural Networks.....	46
2.4.6 PointNet und PointNet++ .....	47
2.4.7 Kurze Betrachtung der genannten Methoden .....	50
2.5 Plausibilitätsprüfung.....	51
2.5.1 Abgrenzung «Validierung», «Verifikation» und «Plausibilität»	51
<b>3 Ableitung des Handlungsbedarfs .....</b>	<b>57</b>
<b>4 Methodik der sphärischen Detektorflächen.....</b>	<b>63</b>
4.1 Gesamtübersicht .....	63
4.2 Einheitliche Orientierung von Punktwolken .....	64
4.2.1 Methode #1: Hauptkomponentenanalyse (PCA) .....	65

4.2.2	Methode #2: Vervielfachung .....	66
4.2.3	Methode #3: Definition eines allgemeingültigen Koordinatensystems .....	67
4.3	Aufbau der sphärischen Detektorfläche .....	70
4.3.1	Einteilung Detektorpixel nach gleichen Winkeln .....	71
4.3.2	Einteilung Detektorpixel nach gleichen Flächen .....	72
4.4	Ermittlung der Detektormatrix für eine Punktwolke .....	74
<b>5</b>	<b>Anwendung #1: Bauteilerkennung .....</b>	<b>77</b>
5.1	Definition der Problemstellung.....	77
5.2	Aufbau der Datenbasis und Erstellung der Detektormatrix .....	78
5.3	KNN Architektur und Training .....	80
5.4	Ermittlung der Prognosegüte der KNNs .....	82
5.5	Automatische Erkennung unbekannter Bauteile .....	87
5.6	Ergebnisse und Bewertung .....	91
<b>6</b>	<b>Anwendung #2: Erkennung von Montagefehlern .....</b>	<b>93</b>
6.1	Definition der Problemstellung.....	93
6.2	Aufbau der Datenbasis .....	97
6.3	Erzeugung der Detektormatrizen .....	99
6.4	Bildvergleich der Detektormatrizen .....	101
6.5	Ergebnisse und Bewertung .....	103
<b>7</b>	<b>Adaption der Methodik der sphärischen Detektorflächen ...</b>	<b>107</b>
7.1	Detektormatrizen für beliebige knotengebundene Größen.....	107
7.2	Normierung der unterschiedlichen Detektormatrizen .....	111
7.2.1	Normierungsstrategie #1 .....	112
7.2.2	Normierungsstrategie #2 .....	113
7.3	Aufbau der «DNA» einer Finite-Elemente- Simulation .....	116
<b>8</b>	<b>Anwendung #3: Plausibilitätsprüfung für FE-Berechnungen .</b>	<b>119</b>
8.1	Definition der Problemstellung.....	119
8.2	Aufbau der Datenbasis .....	120
8.2.1	Bauteil Inliner-Schiene.....	122
8.2.2	Bauteil Mountainbike-Wippe.....	125
8.2.3	Bauteil Kurbelwelle R6.....	128
8.3	Ableitung der «DNA» der FE-Simulation .....	134
8.4	Architektur und Training der Convolutional Neural Networks .....	139
8.4.1	Eigene Architektur mit Matrizenanordnung hintereinander .	139
8.4.2	Eigene Architektur mit Matrizenanordnung untereinander ...	141

8.4.3	Transfer Learning.....	143
8.5	Ermittlung der Prognosegüte der CNNs.....	145
8.6	Ergebnisse .....	145
8.6.1	Eigene Architektur mit Matrizenanordnung hintereinander .	146
8.6.2	Eigene Architektur mit Matrizenanordnung untereinander...	147
8.6.3	Transfer Learning mittels AlexNet Architektur .....	148
8.7	Generalisierungsfähigkeit der trainierten CNNs .....	151
8.8	Vergleich und Bewertung der trainierten CNNs.....	156
<b>9</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>159</b>
<b>10</b>	<b>Summary and outlook.....</b>	<b>165</b>
	<b>Literaturverzeichnis.....</b>	<b>169</b>
	<b>Anhang.....</b>	<b>199</b>



# Formelzeichen- und Abkürzungsverzeichnis

## Formelzeichen

<i>Symbol</i>	<i>Einheit</i>	<i>Beschreibung</i>
$\beta$	deg	Öffnungswinkel des Ausgleichsgewichts einer Kurbelwelle
$\theta$	deg	Polarwinkel Kugelkoordinaten
$\mu$	-	arithmetischer Mittelwert
$\sigma$	-	Standardabweichung
$\varphi$	deg	Azimuthwinkel Kugelkoordinaten
$A$	mm <sup>2</sup>	Fläche
$b$	-	Biaswerte eines künstlichen Neurons
$b_{Wange}$	-	Wangenbreite Kurbelwelle
$C$	-	Regularisationskonstante des SSIM
$d_{pl}$	-	Pleuellagerdurchmesser Kurbelwelle
$I$	-	Anzahl der Eingangsgrößen bzw. Eingangsneuronen eines KNN
$koeff$	-	Rotations- und Translationsmatrix der PCA
$M'$	-	Matrix der Ausgangsdaten der PCA
$\tilde{m}$	-	ungemittelter Schwerpunkt
$M_{original}$	-	Matrix der Eingangsdaten der PCA
$\overline{M_{original}}$	-	arithmetische Mittelwert der Matrix $M_{original}$
$N$	-	Anzahl Neuronen in einer Schicht eines KNN
$n$	-	Drehzahl
$O$	-	Anzahl der Ausgangsgrößen bzw. Ausgangsneuronen eines KNN
$O_1, O_2, \dots, O_8$	-	Oktanten mit Nummerierung in Gray-Code-Ordnung
$r$	mm	Radius
$R_m$	$\frac{N}{mm^2}$	Zugfestigkeit eines Werkstoffs

<b>Symbol</b>	<b>Einheit</b>	<b>Beschreibung</b>
$r_S$	-	Radius des Massenschwerpunkts einer Kurbelwelle
$S_1$	mm	Normierungskennzahl zur Normierung von Matrizen externer Verschiebungen oder Deformationen
$S_2$	N	Normierungskennzahl zur Normierung von Matrizen angreifender Kräfte oder Drücke
$S_3$	$N\ mm$	Normierungskennzahl zur Normierung von Matrizen anliegender Momente
$S_4$	$\frac{N}{mm^2}$	Normierungskennzahl zur Normierung von Matrizen der Spannungswerte
$w$	-	Gewichtungswert der Verbindung zweier künstlicher Neuronen
$\bar{x}$	-	arithmetischer Mittelwert der Werte in x-Richtung
$\bar{y}$	-	arithmetischer Mittelwert der Werte in y-Richtung
$\bar{z}$	-	arithmetischer Mittelwert der Werte in z-Richtung

## Abkürzungen

<b>Abkürzung</b>	<b>Beschreibung</b>
ACC	Accuracy
AGD	Average Geodesic Distance
AI	Artificial Intelligence
AISI	American Iron and Steel Institute
ASME	American Society of Mechanical Engineers
CAD	Computer-aided Design
CNN	Convolutional Neural Network
CoP	Coefficient of Prognosis
corr <sub>2</sub>	zweidimensionaler Korrelationskoeffizient
CT	Computertomografie
DfX	Design for X
DIN	Deutsches Institut für Normung
DIS	Distance from Medial Surface
DOR	Diagnostic Odd Ratio
ESD	Eigen-shape Descriptor
FDR	False Discovery Rate
FE	Finite-Elemente
FN	False Negatives
FNR	False Negative Rate
FOR	False Omission Rate
FP	False Positives
FPR	False Positive Rate
FSD	Fisher-shape Descriptor
GCNN	Geodesic Convolutional Neural Network
GPU	Graphics Processing Unit / Grafikprozessor
GRU	Gated Recurrent Unit
HCI	Human Computer Interaction
HeatSD	Heat Shape Descriptor

<b>Abkürzung</b>	<b>Beschreibung</b>
HKS	Heat Kernel Signature
ISO	International Organization for Standardization
KI	Künstliche Intelligenz
KNN	Künstliches Neuronales Netz
LDA	Lineare Diskriminanz Analyse
LiDAR	Ligth Detection and Ranging
LKW	Lastkraftwagen
LR+	Positive Likelihood Ratio
LR-	Negative Likelihood Ratio
LSTM	Long Short-term Memory
LSVRC	Large Scale Visual Recognition Challenge
ML	Maschinelles Lernen / Machine Learning
NPV	Negative Predictive Value
MSE	Mean Squared Error
MTB	Mountainbike
PCA	Principal Component Analysis / Hauptkomponentenanalyse
PKW	Personenkraftwagen
PPV	Positive Predictive Value
PSNR	Peak Signal to Noise Ratio
RGB-D	Red, Green, Blue, Distance
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SC	Shape Context
SCG	Scaled Conjugate Gradient
SDF	Shape Diameter Function
SI	Spin Image
SPC	Specificity
SSIM	Structural Similarity Index
TB	Einheit Terabyte

<b>Abkürzung</b>	<b>Beschreibung</b>
TNR	True Negative Rate
TPR	True Positive Rate
VDMA	Verband Deutscher Maschinen- und Anlagenbauer
V&V	Verification and Validation



# 1 Einleitung

Die virtuelle Produktentwicklung ist mittlerweile der Standard in den meisten Bereichen des Maschinenbaus. Nach [1] unterstützt die virtuelle Produktentwicklung alle Phasen der Produktentwicklung und umfasst die frühe Phase von der Angebotsbearbeitung, über die Konzeption, die Entwicklung und Konstruktion, die Planung der Fertigungs- und Montageprozesse sowie das gemeinsame Management aller auf das Produkt und die Produktionsplanung bezogenen Informationen in digitaler Form und deren Visualisierung. Dies geht mit der Optimierung jeder einzelnen Phase des Entwicklungsprozesses einher und umfasst auch die Berücksichtigung der nachgelagerten Phasen des Entwicklungsprozesses. Durch die Design for X Philosophie (DfX) finden unterschiedliche Gerechtheiten bzgl. Montage, Ergonomie, Kosten, Normen, Recycling, Beanspruchungen oder Fertigung Berücksichtigung [2].

Die Entwicklung erfolgt zunächst rein virtuell, beispielsweise mittels CAD-Systemen und Finite-Elemente-Simulationen. Teilweise werden Produkte auch direkt virtuell freigegeben (reale Crashversuche sind für die Homologation von Omnibussen beispielsweise nicht mehr erforderlich [3]) und anschließend direkt produziert. Für den Entwicklungsprozess gibt es unterschiedlichste Vorgehensmodelle, welche in der Regel aus Deutschland stammen (vgl. Pahl/Beitz [4], Münchner Vorgehensmodell [5], VDI 2206 [6], VDI 2221 [7], Stage-gate-model [8], etc.). Jedoch wird der Bedeutung der entstehenden Daten eine untergeordnete Rolle beigemessen.

Heute produzieren nahezu alle Firmen, Privatpersonen und Behörden täglich große Mengen an Daten. Teilweise werden diese Daten zwar gespeichert, aber nicht weiter verwendet. Viele Firmen sehen Daten und die damit verbundene Gewinnung und Analyse immer noch als Grundlage für neuartige Geschäftsmodelle, sind sich aber der Bedeutung der Daten als Teil des Kerngeschäfts nicht bewusst [9]. Durch die Nutzung dieser Daten können erhebliche Fortschritte in unterschiedlichsten Bereichen erreicht werden. Dies beginnt bei der Vernetzung der Daten von unterschiedlichen privaten Smartphones, wobei die Zukunft der vollständigen Nutzung aller verfügbaren Daten bisher überhaupt nicht absehbar ist. Auch bei der Entwicklung neuer Produkte können viele Daten aus vorangegangenen Entwicklungen genutzt werden. Die Nutzung von Daten direkter Vorgängerprodukte ist hierbei offensichtlich als sehr sinnvoll zu bewerten, aber auch die Daten der Entwicklung von anderen Produkten können für die Entwicklung eines neuen Produktes relevant sein. Eine durchgängige Nutzung

aller verfügbarer Daten bietet erhebliches Potential zur Verkürzung von Entwicklungszeiten und zur Erreichung einer schnelleren Time-To-Market.

Dies kann aber nur gelingen, wenn die relevanten Daten kontextsensitiv bereitgestellt werden können und das benötigte Wissen überhaupt erreichbar und auffindbar ist. Um große Datenmengen zu nutzen, kommen Algorithmen des Maschinellen Lernen zum Einsatz. Beispiele für diese Algorithmen sind Regressionsmodelle (linear, quadratisch, kubisch, quartisch, etc.), Künstliche Neuronale Netze oder Support Vector Machines.

In der Produktentwicklung sind viele Daten und Informationen direkt mit bestimmten Geometrien verknüpft (Ergebnisse von Finite Elemente Simulationen, Messergebnisse von produzierten Bauteilen, Formelemente an CAD-Bauteilen, uvm.). Bisher existieren jedoch nur sehr wenige Ansätze, um beliebige Geometrien an identische Algorithmen des Maschinellen Lernen zu übergeben. So können CAD-Modelle von unterschiedlichen Normteilen (in unterschiedlichen Größenausprägungen) nicht an das gleiche Neuronale Netz ohne eine entsprechende Vorverarbeitung übergeben werden. Trotz einiger Ansätze für Geometriebeschreibungen aus dem Bereich der Computer Vision, existieren für Geometriedaten aus der Produktentwicklung bzw. dem Sektor des Maschinen- und Anlagenbaus bisher keine derartigen Ansätze oder Vorverarbeitungsschritte. Der wesentliche Unterschied von Geometriedaten aus der Produktentwicklung gegenüber Geometrien aus der Computer Vision liegt in der hohen Relevanz von sehr kleinen geometrischen Details.

## 1.1 Problemstellung

Umso mehr verwundert es, dass der Datennutzung im Bereich der Produktentwicklung bisher relativ wenig Aufmerksamkeit geschenkt wird – obgleich Themen wie «Digitalisierung», «Industrie 4.0», «Digitale Zwillinge» und «Maschinelles Lernen» in aller Munde sind. Jenseits dieser Bestrebungen erfordert die ganzheitliche Datennutzung im Rahmen der Produktentwicklung, Methoden um beliebige Geometriedaten mittels Algorithmen des Maschinellen Lernens verarbeiten zu können. Hierfür stehen bisher so gut wie keine Ansätze zur Verfügung und der Fokus der wenigen bestehenden Ansätze ist keineswegs die virtuelle Produktentwicklung bzw. das Digital Engineering im Maschinen- und Anlagenbau. Ohne die Integration derartiger Methoden droht die deutsche Wirtschaft Spitzenpositionen bzgl. der Entwicklung neuer, effizienter und fortschrittlicher Produkte zu

verlieren. Aus Sicht des Autors lassen sich hierfür vier wesentliche Punkte anführen:

- Die Umstellung der bisherigen Entwicklungsprozesse zur besseren Nutzung von Daten aus vorangegangenen Entwicklungen verläuft sehr schleppend. Teilweise werden Produktdaten vornehmlich für Fehleranalysen und zur Problembeseitigung eingesetzt [10].
- Die Möglichkeiten, welche sich durch eine intelligente Datennutzung ergeben, sind vielen Entwicklungsingenieuren und dem übergeordneten Management nicht bekannt und können daher auch nicht angewendet werden. Es ist eine Wandlung von bestimmten Rollen in der Produktentwicklung zwingend notwendig [P27].
- Ansätze, um beliebige Geometrien aus dem Bereich der Produktentwicklung an Algorithmen des Maschinellen Lernen zu übergeben sind zum gegenwärtigen Zeitpunkt nicht verfügbar.
- Es existieren bisher wenige Beispiele der Integration von Methoden des Data-Minings oder Maschinellen Lernen in Entwicklungsprozesse mit dem Fokus auf Geometrie-basierte Daten.

Zur Freisetzung des vollen Potentials der Datennutzung in modernen Entwicklungsprozessen gilt es, Methoden zu entwickeln, welche die genannten Punkte adressieren. Hierdurch wird es möglich, bestehende Datenbestände aus vorangegangenen Entwicklungen von ähnlichen und weniger ähnlichen Produkten für die Entwicklung von neuen Produkten zu nutzen. Eine durchgängigere Nutzung bietet erhebliches Potential, um neue Entwicklungen schneller und mit weniger unnötigen Iterationen im Produktentwicklungsprozess zu realisieren.

Bestehende Methoden des Maschinellen Lernens sind bereits sehr weit entwickelt und es existieren zahlreiche kommerzielle und open-source Software Umgebungen und Werkzeuge, um Datenanalysen durchzuführen bzw. Datenbestände in Metamodellen (engl. surrogate models) abzubilden. Damit diese performanten Werkzeuge Einzug in die Entwicklungsprozesse halten können, müssen einerseits effiziente Vorverarbeitungsschritte verfügbar sein und es müssen beispielhafte Anwendungen mit exakter Beschreibung des verwendeten Vorgehens verfügbar sein. Denn nur so können datengetriebene Methoden Einzug in die Entwicklungsprozesse der Industrie und Wissenschaft halten. Die vorliegende Arbeit adressiert beide Punkte.

Zum gegenwärtigen Zeitpunkt gibt es wenige klassische Produktentwicklungsingenieure im Sektor des Maschinen- und Anlagenbaus (Maschinenbau-, Elektro-, Wirtschaftsingenieure oder Mechatroniker), welche

einerseits über eine fundierte Ausbildung im Bereich Produktentwicklung und andererseits über Erfahrung mit Algorithmen des Maschinellen Lernen verfügen. Erfahrene Machine Learning Experten (häufig Informatiker) verfügen über wenig Erfahrung im Umgang mit CAD- (Computer Aided Design), FE-Systemen (Finite-Elemente) oder dem methodischen Vorgehen der Produktentwicklung bzw. -entstehung. Laut einer VDMA Umfrage unter ca. 70 Unternehmen im Jahr 2019 [11] waren fehlende Personalressourcen, fehlendes qualifiziertes Datenmaterial und fehlendes qualifiziertes Personal die Hauptgründe, warum in den Unternehmen kein Maschinelles Lernen eingesetzt wird.

## 1.2 Aufgabenstellung

Ziel der vorliegenden Arbeit ist es, einen möglichen Lösungsweg aufzuzeigen, um geometriebasierte Daten aus der Produktentwicklung an Algorithmen des Maschinellen Lernen zu übergeben.

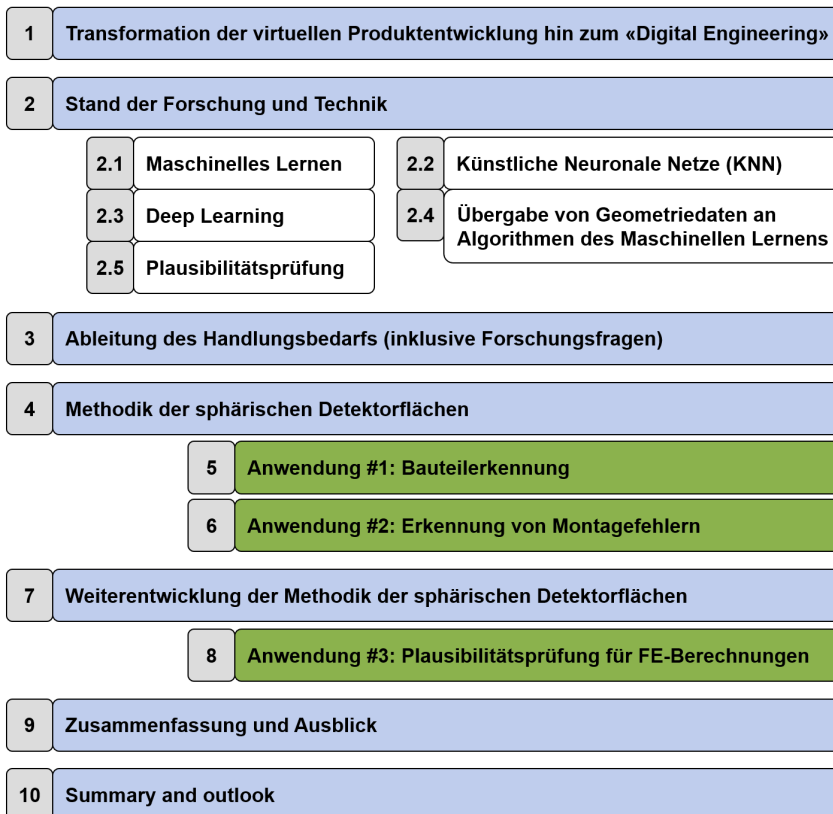


Bild 1: Aufbau der vorliegenden Dissertation

Hierfür soll zunächst auf den aktuellen Stand der Forschung und Technik bezüglich des Maschinellen Lernens und zwei sehr prominente Vertreter (Künstliche Neuronale Netze und Deep Learning) eingegangen werden. Darüber hinaus werden die Übergabe von Geometriedaten an diese Algorithmen und das Themenfeld der Plausibilitätsprüfung näher beleuchtet.

Auf Basis dieses Wissens kann der Handlungsbedarf abgeleitet werden und es wird die Methodik der sphärischen Detektorflächen im Detail erklärt. Anhand von zwei Anwendungsfällen wird die praktische Anwendbarkeit dargelegt und darauf aufbauend die Methodik weiterentwickelt. Dies ermöglicht die Anwendung der Methode der sphärischen Detektorflächen auf das Themenfeld der Plausibilitätsprüfung strukturmechanischer Finite-Elemente-Simulationen in einem dritten Anwendungsfall. Der detaillierte Aufbau der Arbeit ist in Bild 1 dargestellt. Die vorliegende Arbeit liefert einerseits eine neuartige Methodik für unterschiedliche Anwendungsfälle aus dem Bereich des Digital Engineering und legt andererseits die Methodik und die zusätzlich anzuwendenden Algorithmen (insbesondere Künstliche Neuronale Netze/Convolutional Neural Networks) verständlich für Produktentwickler und Produktentwicklerinnen dar. Die Arbeit zeigt, was mit aktuellen Forschungsansätzen und kommerziell verfügbaren Machine Learning Frameworks möglich ist und welche Anforderungen an die Datenbasis daraus hervorgehen.



## 2 Stand der Forschung und Technik

«In God we trust; all others must bring data»

*W. Edwards Deming*

«2019 wird der mobile Datenverkehr in Deutschland monatlich ein Volumen von 259,8 Petabyte erreichen - das entspricht etwa 272 Millionen Gigabyte.» [12]

«Die Datenmenge, die im Jahr 2020 weltweit erstellt, vervielfältigt und konsumiert wird, wird auf etwa 40 Zettabytes geschätzt.» [12]

Diese Zahl wird durch [13] bereits mit ca. 50 Zettabyte prognostiziert und 2025 soll eine Menge von 175 Zettabyte erreicht werden (siehe Bild 2). Bereits hier lässt sich ersehen, dass die exakte zukünftige Entwicklung schwer zu erfassen ist, aber dass der Trend ganz klar auf eine deutliche Zunahme der global verfügbaren Daten hinweist.

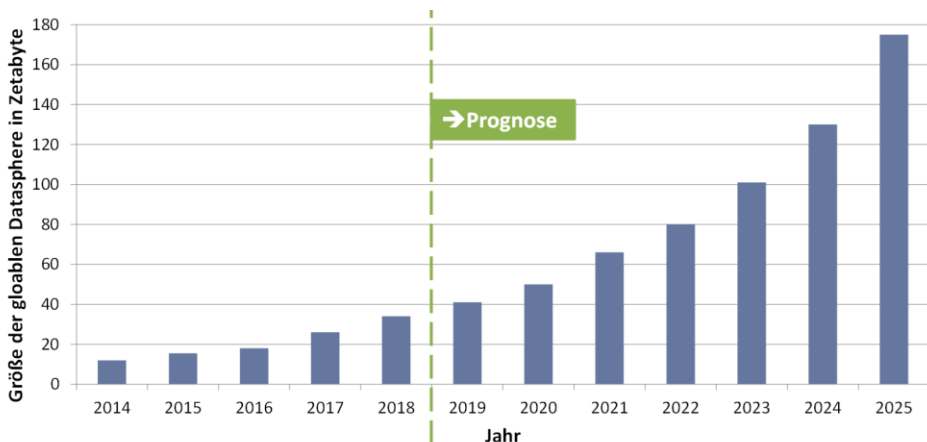


Bild 2: Aufsummierte Daten weltweit (globale Datensphäre) mit Prognose für 2019 bis 2025 in Zettabyte nach [13]

Die Daten entstehen in unterschiedlichen Bereichen und Branchen. Bild 3 zeigt die Aufteilung der Datensphäre der Unternehmen nach unterschiedlichen Branchen. Hierbei fällt auf, dass ein Großteil der Unternehmensdaten auf den Bereich der produzierenden Industrie entfällt (3.584 Exabyte). In den beschriebenen Daten aus der produzierenden Industrie sind ebenfalls Daten aus dem Bereich der Produktentwicklung enthalten, dies umfasst beispielsweise die Ergebnisse von Finite-Elemente-Simulationen des Produktes oder von den zur Herstellung benötigten Prozessen.

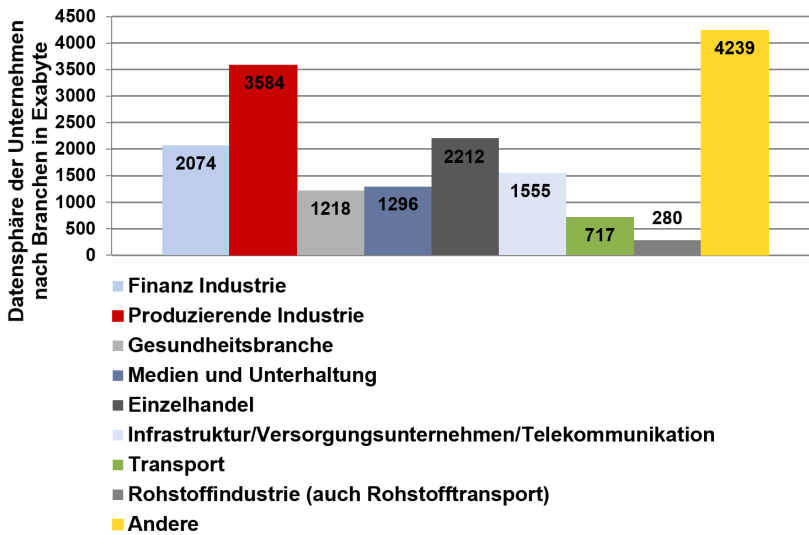


Bild 3: Datensphäre der Unternehmen nach Branchen in Exabyte (Jahr 2018) nach [13]

Daten und deren kontextsensitive Bereitstellung spielen hierbei eine entscheidende Rolle. Durch die reine Aggregation dieser Daten kann jedoch kein Nutzen für neue Produkte oder Dienstleistungen abgeleitet werden. Daher sind die Auswertung, Bündelung und Bereitstellung von Wissen aus Daten eminent wichtig. Im Folgenden soll deshalb kurz auf die Begriffe «Data-Mining» und «Maschinelles Lernen» eingegangen werden. Anschließend werden einige Vertreter aus dem Bereich des Maschinellen Lernen im Detail erläutert. Nach dem aktuellen Stand zur Übergabe von Geometriedaten an Algorithmen des Maschinellen Lernens erfolgt die Abgrenzung der Begriffe «Validierung», «Verifikation» und «Plausibilität». Dies bildet die Grundlage zur Ableitung des Handlungsbedarfs und den Hauptteil der vorliegenden Arbeit.

## 2.1 Maschinelles Lernen

Maschinelles Lernen (Machine Learning, ML) ist ein sehr schnell wachsendes technisches Gebiet an der Schnittstelle zwischen Informatik und Statistik [14]. Die Schwierigkeiten bei der Begriffsdefinition «Machine Learning» werden in Kapitel 2.1.1 aufgezeigt. Anschließend erfolgt in Kapitel 2.1.2 eine kurze Abgrenzung der Begriffe «Regression», «Klassifikation» und «Clustering» als typische Unterkategorien von Algorithmen des Maschinellen Lernens. In Kapitel 2.2 werden Künstliche Neuronale Netze und in Kapitel 2.3 Convolutional Neural Networks als Vertreter des Deep Learning vorgestellt.

### 2.1.1 Definition und Begriffserklärung

Eine klare und eindeutige Abgrenzung der Begriffe ist äußerst schwierig, da zahlreiche Autoren die Begriffe «Data-Mining» und «Machine Learning» als Synonyme verwenden oder die Begriffe gegensätzlich definieren. Anhand einiger Beispiele soll die uneindeutige Definition kurz verdeutlicht werden:

- «Data-Mining» beschreibt die automatisierte oder unterstützende Suche durch einen Computer in elektronisch gespeicherten Daten. [15]
- «Data-Mining» wird als die Suche von Mustern in Datenbeständen verstanden, wobei der Extraktionsprozess automatisch oder automatisiert zu erfolgen hat. [15]
- Die Anwendung von Methoden des «Machine Learning» auf eine große Datenbasis wird als «Data-Mining» bezeichnet. [16]
- Der Begriff «Data-Mining» ist überladen, manchmal bezieht er sich auf den gesamten Prozess der Wissensakquisition und manchmal auf eine bestimmte Phase des «Machine Learning». [17]
- Die exakte Nachbildung des kompletten Prozesses einer Aufgabenstellung ist nicht möglich. Aber die Erkennung von bestimmten Mustern und Zusammenhängen ist möglich, dies ist die Nische des «Machine Learning». [16]
- «Machine Learning» ist die wissenschaftliche Anwendung von Algorithmen und statistischen Modellen, mit denen Computersysteme eine bestimmte Aufgabe effektiv erfüllen, ohne explizite Anweisungen zu verwenden, wobei sie sich stattdessen auf Muster und Inferenz verlassen. [18, 19]
- «Machine Learning» hat die Absicht einem Computer bestimmte Aufgaben anzutrainieren und täglich auftretende Probleme von Experten zu lösen (beispielsweise die automatische Erkennung von Buchstaben). [20]
- «Machine Learning» ist das Feld wissenschaftlicher Studien mit dem Fokus auf Induktionsalgorithmen und andere Algorithmen, welche als «lernend» bezeichnet werden können. [17]
- «Machine Learning» widmet sich der Fragestellung, wie man Computer bauen kann, welche sich automatisch durch Erfahrung verbessern. [14]

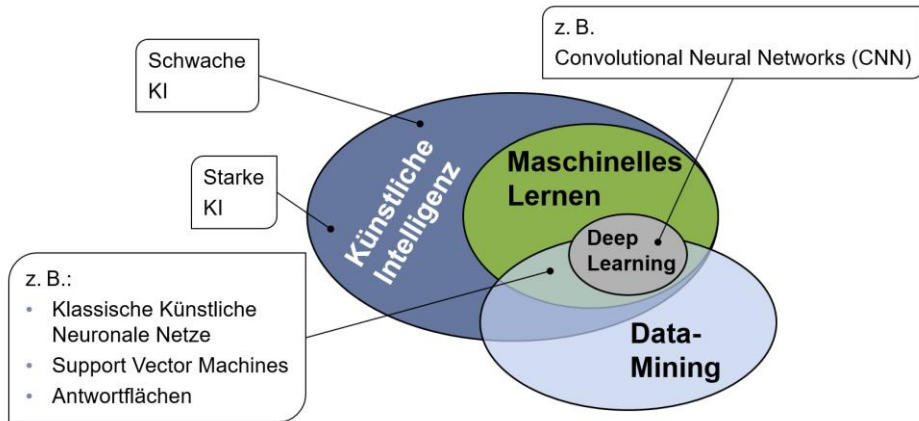


Bild 4: Visualisierung der Überschneidungen der Begriffe «Machine Learning», «Künstliche Intelligenz» (KI), «Data-Mining» und «Deep Learning» in Anlehnung an [21]

Die grafische Darstellung in Bild 4 verdeutlicht die Schwierigkeit der eindeutigen Definition. Es existieren Teilbereiche, in denen die Begriffe «Machine Learning» und «Data-Mining» völlig synonym verwendet werden können. Problemstellungen aus dem Industrieumfeld des Maschinenbaus sind oftmals genau in diesem Zwischenbereich anzutreffen. Darüber hinaus wird auch der Begriff «Künstliche Intelligenz» häufig mit Künstlichen Neuronalen Netzen gleichgesetzt, dies ist wie in Bild 4 gezeigt keinesfalls zulässig. Searle führte in [22] die Begriffe «starke KI» und «schwache KI» ein. «Starke KI» besitzt einen Verstand und ist in der Lage, mindestens so leistungsfähig zu denken wie ein Mensch. Eine «starke KI» stellt für viele Forscher das Ziel auf dem Gebiet der künstlichen Intelligenz dar. Nach [22] ist eine «starke KI» mit Computern nicht zu erreichen. Schwache KI sind Programme und Algorithmen, die menschliche geistige Aktivitäten unterstützen, anstatt sie zu duplizieren [23].

Im folgenden Teil dieser Arbeit soll der Begriff «Machine Learning» verwendet werden, mit dem Hinweis, dass an vielen Stellen auch der Begriff «Data-Mining» als Synonym verwendet werden könnte.

### 2.1.2 Regression, Klassifikation, Clustering etc.

Eine Einteilung der verwendeten Algorithmen kann, ähnlich wie bei der Einsortierung im Rahmen des Begriffs «Data-Mining», nach Methodenarten erfolgen.

Hierbei kann zwischen Anomalieerkennung, Lernen von Assoziationsregeln, Clustern, Klassifikation, Regression und zusammenfassenden Methoden unterschieden werden [24]:

- Methoden der Klassifikation versuchen, Datensätze in bestimmte, vordefinierte Klassen einzusortieren. [25, 26]
- Methoden der Regression versuchen, eine mathematische Beschreibung aufzubauen, welche das Verhalten einer Ausgangsgröße bestmöglich prognostizieren kann [27]. Dies stellt die am häufigsten auftretende Aufgabenstellung der Datenanalyse dar [28].
- Methoden des Clustering beschreiben die unüberwachte Einsortierung von Mustern in eine vordefinierte Anzahl an Clustern/Gruppen. [29]
- Anomalieerkennung beschreibt Methoden, welche Muster in Datenbeständen finden, die nicht zu dem erwarteten Verhalten der anderen Daten passen. [30]
- Lernen von Assoziationsregeln beschreibt Methoden, welche qualitative Regeln aus den Daten ableiten. Hierbei handelt es sich nicht um Klassifikationsregeln. Es werden keine Klassen vordefiniert, sondern das Ziel ist es, die Abhängigkeiten von Bestandteilen der zugrundeliegenden Datensätze zu erfassen. [31]
- Zusammenfassende Methoden haben das Ziel, Schlüsselinhalt aus einer oder mehreren Informationsquellen zu abstrahieren. Es wird zwischen «knowledge-poor» (ohne Berücksichtigung von weiteren Regeln) und «knowledge-rich» (inklusive der Berücksichtigung von Regeln, z. B. in Form von anpassbaren Wissenbasen mit Regeln) unterschieden. [32]

Eine weitere Unterscheidung kann ebenso nach der Notwendigkeit von Ein- und/oder Ausgangsdaten erfolgen. Methoden der Anomalieerkennung und der zusammenfassenden Methoden können entweder separat auf die Eingangs-, Ausgangsdaten oder auf den gesamten Datensatz (bestehend aus Ein- und Ausgangsdaten) angewendet werden. Methoden der Klassifikation oder der Regression benötigen immer Ein- und Ausgangsdaten, um das Modell zu erzeugen (supervised learning). Methoden des Clustering benötigen hierfür lediglich Eingangsdaten (unsupervised learning). Dieser Zusammenhang ist in Bild 5 dargestellt.

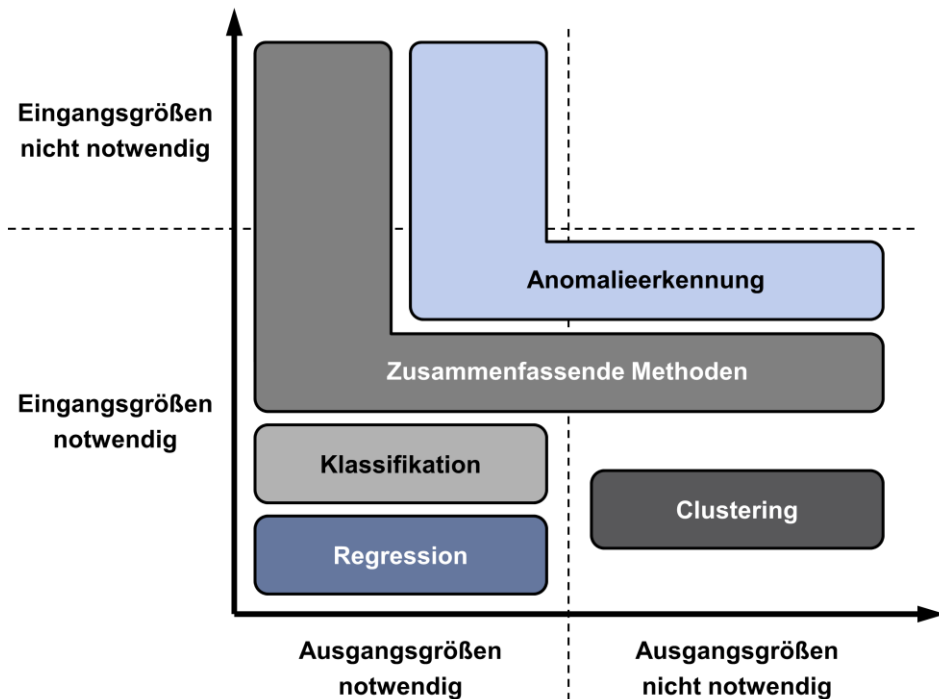


Bild 5: Einteilung der Methodenarten nach den notwendigen Ein- und Ausgangsdaten

Bevor ein Machine Learning Algorithmus verwendet werden kann, muss der in Bild 6 dargestellte Prozess für Regressions- oder Klassifikationsalgorithmen durchlaufen werden. Der Ausgangspunkt ist stets ein Datenbestand, welcher sowohl Eingangs- als auch Ausgangsgrößen enthält. Dieser Datenbestand wird nach einem bestimmten Verteilungsschlüssel auf Trainings-, Validierungs- und Testdaten aufgeteilt. Ein Validierungsdatensatz ist nicht immer erforderlich, bei Künstlichen Neuronalen Netzen (vgl. Kapitel 2.2) kommt dieser aber häufig zum Einsatz. Für das Aufteilungsverhältnis existieren verschiedene Empfehlungen in der Literatur.

Das Ziel der Aufteilung der Datensätze liegt darin begründet, dass eine gewisse Menge an Daten für das Anlernen eines derartigen Algorithmus notwendig ist. Wenn alle verfügbaren Daten für das Training verwendet werden würden, könnte keine Aussage darüber getroffen werden wie gut das Modell bei der Prognose von unbekanntem Daten funktioniert. Aus diesem Grund wird nur ein bestimmter Teil der Daten für das Training verwendet und der Rest fungiert als Test. Zur Aufteilung zwischen Trainings- und Testdaten empfiehlt [33], mehr als die Hälfte der Daten für das Training zu verwenden. Beispielsweise mittels einer 80 - 20 Aufteilung in Prozent. Für

die Aufteilung zwischen Trainings-, Test- und Validierungsdaten verwendet [34] eine 60 - 20 - 20 Aufteilung, diese ist auch in Bild 6 zu finden.

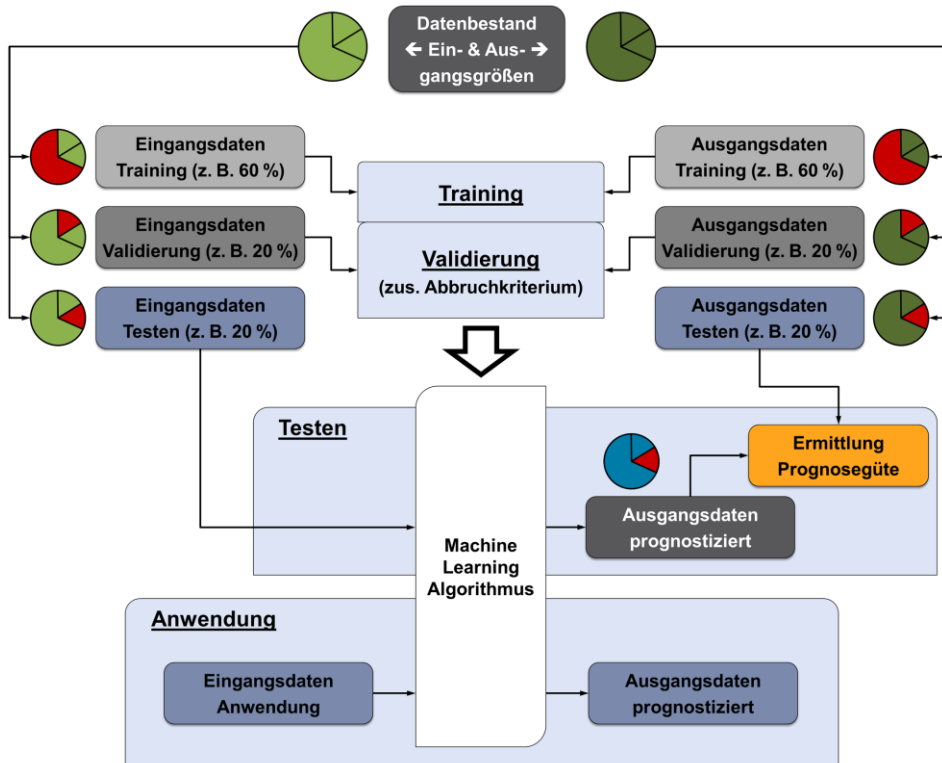


Bild 6: Aufteilung in Trainings-, Validierungs-, Testdaten und Anwendung eines Machine Learning Algorithmus (anwendbar für Regression und Klassifikation)

Die Ein- und Ausgangsdaten des Trainings werden genutzt, um ein Meta-modell / Surrogate Model / einen Algorithmus des Maschinellen Lernens anzulernen [35] (ein Beispiel für ein derartiges Modell sind Künstliche Neuronale Netze). Einige Machine Learning Algorithmen benötigen ausschließlich Training- und Testdaten ohne separate Validierungsdaten. Manche Algorithmen nutzen die Validierungsdatensätze als Abbruchkriterium für den Optimierer, welcher die Modelle an den Datensatz anpasst und darauf trainiert [36]. Convolutional Neural Networks (vgl. Kapitel 2.3) oder Künstliche Neuronale Netze (vgl. Kapitel 2.2) verwenden häufig einen zusätzlichen Validierungsdatensatz als Abbruchkriterium des Optimierers, welcher die Gewichtungen des Netzwerks im Rahmen des Trainings adaptiert. Mit Abschluss des Trainings steht ein funktionsfähiger Machine Learning Algorithmus zur Verfügung, allerdings kann bisher keine Prognosegüte für unbekannte Daten berechnet werden. Dies erfolgt im Rahmen

des Teilschritts «Testen». Hierbei wird ein unbekannter Datensatz aus Eingangsdaten an den Algorithmus übergeben und Ausgangswerte werden prognostiziert [37]. Da für die Eingangsdaten des Testens ebenfalls die tatsächlichen Ausgangsdaten existieren, können die prognostizierten Ausgangsdaten mit den Ausgangsdaten des Testdatensatzes verglichen werden. Bei einem sehr gut trainierten Algorithmus tritt keine oder nur eine sehr geringe Abweichung zwischen beiden Datensätzen auf. Für die Prognosegüte gibt es unterschiedliche Kennwerte, welche verwendet werden können, um die Übereinstimmung der Datensätze in Kennziffern (goodness-of-fit-Parameter [38]) auszudrücken. Die nachfolgende Auflistung stellt einen Auszug von möglichen Kennziffern für «Regression» und «Klassifikation» vor:

- Regression:
  - Mean Squared Error (MSE) [39]
  - Root-Mean-Square Error (RMSE) [40]
  - Coefficient of Prognosis (CoP) [41]
- Klassifikation [39, 42, 43]:
  - True positive rate (TPR), Recall, Sensitivity
  - False negative rate (FNR)
  - False positive rate (FPR)
  - True negative rate (TNR), Specificity (SPC), Selectivity
  - Positive predictive value (PPV), Precision
  - False discovery rate (FDR)
  - False omission rate (FOR)
  - Negative predictive value (NPV)
  - Prevalence
  - Accuracy (ACC)
  - Positive likelihood ratio (LR+)
  - Negative likelihood ratio (LR-)
  - Diagnostic odds ratio (DOR)
  - F1 score

Die Kennziffern der Klassifikation können alle aus der sog. Confusion-Matrix mit den Kennwerten True Positives (TP), False Negatives (FN), False Positives (FP) und True Negatives (TN) berechnet werden, dies verdeutlicht Bild 7: Confusion-Matrix mit berechenbaren Kennziffern nach [39, 42, 43, 44]. Richtig klassifizierte Datensätze sind entweder True Positives oder True Negatives. Die anderen beiden Optionen False Negatives und False Positives entsprechen Fehlklassifikationen und werden in der Statistik auch als Fehler 1. Art ( $\alpha$ -Fehler) und als Fehler 2. Art ( $\beta$ -Fehler) bezeichnet [44].

		Wahrer Zustand		Prävalenz $\frac{TP + FN}{TP + FN + FP + TN}$	Accuracy (ACC) $\frac{TP + TN}{TP + FN + FP + TN}$
		Zustand positiv	Zustand negativ		
prognostizierter Zustand	progn. Zustand positiv	True Positives TP	False Positives FP	Precision (PRC) $\frac{TP}{TP + FP}$	False Discovery Rate (FDR) $\frac{FP}{TP + FP}$
	progn. Zustand negativ	False Negatives FN	True Negatives TN	False Omission Rate (FOR) $\frac{FN}{FN + TN}$	Negative Predictive Value (NPV) $\frac{TN}{FN + TN}$
		True Positive Rate (TPR) $\frac{TP}{TP + FN}$	False Positive Rate (FPR) $\frac{FP}{FP + TN}$	Positive likelihood ratio (LR+) $\frac{TPR}{FPR}$	Diagnostic odds ratio (DOR) $\frac{LR+}{LR-}$
		False Negative Rate (FNR) $\frac{FN}{TP + FN}$	True Negative Rate (TNR) $\frac{TN}{FP + TN}$	Negative likelihood ratio (LR-) $\frac{FNR}{TNR}$	
				F1 score $\frac{PRC \cdot TPR}{2 \cdot PRC + TPR}$	

Bild 7: Confusion-Matrix mit berechenbaren Kennziffern nach [39, 42, 43, 44]

Die nachfolgende Tabelle 1 zeigt für jede der genannten Methodenarten anwendbare Algorithmen (die Auflistung dient nur zur Anschauung und ist daher keinesfalls vollständig) und eine kurze Nennung einer Anwendung für jeden Algorithmus.

Tabelle 1: Methodenarten mit Machine Learning Algorithmus und Anwendungen

Methodenart	beispielhafte Algorithmen des Masch. Lernens	Anwendungsbeispiel
Klassifikation	Support Vektor Machine	Erkennung von nicht verdeckten Frontalansichten menschlicher Gesichter in Graustufen-Bildern [46]
	Convolutional Neural Network	Klassifikation von 1,2 Millionen Bildern des ImageNet LSVRC-2010 Wettbewerbs in 1000 unterschiedliche Klassen [47]
Regression	Quadratische Antwortflächen	Vorhersage der Tangentialkraft und der erzielbaren Oberflächenrauheit beim Drehen von AISI 1045 Stählen [48]
	Künstliche Neuronale Netze	Nachbildung von Deformationen und Verlagerungen mittels Künstlicher Neuronaler Netze zur Integration in eine Toleranzanalyse [P3]
Clustering	k-mean-Algorithmus	Vorhersage der akademischen Leistung von Studenten [49]
	Fuzzy C-Means Algorithmus	Clustering von Gehirnregionen in CT-Scans zur besseren Erkennung von Tumoren [50]
Anomalieerkennung	Hidden Markov Models	Erkennung von Notfällen und auffälligem Verhalten von Personengruppen in Kamerabildern [51]
	Bayesian Networks	Anwendung von Bayesian Networks zur Erkennung von Anomalien in komplexen Systemen [52]
Zusammenfassende Methoden	Parallele Koordinaten-Visualisierung	Erkennung von Clustern in unübersichtlichen parallelen Koordinaten-Visualisierungen [53]
	Kartenanamorphote	Kartenanamorphote-Visualisierung für nichtlineare Vielfach-Lernmodelle [54]

## 2.2 Künstliche Neuronale Netze (KNN)

Das Interesse an Künstlichen Neuronalen Netzen schwankte in den letzten Jahrzehnten sehr häufig. Dies lag einerseits an technischen Schwierigkeiten bei der Umsetzung (z. B. Entwicklung und spätere Anwendung des Backpropagation Algorithmus [55]) und an der schwankenden Menge an Fördergeldern [56]. Auch wenn die Begriffe Künstliche Intelligenz (KI, engl. Artificial Intelligence AI) und Künstliche Neuronale Netze keinesfalls als Synonyme verstanden werden dürfen, bewirkt eine Kürzung der KI-Fördergelder auch einen Rückgang der Gelder für die Entwicklung von KNN. Bei Bild 8 handelt es sich um eine grafische Visualisierung der Begriffe «KI Winter» und das schwankende Interesse und die damit verbundenen Fördergelder für den Zeitraum 1950-2009 (nach [57]) und eine Erweiterung bis zum Jahr 2019. Die Veröffentlichung von Jonathan Grudin [57] beschreibt das Wechselspiel zwischen AI (Artificial Intelligence) und HCI (Human-Computer-Interaction) inklusive der Zeiträume der KI-Winter und das Wechselspiel der Fördergelder sehr anschaulich. In Bild 8 sind die KI-Winter in den Zeiträumen «mittlere 1970er Jahre bis Anfang der 1980er Jahre» und «Anfang der 1990er Jahre bis kurz nach der Mitte der 1990er Jahre» zu sehen. Ein Grund für den zweiten KI-Winter liegt in der Tatsache begründet, dass die US-Regierung bis Ende der 1980er Jahre ca. 2 Milliarden US-Dollar (Gegenwert bzgl. dem Jahr 2009) investiert hatte und es weder autonome Landfahrzeuge noch Produkte mit natürlichem Sprachverständnis gab [57].

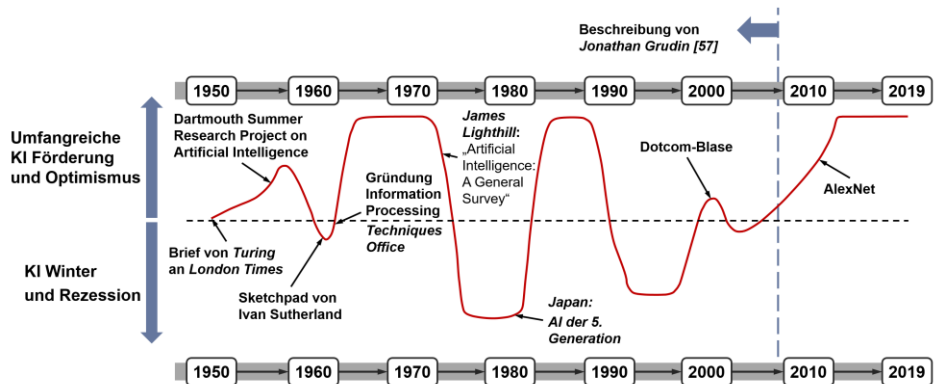


Bild 8: Verdeutlichung des Auf- und Abschwungs der Förderung und des Interesses an Künstlicher Intelligenz, Ergänzung der Veröffentlichung von [57]

Trotz der Einschnitte bei den Fördergeldern wurden Künstliche Neuronale Netze stetig weiterentwickelt und sind daher auf dem heutigen Stand angekommen. Die Nachfrage nach KNN, den verwandten Convolutional

Neural Networks und dem Deep Learning kann durch eine Analyse mittels Google Trends sehr anschaulich verdeutlicht werden. Bild 9 zeigt die zeitliche Entwicklung des Suchinteresses im Raum Deutschland seit dem Jahr 2004 für die drei Themen «Künstliche Neuronale Netze», «Convolutional Neural Networks» und «Deep Learning». Der Rückgang des Interesses an KNN wird durch ein erheblich gesteigertes Interesse für die beiden anderen Begriffe, insbesondere nach der Veröffentlichung des AlexNet [47] im Jahr 2012, erneut beflügelt.

Aus diesem Grund soll in Kapitel 2.2 zunächst auf die Grundlagen Künstlicher Neuronaler Netze eingegangen und mit diesen Grundlagen in Kapitel 2.3 das Themenfeld Deep Learning und Convolutional Neural Networks näher beleuchtet werden.

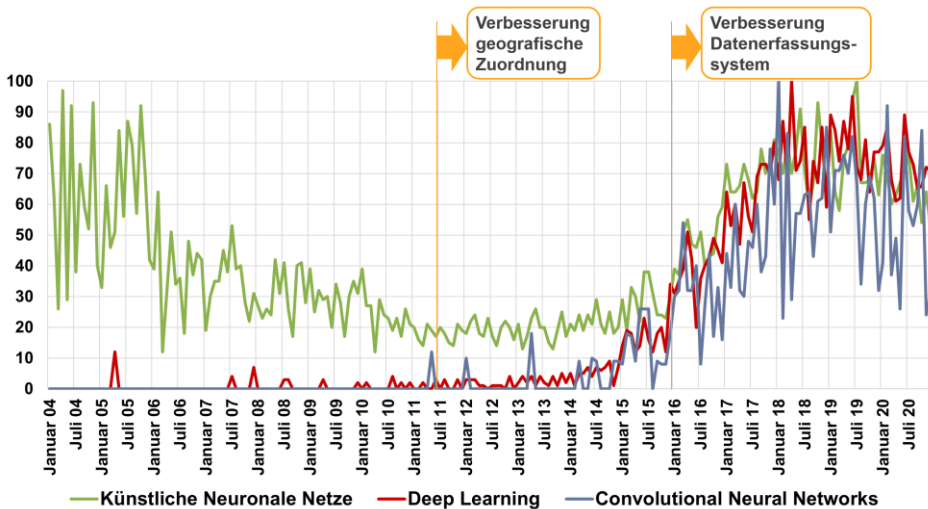


Bild 9: Google Trends, Interesse im zeitlichen Verlauf (Die Werte geben das Suchinteresse relativ zum höchsten Punkt im Diagramm für Deutschland im Zeitraum 01.01.2004-31.12.2020 an. Der Wert 100 steht für die höchste Beliebtheit dieses Suchbegriffs. Der Wert 50 bedeutet, dass der Begriff halb so beliebt ist und der Wert 0 bedeutet, dass für diesen Begriff nicht genügend Daten vorliegen) [58]

### 2.2.1 Das biologische Vorbild

In biologischen neuronalen Netzen werden Informationen an den Kontaktpunkten zwischen Neuronen, den sogenannten Synapsen, gespeichert [59]. Neuronen erhalten Signale und geben eine Antwort weiter [59]. Für das spätere Verständnis von Künstlichen Neuronalen Netzen sind vier Bestandteile eines biologischen Neurons relevant. Wie in Bild 10 dargestellt, sind dies: Dendrite, Synapsen, Zellkörper (Perikaryon) und Axon. Die

Weitergabe von Signalen zwischen den Neuronen erfolgt über Ionen und semipermeable Membranen [59].

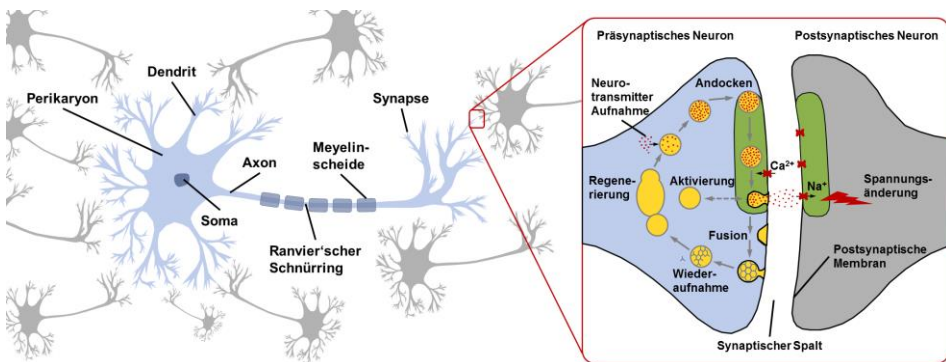


Bild 10: Schematische Darstellung von Nervenzellen (links) und einer Synapse (rechts) nach [mpg.de/synapse]

Der menschliche Körper eines Erwachsenen besteht zu 65 % aus Wasser, wobei sich 55 % davon im Inneren der Zellen und 45 % außerhalb der Zellen befinden [59]. Die Zellen werden von einer Membran (bestehend aus zwei Reihen Molekülen) umschlossen, welche eine Diffusionsbarriere darstellen. Einige Salze in unserem Körper trennen sich in negativ und positiv geladene Ionen. Ein Beispiel hierfür ist das Salz-molekül ( $NaCl$ ), welches in ein positives Natrium-Ion ( $Na^+$ ) und ein negatives Chlor-Ion ( $Cl^-$ ) gespalten wird. Andere wichtige Ionen sind Kalium ( $K^+$ ) und Calcium ( $Ca^{2+}$ ). Die Permeabilität der Ionen durch die Zellmembranen ist unterschiedlich und abhängig von der Anzahl und Größe von Poren in der Zellwand, den sogenannten Membrankanälen oder Ionenkanälen.

Damit ein Nervenimpuls ausgebildet werden kann, müssen Ionen zwischen unterschiedlichen Nervenzellen passieren können. Ionenkanäle besitzen folgende Eigenschaften: Erstens sind diese Kanäle selektiv, d. h. sie lassen nur den Transfer von bestimmten Ionen zu. Ein Natriumkanal ist daher so optimiert, dass nur Natrium-Ionen und keine anderen Ionen, wie etwa Kalium-Ionen, passieren können. Dies gilt umgekehrt ebenfalls für Kaliumkanäle. Zweitens müssen Ionenkanäle regulierbar ausgestaltet sein. Ob ein Kanal geöffnet oder verschlossen ist, hängt vom Spannungsfeld ab. Eine positivere Ladung im Inneren der Membran führt zu einer Öffnung der Membran und in Natriumkanälen können Natrium-Ionen von einer Nervenzelle in die nächste transferiert werden. [60]

Künstliche Neuronale Netze vereinfachen diesen biologischen Prozess, indem Gewichte und Transferfunktionen zur Anwendung kommen. Dieses Vorgehen soll im Folgenden anhand eines Beispiels erklärt werden.

### 2.2.2 Funktionelle Beschreibung & Netzarchitektur

KNNs sind stets aus einer gewissen Anzahl künstlicher Neuronen in verschiedenen Schichten aufgebaut. Es kann zwischen der Eingangsschicht, der Ausgangsschicht und einer oder mehreren verdeckten Schichten unterschieden werden. Dieser Aufbau wird in Bild 11 und Bild 12 gezeigt.

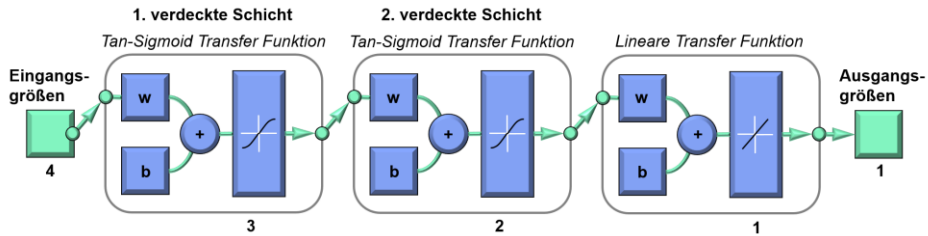


Bild 11: Netzarchitektur des KNN für den Beispieldatensatz, bestehend aus 4 Eingangs- und 1 Ausgangsgröße mit 3 Neuronen in der ersten verdeckten Schicht, 2 Neuronen in der zweiten verdeckten Schicht und 1 Neuron in der Ausgangsschicht ( $w$ : weights/Gewichte und  $b$ : Bias-Werte)

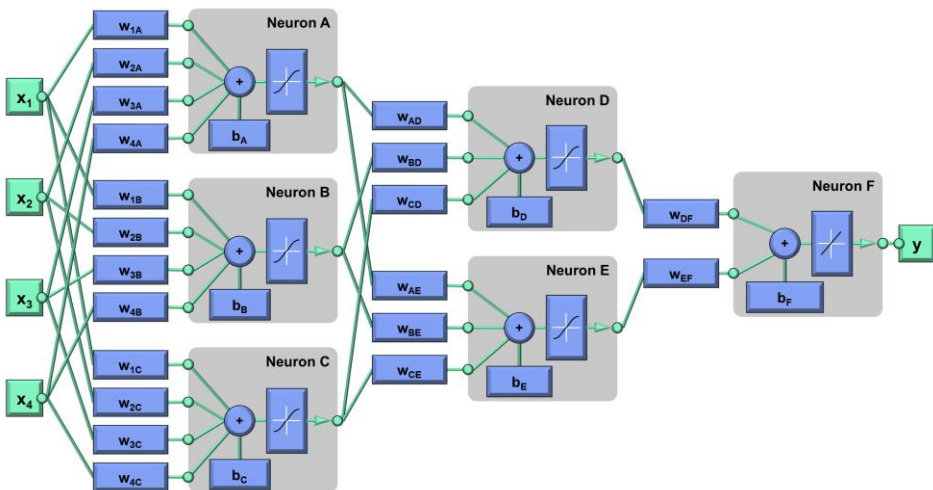
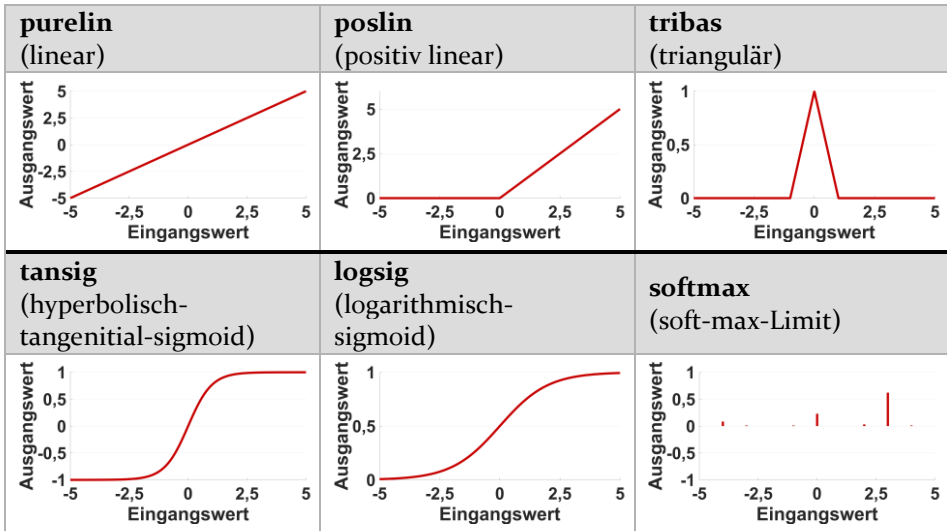


Bild 12: Detaillierte Netzarchitektur des KNN für den Beispieldatensatz

Jedes einzelne Neuron verfügt über eine eigene Transferfunktion, wobei die Transferfunktion bei jedem Neuron in einer Schicht meist identisch ist. Die Anzahl der verfügbaren Transferfunktionen variiert unter den verwendeten Softwarelösungen.

Die folgende Auflistung in Tabelle 2 zeigt eine Auswahl der gängigen Transferfunktionen (diese stehen beispielsweise in der Matlab Neural Network Toolbox zur Verfügung):

Tabelle 2: Übliche Transfer-Funktionen für Künstliche Neuronen



Die Berechnung eines Ausgabewertes für bestimmte Eingangswerte des KNN soll im Folgenden anhand eines einfachen Beispiels und einem ersten Testdatensatzes dargestellt werden. Der Eingabevektor an das KNN besitzt vier Zahlenwerte im ursprünglichen Wertebereich (vgl. Bild 13).

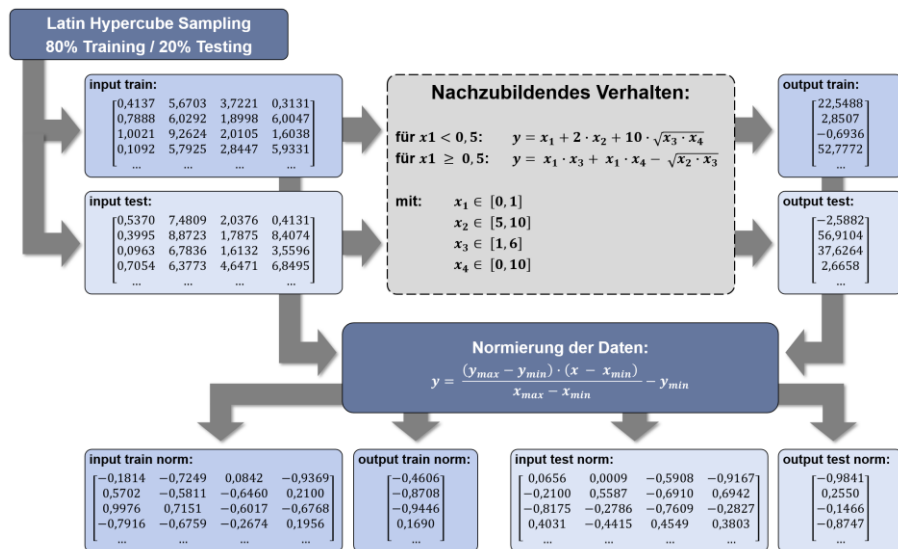


Bild 13: Beispieldatensatz zur Erklärung der Funktionsweise eines KNN

Gleiches gilt für den Ausgabewert, welchen das KNN berechnen sollte. Bild 14 zeigt die optimierten Gewichtungen und Bias-Werte. Wie eine Berechnung des Ausgabewertes dieses Netzes funktioniert, soll anhand der Folgenden detaillierten Berechnung nachvollzogen werden.

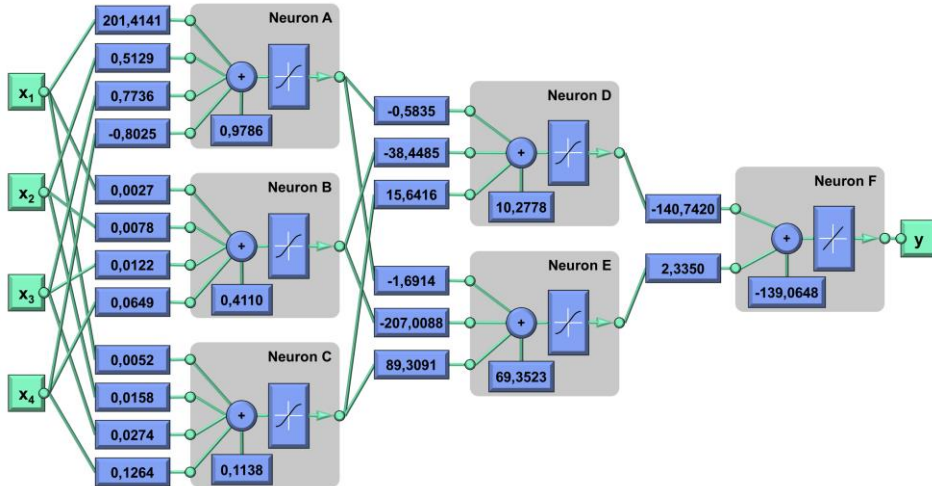


Bild 14: Trainierte Netzarchitektur des KNN für den Beispieldatensatz

In Gleichung (1) und (2) ist die erste Zeile/ das erste Sample der Testdaten dargestellt.

$$input\ test = [ 0,5370\ 7,4809\ 2,0376\ 0,4131 ] \quad (1)$$

$$output\ test = [ -2,5882 ] \quad (2)$$

Die Normierung der Eingangsdaten erfolgt mittels Gleichung (3). Die Maximal- und Minimalwerte ergeben sich aus den Trainingsdaten, mit denen die Gewichtungen des KNN bestimmt wurden. In diesem Beispiel sind dies die minimalen und maximalen Werte jeder Spalte der kombinierten Trainings- und Testdaten (in Bild 13 sind nicht alle dieser Werte dargestellt).

$$Y_{norm} = \frac{(Y_{max} - Y_{min}) \cdot (X - X_{min})}{(X_{max} - X_{min})} + Y_{min} \quad (3)$$

$Y_{norm}$  normierter Zahlenwert

$Y_{max}$  Maximalwert der Normierung

$Y_{min}$  Minimalwert der Normierung

$X$  Zahlenwert

$X_{max}$  Maximalwert des Zahlenwerts

$Y_{min}$  Minimalwert des Zahlenwerts

Da sich die normierten Werte in einem Wertebereich von -1 bis +1 befinden sollen, ergeben sich folgende Werte für die Normierung. Jede Spalte der Eingangsdaten wird separat normiert, daher werden vier Werte benötigt.

$$X_{max} = [ 1,0033 \ 9,9731 \ 6,0075 \ 9,9249 ] \quad (4)$$

$$X_{min} = [ 0,0052 \ 4,9841 \ 1,0165 \ 0 ]$$

$$Y_{max} = [ 1,0000 ]$$

$$Y_{min} = [ -1,0000 ]$$

Somit ergibt sich für den Eingangsvektor aus Formel (1) folgender normierter Eingangsvektor des KNN:

$$input\ test\ normiert = [ 0,0656 \ 0,0009 \ -0,5908 \ -0,9167 ] \quad (5)$$

Zur Berechnung des Ausgabewertes von Neuron A, B und C werden zunächst alle Eingangswerte (Y) mit den multiplizierten Gewichtungen (w) aufaddiert und schließlich der Bias-Wert (b) von Neuron A addiert. Diese gebildete Summe wird daraufhin einer Tangens-Sigmoid-Funktion (tansig) übergeben. Die in Klammern dargestellten Ergebnisse entsprechen der Berechnung mit 4 Nachkommastellen. Das Ergebnis davor entspricht dem Ergebnis einer Berechnung mit 64-Bit Fließkommazahlen auf 4 Nachkommastellen gerundet. Insbesondere in den hinteren Schichten des KNN

können durch die Berechnung mit 4 Nachkommastellen Ungenauigkeiten entstehen, dies wird im weiteren Verlauf beleuchtet.

$$\begin{aligned}
 \sum_{Neuron A} &= Y_1 \cdot w_{11} + Y_2 \cdot w_{12} + Y_3 \cdot w_{13} + Y_4 \cdot w_{14} + b_1 \\
 &= 0,0656 \cdot 201,4141 + 0,0009 \cdot 0,5129 \\
 &\quad - 0,5908 \cdot 0,7736 + 0,9167 \cdot 0,8025 + 0,9786 \\
 &= 14,4749 \quad (14,4704)
 \end{aligned} \tag{6}$$

$$Output_{Neuron A} = \text{tansig} \left( \sum_{Neuron A} \right) = 1,0000 \quad (1,0000) \tag{7}$$

$$\begin{aligned}
 \sum_{Neuron B} &= 0,0656 \cdot 0,0027 + 0,0009 \cdot 0,0078 - 0,5908 \\
 &\quad \cdot 0,0122 - 0,9167 \cdot 0,0649 + 0,4110 \\
 &= 0,3445 \quad (0,3445)
 \end{aligned} \tag{8}$$

$$Output_{Neuron B} = \text{tansig} \left( \sum_{Neuron B} \right) = 0,3315 \quad (0,3315) \tag{9}$$

$$\begin{aligned}
 \sum_{Neuron C} &= 0,0656 \cdot 0,0052 + 0,0009 \cdot 0,0158 - 0,5908 \\
 &\quad \cdot 0,0274 - 0,9167 \cdot 0,1264 + 0,1138 \\
 &= -0,0179 \quad (-0,0179)
 \end{aligned} \tag{10}$$

$$Output_{Neuron C} = \text{tansig} \left( \sum_{Neuron C} \right) = -0,0179 \quad (-0,0179) \tag{11}$$

Die berechneten Ausgabewerte der Neuronen A, B und C dienen in der zweiten verdeckten Schicht des KNN als Eingangsgröße für die Neuronen D und E. Somit ergeben sich für die Neuronen D und E folgende Ausgabewerte:

$$\sum_{Neuron D} = -1,0000 \cdot 0,5835 - 0,3315 \cdot 38,4485 - 0,0179 \cdot 15,6416 + 10,2778 = -3,3315 \quad (-3,3314) \quad (12)$$

$$Output_{Neuron D} = \text{tansig} \left( \sum_{Neuron D} \right) = -0,9974 \quad (-0,9974) \quad (13)$$

$$\sum_{Neuron E} = -1,0000 \cdot 1,6914 - 0,3315 \cdot 207,008 - 0,0179 \cdot 89,3091 + 69,3523 = -2,5621 \quad (-2,5612) \quad (14)$$

$$Output_{Neuron E} = \text{tansig} \left( \sum_{Neuron E} \right) = -0,9882 \quad (-0,9881) \quad (15)$$

Analog zur zweiten verdeckten Schicht des KNN werden in der dritten verdeckten Schicht die Ausgabewerte der Neuronen D und E als Eingangswerte für Neuron F verwendet.

$$\begin{aligned} \sum_{Neuron F} &= 0,9974 \cdot 140,7420 - 0,9881 \cdot 2,3350 - 139,0648 \\ &= -0,9893 \quad (-0,9959) \end{aligned} \quad (16)$$

$$Output_{Neuron F} = \text{tansig} \left( \sum_{Neuron F} \right) = -0,9893 \quad (-0,9959) \quad (17)$$

Mit Gleichung (3) kann der Ausgabewert von Neuron F wieder zurück in den ursprünglichen Wertebereich transformiert werden. Die Maximal- und Minimalwerte ergeben sich wieder aus den Trainingsdaten, mit denen die Gewichtungen des KNN bestimmt wurden. Somit ergibt sich der Ausgabewert des KNN mit Gleichung (19). Bild 14 zeigt das trainierte KNN mit den berechneten Gewichtungen und Bias-Werten.

$$X_{max} = [ 1,0000 ] \quad (18)$$

$$X_{min} = [ -1,0000 ]$$

$$Y_{max} = [ 92,6817 ]$$

$$Y_{min} = [ -3,3523 ]$$

$$Output_{KNN} = -2,8391 \quad (-3,1554) \quad (19)$$

### 2.2.3 Lernvorgänge

Die Anpassung der Gewichtungswerte  $w$  an den Datensatz des Trainings stellt ein Minimierungsproblem über einen mehrdimensionalen Parameterraum dar - nämlich den Vektorraum, der von den Gewichten aufgespannt wird [61]. Werbos entwickelte 1971 einen Backpropagation-Algorithmus für Künstliche Neuronale Netze und publizierte diesen in seiner Doktorarbeit im Jahr 1974 [62]. Die dort publizierten Ideen wurden später durch Parker [63] entdeckt und einer breiteren Öffentlichkeit bekannt gemacht [64]. Kurz darauf wurde die Idee der Backpropagation durch Rumelhart, Hinton und Williams aufgegriffen und damit noch bekannter [65].

In der einfachsten Form beginnt die Backpropagation damit, einen Eingangsvektor an das Netzwerk zu übergeben. Dieser wandert vorwärts durch das Netzwerk und es wird ein Ausgangsvektor des Netzwerks berechnet. Da die Ausgangswerte des Netzwerks, die herauskommen sollten, bekannt sind, kann ebenfalls der Fehler jedes Ausgangs berechnet werden. Der nächste Schritt besteht darin, die Auswirkungen der Fehler rückwärts durch das Netzwerk zu tragen, um jedem Neuron eine «quadratische Fehlerableitung» zuzuordnen. Es wird ein Gradient von jeder «quadratischen Fehlerableitung» berechnet und schließlich werden die Gewichte jedes Neurons basierend auf dem entsprechenden Gradienten aktualisiert. Anschließend wird ein neuer Eingangsvektor an das Netzwerk übergeben und der Prozess wiederholt sich. [66]

Die initialen Gewichtungswerte werden in der Regel auf kleine Zufallszahlen eingestellt. Der Backpropagation-Algorithmus funktioniert mit mehrschichtigen Neuronalen Netzen nicht richtig, wenn die initialen Gewichtungswerte entweder Null oder schlecht gewählte Werte ungleich Null sind. [67]

Das Ziel der Backpropagation besteht darin, die Anzahl der freien Parameter (Gewichte) im Netzwerk so weit wie möglich zu reduzieren, ohne die Berechnungsleistung des Netzwerks zu stark einzuschränken [68].

Daher können unterschiedliche Optimierungsalgorithmen verwendet werden. Im Folgenden soll eine Auswahl von möglichen Algorithmen kurz vorgestellt werden:

- **Levenberg-Marquardt Backpropagation-Algorithmus**
  - Entwicklung: Der Optimierungsalgorithmus wurde unabhängig von Kenneth Levenberg [69] und Donald Marquardt [70] entwickelt und ermöglicht die Minimierung einer nichtlinearen Funktion [71].
  - Speicherbedarf: Berechnung der Hesse-Matrix erfordert  $O(N^2)$  mit  $N$ : Anzahl der zu optimierenden Gewichtungen im Netzwerk. Die Berechnungskomplexität liegt sogar bei  $O(N^3)$ . [72]
  
- **Scaled Conjugate Gradient Backpropagation-Algorithmus**
  - Entwicklung: Die Basis des Algorithmus bildet der Conjugate Gradient Optimierungsalgorithmus [73], welcher so erweitert wurde, dass keine weiteren Parameter durch den Nutzer definiert werden müssen und eine aufwändige Line-Search bei jeder Iteration vermieden wird [72].
  - Speicherbedarf:  $O(N)$  mit  $N$ =Anzahl der zu optimierenden Gewichtungen im KNN. [72]
  
- **Gradient Descent with Momentum Backpropagation-Algorithmus**
  - Entwicklung: Neben dem typischen Gradient Descent Optimierungsalgorithmus wird ein zusätzlicher Momentum Term integriert. Ohne Momentum Term kann der Optimierungsalgorithmus in einem flachen lokalen Minimum stecken bleiben. Mit Momentum kann der Algorithmus durch ein solches Minimum gleiten. [74]
  - Speicherbedarf:  $O(N)$  mit  $N$ =Anzahl der zu optimierenden Gewichtungen, Bias-Werte etc. des KNN.

Der schnellste Algorithmus ist meistens der Levenberg-Marquardt Backpropagation-Algorithmus. Allerdings ist dieser für das Training von großen KNN mit tausenden von Gewichtungen meist weniger effizient, da der Algorithmus mehr Arbeitsspeicher und mehr Zeit für das Lösen des

Optimierungsproblems benötigt. Für das Training von großen KNN und zur Mustererkennung/Klassifikation ist der Scaled Conjugate Gradient Backpropagation Algorithmus eine gute Wahl. [75]

Wenn der Umfang der Eingabedaten spärlich ist, können die besten Ergebnisse wahrscheinlich mit einem Algorithmus erzielt werden, welcher die Lernrate adaptiv anpasst. [76]

#### 2.2.4 Vor- und Nachteile

Künstliche Neuronale Netze bieten zahlreiche Vorteile, allerdings gibt es auch einige Nachteile gegenüber anderen Algorithmen des Maschinellen Lernen bzw. Verfahren aus der Statistik. Im Folgenden soll zunächst auf die Nachteile und anschließend auf die Vorteile Künstlicher Neuronaler Netze eingegangen werden. Einige der genannten Punkte gelten ebenso für Deep Neuronal Networks, hierauf soll in Kapitel 2.3.4 explizit erneut eingegangen werden.

Teilweise synonym nennen Livingstone et al. [77], Kruse et al. [78], Agarwall [79] und Garro & Vazques [80] folgende Nachteile von KNN:

- **Overfitting** [77, 79]  
Während des Trainingsprozesses kann es zu einer Überanpassung der Gewichte auf den vorliegenden Trainingsdatensatz kommen. Dies hat zur Folge, dass neue, unbekannte Daten mit sehr geringer Prognosegüte durch das Netz interpretiert werden. Aus diesem Grund gilt es entsprechende Strategien zur Vermeidung von Overfitting in den Trainingsprozess bzw. die finale Auswahl geeigneter Modelle zu integrieren (beispielsweise die Verwendung von unabhängigen Test- oder Validierungsdaten).
- **Interpretation und Analyse** [77, 78]  
Im Gegensatz zu anderen Algorithmen des Maschinellen Lernen liefern KNN keine für den Menschen einfach zu verstehenden Gleichungen zurück. Vielmehr steckt die Fähigkeit zur Prognose des Netzes in den angelernten Gewichtungen und Bias-Werten der hintereinander angeordneten verdeckten Schichten und den implementierten Transferfunktionen. Aus diesem Grund werden Neuronale Netze häufig als «Black-Box-Modell» bezeichnet. Ansätze zur Interpretation können beispielsweise Sensitivitätsanalysen liefern, diese geben Aufschluss darüber welchen Einfluss bestimmte

Schwankungen der einzelnen Eingangsgrößen auf die Ausgabe des Neuronalen Netzes haben [78].

- **Zufallseffekte, Trainingsinstabilität, Konvergenzprobleme** [77, 79]

Der Prozess des Trainings von Neuronalen Netzen bringt weitere Probleme mit sich. Einerseits werden die initialen Gewichtungen und Bias-Werte üblicherweise mittels Pseudo-Zufallszahlen nach bestimmten vorgegeben Verteilungen initialisiert. Andererseits werden Trainings-, Test- und Validierungsdatensätze zumeist per Zufall nach einer bestimmten Aufteilung zugeordnet. Dies und die Wahl einer geeigneten Lernrate für unterschiedliche Optimierungsalgorithmen für die vorliegende Netzarchitektur und die zugehörigen Daten kann zu Instabilitäten während dem Training führen. Dies äußert sich häufig in schlechtem Konvergenzverhalten oder sprunghaftem Verhalten im Rahmen des Trainings des KNN.

- **Notwendigkeit großer Datenmengen, Trainingszeit** [79, 80]

Für das Anlernen einer großen Menge an Gewichtungen und Bias-Werten werden sehr viele Datensätze benötigt. Die Menge an Daten kann bei KNN größer als bei anderen Algorithmen des Maschinellen Lernen sein. Die große Menge an Daten geht mit einer längeren Trainingszeit und der Notwendigkeit zur effizienten Parallelisierung des Trainingsprozesses und der verwendeten Software einher.

Bei der Anwendung von KNN sollten diese Nachteile stets Berücksichtigung finden und entsprechende Maßnahmen zur Minimierung der genannten negativen Effekte umgesetzt werden. Dies gilt insbesondere für die Vermeidung des Overfittings und der genauen Analyse von Effekten während dem Training. Die große Menge an Daten und Trainingszeit sowie die schwierigere Interpretation der KNN sind zum jetzigen Zeitpunkt ständige Wegbegleiter. Nichtsdestotrotz bieten KNN einige entscheidende Vorteile gegenüber anderen Algorithmen des Maschinellen Lernen:

- **Abbildung komplexer, nicht-linearer Zusammenhänge** [77, 79]

Unter der Annahme, dass genügend Daten vorhanden sind, können KNN nicht-lineare Zusammenhänge in den Daten erkennen und entsprechend abbilden. Hierdurch können vielfältige Problemstellungen aus unterschiedlichsten Bereichen durch KNN abgebildet werden.

- **Einige KNN gelten als «touring complete»** [79]  
«Touring completeness» beschreibt die Fähigkeit eines Algorithmus, bei ausreichender verfügbarer Datenmenge jeden beliebigen Lernalgorithmus zu simulieren. Der Knackpunkt ist, dass die Datenmenge, die zum Lernen selbst einfacher Aufgaben benötigt wird, oft außerordentlich groß ist und eine entsprechende Erhöhung der Trainingszeit zur Folge hat. Unter anderem gelten einige Rekurrente Neuronale Netze als «Touring complete». [79]
- **Verfügbarkeit an performanter Software und Hardware**  
Zum gegenwärtigen Zeitpunkt existieren diverse Frameworks für die Implementierung, das Training und die Anwendung von KNN (Auszug: TensorFlow [81], Keras [82], CNTK [83], Caffe [84], Theano [85], Sci-Kit Learn [86], etc.). Das Training von KNNs wird insbesondere durch den Fortschritt in der GPU-Entwicklung (graphics processing unit) und durch entsprechende Parallelisierungsplattformen (NVIDIA CUDA [87] und cuDNN [88]) erheblich beschleunigt.

## 2.3 Deep Learning

Konventionelle Machine Learning Verfahren waren lange Zeit nicht in der Lage, natürliche Daten in ihrer Rohform zu verarbeiten. Jahrzehntlang hat der Bau von Mustererkennungs- oder Machine Learning Systemen sorgfältiges Engineering und domänenspezifisches Fachwissen für die Entwicklung und Extraktion geeigneter Features erfordert. Die so ermittelten Features konnten anschließend als Repräsentation der Ursprungsdaten für nachgelagerte Analysen oder Algorithmen verwendet werden. [89]

Eine grobe Abgrenzung der Begriffe «Künstliche Intelligenz», «Maschinelles Lernen», «Representation Learning» und «Deep Learning» ist in Bild 15 grafisch dargestellt und kann [90] entnommen werden. Hieraus kann insbesondere die Zugehörigkeit von bestimmten Begriffen zu anderen Begriffen schnell erkannt werden.

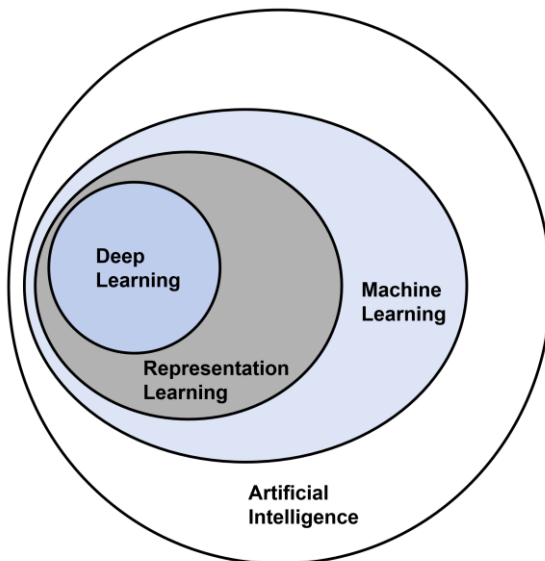


Bild 15: Abgrenzung der Begriffe «Artificial Intelligence», «Machine Learning», «Representation Learning» und «Deep Learning» nach [90]

«**Künstliche Intelligenz**» wird von vielen Autoren unterschiedlich und teilweise widersprüchlich definiert (vgl. z. B. [91, 92, 93, 94, 95, 96, 97, 98]). Laut [99] können diese Definitionen in folgende vier Kategorien einsortiert werden:

- «Systeme, die wie Menschen denken» [91, 93]
- «Systeme, die rationell denken» [92, 98]
- «Systeme, die wie Menschen handeln» [94, 96]
- «Systeme, die rationell handeln» [95, 97]

«**Representation Learning**» sind Methoden, die es erlauben, eine Maschine mit Rohdaten zu füttern und automatisch die für die Erkennung oder Klassifizierung erforderlichen Repräsentationen (Features) zu entdecken [89].

«**Deep Learning**» sind Methoden des «Representation Learning» mit mehreren nacheinander gelagerten Ebenen an Features, welche ausgehend von den Eingangsdaten durch einfache, aber nicht-lineare Module erzeugt werden [89]. Der Begriff «Deep Learning» wurde erstmals durch Aizenberg et al. [100] im Jahr 2000 verwendet und hat sich darauf hin sehr schnell verbreitet.

Im Folgenden soll in Kapitel 2.3.1 eine Abgrenzung unterschiedlicher Netzarchitekturen Neuronaler Netze erfolgen und in Kapitel 2.3.2 auf den bekanntesten Vertreter des Deep Learning – die Convolutional Neural Networks (CNN) – eingegangen werden. Nach einer Erklärung der üblichen Lernverfahren, inklusive dem sogenannten Transfer Learning in Kapitel 2.3.3 soll schließlich in Kapitel 2.3.4 auf die Vor- und Nachteile des Deep Learning eingegangen werden.

### 2.3.1 Erklärung

Generell werden unter dem Begriff «Deep Learning» bestimmte Arten und Ausprägungen von Neuronalen Netzen verstanden. «Deep Learning» hat sich als sehr gut bei der Entdeckung von komplexen Strukturen in hochdimensionalen Daten erwiesen und ist daher in vielen Bereichen von Wissenschaft, Wirtschaft und Verwaltung anwendbar [89]. Die folgende Auflistung repräsentiert den aktuellen Stand der Forschung bzgl. Neuronalen Netzen, welche dem Themenfeld «Deep Learning» zugeordnet werden können (es besteht kein Anspruch auf Vollständigkeit, da sehr viele Unterarten von Neuronalen Netzen existieren und ständig neue Formen und Architekturen von Forschern entworfen werden):

- **Deep Feedforward Networks/Deep Multilayer Perceptrons:**  
Die Abgrenzung zwischen klassischen Neuronalen Netzen und Deep Multilayer Perceptrons ist fließend. Zum gegenwärtigen Zeitpunkt werden Neuronale Netze mit drei Schichten (Eingangs- und Ausgangsschicht und einer zusätzlichen verdeckten Schicht) als «Shallow Neural Networks» bezeichnet und Netze mit mehr als einer verdeckten Schicht als «Deep Neural Network» bezeichnet (vgl. [101]).
- **Convolutional Neural Networks (CNN):**  
Die Ursprünge der Convolutional Neural Networks gehen auf Arbeiten von D.H. Hubel und T.N. Wiesel aus den 1950er und 1960er Jahren zurück, welche den Aufbau der Sehrinde von Katzen [102] sowie Makaken- und Klammeraffen [103] untersuchten. Die beiden Autoren entwickelten aus diesen Erkenntnissen ein hierarchisches Modell des Sehnervensystems [104, 105]. Diese Ideen wurden von K. Fukushima aufgegriffen und als «neocognitron» 1980 veröffentlicht [106]. Bei Fukushima waren bereits die beiden wesentlichen Elemente eines Convolutional Neural Networks enthalten: Convolution-/Faltungsschichten und Downsampling-/Modell-

reduktionsschichten. Da es sich bei Convolutional Neural Networks um die bekanntesten Vertreter des modernen Deep Learnings handelt und dies für den weiteren Teil der vorliegenden Arbeit von Bedeutung sind, soll in Kapitel 2.3.2 näher auf den Aufbau/Architektur von CNN eingegangen werden.

- **Deep Recurrent Neural Networks (Deep RNN):**

Recurrent Neural Networks basieren auf den Arbeiten von Little [107], Rumelhart et al. [108] und Hopfield [109]. Es handelt sich hierbei um Neuronale Netze, welche sequenzielle Informationen verarbeiten können. Im Gegensatz zu den Convolutional Neural Networks (Feedforward-Netze), enthalten Recurrent Neural Networks Schleifen in der Netzstruktur [110]. Hierdurch können Verzögerungen (Delays) von Signalen abgebildet werden und RNN können zeitliche Abfolgen in den Trainingsdaten erkennen und nachbilden. Die Architekturen reichen von vollständig miteinander verbundenen Schichten bis hin zu teilweise verbundenen Schichten eines Netzwerks, dies schließt auch mehrschichtige Feedforward-Netze mit unterschiedlichen Ein- und Ausgangsschichten mit ein [111]. Im Folgenden soll kurz auf zwei sehr erfolgreiche Vertreter der Rekurrenten Neuronalen Netze, LSTM- und GRU-Netze eingegangen werden.

- **Long short-term memory (LSTM):**

Die Entwicklung der LSTM-Netze ist stark mit J. Schmidhuber verbunden. Der ursprüngliche Entwurf wurde 1997 durch Hochreiter & Schmidhuber [112] vorgestellt und durch Geers et al. um das «forget gate/keep gate» ergänzt [113]. 2000 wurden ebenfalls durch Geers et al. sogenannte «peephole connections» (Verbindungen von der LSTM-Zelle zu den gates) ergänzt [114] (gleichnamige Veröffentlichung wie [113]).

- **Gated recurrent units (GRU):**

GRU Netze beinhalten sog. «gating mechanisms», welche erstmals durch [115] vorgestellt wurden. Obwohl GRU sehr nahe mit LSTM verwandt sind, weisen diese eine geringere Anzahl von Parametern auf [116]. Dies liegt in dem Nichtvorhandensein eines Ausgabegates begründet. Ein GRU Netzwerk besitzt zwei Gates (Reset und Update), das Reset-Gate bestimmt, wie neue Eingangswerte mit den bestehenden Memory-Werten kombiniert werden und das Update-Gate bestimmt, inwieweit die aktuellen Memory-Werte

beizubehalten sind [117]. Nach [118] sind GRU in der Lage bessere Ergebnisse als LSTMs zu erzielen. Ähnliche Erkenntnisse zeigen [119], wobei hier LSTMs bei Aufgaben des Language Modellings bessere Ergebnisse erzielen.

### 2.3.2 Netzarchitektur CNN

In Kapitel 2.2 wurden bereits Künstliche Neuronale Netze vorgestellt. Diese weisen gegenüber CNN jedoch einen gravierenden Nachteil auf: KNN sind nicht Translations-invariant [120]. Das heißt, dass Eingangsdaten immer an den gleichen Eingangsneuronen eines KNN übergeben werden müssen, sonst kann das Netz die Informationen nicht korrekt verarbeiten. Bilder können beispielsweise nicht gedreht oder andere Ausschnitte eines Bildes übergeben werden. Aus diesem Grund wird einem klassischen Neuronalen Netz mindestens eine «Convolution» (Faltung) vorgeschaltet. Unter «Convolution» wird die Multiplikation einer größeren Matrix mit einer kleineren verstanden [120]. Da dies jedoch mathematisch nicht möglich ist, wird jeweils ein Ausschnitt der großen Matrix mit der kleineren Matrix multipliziert. Dieser Vorgang wird mit zahlreichen Ausschnitten der großen Matrix fortgesetzt, solange bis die gesamte große Matrix verarbeitet wurde. Bevor die so extrahierten Werte an ein KNN weitergegeben werden können, muss der Umfang der Daten reduziert werden. Hierfür kommt «Pooling» (Bündelung) zum Einsatz [121]. Meistens wird «Max Pooling» angewendet, hierbei wird aus einem Ausschnitt der resultierenden Matrix jeweils der maximale Wert weiter verwendet [121]. Andere Pooling Arten sind «Average Pooling», «Stochastic Pooling» [122] oder «Mixed Pooling» [123]. Anstelle von Pooling können ebenfalls andere Convolution Schichten verwendet werden, jedoch mit einer größeren Schrittweite [124, 125] (Erklärung der Schrittweite erfolgt später anhand von Bild 16).

Das Training von Convolutional Neural Networks ist heute sehr schnell mittels Grafikprozessoren (GPU: graphics processing unit) möglich und es können Netze 10-20x schneller trainiert werden [89].

Die Anwendungsgebiete von CNN sind vielfältig und es sollen hier nur einige Beispiele genannt werden:

Tabelle 3: Anwendungsbeispiele für den Einsatz von CNN

Reduktion von Rauschen in Bildern	[126]
Gesichtserkennung	[127, 128]
Erkennung von Gesichtern und Gesichtszügen	[129]
Objekterkennung mittels Reccurrent CNN	[130]
Klassifikation von handschriftlichen Ziffern	[131]
Handschrifterkennung mehrerer Sprachen	[132]
Berechnung von Tiefeninformationen aus zwei Bildern	[133]
Erkennung interstitieller Lungenerkrankungen anhand von CT-Bildern	[134]
Erkennung menschlicher Bewegungen in Aufnahmen von Überwachungskameras	[135]
Segmentierung von Knieknorpeln mittels triplanarer CNN	[136]
Automatisches Verknüpfen von gesprochenen Sätzen	[137]
Klassifizierung von Sätzen auf Basis von Wortvektoren	[138]
Klassifikation von Videoaufnahmen	[139]
Visuelle Dokumenten Analyse	[140]
Bildklassifikation (AlexNet)	[47]
Spracherkennung	[141]

Das prinzipielle Konzept der Convolution und des Max Poolings soll anhand eines einfachen Beispiels anhand von Bild 16 verdeutlicht werden. Neben der Eingangsmatrix (unten links) ist exemplarisch ein zweidimensionaler Laplace Filter dargestellt, wie er beispielsweise zur Kantenerkennung verwendet werden kann. Dieser Filter wird sukzessive über die Eingangsmatrix verschoben und jeweils die Filtermatrix mit einem gleich großen Bereich der Eingangsmatrix punktweise multipliziert und

aufsummiert. Der Filter wird sowohl horizontal als auch vertikal verschoben (hier mit einer Schrittweite von 1). Der Rechenweg für 3 Werte der Matrix nach der Faltung ist exemplarisch in Bild 16 angegeben. Anschließend erfolgt ein Max Pooling, ebenfalls mit einer vertikalen und horizontalen Schrittweite von 1 (die Schrittweite wird auch als «stride» bezeichnet). Im dargestellten Beispiel kann die Eingangsmatrix der Größe 6x6 auf eine Größe von 3x3 reduziert werden (25 % der initialen Größe).

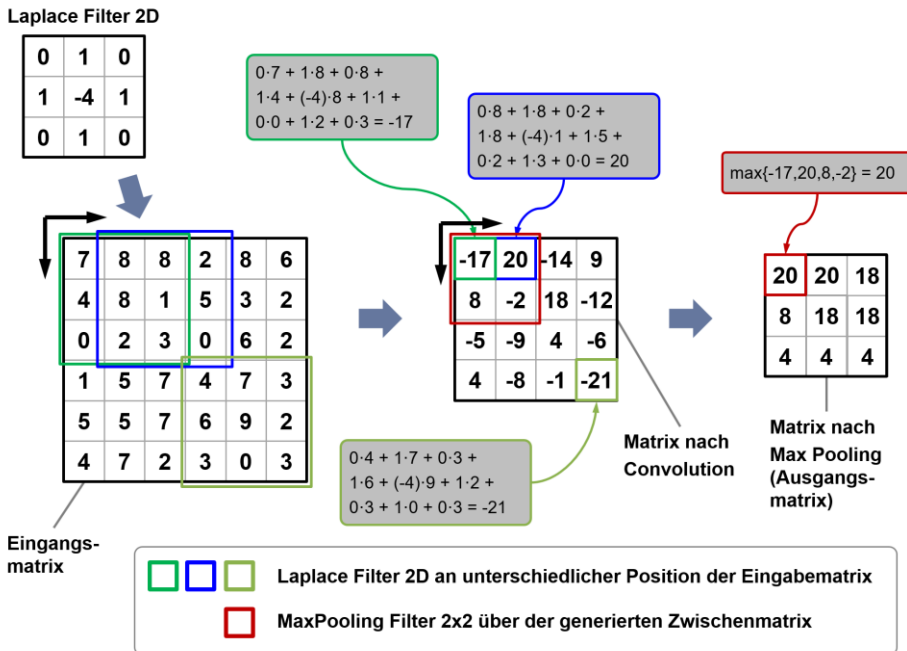


Bild 16: Convolution einer Eingangsmatrix mittels Filter (hier Laplace 2D) und MaxPooling (Größe 2x2), Schrittweite der Filter jeweils 1

Die Dimensionen der Eingangsmatrix können durch die Kombination aus verschiedenen Convolution und Pooling Schichten sukzessive reduziert werden. Nach dieser Reduktion kommen eine Reihe von vollvernetzten Neuronenschichten zum Einsatz, dies entspricht somit einen zuvor beschriebenen KNN.

Exemplarisch ist dieses Vorgehen in Bild 17 für die Erkennung von handschriftlichen Zahlen dargestellt. Die Eingangsmatrix der Größe 64x64 wird aus der handschriftlichen Zahl abgeleitet (hier als schwarz-weiß Bild, eine Umsetzung mittels Graustufen Bildern ist ebenso einfach möglich). Durch die Convolution und Pooling Schichten nimmt die Größe der initialen

Eingangsdaten ( $64 \times 64 = 4.096$ ) immer weiter ab, bis schließlich vor der vollvernetzten Schicht eine Matrix der Größe  $14 \times 14 = 196$  resultiert.

Mittels der Convolution Schichten wird in der Regel nur eine sehr geringe Reduktion der ursprünglichen Matrix erreicht (Ausnahme: es werden größere Schrittweiten der Filter verwendet). Die Hauptächliche Reduktion der Matrixgröße wird in dem dargestellten Beispiel durch die Pooling Schichten erreicht. Die Max Pooling Layer halbieren jede Dimension der Matrix. Somit erfolgt eine Reduktion auf 25 % der ursprünglichen Matrixwerte.

Die 196 Zahlenwerte nach der zweiten Max Pooling Schicht werden anschließend an eine Neuronenschicht mit 196 Neuronen transferiert und anschließend an weitere Neuronenschichten weitergeben. Am Ende des Netzes resultieren 10 Ausgangsgrößen, welche den 10 Ziffern von 0 bis 9 mittels einer Klassifizierungsschicht (softmax Schicht) zugeordnet werden können. Ähnlich wie in Bild 16 kommt aus Gründen der Vereinfachung auch hier eine Schrittweite von 1 in den Convolution und Pooling Schichten zum Einsatz.

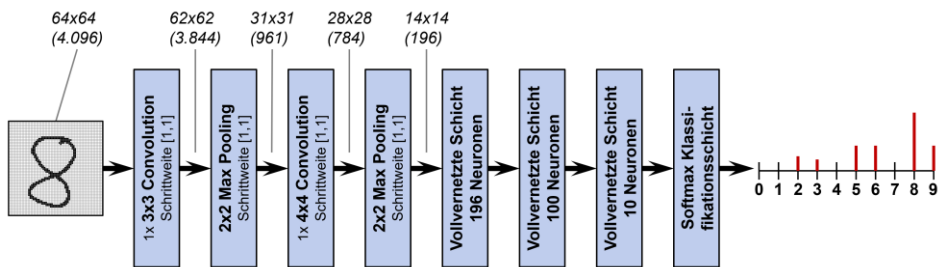


Bild 17: Exemplarischer Aufbau eines CNN zur Klassifikation handgeschriebener Zahlen

### 2.3.3 Lernvorgänge (auch Transfer Learning)

Die meisten Deep Learning Anwendungen verwenden Feedforward Neural Network Architekturen, welche versuchen den Zusammenhang zwischen Eingangs- und Ausgangsdaten herzustellen [89]. Für das Training der Gewichtungen und der Filter wird der Backpropagation Algorithmus nach Rumelhart, Hinton und Williams [65, 108] zum Einsatz, die Ursprünge der Entwicklung gehen zurück auf die Dissertation von Werbos [62], welche allerdings zum Zeitpunkt der Veröffentlichung wenig Beachtung fand [66]. Bei diesem Vorgehen werden die Eingangsgrößen des Trainingsdatensatzes an das Neuronale Netz übergeben und es wird die Ausgangsgröße prognostiziert. Diese Ausgangsgröße kann dann mit der tatsächlich erwarteten Ausgangsgröße des Trainingsdatensatzes verglichen werden (Ermittlung

der Abweichung, des Fehlers oder der Kostenfunktion). Der Backpropagation Algorithmus kann durch das Rückwärtslaufen dieser Informationen einen Gradienten berechnen [90]. Mittels dieser Gradienten werden die zu optimierenden Größen (Gewichtungen, Filter, Bias-Werte etc.) in jeder Iteration des Trainingsprozesses angepasst. Die Veröffentlichung der Backpropagation Methode war laut Widrow & Lehr ohne Frage die einflussreichste Entwicklung auf dem Gebiet der Neuronalen Netze der letzten Jahre [66]. In der Praxis ist das Festsitzen in lokalen Minima durch den Optimierungsalgorithmus für große Neuronale Netze selten ein Problem, dies belegen auch theoretische und empirische Untersuchungen [89].

Häufig wird nicht der gesamte zur Verfügung stehende Trainingsdatensatz genutzt, sondern es werden Teile («batches») dieses Datensatzes genutzt. Dies hat den Vorteil, dass weniger Berechnungsleistung benötigt wird und der Umfang an zur Verfügung stehenden Daten wird nacheinander an das Netz übergeben und jeweils in mehreren Schritten die zu optimierenden Parameter angepasst. Die Batchsize sollte jedoch immer so groß wie möglich gewählt werden. Anstelle einer Anpassung (Reduktion) der Lernrate des Optimierungsalgorithmus empfehlen Smith et al. [142] stattdessen die Batchsize zu erhöhen. Dies führt zu genauso guten Ergebnissen und resultiert in weniger notwendigen Anpassungen der Parameter, was gleichbedeutend mit einer besseren Parallelisierbarkeit zu setzen ist [142].

Bisher wurde immer davon ausgegangen, dass ausreichend Daten vorliegen, um ein entsprechendes Neuronales Netz mit hoher Prognosegüte zu trainieren. Das Netz wird ausgehend von Initialisierungswerten (z. B. normalverteilte Zufallszahlen zwischen 1 und 0) der Gewichtungen trainiert und es werden diese Werte auf den Datensatz hin optimiert. Ein anderer Ansatz ist das sogenannte Transfer Learning. Hierbei werden Netze mit einem sehr großen Datensatz angelernt, welcher eventuell gar keine Verbindung zu dem eigentlich zu trainierenden Datensatz aufweist. Anschließend erfolgte eine weitere Optimierung der zuvor optimierten Parameter mittels des tatsächlichen Datensatzes. Teilweise können auch nur bestimmte Bereiche des Netzes nachtrainiert werden und der Rest des Netzes bleibt vollständig unverändert. Die bisherigen Verfahren des Transfer Learning fokussieren sich auf die Verbesserung der Generalisierbarkeit für verschiedene Verteilungen zwischen dem eigentlichen Datensatz und dem für das initiale Training verwendeten Datensatz [143]. Transfer Learning besitzt nach Torrey & Shavlik [144] folgende Vorteile: Eine höhere Prognosegüte beim Start des Trainings, eine schnellere Steigerung der Prognosegüte während dem Trainingsprozess und eine höhere Asymptote als Trainingsergebnis (dies verdeutlicht Bild 18).

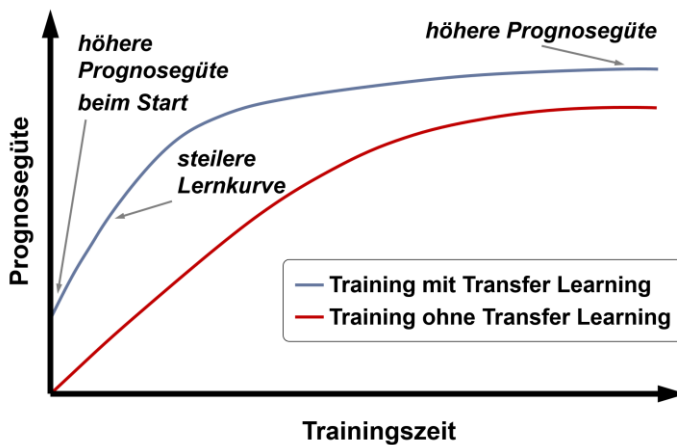


Bild 18: Einfluss des Transfer Learning auf die zu erreichenden Prognosegüten und drei Vorteile dieses Vorgehens nach [144]

#### 2.3.4 Vor- und Nachteile

Deep Learning und die entsprechenden Methoden verfügen sowohl über sehr gewichtige Vorteile, besitzen aber auch einige Nachteile. Aus diesem Grund soll im Folgenden eine kurze Auflistung der Vor- und Nachteile von Deep Learning vorgenommen werden:

##### Vorteile:

- Deep Learning ermöglicht die automatische Generierung von aussagekräftigen Feature Extraktoren (z. B. Filter) auf Basis von Daten [89]. Dies ermöglicht eine bessere Vorhersage, da die extrahierten Features relevante Merkmale des zugrundeliegenden Problems beinhalten.
- Bei entsprechender Absicherung kann eine große Robustheit gegenüber Schwankungen in den zu Grunde liegenden Daten erreicht werden. Insbesondere eine hohe Robustheit gegenüber additiven und strukturierten Störeinflüssen sollte das Ziel von Robustheitsanalysen darstellen. [145]
- Die zuvor beschriebenen Ansätze des Deep Learnings können auf sehr viele andere Bereiche und Domänen übertragen werden (z. B. Signal Processing [146], Medizinische Anwendungen [147], Social Media [148] etc.).

- Berechnungen können mit GPUs und Verwendung von entsprechender Hardware parallelisiert durchgeführt werden [149] und sind für große Datenmengen skalierbar [150].
- Sehr gute Prognosegüten bei der Verarbeitung von sehr großen Datenmengen bzw der Verwendung von speziell segmentierten Daten (Beispiel bzgl. Segmentierung [151]).
- Das Vorgehen des Deep Learnings ist sehr flexibel und kann auf zukünftige Aufgaben und Herausforderungen angepasst werden. Insbesondere mit Fokus auf den Bereich «Big Data» [152].

### **Nachteile:**

- Die Bewertung der Prognosegüte eines Modells erfolgt anhand von bestimmten Metriken. Bei manchen Randbedingungen sollten andere Metriken gewählt werden als bei anderen Randbedingungen. Es kann vorkommen, dass ein Modell eine optimale Güte bei einer bestimmten Metrik aufweist, bei einer anderen Metrik ist das identische Modell aber schlechter bewertet. [153]
- Deep Learning Methoden benötigen sehr große Datenmengen. [154]
- Methoden des Deep Learnings benötigen generell mehr Zeit für den Prozess des Trainings. [155]
- Es sind teure Grafikkarten mit großer Rechenleistung notwendig, um entsprechende Modelle mit vielen Daten trainieren zu können. [156]
- Es existieren bisher wenig standardisierte Prozesse um geeignete Methoden, Algorithmen und deren Einstellungen für eine bestimmte Problemstellung auszuwählen. Dies erschwert den Zugang Personen, welche sich bisher nicht mit der Architektur von Neuronalen Netzen beschäftigt oder wenig Erfahrung mit den verwendeten Trainingsmethoden haben. [157]
- Eine Beschreibung der Generalisierungsfähigkeit Künstlicher Neuronaler Netze stellt eine große Herausforderung für klassische Methoden der statistischen Lerntheorie dar. [158]
- Trotz des Erfolgs tiefer Neuronaler Netze stehen keine theoretischen Fortschritte gegenüber, die das Verhalten dieser Netze zufriedenstellend erklären. [159]

## 2.4 Übergabe von Geometriedaten an Machine Learning Algorithmen

In vielen Bereichen kann es sinnvoll sein, dreidimensionale Geometriedaten an Algorithmen des Maschinellen Lernens zu übergeben und mit unterschiedlichen Zielgrößen zu kombinieren. Dies ermöglicht den Aufbau von Regressionsmodellen zwischen den Eingangs- und Ausgangsdaten oder die Klassifikation bzw. das Clustering von übergebenen Eingangsdaten. Beispiele für dreidimensionale Geometriedaten sind: CAD-Modelle [160], numerische Simulationsdaten [161], Scan Daten von optischen oder taktilen Messverfahren [162, 163], LiDAR Aufnahmen [164, 165, 166], Magnetische Felder und Aufnahmen von Coronal Diagnostic Spectrometern [167] etc. Fang et al. [168] beschreibt die Vielzahl der vorliegenden Daten und die Bereiche, aus denen diese Daten stammen: Ingenieurwissenschaften, Unterhaltungsbranche, Medizinbildung und Biologie [169, 170, 171, 172, 173, 174, 175, 176].

Die Vielzahl dieser Daten verdeutlicht bereits die Notwendigkeit zur Übergabe an Algorithmen des Maschinellen Lernen, insbesondere für die Auswertung oder Nachbildung der in den Daten enthaltenen Zusammenhänge. Für die Übergabe dieser Daten an die zuvor beschriebenen Algorithmen stehen bereits einige Verfahren zur Verfügung, diese sollen im Folgenden (Kapitel 2.4.1 - 2.4.6) kurz vorgestellt werden. Die meisten der vorgestellten Verfahren sind dem Bereich der Objekterkennung zuzuordnen, Methoden aus dem Bereich der Formsegmentierung oder Ähnlichkeitsanalyse werden entsprechend gekennzeichnet.

### 2.4.1 Volumetric Convolutional Neural Networks

Volumetric CNNs stellen die Vorreiter der im Weiteren vorgestellten Netze bzw. Vorverarbeitungsschritte dar. Begrenzt wird die Auflösung der Volumetric CNNs durch die hohe Menge an notwendigen Daten für das Training und durch den mit der Convolution verbundenen Rechenaufwand [177].

Ein Vertreter der Volumetric CNNs ist VoxNet [178]. Bei VoxNet handelt es sich um ein 3D Convolutional Neural Network, welches um eine zusätzlich vorgeschaltete Volumetric Occupancy Grid Schicht erweitert wird. Occupancy Grids stellen den Zustand der Umgebung als 3D-Gitter aus Zufallsvariablen dar (jede Zufallszahl ist einem Voxel zugeordnet). Die «Occupancy» kann mit unterschiedlichen Modellen abgebildet werden, beispielsweise mit binären Zufallszahlen für die Voxel, die entweder den Wert 0

oder 1 annehmen können. Anschließend können Wahrscheinlichkeiten für die Occupancies berechnet werden, um unterschiedliche Zustände für das untersuchte Voxel abzubilden (z. B. *frei*, *belegt*, *unbekannt*) [179]. Die probabilistische Schätzung der betreffenden Zufallszahl ist von eingehenden Daten in das Occupancy Grid und weiteren Vorkenntnissen abhängig. [178]

Der Vorteil von Occupancy Grids ist, dass eine Unterscheidung zwischen freien, belegten und unbekannten Bereichen, auch aus mehreren Blickrichtungen und unter der Berücksichtigung von Bildfolgen – also der zeitlichen Abfolge bei Kamera-/Sensoraufnahmen – ermöglicht wird. Darüber hinaus können Volumetric CNNs mit Occupancy Grids verschiedene Quellen von 3D-Daten verarbeiten (LiDAR-Punktwolken, RGB-D-Punktwolken, voxelisierte CAD-Modelle, etc.) [180].

Ein Problem der Objekterkennung von 3D-Objekten ist die Aufrechterhaltung einer konsistenten Orientierung. Die Entwickler Maturana & Scherer von VoxNet [178] verwenden hierfür eine Methode zur künstlichen Generierung von zusätzlichen Trainingsdaten, dieses Vorgehen ist in Bild 19 dargestellt.

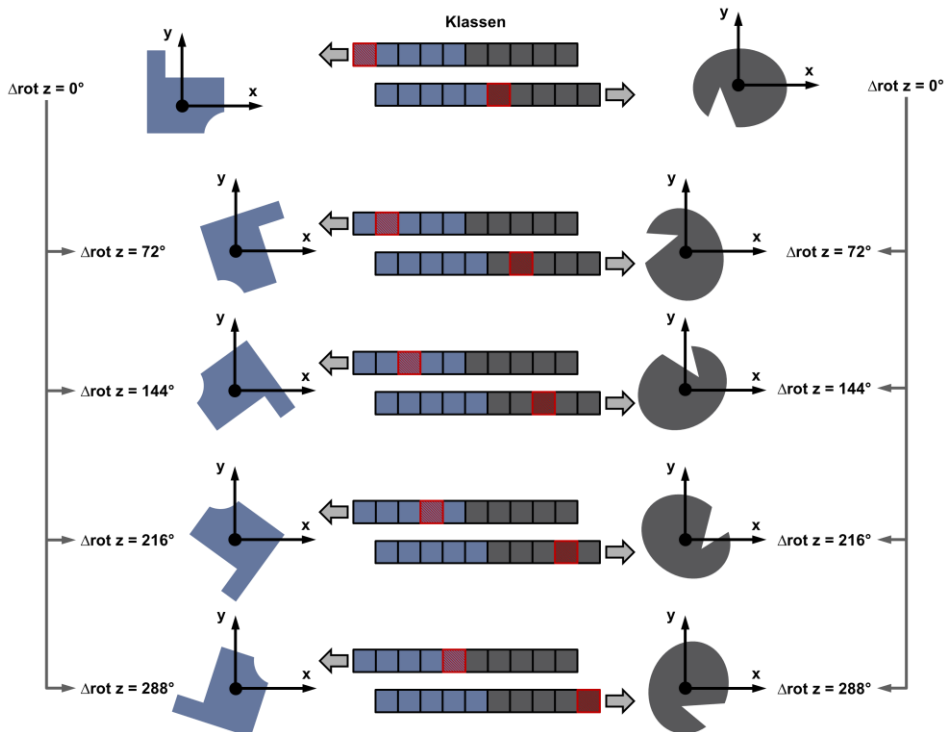


Bild 19: Ansatz zur Berücksichtigung von unterschiedlichen Orientierungen um die Z-Achse zur Klassifikation mittels CNN nach [178]

Der vorliegende Trainingsdatensatz wird  $n$ -mal künstlich um die z-Achse rotiert und anschließend angelernt (durch die Anwendung für Roboter, welche auf der Bodenoberfläche fahren, muss nur eine Rotation um die Z-Achse betrachtet werden). Jede Objektklasse wird ebenfalls  $n$ -mal vergeben. Bei der eigentlichen Anwendung des angelernten Modells werden die  $n$  Klassen wieder zu einer kombiniert. Maturana & Scherer [178] verwenden  $n = 12$  oder  $n = 18$  künstlich erzeugte Rotationen.

### 2.4.2 Multiview Convolutional Neural Networks

Multiview CNNs verwenden mehrere Ansichten auf ein Objekt oder eine Punktwolke, um anschließend eine Klassifikation durchzuführen. Dies ist notwendig, da die Ausrichtung eines 3D-Objekts im Raum schwierig und manchmal unzureichend definiert ist [181]. Su et al. [181] verwenden zur Generierung der gerenderten Ansichten ausgehend von Polygonnetzen das Phong Reflexions-Modell [182].

Im ersten Versuch werden 12 Ansichten eines Objekts durch die Rotation einer virtuellen Kamera in  $30^\circ$ -Schritten um das Objekt herum erzeugt. Die Kameras schauen unter einem Winkel von  $30^\circ$  (bezogen auf die Grundebene) auf den Schwerpunkt des Polygonnetzes. Somit stehen für ein Objekt 12 gerenderte Bilder zur Verfügung. Im zweiten Versuch werden 20 virtuelle Kameras an den 20 Ecken eines, das Objekt umschließenden, Iko-saeders platziert. Diese Kameras schauen ebenfalls auf den Schwerpunkt des Polygonnetzes. Jede der 20 Kameras wird um die Sichtachse um  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  und  $270^\circ$  gedreht und es wird jeweils ein Bild gerendert. Somit stehen für ein Objekt  $20 \cdot 4 = 80$  gerenderte Bilder zur Verfügung. Jedes Bild wird einzeln verwendet, um ein erstes CNN anzulernen. Die Ergebnisse dieses ersten CNNs werden dann für alle Ansichten (hier entweder 12 oder 80) mittels «view pooling» in Eingabewerte für ein zweites CNN umgewandelt und dieses anschließend trainiert. Das zweite CNN führt die eigentliche Klassifikation des ursprünglichen Objekts durch.

Einen ähnlichen Ansatz verwenden Ge et al. [183], um die Handposition und -orientierung (auch der Finger und der Fingergelenke), mittels drei Projektionen einer Punktwolke auf vorgegebene Ebenen, zu ermitteln.

In einem direkten Vergleich mit Volumetric CNNs stellen Qi et al. [180] fest, dass die damals verfügbaren Multiview CNNs bessere Ergebnisse als die Volumetric CNNs liefern. Darüber hinaus stellen die Autoren fest, dass die 3D-Auflösung der Volumetric CNNs die Engstelle für die Leistungsfähigkeit dieser Netzarchitektur darstellt.

### 2.4.3 Feature-Based Convolutional Neural Networks

Der Umfang an verfügbaren dreidimensionalen Datensätzen nimmt ständig zu und der Ursprung der Daten sind z. B. Ingenieurwissenschaften, Unterhaltungsindustrie, Medizin oder Biologie [168]. Fang et al. [168] haben eine Methode entwickelt, um prägnante und geometrisch aufschlussreiche Formfeatures aus den dreidimensionalen Daten abzuleiten. Die extrahierten Formfeatures können anschließend für das Training eines CNN verwendet werden. Bei diesem Verfahren handelt es sich um einen Vertreter aus dem Bereich der Objektsegmentierung. Die beschriebene Methode muss laut Fang et al. [168] folgende Punkte berücksichtigen:

- die hohe Komplexität von dreidimensionalen Modellen [170, 174, 184, 185]
- die strukturellen Unterschiede in 3D-Modellen [170, 185, 186]
- Rauschen, Unvollständigkeit, unterschiedliche Orientierungen und Verdeckung von Objekten [170, 174, 185, 186]

Die entwickelte Methode besteht aus vier Komponenten. Die erste Komponente ist eine Datenbank, in der eine große Menge an dreidimensionalen Formen bzw. Geometrien gespeichert werden. Die zweite Komponente extrahiert die Formmerkmale für jede Form, wobei jeweils zwei Merkmale extrahiert werden: «heat kernel signature» (HKS) und «heat shape descriptor» (HeatSD). Die dritte Komponente ist ein tiefes Neuronales Netzwerk zum Erlernen der Formmerkmale der Formen/Geometrien. Eine Auswahl der HeatSD wird an eine Hauptkomponentenanalyse (PCA) und an eine lineare Diskriminanzanalyse (LDA) weitergegeben, um schließlich einen Eigen-shape descriptor (ESD) und einen Fisher-shape descriptor (FSD) zu erzeugen. Die so erzeugten Deskriptoren werden als Eingangsdaten für ein tiefes Neuronales Netz verwendet und dienen somit als Vorverarbeitung der Formen bzw. Geometrien, um diese anschließend an das Neuronale Netz übergeben zu können.

Einen ähnlichen Ansatz verfolgen Guo et al. [187] mit dem Lernen einer kompakten und effektiven Netzsdarstellung aus Low-Level-Features.

Hierbei werden sieben Arten von geometrischen Merkmalen, die in bestehenden Studien weit verbreitet sind, extrahiert:

- Krümmung [188]
- Hauptkomponentenanalyse (PCA) [189]
- shape diameter function (SDF) [190]
- distance from medial surface (DIS) [191]
- average geodesic distance (AGD) [192]
- shape context (SC) [193]
- spin image (SI) [194]

Durch die Berechnung der genannten Merkmale können die Eigenschaften jedes einzelnen Dreiecks (einer tesselierten Dreiecksflächen-Geometrie) aus mehreren Perspektiven eindeutig beschrieben werden. Dies ermöglicht das Antrainieren eines CNN zur Berechnung der Wahrscheinlichkeiten der Zugehörigkeit zu verschiedenen Objektteilen. Somit kann jedem Dreieck der tesselierten Oberfläche eines Objekts eine Zugehörigkeit zu einer bestimmten Klasse zugewiesen werden. Experimentelle Ergebnisse mit typischen Benchmarks zeigen, dass der vorgeschlagene Ansatz robust ist und für verschiedene 3D-Netze funktioniert [187].

#### 2.4.4 Shape Google

Feature-basierte Ansätze sind in der «Computer Vision» sehr beliebt, werden aber erst durch Shape Google ([195]) auf Problemstellungen aus dem Bereich der Formensuche von nicht starren Körpern übertragen.

Bisherige Ansätze ermöglichen die Beschreibung von Bildern durch eine Ansammlung von «visuellen Worten» welche mittels Text Suche nach dem sog. «Bag of features»-Ansatz weiterverarbeitet werden [196]. Shape Google verwendet hierfür einen Multiskalen-Feature-Descriptor (vgl. [197]). Dieses Vorgehen verwendet Wärmekern (heat kernel), welche in engem Zusammenhang mit Diffusionsgeometrien [198] und spektralen Methoden der Formanalyse stehen (z. B. [176], [199], [200] und [201]).

Die Schwierigkeit in der Überführung von Feature-basierenden Ansätzen von zweidimensionalen Bildern auf dreidimensionale non-rigid Geometrien sind vielseitig. Aus Geometrien können für gewöhnlich weniger Features extrahiert werden als aus Bildern (vgl. [195]). Folglich ist es schwieriger eine große Anzahl an stabilen und wiederholgenauen Features aus Geometrien zu extrahieren und die räumliche Abhängigkeit der ermittelten Features ist deutlich höher als bei ermittelten Features aus Bildern [195].

Shape Google extrahiert entsprechende Features aus den Geometrien und führt eine Analyse zur Generierung von Bag of Features durch. Es handelt sich um einen Vertreter aus Bereich der Ähnlichkeitsanalyse. Bereits hierdurch können bestimmte Geometrien voneinander unterschieden werden, manche ermittelten Features eignen sich jedoch nicht um ähnliche Geometrie voneinander zu unterscheiden. In [196] wird dies für Geometrien von menschlichen Körpern und die Körper von Zentauren verdeutlicht. Durch die Ergänzung der räumlichen Sensitivität der einzelnen Features können auch ähnliche Geometrien voneinander unterschieden werden. Mit Shape Google können Geometrien schlussendlich effizient durch Binärcode repräsentiert werden, gleichzeitig sind die ermittelten Features der Geometrien robust gegenüber geometrischen Adaptionen [196].

#### **2.4.5 Spektrale Convolutional Neural Networks**

Nach Masci et al. [202] haben die Arbeiten von Coifman und Lafon [198] und Levy [203] über Diffusion und Spektralgeometrie zu Entstehung von intrinsischen spektralen Formdeskriptoren geführt, welche durch Ihre Umsetzung einerseits kompakt (dense) und andererseits isometrisch-invariant sind. Prominente Vertreter hierfür sind heat kernel signatures (HKS) [197] und wave kernel signatures (WKS) [204].

Eine Umsetzung von spektralen CNN setzt Masci [202] mit GCNN (Geodesic CNN) um, hierbei handelt es sich um eine Erweiterung von CNNs auf nicht-euklidische Bereiche (manifolds), basierend auf lokalen geodätischen Koordinatensystemen. Der Vorteil dieses Vorgehens liegt in der Generalisierbarkeit, es können bestimmte Geometrien und Formen angelernt werden und das Vorgehen dann auf weitere unbekannte Geometrien und Formen übertragen werden. Das beschriebene Verfahren ist dem Bereich der Ähnlichkeitsanalyse zuzuordnen.

Die Architektur der Neuronalen Netze wird verändert, indem andere Schichten verwendet werden. Nach einem linearen Eingangslayer werden Geodesic-Convolutionlayer eingesetzt, anschließend erfolgt Angular-Max-Pooling, Fourier transform magnitude Layer und Kovarianzlayer. Spektrale Deskriptoren entstehen durch die nacheinander aufbauende Kombination aus diesen Schichten. Eine Adaption der Methode auf Punktwolken beschreibt [202] als möglich, setzt diese jedoch nicht um. Bruna et al. [205] wenden spektrale CNN auf Graphen an.

### 2.4.6 PointNet und PointNet++

Im Gegensatz zu den vorherigen Ansätzen, verwenden PointNet und PointNet++ Punktwolken als Eingangsdaten für die Neuronalen Netze. Damit verwenden diese beiden Vertreter einen deutlich anderen Ansatz als die bisher vorgestellten Netze und auch gegenüber Vorgehen, bei denen Punktwolken zunächst aufwändig voxelisiert werden. Bei Voxeln handelt es sich um eine Menge von Würfeln, beschrieben durch deren Eckpunkte. Mittels Voxeln können Geometrien ähnlich der Methode der Finiten Elemente angenähert werden. Ein weiterer Vorteil liegt in der Überführung einer beliebigen Geometrie in eine fixe Menge an Voxeln (hierfür muss die Kantenlänge der Voxel an die Geometrie angepasst werden oder die Geometrie entsprechend auf einen vorgegebenen Größenbereich skaliert werden).

PointNet von Qi et al. extrahiert mittels unterschiedlicher Optimierungsfunktionen bzw. Kriterien bedeutende Punkte aus der vorgegebenen ungeordneten Punktwolke [177]. Durch die letzten Schichten des Netzwerks kann dann mittels dieser extrahierten Punkte eine Formklassifikation oder Formsegmentierung umgesetzt werden. Es bestehen drei wesentliche Eigenschaften der initialen Punktwolke:

- Ungeordnete Punktwolke: Ein Vertauschen der einzelnen Punkte muss möglich sein
- Interaktion der Punkte untereinander: Einzelne Punkte sind nicht isoliert zu betrachten, sondern benachbarte Punkte haben einen geometrischen Zusammenhang
- Invarianz gegenüber Transformationen: Eine Rotation oder Translation der gesamten Punktwolke darf keinen Einfluss auf das spätere Ergebnis des Netzes haben

Bild 20: Aufbau PointNet zur Klassifikation dreidimensionaler Punktwolken in  $k$ -Klassen nach [177] verdeutlicht den Aufbau des PointNet mit den diversen integrierten Neuronalen Netzen (entweder als «T-Net» oder als «KNN» bezeichnet). Eine übergebene Eingangsmatrix ( $n$  Punkte mit jeweils  $x$ -,  $y$ - und  $z$ -Koordinaten) wird mittels Eingangstransformation verarbeitet. Das Ziel dieser Transformation ist es, ein Abbild der Punktwolke zu generieren, welches unabhängig von geometrischen Transformationen der ursprünglichen Punktwolke ist. Hierdurch sollen beispielsweise Translationen oder Rotationen der Punktwolke eliminiert werden. Bei der Transformation handelt es sich um eine einfache Matrixoperation, wobei die Parameter dieser Transformationsmatrix mittels eines Neuronalen Netzes (T-Net) auf Basis des Ursprungsdatensatzes erzeugt werden. Jeder Datenpunkt wird anschließend einem identischen Neuronalen Netz mit

einer höheren Anzahl an verdeckten Neuronen übergeben, so entsteht die Feature Matrix der Größenordnung  $n \cdot 64$ . Die erzeugten Features werden erneut einer Transformation unterzogen, das Vorgehen ist identisch zur Eingangstransformation, allerdings mit einer deutlich größeren Transformationsmatrix. Das Ziel dieser Transformation ist es, die Features von unterschiedlichen Punktwolken für die nachfolgenden Operationen einheitlich auszurichten. Die transformierte Featurematrix wird anschließend einem weiteren KNN mit nochmals erhöhter Neuronenanzahl in den einzelnen Schichten übergeben. Die so erhaltene Matrix der Größe  $n \cdot 1.024$  kann mittels max pooling auf eine Größe von  $1 \cdot 1.024$  reduziert werden. Hierbei handelt es sich um die globalen Features der übergebenen Punktwolke, die Feature sind unabhängig von der ursprünglich übergebenen Anzahl an Punkten. Eine Klassifikation in  $k$ -Klassen erfolgt durch ein weiteres KNN. Der vorgestellte Ansatz kann noch für Segmentierungen der Punktwolke erweitert werden, an dieser Stelle wird hierauf verzichtet und es wird auf [177] verwiesen.

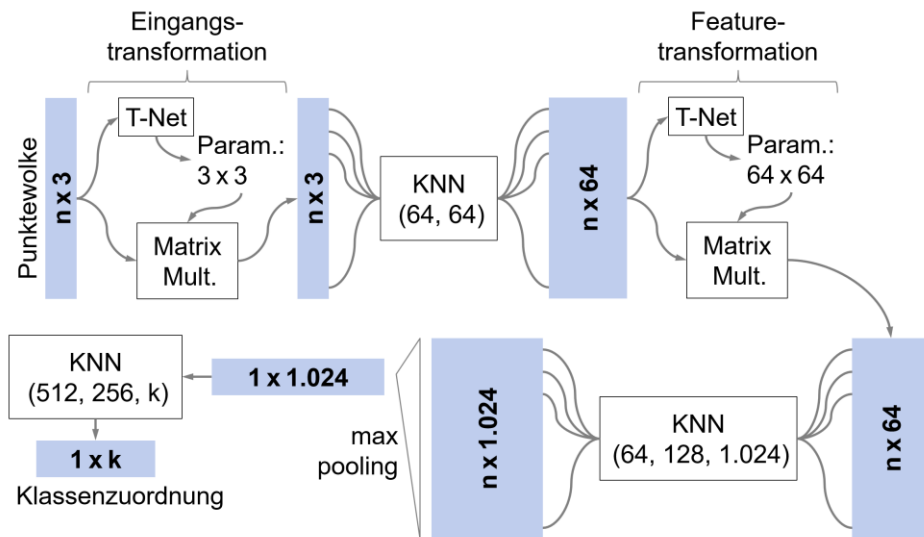


Bild 20: Aufbau PointNet zur Klassifikation dreidimensionaler Punktwolken in  $k$ -Klassen nach [177]

PointNet erfasst jedoch keine lokalen Strukturen, die durch die zugrundeliegende Punktwolke durch ihre Lage im Raum induziert werden, hierdurch wird die Fähigkeit feingranulare Muster zu erkennen und die Generalisierbarkeit eingeschränkt [206]. Aus diesem Grund wurde PointNet++ als Erweiterung durch die Autoren von PointNet entworfen. Es handelt sich um ein hierarchisches Neuronales Netz, welches PointNet rekursiv anwendet.

Die Funktionsweise von PointNet++ beschreiben die Autoren Qi et al. folgendermaßen: Die hierarchische Struktur von PointNet++ besteht aus einer Reihe von festgelegten Abstraktionsebenen. Auf jeder Abstraktionsebene wird ein Teil von Punkten verarbeitet und abstrahiert, um einen neuen Teil mit weniger Elementen zu erhalten.

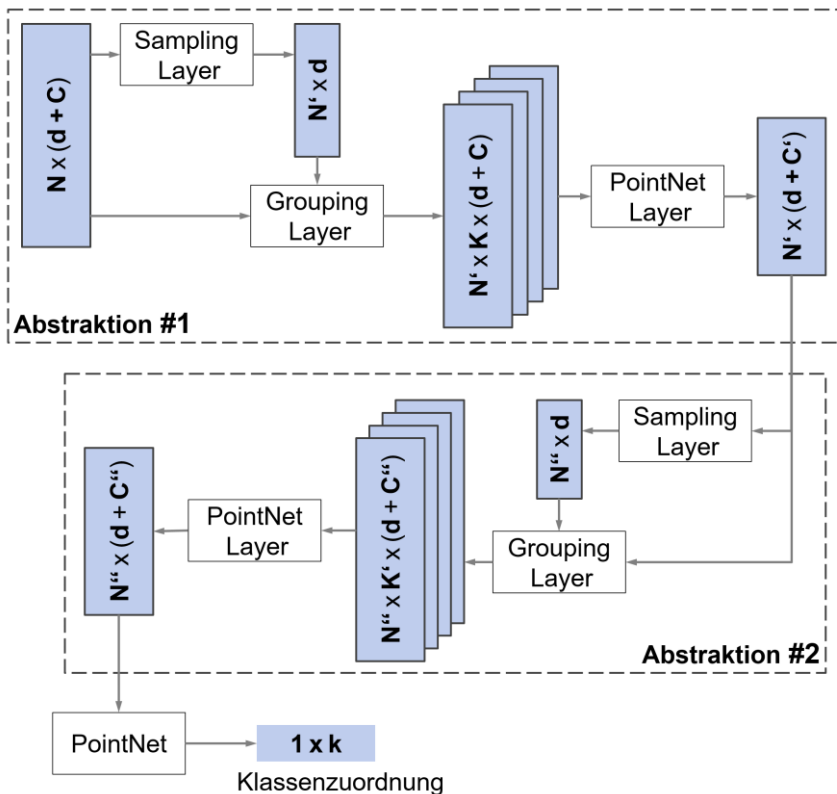


Bild 21: Aufbau PointNet++ zur Klassifikation von  $d$ -dimensionalen Punktwolken in  $k$ -Klassen nach [206]

Abstraktionsebene besteht aus drei Schichten: Sampling Layer, Grouping Layer und PointNet Layer. Das Sampling Layer selektiert einen Teil von Punkten, diese bilden die Mittelpunkte von lokalen Bereichen/Clustern. Das Grouping Layer ordnet benachbarte Punkte den zuvor selektierten Mittelpunkten zu. Das PointNet Layer verwendet ein Mini-PointNet, um lokale Muster in Featurevektoren zu kodieren. PointNet++ ermöglicht die intelligente Aggregation von multidimensionalen Informationen auf Basis der lokalen Dichte an Punkten der zugrundeliegenden Punktwolke, die Punktwolke muss hierfür keine einheitliche Dichte an Punkten aufweisen, sondern kann in der Auflösung variieren. Der grundlegende Aufbau von PointNet++ ist in Bild 21 veranschaulicht.

Der Ausgangspunkt ist ein Punktwolke der Dimension  $N \cdot (d + C)$  ( $C$  sind zu einem Punkt gehörige Features, beispielsweise eine gemittelte Entfernung zu einer bestimmten Anzahl an räumlich nahe liegenden Punkten). Durch das Sampling Layer werden  $N$  Punkte selektiert, welche den Mittelpunkt von  $N$  Clustern darstellen. Das Grouping Layer ordnet die initialen Punkte den ermittelten Clustern zu. Die Anzahl an Punkten je Cluster kann von Cluster zu Cluster variieren. Durch die Übergabe der so erzeugten Matrix an ein PointNet Layer wird die Dimension der Matrix auf  $N \cdot (d + C)$  reduziert. Durch die Übergabe an ein oder mehrere Abstraktion-Einheiten kann die Größe sukzessive reduziert werden. Schlussendlich kann durch ein PointNet eine Klassifikation in  $k$ -Klassen bzw. eine Segmentierung erfolgen.

### 2.4.7 Kurze Betrachtung der genannten Methoden

Es kann gezeigt werden, dass bereits eine gewisse Anzahl an Methoden existieren, um Geometriedaten unterschiedlicher Form an Algorithmen des Maschinellen Lernens zu übergeben. Allen hier beschriebenen Ansätzen ist gemein, dass ausschließlich die Informationen der Geometrie oder Features Beachtung finden, die direkt aus der Geometrie abgeleitet werden können. Mit der Geometrie verknüpfte Werte können mit den hier beschriebenen Methoden nicht an Algorithmen des Maschinellen Lernens übergeben werden. Ein Beispiel für derartige Werte sind berechnete Spannungen im Rahmen einer FE-Simulation. Diese sind direkt mit der Geometrie verknüpft und sollten daher auch zusammen mit dieser im Rahmen einer Analyse an Algorithmen übergeben werden. Eine detailliertere Betrachtung der Schwächen der bisherigen Ansätze erfolgt in Kapitel 3.

## 2.5 Plausibilitätsprüfung

In der Produktentwicklung im Allgemeinen und in der virtuellen Produktentwicklung im Besonderen steht eine große Menge an Daten aus FE-Simulationen zur Verfügung. Hierbei handelt es sich um Daten, welche direkt mit einer Geometrie verknüpft sind und aus diesem Grund im Rahmen einer automatischen Auswertung auch direkt mit dieser zusammen an entstprechende Algorithmen des Maschinellen Lernens (z. B. Künstliche Neuronale Netze) übergeben werden sollten. Dies kann eine automatische Überprüfung von FE-Simulationen bezüglich bestimmter Kriterien ermöglichen. Ein Vertreter einer solchen Überprüfung ist die Plausibilitätsprüfung.

Um zu begreifen, was unter einer Plausibilitätsprüfung für FE-Simulationen zu verstehen ist, gilt es, zunächst einige Begriffe zu erläutern. Im Zusammenhang mit numerischen Simulationen und auch der Generierung von mathematischen Vergleichsmodellen ist es wichtig, die Begriffe Verifikation und Validierung voneinander abzugrenzen. Ein Beispiel für ein derartiges mathematisches Vergleichsmodell oder Metamodell sind Künstliche Neuronale Netze (siehe auch Kapitel 2.2). In Kapitel 2.5.1 gilt es, zunächst eine Abgrenzung der Begrifflichkeiten zu definieren und anschließend den Begriff der Plausibilitätsprüfung in diesen Kontext einzusortieren. Dies umfasst ebenso die Einordnung des Begriffs «Plausibilitätsprüfung» für Simulationsergebnisse.

### 2.5.1 Abgrenzung «Validierung», «Verifikation» und «Plausibilität»

Die Abgrenzung der Begriffe «Validierung» und «Verifikation» soll anhand von gängigen Normen erfolgen (ASME V&V 10 [207], ISO 9000 [208] und DIN 25478 [209]).

ISO 9000 definiert «Validierung» als die „Bestätigung durch Bereitstellung eines objektiven Nachweises, dass die Anforderungen für einen beabsichtigten Gebrauch oder eine spezifische beabsichtigte Anwendung erfüllt worden sind“. Hierbei kann der objektive Nachweis durch das Ergebnis eines Tests oder anderweitig erbracht werden, z. B. durch alternative Berechnungen oder Überprüfung von Dokumenten. Die Anwendungsbedingungen für die Validierung können sowohl echt als auch simuliert sein. [vgl. 208]

DIN 25712 stellt klar, dass ein Berechnungssystem für jeden Anwendungsfall zu validieren ist. „Die Validation eines Berechnungssystems muss

grundsätzlich basieren auf Vergleichen der Ergebnisse der Nachberechnung von Benchmarks mit den Messergebnissen dieser Benchmarks“. DIN 25712 nennt Benchmarks und Referenzergebnisse aus dem Bereich der Nukleartechnik (z. B. Messung von Neutronenmultiplikation oder Reaktivitätswerte von Neutronenabsorbern).

Der Begriff der Verifizierung wird in ISO 9000 als „Bestätigung durch Bereitstellung eines objektiven Nachweises, dass festgelegte Anforderungen erfüllt worden sind“ definiert. Der genannte objektive Nachweis wird identisch wie bei der Validierung definiert.

DIN 25712 definiert die Verifizierung im Kontext von Berechnungssoftware auf Rechnern: „Nach seiner Installation auf einem Rechner ist ein Berechnungssystem zwecks Qualitätssicherung zu verifizieren. Eine erneute Verifikation ist erforderlich, wenn Änderungen am Berechnungssystem, am Compiler oder am Betriebssystem vorgenommen werden. Die zu einer Verifikation durchgeführten Prüfungen und die dabei erhaltenen Ergebnisse sind entsprechend zu dokumentieren. Die zur Verifikation durchzuführenden Prüfungen bestehen üblicherweise in der Nachrechnung von Standardbeispielen (im Allgemeinen vom Entwickler des Berechnungssystem zur Verfügung gestellt), die oft auf Benchmarks oder theoretischen Benchmarks beruhen.“

Nach ASME V&V<sup>10</sup> lassen sich die beiden Begriffe «Verifikation» und «Validierung» anhand eines Prozesses für strukturmechanische FE-Simulationen abgrenzen (siehe Bild 22). Ausgangspunkt ist ein zu untersuchender Zustand eines technischen Systems, z. B. eines Bauteils, welches einer bestimmten mechanischen Belastung unterzogen werden soll. Mittels Abstraktion entsteht das «konzeptionelle Modell» des zu untersuchenden Systems.

Bei der Abstraktion werden fundamentale Annahmen getroffen, welche die Interpretation der späteren Ergebnisse beeinflussen. Typische Annahmen sind die Anzahl an zu betrachtenden Komponenten, die Ansätze zur Materialmodellierung, die Vereinfachung der Geometrie, die Kontakteinstellungen zu weiteren Bauteilen oder sich selbst in der Simulation sowie die Definition und Aufbringung der Randbedingungen (Lagerungen, angreifende Kräfte, usw.). Anschließend erfolgt eine Aufteilung in die beiden Bereiche «mathematische Modellierung» und «physikalische Modellierung». Im Bereich der mathematischen Modellierung wird das «mathematische Modell» aufgestellt, welches das «konzeptionelle Modell» in mathematische Gleichungen und Formeln überführt.

Das «Berechnungsmodell» ist die numerische Implementierung des «mathematischen Modells», welches mittels Rechnereinsatz gelöst werden soll.

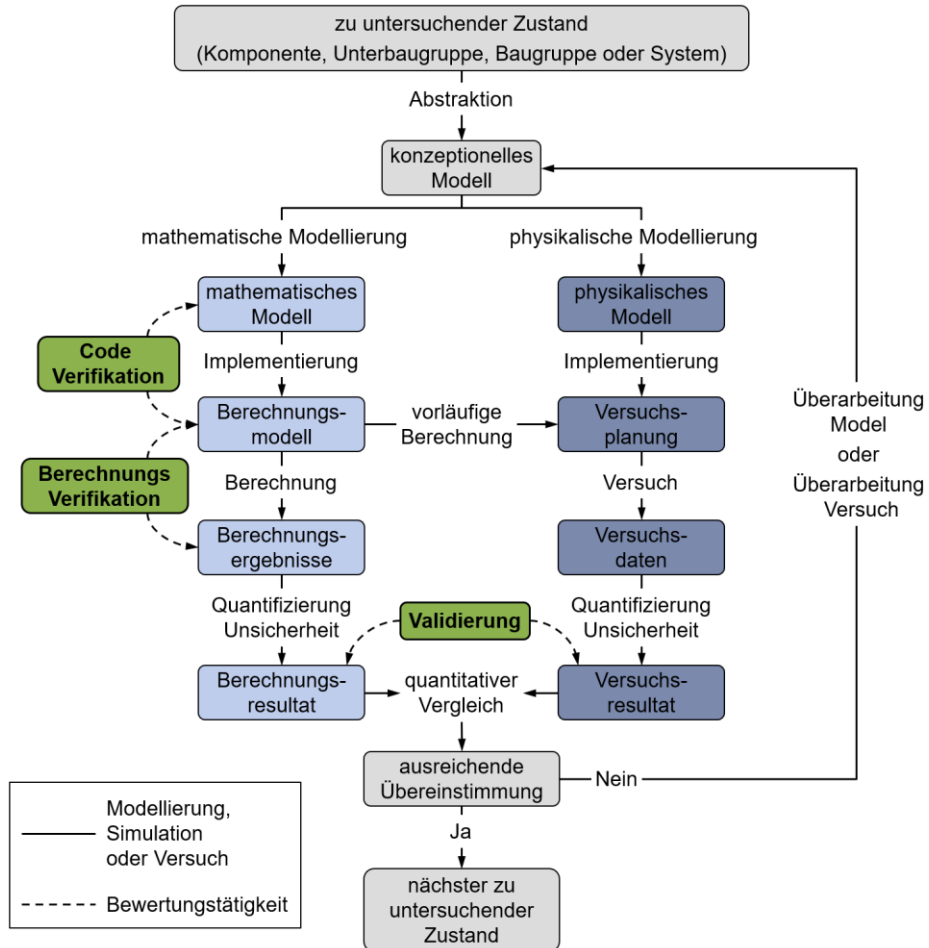


Bild 22: Abgrenzung der Begriffe «Verifikation» und «Validierung» nach [207]

Mittels «Code Verifikation» wird die korrekte Überführung des «mathematischen Modells» in das «Berechnungsmodell» überprüft. Nach der Berechnung bzw. dem Lösen der mathematischen Gleichungen steht das «Berechnungsergebnisse» zur Verfügung. Der Abgleich zwischen «Berechnungsmodell» und «Berechnungsergebnisse» wird als «Berechnungsverifikation» verstanden. Durch die Berücksichtigung bzw. Quantifizierung der Unsicherheiten entsteht das «Berechnungsergebnisse» und auf der Seite der physikalischen Modellierung das «Versuchsergebnisse». Der Abgleich zwischen «Berechnungsergebnisse» und «Versuchsergebnisse» wird als

«Validierung» bezeichnet. Das Ziel der «Validierung» ist es, die Vorhersagegenauigkeit für den angedachten Verwendungsfall zu überprüfen. [207]

Unter Plausibilität wird ein Beurteilungskriterium für Aussagen verstanden. Böhnert & Reszke [210] betrachten den Begriff der Plausibilität bei der Entstehung von wissenschaftlichen Tatsachen und zeigen, dass der Begriff «plausibel» zwischen den Begriffen absurd und offenkundig einzuordnen ist. Sollte zu einer Position keine Alternative mehr vorliegen, so liegt mehr als nur Plausibilität vor [210], dies verdeutlicht Bild 23. Zusätzlich wird der Begriff der Plausibilität für Simulationsergebnisse ergänzt.

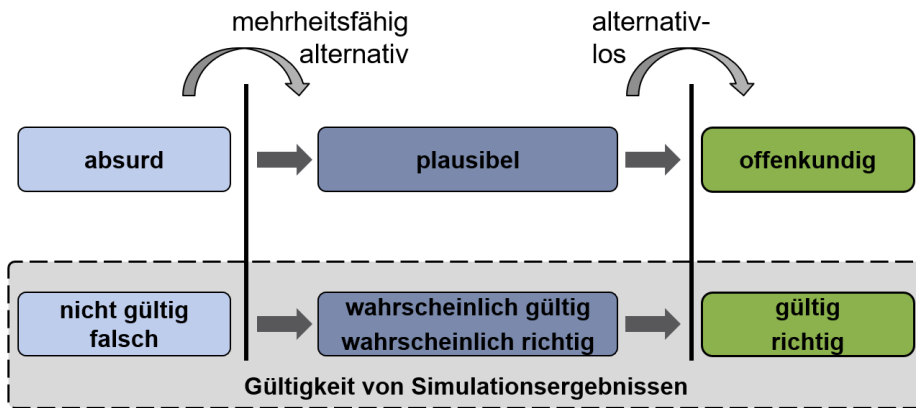


Bild 23: Abgrenzung des Begriffs «Plausibilität» nach [210], ergänzt um die Gültigkeit von Simulationsergebnissen

Darüber hinaus wird in VDI Richtlinie 3843 Blatt 1 [211] ein Test auf Plausibilität für die Modellbildung schwingungsfähiger Systeme mittels Eigenfrequenzen, Eigenschwingungsformen, Vergleichberechnungen, Abschätzungen und Grenzwertbetrachtungen erwähnt.

Liegen plausible Simulationsergebnisse vor, kann davon ausgegangen werden, dass keine offensichtlichen Fehler enthalten sind. Die Simulationsergebnisse sind also wahrscheinlich gültig bzw. richtig. [P13, P22]

Eine Plausibilitätsprüfung kann sowohl für Simulationsergebnisse als auch für Versuchsergebnisse verwendet werden. In [P13] und [P22] wurde die ASME V&V 10 um eine Plausibilitätsprüfung zwischen den Schritten der Verifikation und der nachgelagerten Validierung ergänzt (siehe Bild 24). Im Rahmen der Plausibilitätsprüfung erfolgt ein Abgleich mit einem Referenzwert, hierfür werden zusätzliche Daten benötigt.

Auf Basis dieser Daten kann, z. B. mittels Maschinellen Lernen, ein Vergleichswert ermittelt werden, wodurch wiederum die Ergebnisse des Versuchs oder der Simulation bewertet werden können.

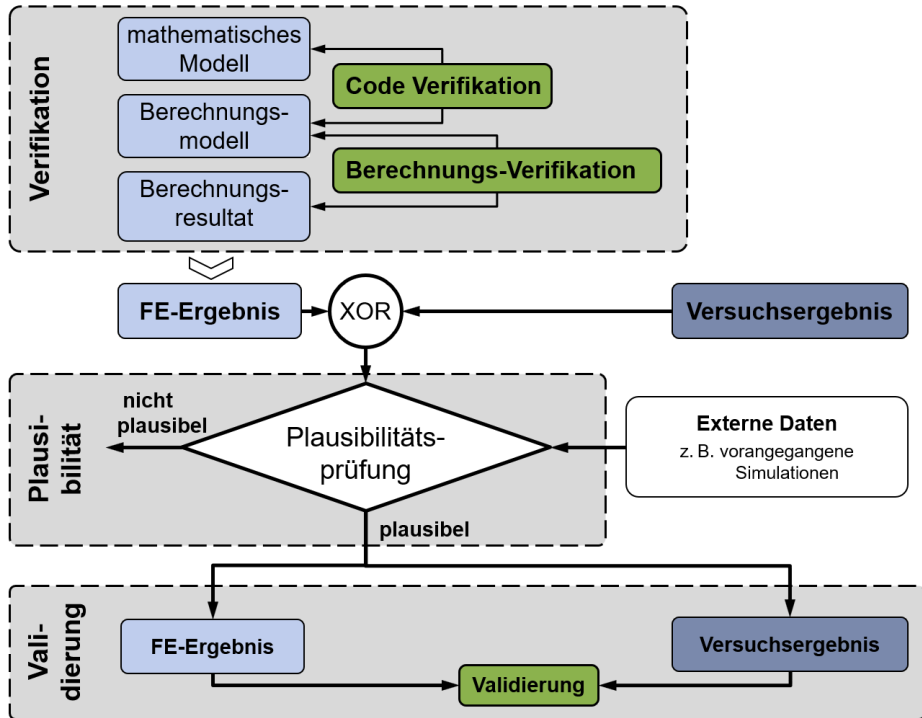


Bild 24: Ergänzung [207] um den Begriff «Plausibilitätsprüfung» nach [P13]



### 3 Ableitung des Handlungsbedarfs

Die Produktentwicklung umfasst heute sehr viele unterschiedliche Bereiche und sie unterliegt einem ständigen Wandel auf Grund von sich verändernden Anforderungen. Laut Joscha Ernst in [1] sind dies: geänderte Kundenanforderungen, technologisch höherwertige Produkte, zunehmende Modellvielfalt, höherer Funktionsumfang der Produkte, zunehmende Globalisierung sowie eine Internationalisierung der Produktentwicklung. Dies alles führt laut Ernst zu einem Anstieg der Komplexität in der Produktentwicklung. Dies lässt sich mit den drei Verben «schneller», «höher», «weiter» zusammenfassen. Gegenüber [1] hat die Bedeutung von Daten in allen Bereichen des Lebens stark zugenommen und der Fokus der Produktentwicklung muss sich vermehrt auf die Nutzung aktueller und bestehender Daten aus vorangegangenen Entwicklungen richten (Digital Engineering).

Die durchgängige Nutzung von Daten in allen Bereichen der Produktentwicklung bietet erhebliches Potential zur Steigerung der Qualität und zur Verkürzung der benötigten Entwicklungszeit. Hierfür muss im Idealfall das gesamte Wissen aus vorangegangenen Entwicklungen kontextsensitiv bereitgestellt werden. Bedingt durch riesige Datenmengen und verteilte Speicherung sehr unterschiedlicher Daten ist dies ohne den Einsatz von Datenanalysemethoden unmöglich. Großes Potential wird in den Algorithmen des Maschinellen Lernen mit der Möglichkeit zur Generalisierung des verfügbaren Wissens gesehen. Derartige Algorithmen können also nicht nur bestimmte Ergebnisse von Datensätzen aus der Vergangenheit zur Verfügung stellen (inklusive einer gewissen Unsicherheit/Schwankung), sondern sie können auch Aussagen über einen unbekanntem Datensatz liefern, welcher zuvor nie zur Verfügung stand (also die Interpolation bzw. Extrapolation von Daten).

Hierfür muss es im Idealfall allerdings möglich sein, alle Daten für Algorithmen des Maschinellen Lernen nutzbar zu machen. Dies ist leider in vielen Bereichen der Produktentwicklung nicht durchgängig für alle Daten und über verschiedene Entwicklungsprozesse von unterschiedlichen Bauteilen und Baugruppen hinweg möglich. Oftmals können nur bestimmte Bestandteile übernommen werden und große Mengen an Daten liegen bisher ungenutzt auf Laufwerken zur Langzeitsicherung. In vielen Fällen erfolgt die kostenintensive Speicherung nur aus juristischen Gründen der Nachweispflicht. Zur weiteren Verbesserung und Beschleunigung der Produktentwicklung wäre es daher sinnvoll, die große Menge an verfügbaren Daten für die Entwicklung neuer Produkte nutzbar zu machen.

Im Bereich der Produktentwicklung sind viele Daten direkt oder indirekt mit Geometrien verknüpft, beispielweise CAD-Daten, Simulationsdaten, Scan-Daten, Versuchsergebnisse, Ergebnisse von abweichungsbehafteten Toleranzanalysen, und viele mehr. Um diese Datenbestände aus vorangegangenen Entwicklungen für neue Produkte und deren Produktentwicklungsprozesse nutzbar zu machen, bedarf es eines universell nutzbaren Verfahrens zur Übertragung von beliebigen Geometriedaten und daran angehängter zusätzlicher Informationen an Algorithmen des Maschinellen Lernens (z. B. Deep Neural Networks). Diese Algorithmen sind in der Lage, auch widersprüchliche Daten zu verarbeiten, und ermöglichen die Abbildung einer sehr großen Menge an Datensätzen und des darin enthaltenen Wissens in sehr kompakten, relativ kleinen Modellen (bezüglich des Umfangs der ursprünglichen Daten).

Gegenwärtig existieren aus akademischer Sicht folgende Einschränkungen für die durchgängige Nutzung geometriebasierter Daten in der Produktentwicklung:

- **Einheitliche Orientierung:**

In der Produktentwicklung kommt es häufig vor, dass identische Bauteile mehrfach in unterschiedlicher Translation und Rotation verwendet werden, beispielsweise bei Normteilen (z. B. Schrauben, Muttern, Unterlegscheiben, Wellensicherungsringe, Profilverbinder, Scharniere etc.) in umfangreichen CAD-Baugruppen. Innerhalb von CAD-Systemen sind die Orientierungen der Bauteile bekannt, bei der Weitergabe an andere Programme (FE-Programme) oder bei der Umwandlung in neutrale Datenaustauschformate (.step-Dateien, .jt-Dateien, etc.) können diese Informationen jedoch verloren gehen. Gerade historische Daten werden häufig in neutralen Austauschformaten archiviert. Zur Herbeiführung einheitlicher Orientierungen existieren erste Ansätze [212], allerdings sind diese Methoden häufig sehr aufwändig oder erfolgen händisch. Außerdem liegt der Fokus der Entwicklung nicht auf Daten aus der Produktentwicklung. Andere Algorithmen wie die Hauptkomponentenanalyse können einheitliche Orientierungen für unsymmetrische Bauteile sehr effizient erzeugen, allerdings kann es zu unerwünschten Ausrichtungen bei der Verwendung von symmetrischen Bauteilen kommen (dies verdeutlicht Kapitel 4.2.1).

- **Dimensionsreduktion:**

Bisherige Ansätze zur Geometrieerkennung bzw. -klassifikation und der dafür notwendigen Dimensionsreduktion beziehen sich auf

Datensätze, welche kaum Bezug zur Produktentwicklung haben. Beispiele hierfür sind die ModelNet10- und ModelNet40-Datensätze, welche Objekte aus dem Bereich der Computergrafik bzw. abstrahierte Scans von realen Objekten enthalten. Dies Bauteile liegen aber lediglich als beschreibende Geometrie (off-Dateien) oder als voxelisierte Geometrie vor. Diese beiden Arten von Geometriebeschreibungen kommen in Prozessketten der virtuellen Produktentwicklung nur sehr selten oder gar nicht vor.

Es muss folglich eine Methode entwickelt werden, um typische Datenformate der Produktentwicklung in ihrer Dimension zu reduzieren.

- **Aufbau einer großen Datenbasis mit gelabelten Datensätzen:** Umfangreiche Datensätze aus dem Bereich der Produktentwicklung (bestehend aus Eingangs- und Ausgangsgrößen) sind im akademischen Bereich kaum verfügbar. Zumeist existieren kleine Simulationsstudien und akademische Demonstratoren oder Konstruktionen von Prüfständen, welche häufig Geheimhaltungsvereinbarungen mit Firmen unterliegen. Größere Datenmengen existieren zumeist nur als Prüfstandsergebnisse von unterschiedlichen Versuchsreihen. Diese vorliegenden Daten können keinesfalls mit dem Umfang verfügbarer Daten im industriellen Umfeld verglichen werden. Eine große Menge an gelabelten Daten liegt in der Regel weder im akademischen noch im industriellen Umfeld direkt verfügbar vor. Häufig müssen unterschiedliche Datensätze oder IT-Systeme verknüpft werden, um derartige Daten zu erhalten. Um die entwickelten Methoden in dieser Arbeit testen und evaluieren zu können, muss eine entsprechend umfangreiche Datenbasis an gelabelten Daten aufgebaut werden.
- **Adaption von Algorithmen des Maschinellen Lernen an die Anforderungen der vorliegenden Datensätze:** Verwendete Algorithmen gilt es, an die vorliegende Aufgabe anzupassen. Dies betrifft einerseits die Anpassung von Modellen (z. B. Netzarchitektur von Neuronalen Netzen) und andererseits die Anpassung der Vorverarbeitung der Daten (z. B. die Normierung). Einige Quellen beschreiben die Schritte der Vorverarbeitung als oft vernachlässigt [213], zeitintensiv [214] jedoch zugleich als notwendig [215] und sehr lohnenswert [214, 216]. Im Rahmen der vorliegenden Dissertation werden keine neuen Algorithmen des Maschinellen Lernen entwickelt, vielmehr geht es

darum entsprechende Vorverarbeitungsschritte für bereits existierende Algorithmen zu entwickeln. Mittels dieser Vorverarbeitung und Normierung können geometriebasierte Daten aus der Produktentwicklung an unterschiedliche Algorithmen des Maschinellen Lernen für Analsen übergeben werden.

Aus den dargelegten Punkten ergeben sich folgende Herausforderungen, welche im Rahmen dieser Dissertation näher beleuchtet und im Rahmen von vier Forschungsfragen erörtert werden:

- Es gilt einen Vorverarbeitungsschritt für die Übergabe von geometriebasierten Daten oder Daten, welche direkten Bezug zur Geometrie aufweisen, an Algorithmen des Maschinellen Lernen zu entwickeln. Insbesondere die Verknüpfung von Daten mit der Geometrie und die Übergabe an Algorithmen ist bisher nicht anwendbar, da keine entsprechenden Ansätze existieren. Die vorliegende Arbeit beschäftigt sich sowohl mit der Berücksichtigung unterschiedlich orientierter, aber geometrisch identischer Bauteile als auch mit der Dimensionsreduktion von beliebigen geometrisch ausgeprägten Bauteilen auf eine einheitliche Form.  
➔ **Wie kann die Übergabe von Geometrieinformationen und mit der Geometrie verknüpften Kenngrößen an Algorithmen des Maschinellen Lernens erfolgen?**
- Es erfolgt die Darlegung eines Vorgehens zur Umsetzung einer derartigen Methodik inklusive aller relevanten Teilschritte. Dies ermöglicht es anderen Personen, Forschungseinrichtungen und Unternehmen (auch kleinen), die entwickelte Methodik anzuwenden und so von den Vorteilen neuartiger datengetriebener Methoden in der Entwicklung zu profitieren. Hierfür werden ebenfalls Aspekte für das Training und die Adaption der Algorithmen des Maschinellen Lernen auf die vorliegenden Daten beleuchtet und detailliert beschrieben.  
➔ **Welche Schritte sind notwendig, um eine Prozesskette von der Datenerstellung bis hin zur Datenanalyse aufzubauen und wo liegen potenzielle Schwierigkeiten?**
- Zur Darlegung der Umsetzbarkeit werden drei unterschiedliche Anwendungsbeispiele entlang des Produktentstehungsprozesses aufgebaut sowie die Anwendung der entwickelten Methodik und

das Training bzw. die Evaluierung entsprechender Modelle dargelegt. Durch die Wahl unterschiedlicher Anwendungsbeispiele soll in dieser Arbeit die Übertragbarkeit der entwickelten Methode auf unterschiedliche Schritte entlang des Produktentstehungsprozesses aufgezeigt werden. Die einzelnen Beispiele unterscheiden sich dabei erheblich bzgl. des Umfangs der Originaldaten (Geometriebeschreibung) und der zusätzlich mit der Geometrie verknüpften Informationen.

→ **Welche Adaptionen an der Methodik sind notwendig, um unterschiedliche Anwendungsfälle und damit unterschiedliche Datenstände mit dem beschriebenen Vorgehen zu verarbeiten?**

- Das übergeordnete Ziel besteht in der Nutzbarmachung von Datenbeständen des Produktentstehungsprozesses für den Entwicklungsprozess von neuen Produkten. Durch die Berücksichtigung und Nutzbarmachung von Daten wird die virtuelle Produktentwicklung erweitert und es erfolgt die Transformation in Richtung des Digital Engineering. Hierfür soll die vorliegende Arbeit Anstöße und Ansätze zur Umsetzung aufzeigen.  
→ **Welche Schritte sind notwendig, um die entwickelte Methodik sowohl in der Industrie als auch im akademischen Bereich einzusetzen.**

Im Rahmen der vorliegenden Arbeit soll ein Beitrag zur durchgängigen Datennutzung in der Produktentwicklung und -entstehung (im Rahmen des Digital Engineering) geleistet werden. Der Autor widmet sich der Umwandlung von Geometriepäsentationen zur Verarbeitung mittels Algorithmen des Maschinellen Lernen und der Adaption von Algorithmen in drei Anwendungsbeispielen.



## 4 Methodik der sphärischen Detektorflächen

Die Methodik der sphärischen Detektorflächen stellt eine Dimensionsreduktionsmethode für dreidimensionale Punktwolken dar. Hiermit können beliebig große und beliebig orientierte dreidimensionale Punktwolken in zweidimensionale numerische Matrizen überführt werden. Dieses Vorgehen wird in Kapitel 3 vorgestellt und die einzelnen Teilschritte weiter detailliert beleuchtet. Als Anschauungsobjekt dienen die in Bild 25 gezeigten Zylinderkopfschrauben nach ISO 4762 – M10x50 in acht unterschiedlichen Orientierungen im Raum. Die Methodik der sphärischen Detektorflächen ermöglicht es diese acht Bauteile in eine einheitliche Orientierung zu transformieren. Es handelt sich um CAD-Modelle, welche mittels ANSYS Workbench 19.2 einheitlich, mit 10-knotigen Tetraeder-Elementen mit einer Elementgröße von 1 mm, vernetzt wurden. Jede Zylinderkopfschraube wird durch 73.620 dreidimensionale Punkte in kartesischen Koordinaten repräsentiert.

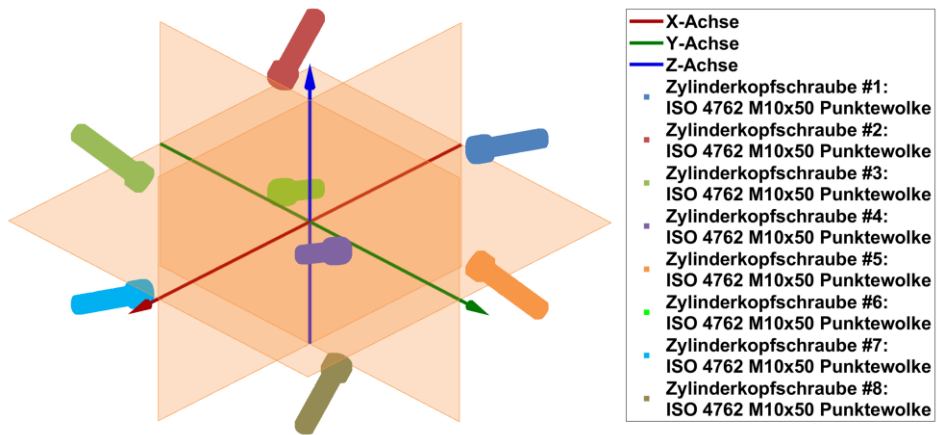


Bild 25: Zylinderkopfschrauben ISO 4762 – M10x50 in 8 unterschiedlichen Orientierungen

### 4.1 Gesamtübersicht

Im Folgenden wird ein Überblick über die Methode der sphärischen Detektorflächen gegeben. Eine Übersicht über die Hauptschritte und die jeweils anwendbaren Methoden und Einteilungsmöglichkeiten der Detektorfläche zeigt Bild 26. Zunächst gilt es, eine übergebene dreidimensionale Punktwolke in kartesischen Koordinaten in eine einheitliche Orientierung zu

überführen, dies erfolgt mit drei unterschiedlichen Methoden in Kapitel 4.2. Anschließend wird der Aufbau der Detektorfläche, inklusive zwei alternativen Einteilungen der sphärischen Fläche, in Kapitel 4.3 vorgestellt. Abschließend erfolgt die Generierung der Detektormatrix inklusive der Auffaltung der sphärischen Detektorfläche in eine zweidimensionale numerische Matrix. Dies erfolgt in Kapitel 4.4 für die zuvor dargestellten Zylinderkopfschrauben aus Bild 25. Hierbei werden auch die unterschiedlichen Methoden der Orientierung und die unterschiedliche Einteilung der Detektorfläche durchiteriert. Somit ergeben sich 6 unterschiedliche numerische Matrizen für jede der 8 Zylinderschrauben als Abbildung der dreidimensionalen Punktwolke der Zylinderkopfschrauben.

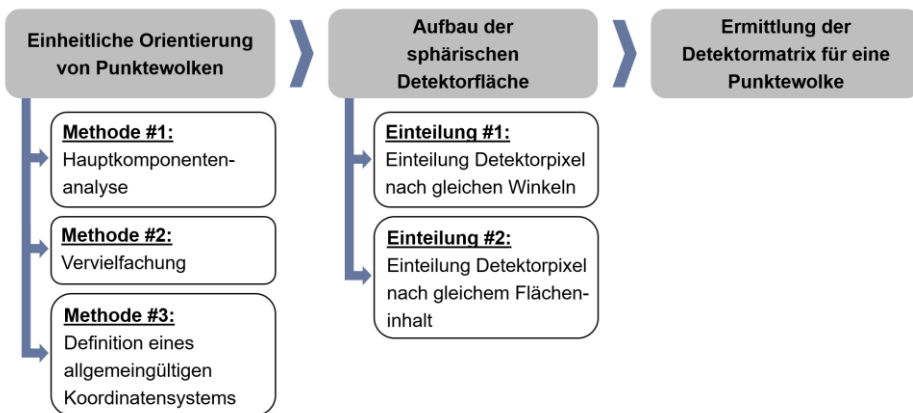


Bild 26: Übersicht der Methodik zur Generierung einer Detektormatrix aus einer dreidimensionalen Punktwolke

## 4.2 Einheitliche Orientierung von Punktwolken

In typischen Konstruktionen des Maschinenbaus werden identische Bauteile, wie z. B. Normteile, mehrfach an komplett unterschiedlichen Stellen einer Baugruppe verbaut. So werden beispielsweise komplett identische Schrauben mehrfach verwendet, um ein Bauteil mit einem anderen zu verschrauben. Dieser Umstand hat auch Einfluss auf die Übergabe von Geometrierepräsentationen an Algorithmen des Maschinellen Lernen. Aus diesem Grund gilt es zunächst einheitliche Orientierungen von gleichen Bauteilen im dreidimensionalen Raum herbeizuführen. Im Folgenden werden drei unterschiedliche Methoden vorgestellt, um Bauteile einheitlich zu orientieren. Den Startpunkt stellt die Hauptkomponentenanalyse in Kapitel 4.2.1 dar. Diese bildet die Grundlage für die beiden, darauf aufbauenden

Methoden der künstlichen Vervielfachung (Kapitel 4.2.2) und einer eigenen Methode zur Definition eines allgemeingültigen Koordinatensystems in Kapitel 4.2.3.

#### 4.2.1 Methode #1: Hauptkomponentenanalyse (PCA)

Die zentrale Idee der Hauptkomponentenanalyse besteht darin, die Dimensionalität eines Datensatzes, der aus einer großen Anzahl voneinander abhängiger Variablen besteht, zu reduzieren und gleichzeitig die im Datensatz vorhandene Streuung so weit wie möglich zu erhalten. Dies wird durch die Transformation der ursprünglichen Variablen in einen neuen Satz aus Variablen erreicht. Die neuen Variablen, die Hauptkomponenten, sind voneinander unabhängig und so angeordnet, dass die ersten wenigen Variablen die meiste der in allen ursprünglichen Variablen enthaltenen Streuung beinhalten. [217]

Die Hauptkomponentenanalyse ist ein Verfahren der multivariaten Statistik, das versucht, den ursprünglichen Datensatz mittels Linearkombinationen genähert abzubilden. Anhand des vorliegenden Beispiels der dreidimensionalen Punktwolken soll kurz die Hauptkomponentenanalyse (Principal Component Analysis → PCA) aufgezeigt werden.

Das Ziel der Hauptkomponentenanalyse ist es, eine Eingangsmatrix mittels einer Transformation in eine Ausgabematrix zu überführen. Eine kurze mathematische Beschreibung liefert Gleichung (20).

$$M' = (M_{original} - \overline{M_{original}}) \cdot koeff \quad (20)$$

$M'$  Matrix der Ausgangsdaten der PCA

$M_{original}$  Matrix der Eingangsdaten der PCA

$\overline{M_{original}}$  arithmetischer Mittelwert der Matrix  $M_{original}$

$koeff$  Rotations- und Translationsmatrix der PCA

Mit diesem Vorgehen werden die in Bild 25 dargestellten 8 Zylinderkopfschrauben eingedreht und ergeben die in Bild 27 dargestellten Punktwolken. Es ist ersichtlich, dass keine konsistent einheitliche Orientierung entsteht, da identische Bauteile noch gespiegelt an den Hauptebenen vorliegen können. In Bild 27 tritt eine ungewollte Spiegelung an der Hauptebene zwischen Y- und Z-Achse auf. Dies betrifft z. B. die uneinheitliche Orientierung der beiden Zylinderkopfschrauben #1 und #6.

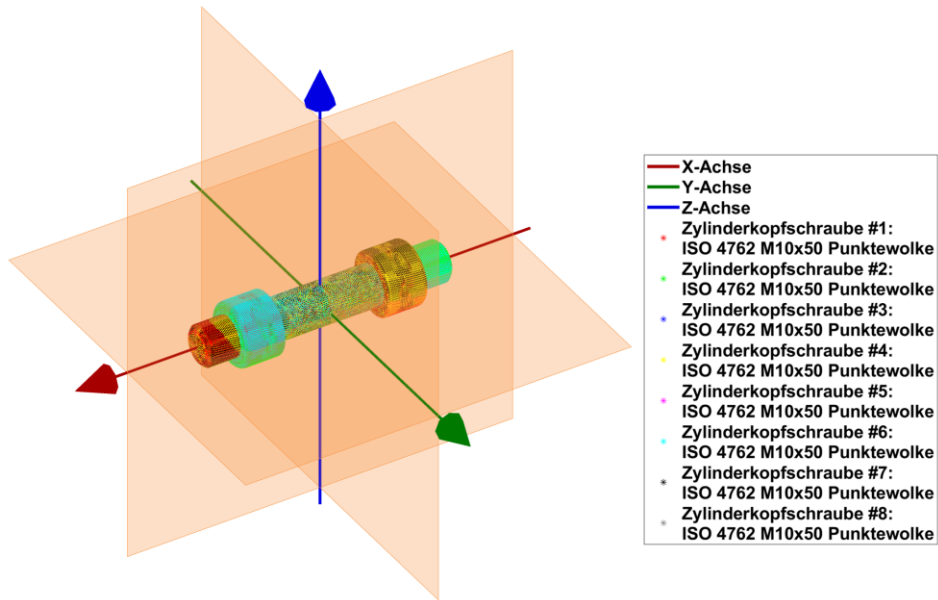


Bild 27: Eingedrehte Bauteile nach der PCA

Es ist ersichtlich, dass eine Hauptkomponentenanalyse allein noch nicht zu einer eindeutigen Orientierung von beliebig rotierten und verschobenen Bauteilen führt. Aus diesen Gründen erfolgt durch Methode #2 (Kapitel 4.2.2) und Methode #3 (Kapitel 4.2.3) eine Erweiterung der Hauptkomponentenanalyse, um eine eindeutige Orientierung zu erhalten.

#### 4.2.2 Methode #2: Vervielfachung

Zwei identische Bauteile, welche ursprünglich in unterschiedlichen Quadranten des aufgespannten Raumes liegen und mittels der Hauptkomponentenanalyse eingedreht werden, können um jede der 3 Koordinatenachsen um einen Winkel von  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  oder  $270^\circ$  verdreht sein. Ob und in welche Richtung ein Bauteil verdreht ist, hängt von der Form der Punktwolke und der ursprünglichen Orientierung im Raum ab. Zwei der in Bild 27 dargestellten Zylinderkopfschrauben sind lediglich an der Y-Z-Ebene gespiegelt/Verdrehung um die Z-Achse. Somit ergeben sich  $4^3 = 64$  mögliche Orientierungen. Methode #2 erzeugt im Anschluss an die Hauptkomponentenanalyse künstlich diese zusätzlichen Orientierungen. Die Punktwolke jedes Bauteils wird also 63-mal rotiert, anschließend werden alle so

erzeugten Punktwolken zu einer großen Punktwolke kombiniert. Dieses Vorgehen ist in Bild 28 für die beiden Zylinderkopfschrauben #1 und #6 im Anschluss an die Hauptkomponentenanalyse aus Bild 27 durchgeführt.

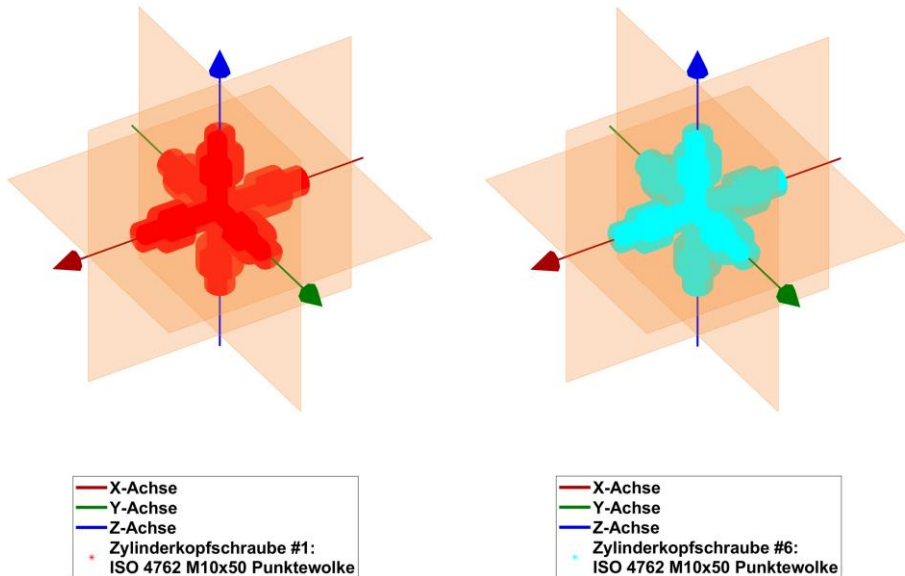


Bild 28: Eingedrehte Zylinderkopfschrauben #1 und #6 nach Methode #2 (Vervielfachung)

Es ist ersichtlich, dass so eine einheitliche Orientierung der Bauteile möglich ist - allerdings zu Lasten der Größe der Punktwolke. Zusätzlich kommt es zu einer Vermischung von Geometrien im Bereich der Überlagerung der Bauteile. Es muss folglich mit einem gewissen Informationsverlust in diesen Bereichen gerechnet werden.

### 4.2.3 Methode #3: Definition eines allgemeingültigen Koordinatensystems

Ähnlich wie Methode #2 setzt Methode #3 [S4] direkt nach der Hauptkomponentenanalyse an. Es wird zunächst der arithmetische Schwerpunkt  $m$  der Punktwolke nach Formel (21) für jede der drei Koordinatenrichtungen in Kartesischen Koordinaten berechnet. Die vorliegende Punktwolke wird in den Nullpunkt verschoben, der Schwerpunkt der Punktwolke liegt somit im Koordinatenursprung mit  $x = 0$ ,  $y = 0$  und  $z = 0$ .

Anschließend werden die Minimal- und Maximalwerte der drei Koordinatenrichtungen ermittelt und aus diesen der ungemittelte Schwerpunkt

nach Formel (22) gebildet. Sollten der gemittelte Schwerpunkt und der ungemittelte Schwerpunkt identisch bzw. nur geringfügig voneinander abweichen (Abstand in x-, y- und z-Koordinaten  $< 0,04 \text{ mm}$ , werden nach dem Zufallsprinzip 200 Punkte aus der Punktwolke entfernt und anschließend erneut die Schwerpunkte berechnet. Dieser Sonderfall tritt bei gängigen Normteilen in der Regel nicht auf, ist zur Vollständigkeit aber in der Methode #3 enthalten.

$$m = [\bar{x} \quad \bar{y} \quad \bar{z}] \quad (21)$$

$\bar{x}$  arithmetischer Mittelwert der x-Koordinate

$\bar{y}$  arithmetischer Mittelwert der y-Koordinate

$\bar{z}$  arithmetischer Mittelwert der z-Koordinate

$$\tilde{m} = \left[ \frac{\min x + \max x}{2} \quad \frac{\min y + \max y}{2} \quad \frac{\min z + \max z}{2} \right] \quad (22)$$

Der ungemittelte Schwerpunkt  $\tilde{m}$  befindet sich in einem von 8 Oktanten des Kartesischen Koordinatenraums.

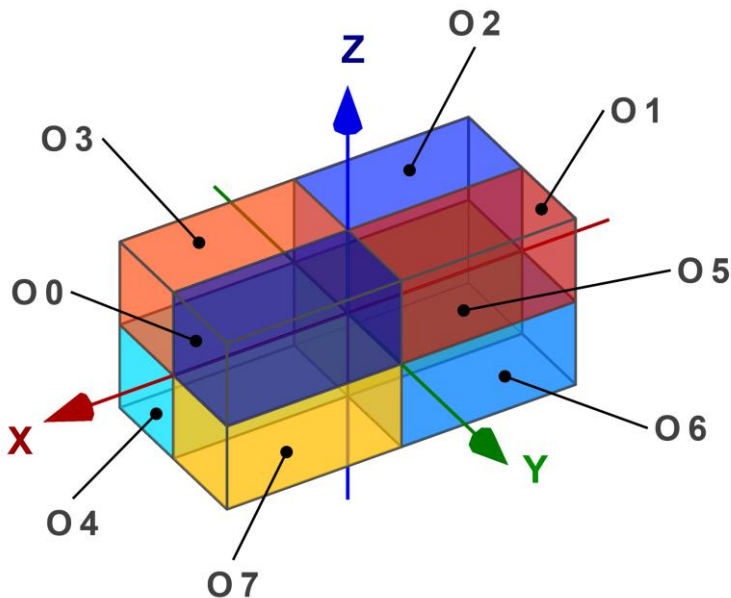


Bild 29: Definition der Oktanten mit Nummerierung in Gray-Code-Ordnung

Die Zuordnung der Oktanten erfolgt durch Vorzeichen-Paare in Gray-Code-Ordnung [218] und ist in Bild 29 dargestellt. Sollte  $\tilde{m}$  bereits im nullten Oktanten anzutreffen sein, ist keine weitere Spiegelung notwendig und das Bauteil bzw. die Punktwolke ist bereits korrekt ausgerichtet. Sollte sich  $\tilde{m}$  in einem anderen Oktanten befinden, kann mit den Transformationsvorschriften aus Tabelle 4 die Punktwolke so lange an unterschiedlichen Koordinatenebenen gespiegelt werden, bis  $\tilde{m}$  sich im nullten Oktanten befindet.

Tabelle 4: Verwendete Transformationsvorschrift für die Oktanten

Oktant	Vorzeichen (x-, y-, und z-Achse)	Verwendete Transformationsvorschrift
#1	(+, +, +)	▪ keine Transformation
#2	(-, +, +)	▪ Spiegelung an der Y-Z-Ebene ▪ Spiegelung an der X-Z-Ebene
#3	(-, -, +)	▪ Spiegelung an der X-Z-Ebene
#4	(+, -, +)	▪ Spiegelung an der X-Y-Ebene ▪ Spiegelung an der X-Z-Ebene
#5	(-, -, -)	▪ Spiegelung an der Y-Z-Ebene ▪ Spiegelung an der X-Z-Ebene ▪ Spiegelung an der X-Y-Ebene
#6	(-, +, -)	▪ Spiegelung an der Y-Z-Ebene ▪ Spiegelung an der X-Y-Ebene
#7	(+, +, -)	▪ Spiegelung an der X-Y-Ebene

Dies ermöglicht eine einheitliche Orientierung aller Zylinderkopfschrauben aus Bild 25. Die Orientierungen der beiden Schrauben #1 und #6 sind nach dem Ablauf von Methode #3 identisch und sind in Bild 30 zu erkennen. In dieser Darstellung sind die roten Punkte von Zylinderkopfschraube #1 größer dargestellt, aus diesem Grund ist ein roter Rand um die Punkte von Zylinderkopfschraube #6 erkennbar, dies dient ausschließlich Visualisierungsgründen. Somit steht mit Methode #3 ein Vorgehen zur Verfügung, welches es ermöglicht, einheitliche Orientierungen von Punktwolken zu erzeugen und gleichzeitig die Anzahl der Punkte in der Punktwolke nicht zu verändern.

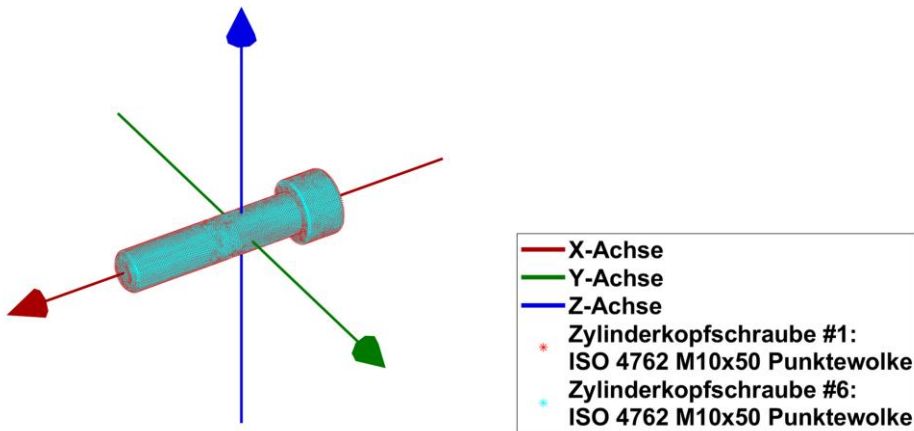


Bild 30: Identisch orientierte Zylinderkopfschrauben #1 und #6 nach Methode #3

### 4.3 Aufbau der sphärischen Detektorfläche

Im Anschluss an die einheitliche Orientierung der Punktwolken gilt es, eine Methode zur Dimensionsreduktion anzuwenden, welche einerseits unterschiedlich umfangreiche und andererseits räumlich anders angeordnete Punktwolken auf eine fixe Anzahl von Kenngrößen reduziert. Dies ist der Hauptbestandteil der Methode der sphärischen Detektorflächen. Zunächst soll kurz der Hintergrund beleuchtet werden und anschließend in Kapitel 4.3.1 auf eine Einteilung mit gleichen Winkeln und in Kapitel 4.3.2 auf eine Einteilung mit konstanter Fläche der einzelnen Detektorpixel eingegangen werden.

Im ersten Schritt der Methode erfolgt eine Umwandlung der Kartesischen  $x$ -,  $y$ - und  $z$ -Koordinaten der Punktwolke in sphärische  $\varphi$ -,  $\theta$ - und  $r$ -Koordinaten. Anschließend wird eine sphärische Fläche um das Bauteil herum aufgespannt. Der Ursprung der Kugel liegt dabei im Nullpunkt des Koordinatensystems der Punktwolke. Dies ist gleichzeitig auch der gewichtete Schwerpunkt der Punktwolke. Auf der Oberfläche der Sphäre ist eine definierbare Anzahl an Flächen (Detektorpixel) angeordnet. Die Anzahl der Pixel ist willkürlich, definiert aber den Grad der Dimensionsreduktion der Punktwolke. Werden  $36 \times 36$  Pixel verwendet, erfolgt im späteren Verlauf eine Dimensionsreduktion auf  $36 \cdot 36 = 1.296$  numerische Werte. Durch die Umwandlung der kartesischen Koordinaten in Kugelkoordinaten ist die Zuordnung eines jeden Punktes der Punktwolke zu einem bestimmten Pixel auf der Oberfläche der Sphäre nur noch von den beiden Winkeln  $\varphi$  und  $\theta$  abhängig. Der Radius  $r$  hat hierfür keine Bedeutung und

es ist somit völlig unbedeutend, wie groß die Punktwolken/Bauteile innerhalb der Sphäre sind bzw. wie groß die Sphäre definiert werden muss. Diese ist im weiteren Verlauf nur aus Visualisierungsgründen um das Bauteil herum angeordnet. Zur weiteren Visualisierung soll erneut die Zylinderkopfschraube #1 aus Bild 30 herangezogen werden.

#### 4.3.1 Einteilung Detektorpixel nach gleichen Winkeln

Eine Einteilung der sphärischen Detektorfläche kann durch die Vorgabe einer konstanten Winkelaufteilung für beide Winkel  $\varphi$  und  $\theta$  erfolgen. Für eine Einteilung in  $36 \times 36$  Pixel gilt es die Winkel in  $\Delta\varphi = \Delta\theta = 10^\circ$  Abschnitte einzuteilen.

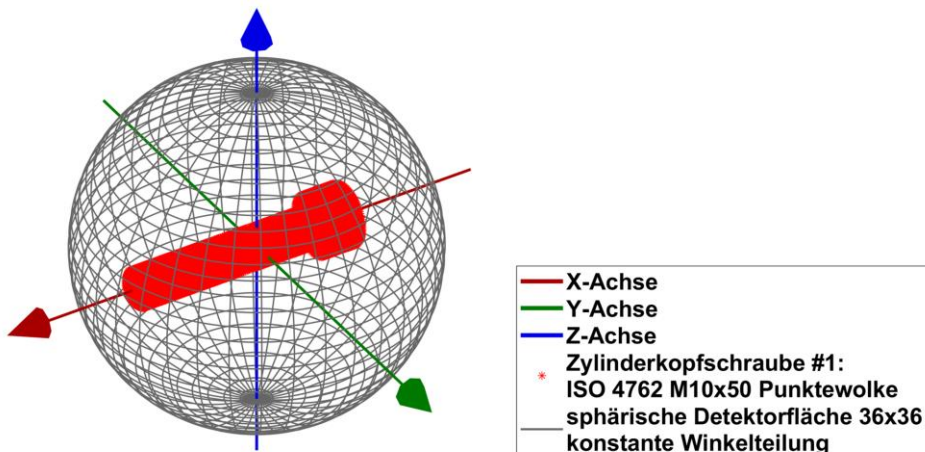


Bild 31: Zylinderkopfschrauben #1 mit sphärischer Detektorfläche  $36 \times 36$  mit konstanter Winkelteilung

Dies ist in Bild 31 mit einer aufgespannten Detektorfläche mit  $36 \times 36$  Pixeln mit konstanter Winkeinteilung und Zylinderkopfschraube #1 im Inneren dargestellt.

Es ist ersichtlich, dass die Größe der einzelnen Pixel sehr unterschiedlich ausfällt. Insbesondere ein Vergleich zwischen Pixeln am oberen oder unteren Rand der Sphäre mit denen im Bereich des «Äquators» zeigt erhebliche Unterschiede und hat Auswirkungen auf die Erzeugung der späteren Detektormatrix in Kapitel 4.4.

### 4.3.2 Einteilung Detektorpixel nach gleichen Flächen

Im Gegensatz zur Einteilung der Winkel nach konstanten Winkelabschnitten kann eine Einteilung der Winkel auch nach konstanter Fläche der einzelnen Detektorpixel erfolgen. Soll eine  $36 \times 36$ -Sphäre erzeugt werden, wird zunächst die Oberfläche eines  $10^\circ$ -Winkelsegments nach Formel (23) berechnet. Anschließend kann die Fläche eines einzelnen Pixels mit Formel (24) berechnet werden.

$$A_{10^\circ\text{-Segment}} = r^2 \cdot \int_{\varphi=0}^{\frac{2\cdot\pi}{36}} d\varphi \cdot \int_{\theta=0}^{\pi} \sin \theta d\theta = \frac{r^2 \cdot \pi}{9} \quad (23)$$

$r$  Radius

$\varphi$  Azimuthwinkel

$\theta$  Polarwinkel

$A$  Fläche

$$A_{\text{Pixel}} = \frac{A_{10^\circ\text{-Segment}}}{36} = \frac{r^2 \cdot \pi}{324} \quad (24)$$

Der zugehörige Azimuthwinkel  $\varphi$  jedes Pixels ist bereits durch die Einteilung in  $10^\circ$ -Segmente vorgegeben. Für das erste Segment ergeben sich die beiden Grenzen von  $\varphi = 0^\circ$  und  $\varphi = 10^\circ$  bzw.  $\varphi = \frac{2\cdot\pi}{36}$ .

$$A_{\text{Pixel}} = r^2 \cdot \int_{\varphi=0}^{\frac{2\cdot\pi}{36}} d\varphi \cdot \int_{\theta_n}^{\theta_{n+1}} \sin \theta d\theta \quad (25)$$

$$\begin{aligned} \theta_{n+1} &= \arccos\left(-\frac{r^2 \cdot \pi \cdot 36}{324 \cdot r^2 \cdot 2\pi} + \cos \theta_n\right) \\ &= \arccos\left(-\frac{1}{18} + \cos \theta_n\right) \end{aligned} \quad (26)$$

Der Polarwinkel  $\theta$  kann durch das Auflösen von Formel (25) nach  $\theta_{n+1}$  berechnet werden. Dies ist in Formel (26) gezeigt und in Bild 32 grafisch verdeutlicht.

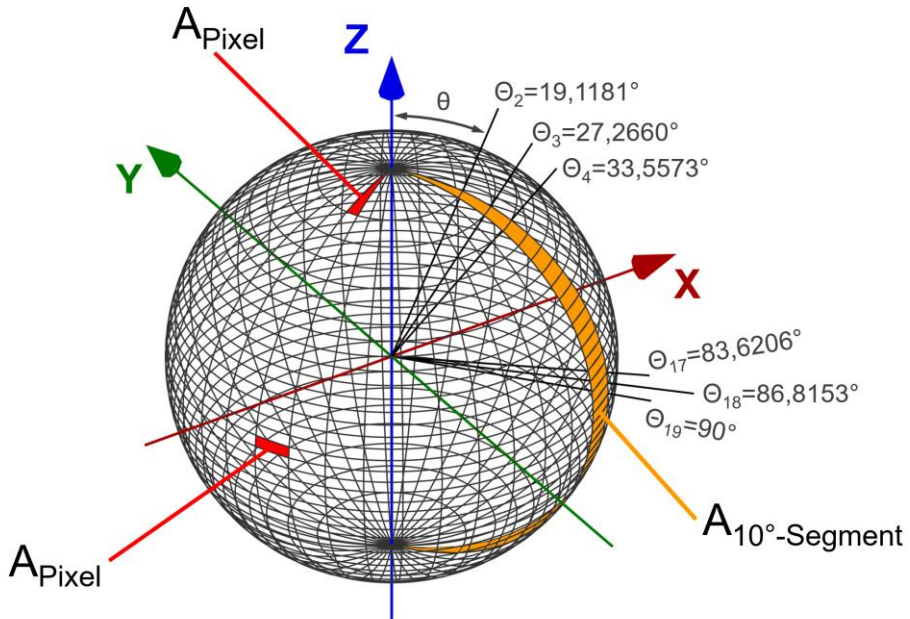


Bild 32: Zuordnung der Flächeninhalte auf einer sphärischen Detektorfläche mit 36x36 Pixeln

Tabelle 5:  $\theta$ -Winkel für eine sphärische Detektorfläche mit 36x36 Pixeln, gerundet auf 4 Nachkommastellen

1. Segment	$\theta_n$	$\theta_{n-1}$	$\theta_{n+1} - \theta_n$
Pixel 1	0°	19,1881°	19,1881°
Pixel 2	19,1881°	27,2660°	8,0779°
Pixel 3	27,2660°	33,5573°	6,2913°
...	...	...	...
Pixel 17	83,6206°	86,8153°	3,1946°
Pixel 18	86,8153°	90°	3,1847°
...	...	...	...
Pixel 35	152,7339°	160,8119°	8,0779°
Pixel 36	160,8119°	180°	19,1881°

Die 36 Pixel des ersten Winkelsegments mit den zugehörigen  $\theta$ -Winkeln sind in Tabelle 5 aufgelistet. Durch die Vorgabe der Randbedingung einer konstanten Pixelfläche ergeben sich an den «Polen» deutlich größere Winkelabstände des  $\theta$ -Winkels als bei den nahe am «Äquator» gelegenen Pixeln, dies wird durch die Differenz der beiden Winkel ( $\theta_{n+1} - \theta_n$ ) in der vierten Spalte von Tabelle 5 klar ersichtlich (vgl. auch Bild 33).

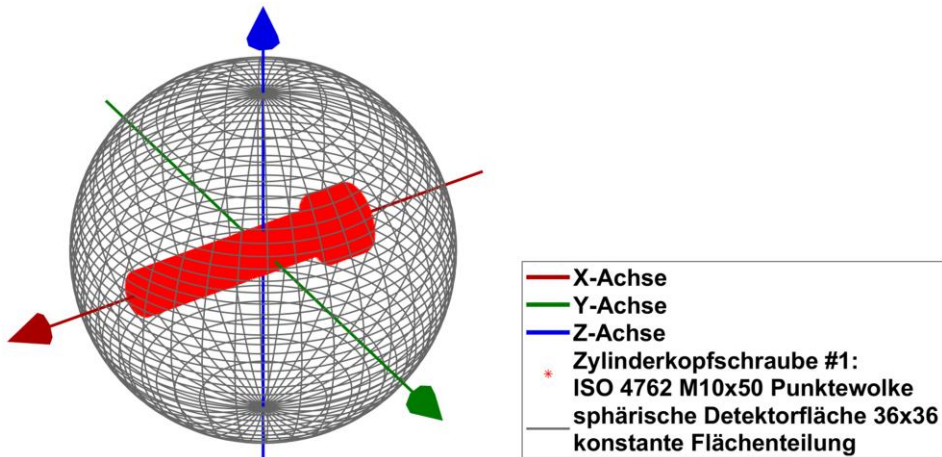


Bild 33: Zylinderkopfschraube #1 mit sphärischer Detektorfläche, 36x36 Pixel mit identischem Flächeninhalt

#### 4.4 Ermittlung der Detektormatrix für eine Punktwolke

Der Ablauf zur Erzeugung der Detektormatrix für eine Punktwolke ist anhand eines vernetzten Blocks durch fünf Schritte (von a) bis e) in Bild 34 dargestellt. Nach dem Aufspannen der Detektorfläche (Bild 34b) werden alle Punkte des Bauteils auf die Oberfläche der Sphäre projiziert. Der Ursprung der Sphäre ist der Massenschwerpunkt der Punktwolke. Folglich existieren einzelne Pixel, in die mehrere Punkte projiziert werden, und andere mit deutlich weniger oder gar keinen Punkten (vgl. Bild 34c). Die projizierten Punkte in jedem Pixel werden aufsummiert und gespeichert. Anschließend kann die Oberfläche der dreidimensionalen Sphäre, ähnlich wie die Abwicklung der Erdkugel als Weltkarte, in eine zweidimensionale Darstellung überführt werden (Bild 34d). Der Zahlenwert der Anzahl der projizierten Punkte in jedem Pixel kann somit in eine numerische Matrix in der Größe der Detektorpixel der Detektorfläche überführt werden (Bild 34e).

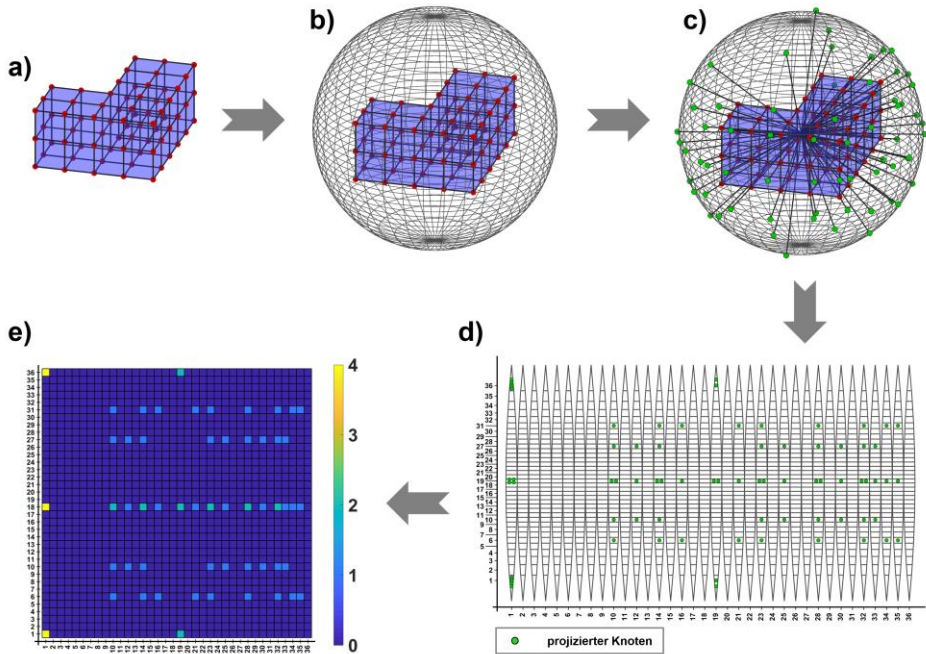


Bild 34: Erzeugung der Detektormatrix in 5 Schritten

Die Umwandlung einer dreidimensionalen Punktwolke in eine zweidimensionale numerische Matrix (die Detektorfläche) bildet das Ergebnis der vorgestellten Methodik. Diese ermöglicht mittels Dimensionsreduktion die Übergabe an Algorithmen des Maschinellen Lernens. Exemplarisch sind die beiden Matrizen für die Zylinderkopfschraube mit konstanter Winkелеinteilung der Pixel (vgl. Bild 31) und mit konstanter Flächeneinteilung der Pixel (vgl. Bild 33) in Bild 35 dargestellt. In dieser Darstellung ist zu beachten, dass durch die Farbskalierung der Schaft als solcher nicht in den Matrizen erkennbar ist. Nichtsdestotrotz erfolgt eine Projektion der Knoten des Schafts und die Werte zwischen dem Schraubenkopf und dem Ende des Gewindes sind größer null. Dies wird durch die beiden Vergrößerungen aus den Matrizen mittels Heatmap dargestellt.

Es ist erkennbar, dass die Werte oberhalb des Schraubenkopfes bei konstanter Pixelfläche zwar erheblich reduziert sind, jedoch weiterhin die Geometrie der Zylinderkopfschraube repräsentieren. Bei einem direkten Vergleich der beiden Einteilungen (konstanter Flächeninhalt bzw. konstante Winkелеinteilung) fällt auf, dass die Einteilung mit konstanter Pixelfläche in einer gleichmäßigeren Aufteilung der projizierten Punkte auf die einzelnen Detektorpixel resultiert. Dieses Verfahren sollte daher für die weiteren Schritte im Rahmen dieser Dissertation verwendet werden. Es

#### 4 Methodik der sphärischen Detektorflächen

erfolgt dadurch eine detailliertere Abbildung der Geometrie ohne Verzerrung bestimmter Bereiche. Eine derartige Verzerrung ist nicht sinnvoll, da die geometriebasierten Daten an Algorithmen des Maschinellen Lernens übergeben werden sollen und diese Algorithmen eine eventuell vorhandene Verzerrung nicht kompensieren sollen.

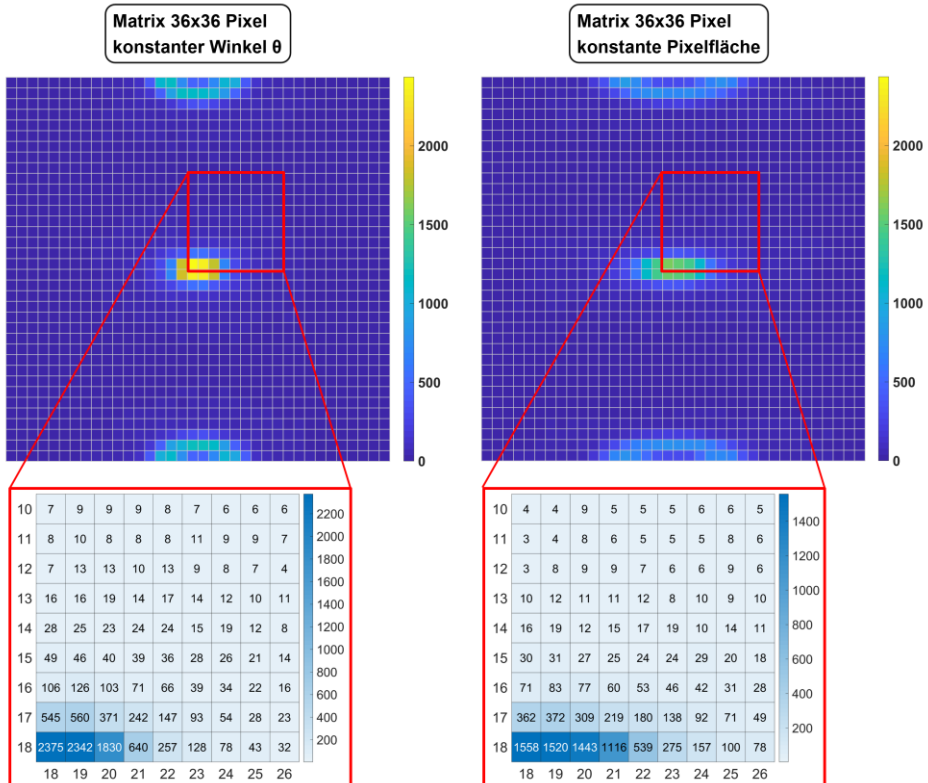


Bild 35: Detektormatrizen mit der aufsummierten Anzahl an projizierten Knoten pro Pixel

## 5 Anwendung #1: Bauteilerkennung

Mit der vorgestellten Methodik der sphärischen Detektorflächen ist die Umwandlung von dreidimensionalen Punktwolken in eine zweidimensionale numerische Matrix mit zuvor spezifizierter Größe möglich.

Zur Darstellung der Anwendbarkeit soll im Folgenden eine automatische Bauteilerkennung mittels dieser Methodik umgesetzt werden. Hierfür gilt es, zunächst die Problemstellung in Kapitel 5.1 zu beschreiben, den Aufbau der Datenbasis in Kapitel 5.2 vorzustellen und die Detektormatrizen zu erzeugen. In Kapitel 5.3 wird die Architektur des verwendeten Neuronalen Netzes beschrieben. Im Anschluss an die Ermittlung der Prognosegüten für verschiedene KNN Architekturen (Kapitel 5.4) wird ein Ansatz zur Erkennung von unbekanntem Bauteilen mittels Autoencodern in Kapitel 5.5 vorgestellt. Abschließend erfolgt in Kapitel 5.6 die Bewertung und Diskussion der Ergebnisse des ersten Anwendungsbeispiels.

### 5.1 Definition der Problemstellung

In der virtuellen Produktentwicklung werden sehr häufig Bauteile im CAD modelliert und in Baugruppen assembliert. Diese Baugruppen liegen normalerweise in CAD-spezifischen Formaten vor, welche bei jedem CAD-Software-Hersteller variieren. CAD-Modelle werden häufig in neutrale Austauschformate konvertiert, einerseits zum Austausch zwischen unterschiedlichen Software-Programmen und andererseits zur Langzeitsicherung von Modellen (beispielsweise im Kraftwerks-, Schiff- oder Flugzeugbau). Vor der Konvertierung kann entweder anhand einer gut gepflegten Bauteilbezeichnung oder anhand der CAD-Modellierungshistorie jedes einzelne Bauteil erkannt und klassifiziert werden. Nach der Umwandlung in neutrale Austauschformate ist die Modellierungshistorie (CAD-Modellbaum) der einzelnen Bauteile nicht mehr vorhanden und kann daher auch nicht für weitere Analysen herangezogen werden. Insbesondere auch die Langzeitsicherung von digitalen Modellen der Produktentwicklung erfolgt sehr häufig in den erwähnten neutralen Austauschformaten, wie beispielsweise *.step* oder *.jt*. Ähnliches gilt auch für erzeugte Vernetzungen von Finite-Elemente-Software, hierbei erfolgt die Abspeicherung von Netzinformationen häufig in *ASCII*-Textdateien oder im *.stl* Format. Manchmal kann es notwendig sein, diese Dateiformate und die darin enthaltenen Bauteile zuzuordnen bzw. zu klassifizieren. Anwendungsfälle können die

Suche nach Gleichteilen in vorhandenen Datenbeständen oder die Bauteilerkennung innerhalb von FE-Programmen sein.

Das Ziel besteht in der Überführung von Bauteilen aus neutralen Austauschformaten in eine maschinenlesbare und einheitliche Definition, um eine Bauteilerkennung anhand dieser Daten durchführen zu können.

## 5.2 Aufbau der Datenbasis und Erstellung der Detektormatrix

Für den Aufbau einer Datenbasis werden 73 CAD-Bauteile aus einer Normteillbibliothek (Autodesk Inventor Professional 2017) verwendet. Die genaue Auflistung der Bauteile kann Tabelle 26, Tabelle 27 und Tabelle 28 im Anhang entnommen werden. Es handelt sich um gängige Normteile des Maschinenbaus:

- Zylinderkopfschrauben
- Zylinderschrauben mit Schlitz
- Sechskantschrauben
- Sechskantschrauben mit Schaft
- Sechskantschrauben mit Flansch
- Spannstifte
- Zylinderstifte
- Sicherungsringe für Wellen
- Sicherungsringe für Bohrungen
- diverse Sechskantmuttern

Die 73 Bauteile werden als *.step*-Dateien aus dem CAD-System Inventor Professional 2017 exportiert und mittels ANSYS Workbench 18.0 mit 10-knotigen Tetraederelementen (ANSYS Elementtyp SOLID187) mit einer Kantenlänge von 0,5 mm einheitlich vernetzt. Die kartesischen Koordinaten der Knoten des FE-Netzes werden mittels der entwickelten Orientierungsmethode #3 aus Kapitel 4.2.3 ausgerichtet und anschließend mit der beschriebenen Methodik der sphärischen Detektorflächen mit konstanter Pixel-Fläche mit 36x36 Pixeln in zweidimensionale numerische Matrizen umgewandelt.

Somit stehen 73 Datensamples mit jeweils  $36 \cdot 36 = 1.296$  Werten als Trainingsdatensatz zur Verfügung. Bild 36 zeigt die Punktwolken der Bauteile und die erstellten Bauteilmatrizen für Zylinderkopfschrauben nach ISO 4762 – M6 x 20 und ISO 4762 – M8 x 20, für Spannstifte nach ISO 13337 – 6 x 20 und ISO 13337 – 8 x 20 und für Sicherungsringe für Bohrungen nach DIN 472 – 20 x 1,5 und DIN 472 – 50 x 2.

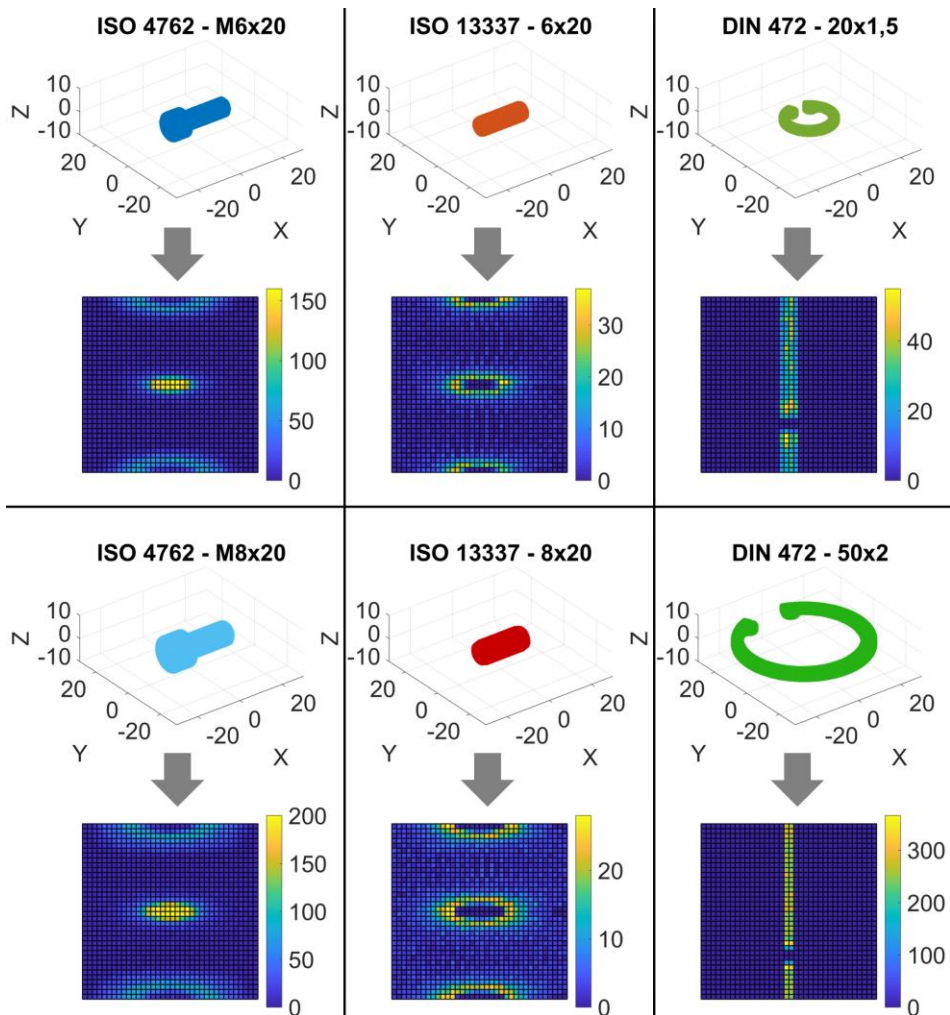


Bild 36: Ausgewählte Bauteile des Trainingsdatensatzes mit Punktwolken und abgeleiteten Detektormatrizen mit  $36 \times 36$  Pixeln

Zur Schaffung eines umfangreichen Testdatensatzes werden die 73 Bauteile als *.step*-Dateien erneut mittels ANSYS Workbench 18.0 einheitlich vernetzt. Die 73 Bauteile sind identisch mit den Bauteilen aus dem Trainingsdatensatz und bilden die Grundlage für weitere Manipulationsschritte.

Hierfür werden von jedem Bauteil 16 zufällige Orientierungen abgeleitet. Dabei wird das Bauteil um drei zufällige Winkel zwischen  $-90^\circ$  und  $+90^\circ$  sukzessive um die X-Achse, Y-Achse und zuletzt um die Z-Achse rotiert. Anschließend erfolgt eine Translation des Bauteils um 150 mm in allen drei Koordinatenachsen. Somit stehen für das Testen der trainierten Modelle  $73 \cdot 16 = 1.168$  Datensamples zur Verfügung. Ein Sample entspricht einem Datensatz, welcher nachgelagert an das Künstliche Neuronale Netz übergeben wird. Diese Datensätze werden ebenfalls mit der Orientierungsmethode #3 ausgerichtet und von diesen Punktwolken wird jeweils eine  $36 \times 36$ -Detektormatrix abgeleitet.

Es stehen somit 73 Datensätze (73 Normteile) für das Training und weitere  $73 \cdot 16 = 1.168$  Datensätze (73 Normteile) für das Testen zur Verfügung. Für die Erkennung von unbekanntem Bauteilen mittels Autoencodern (Kapitel 5.5) stehen weitere 8 unbekannte Bauteile zur Verfügung, diese werden später in Bild 41 kurz vorgestellt.

### 5.3 KNN Architektur und Training

Bei der Bauteilerkennung handelt es sich um eine Klassifikationsaufgabe, bei der Eingangsgrößen mit jeweils 1.296 Werten einer von 73 Klassen zuzuordnen sind. Jeder Klasse steht für ein eigenes Bauteil mit spezifischer Geometrie, folglich besitzt die Zylinderkopfschraube mit sieben geometrischen Ausprägungen sieben unterschiedliche Klassen. Aufgrund der vorliegenden Datenlage können unterschiedliche Klassifizierer aus dem Bereich des Machinellen Lernens verwendet werden. Exemplarisch soll im Folgenden eine Anwendung mittels Künstlicher Neuronaler Netze betrachtet werden. Aufgrund der beschränkten Anzahl von Eingangsgrößen sollen hierbei Neuronale Netze mit einer geringen Anzahl von verdeckten Neuronenschichten zur Klassifikation eingesetzt werden. Die untersuchten Netze besitzen entweder zwei oder drei verdeckte Neuronenschichten. Zur Ermittlung einer gut geeigneten Neuronenanzahl existieren lediglich unspezifische Daumenregeln.

Nach Masters [219] kann die Anzahl der notwendigen Neuronen bei einem Netz mit einer verdeckten Neuronenschicht folgendermaßen bestimmt werden:

$$N = \sqrt{I \cdot O} \quad (27)$$

$N$  Anzahl empfohlener Neuronen in der verdeckten Schicht

$I$  Anzahl der Eingangsgrößen bzw. Eingangsneuronen

$O$  Anzahl der Ausgangsgrößen bzw. Ausgangsneuronen

Für ein Neuronales Netz mit zwei verdeckten Schichten definiert Masters [219] die Anzahl der Neuronen in der ersten und zweiten Schicht wie folgt:

$$N_1 = O \cdot \left( \sqrt[3]{\frac{I}{O}} \right)^2 \quad (28)$$

$$N_2 = O \cdot \sqrt[3]{\frac{I}{O}} \quad (29)$$

$N_1$  Anzahl empfohlener Neuronen in der 1. verdeckten Schicht

$N_2$  Anzahl empfohlener Neuronen in der 2. verdeckten Schicht

$I$  Anzahl der Eingangsgrößen bzw. Eingangsneuronen

$O$  Anzahl der Ausgangsgrößen bzw. Ausgangsneuronen

Somit ergeben sich mit  $I = 1.296$  und  $O = 73$  für  $N_1 = 83$  Neuronen und für  $N_2 = 78$  Neuronen. Da es sich bei diesen Angaben lediglich um Daumenregeln für Netze mit maximal zwei verdeckten Neuronenschichten handelt und auch Masters [219] die Ermittlung der optimalen Anzahl durch Ausprobieren nahe legt, soll im Folgenden eine erste Untersuchung mit folgenden Randbedingungen erfolgen.

In der ersten Untersuchung sollen folgende Neuronenanzahlen für bis zu drei verdeckte Schichten vollfaktoriell in 5er Schritten betrachtet werden:

- Neuronen in der ersten verdeckten Schicht: 80 bis 400 in 5er Schritten
- Neuronen in der zweiten verdeckten Schicht: 60 bis 200 in 5er Schritten
- Neuronen in der dritten verdeckten Schicht: 0 bis 100 in 5er Schritten (eine Neuronenanzahl von 0 bedeutet, dass lediglich zwei verdeckte Neuronenschichten untersucht werden)

Die untersuchte Netzarchitektur ist ein feedforward-Netzwerk, welches mittels scaled conjugate gradient (scg) backpropagation-Optimierungsalgorithmus angelernt wird. Als Abbruchkriterien werden ein minimaler Gradient des Optimierungsalgorithmus bzgl. dem Mean-Squared-Error von  $1e^{-9}$  und eine maximale Epochenanzahl von 1.000 definiert. Das Netz verwendet Tangens-hyperbolicus-Transferfunktionen und normiert die Eingangsdaten mittels der Minimal- und Maximalwerte. Die numerischen Matrizen der Größe  $36 \times 36$  werden zu einem Vektor mit 1.296 Werten umgewandelt. Das Training der 39.585 Netze erfolgt mittels GPU auf einer Nvidia Quadro K600 bzw. einer Nvidia Quadro P2000.

## 5.4 Ermittlung der Prognosegüte der KNNs

Zur Ermittlung der Prognosegüte der Neuronalen Netze wird die Anzahl der falsch klassifizierten Bauteile des Testdatensatzes verwendet. Da der Testdatensatz 1.168 Datensamples enthält, kann die Anzahl an falsch klassifizierten Bauteilen zwischen 0 und 1.168 Bauteilen liegen. Die Netze sind umso besser, je weniger Bauteile falsch klassifiziert werden. Zur Visualisierung der Ergebnisse des ersten Trainingsdurchlaufs werden alle Fehler der 39.585 trainierten Netze in Bild 37 dargestellt.

Die Fehlerrate wird farblich skaliert und ist über den drei Achsen der Neuronenanzahl in der ersten, zweiten und eventuell vorhandenen dritten verdeckten Neuronenschicht dargestellt. In Bild 37 sind alle Netze mit einer Fehlerrate von mehr als 33 falsch klassifizierten Bauteilen des Testdatensatzes transparent dargestellt. Eine Zahl kleiner 33 bedeutet, dass höchstens zwei Bauteile (jeweils 16 Mal unterschiedlich orientiert vorhanden) nicht richtig klassifiziert werden. Nicht transparent dargestellte Punkte stellen die Netze mit besserer Prognosegüte (mit weniger falsch klassifizierten Bauteilen des Testdatensatzes) dar.

Es ist ersichtlich, dass die Neuronenanzahl einen erheblichen Einfluss auf die Prognosegüte der trainierten Netze aufweist. Viele der Netze mit geringer Fehlerquote liegen im mittleren Bereich der Neuronenanzahl der dritten verdeckten Schicht.

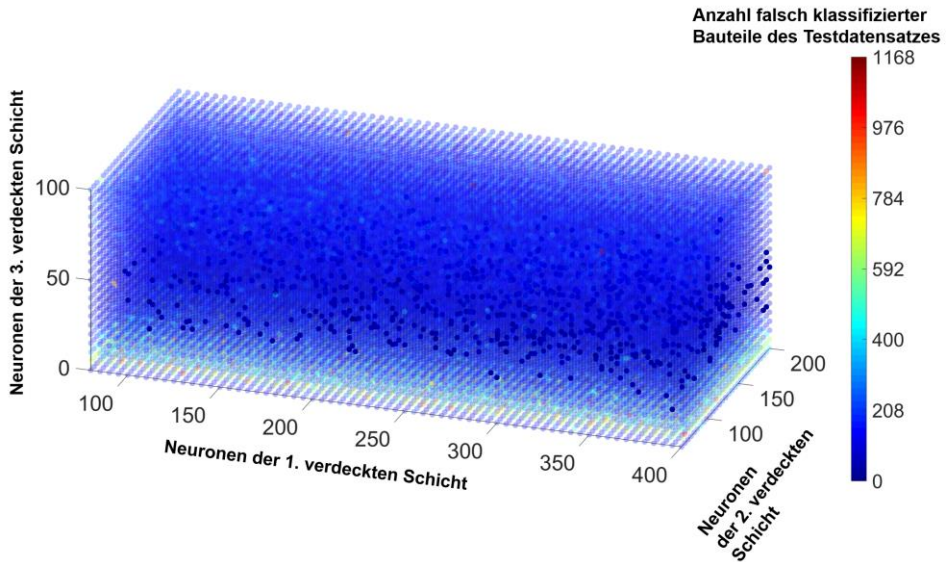


Bild 37: Anzahl der falsch klassifizierten Bauteile des Testdatensatzes aufgetragen über der Neuronenanzahl der 39.585 trainierten Netze in den bis zu drei verdeckten Schichten (Netze mit  $>33$  Fehlern sind transparent dargestellt)

Aus diesem Grund wird die Anzahl an Netzen mit hoher Prognosegüte ( $\leq 33$  bzw.  $\leq 65$  falsch klassifizierte Bauteile des Testdatensatzes, dies entspricht zwei bzw. vier falsch klassifizierte Bauteilen in 16 unterschiedlichen Orientierungen) in Abhängigkeit der Neuronenanzahl in der dritten verdeckten Schicht analysiert. Diese Ergebnisse zeigt Bild 38. Die Neuronalen Netze mit 35 bzw. 40 Neuronen weisen in der dritten verdeckten Schicht sehr häufig gute Prognosegüten auf.

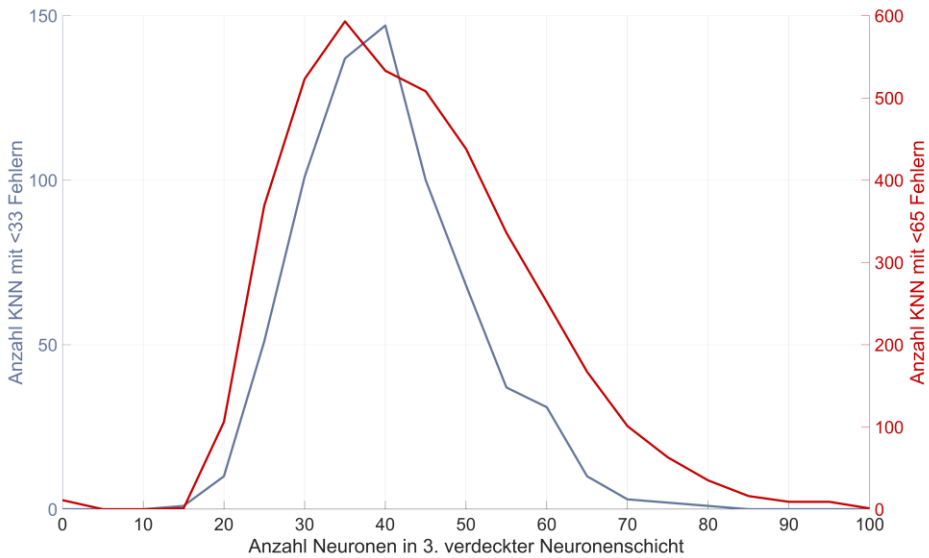


Bild 38: Anzahl der ANNs mit weniger als 33 bzw. weniger als 65 Fehlern bei der Klassifikation des Testdatensatzes aufgetragen über der Neuronenanzahl der dritten verdeckten Neuronenschicht

Im nächsten Schritt gilt es daher die Abhängigkeit der Neuronenanzahlen der ersten und zweiten verdeckten Schicht unter konstanter Neuronenanzahl der dritten verdeckten Schicht zu analysieren.

Bild 39 zeigt die Anzahl der falsch klassifizierten Bauteile des Testdatensatzes über der Neuronenanzahl der ersten beiden verdeckten Schichten. Die Neuronenanzahl der dritten Schicht wird in Bild 39 oben auf 30 Neuronen, mittig auf 35 Neuronen und in Bild 39 unten auf 40 Neuronen fixiert. Anhand der Ergebnisse kann folgende Empfehlung für die Wahl der entsprechenden Neuronenanzahl in den jeweiligen verdeckten Schichten zur Erreichung einer niedrigen Rate an falsch klassifizierten Bauteilen festgehalten werden:

- Neuronenanzahl der 1. verdeckten Schicht: 100 - 340
- Neuronenanzahl der 2. verdeckten Schicht: 120 - 170
- Neuronenanzahl der 3. verdeckten Schicht: 40

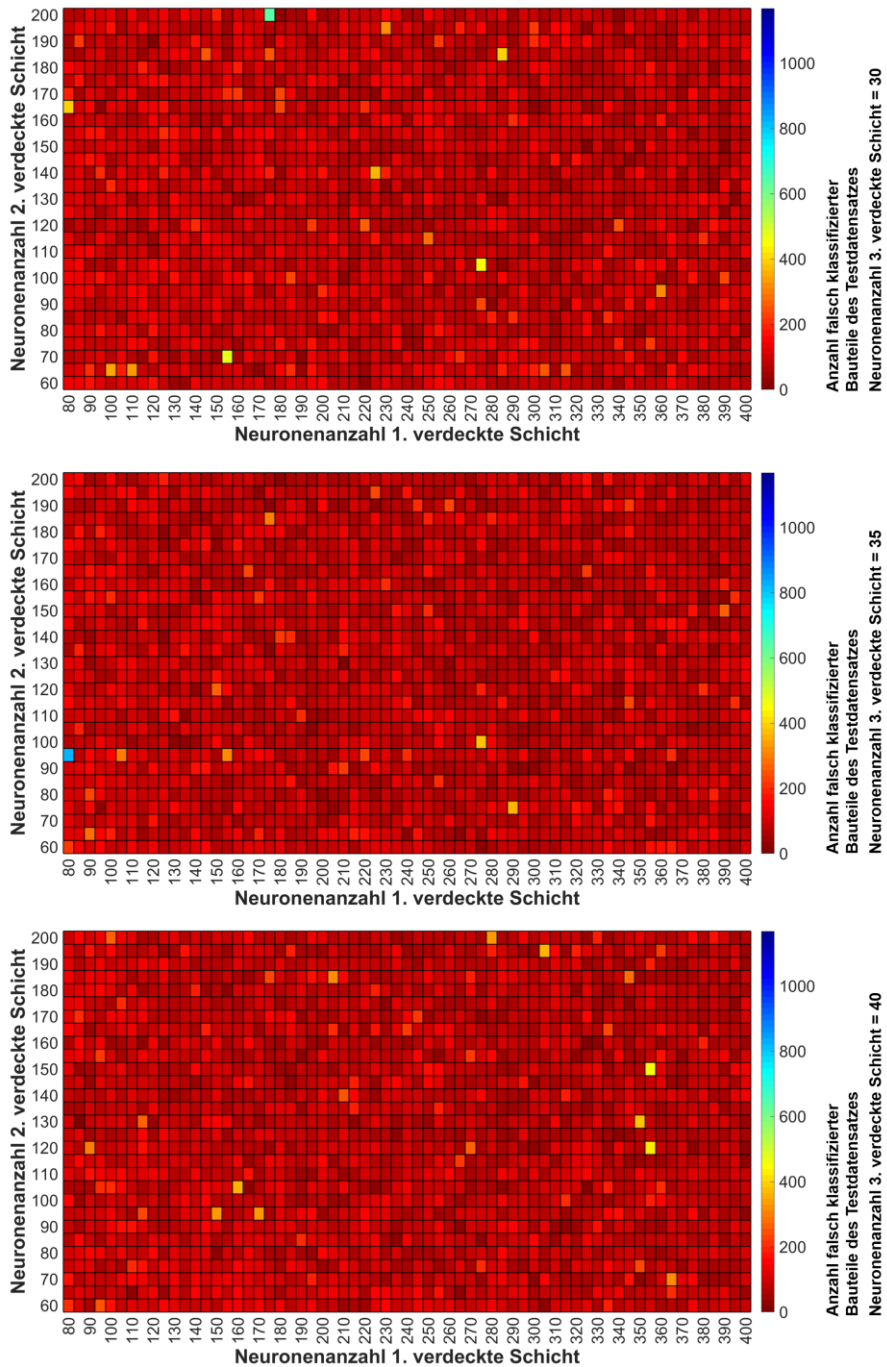


Bild 39: Anzahl der falsch klassifizierten Bauteile des Testdatensatzes aufgetragen über der Neuronenanzahl der 1. und 2. verdeckten Schicht (Neuronen der 3. verdeckten Schicht =30, =35 bzw. =40)

Es ist erkennbar, dass in einigen Bereichen sehr gute Ergebnisse erzielt werden können. Im Rahmen der Parameterstudie mit 5-Neuronen-Schrittweite konnten bereits mehrere KNNs mit einer Fehlerrate von «0 falsch klassifizierten Bauteilen» trainiert werden. Eine Auflistung dieser Netze zeigt Tabelle 6. Eine weitere Untersuchung von KNNs mit einer weiteren Detaillierung der Neuronenanzahl in 1-Neuronen-Schrittweite wäre möglich, da aber bereits so sehr gute Ergebnisse erreicht werden, soll hierauf verzichtet werden.

Tabelle 6: Trainierte KNNs mit einer Fehlerrate von 0 falsch klassifizierten Bauteilen des Testdatensatzes

KNN Nr.	Neuronenanzahl der 1. verdeckten Schicht	Neuronenanzahl der 2. verdeckten Schicht	Neuronenanzahl der 3. verdeckten Schicht
#1	85	130	40
#2	90	100	45
#3	120	190	30
#4	210	130	35
#5	220	65	45
#6	280	125	45
#7	325	105	40
#8	325	150	30
#9	325	160	40
#10	390	150	30
#11	395	120	25

## 5.5 Automatische Erkennung unbekannter Bauteile

Die Erkennung der Bauteile des Testdatensatzes liefert sehr gute Klassifikationsergebnisse. Allerdings werden in der späteren Anwendung einer derartigen Bauteilerkennung nicht ausschließlich bekannte Bauteile an das KNN übergeben. Bei der Suche von Gleichteilen in umfangreichen Datenbeständen der virtuellen Produktentwicklung gilt es ebenso, bisher unbekannte Bauteile als «unbekannt» zu klassifizieren. Für diesen Anwendungsfall können Neuronale Netze als Autoencoder verwendet werden. Sie können zur Dimensionsreduktion herangezogen werden, um anschließend mit den reduzierten Daten statistische Kenngrößen abzuleiten. Bild 40 verdeutlicht den Aufbau eines Autoencoders.

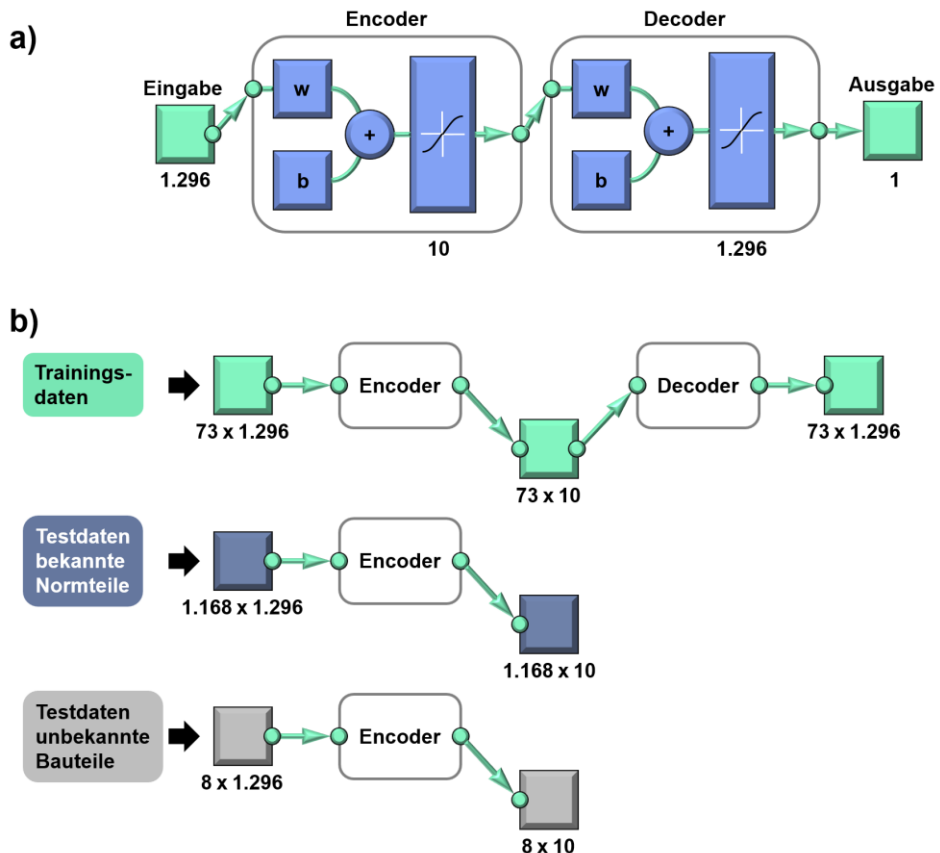


Bild 40: Anlernen eines Autoencoders zur Dimensionsreduktion der Datensätze zur nachgelagerten Erkennung von unbekanntem Bauteilen [a] Aufbau Autoencoder aus Encoder und Decoder, b) Anwendung eines Encoders zur Erkennung unbekannter Bauteile]

Ein Autoencoder besteht sowohl aus einem Encoder als auch einem Decoder. Dem Neuronalen Netz werden die Trainingsdaten (hier die 73 Datensätze der  $36 \times 36$  Matrizen, entsprechend als  $73 \times 1.296$  Matrix) übergeben und eine Neuronenanzahl von 10 vorgegeben. Das Netz soll den Datensatz mittels anzulernender Gewichtungen und Bias-Werte zunächst auf eine Größe von 10 Datenwerten je Sample mittels des Encoders reduzieren und anschließend mit dem Decoder wieder auf die ursprüngliche Größe der Eingangsmatrix bringen. Bei dieser Trainingskonfiguration entsprechen die Eingangsdaten exakt den Ausgangsdaten. Der Vorteil dieser Anwendung besteht darin, dass in einem zweiten Schritt nur der Encoder weiterverwendet wird. Somit können auch neue und bisher unbekannte Daten mittels Dimensionsreduktion auf 10 Kennwerte je Datensample reduziert werden. Für die bekannten Bauteile sind die Bereiche, in denen diese 10 Kennwerte variieren, bzw. die Werte der Trainingsdaten bekannt. Ein bisher unbekanntes Bauteil, das an den Encoder übergeben wird, erzeugt ebenfalls diese 10 Kennwerte. Allerdings weichen die 10 Kennwerte erfahrungsgemäß von denen der bekannten Bauteile ab. Somit kann mit einer entsprechenden Untersuchung der resultierenden Kennwerte nach dem Encoder für eine Erkennung unbekannter Bauteile verwendet werden.

Dieser Ablauf ist im unteren Teil von Bild 40 für die Trainingsdaten, die Testdaten bestehend aus zuvor angelernten Normteilen und für Testdaten bisher unbekannter Bauteile dargestellt. Die übergebenen Daten mit jeweils 1.296 Eingangswerten werden durch den Encoder mit 10 Neuronen auf 10 Ausgabewerte dieser Neuronen reduziert. Der Decoder wandelt diese 10 Werte mittels 1.296 Neuronen wieder in die Ausgangsdimension mit 1.296 Werten um. Daraus ergeben sich folgende reduzierte Datensätze mit Größenangabe:

- Trainingsdaten von 73 Normteilen ( $73 \times 10$ ).
- Testdaten der zufällig rotierten und translatorisch verschobenen Normteile, welche bereits im Trainingsdatensatz enthalten waren ( $1.168 \times 10$ ).
- Testdaten mit 8, bisher unbekanntem Bauteilen ( $8 \times 10$ ). Die einzelnen Bauteile sind in Bild 41 als CAD-Baugruppe dargestellt und markiert.

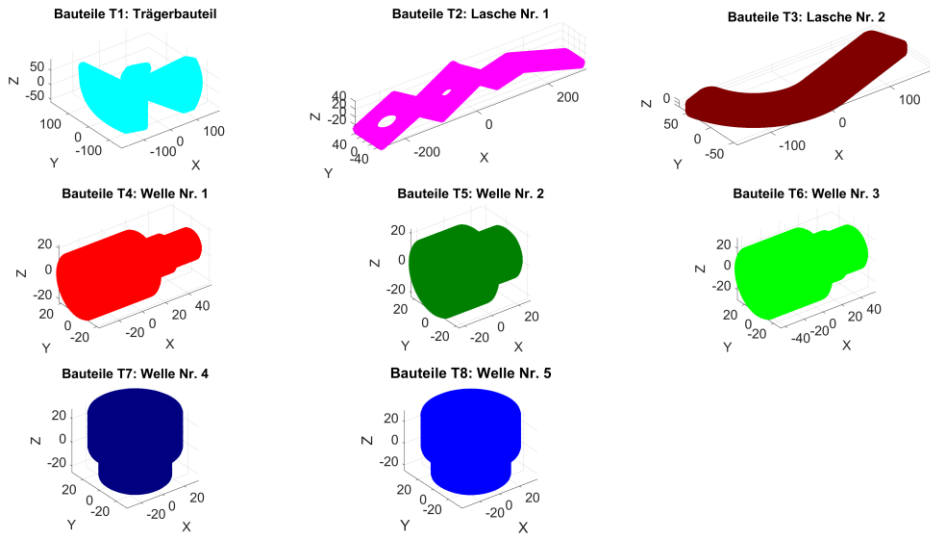


Bild 41: Bisher unbekannte Testbauteile, welche als unbekannt klassifiziert werden sollen

Das Vorgehen zur Unterscheidung von unbekanntem Bauteilen und solchen Bauteilen, welche bereits im Trainingsdatensatz enthalten waren, zeigt Bild 42. Die Grundlage bilden die zuvor mittels Encoder verkleinerten Datensätze. Datensatz *A* ist der Trainingsdatensatz und enthält die später zu klassifizierenden Normteile, Datensatz *B* ist ein unbekanntes Bauteil, welches als solches zu klassifizieren ist. Würde die  $36 \times 36$ -Matrix dieses Bauteils an ein Klassifikations-KNN aus Tabelle 6 übergeben werden, würde dieser Datensatz fälschlicherweise als ein bestimmtes Normteil des Trainingsdatensatzes klassifiziert. Dies gilt es, mit einer vorgeschalteten Erkennung von unbekanntem Bauteilen zu verhindern.

Zur Erstellung der Matrix *C* wird der gemittelte quadratische Fehler zwischen dem unbekanntem Bauteildatensatz *B* und jeder Zeile (mit jedem Bauteil) des reduzierten Trainingsdatensatzes berechnet. Im nächsten Schritt wird das globale Minimum des Datensatzes *C* gebildet. Dieser einzelne Zahlenwert *D* kann für bekannte Bauteile (die bereits im Trainingsdatensatz enthalten waren) durch numerische Fehler geringfügig von dem exakten Wert 0 abweichen.

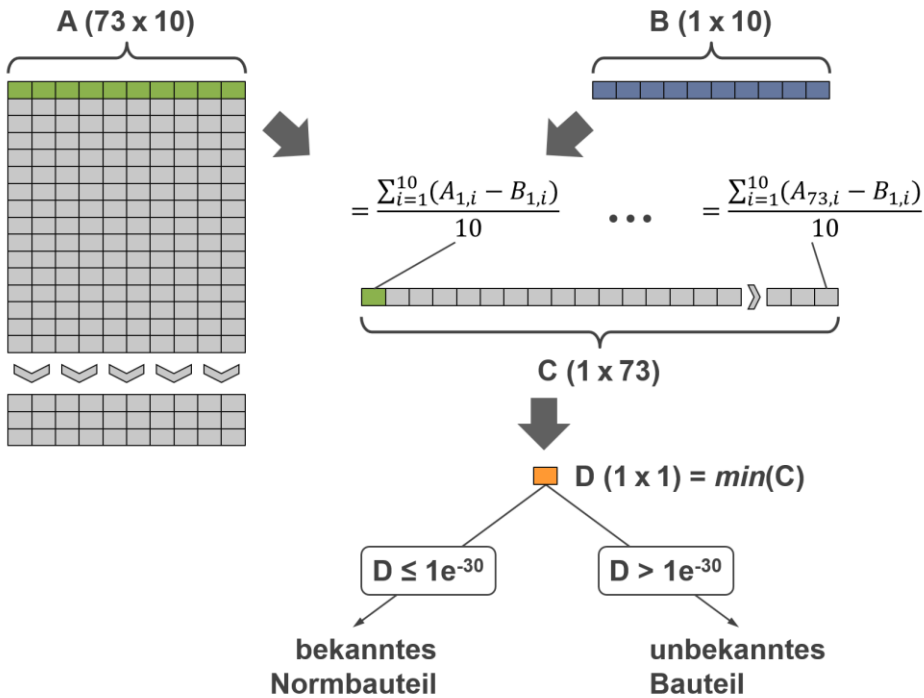


Bild 42: Vorgehen zur Klassifikation unbekannter Bauteile, welche nicht in einem zuvor angelernten Trainingsdatensatz enthalten waren

Für den vorliegenden Datensatz und die untersuchten Testbauteile, bestehend aus 1.168 bekannten Normteilen und 8 unbekanntem Bauteile (siehe Bild 41), liefert ein Toleranzschwellwert von  $1e^{-30}$  sehr gute Ergebnisse. Hiermit ist es möglich, mittels Autoencoder und paarweisem Vergleich der mittleren quadratischen Abweichung unbekannte Bauteile sicher zu erkennen.

Die ermittelten Werte  $D$  der Bauteile des Testdatensatzes können Tabelle 29 im Anhang entnommen werden. Es fällt auf, dass alle unbekanntem Bauteile sicher als unbekannt klassifiziert werden, wohingegen fast alle bekannten Bauteile sehr genau als bekannt klassifiziert werden können. Lediglich die 16 Muttern der Norm ISO 4035 – M12 zeigen einen geringfügigen Ausschlag. Dieser liegt jedoch weit unter dem Wert der unbekanntem Bauteile und bewegt sich mit einem Kennwert von  $9,8636e^{-33}$  sehr nahe an der 0, welche die bekannten Bauteile definiert.

## 5.6 Ergebnisse und Bewertung

Das vorliegende Anwendungsbeispiel zur Klassifikation von Bauteilen bzw. Normteilen zeigt, wie die Methode der sphärischen Detektorflächen verwendet werden kann, um Geometrieinformationen von unterschiedlichen Bauteilen an Algorithmen des Maschinellen Lernen zu übergeben. Mit der vorgestellten Methodik ist es möglich, CAD-Geometrie aus neutralen Austauschformaten durch einen Vernetzungsalgorithmus in eine Punktwolke umzuwandeln. Die Punktwolke kann anschließend mittels der entwickelten Methodik in eine einheitliche Orientierung überführt werden. Durch die Projektion auf eine sphärische Detektorfläche entsteht für jedes Bauteil eine charakteristische numerische Matrix der Größe  $36 \times 36$ , welche in einen Zeilenvektor der Größe 1.296 überführt werden kann. Mittels des Encoder-Teils eines Autoencoders und der Untersuchung des gemittelten quadratischen Fehlers zwischen einem zu klassifizierenden Bauteil und den Trainingsdaten können unbekannte Bauteile sicher erkannt werden. Anschließend können die vorliegenden Vektoren durch ein Neuronales Netz mit 3 verdeckten Schichten sicher einer der 73 Normteile zugeordnet werden. Zusammenfassend liegt eine Methode zur automatischen Bauteilklassifikation mittels Künstlicher Neuronaler Netze vor. Andere Algorithmen des Maschinellen Lernen wären ebenso anwendbar, da die Daten in eine einheitliche Form mit fixer Größe überführt werden können.



## **6 Anwendung #2: Erkennung von Montagefehlern**

Mit dem ersten Anwendungsbeispiel – der Bauteilerkennung – konnte gezeigt werden, dass eine Umsetzung der Methodik der sphärischen Detektorflächen in der virtuellen Produktentwicklung umsetzbar ist. Um die Übertragbarkeit auf andere Bereiche der Produktentstehung zu verdeutlichen, soll das zweite Anwendungsbeispiel den Bereich der Überwachung von Montageprozessen betrachten. Als Demonstrator soll die Montage von Kabelbäumen in der Automobilindustrie herangezogen werden. Diese Tätigkeit belastet den Schulter- und Nackenbereich der Monteure stark [220] und kann zu Montagefehlern führen, welche durch das Qualitätsmanagement möglichst schnell erkannt werden müssen. Im Folgenden wird dargestellt, wie die Methode der sphärischen Detektorflächen verwendet werden kann, um Scan Daten eines optischen Vermessungssystems in eine numerische Matrix umzuwandeln. Diese Matrix wird anschließend zur Klassifikation möglicher Montagefehler herangezogen. Das Ziel ist eine automatische Erkennung von falsch verlegten Kabelbäumen an dem Verbrennungsmotor eines Nutzfahrzeugs. Die Erzeugung der Scan-Daten erfolgt rein virtuell und berücksichtigt keine Scan-Fehler oder unzureichend abgedeckte Oberflächen der Baugruppe. Eine Berücksichtigung von Messunsicherheit und eine Überlagerung von Punktwolken von unterschiedlichen Scan-Kameras findet hingegen bereits Berücksichtigung.

### **6.1 Definition der Problemstellung**

Falsch verlegte Kabelbäume, beispielsweise entlang eines Verbrennungsmotors, können zum Aufscheuern von Adern des Kabelbaums oder zum Öleintritt in den Kabelstrang an den Steckverbindern führen – vgl. z. B. [221]. Besonders relevant wird diese Thematik, wenn an derselben Montagelinie unterschiedliche Motoren bzw. Motoren mit individuellen Anbauteilen und Kabelbäumen mit unterschiedlichen Verlegewegen montiert werden. Dies ist insbesondere bei Nutzfahrzeugmotoren und bei vermehrter Individualisierung und Aspekten bedingt durch die Industrie 4.0 häufig der Fall, da diese Aggregate neben LKWs auch in Land-, Forst- und Baumaschinen oder als stationäre Motoren in Blockheizkraftwerken zum Einsatz kommen. Gleichzeitig ist der Kabelbaum ein volumenmäßig relativ kompaktes Bauteil im Vergleich zu den großen Bauteilen eines Aggregats, wie etwa Zylinderkopf, Kurbelgehäuse oder Ölwanne. Folglich muss die

Methodik der sphärischen Detektorflächen in der Lage sein, diese Feinheiten abzubilden.

Die Generierung entsprechender Punktwolken für die spätere Erzeugung der Detektormatrizen erfolgt im Rahmen von Anwendungsbeispiel #2 virtuell. Der Ausgangspunkt ist ein bereits sehr detailliertes Motorenmodell, welches durch Artem Slyusarev erstellt wurde und über die Plattform GrabCAD verfügbar ist. Es handelt sich um einen Vierzylinder-Dieselmotor des Herstellers Kubota (V3600). Einige Bauteile wurden neu modelliert, um ein Modell mit durchgängiger Oberfläche ohne fehlende Flächensegmente zu erhalten. Darüber hinaus wurden 3 unterschiedliche Kabelbäume modelliert (eine korrekte Verlegung und 2 falsch verlegte Kabelbäume). Bild 43 zeigt das CAD-Modell des Motors mit korrekt verlegtem Kabelbaum. Der Kabelbaum besteht aus dicken Leitungen zur Anbindung von Lichtmaschine und Anlasser, dünnen Sensor- und Steuerleitungen sowie einem Schutzschlauch, in welchem die Leitungen zusammenlaufen.

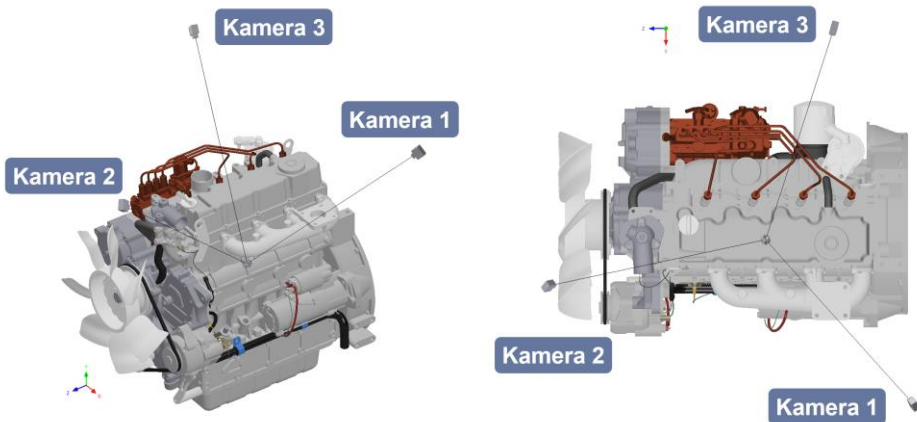


Bild 43: Adaptiertes CAD-Modell (Original: Diesel engine Kubota v3600 von Artem Slyusarev aus GrabCAD) erweitert um Kabelbaum und drei virtuelle Scan-Kamera Positionen, Orthogonale Ansicht (links) und Draufsicht (rechts)

Zur Erzeugung der späteren Punktwolken werden virtuelle Positionen für die Scan-Kameras definiert, in Bild 43 sind drei Positionen als Kameraobjekte sowohl in einer orthogonalen wie auch in der Draufsicht dargestellt. Die verwendeten drei Kamerapositionen stellen ein Minimum dar, um mit den vorgegebenen Öffnungswinkeln der Kameras das gesamte Objekt erfassen zu können (vgl. nachfolgendes Kapitel 6.2). Durch die Kombination von drei Kamerapositionen mit drei unterschiedlichen CAD-Modellen ergeben sich neun Ansichten der Kameras, welche in Bild 44 dargestellt sind.

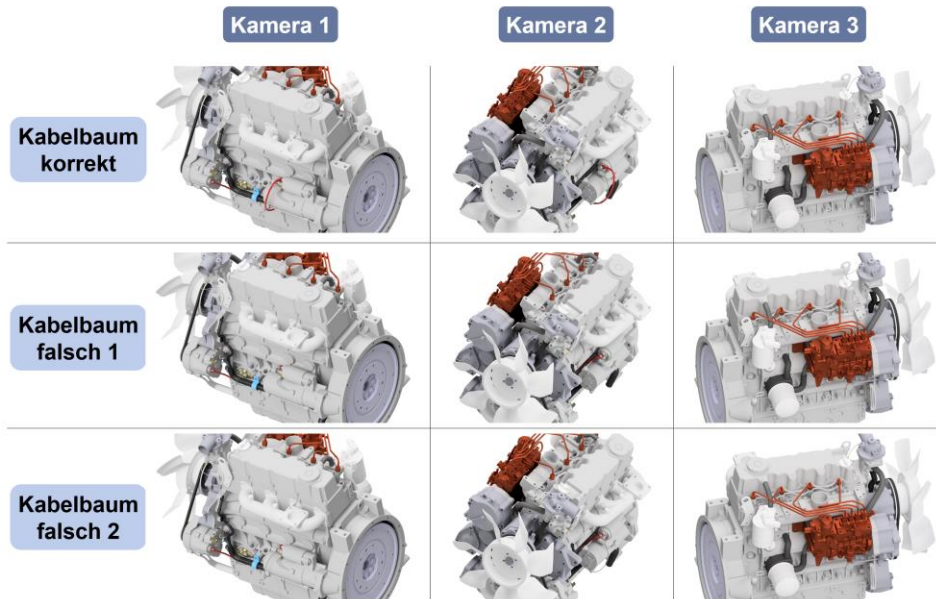


Bild 44: Ansichten der drei Kameras auf die CAD-Modelle mit korrekt und zwei falsch verlegten Kabelbäumen

Die Unterschiede der verlegten Kabelbäume mit einzelnen Kabeln sind in den Ansichten von Kamera 1 und Kamera 2 erkennbar. Details der Verlegung sind in Bild 45 (für Kamera 1) und in Bild 46 (für Kamera 2) dargestellt. In diesen beiden Bildern kann sowohl die Verlegung der Plusleitung des Anlassers als auch ein deutlich dünneres Kabel eines Sensors erkannt werden. In der ersten Variante des falsch verlegten Kabelbaums laufen beide Kabel zwischen Anlasser und Motorblock. In der zweiten fehlerhaften Variante ist die dünnere Sensorleitung korrekt verlegt, wohingegen die dickere Plusleitung erneut zwischen Anlasser und Motorblock verläuft. Korrekt verlegt, laufen beide Kabel vor dem Anlasser vorbei, wodurch sie weder den heißen Motorblock berühren noch Kontakt zu benachbarten Bauteilen haben.

6 Anwendung #2:  
Erkennung von Montagefehlern

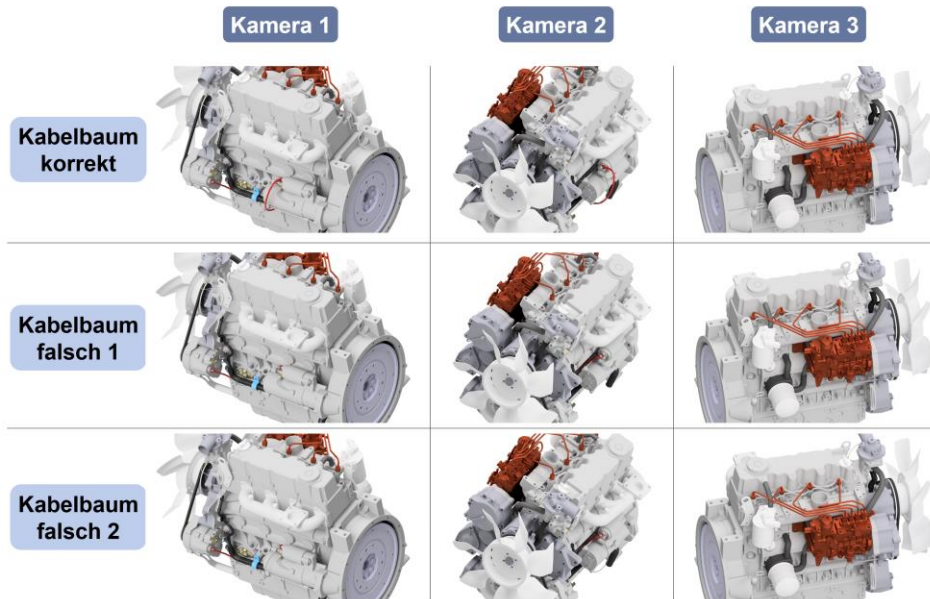


Bild 45: Detailbilder Kamera 1 mit den verlegten Kabelbäumen

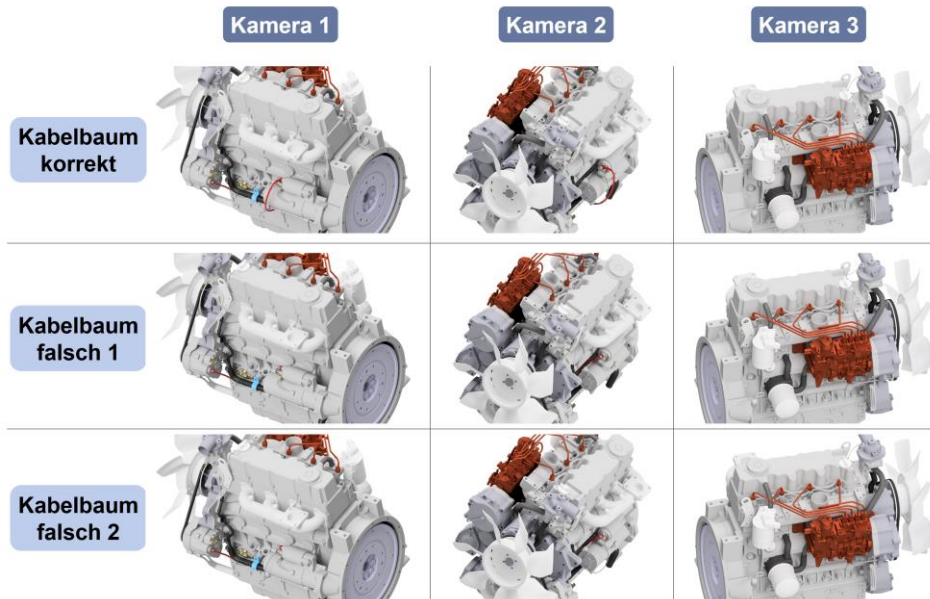


Bild 46: Detailbilder Kamera 2 mit den verlegten Kabelbäumen

## 6.2 Aufbau der Datenbasis

Jede Kameraposition und das identische Ziel der Kameras sind im CAD-Modell definiert worden. Zur Erzeugung der Punktwolke über Ray-Tracing muss zunächst der Öffnungswinkel der Kamera sowie die Auflösung der virtuellen Scan-Kamera definiert werden. Gute Scan-Verfahren arbeiten mit einer Auflösung von ca. 12 Megapixeln. Außerdem weisen sie Auflösungen von  $4.260 \times 2.840$  Pixeln, einem horizontalen Öffnungswinkel von  $\approx 70^\circ$  und einem vertikalen Öffnungswinkel von  $\approx 50^\circ$  auf. Mit diesen Informationen kann im Abstand von  $0,5 \text{ mm}$  von der Kameraposition in Richtung des Kameraziels eine Stützfläche mit  $4.260 \cdot 2.840 = 12.098.400$  Stützpunkten aufgespannt werden. Mittels eines Ray-Tracings wird der Schnittpunkt eines Strahls ermittelt, der von der Kameraposition durch jeden Stützpunkt auf die CAD-Baugruppe trifft. Die Koordinaten des ersten Schnittpunkts des Strahls mit der CAD-Baugruppe werden entsprechend in kartesischen Koordinaten abgespeichert.

Wie in Bild 44 zu erkennen ist, sind die Kameras nicht vollständig auf die CAD-Baugruppe ausgerichtet. Folglich gibt es nicht exakt 12.098.400 Schnittpunkte mit der CAD-Baugruppe, sondern es treffen viele Rays überhaupt kein Bauteil und geben somit auch keinerlei Schnittpunkte zurück. Der Umfang der ermittelten Schnittpunkte des Ray Tracings mittels Programmierschnittstelle der CAD Software Autodesk Inventor Professional 2017 sind in Tabelle 7 aufgelistet:

Tabelle 7: Ermittelte Schnittpunkte des Ray Tracings mit den CAD-Baugruppen

	X-, Y- und Z-Koordinaten der Schnittpunkte des Ray Tracings mit den CAD-Baugruppen		
	Kamera 1	Kamera 2	Kamera 3
Kabelbaum korrekt	3.689.619x3	4.085.044x3	4.085.449x3
Kabelbaum-falsch 1	3.685.675x3	4.042.694x3	4.085.449x3
Kabelbaum falsch 2	3.686.459x3	4.050.099x3	4.085.449x3

Die Punktwolken aller drei Kameras sind für die unterschiedlichen Baugruppen in Bild 47 grafisch visualisiert. Unterschiede der Baugruppen sind beispielsweise bei der Zuleitung des Anlassers (jeweils im rechten unteren

Teil der geplotteten Punktwolke) zu erkennen. Die korrekte Verlegung der Plusleitung und des Sensorkabels muss an der Außenseite erfolgen. Bei Baugruppe «falsch 1» ist die Plusleitung und das Sensorkabel zwischen Anlasser und Kurbelgehäuse verlegt und bei Baugruppe «falsch 2» ist nur die Plusleitung falsch verlegt. Eine weitere Unterscheidung stellt die Verlegung der obersten Sensorleitung dar. Bei Baugruppe «falsch 1» ist diese Sensorleitung hinter der Anhebelasche des Motorblocks verlegt, erkennbar ist dies anhand des Verlaufs der orange eingefärbten Leitung in der Nähe der blauen Haltetasche in Bild 47. Auch in Bild 44 ist dies in den Perspektiven von Kamera 1 und Kamera 2 erkennbar.

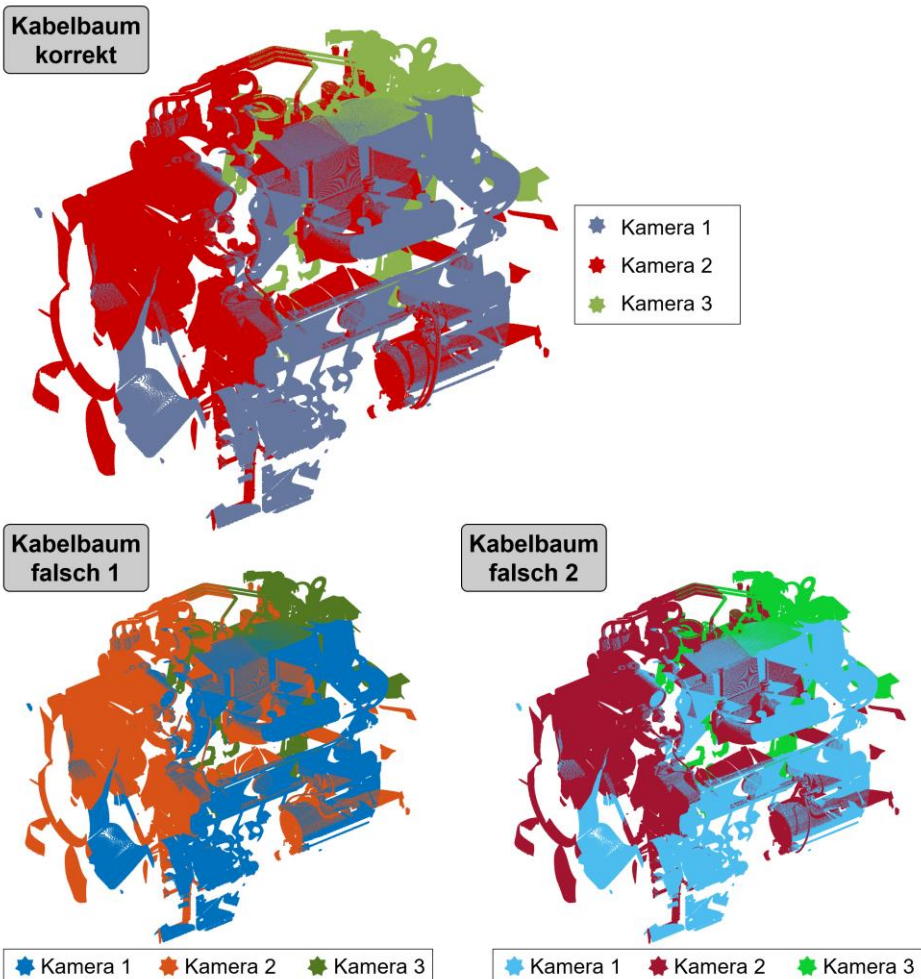


Bild 47: Kombinierte Punktwolken der Kameras für den korrekt und 2x falsch verlegten Kabelbaum

### 6.3 Erzeugung der Detektormatrizen

Durch die Kombination der drei Kamerapunktewolken für jede der drei CAD-Baugruppen ergeben sich drei kombinierte Punktewolken. Auf jede dieser Punktewolken kann die entwickelte Methodik der sphärischen Detektorflächen angewendet werden. Bild 48 zeigt dies mit einer Detektorfläche mit 200x200 Pixeln.

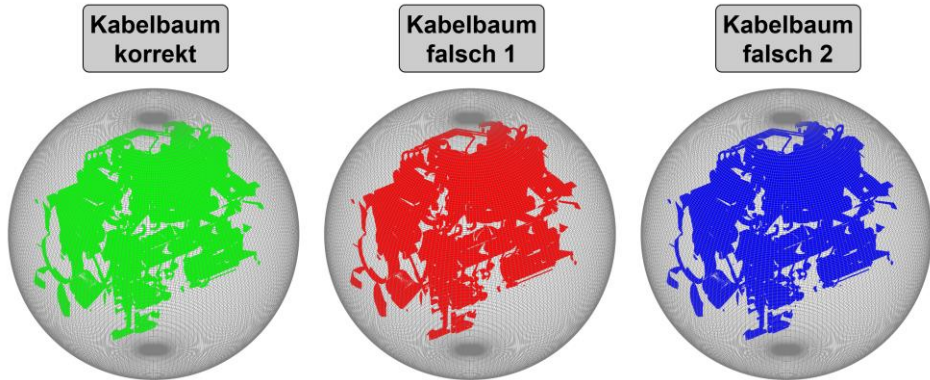


Bild 48: Aufgespannte Detektorfläche mit 200x200 Detektorpixeln für die 3 unterschiedlichen CAD-Baugruppen der Kabelbäume

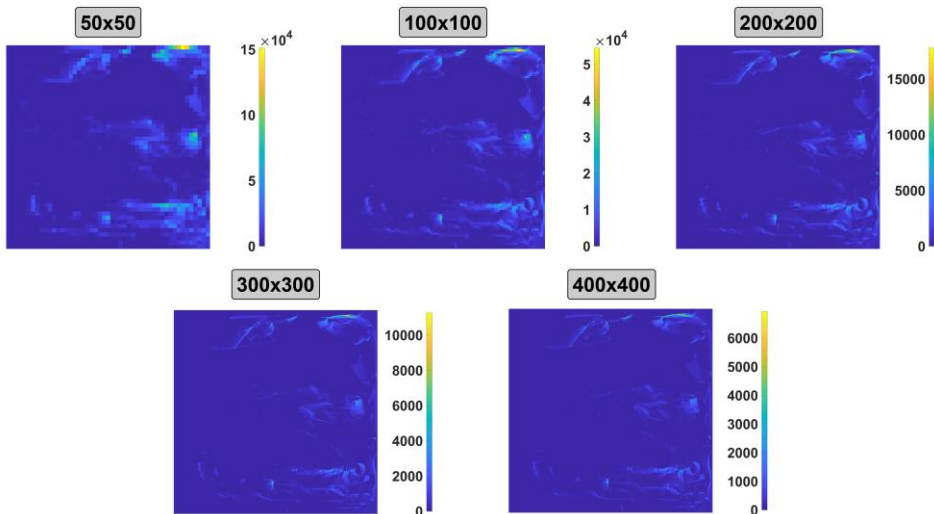


Bild 49: Detektormatrizen für das Modell des korrekt verlegten Kabelbaums in den Größen 50x50, 100x100, 200x200, 300x300 und 400x400 jeweils mit der Anzahl der projizierten Punkte in jedes Pixel

Andere Größen der Detektorfläche sind ebenso anwendbar, die Matrizen der Punktwolke sind für die Baugruppe mit korrekt verlegtem Kabelbaum für die Pixelanzahl 50x50, 100x100, 200x200, 300x300 und 400x400 in Bild 49 dargestellt. Die maximale Anzahl an projizierten Punkten in einem Pixel nimmt mit steigender Pixelanzahl ab, dies ist an den Farbbalken jeder Matrix direkt ersichtlich. Zusätzlich ist erkennbar, dass die Geometrie des Motorblocks mit steigender Pixelanzahl deutlich besser abgebildet wird.

Die Gesamtanzahl an numerischen Werten je Matrix ist in Tabelle 8 dargestellt:

Tabelle 8: Unterschiedliche Modellreduktionen aufgrund der Größe der Detektorflächen

Anzahl Detektorpixel	Anzahl numerischer Werte nach Modellreduktion
50x50	2.500
100x100	10.000
200x200	40.000
300x300	90.000
400x400	160.000

Für die weiteren Betrachtungen werden Detektorflächen der Größe 200x200 verwendet, dies liegt in der detaillierten Abbildung der Geometrie bei gleichzeitig vertretbarer Anzahl an numerischen Werten nach der Modellreduktion begründet.

Alle Messverfahren besitzen eine teilweise nicht unerhebliche Messunsicherheit. Nach Keferstein & Marxer [222] kann von folgenden Messunsicherheiten in  $\mu\text{m}$  für verschiedene optische Messverfahren ausgegangen werden:

Tabelle 9: Messunsicherheit von optischen Messverfahren aus [222]

Messverfahren	Messbereich	Messunsicherheit
Streifenprojektion	1 mm bis 2.000 mm	2 $\mu\text{m}$ bis 600 $\mu\text{m}$
Laser Tracker	2.500 mm bis 60.000 mm	11 $\mu\text{m}$ bis 550 $\mu\text{m}$
Laser Scanner	3 mm bis 200 mm	0,7 $\mu\text{m}$ bis 3 $\mu\text{m}$
Photogrammetrie	300 mm bis 10.000 mm	10 $\mu\text{m}$ bis 300 $\mu\text{m}$

Im Folgenden soll die Messunsicherheit eines Messsystems künstlich ergänzt werden. Hierfür wird jeder Messpunkt in den drei Koordinatenachsen zufällig um einen bestimmten Wert verschoben. Der Verschiebungswert wird sukzessive von  $0\text{ mm}$  auf  $\pm 0,015\text{ mm}$ ,  $\pm 0,03\text{ mm}$ ,  $\pm 0,06\text{ mm}$  und  $\pm 0,3\text{ mm}$  erhöht. Eine Auflistung dieser Datensätze ist in Bild 50 zu erkennen.

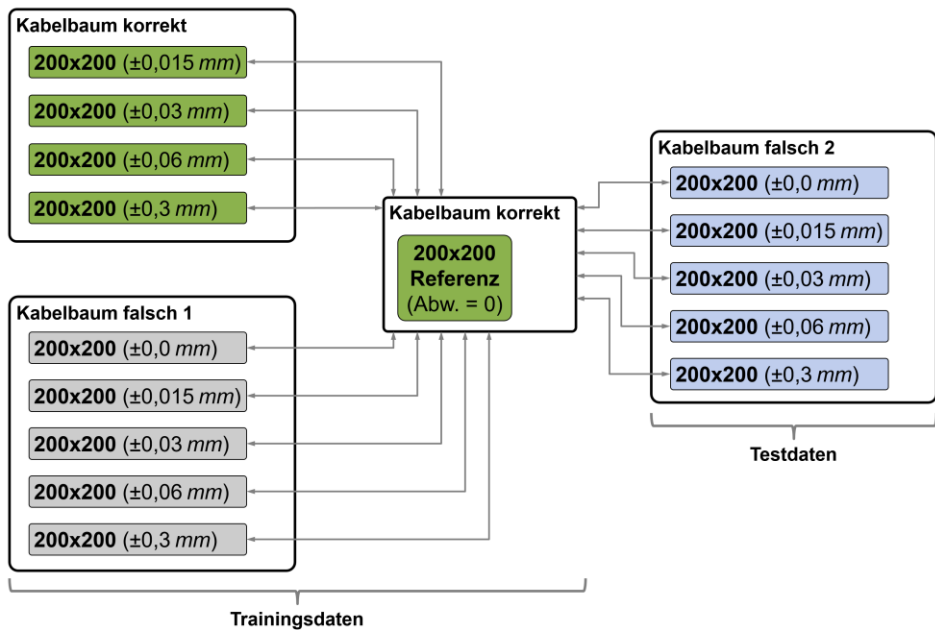


Bild 50: Übersicht der zur Verfügung stehenden Matrizen für den nachgelagerten paarweisen Vergleich mit der Referenzmatrix des korrekten Kabelbaums ohne künstlich beaufschlagte Scan-Abweichung

## 6.4 Bildvergleich der Detektormatrizen

Die erzeugten Detektormatrizen stehen für weitere Auswertungen bereit, allerdings sollen nun keine Algorithmen des Maschinellen Lernen sondern Metriken zum Vergleich der Matrizen herangezogen werden. Hierfür werden alle erzeugten Matrizen, auch diese mit ergänzter künstlicher Messunsicherheit, mit der Referenzmatrix des korrekt verlegten Kabelbaums ohne zusätzliche Abweichungen verwendet. Mittels der Matrizen des korrekt verlegten Kabelbaums und denen des Kabelbaums «falsch 1» können Schwellenwerte für die Bildvergleichsmetriken abgeleitet werden. Anschließend können die Matrizen des Kabelbaums «falsch 2» als Testdatensatz verwendet werden. Dieses Vorgehen und die zur Verfügung stehenden Matrizen sind in Bild 50 dargestellt. Durch den verwendeten Bildvergleich

zwischen Test- und Trainingsdatensatz soll überprüft werden, ob falsch montierte Kabelbäume sicher erkannt werden können.

Als Bildvergleichsmetriken werden der Structural Similarity Index (SSIM), der 2-D Korrelationskoeffizient (corr2) und das Peak Signal-to-Noise Ratio (PSNR) zwischen zwei Matrizen  $A$  und  $B$  verwendet. Die mathematische Beschreibung dieser Vergleichsmetriken ist in Formel (30) – Formel (32) gezeigt.

$$\begin{aligned} & \text{corr2}(A, B) \\ &= \frac{\sum_{m=1}^y \sum_{n=1}^x (A_{mn} - \mu_A) \cdot (B_{mn} - \mu_B)}{\sqrt{(\sum_{m=1}^y \sum_{n=1}^x (A_{mn} - \mu_A)^2) \cdot (\sum_{m=1}^y \sum_{n=1}^x (B_{mn} - \mu_B)^2)}} \end{aligned} \quad (30)$$

corr2( $A, B$ ) Korrelationskoeffizienten von Matrix  $A$  mit  $B$

$\mu_A$  Mittelwert von Matrix  $A$

$\mu_B$  Mittelwert von Matrix  $B$

$x$  maximale Dimension Matrix in  $x$ -Richtung

$y$  maximale Dimension Matrix in  $y$ -Richtung

$$\text{PSNR}(A, B) = 10 \log_{10} \left( \frac{L^2}{\text{MSE}} \right) \quad (31)$$

$$\text{mit MSE} = \sum_{m=1}^y \sum_{n=1}^x (A_{mn} - B_{mn})^2$$

PSNR( $A, B$ ) Peak signal-to-noise ratio

MSE mean squared error

$$\text{SSIM}(A, B) = \frac{(2 \cdot \mu_A \cdot \mu_B + C_1) \cdot (2 \cdot \sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1) \cdot (\sigma_A^2 + \sigma_B^2 + C_2)} \quad (32)$$

$$\text{mit } C_1 = (0,01 \cdot L)^2$$

$$\text{und } C_2 = (0,03 \cdot L)^2$$

SSIM(A, B) SSIM Wert von A mit B nach [223]

$\mu_A$  Mittelwert von Matrix A

$\mu_B$  Mittelwert von Matrix B

$\sigma_A$  Standardabweichung von Matrix A

$\sigma_B$  Standardabweichung von Matrix B

$\sigma_{AB}$  Kreuzkovarianz von Matrix A und B

$C_1$  Regularisierungskonstante 1

$C_2$  Regularisierungskonstante 2

$L$  Dynamischer Bereich, hier  $L = 1$

## 6.5 Ergebnisse und Bewertung

Wie bereits in Bild 50 gezeigt, werden alle zur Verfügung stehenden Matrizen mit der Referenzmatrix des korrekt verlegten Kabelbaums ohne künstlich ergänzte Abweichungen miteinander verglichen. Hierfür werden die Bildvergleichsmetriken SSIM, corr2 und PSNR verwendet. Zunächst werden die Metriken des «korrekt» verlegten und des Kabelbaums «falsch 1» ermittelt. Diese dienen zur Definition eines Schwellenwertes, ab dem ein verlegter Kabelbaum als falsch klassifiziert werden muss. Anschließend werden die ermittelten Werte der Metriken des Kabelbaums «falsch 2» ermittelt und es können die zuvor festgelegten Metriken mittels dieses Testdatensatzes bewertet werden. Die Ergebnisse dieses Vorgehens sind in Bild 51 für die drei Metriken separat über der steigenden künstlich ergänzten Messabweichung dargestellt. Die Festlegung der Schwellenwerte muss für jeden Anwendungsfall individuell definiert werden.

Die Schwellwerte für den vorliegenden Fall wurden wie folgt definiert:

- Schwellwert corr = 0,9993
- Schwellwert PSNR = -20
- Schwellwert SSIM = 0,975

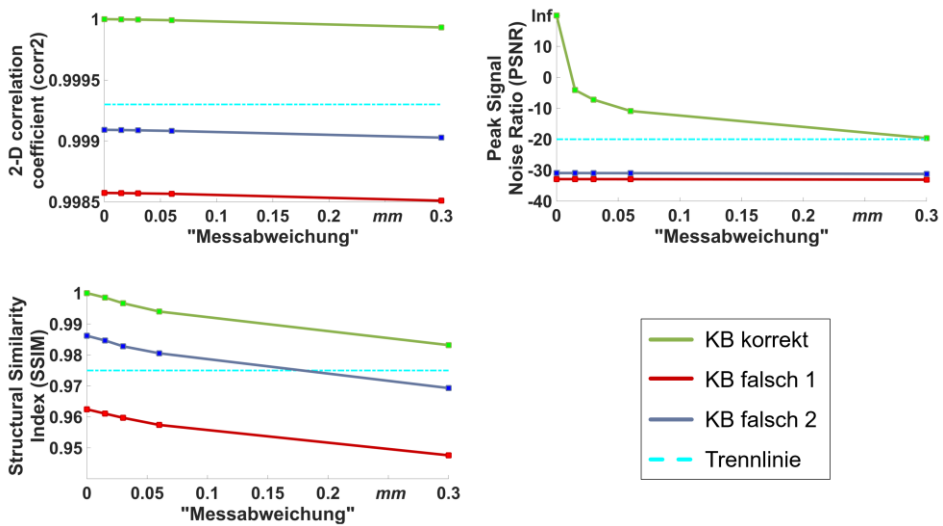


Bild 51: Bildvergleichsmetriken (corr2, PSNR und SSIM) des Vergleichs zwischen Referenzmatrix und den anderen Matrizen

Es zeigt sich, dass eine Unterscheidung eines korrekt verlegten Kabelbaums für geringe ergänzte Messunsicherheit bis ca.  $\pm 0,15 \text{ mm}$  mit allen drei Metriken umsetzbar ist. Mit weiter ansteigender Messunsicherheit ist eine sichere Unterscheidung nur noch mittels corr2 und PSNR möglich.

Somit können Scan-Daten als Eingangsgröße für die Methode der sphärischen Detektorflächen genutzt und geringe Änderungen an einem CAD-Modell bzw. in den daraus abgeleiteten Punkten eines Messverfahrens mittels recht einfacher Bildvergleichsmetriken erkannt werden. Insbesondere die Metriken corr2 und PSNR sind hierfür geeignet und liefern im zweiten vorliegenden Anwendungsfall gute Ergebnisse.

Folgende Veränderungen weist das zweite Anwendungsbeispiel gegenüber dem ersten auf:

- Verwendung von Punkten auf der Oberfläche von CAD-Bauteilen (keine Volumenvernetzung).
- Keine gleichmäßige Punkteverteilung, da kein Vernetzer mit konstanter Elementgröße zur Anwendung kommt, sondern ein Ray-Tracing bei dem die Dichte der Punkte sowohl von der Geometrie als auch von der Entfernung des Objekts von der Kameraposition abhängt.
- Anwendung einer deutlich größeren Detektorfläche mit einer Größe von 200x200 Pixeln.
- Keine Anwendung von Algorithmen des Maschinellen Lernens.



## 7 Adaption der Methodik der sphärischen Detektorflächen

In den vorangegangenen Kapiteln wurden die Methodik der sphärischen Detektorflächen in Kapitel 4 sowie zwei Anwendungsbeispiele: Bauteilerkennung in Kapitel 5 und Erkennung von Montagefehlern in Kapitel 6 aufgezeigt. Es konnte gezeigt werden, dass eine Übergabe von Geometriedaten in Form von Punktwolken an Algorithmen des Maschinellen Lernen umsetzbar ist bzw. auch andere statistische Kennwerte zur Klassifikation der erzeugten Detektormatrizen herangezogen werden können. Bisher werden jedoch ausschließlich Punktwolken ohne weitere Informationen verwendet. Im Rahmen der Adaption der Methodik der sphärischen Detektorflächen soll eine Erweiterung erfolgen, um auch andere mit der Geometrie verknüpfte Informationen in Detektormatrizen umzuwandeln. Nach der Adaption der Methodik in Kapitel 7 erfolgt in einem dritten Anwendungsbeispiel in Kapitel 8 eine Umsetzung auf umfangreiche Datenbestände aus Finite-Elemente-Simulationen.

### 7.1 Detektormatrizen für beliebige knotengebundene Größen

Ein großer Vorteil der zuvor beschriebenen Methodik der Detektorflächen besteht in der Zuordnung von Punkten aus einer Punktwolke zu bestimmten Pixeln auf der Detektorfläche. Folglich ist es ebenso möglich, Informationen, welche mit den Punkten der Punktwolke verknüpft sind, in die Pixel der Detektorfläche bzw. -matrix zu aggregieren. Alle Eingangs- und Zielgrößen einer Finite-Elemente-Simulation sind mit den Knoten eines Finite-Elemente-Netztes verknüpft. Damit ist gemeint, dass alle Randbedingungen im Pre-Processing-Schritt auf individuelle Knotenlasten oder gesperrte Freiheitsgrade der einzelnen Knoten umgewandelt werden. Darüber hinaus werden die Zielgrößen (z. B. Deformationen, Spannungen, Reaktionskräfte etc.) stets für die Knoten und Zwischenknoten der Elemente berechnet. Folglich ist es für eine strukturmechanische Simulation möglich, Detektormatrizen für jede der genannten Größen zu generieren.

Exemplarisch soll dies für ein T-förmiges Bauteil kurz vorgestellt werden. Die Abmessungen und die definierten Randbedingungen des Bauteils sind in Bild 52 dargestellt.

Das Bauteil wird an der fixen Einspannung gehalten (alle rotatorischen und translatorischen Freiheitsgrade sind gesperrt) und wird an der Fläche der Krafteinleitung mit einer Kraft von  $-2000\text{ N}$  in Y-Richtung belastet. Die Vernetzung und Simulation des Bauteils erfolgt in ANSYS Workbench 19.2 und es kommen Hexaeder-Elemente mit 20 Knoten zum Einsatz.

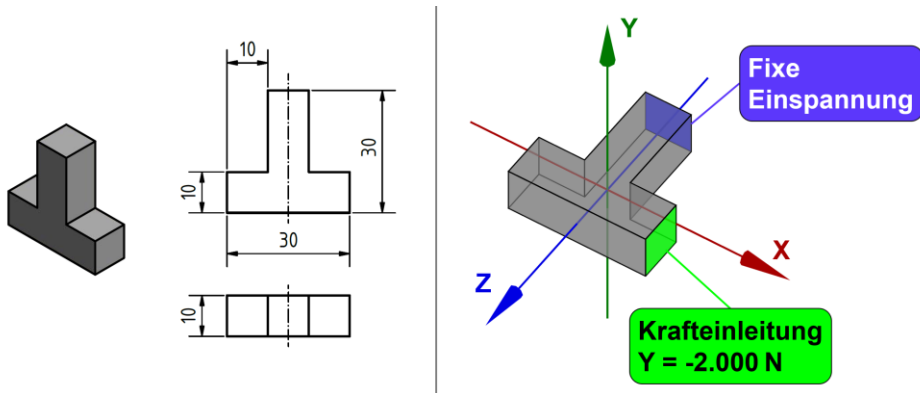


Bild 52: T-förmiges Bauteil mit Abmessungen und Randbedingungen

Das vernetzte Bauteil ist in den oberen beiden Ansichten in Bild 53 dargestellt (auf der linken Seite sind nur die Eckknoten jedes Elements in rot dargestellt, auf der rechten Seite sind zusätzlich die Zwischenknoten in türkis ergänzt). Analog zu Kapitel 4 wird eine Detektorfläche der Größe  $36 \times 36$  Pixel um das Bauteil herum erzeugt. Der Mittelpunkt der Detektorsphäre liegt im Koordinatenursprung bei  $x = 0$ ,  $y = 0$  und  $z = 0$ . ANSYS Workbench berechnet nicht alle Ergebnisgrößen einer FE-Simulation für alle Knoten. Die resultierenden Spannungen werden typischerweise nur in den Eckknoten jedes Elements berechnet. Andere Zielgrößen, wie z. B. die Deformationen werden dagegen auch in den Zwischenknoten berechnet und stehen für weitere Auswertungen direkt zur Verfügung.

Folglich lassen sich die Eingangs- und Ausgangsgrößen einer strukturellen FE-Simulation in eine Vielzahl numerischer Matrizen umwandeln. Beispielsweise können, ähnlich der Projektion der Knoten des FE-Netzes, auch nur diejenigen Knoten projiziert werden, welche eine gesperrte Verschiebung oder Rotation aufweisen. Anstelle der Aufsummation der Anzahl projizierter Punkte in einem Pixel der Detektorfläche ist es ebenso möglich Zielgrößen wie Verschiebung in  $mm$  oder Spannungen in  $MPa$  direkt aufzusummieren.

Eine exakte Auflistung und kurze Erklärung zur Erzeugung von 44 unterschiedlichen Matrizen für eine strukturelle Simulation ist in Tabelle 30 bis Tabelle 33 im Anhang zu finden.

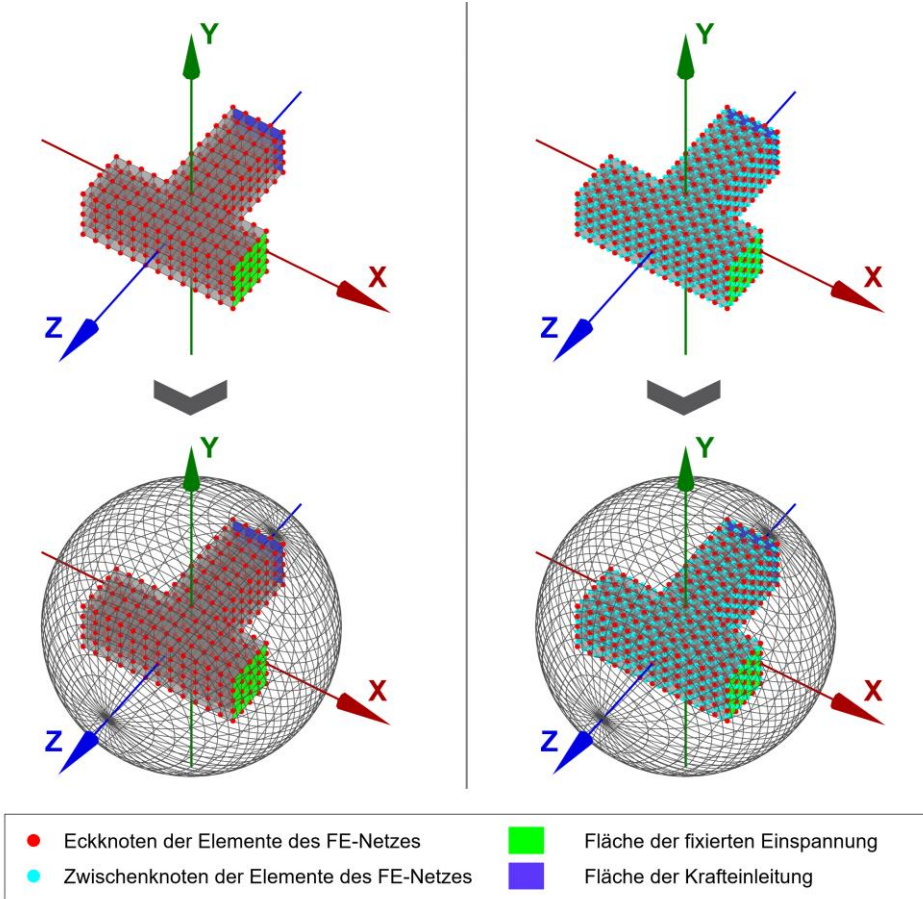


Bild 53: Vernetztes T-förmiges Bauteil mit Detektorfläche der Größe  $36 \times 36$

Die folgende Übersicht der gängigsten Randbedingungen (Tabelle 10) listet diese Matrizen auf und gibt einen kurzen Überblick über die enthaltenen Inhalte und die Anzahl an so verfügbaren Matrizen:

Tabelle 10: Auflistung der erzeugbaren Matrizen aus einer strukturmechanischen FE-Simulation

Art	Name	Richtung	Vorzeichen	Matrizenanzahl
Lagerung	Knotenposition	-	kein	1
	Verschiebung	X, Y, Z	kein	3
	Rotation	X, Y, Z	kein	3
	externe Verschiebung	X, Y, Z	±	6
Lasten	Kraft	X, Y, Z	±	6
	Moment	X, Y, Z	±	6
	Druck	normal	±	2
post-processing	Verschiebung	X, Y, Z	±	6
	Normalspannung	X, Y, Z	±	6
	maximale Hauptspannung	-	±	2
	minimale Hauptspannung	-	±	2
	von Mises Vergleichsspannung	-	kein	1
				$\sum = 44$

Zur Verdeutlichung des Vorgehens sollen einige dieser 44 Matrizen näher beleuchtet werden. Exemplarisch wird jeweils eine Matrix aus den Bereichen Lagerung, Lasten, post-processing und dem zugrunde liegenden Netz in Bild 54 gezeigt. Es handelt sich hierbei um die Matrix, welche das FE-Netz repräsentiert, die Matrix der gesperrten Translationen in X-Richtung, die Matrix der angreifenden Kraft in negativer Y-Richtung und die Matrix der aufsummierten negativen Normalspannung in X-Richtung. Die Addition der negativen Kräfte in Y-Richtung erfolgt durch die Applikation der

Kraft in Y-Richtung von  $-2000\text{ N}$  für jeden projizierten Knoten. Ebenso denkbar ist eine Addition der Äquivalenzkräfte, welche die angreifende Kraft auf die Fläche der betroffenen Elemente und anschließend auf die einzelnen Knoten dieser Fläche verteilt, aus Gründen der Nachvollziehbarkeit wird hier darauf verzichtet.

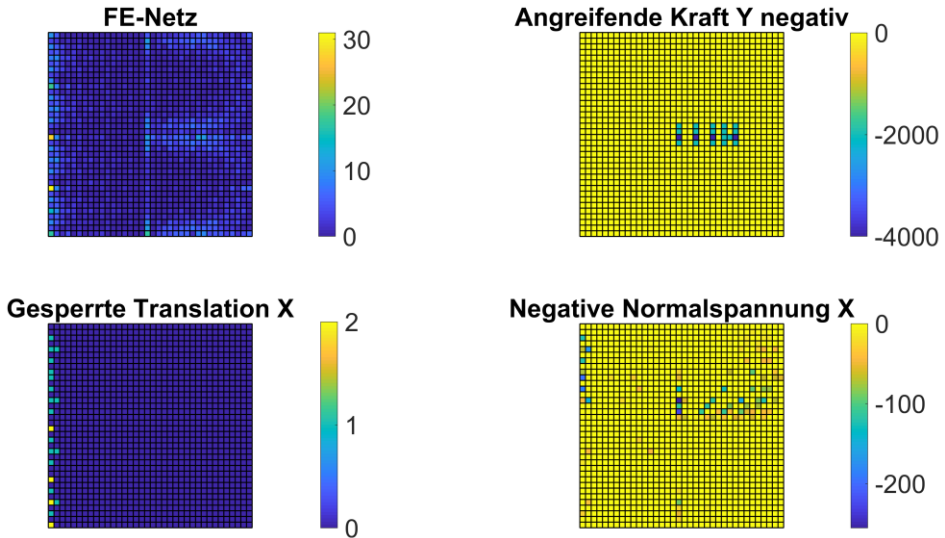


Bild 54: Matrizen des T-förmigen Bauteils für FE-Netz, angreifende Kraft in negativer Y-Richtung, gesperrte Translation in X-Richtung und negative Normalspannung in X-Richtung der Größe  $36 \times 36$

Mit dieser Erweiterung der Methodik der sphärischen Detektorflächen ist es möglich, beliebige Eingangs- und Ausgangsparameter einer FE-Simulation in numerische Matrizen einer bestimmten selbstdefinierten Größe umzuwandeln.

## 7.2 Normierung der unterschiedlichen Detektormatrizen

Um die beschriebenen Matrizen einer FE-Simulation sinnvoll an Algorithmen des Maschinellen Lernen zu übergeben, müssen diese in einen vorgegebenen Wertebereich normiert werden. Häufig ist dies der Bereich von  $-1$  bis  $+1$  oder von  $0$  bis  $+1$ . In Bild 54 ist bereits erkennbar, dass der Wertebereich der einzelnen Matrizen sehr unterschiedlich ist und von unterschiedlichen Faktoren abhängt (vgl. Tabelle 11) Es gilt also eine möglichst allgemeingültige Normierungsstrategie zu entwickeln, welche auf die meisten später zu untersuchenden Simulationen angewendet werden kann. Durch die Integration von Tangens-hyperbolicus-Funktionen kann

unerwartet hohen Werten sehr effektiv begegnet werden. Es gilt also, keine 100%-ige Lösung unter Berücksichtigungen aller Spezialfälle zu entwickeln. Im Folgenden sollen zwei unterschiedliche Normierungsstrategien vorgestellt (Strategie #1 in Kapitel 7.2.1 und Strategie #2 in Kapitel 7.2.2) und kurz auf deren individuelle Vor- und Nachteile eingegangen werden.

Tabelle 11: Abhängigkeiten der Wertebereiche der einzelnen Matrizen

Art der Matrizen	Beschreibung des Wertebereichs
Knotenmatrix	Abhängig von der Größe und Gestalt des Bauteils und der Feinheit des FE-Netzes.
Lagerungen	Abhängig von der Größe, Gestalt und Anzahl der gelagerten Flächen und der dort anzutreffenden Vernetzung.
Lasten	Abhängig von der Größe und Anzahl der gelagerten Flächen, der dort anzutreffenden Vernetzung und der Größe der angreifenden Kräfte, Momente, Verschiebungen bzw. Drücke.
Zielgrößen	Abhängig von der Größe des Bauteils und der Feinheit des FE-Netzes sowie den angreifenden Belastungen und den definierten Lagerungen.

### 7.2.1 Normierungsstrategie #1

Normierungsstrategie #1 betrachtet jede der Matrizen einzeln und ohne jeglichen Bezug zu anderen Matrizen der gleichen Simulation oder zur gleichen Matrizenart einer anderen durchgeführten FE-Simulation. Es erfolgt eine typisches min-max-Mapping in einen vorgegebenen Wertebereich von beispielsweise  $[0, +1]$  oder  $[-1, +1]$ . Um einen Zahlenwert  $x$  in den Zahlenwert  $y$  im Zielwertebereich umzuwandeln kann folgende Formel (33) verwendet werden:

$$y = \frac{(y_{max} - y_{min}) \cdot (x - x_{min})}{(x_{max} - x_{min})} + y_{min} \quad (33)$$

$x$  Zahlenwert im ursprünglichen Wertebereich  
zwischen  $[x_{min}, x_{max}]$

$y$  Zahlenwert im neuen Wertebereich  
zwischen  $[y_{min}, y_{max}]$

Der Vorteil von Normierungsstrategie #1 besteht in der einfachen Umsetzung ohne weitere Berücksichtigungen des globalen Zusammenhangs zu anderen Simulationen oder anderen Matrizen. Diese Strategie kann sehr schnell und einfach umgesetzt werden und funktioniert auch für sehr unterschiedliche Simulationen, welche einheitlich normiert werden sollen. Allerdings können durch diese Methode zwei Matrizen von unterschiedlichen Simulationen identisch aussehen. Wenn der in Bild 53 dargestellte T-förmige Körper global feiner vernetzt wird, entsteht nach der Normierung die identische Matrix wie bei einem einheitlich gröber vernetzten Bauteil. Dies kann in bestimmten Fällen zu Problemen führen und sollte daher bei der Umsetzung dieser Normierungsstrategie beachtet werden.

### 7.2.2 Normierungsstrategie #2

Im Gegensatz zu Normierungsstrategie #1 verwendet Strategie #2 einen Ansatz, welcher auf den vorliegenden Datenbestand an Simulationen und deren Matrizen angepasst werden muss. Die Normierung der unterschiedlichen Matrizen für Netz, Lagerungen, Lasten und Ergebnisgrößen erfolgt individuell:

#### **Knotenmatrix:**

Jede Knotenmatrix wird durch einen fix definierten Wert der maximal zu erwartenden Knoten in einem Pixel der Detektorfläche geteilt. Dieser Maximalwert sollte geringfügig über dem tatsächlich in den Daten vorhandenen Maximalwert liegen. Zusätzlich wird die Matrix anschließend an eine Tangens-hyperbolikus-Funktion übergeben, um eventuell höhere Werte sicher verarbeiten zu können.

#### **Translations- und Rotationsmatrizen:**

Diese Matrizen werden, analog zur Normierungsstrategie #1, durch den Maximalwert der einzelnen Matrix geteilt. Dies ist umsetzbar, da in den Matrizen weiterhin eine Aussage über fehlende oder punktuelle Lagerungen enthalten bleibt.

#### **externe Verschiebungsmatrizen:**

Die Matrizen werden durch Kennzahl  $S_1$  geteilt und anschließend an eine Tangens-hyperbolikus-Funktion übergeben.

#### **Kräfte matrizen:**

Die Matrizen werden durch Kennzahl  $S_2$  geteilt und anschließend an eine Tangens-hyperbolikus-Funktion übergeben.

**Momentenmatrizen:**

Die Matrizen werden durch Kennzahl  $S_3$  geteilt und anschließend an eine Tangens-hyperbolikus-Funktion übergeben.

**Druckmatrizen:**

Die Matrizen werden durch Kennzahl  $S_2$  geteilt und anschließend an eine Tangens-hyperbolikus-Funktion übergeben.

**Deformationsmatrizen:**

Die Matrizen werden durch Kennzahl  $S_1$  geteilt und anschließend an eine Tangens-hyperbolikus-Funktion übergeben.

**Spannungsmatrizen:**

Die Matrizen werden durch Kennzahl  $S_4$  geteilt und anschließend an eine Tangens-hyperbolikus-Funktion übergeben.

Die Definition der Kennzahlen  $S_1$  bis  $S_4$  kann den Gleichungen (34), (35), (36) und (37) entnommen werden. Eine Erklärung der jeweiligen Indizes und Abkürzungen findet sich im Anschluss an Gleichung (42).  $S_1$  bis  $S_4$  stellen maximal anzutreffende Werte der einzelnen Größen dar. So kann etwa für Verschiebungen in der FE-Simulation angenommen werden, dass diese bei plausiblen strukturmechanischen Simulationen nicht größer als ein Hundertstel der maximalen Bauteilabmessung beträgt (daher die Definition von Kennzahl  $S_1$ ).

Analog dazu wird mittels  $S_2$  ein maximal anzutreffender Wert für angreifende Kräfte bzw. Drücke definiert.  $S_3$  definiert einen Kennwert für Momente und  $S_4$  für Spannungen. Diese Kennwerte werden immer bezüglich der maximalen Abmessungen des Bauteils spezifiziert.

$$S_1 = \frac{L_{max}}{100} \tag{34}$$

$$\begin{aligned} S_2 &= 1,5 \cdot \frac{R_m \cdot maxFläche}{10.000 \cdot AnteilFläche} \tag{35} \\ &= 1,5 \cdot \frac{R_m \cdot maxFläche}{10.000 \cdot \frac{maxFläche}{2 \cdot x_{dim} \cdot y_{dim} + 2 \cdot x_{dim} \cdot z_{dim} + 2 \cdot y_{dim} \cdot z_{dim}}} \\ &= \frac{3}{10.000} \cdot R_m \cdot (x_{dim} \cdot y_{dim} + x_{dim} \cdot z_{dim} + y_{dim} \cdot z_{dim}) \end{aligned}$$

$$S_3 = \frac{1,5 \cdot \frac{L_{max}^3 \cdot R_m}{1,72^2}}{1.000 \cdot 10.000 \cdot AnteilFläche} \quad (36)$$

$$\approx 3.000.000 \cdot L_{max}^3 \cdot R_m \cdot AnteilFläche$$

$$AnteilFläche = \frac{maxFläche}{2 \cdot (x_{dim} \cdot y_{dim} + x_{dim} \cdot z_{dim} + y_{dim} \cdot z_{dim})} \quad (37)$$

Die Bestimmung von *maxFläche* soll anhand eines kurzen Beispiels mit exemplarischen Werten verdeutlicht werden. Zunächst werden die maximalen Dimensionen für alle drei Koordinatenrichtungen  $x_{dim}$ ,  $y_{dim}$  und  $z_{dim}$  ermittelt (Gleichung (38)). Im zweiten Schritt werden diese Dimensionen der Größe nach sortiert und der größte Wert mit dem zweitgrößten Wert multipliziert, um *maxFläche* zu erhalten (Gleichung (41)).

$$x_{dim} = x_{max} - x_{min} = 450 - (-15) = 465 \quad (38)$$

$$y_{dim} = y_{max} - y_{min} = -10 - (-55) = 45$$

$$z_{dim} = z_{max} - z_{min} = 125 - 30 = 95$$

$$M_1 = \max\{x_{dim}, y_{dim}, z_{dim}\} \quad (39)$$

$$M_2 = \{\{x_{dim}, y_{dim}, z_{dim}\} \mid \{x_{dim}, y_{dim}, z_{dim}\} \notin M_1\} = \{45, 95\} \quad (40)$$

$$maxFläche = M_1 \cdot \max\{M_2\} \quad (41)$$

$$S_4 = 4 \cdot R_m \quad (42)$$

$L_{max}$  maximale Bauteillänge

$R_m$  Zugfestigkeit des simulierten Werkstoffs in MPa

$x_{dim}$  maximale Dimension in X-Richtung =  $x_{max} - x_{min}$

$y_{dim}$  maximale Dimension in Y-Richtung =  $y_{max} - y_{min}$

$z_{dim}$  maximale Dimension in Z-Richtung =  $z_{max} - z_{min}$

Mit der vorgestellten Methodik ist es möglich, die Detektormatrizen der Eingangs- und Ausgangsgrößen einer strukturmechanischen FE-Simulation in Abhängigkeit der tatsächlichen Bauteilabmessungen zu normieren. Der Wertebereich der so erzeugten Detektormatrizen liegt zwischen  $-1$  und  $+1$ .

### 7.3 Aufbau der «DNA» einer Finite-Elemente-Simulation

Durch die Generierung von unterschiedlichen Matrizen für alle Eingangs- und Ausgangsgrößen einer FE-Simulation und eine nachgelagerte Normierung der Matrizen kann dieser Inhalt an Algorithmen des Maschinellen Lernens übergeben werden. Hierfür können verschiedenste Algorithmen zur Anwendung kommen. Bevor dies anhand eines umfangreichen dritten Anwendungsbeispiels in Kapitel 8 aufgezeigt wird, soll hier noch kurz auf den Begriff der «DNA» einer FE-Simulation eingegangen werden. Die Bezeichnung der «DNA» liegt darin begründet, dass die vorliegenden Matrizen visualisiert einem DNA-Muster ähneln. Zudem sind darin analog zu einer echten DNA-Analyse alle relevanten Informationen der zugrundeliegenden FE-Simulation „kodiert“ enthalten. Zur Verdeutlichung des Aufbaus dient Bild 55 mit einer vereinfachten Darstellung der einzelnen Matrizen als farbliche Blöcke.

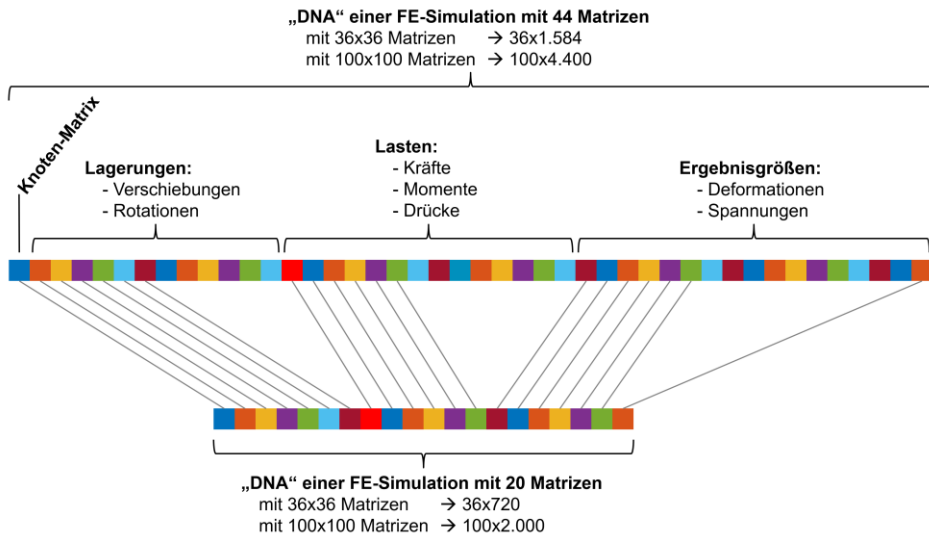


Bild 55: «DNA» einer FE-Simulation mit 44 und 20 Detektormatrizen

Nach der Normierung liegt eine bestimmte Anzahl an numerischen Matrizen vor, dies können beispielsweise 44 Stück sein, wenn für ein Anwendungsbeispiel nicht alle benötigt werden, kann auch eine reduzierte Anzahl (z. B. 20) aus diesen 44 verwendet werden. Durch die Aneinanderreihung dieser Matrizen entsteht eine große Matrix an numerischen Werten, welche als Repräsentation der FE-Simulation dient und z. B. an Künstliche Neuronale Netze übergeben werden kann. Der große Vorteil einer derartigen Konvertierung der FE-Simulation besteht in der Dimensionsreduktion auf eine fixe Anzahl numerischer Werte. Diese Modellreduktion ist notwendig, um sinnvoll Algorithmen des Maschinellen Lernen mit diesen Daten zu trainieren, zu evaluieren und schlussendlich anzuwenden. Zur Verdeutlichung der Umsetzbarkeit soll im nachfolgenden Kapitel 8 die entwickelte Methode auf einen exemplarischen Anwendungsfall übertragen werden.



## 8 Anwendung #3: Plausibilitätsprüfung für FE-Berechnungen

Mit den ersten beiden Anwendungsbeispielen konnte gezeigt werden, dass die Methode der sphärischen Detektorflächen in der Lage ist, Punktwolken in eine Matrix fixer Größe umzuwandeln und hierbei die Geometrie so abzubilden, dass eine Klassifikation der Punktwolken anhand dieser Matrizen möglich ist. Gerade im Bereich der Verknüpfung von virtueller Produktentwicklung mit durchgängiger und ganzheitlicher Datennutzung («Digital Engineering») kommt es aktuell zu einem starken Wandel und einer Veränderung der bisherigen Rollen der beteiligten Personen und der verwendeten Werkzeuge [P27]. Die Nutzbarmachung von bisher ungenutzten Datenbeständen (z. B. Simulationsdaten) bietet erhebliches Potenzial, um das darin enthaltene Wissen für zukünftige Entwicklungen und Simulationen bereitzustellen. Aktuell existieren bereits kommerzielle und akademische Lösungen, um Simulationsergebnisse von ähnlichen Simulationen miteinander zu vergleichen [P8, P13, 224]. Ein Ansatz zur Berücksichtigung von beliebigen FE-Simulationen fehlt bisher vollständig, dieses Anwendungsbeispiel soll hierfür einen Weg aufzeigen und im Rahmen eines umfangreichen Datensatzes eine erste Machbarkeit am Beispiel einer Plausibilitätsprüfung darlegen. In Kapitel 8 wird zunächst die Problemstellung vorgestellt und in Kapitel 8.2 der Aufbau der Datenbasis beschrieben. Nach der Ableitung und Normierung der Detektormatrizen in Kapitel 8.3 erfolgt der Aufbau der CNNs und das Training der gewählten Architekturen in Kapitel 8.4. Abschließend erfolgt die Ermittlung der Prognosegüte (Kapitel 8.5), eine Untersuchung der Generalisierungsfähigkeit (Kapitel 8.7) und eine Zusammenfassung und Bewertung in Kapitel 8.

### 8.1 Definition der Problemstellung

Im Rahmen des dritten Anwendungsbeispiels soll eine Plausibilitätsprüfung für strukturmechanische FE-Simulationen vorgestellt werden. Eine nicht-plausible FE-Simulation enthält offensichtliche Fehler, welche ein erfahrener Berechnungsingenieur/-in erkennen würde. Für weniger erfahrene Simulationsanwender ist eine Erkennung dieser Fehler aber nicht immer direkt möglich. So kann durch eine automatische Plausibilitätsprüfung eine wichtige Hilfestellung gegeben werden. Es werden Datenbestände aus plausiblen und nicht-plausiblen Simulationen aufgebaut und deren Eingangs- und Ausgangsgrößen in numerische Matrizen fixer Größe

umgewandelt. Nach einer Normierung werden diese Datensätze genutzt, um Convolutional Neural Networks zu trainieren, zu validieren und zu testen. Das Ziel besteht in der Umsetzung einer automatischen Plausibilitätsprüfung.

## 8.2 Aufbau der Datenbasis

Der Aufbau der Datenbasis an FE-Simulationen startet mit der Definition von unterschiedlichen CAD-Bauteilen, welche sich in Form und Größe deutlich unterscheiden. Jedes Bauteil wird als parametrisches CAD-Modell erstellt, um automatisiert Geometrievariationen zu generieren. Die Auswahl der drei unterschiedlichen Geometrien erfolgt nach der Verfügbarkeit von Randbedingungen für die späteren FE-Simulationen. Folgende Bauteile stehen als parametrisierte CAD-Modelle zur Verfügung und werden in den genannten Kapiteln näher beschrieben:

- Speed Skating Inliner-Schiene  
(vgl. Kapitel 8.2.1)
- Umlenkwappe Mountainbike, Hinterrad-Federung  
(vgl. Kapitel 8.2.2)
- Kurbelwelle eines Reihensechszylindermotors  
(vgl. Kapitel 8.2.3)

Bevor eine umfangreiche Simulationsstudie für jedes der genannten Bauteile durchgeführt werden kann, wird ein Versuchsplan erstellt. Das Vorgehen unterscheidet sich von anderen Vorgängen zur Erzeugung eines Versuchsplans für Experimente und soll daher kurz theoretisch vorgestellt werden. Ein Hauptmerkmal besteht darin, dass zwei D-optimale Versuchspläne des gleichen Versuchsraum erstellt werden und einer der so entstehenden Versuchspläne per Zufall in zwei andere Versuchspläne aufgeteilt wird (dieses Vorgehen verdeutlicht Bild 56). Der Vorteil dieses Vorgehens besteht in einer besseren Verteilung der Testdaten im Versuchsraum. Dieser Vorteil wird mit steigender Anzahl an Versuchen weniger relevant, da tendenziell auch deutlich mehr Versuchspunkte für das nachgelagerte Testen existieren. Nach der anfänglichen Definition der Gesamtanzahl geplanter Simulationen werden zwei D-optimale Versuchspläne mit einer Aufteilung von 80 % zu 20 % erstellt. Der 20 %-Datensatz kann direkt zur Erzeugung der FE-Simulationen für das spätere Testen verwendet werden. Der zweite, größere Versuchsplan wird per Zufall in einem Verhältnis von 90 zu 10 aufgeteilt.

Es ergeben sich somit die beiden Versuchspläne für das Training und die Validierung. Die Aufteilung der Gesamtanzahl von Simulationen erfolgt nach der Aufteilung 72 % zu 8 % zu 20 % auf die Datensätze Training, Validierung und Testen. Jeder dieser Versuchspläne wird verwendet, um damit strukturmechanische FE-Simulationen mit den parametrischen CAD-Modellen in ANSYS Workbench durchzuführen.

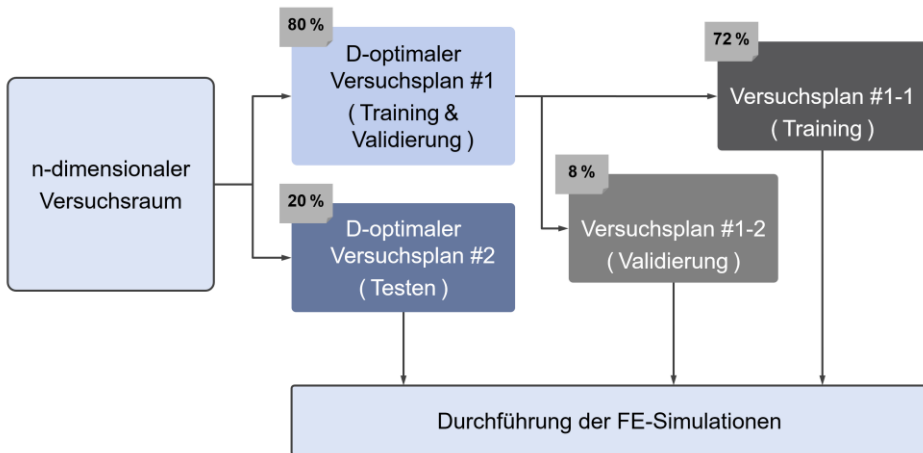


Bild 56: Prozentuale Aufteilung der Datensätze für Training, Validierung und Testen

Für einen exemplarischen Versuchsplan mit 2 Parametern (Elementgröße des FE-Netzes und eine angreifende Kraft) sehen die drei resultierenden Versuchspläne für Training, Validierung und Testen wie in Bild 57 aus. Das Ziel des D-optimalen Versuchsplans, die Versuchspunkte möglichst weit voneinander entfernt zu legen, ist deutlich erkennbar. Ebenso ist ersichtlich, dass einige Testpunkte in dieser Konstellation auf den gleichen Versuchspunkten wie die des Trainings- bzw. Validierungsversuchsplan liegen. Der entscheidende Vorteil dieses Vorgehens besteht in der optimalen Ausnutzung der Grenzen des Versuchsraums, sowohl für die kombinierten Trainings-Validierungs-Daten als auch für die Test-Daten. In Bild 57 ist ebenfalls der Bereich der nicht-plausiblen Versuchspunkte aufgrund von zu geringer angreifender Kraft oder zu groß gewählter Elementgröße des FE-Netzes dargestellt.

Das Ziel bei der Festlegung der nicht-plausiblen Faktorstufen des Versuchsplans ist dabei durchgängig, dass ein ausgewogenes Verhältnis zwischen plausiblen und nicht-plausiblen Versuchspunkten erreicht wird. Es soll kein verzerrter (unbiased) Datensatz erzeugt werden.

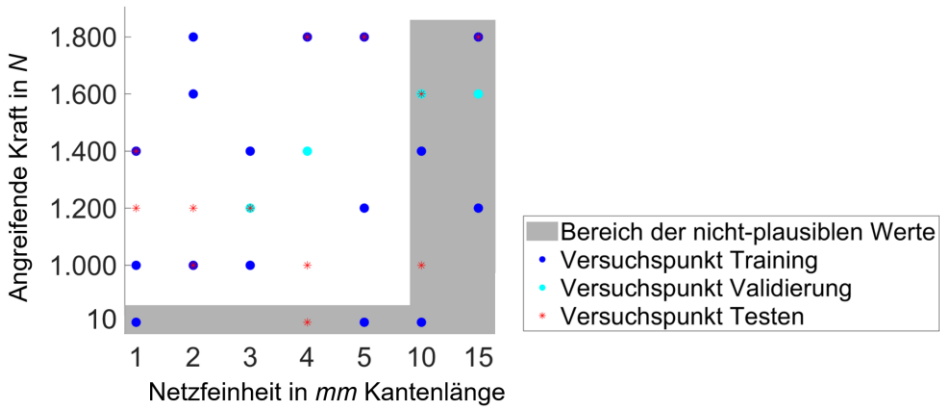


Bild 57: Erklärung zur Erzeugung des Versuchsplan (hier 2-dimensionaler Versuchsraums)

In dem in Bild 57 dargestellten Beispiel liegt das Verhältnis von plausiblen zu nicht-plausiblen Datensätzen für die Trainings-Validierungsdaten bei 12 zu 8 und für die Testdaten bei 8 zu 4.

### 8.2.1 Bauteil Inliner-Schiene

Das erste parametrische Bauteil des Datensatzes ist eine Speed Skating Inliner-Schiene. An dieser Schiene werden 4 Achsen für die Rollen und der Schuh befestigt. Der Durchmesser der zu montierenden Räder beträgt 76 mm, 80 mm oder 90 mm. Die Wandstärke der Aluminiumschiene kann jeweils entweder 2 mm, 3 mm, 4 mm oder 5 mm betragen. Es ergeben sich vollfaktoriell 12 Geometriemodelle, welche in diesem ersten Datensatz auch alle für jeden Punkt des Versuchsplans berechnet werden. Folglich sind beim ersten Bauteil die gleichen Geometriemodelle im Trainings-, Validierungs- und Testdatensatz enthalten. Dies ist eine Besonderheit des ersten Bauteils und tritt in dieser Form nur hier auf. Mit den zuvor erwähnten Versuchsplänen werden folgende Faktoren mit den zugehörigen Faktorstufen in Tabelle 12 definiert.

Der Aufbau der FE-Simulation erfolgt invers zur eigentlichen Belastung. An den Montagepunkten des späteren Schuhs wird die Schiene fix eingespannt, es werden an diesen Flächen alle Translationen und Rotationen um X, Y und Z blockiert. Die Belastungen durch eine Person werden in Form von Kraft 1 in die Simulation integriert und Kraft 2 gibt eine Belastung an der vordersten Rolle der Schiene vor.

Beispielsweise in Folge einer leichten Kollision des ersten Rades im Rahmen des üblichen Betriebs der Skates. Normalerweise wird die Schiene im unteren Bereich zusätzlich durch die verschraubten Achsen verstärkt, in der Simulation wird dies durch gesperrte Bewegung in X-Richtung an der Außenkante der Bohrungen für die Achsen realisiert. Den Aufbau der Simulation verdeutlicht exemplarisch Bild 58.

Tabelle 12: Versuchsplan Inliner-Schiene (nicht-plausible Faktorstufen sind farblich hervorgehoben)

Faktor- stufen Nr.	Faktoren			
	Elementgröße	Kraft 1 Y	Kraft 1 Z	Kraft 2 Y
1	1 mm	0 N	1.000 N	-1.000 N
2	1,5 mm	200 N	1.500 N	-800 N
3	2 mm	400 N	2.000 N	-600 N
4	3 mm	600 N	3.000 N	-400 N
5	4 mm	800 N	20.000 N	-60 N
6	5 mm	-	-	-
7	10 mm	-	-	-
8	15 mm	-	-	-

Sobald ein Parameter des Versuchspunktes «nicht-plausibel» ist, ist die zugehörige Simulation als «nicht-plausibel» zu bewerten. Hieraus ergibt sich ein ausgewogenes Verhältnis aus «plausiblen» und «nicht-plausiblen» Simulationen, obwohl mehr Faktorstufen eines Parameters in die Kategorie «plausibel» fallen.

Der so umgesetzte Versuchsplan liefert folgende Zusammensetzung an «plausiblen» (48,125 %) und «nicht-plausiblen» (51,875 %) FE-Simulationen (folglich kann durch stetige Auswahl der Kategorie «nicht-plausibel» durch einen Algorithmus bereits eine Prognosegüte von 51,875% erreicht werden):

- **Training** (6.912 Simulationen):
  - 3.326 FE-Simulationen plausibel (48,119 %)
  - 3.586 FE-Simulationen nicht-plausibel (51,881 %)
- **Validierung** (768 Simulationen):
  - 370 FE-Simulationen plausibel (48,177 %)
  - 398 FE-Simulationen nicht-plausibel (51,823 %)
- **Testen** (1.920 Simulationen):
  - 924 FE-Simulationen plausibel (48,125 %)
  - 996 FE-Simulationen nicht-plausibel (51,875 %)

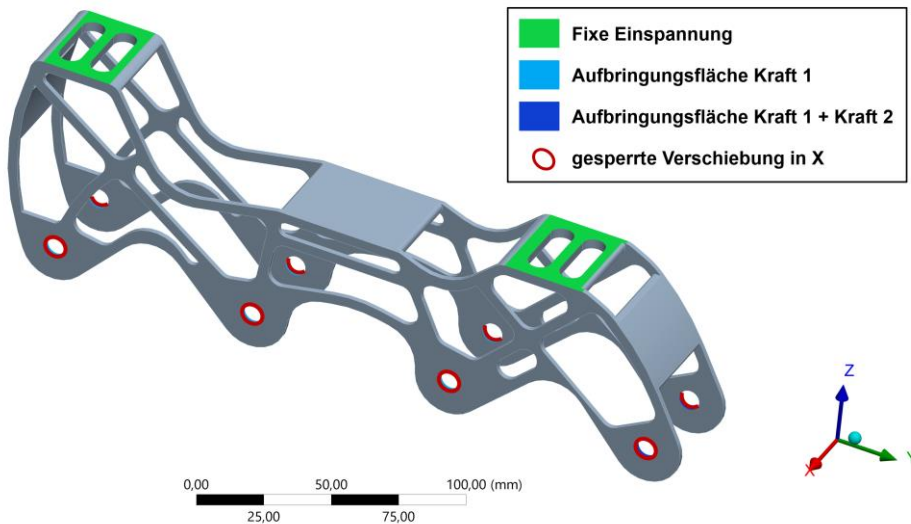


Bild 58: Aufbau der FE-Simulation für das Bauteil Inliner Schiene

Für jede der  $6.912 + 768 + 1.920 = 9.600$  strukturmechanischen Simulationen stehen das FE-Netz, die Knoten der fixierten Einspannung, die Knoten der gesperrten Kanten in X-Richtung, die Knoten der Angriffsflächen von Kraft 1 und 2 sowie die resultierenden Deformationen an allen Knoten und die Spannungen der Eckknoten aller Elemente zur Verfügung.

### 8.2.2 Bauteil Mountainbike-Wippe

Das zweite parametrische Bauteil des Datensatzes ist ein Umlenkhebel/Wippe einer Hinterradfederung eines Mountainbikes (MTB), mittels dieser Wippe wird der Bewegungsweg der Hinterradschwinge auf den Stoßdämpfer und die Feder umgelenkt. Rotatorisch gelagert ist die Wippe am Rahmen des Mountainbikes. Es handelt sich hierbei um ein massives Aluminiumbauteil mit Wandstärken zwischen 8 mm und 10 mm sowie Vertiefungslaschen zwischen 4 mm und 6 mm Tiefe.

Tabelle 13: Versuchsplan MTB-Wippe (nicht-plausible Faktorstufen sind farblich hervorgehoben)

Faktor- stufen Nr.	Faktoren		
	Elementgröße	Neigungswinkel	Kraftwert
1	1 mm	0°	10 N
2	1,5 mm	2,5°	40 N
3	2 mm	5°	1.000 N
4	2,5 mm	7,5°	1.500 N
5	3 mm	10°	2.000 N
6	15 mm	12,5°	2.500 N
7	-	15°	3.000 N
8	-	17,5°	3.500 N
9	-	20°	4.000 N
10	-	22,5°	10.000 N
11	-	25°	-
12	-	30°	-
13	-	35°	-
14	-	65°	-

Neben den Bohrungsdurchmessern ( $\varnothing 6 \text{ mm} - \varnothing 10 \text{ mm}$ ) variiert ebenfalls der Öffnungswinkel der beiden Arme von  $14^\circ$  bis  $20^\circ$  und der Achsabstand der äußeren Bohrungen von  $70 \text{ mm}$  bis  $80 \text{ mm}$ .

Vom Bauteil MTB-Wippe werden 27 Geometrien erzeugt, wobei 5 dieser Geometrien für den Testdatensatz verwendet werden. Anders als bei Bauteil Inliner-Schiene enthält der Testdatensatz der MTB-Wippe ähnliche, aber unbekannte Bauteile gegenüber dem Trainings- bzw. Validierungsdatensatz. Mit den zuvor erwähnten Versuchsplänen werden folgende Faktoren mit den zugehörigen Faktorstufen in Tabelle 13 definiert.

Mithilfe des Neigungswinkels und des Kraftwerts kann die Aufteilung der Kraft in  $F_x$ ,  $F_y$  und  $F_z$  nach Gleichungen (43) - (45) berechnet werden.

$$F_x = \frac{32 \cdot \text{Kraftwert} \cdot \cos(\text{Neigungswinkel})}{\sin(\alpha) \cdot l - 22,5} \quad (43)$$

$$F_y = \frac{32 \cdot \text{Kraftwert} \cdot \sin(\text{Neigungswinkel})}{\sin(\alpha) \cdot l - 22,5} \quad (44)$$

$$F_z = 1,7 \cdot \text{Kraftwert} \quad (45)$$

$\alpha$  mittlerer Schenkelwinkel (hier =  $81,5^\circ$ )

$l$  mittlere Schenkellänge (hier =  $65 \text{ mm}$ )

Der Aufbau der FE-Simulation erfolgt analog zu einer Belastungsrichtung, bei der ein Mountainbike nach einem Sprung auf dem Hinterrad aufkommt und die Belastung über die Wippe in den Stoßdämpfer eingeleitet wird. Ein schräges Aufkommen des Hinterrads wird durch den Neigungswinkel ebenfalls mitberücksichtigt. Zu den tatsächlich am Rahmen auftretenden Belastungen gibt es nur sehr wenige verfügbare Angaben. Nach DIN EN ISO 4210-6 werden  $2.800 \text{ N}$  auf das Hinterrad eines umgedrehten Fahrrads aufgebracht, um die Reifenfreigängigkeit zu testen [225]. Eine Belastung von bis zu  $14.000 \text{ N}$  wird durch [226] für vertikale Kräfte angegeben, welche durch Unebenheiten in den Rahmen eines Mountainbikes eingeleitet werden. Für das vorliegende Beispiel der MTB-Wippe werden Kräfte von  $1.000 \text{ N}$  bis  $4.000 \text{ N}$  direkt an der Aufnahmebohrung der Wippe angenommen. Größere und kleinere Werte werden als nicht-plausibel klassifiziert. Den Aufbau der Simulation verdeutlicht exemplarisch Bild 59.

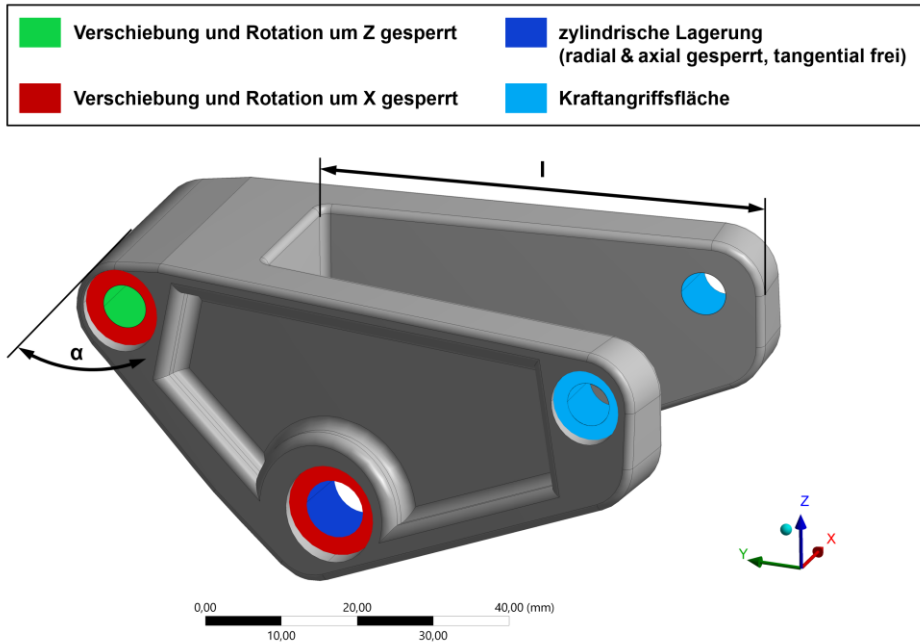


Bild 59: Aufbau der FE-Simulation für das Bauteil MTB Wippe

Der umgesetzte Versuchsplan liefert folgende Zusammensetzung an plausiblen (54,167 %) und nicht-plausiblen (45,833 %) FE-Simulationen:

- **Training** (16.632 Simulationen):
  - 9.009 FE-Simulationen plausibel (54,167 %)
  - 7.623 FE-Simulationen nicht-plausibel (45,833 %)
- **Validierung** (1.848 Simulationen):
  - 1.001 FE-Simulationen plausibel (54,167 %)
  - 847 FE-Simulationen nicht-plausibel (45,833 %)
- **Testen** (4.200 Simulationen):
  - 2.275 FE-Simulationen plausibel (54,167 %)
  - 1.925 FE-Simulationen nicht-plausibel (45,833 %)

Für jede der  $16.632 + 1.848 + 4.200 = 22.680$  strukturmechanischen Simulationen stehen das FE-Netz, die Knoten der Flächen mit gesperrter Verschiebung, die Knoten der Flächen der zylindrischen Lagerung, die Knoten der Angriffsflächen der Kraft sowie die resultierenden Deformationen an allen Knoten und die Spannungen der Eckknoten aller Elemente zur Verfügung.

### 8.2.3 Bauteil Kurbelwelle R6

Das dritte parametrische Bauteil des Datensatzes ist eine Kurbelwelle eines 6-Zylinder-Reihenmotors (vgl. Bild 6o), welche die translatorischen Bewegungen der Kolben in eine rotatorische Bewegung umwandelt. Es handelt sich um ein längliches massives Bauteil aus geschmiedetem Stahl. Die Kurbelwelle ist an 7 Lagerstellen rotatorisch gelagert, die Sperrung der übrigen translatorischen und rotatorischen Bewegungen in Z-Richtung erfolgt durch eine zusätzliche Lagerung an der Keilwellenverzahnung. Neben den durch die Ausgleichsgewichte hervorgerufenen Ausgleichkräfte werden die Gaskräfte auf den jeweiligen Pleuelsitzen aufgebracht. Die Ausgleichskräfte wirken durch die Rotation der Kurbelwelle im Betrieb stets nach außen, die Gaskräfte besitzen jeweils eine X- und Y-Komponente und variieren je nach Zündreihenfolge des Motors und untersuchtem Zeitpunkt im 4-Takt-Verbrennungsprozess.

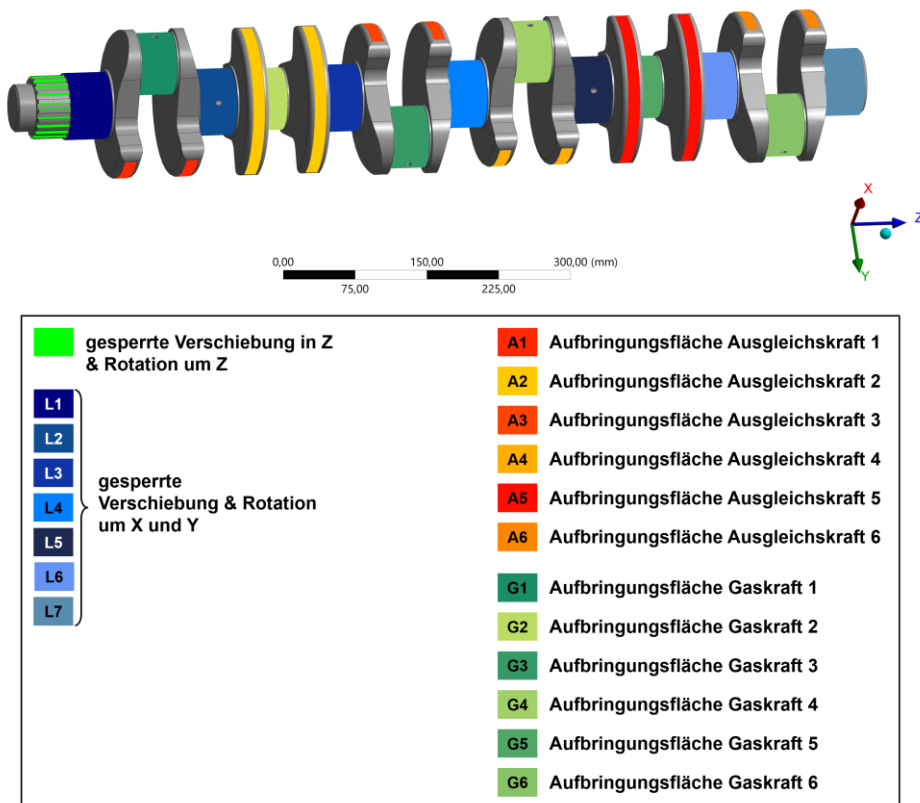


Bild 6o: Aufbau der FE-Simulation für das Bauteil Kurbelwelle R6

Das Modell der Kurbelwelle ist parametrisch aufgebaut und ermöglicht die Manipulation der folgenden Geometriegrößen in den angegebenen Grenzen (Tabelle 14 zeigt die Faktorstufen des gewählten Versuchsplan):

- Kurbelradius (44 mm - 51 mm)
- Durchmesser der Wellenzapfen (Ø50 mm - Ø70 mm)
- Durchmesser der Hubzapfen (Ø45 mm - Ø65 mm)
- Breite des Hauptwellenzapfenlagers (35 mm - 45 mm)
- Breite der Nebenwellenzapfenlager (32 mm - 40 mm)
- Lagerbreite Pleuel (30 mm - 38 mm)
- Wangenbreite (18 mm - 24 mm)

Tabelle 14: Versuchsplan Kurbelwelle R6 (nicht-plausible Faktorstufen farblich hervorgehoben)

Faktorstufen Nr.	Faktoren					
	Kolbendurchmesser	Kurbelradius	Zünddruck	Maximaldrehzahl	Durchmesser Wellenzapfen	Elementgröße
1	Ø80 mm	44 mm	180 bar	4.500 min <sup>-1</sup>	Ø50 mm	3 mm
2	Ø82 mm	46 mm	190 bar	6.500 min <sup>-1</sup>	Ø55 mm	3,5 mm
3	Ø84 mm	48 mm	200 bar	-	Ø60 mm	4 mm
4	Ø86 mm	49 mm	210 bar	-	Ø65 mm	4,5 mm
5	Ø87 mm	50 mm	220 bar	-	Ø70 mm	5 mm
6	Ø88 mm	51 mm	1.900 bar	-	-	10 mm
7	Ø89 mm	-	-	-	-	-
8	Ø90 mm	-	-	-	-	-

Die Anpassung der anderen Geometriegrößen richtet sich nach dem vorgegebenen Durchmesser des Wellenzapfens. Eine Auflistung der Parameter «Durchmesser Hubzapfen», «Breite des Hauptwellenzapfens», «Breite der Nebenwellenzapfen», «Lagerbreite am Wellenzapfen» und der «Wangenbreite» in Abhängigkeit des gewählten «Durchmesser Wellenzapfen» zeigt Tabelle 15.

Tabelle 15: Definition der anderen Geometrieparameter der Kurbelwelle in Abhängigkeit des Durchmessers des Wellenzapfens

<b>Durchmesser Wellenzapfen</b>	$\varnothing 50 \text{ mm}$	$\varnothing 55 \text{ mm}$	$\varnothing 60 \text{ mm}$	$\varnothing 65 \text{ mm}$	$\varnothing 70 \text{ mm}$
Durchmesser Pleuellager	$\varnothing 45 \text{ mm}$	$\varnothing 50 \text{ mm}$	$\varnothing 55 \text{ mm}$	$\varnothing 60 \text{ mm}$	$\varnothing 65 \text{ mm}$
Lagerbreite Pleuel	$30 \text{ mm}$	$32 \text{ mm}$	$34 \text{ mm}$	$36 \text{ mm}$	$38 \text{ mm}$
Lagerbreite Hauptlager	$35 \text{ mm}$	$36 \text{ mm}$	$39 \text{ mm}$	$42 \text{ mm}$	$45 \text{ mm}$
Lagerbreite Nebenlager	$32 \text{ mm}$	$34 \text{ mm}$	$36 \text{ mm}$	$38 \text{ mm}$	$40 \text{ mm}$
Wangenbreite	$18 \text{ mm}$	$19 \text{ mm}$	$20 \text{ mm}$	$22 \text{ mm}$	$24 \text{ mm}$

Für die nachgelagerte Ermittlung der Ausgleichskräfte werden die Massen von Pleuel und Kolben benötigt. Mittels Tabelle 16 und Tabelle 17 werden vereinfacht Massen für Pleuel und Kolben in Abhängigkeit von «Kolbendurchmesser» bzw. «Kurbelradius» und «Hauptlagerdurchmesser» herangezogen. Die jeweiligen Massen sind von gängigen Pleueln und Kolben von PKW-Dieselmotoren mittels Inter- und Extrapolation abgeleitet.

Tabelle 16: Kolbenmasse in Gramm in Abhängigkeit des Kolbendurchmessers

<b>Kolbendurchmesser</b>							
80 mm	82 mm	84 mm	86 mm	87 mm	88 mm	89 mm	90 mm
⇕	⇕	⇕	⇕	⇕	⇕	⇕	⇕
775 g	832 g	872 g	922 g	938 g	954 g	979 g	1.012 g

**Kolbenmasse**

Tabelle 17: Pleuelmasse in Gramm in Abhängigkeit von Kurbelradius und Hauptlagerdurchmesser der Kurbelwelle

		<b>Kurbelradius</b>					
		44 mm	46 mm	48 mm	49 mm	50 mm	51 mm
<b>Hauptlager Ø</b>	50 mm	819 g	839 g	859 g	869 g	879 g	889 g
	55 mm	946 g	968 g	990 g	1.001 g	1.012 g	1.023 g
	60 mm	1.234 g	1.265 g	1.296 g	1.312 g	1.328 g	1.343 g
	65 mm	1.391 g	1.425 g	1.460 g	1.477 g	1.494 g	1.512 g
	70 mm	1.558 g	1.596 g	1.634 g	1.653 g	1.672 g	1.690 g

Die Masse eines einzelnen Gegengewichts an einer Wange der untersuchten Kurbelwelle kann näherungsweise mit Formel (46) berechnet werden. Mit dieser Masse kann dann die resultierende Ausgleichskraft von zwei Ausgleichsmassen, also die Ausgleichskraft an einem Zylinder, mit Formel (47) berechnet werden.

$$m_{\text{Ausgleichsmasse}} = \left( r + \left( \frac{d_{\text{pl}}}{2} + 0,5 \right) \right)^2 \cdot \pi \cdot b_{\text{Wange}} \cdot \frac{\beta}{360^\circ} \cdot \rho_{\text{Stahl}} \quad (46)$$

$$F_{\text{Ausgleich Zylinder}} = 2 \cdot m_{\text{Ausgleichsmasse}} \cdot (2 \cdot \pi \cdot n)^2 \cdot r_s \quad (47)$$

$$\text{mit } r_s = \frac{\left( r + \left( \frac{d_{\text{pl}}}{2} + 0,5 \right) \right) \cdot \sin(78^\circ)}{\pi \cdot \frac{\beta}{360^\circ}}$$

$r$  Kurbelradius

$d_{\text{pl}}$  Pleuellagerdurchmesser

$b_{\text{Wange}}$  Wangenbreite

$\beta$  Öffnungswinkel des Ausgleichsgewichts

$\rho_{\text{Stahl}}$  Dichte von Stahl (hier =  $0,00786 \frac{\text{kg}}{\text{cm}^3}$ )

$r_s$  Radius des Massenschwerpunkts

$n$  Drehzahl der Kurbelwelle in  $\frac{1}{\text{s}}$

Die Berechnung der resultierenden Kräfte, welche durch die Gaskräfte der sechs Zylinder des Verbrennungsmotors auf die Hubzapfen hervorgerufen werden, können nach [227, 228, S9] berechnet werden. Es handelt sich um einen funktionalen Zusammenhang, bestehend aus folgenden Eingangsgrößen:

- Kurbelwinkel
- Zylinderanzahl bzw. Winkelversatz der Hubzapfen auf der Kurbelwelle
- Kolbendurchmesser
- Zünddruck
- Pleuellänge
- Pleuelstangenverhältnis (hier = 0,3)
- Kurbelradius
- Kolbenmasse
- Pleuelmasse
- Drehzahl der Kurbelwelle

Der Aufbau der FE-Simulationen der Kurbelwelle ist aufgrund einer höheren Anzahl an Randbedingungen und angreifenden Kräften aufwendiger als in den beiden anderen Parameterstudien der MTB-Wippe und der Inliner-Schiene. Die Kurbelwelle wird über sieben Lagerstellen gelagert, wobei an jeder Lagerstelle die Verschiebung in und die Rotation um die Z-Richtung weiterhin frei bleibt. Diese beiden Freiheitsgrade werden über die Lagerung der Keilwelle am Ende der Kurbelwelle gesperrt. Die beiden Ausgleichsmassen an jedem Zylinder resultieren in einer entsprechenden Ausgleichskraft, welche nach außen in Richtung der Fliehkraft aufgebracht wird. Zusätzlich greifen an jedem Hubzapfen der Kurbelwelle die resultierenden Kräfte der Gaskraft in X- und Y-Richtung an. In der FE-Parameterstudie wird die Höchstkraft der Zündung jedes Zylinders mit den zusätzlich resultierenden Kräften durch den 4-Takt-Prozess an den anderen Zylindern aufgebracht. Da die größten Kräfte am zündenden Zylinder des 6-Zylinderreihenmotors auftreten, werden sechs Simulationen für jede Parameterkombination aufgebaut, um die Zündung jedes einzelnen Zylinders zu berücksichtigen. Den Aufbau der Randbedingungen der FE-Simulation der Kurbelwelle verdeutlicht Bild 60.

Der umgesetzte Versuchsplan liefert folgende Zusammensetzung an plausiblen und nicht-plausiblen FE-Simulationen:

- Training (7.776 Simulationen):
  - 5.184 FE-Simulationen plausibel (66,667 %)
  - 2.592 FE-Simulationen nicht-plausibel (33,333 %)
- Validierung (864 Simulationen):
  - 576 FE-Simulationen plausibel (66,667 %)
  - 288 FE-Simulationen nicht-plausibel (33,333 %)
- Testen (2.160 Simulationen):
  - 1.446 FE-Simulationen plausibel (66,994 %)
  - 714 FE-Simulationen nicht-plausibel (33,056 %)

Für jede der  $7.776 + 864 + 2.160 = 10.800$  strukturmechanischen Simulationen stehen das FE-Netz, die Knoten der 7 Lagerungen mit gesperrter Verschiebung und Rotation, die Knoten der Flächen der Keilwelle mit gesperrter Verschiebung und Rotation, die Knoten der 12 Angriffsflächen der Kräfte der Ausgleichsgewichte, die Knoten der 6 Hubzapfen der angreifenden resultierenden Gaskräfte sowie die resultierenden Deformationen an allen Knoten und die Spannungen der Eckknoten aller Elemente zur Verfügung.

### 8.3 Ableitung der «DNA» der FE-Simulation

Nach der Durchführung der FE-Parameterstudien stehen insgesamt 43.080 Simulationen mit den relevanten Eingangs- und Ausgangsgrößen als Textdateien im .txt-Format zur Verfügung. Einen Überblick über die Zusammensetzung dieser Simulationen liefert Tabelle 18.

Tabelle 18: Übersicht der zur Verfügung stehenden FE-Simulationen mit allen relevanten Eingangs- und Ausgangsgrößen

	Training		Validierung		Testen		
	plausibel (p)	nicht-plausibel (np)	plausibel (p)	nicht-plausibel (np)	plausibel (p)	nicht-plausibel (np)	
Inliner-schiene	3.326	3.586	370	398	924	996	$\sum_p = 4.620$ $\sum_{np} = 4.980$
	$\sum = 6.912$		$\sum = 768$		$\sum = 1.920$		
	$\sum = 9.600$						
MTB-Wippe	9.009	7.623	1.001	847	2.275	1.925	$\sum_p = 12.285$ $\sum_{np} = 10.395$
	$\sum = 16.632$		$\sum = 1.848$		$\sum = 4.200$		
	$\sum = 22.680$						
R6-Kurbelwelle	5.184	2.592	576	288	1.446	714	$\sum_p = 7.206$ $\sum_{np} = 3.594$
	$\sum = 7.776$		$\sum = 864$		$\sum = 2.160$		
	$\sum = 10.800$						

---

Gesamtanzahl Simulationen = 43.080

---

Gesamtanzahl plausibler Simulationen = 24.111  $\Rightarrow$  55,968 %

---

Gesamtanzahl nicht-plausibler Simulationen = 18.969  $\Rightarrow$  44,032 %

---

Die Simulationen erfolgten mittels ANSYS Workbench 19.2 und der Gesamtumfang der Simulationsdaten beträgt 12,991 TB. Die Aufteilung auf die einzelnen Bauteilgruppen und Datensätze zeigt Tabelle 19. Mit der zuvor in Kapitel 7 beschriebenen Methode ist es möglich die «DNA» jeder dieser 43.080 Simulationen abzuleiten. Aufgrund der vorliegenden Simulationsdaten (weder Drücke, Momente noch externe Verschiebungen) sollen lediglich 20 Matrizen verwendet werden, um damit nachgelagert eine Klassifikation in «plausibel» bzw. «nicht-plausibel» durchzuführen.

Tabelle 19: Datenumfang Rohdaten der zur Verfügung stehenden FE-Simulationen in TB

<b>Bauteilart</b>	<b>Training</b>	<b>Validierung</b>	<b>Testen</b>
Inliner-Schiene	1,080 TB	0,120 TB	0,410 TB
MTB-Wippe	2.009 TB	0,224 TB	0,558 TB
R6-Kurbelwelle	6,210 TB	0,690 TB	1,690 TB
$\Sigma$	9,299 TB	1,034 TB	2,658 TB
$\Sigma = 12,991 TB$			

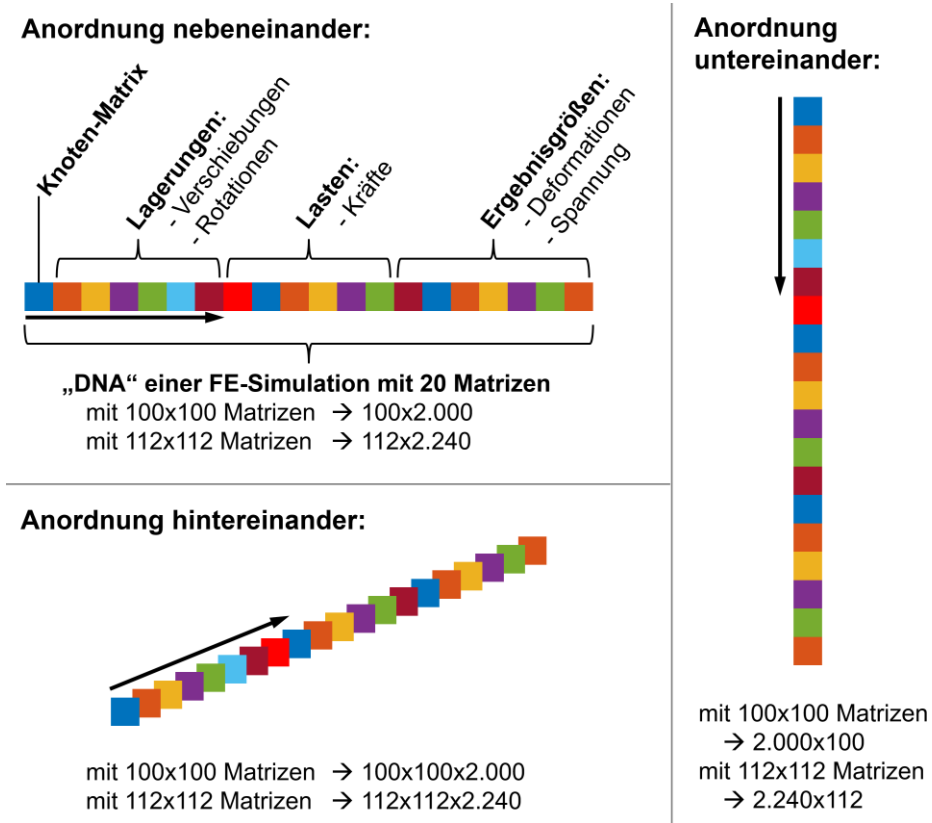


Bild 61: Unterschiedliche Anordnungen der Detektormatrizen zur Bildung der «DNA» einer FE-Simulation

Die Anordnung der einzelnen Matrizen für das nachgelagerte CNN kann hierbei nebeneinander, untereinander oder hintereinander erfolgen, dies verdeutlicht Bild 61 anhand von farblichen Matrizen für die verwendeten 20 Matrizen. Zusätzlich zur Anordnung der Matrizen kann ebenfalls die Größe der einzelnen Matrizen angepasst werden. Für die folgenden Untersuchungen werden zwei unterschiedliche Matrizengrößen betrachtet. Zum einen eine Größe von 100x100 Pixeln und zum anderen eine vergrößerte Variante mit 112x112 Pixeln. Die größere Variante entsteht durch das Spiegeln von Elementen der 100x100 Matrix auf die gegenüberliegenden Seiten der Matrix. Dies verdeutlicht Bild 62. Es dient der Integration der zusätzlichen Information, dass die Pixel an den Rändern der Matrix auf der Kugeloberfläche nebeneinander liegen, in der Matrix aber keinerlei lokale Verbindung mehr besitzen.

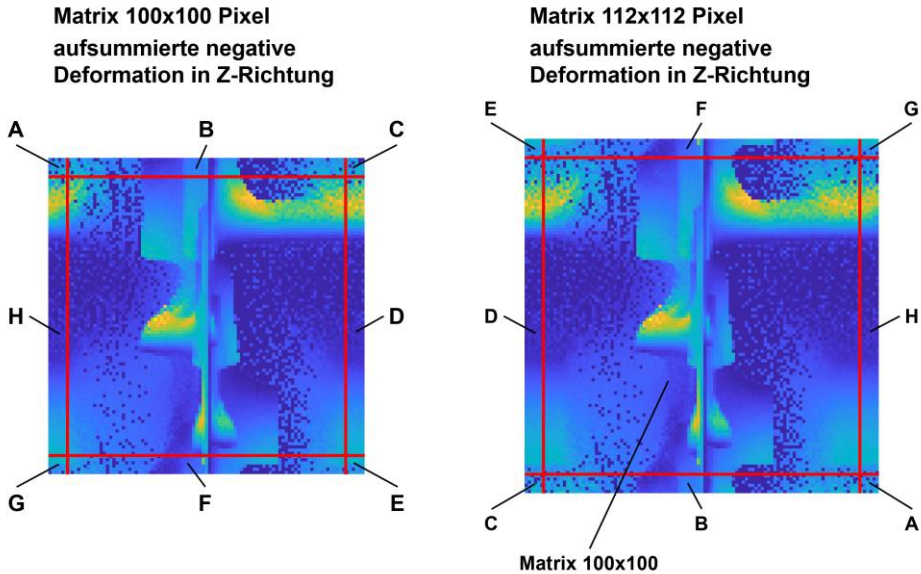


Bild 62: Erweiterung der Detektormatrix von 100x100 auf 112x112 Pixel

Bevor mit den Matrizen ein Algorithmus des Maschinellen Lernen ange-  
lernt werden kann, müssen die Matrizen normiert werden. Es werden so-  
wohl Normierungsstrategie #1 (vgl. Kapitel 7.2.1) als auch Normierungsstra-  
tegie #2 (vgl. Kapitel 7.2.2) verwendet. Das Ergebnis nach der Normierung  
ist für eine plausible Simulation der Kurbelwelle sowohl für 100x100 als  
auch für 112x112 und für die beiden Normierungsstrategien in Bild 63 ge-  
zeigt. Hierbei werden die einzelnen Matrizen mit Abkürzungen belegt, z. B.  
«ux» für die gesperrten Verschiebungen in X-Richtung oder «Fy neg» für  
die aufgebrachten negativen Kräfte in Y-Richtung.

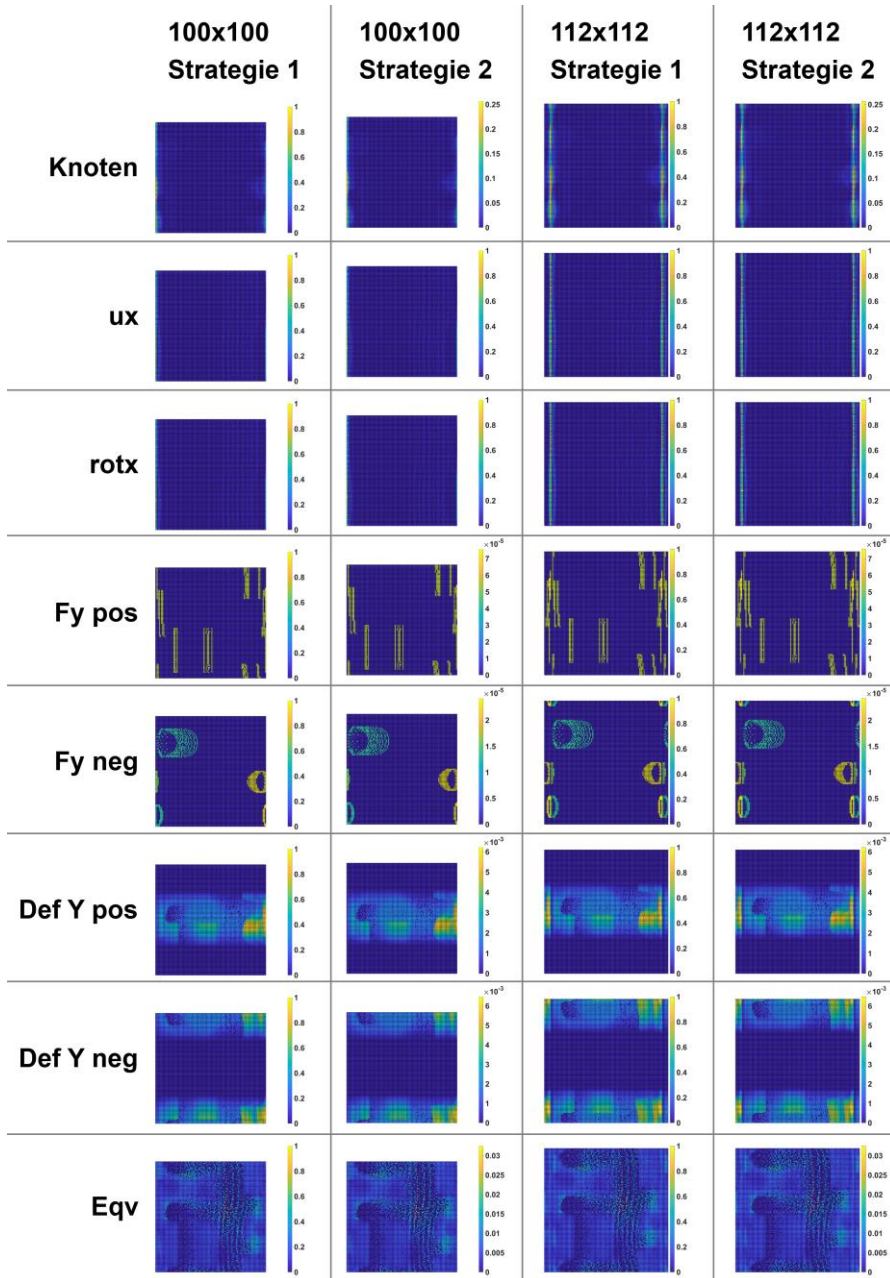


Bild 63: Vergleich der Normierungsstrategien #1 und #2 mit unterschiedlicher Matrizen-größe für die FE-Größen: Netz (Knoten), gesperrte Verschiebung in X-Richtung (ux), gesperrte Rotation in X-Richtung (rot x), aufgebrachte positive Kraft in Y-Richtung (Fy pos), aufgebrachte negative Kraft in Y-Richtung (Fy neg), positive Deformation in Y-Richtung (Def Y pos), negative Deformation in Y-Richtung (Def Y neg), von Mises Vergleichsspannung (Eqv)

## 8.4 Architektur und Training der Convolutional Neural Networks

Nach der Erzeugung der FE-Simulationen, der Umwandlung in numerische Matrizen mittels der Methode der sphärischen Detektorflächen, der Normierung dieser Matrizen und der Anordnung der 20 Matrizen für eine Simulation gilt es einen geeigneten Algorithmus des Maschinellen Lernen auszuwählen, um eine automatische Plausibilitätsprüfung auf Basis dieser Daten umsetzen zu können. Die Anzahl an Eingangsparametern limitiert die Auswahl verfügbarer Algorithmen bereits erheblich. Bedingt durch die Matrizengröße von  $112 \times 112$  Pixeln mit 20 unterschiedlichen Matrizen existieren bereits 250.880 numerische Eingangswerte. Zudem existieren im Trainingsdatensatz 31.320 Simulationen mit jeweils 250.880 der erwähnten Eingangsparameter. Aus diesem Grund soll im Folgenden eine Umsetzung mittels Convolutional Neural Networks erfolgen. In einem ersten Schritt wird in Kapitel 8.4.1 und Kapitel 8.4.2 eine selbst gewählte CNN-Architektur mit unterschiedlicher Matrizenanordnung vorgestellt und mit den beschriebenen Daten vollständig angeleert. In diesem ersten Schritt sind leichte Anpassungen an der Netzarchitektur notwendig, beispielsweise um auch die Datensätze mit den  $100 \times 100$ -Matrizen an ein Netz mit ähnlicher Architektur wie die Datensätze mit  $112 \times 112$ -Matrizen übergeben zu können. In einem zweiten Schritt soll in Kapitel 8.4.3 das Transfer Learning eines bereits zuvor mit anderen Daten angeleerten Netzes vorgestellt werden. Es soll damit gezeigt werden, dass nicht immer eine derartig große Menge an Simulationsdaten notwendig ist, um eine automatische Plausibilitätsprüfung für FE-Simulationen umzusetzen.

### 8.4.1 Eigene Architektur mit Matrizenanordnung hintereinander

Die erste verwendete Architektur orientiert sich an dem Aufbau des VGG16-Netzes [229], gegenüber dem Aufbau des VGG16 Netzes gibt es jedoch einige Unterschiede. Zum einen wird eine Sequenz von Convolution-Schichten weniger benötigt, da die Größe einer einzigen Matrix anstelle von  $224 \times 224$  bei VGG16 hier nur  $100 \times 100$  bzw.  $112 \times 112$  beträgt und somit eine ausreichende Faltung bereits mit einer Convolution-Sequenz weniger erreicht werden kann. Zum anderen wird die Anzahl der Eingangskanäle von drei (RGB Bilder beim VGG16) auf 20 Eingangskanäle für die Matrizen einer FE-Simulation erhöht. Damit in den Convolution-Schichten keine Vermischung der 20 Eingangskanäle (jeweils eine Detektormatrix) miteinander stattfindet, werden «Grouped Convolution 2D Layer» mit jeweils 20

Gruppen anstelle von «Convolution 2D Layern» verwendet. Dies ermöglicht die Komprimierung jeder Eingangsmatrix ohne Vermischung mit anderen Kanälen/Eingangsmatrizen bis zum Erreichen der vollvernetzten Schichten des CNN.

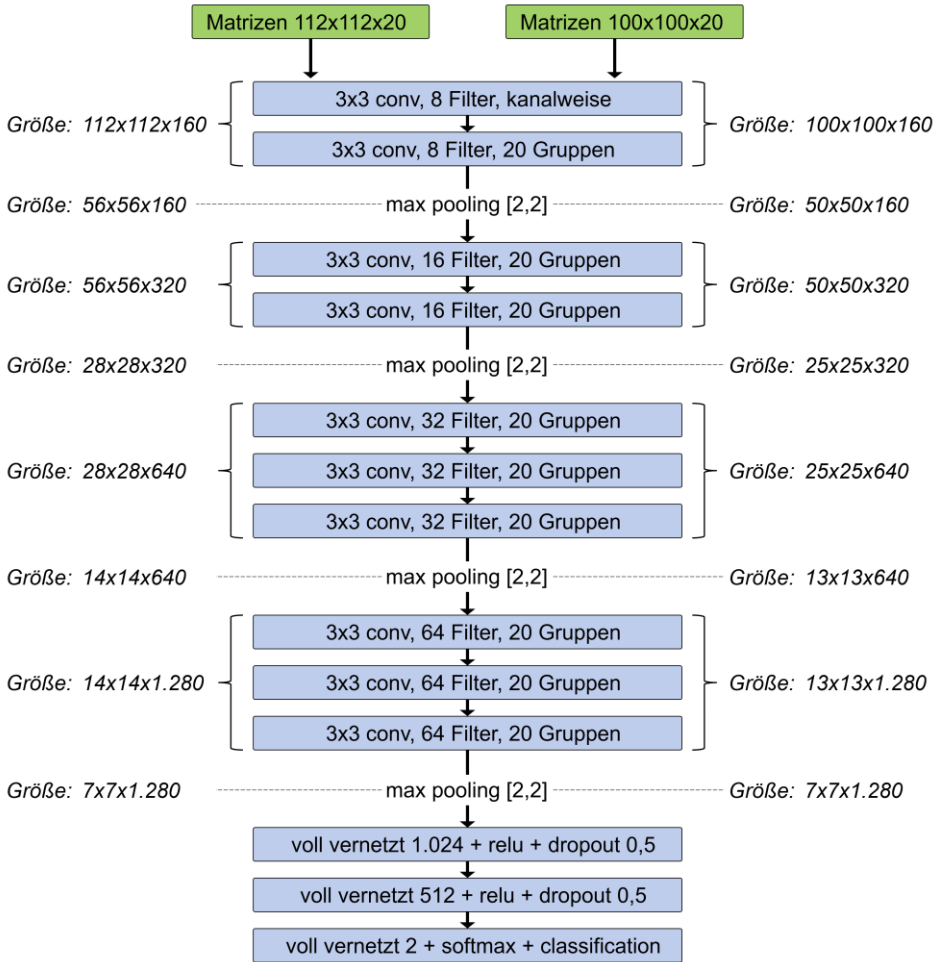


Bild 64: Aufbau der einzelnen Schichten der zu trainierenden CNNs mit der zugehörigen Größe der Bilder nach jeder Sequenz an Convolution-Schichten

Der Netzarchitektur werden die 20 Eingangsmatrizen hintereinander entweder als  $100 \times 100$  oder  $112 \times 112$  übergeben. In allen 10 Convolution-Schichten des Netzwerks werden Filter der Größe  $3 \times 3$  verwendet und jeder Filter immer um ein Pixel verschoben. Durch die Wahl von Filtern mit der

Schrittweite 1 und einem Padding am Außenrand von ebenfalls jeweils 1 bleibt die Größe der Matrizen für die nächste Schicht identisch. Eine Minimierung der ursprünglichen Größe von  $100 \times 100$  bzw.  $112 \times 112$  erfolgt ausschließlich durch die Max-Pooling-Schichten, bei denen immer  $2 \times 2$  Werte zu einem reduziert werden. Somit werden die Matrixdimensionen jeweils halbiert und die Matrizen schrumpfen auf 25% der ursprünglichen Größe. Der Aufbau dieses CNN ist in Bild 64 dargestellt.

In einer ersten Untersuchung soll der Einfluss von Matrizengröße und Normierungsstrategie untersucht werden. Es werden daher vollfaktoriell vier CNN mit nahezu identischen Einstellungen trainiert. Das Training erfolgt hierbei mittels GPU auf einer NVIDIA Tesla K40c. Die weiteren Einstellungen sind Minibatch-Größe (= 256 für  $100 \times 100$  Matrizen und = 200 für  $112 \times 112$  Matrizen), maximale Epochen (= 50) und eine initiale Lernrate (0,001) mit einer Verringerung um den multiplikativen Faktor von 0,1 nach jeweils 5 Epochen. Die Validierungsdaten werden verwendet, um ein mögliches Overfitting während des Trainings erkennen zu können, die Evaluierung der Prognosegüte erfolgt mittels der Testdatensätze. Die Testdatensätze werden nicht für das Training verwendet und bilden den potenziellen Ergebnisraum dank des D-optimalen Versuchsplans gut ab.

#### 8.4.2 Eigene Architektur mit Matrizenanordnung untereinander

Die zweite verwendete Art von CNN Architektur weist gegenüber den ersten vier Netzen einige Änderungen auf. Zum einen werden deutlich weniger Convolution-Schichten gewählt, allerdings werden größere Filter und von diesen eine höhere Anzahl verwendet. Darüber hinaus werden die Filter nach einem anderen Muster über die Matrizen verschoben, da eine Vermischung der Inhalte der 20 einzelnen Matrizen nicht gewünscht ist. Hierfür wird eine unterschiedliche Schrittweite (stride) für die Bewegung der Filter über die zwei Dimensionen der jeweiligen Matrix verwendet. In Bild 65 ist dies für die «DNA» einer FE-Simulation mit 20 untereinander angeordneten Matrizen und zwei unterschiedliche Schrittweiten dargestellt. Bei einer Filtergröße von 5 und einer Schrittweite von 5 Werten in Richtung  $a$  und 1 Wert in Richtung  $b$  kommt es zu keiner Vermischung der beiden Matrizen der Knoten und der gesperrten Verschiebung in  $x$ -Richtung ( $u_x$ ). Anders sieht es bei einer gewählten Schrittweite von 3 in Richtung  $a$  aus, hierbei kommt es zu einer Vermischung der beiden Matrizen. Um dies zu verhindern, werden für die zweite Art von Architektur Schrittweiten gewählt, welche ein ganzzahliger Teiler der Matrix- mit der Filtergröße sind.

Darüber hinaus werden für diese Architektur die Matrizen der Dimension 100x100 Pixel verwendet. Dies reduziert die Größe der übergebenen Matrizen um ca. 25 % ( $20 \cdot 100 \cdot 100 = 200.000$  gegenüber  $20 \cdot 112 \cdot 112 = 250.880$ ).

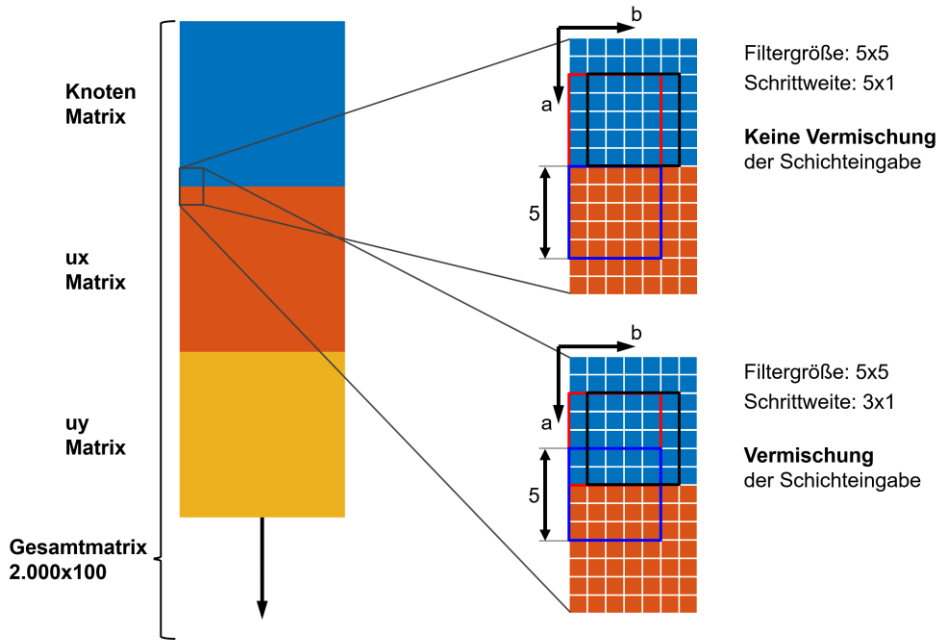


Bild 65: Erklärung zur Wahl einer geeigneten Schrittweite (Stride)

Der gewählte Netzaufbau ist in Bild 66 zu erkennen. Es wird stets immer nur eine Convolution-Schicht verwendet und es existieren keinerlei max pooling layer. Die Reduktion des Umfangs der ursprünglichen Matrix mit 2.000x100 erfolgt nur durch die Wahl der Filter und deren individuelle Schrittweite.

In den voll vernetzten Schichten wird entweder eine 250-100-2- oder eine 512-256-2-Neuronenkombination verwendet. Die weiteren Einstellungen sind:

- Minibatch-Größe (= 256)
- maximale Epochen (= 50)
- initiale Lernrate (0,001) mit einer Verringerung um den multiplikativen Faktor von 0,1 nach jeweils 5 Epochen

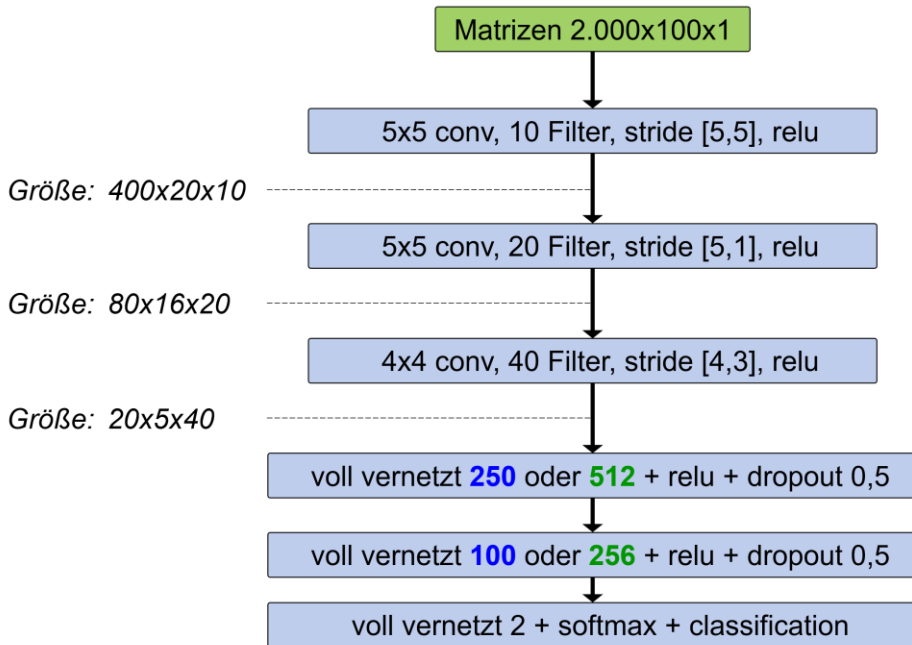


Bild 66: Aufbau der zweiten CNN-Art mit größeren Filtern und weniger Schichten

### 8.4.3 Transfer Learning

Neben der Umsetzung von neuen CNN-Architekturen soll auch der Ansatz des Transfer Learnings verwendet werden, um zu evaluieren, ob eine Reduktion des Datensatzes möglich ist und dennoch gute Prognosegüten des CNN erreicht werden können. In Voruntersuchungen (z. B. [S8]) konnte gezeigt werden, dass an einem kleineren Datensatz ein Transfer Learning mit dem AlexNet [47] gute Ergebnisse für die Plausibilitätsprüfung erzielen kann. Aus diesem Grund soll der Input der Matrizen für das AlexNet anders als bei den bisher beschriebenen CNN-Architekturen aufgebaut werden. Die Werte der 20 Matrizen jeder FE-Simulation werden in eine  $227 \times 227 \times 3$ -Eingabematrix umgewandelt. Dies liegt in der Definition der Eingangsgröße der Bilder für das AlexNet begründet und bedeutet, dass die Größe der Matrizen von  $100 \times 100$  auf  $87 \times 87$  reduziert werden muss. Aus diesem Grund werden Detektorflächen mit weniger Pixeln auf der Oberfläche verwendet und damit die  $87 \times 87$  Matrizen erzeugt. Die Anordnung der Werte erfolgt entlang der Ebenen des neuen Formats in Tiefenrichtung (vgl. Bild 67).

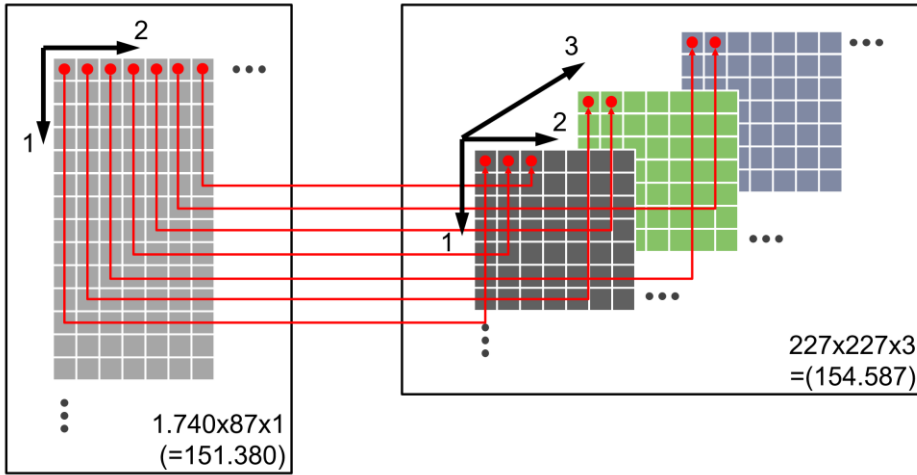


Bild 67: Umwandlung der 20 (8x8) Matrizen in die Eingabe für das AlexNet mit  $227 \times 227 \times 3$

Für das Transfer Learning der Plausibilitätsprüfung für FE-Simulationen werden alle vorderen Convolution-Schichten des trainierten AlexNet übernommen und nur die hinteren drei vollvernetzten Schichten angeleert. Es werden  $2.048-2.048-2$  Neuronen in diesen Schichten verwendet. Die weiteren Einstellungen sind Minibatch-Größe (= 200), maximale Epochen (= 50) und eine initiale Lernrate (0,001) mit einer Verringerung um den multiplikativen Faktor von 0,1 nach jeweils 5 Epochen.

Darüber hinaus wird die Menge, der für das Training zur Verfügung stehenden Daten schrittweise reduziert, bis nur noch 10% der ursprünglichen Datenmenge der Trainingsdaten verwendet werden:

- 100% der 31.320 Trainingsdatensätze
- 90% der 31.320 Trainingsdatensätze  $\Rightarrow$  28.188
- 80% der 31.320 Trainingsdatensätze  $\Rightarrow$  25.056
- 70% der 31.320 Trainingsdatensätze  $\Rightarrow$  21.924
- 60% der 31.320 Trainingsdatensätze  $\Rightarrow$  18.792
- 50% der 31.320 Trainingsdatensätze  $\Rightarrow$  15.660
- 40% der 31.320 Trainingsdatensätze  $\Rightarrow$  12.528
- 30% der 31.320 Trainingsdatensätze  $\Rightarrow$  9.396
- 20% der 31.320 Trainingsdatensätze  $\Rightarrow$  6.264
- 10% der 31.320 Trainingsdatensätze  $\Rightarrow$  3.132

Mit dieser Untersuchung soll gezeigt werden, wie viele Datensätze für das Anlernen eines CNN mittels Transfer Learning notwendig sind, bzw. ab welchem Prozentsatz an Trainingsdaten keine zufriedenstellenden Ergebnisse mehr erreicht werden können.

## 8.5 Ermittlung der Prognosegüte der CNNs

Jede der zuvor beschriebenen CNN-Architekturen wird mittels der zur Verfügung stehenden Trainingsdaten angelernet und während des Trainingsprozesses wird stets der Validierungsdatensatz herangezogen, um ein potenzielles Overfitting schnell zu erkennen. Die verschiedenen Umfänge des Testdatensatzes werden zur Evaluierung der Prognosegüte verwendet. Hierbei werden bei jedem Netz der gesamte Testdatensatz und die Datensätze der einzelnen Bauteile (Inliner-Schiene, MTB-Wippe und R6-Kurbelwelle) separat ausgewertet. Für die Auswertung wird jeweils die Confusion-Matrix mit den zugehörigen bzw. daraus abgeleiteten Größen herangezogen:

- True Positives (TP)
- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)
- Accuracy  $\frac{TP+TN}{TP+TN+FP+FN}$

## 8.6 Ergebnisse

Im Folgenden sollen die Ergebnisse der unterschiedlichen CNN-Architekturen in Kapitel o - 8.6.3 vorgestellt und zunächst allein für sich bewertet werden. Es handelt sich um die CNNs mit hintereinander angeordneten, untereinander angeordneten und die für das AlexNet speziell angeordneten Matrizen. Im Anschluss daran erfolgt in Kapitel 8.7 eine Untersuchung der Generalisierungsfähigkeit der trainierten CNN und in Kapitel o ein Vergleich der Ergebnisse untereinander auf Basis des Testdatensatzes.

### 8.6.1 Eigene Architektur mit Matrizenanordnung hintereinander

Zur Auswertung stehen die 4 CNNs mit einer Kombination aus Matrizengröße  $100 \times 100$  oder  $112 \times 112$  sowie entweder mittels Normierungsstrategie #1 oder Normierungsstrategie #2 normierten Daten zur Verfügung. Die Ergebnisse der Confusion-Matrizen und die abgeleitete Größe der Accuracy sind in Tabelle 20 für alle Bauteile sowohl einzeln als auch gesammelt und für alle trainierten CNNs aufgelistet. Bei der Betrachtung der Accuracy für alle Bauteile fällt auf, dass die Normierungsstrategie einen erheblichen Einfluss auf die Prognosegüte der Netze aufweist. Normierungsstrategie #2 ist stets deutlich besser als die beiden anderen Netze mit Normierungsstrategie #1. Mit Normierungsstrategie #2 erzielen die beiden Netze mit Matrizengröße  $100 \times 100$  und  $112 \times 112$  sehr gute Ergebnisse nahe einer Accuracy von 100 %. Durch die Erweiterung der Matrizen kann keine Verbesserung der Prognosegüte erreicht werden, die Erweiterung hat eher einen nachteiligen Effekt und senkt die Accuracy von 99,21 % auf 96,93 %. Dies ist insbesondere auf eine schlechtere Prognosegüte bei den Bauteilen Kurbelwelle und Inliner zurückzuführen. Die Accuracy der MTB-Wippe liegt in beiden Fällen bei 100 %.

Sollte eine weitere Senkung der falsch klassifizierten Bauteile gewünscht sein, müsste zunächst die Prognosegüte der Bauteile der Kurbelwelle weiter gesteigert werden, wobei hier insbesondere die Senkung der False Negatives zielführend erscheint. Für die Bewertung des Trainingserfolgs und die Anpassung der Gewichtungen wird bisher die Accuracy verwendet, anstelle der Accuracy ist eine adaptierte Kennzahl ebenso denkbar. Hierbei sollten insbesondere bei der Plausibilitätsprüfung fälschlicherweise plausibel klassifizierte Ergebnisse stärkeren Einfluss auf die Prognosegüte haben als die Anzahl der Ergebnisse mit fälschlicherweise nicht-plausibel klassifizierten Simulationen.

Es kann festgehalten werden, dass im Schnitt sehr gute Prognosegüten von allen CNNs erreicht werden. Normierungsstrategie #2 und die Matrizengröße  $100 \times 100$  liefern bessere Ergebnisse als Normierungsstrategie #1 oder die erweiterten Matrizen mit  $112 \times 112$  Pixeln.

Tabelle 20: Ermittelte Prognosegüten der CNNs mit hintereinander angeordneten Matrizen auf Basis der Testdaten: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) und Accuracy

		TP	TN	FP	FN	Accuracy
100x100 Norm. #1	Inliner	994	924	2	0	99,90 %
	MTB	950	2.275	975	0	76,79 %
	Kurbelwelle	669	1.353	45	93	93,61 %
	Alle Bauteile	2.613	4.552	1.022	93	86,53 %
100x100 Norm. #2	Inliner	992	919	4	5	99,53 %
	MTB	1.925	2.275	0	0	100 %
	Kurbelwelle	695	1.409	19	37	97,41 %
	Alle Bauteile	3.612	4.603	23	42	99,21 %
112x112 Norm. #1	Inliner	990	888	6	36	97,81 %
	MTB	950	2.275	975	0	76,79 %
	Kurbelwelle	693	1.392	21	54	96,53 %
	Alle Bauteile	2.633	4.555	1.002	90	86,81 %
112x112 Norm. #2	Inliner	963	887	33	37	96,35 %
	MTB	1.925	2.275	0	0	100 %
	Kurbelwelle	689	1.287	25	159	91,48 %
	Alle Bauteile	3.577	4.449	58	196	96,93 %

### 8.6.2 Eigene Architektur mit Matrizenanordnung untereinander

Zur Auswertung stehen die beiden anderen CNN-Architekturen mit unterschiedlicher Anzahl an Neuronen in den voll-vernetzten letzten Schichten zur Verfügung. Diese beiden Architekturen werden im Folgenden als CNN 2 und CNN 3 bezeichnet. CNN 2 ist mit deutlich weniger Neuronen (Konfiguration 250-100-2) versehen als CNN 3 (Konfiguration 512-256-2). Die Reduktion der Anzahl an Convolution Schichten hat einen geringen

Einfluss auf die Prognosegüte der beiden CNNs. Diese verfügen über Accuracies von 98,71 % bzw. 98,66 % bezogen auf den gesamten Testdatensatz mit allen Bauteilen. Bei der Betrachtung der Accuracies der einzelnen Bauteile in Tabelle 21 ist erkennbar, dass eine höhere Anzahl von Neuronen in den voll-vernetzten Schichten zu höheren Prognosegüten der Bauteile Inliner-Schiene und MTB-Wippe führt, allerdings geht dies zu Lasten der Accuracy des Bauteils Kurbelwelle. Dies führt bei der Untersuchung der Accuracy mit allen Bauteilen des Testdatensatzes zu einem geringfügig schlechteren Ergebnis als bei CNN 2. Da die Accuracies aber sehr nahe bei 100 % und sehr nahe beieinander liegen, können beide Netze als sehr gut bewertet werden.

Tabelle 21: Ermittelte Prognosegüten der CNNs mit untereinander angeordneten Matrizen auf Basis der Testdaten: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) und Accuracy

		TP	TN	FP	FN	Accuracy
2.000x100 CNN 2	Inliner	976	922	20	2	98,85 %
	MTB	1.925	2.274	0	1	99,98 %
	Kurbelwelle	681	1.395	33	51	96,11 %
	Alle Bauteile	3.582	4.591	53	54	98,71 %
2.000x100 CNN 3	Inliner	992	924	4	0	99,79 %
	MTB	1.925	2.275	0	0	100 %
	Kurbelwelle	668	1.385	46	61	95,05 %
	Alle Bauteile	3.585	4.584	50	61	98,66 %

### 8.6.3 Transfer Learning mittels AlexNet Architektur

Zur Auswertung stehen 10 trainierte CNNs, welche mittels unterschiedlicher Umfänge an Trainingsdaten durch Transfer Learning und der AlexNet Architektur trainiert wurden. Der Umfang der Trainingsdaten nimmt sukzessive von 100 % auf 10 % ab (in 10 % - Schritten). Für die Ermittlung der Confusion Matrizen und der Accuracy in Tabelle 22 und Tabelle 23 wird der identische Testdatensatz wie auch bei der Auswertung der anderen CNNs in Kapitel 8.6.1 und 8.6.2 verwendet.

Tabelle 22: Ermittelte Prognosegüten auf Basis der Testdaten der mittels Transfer Learning trainierten CNNs: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) und Accuracy – Teil 1 von 2

		<b>TP</b>	<b>TN</b>	<b>FP</b>	<b>FN</b>	<b>Accuracy</b>
100 % Trainingsdaten	Inliner	771	923	225	1	88,23 %
	MTB	1.924	2.275	1	0	99,98 %
	Kurbelwelle	540	1.384	174	62	89,07 %
	Alle Bauteile	3.235	4.582	400	63	94,41 %
90 % Trainingsdaten	Inliner	768	924	228	0	88,13 %
	MTB	1.925	2.275	0	0	100 %
	Kurbelwelle	535	1.420	179	26	90,51 %
	Alle Bauteile	3.228	4.619	407	26	94,77 %
80 % Trainingsdaten	Inliner	769	924	227	0	88,18 %
	MTB	1.924	2.275	1	0	99,98 %
	Kurbelwelle	355	1.446	359	0	83,38 %
	Alle Bauteile	3.048	4.645	587	0	92,91 %
70 % Trainingsdaten	Inliner	773	907	223	17	87,50 %
	MTB	1.851	2.265	74	10	98,00 %
	Kurbelwelle	357	1.443	357	3	83,33 %
	Alle Bauteile	2.981	4.615	654	30	91,74 %
60 % Trainingsdaten	Inliner	770	919	226	5	87,97 %
	MTB	1.765	2.268	160	7	96,02 %
	Kurbelwelle	329	1.446	385	0	82,18 %
	Alle Bauteile	2.864	4.633	771	12	90,54 %

Tabelle 23: Ermittelte Prognosegüten auf Basis der Testdaten der mittels Transfer Learning trainierten CNNs: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) und Accuracy – Teil 2 von 2

		<b>TP</b>	<b>TN</b>	<b>FP</b>	<b>FN</b>	<b>Accuracy</b>
50 % Trainingsdaten	Inliner	768	924	228	0	88,13 %
	MTB	1.761	2.261	164	14	95,76 %
	Kurbelwelle	330	1.446	384	0	82,22 %
	Alle Bauteile	2.859	4.631	776	14	90,46 %
40 % Trainingsdaten	Inliner	768	924	228	0	88,13 %
	MTB	1.773	2.247	152	28	95,71 %
	Kurbelwelle	319	1.446	395	0	81,71 %
	Alle Bauteile	2.860	4.617	775	28	90,30 %
30 % Trainingsdaten	Inliner	768	924	228	0	88,13 %
	MTB	1.773	2.205	152	70	94,71 %
	Kurbelwelle	327	1.446	387	0	82,08 %
	Alle Bauteile	2.868	4.575	767	70	89,89 %
20 % Trainingsdaten	Inliner	727	902	269	22	84,84 %
	MTB	1.050	2.244	875	31	78,43 %
	Kurbelwelle	238	1.446	476	0	77,96 %
	Alle Bauteile	2.015	4.592	1.620	53	79,79 %
10 % Trainingsdaten	Inliner	0	924	996	0	48,13 %
	MTB	0	2.275	1.925	0	54,17 %
	Kurbelwelle	0	1.446	714	0	66,94 %
	Alle Bauteile	0	4.645	3.635	0	56,10 %

Es ist erkennbar, dass die Accuracies der Einzelteile und des Gesamttestdatensatzes mit allen Bauteilen schrittweise mit Reduzierung des Trainingsdatensatzes abnehmen. Der reduzierte Datensatz mit 40 % der Trainingsdaten liefert immer noch Accuracies von über 90 %.

Ein extremer Abfall der Accuracy ist erst bei 10 % des Testdatensatzes erkennbar, dieses Netz erkennt keinen Zusammenhang mehr in den Daten, sondern klassifiziert alle Bauteile als Negatives (hier plausible Simulationen). Dies ist an der Anzahl von 0 Bauteilen in den beiden Klassen TP und FN erkennbar. Die Accuracy von 56,10 % entspricht damit exakt der Konstellation des Testdatensatzes mit 4.645 plausiblen und 3.635 nicht-plausiblen Simulationen.

Es ist möglich, mit wenig Aufwand und wenigen notwendigen Anpassungen ein bestehendes vortrainiertes CNN anzupassen und mit einem neuen Datensatz nachzutrainieren. Erwartungsgemäß sinkt die Prognosegüte des Netzes bei einer Reduktion des Trainingsdatensatzes, allerdings liegen die Ergebnisse selbst bei einem auf 60 % reduzierten Trainingsdatensatz noch bei deutlich über 90% Accuracy. Gerade für den Fall, dass weniger Daten vorhanden sind, ist das Transfer Learning eine sinnvolle und gut funktionierende Lösung.

## 8.7 Generalisierungsfähigkeit der trainierten CNNs

Die Klassifikation der Bauteile des Testdatensatzes ist mit hoher Prognosegüte möglich. Allerdings handelt es sich hierbei um Bauteile, welche nicht identisch aber relativ ähnlich zu den Bauteilen des Trainingsdatensatzes sind. Im Folgenden soll die Fähigkeit der trainierten Netze zur Generalisierung bzw. Extrapolation auf andere Bauteilarten untersucht werden. Hierfür werden vier unterschiedliche Arten von Testbauteilen mittels mehrerer FE-Simulationen betrachtet. Es handelt sich um grundlegend andere Bauteile (Profilverbinder, Profilverbinder 80x80 und Hebel eines Ausrücklagers) sowie um eine Kurbelwelle eines 4-Zylinder Reihenmotors. In Bild 68 sind die vier Bauteilarten abgebildet.

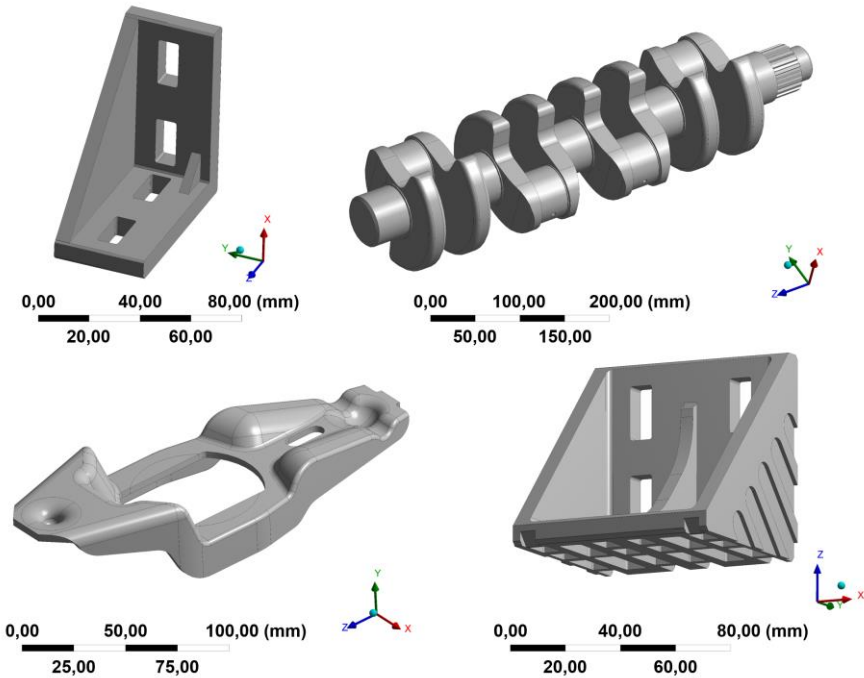


Bild 68: Arten von Testbauteilen: Profilverbinder einfach, Kurbelwelle R4, Ausrücklagerhebel und Profilverbinder 80 mm x 80 mm

Analog zur bisherigen Vorgehensweise werden sowohl «plausible», als auch «nicht plausible» Simulationen erstellt. Die Zusammensetzung aus plausiblen (29) und nicht plausiblen Simulationen (39) beträgt 42,65 % zu 57,35 %. Die variablen Parameter der Simulationen sind im Folgenden aufgelistet, in Klammern ist jeweils die Anzahl an Simulationen dieses Bauteils angegeben:

- **Profilverbinder einfach** (20 Simulationen, nicht-plausible Faktorstufen sind farblich hervorgehoben):

		Faktorstufen Nr. (vollfaktoriell)				
		1	2	3	4	5
Elementgröße		1 mm	1,5 mm	2 mm	5 mm	10 mm
Kraft	X	65 N	400 N	650 N	6.500 N	-
	Y	-60 N	-360 N	-600 N	-6.000 N	-
	Z	45 N	280 N	450 N	4.500 N	-

- **Profilverbinder 80x80**

(20 Simulationen, nicht-plausible Faktorstufen sind farblich hervorgehoben):

		Faktorstufen Nr. (vollfaktoriell)				
		1	2	3	4	5
Elementgröße		1 mm	1,5 mm	2 mm	5 mm	10 mm
Kraft	X	-750 N	-7.500 N	-75.000 N	-75.000 N	-
	Y	-300 N	-3.000 N	-3.000 N	-30.000 N	-
	Z	-500 N	-5.000 N	-5.000 N	-50.000 N	-

- **Blechhebel zur Montage eines Ausrücklagers einer Kupplung**

(24 Simulationen, nicht-plausible Faktorstufen sind farblich hervorgehoben):

		Faktorstufen Nr. (vollfaktoriell)					
		1	2	3	4	5	6
Elementgröße		0,75 mm	1 mm	1,5 mm	2 mm	5 mm	10 mm
Kraft		-150 N	-300 N	-450 N	-900 N	-	-

- **R4-Kurbelwelle**

(4 Simulationen, nicht-plausible Faktorstufen sind farblich hervorgehoben):

	Nr. der Simulation			
	1	2	3	4
Elementgröße	3 mm	3,5 mm	4 mm	10 mm
Kraft der Ausgleichsmasse	65.000 N	55.592 N	44.192 N	125.650 N
Zünddruck	180 bar	190 bar	1.900 bar	190 bar
max. Drehzahl	4.500 min <sup>-1</sup>	4.500 min <sup>-1</sup>	4.500 min <sup>-1</sup>	6.500 min <sup>-1</sup>
Kolben-Ø	80 mm	89 mm	80 mm	84 mm
Kurbelradius	48 mm	49 mm	51 mm	51 mm
Hauptlager-Ø	65 mm	60 mm	50 mm	60 mm

Alle vorgestellten Testbauteile zur Überprüfung der Generalisierbarkeit und deren Simulationen werden, analog zum zuvor beschriebenen Vorgehen, in Detektormatrizen umgewandelt und diese entsprechend normiert. Hierfür werden die  $R_m$ -Werte (Zugfestigkeit) der einzelnen Bauteile benötigt: Profilverbinder einfach und Profilverbinder 80x80 ( $R_m = 310 \text{ MPa}$ ), Ausrücklagerhebel ( $R_m = 460 \text{ MPa}$ ) und Kurbelwelle R4 ( $R_m = 1000 \text{ MPa}$ ). Die normierten Matrizen werden ebenfalls allen zuvor trainierten CNNs übergeben und die Prognosegüte entsprechend ausgewertet (vgl. Tabelle 24).

Tabelle 24: Ermittelte Prognosegüten der mittels Transfer Learning trainierten CNNs für die zusätzlichen Bauteile zur Überprüfung der Generalisierung: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) und Accuracy

CNN	TP	TN	FP	FN	Accuracy
100x100 Norm. #1	31	8	8	21	57,35 %
100x100 Norm. #2	32	15	7	14	69,12 %
112x112 Norm. #1	39	0	0	29	57,35 %
112x112 Norm. #2	19	13	20	16	47,06 %
2.000x100 CNN 2	32	7	7	22	57,35 %
2.000x100 CNN 3	29	16	10	13	66,18 %
Transfer 100% Train. Daten	27	3	12	26	44,12 %
Transfer 90% Train. Daten	28	2	11	27	44,12 %
Transfer 80% Train. Daten	27	5	12	24	47,06 %
Transfer 70% Train. Daten	27	6	12	23	48,53 %
Transfer 60% Train. Daten	26	11	13	18	54,40 %
Transfer 50% Train. Daten	26	10	13	19	52,94 %
Transfer 40% Train. Daten	25	17	14	12	61,76 %
Transfer 30% Train. Daten	26	7	13	22	48,35 %
Transfer 20% Train. Daten	26	17	13	12	63,24 %
Transfer 10% Train. Daten	0	29	39	0	42,65 %

Es ist erkennbar, dass die guten Prognosegüten nahe 100 % Accuracy nicht erreicht werden können. Mit dem Hintergrund, dass eine Accuracy von 57,35 % bereits durch reines Raten erreicht werden kann, zeigen viele der trainierten CNNs schlechte Ergebnisse. Es gibt Ausnahmen, wie beispielsweise das CNN  $100 \times 100$  mit Normierungsstrategie #2 oder das CNN  $2.000 \times 100$ . Generell ist festzuhalten, dass die trainierten CNNs nicht im Stande sind, das Wissen aus den Trainingssimulationen auf die zusätzlichen und völlig unbekanntem Bauteile des Generalisierungstest zu übertragen. Allerdings ist der Trend erkennbar, dass einige Fehler in den unbekanntem Simulationen bereits erkannt werden, jedoch noch zu schwach, um mit diesen Netzen direkt in eine Anwendung mit vollständig neuen Bauteilen zu gehen.

Für die beiden CNNs mit höherer Prognosegüte sind die Einträge der Confusion Matrix bauteilweise in Tabelle 25 dargestellt. Es werden folgende Abkürzungen für die Bauteilarten verwendet: Profilverbinder einfach (B<sub>1</sub>), Profilverbinder  $80 \times 80$  (B<sub>2</sub>), Ausrücklagerhebel (B<sub>3</sub>) und die Vierzylinder-Kurbelwelle (R<sub>4</sub>).

Tabelle 25: Prognosegüten auf Basis der zusätzlichen Testdaten zur Ermittlung der Generalisierungsfähigkeit: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) und Accuracy

		TP	TN	FP	FN	Accuracy
$100 \times 100$ Norm. #2	B <sub>1</sub>	11	0	0	9	55,00 %
	B <sub>2</sub>	12	3	2	3	75,00 %
	B <sub>3</sub>	7	12	5	0	79,10 %
	R <sub>4</sub>	2	0	0	2	50,00 %
	Alle Bauteile	32	15	7	14	69,12 %
$2.000 \times 100$ CNN 3	B <sub>1</sub>	6	9	5	0	75,00 %
	B <sub>2</sub>	10	6	4	0	80,00 %
	B <sub>3</sub>	12	0	0	12	50,00 %
	R <sub>4</sub>	1	1	1	1	50,00 %
	Alle Bauteile	29	16	10	13	66,18 %

## 8.8 Vergleich und Bewertung der trainierten CNNs

Im Folgenden soll eine Bewertung aller vorgestellten Ergebnisse aus Kapitel 8.6.1, 8.6.2, 8.6.3 und 8.7 durch einen gegenseitigen Vergleich stattfinden. Der Vergleich soll anhand des Kennwerts der Accuracy erfolgen, weitere Kenngrößen aus der Confusion-Matrix oder eigene Kennwerte werden an entsprechenden Stellen herangezogen.

Das Training aller CNN-Architekturen erfolgte mittels GPU-Berechnung auf einer Tesla K40c. Durch die parallele Auswertung mit einem Validierungsdatensatz und deren Ergebnissen während des Trainingsprozesses (vgl. Bild 69 bis Bild 84 im Anhang) könnte ein Overfitting sicher erkannt werden und wird aus diesem Grund ausgeschlossen. Dies bestätigt sich auch bei der Evaluierung der CNNs mittels der Testdatensätze. Hierbei zeigt sich, dass nahezu alle Architekturen in der Lage sind, den Zusammenhang der Trainingsdaten abzubilden und diesen auf die für das Netz unbekannt Bauteile des Testdatensatzes anzuwenden. Mittels der vorgestellten Normierungsstrategie #2 können bessere Prognosegüten nahe 100 % Accuracy erreicht werden. Durch die geringere Menge an zu trainierenden Filtern und Gewichtungen gegenüber den CNNs mit adaptierter VGG16-Architektur stellt das CNN 2 eine sehr gute Alternative dar. Erst bei der Umsetzung des Transfer Learnings und einer Reduktion der Trainingsdaten auf unter 40 % im Vergleich zu den Trainingsdaten der anderen Netze lassen sich niedrigere Accuracy-Werte feststellen.

Es kann festgehalten werden, dass die Methode der sphärischen Detektorflächen mit einer geeigneten Normalisierung der Detektormatrizen zur Durchführung einer Plausibilitätsprüfung mit hoher Genauigkeit von ähnlichen Simulationen verwendet werden kann. Eine derartige Prüfung muss allerdings nicht für jede Bauteilart separat durchgeführt werden, sondern es kann ein CNN trainiert werden, das die Zusammenhänge von unterschiedlichen Bauteilarten beinhaltet, beispielsweise für alle FE-Simulationen, die in einer Abteilung oder Firma durchgeführt werden. Dies stellt eine erhebliche Neuerung dar und unterstreicht die Neuheit der entwickelten Methode. Es ist somit möglich, bisher ungenutzte Simulationsdatenbestände für das Digital Engineering nutzbar zu machen. In diesem Zusammenhang sei kurz auf die Fehlerraten bei der Bildklassifikation von CNN und menschlichen Personen hingewiesen. Im Jahr 2015 waren CNNs bereits in der Lage, geringere Fehlerraten für die Testdatensätze bei der Klassifikation von Bildern des ImageNet Datensatzes zu erreichen (Microsoft PReLU 4,94 % [230], Google BN-x30 Ensemble mit Batch Normalisierung 4,82 %

[231], Baidu brute-force Ansatz 4,58 % [232], menschliche Fehlklassifikationsrate 5,1 % [233].

Die Überprüfung der Generalisierungsfähigkeit der trainierten CNNs hat gezeigt, dass es nicht möglich ist, mittels den drei Bauteilarten Inliner Schiene, MTB Wippe und R6-Kurbelwelle auf andersartige Bauteile zu schließen. Die Vermutung liegt nahe, dass erheblich mehr unterschiedliche Bauteile und deren Simulationen angelernt werden müssen, bevor ein CNN in der Lage ist, die gelernten Assoziationen auf beliebige Bauteile zu übertragen. Die vorgestellte Methodik liefert jedoch die Werkzeuge, die notwendig sind, um den Ansatz der Plausibilitätsprüfung entsprechend zu skalieren. Mit einer entsprechend umfangreichen Datenbasis können dann die weiteren Schritte unternommen werden. Durch die rasante Entwicklung bei Hard- und Softwareentwicklung werden zukünftig immer mächtigere Werkzeuge für die Produktentwicklung zur Verfügung stehen. Die Herausforderung vor der sowohl Industrie, als auch die akademische Wissenschaft stehen, ist die effiziente Bereitstellung von umfangreichen und sinnvoll aufgebauten Datensätzen.



## 9 Zusammenfassung und Ausblick

Die durchgängige Nutzung von Daten und die Nutzung der in diesen Daten versteckten Potentiale sind gegenwärtig die größten Herausforderungen in diversen Sektoren. Dies beginnt im sozialen Sektor und zieht sich quer durch viele andere Bereiche, wie etwa dem öffentlichen und privaten Nahverkehr, der industriellen Produktion und natürlich auch alle Bereiche der Produktentwicklung und Produktentstehung. In der vorliegenden Arbeit wurde aufgezeigt, dass im Bereich des Digital Engineering erhebliches Potential liegt, sofern geometriebasierte Daten für Algorithmen des Maschinellen Lernen nutzbar gemacht werden können. Unter Digital Engineering wird die Erweiterung der virtuellen Produkt- und Prozessentwicklung mit datengestützten Modellen verstanden, wobei der Nutzung von bestehenden Datenbeständen ein erheblicher Beitrag zukommt.

Anhand der Betrachtung des aktuellen Standes der Wissenschaft und Forschung konnte gezeigt werden, dass die Übergabe von Geometriedaten unterschiedlicher Bauteile an Algorithmen des Maschinellen Lernens ein bisher nur wenig beleuchtetes Thema darstellt. Insbesondere die Nutzung von industriellen Bauteilen (CAD-Bauteile, Scans von realen Bauteilen) oder die Weitergabe von umfangreichen Simulationsdaten unterschiedlicher Finite-Elemente-Analysen findet bisher keine Berücksichtigung. Dies liegt auch in der fehlenden Verfügbarkeit derartiger Daten und geeigneter Ansätze begründet. Nichtsdestotrotz existieren aber sehr umfangreiche Datenmengen derartiger Daten in den Langzeitspeichern von diversen Unternehmen aus dem industriellen Umfeld und die Nutzung dieser Datenbestände ist sowohl aus akademischer wie auch aus wirtschaftlicher Sicht sehr interessant und vielversprechend.

Aus diesem Grund wurde eine Methode entwickelt, mit deren Hilfe eine Übergabe geometriebasierter Größen an Algorithmen des Maschinellen Lernen ermöglicht wird. Der Ausgangspunkt dieser Methode sind dreidimensionale Punktwolken, wie sie in diversen Bereichen entstehen (FE-Vernetzung, Scan-Verfahren, etc.). Ebenso können Punktwolken aus detaillierteren Geometriemodellen (z. B. CAD-Modelle) abgeleitet werden. Die Punkte einer Punktwolke werden auf eine sphärische Oberfläche projiziert, welche mit einer bestimmten Anzahl an Detektorbereichen (Pixel) versehen ist. Die Anzahl der projizierten Punkte in jedem dieser Pixel oder mit diesen Punkten assoziierte Größen können in jedem Detektorpixel der sphärischen Oberfläche aufsummiert werden. Durch eine Umwandlung

der kartesischen Koordinaten in Kugelkoordinaten ist es möglich, beliebig große oder kleine Bauteile auf eine fix vorgegebene Anzahl an Kennwerten zu reduzieren. Somit steht eine Methode zur Dimensionsreduktion für aus Geometrien abgeleitete Punktwolken bzw. mit den Punkten verknüpften Kennwerten zur Verfügung.

Anhand von drei unterschiedlichen Anwendungsbeispielen aus dem Bereich der Produktentstehung (Bauteilerkennung von CAD-Bauteilen, Erkennung von Montagefehlern bei Kabelbäumen und der Plausibilitätsprüfung von Finite-Elemente-Simulationen) wurde die Umsetzbarkeit der entwickelten Methode verdeutlicht. Hierfür wurde eine Methode zur einheitlichen Orientierung von dreidimensionalen Punktwolken für volumenvernetzte CAD-Bauteile entwickelt und umgesetzt. Als prominente Vertreter von Algorithmen des Maschinellen Lernen wurden Künstliche Neuronale Netze (Shallow Neural Networks), Convolutional Neural Networks (tiefe Neuronale Netze aus dem Bereich des Deep Learnings) und statistische Kennziffern bzw. Bildvergleichsmetriken verwendet.

Anhand der Anwendungen konnten die eingangs bei der Ableitung des Handlungsbedarfs aufgezeigten Bedarfe, Berücksichtigung finden. Dies waren:

- **Vorverarbeitungsschritt für die Übergabe geometriebasierter Daten an Algorithmen des Maschinellen Lernen:**

Durch die Entwicklung der Methode der sphärischen Detektorflächen, inklusive einer Methodik zur Herbeiführung einer einheitlichen Orientierung können geometriebasierte Daten an Algorithmen des Maschinellen Lernens übergeben werden. Die entwickelte Methodik ermöglicht es direkt, vernetzte CAD-Daten, Scan-Daten oder Netze von Finite-Elemente-Simulationen in eine einheitliche Form zu überführen. Hierbei ist es vollständig unerheblich, wie groß die zu berücksichtigenden Bauteile sind oder ob sowohl kleine als auch große Bauteile gleichzeitig berücksichtigt werden sollen. Darüber hinaus wurde eine Methodik entwickelt, welche es ermöglicht einheitliche Orientierungen von individuell im dreidimensionalen Raum angeordneten Bauteilen herbeizuführen. Die Ableitung der sogenannten Detektormatrix ermöglicht die Dimensionsreduktion von dreidimensionalen Punktwolken auf eine zweidimensionale Matrix fixer Größe. Dadurch können unterschiedliche Geometrien einheitlich an Algorithmen des Maschinellen Lernens übergeben werden. Somit steht eine Methodik zur Verfügung um die erste Forschungsfrage aus Kapitel 3 zu beantworten:

*Wie kann die Übergabe von Geometrieinformationen und mit der Geometrie verknüpften Kenngrößen an Algorithmen des Maschinellen Lernens erfolgen?*

▪ **Darlegung des entwickelten Vorgehens an drei Anwendungsbeispielen:**

Das entwickelte Vorgehen konnte im Rahmen einer Bauteilerkennung von CAD-Bauteilen, der Verarbeitung von Scan-Daten von optischen Vermessungssystemen und im Rahmen einer Plausibilitätsprüfung für strukturmechanische Finite-Elemente-Simulationen aufgezeigt werden. Hierdurch konnte die Vielfältigkeit der Anwendungsfelder der entwickelten Methodik exemplarisch aufgezeigt werden. In allen Anwendungsbeispielen zeigten sich sehr gute Ergebnisse und die Methodik ermöglicht die Nutzung von bisher ungenutzten Datenbeständen der virtuellen Produktentwicklung für zukünftige Entwicklungen. Die Methodik stellt somit einen Baustein des Digital Engineering dar, bei der der Nutzung von Daten ein vermehrtes Interesse zuteilwird. Dies beantwortet die Fragen, *welche Schritte notwendig sind, um eine Prozesskette von der Datenerstellung bis hin zur Datenanalyse aufzubauen, und wo die potenziellen Schwierigkeiten darin liegen?*

▪ **Nutzbarmachung von Datenbeständen des Produktentstehungsprozesses für den Entwicklungsprozess neuer Produkte:**

Durch die Übergabe geometriebasierter Daten an Algorithmen des Maschinellen Lernens können Datenbestände deutlich einfacher nutzbar gemacht werden. Beispielsweise lassen sich mittels der Bauteilerkennung alte CAD-Daten nach identischen Bauteilen durchsuchen oder es kann ein Ähnlichkeitswert eines Bauteils bezüglich anderer Bauteile ermittelt werden. Dies erlaubt eine bessere Nutzung von historischen Datenbeständen und führt zur Vermeidung der Entwicklung sehr ähnlicher Bauteile durch verschiedene Personen zu verschiedenen Zeitpunkten. Durch die «Simulations-DNA» steht eine Möglichkeit zur Verfügung, um Simulationen automatisiert zu bewerten bzw. kann anhand der Bildwerte eine Ähnlichkeitskennzahl für FE-Simulationen ermittelt werden. Die weiteren Anwendungsfelder der Methode sind sehr vielseitig und können durch entsprechende Experten auch an ganz anderen Stellen im Bereich der Produktentstehung oder darüber hinaus identifiziert werden. Dies ist die Antwort auf die Frage, *welche*

*Adaptionen an der Methodik notwendig sind, um unterschiedliche Anwendungsfälle und damit unterschiedliche Datenbestände zu verarbeiten.*

- **Aufzeigen der relevanten Aspekte und Teilschritte für die Umsetzbarkeit der entwickelten Methodik durch Dritte:**

Im Rahmen der vorliegenden Arbeit wurden alle relevanten Teilschritte und das individuelle Vorgehen zur Umsetzung anhand der drei Anwendungsbeispiele sehr detailliert beleuchtet. Dies ermöglicht die weitere Umsetzung der entwickelten Methode sowohl im akademischen als auch im industriellen Umfeld. Insbesondere für Personen mit weniger Erfahrung im Bereich der datengetriebenen Produktentwicklung und des Maschinellen Lernens soll die Arbeit einen Denkanstoß liefern und die Potentiale, die in den Daten stecken, verdeutlichen. Durch die detaillierte Beschreibung des Vorgehens lassen sich in der Produktentwicklung tätige Personen animieren, über den Tellerrand hinauszublicken. Durch neue Methoden des Digital Engineering können bestehende Prozesse noch weiter verbessert werden und es erlaubt die Erschließung neuer Geschäftsfelder durch die durchgängige Datennutzung, insbesondere auch im Bereich der Produktentwicklung und -entstehung. Somit ist ersichtlich, *welche Schritte notwendig sind, um die entwickelte Methodik sowohl in der Industrie als auch im akademischen Bereich einzusetzen.*

Weiterentwicklungsmöglichkeiten der Methodik sind vielfältig. Aus Sicht des Digital Engineering mit Fokus auf die Produktentwicklung sind die folgenden Punkte besonders relevant:

Aktuell erfolgt die Projektion der Punktwolke auf eine Detektorfläche mit Pixeln, welche entweder durch einen einheitlichen Winkel oder mittels einheitlicher Fläche jedes Detektorpixels definiert werden. Anstelle dieser beiden Verfahren können weitere Varianten entwickelt werden (z. B. in Abhängigkeit der Vernetzungsdichte des verwendeten FE-Netzes, etc.).

Der Umfang der verwendeten Daten für das Training kann noch erheblich erweitert werden, darüber hinaus kann die Methode zur Projektion von FE-Daten auf Baugruppen erweitert werden. Hierfür müssen zusätzliche Matrizen für die Kontakte zwischen den unterschiedlichen Bauteilen und eine entsprechende Normierung dieser Kontaktmatrizen entwickelt werden. Gleiches gilt auch für zusätzliche Ergebnisgrößen aus diesen FE-Simulationen.

Es kann ebenfalls eine Evaluierung erfolgen, inwieweit die vorliegende Methodik als automatischer Absicherungsschritt in bestehenden Prozessabsicherungsketten integriert werden kann oder welche Änderungen vorgenommen werden müssen, um dies zu erreichen. Hierbei kann es hilfreich sein, einen Algorithmus des Maschinellen Lernens zu verwenden, bei dem Menschen die Entscheidung des Algorithmus besser nachvollziehen können als dies bei Künstlichen Neuronalen Netzen zum gegenwärtigen Zeitpunkt der Fall ist.

Durch die aktuell rasante Entwicklung potenter GPUs und die damit verbundenen Möglichkeiten zur Datenanalyse durch vielfältige Software Frameworks können in der nahen Zukunft noch deutlich schwierigere Probleme gelöst werden. Hierfür ist jedoch eine Verfügbarkeit umfangreicher Datenbestände erforderlich. Aus diesem Grund appelliert der Autor an alle Institutionen, relevante Daten strukturiert zu speichern. Dies gilt auch für Daten, welche zum gegenwärtigen Zeitpunkt noch nicht genutzt werden können. Insbesondere dann, wenn sich mit diesen Daten bzw. einer sinnvollen Auswertung der Daten ein Einspar- oder Optimierungspotential absehen lässt. So können die entwickelten Detektormatrizen auch als Feature für deutlich umfangreichere Auswertungen eingesetzt werden, vergleichbar mit anderen statistischen Kenngrößen wie dem Mittelwert oder der Standardabweichung.



## 10 Summary and outlook

The end-to-end use of data and the exploitation of the potential hidden in this data are currently the greatest challenges in various sectors. This starts in the social sector and extends to many other areas, such as public and private transport, industrial production and, of course, all areas of product development and product creation. In this thesis, it has been shown that there is considerable potential in the area of digital engineering, provided that geometry-based data can be made usable for machine learning algorithms. Digital engineering is understood as the extension of virtual product and process development with data-supported models, whereby the use of existing data sets makes a significant contribution.

By examining the current state of science and research, it could be shown that the transfer of geometry data of different components to machine learning algorithms is a topic that has been little explored so far. In particular, the use of industrial components (CAD components, scans of real components) or the transfer of extensive simulation data of different finite element analyses has not been considered so far. This is also due to the lack of availability of such data. Nevertheless, extremely large amounts of such data exist in the long-term memories of various companies in the industrial environment and the use of this data is very interesting and promising from both an academic and an economic point of view.

For this reason, a method has been developed to enable the transfer of geometry-based data to machine learning algorithms. The starting point of this method are three-dimensional point clouds as they are generated in various fields (finite element simulation meshing, scanning methods, etc.). Point clouds can also be derived from more detailed geometry models (e. g. CAD models). The points of a point cloud are projected onto a spherical surface, which is provided with a certain number of detector areas (pixels). The number of projected points in each of these pixels or quantities associated with these points can be summed up in each detector pixel of the spherical surface. By converting the Cartesian coordinates into spherical coordinates, it is possible to reduce arbitrarily large or small components to a fixed number of characteristic values. Thus, a method for dimension reduction is available for point clouds derived from geometries or characteristic values linked to these points.

The feasibility of the developed method was illustrated by means of three different application examples from the field of product development (part recognition of CAD parts, recognition of assembly errors in cable harnesses and the plausibility check of finite element simulations). For this purpose, a method for the uniform orientation of three-dimensional point clouds for volume meshed CAD parts was developed and implemented. Artificial neural networks (Shallow Neural Networks), Convolutional Neural Networks (deep neural networks from the field of Deep Learning) and statistical indices or image comparison metrics were used as prominent representatives of machine learning algorithms.

Based on the applications, the needs identified at the beginning of this thesis could be considered. These were:

- **Preprocessing step for the transfer of geometry-based data to machine learning algorithms:**

By the developed method of spherical detector surfaces, including a methodology for inducing a uniform orientation, geometry-based data can be passed to machine learning algorithms. The developed methodology allows the transfer of meshed CAD data, scan data or meshes of finite element simulations into a unified form. Here, it is completely irrelevant how large the considered parts are or whether both small and large components are to be considered simultaneously. In addition, a methodology has been developed which creates uniform orientations of parts arranged individually in three-dimensional space. The implementation of the so-called detector matrix allows the dimensional reduction of three-dimensional point clouds to a two-dimensional matrix of fixed size. Thus, different geometries can be uniformly transferred to machine learning algorithms. Thus, a methodology is available to answer the first research question from chapter 3: *How can geometry information and characteristics associated with geometry be passed to machine learning algorithms?*

- **Three applications of the developed methodology:**

The developed methodology could be demonstrated in the context of a part recognition of CAD parts, the processing of scan data of optical measuring systems and in the context of plausibility checks for structural mechanic finite element simulations. In this way, the diversity of application fields of the developed methodology could

be demonstrated. All application examples showed very good results and the methodology enables the use of previously unused data sets of virtual product development for future developments. The methodology thus represents a building block of digital engineering in which the use of data is attracting increased interest. *This answers the questions which steps are necessary to build up a process chain from data creation to data analysis, and where potential difficulties can be observed?*

- **Utilization of data from the product development processes of previous products for the development process of new products:**

By passing geometry-based data to machine learning algorithms, such data sets can be made much easier to use. For example, part recognition can be used to search old CAD data for identical parts or to determine a similarity value of a part with respect to other parts. This allows a better use of historical data sets and leads to the avoidance of the development of very similar parts or components by different persons at different times. The "simulation DNA" provides a possibility to evaluate simulations automatically or to determine a similarity index for finite element simulations based on these image values. The further fields of application of the method are very versatile and can be identified by corresponding experts also at different stages of the product development process or beyond. *This is the answer to the question, which adaptations to the methodology are necessary to process different use cases and thus different data sets.*

- **Demonstration of the relevant aspects and sub-steps for the feasibility of the developed methodology by third parties:**

Within the scope of the present thesis, all relevant sub-steps and the individual procedure for implementation were shown in great detail on the basis of the three application examples. This enables the further implementation of the developed method in the academic as well as in the industrial environment. Especially for people with less experience in the field of data-driven product development and machine learning, the work should provide input for thought and illustrate the potentials that lie in the data. By describing the approach in detail, product developers can be encouraged to think outside the box. New methods of digital engineering can be used to improve existing processes even further, and it allows the

development of new business models through the end-to-end use of data, especially in the area of product development and creation. *Thus, it can be seen which steps are necessary to apply the developed methodology in industry as well as in academia.*

Further extensions of the developed methodology are manifold. From the perspective of digital engineering with a focus on product development, the following points are particularly relevant:

Currently, the projection of the point cloud onto the detector surface is done with pixels defined either by a uniform angle or by means of uniform area of each detector pixel. Instead of these two methods, further variants can be developed (e. g. depending on the mesh density of the used FE mesh).

The scope of the used data for the training can be extended considerably. Instead of finite element simulations of single parts the methodology can be extended onto assemblies of multiple parts. For this purpose, additional matrices for the contacts between the different components and a corresponding normalization of these contact matrices must be developed. The same applies to additional result variables from these finite element simulations.

It can also be evaluated to what extent the above methodology can be integrated as an automatic safeguarding step in existing process chains or which changes have to be made in order to achieve this. Here it can be helpful to use a machine learning algorithm, where humans can better understand the decision of the algorithm than it is the case with artificial neural networks.

Due to the current rapid development of powerful GPUs and the associated possibilities for data analysis through a variety of software frameworks, much more difficult problems can be solved soon. However, this requires the availability of extensive data sets. For this reason, the author appeals to all institutions to store relevant data in a structured way. This also applies to data that cannot be used at present. This is especially true if this data or a meaningful evaluation of the data can be used to foresee a potential for savings or future optimizations. Thus, the developed detector matrices can also be used as a feature for significantly more extensive evaluations, comparable to other statistical parameters such as the mean value or the standard deviation.

## Literaturverzeichnis

- [1] Eigner, Martin; Roubanov, Daniil; Zafirov, Radoslav: Modellbasierte virtuelle Produktentwicklung. Berlin, Heidelberg: Springer-Verlag, 2014.
- [2] Meerkamm, Harald: Design for X - A Core Area of Design Methodology. In: Journal of Engineering Design, Volume 5, Issue 2 (1994), S. 165-181, DOI: 10.1080/09544829408907881.
- [3] DEKRA Sonstige Crashversuche - Vom Rollover bis zum Brandschutz. <https://www.dekra.de/de/sonstige-crashversuche/> [Stand 26.01.2020, 11:00].
- [4] Pahl, Gerhard; Beitz, Wolfgang; Feldhusen, Jörg; Grote, Karl-Heinz. Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung. Berlin Heidelberg: Springer-Verlag, 2003.
- [5] Lindemann, Udo: Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden. Berlin Heidelberg: Springer-Verlag, 2009.
- [6] VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme. Berlin: Beuth-Verlag, 2004.
- [7] VDI-Richtlinie 2221: Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung. Berlin: Beuth-Verlag, 2019.
- [8] Cooper, Robert: Stage-gate systems: A new tool for managing new products. In Business Horizons. Volume 33, Issue 3 (1990), S. 44-55.
- [9] Kiel, Daniel, Arnold, Christian, Voigt, Kai-Ingo: The influence of the Industrial Internet of Things on business models of established manufacturing companies - A business level perspective. In Technovation, Volume 68 (2017), S. 4-19.
- [10] Holmström, Helena; Bosch, Jan: Towards Data-Driven Product Development: A Multiple Case Study on Post-deployment Data Usage in Software-Intensive Embedded Systems. In: Fitzgerald B., Conboy K., Power K., Valerdi R., Morgan L., Stol KJ. (eds): Lean Enterprise Software and Systems. Lecture Notes in Business Information Processing, Volume 167 (2013). Berlin Heidelberg: Springer-Verlag.

- [11] VDMA Report Machine Learning im Maschinenbau 2019 bis 2022 - Ausgewählte Ergebnisse zu Lösungen, Erfahrungen und Entwicklungspotentialen. VDMA, 14.06.2019.
- [12] Wittpahl, V.: Digitalisierung: Bildung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.
- [13] Reinsel, David; Gantz, John; Rydning, John: Data Age 2025 - The Digitization of the World - From Edge to Core. IDC White Paper. URL: <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>. [Stand 17.04.2019, 20:03].
- [14] Jordan, Micheal I.; Mitchell, Tom M.: Machine learning: Trends, perspectives, and prospects. Science, Volume 349, Issue 6245 (2015), S.255-260.
- [15] Witten, Ian H.; Frank, Eibe; Hall, Mark A.; Pal, Christopher J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2016.
- [16] Alpaydin, Ethem: Introduction to Machine Learning (Adaptive Computation and Machine Learning). MIT Press, 2010.
- [17] Kohavi, Ron; Provost, Foster: Machine Learning - Glossary of Terms, Volume 30, Issue 2-3 (1998), S. 271-274.
- [18] Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A.: Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In Artificial Intelligence in Design'96, S. 151-170. Springer, Dordrecht. 1996.
- [19] Bishop, Christopher: Pattern Recognition and Machine Learning. Springer New York, 2006.
- [20] Fernandes de Mello, Rodrigo; Ponti, Moacir Antonelli: Machine Learning - A Pratical Approach on the Statistical Learning Theory. Springer International, 2018.
- [21] StackExchange Forum: Distinction between AI, ML, Neural Networks, Deep learning and Data mining. URL: <https://softwareengineering.stackexchange.com/questions/366996/distinction-between-ai-ml-neural-networks-deep-learning-and-data-mining> , [Stand 11.04.2019, 10:37].
- [22] Searle, John R.: Minds, Brains, and Programs. In Behavioral and Brain Sciences Journal, Volume 3, Issue 3 (1980), S. 417-457.

- [23] Nilsson, Nils J.: The quest for artificial intelligence. Cambridge University Press, 2009.
- [24] Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic: From Data Mining to Knowledge Discovery in Databases. *AI magazine*, Volume 17, Issue 3 (1996), S. 37-37.
- [25] Weiss, Sholom M.; Kulikowski, Casimir A.: Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. Volume 123. San Mateo, CA: Morgan Kaufmann, 1991.
- [26] Hand, David J.: Discrimination and classification. *Wiley Series in Probability and Mathematical Statistics*. Wiley Chichester, 1981.
- [27] Draper, Norman R.; Smith, Harry: Applied regression analysis. Volume 326. John Wiley & Sons, 2014.
- [28] Brandt, Siegmund; Datenanalyse. Spektrum Heidelberg, 1999.
- [29] Jain, Anil K.; Murty, M. Narasimha; Flynn, Patrick J.: Data Clustering: A Review. In *ACM computing surveys (CSUR)*, Volume 31, Issue 3 (1999), S. 264-323.
- [30] Chandola, V; Banerjee, Arindam; Kumar, Vipin: Anomaly Detection: A Survey. In *ACM computing surveys (CSUR)*, Volume 41, Issue 3 (2009), S. 15-86.
- [31] Agrawal, Rakesh; Imielinski, Tomasz; Swami, Arun: Mining Association Rules between Sets of Items in Large Databases. In *Acm sigmod record*, Volume 22, Issues 2 (1993), S. 207-216.
- [32] Hahn, Udo; Mani, Inderjeet: The Challenges of Automatic Summarization. In *Computer*, Volume 33, Issue 11 (2000), S. 29-36.
- [33] Witten, Ian H.; Frank, Eibe; Hall, Mark A.: *Data Mining - Practical Machine Learning Tools and Techniques*. Burlington: Elsevier, 2011.
- [34] Nabil, Mahmoud; Aly, Mahamed; Atiya, Amir F.: ASTD: Arabic Sentiment Tweets Dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015)*, S. 2515-2519.
- [35] Razavi, Saman; Tolson, Bryan A.; Burn, Donal H.: Review of surrogate modeling in water resources. In *Water Resources Research*, Volume 48 (2012), W07401.

- [36] James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert: An Introduction to Statistical Learning - with Applications in R. New York: Springer, 2013.
- [37] Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin: Introduction to Data Mining. Boston: Pearson, 2006.
- [38] Hosmer, David W.; Lemeshow, Stanley: Goodness of fit tests for multiple logistic regression model. In Communications in Statistics - Theory and Methods, Volume 9, Issue 10 (1980), S. 1043-1069.
- [39] Sammut, Claude; Webb, Geoffrey I.: Encyclopedia of Machine Learning. Boston: Springer, 2010.
- [40] Shekhar, Shashi; Xiong, Hui; Zhou, Xun: Encyclopedia of GIS. Cham: Springer, 2017.
- [41] Will, Johannes; Most, Thomas: Metamodell of optimized Prognosis (MoP) - an Automatic Approach for User Friendly Parameter Optimization. Weimarer Optimierungs-und Stochastiktag 6, 2009.
- [42] Fawcett, Tom: An introduction to ROC analysis. Pattern Recognition Letters, Volume 27 (2006), S. 861-874.
- [43] Powers, David M.W.: Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. In Journal of Machine Learning Technologies, Volume 2, Issue 1 (2011), S. 37-63.
- [44] Kohn, Wolfgang: Statistik - Datenanalyse und Wahrscheinlichkeitsrechnung. Springer-Verlag, Berlin Heidelberg. 2005.
- [45] Stehman, Steven V.: Selecting and interpreting measures of thematic classification accuracy. In Remote Sensing Environment, Volume 62, Issue 1 (1997), S. 77-89.
- [46] Osuna, Edgar; Freund, Robert; Girosi, Federico: Training Support Vector Machines: an Application to Face Detection. In: Proceedings of CVPR'97, Volume 97, Issue 130-136 (1997), S. 99-106.
- [47] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in neural information processing systems (2012), S. 1097-1105.

- [48] Noordin, M.Y.; Venkatesh, V.C.; Sharif, S.; Elting, E.; Abdullah, A.: Application of response surface methodology in describing the performance of coated carbide tools when turning AISI 1045 steel. In *Journal of Materials Processing Technology*, Volume 145, Issue 1 (2004), S.46-58.
- [49] Oyelade, O.J.; Oladipupo, O.O.; Obagbuwa, I.C.: Application of k-Means Clustering algorithm for prediction of Students' Academic Performance. In *International Journal of Computer Science and Information Security (IJCSIS)*, Volume 7, Issue 1 (2010), S. 292-295.
- [50] Chaira, Tamalika: A novel intuitionistic fuzzy C means clustering algorithm and its application to medical images. In *Applied Soft Computing* Volume 11 (2011), S. 1711-1717.
- [51] Andrade, Ernesto L.; Blunsden, Scott; Fisher, Robert B.: Hidden Markov Models for Optical Analysis of Crowds. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Volume 1 (2006), S. 460-463.
- [52] Schutzrecht US 9349103-B2 (2019-04-26). Eberhardt, John S.; Radano, Todd A.; Peterson, Benjamin E.: Application of machine learned Bayesian networks to detection of anomalies in complex systems.
- [53] Artero, Almir Olivette; Oliveira, Maria Cristina Ferreira de; Levkowitz, Haim: Uncovering Clusters in Crowded Parallel Coordinates Visualizations. In *IEEE Symposium on Information Visualization* (2004), S. 81-88.
- [54] Vellido, Alfredo; Garcia, David L.; Nebot, Angela: Cartogram visualization for nonlinear manifold learning models. In *Data Mining and Knowledge Discovery*, Volume 27, Issue 1 (2013), S. 22-54.
- [55] Schmidhuber, Jürgen: Deep learning in neural networks: An overview. In *Journal of Neural networks*, Volume 61 (2015), S. 85-117.
- [56] Brachman, Ronald, J.: (AA)AI: More than the Sum of Its Parts. In *AI Magazine*, Volume 27, Issue 4 (2006), S. 19-34.
- [57] Grudin, Jonathan: AI and HCI: Two Fields Divided by a Common Focus. In *AI Magazine*, Volume 30, Issue 4 (2009), S. 48-57.

- [58] Google Trends Analyse für die Begriffe «Künstliche Neuronale Netze», «Deep Learning» und «Convolutional Neural Networks»: <https://trends.google.de/trends> [Stand 03.01.2021, 13:20].
- [59] Rojas, R.: Neural networks: A systematic introduction. Berlin. London: Springer-Verlag, 2015.
- [60] Beck, H.: Biologie des Geistesblitzes - speed up your mind!. Berlin, Heidelberg: Springer Spektrum, 2013.
- [61] Riedmiller, Martin: Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. In Computer Standards & Interfaces, Volume 16, Issue 3 (1994), S. 265-278.
- [62] Werbos, Paul J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Dissertation, Harvard University, 1974.
- [63] Parker, David B.: Learning-logic, invention report. File 1, (1982), S. 81-64.
- [64] Parker, David B.: Learning Logic Technical Report TR-47. Center of Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, 1985.
- [65] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J.: Learning internal representations by error propagation. In ICS Report Number 8506 (1985). Institute for Cognitive Science, University of California San Diego.
- [66] Widrow, Bernard; Lehr, Michael A.: 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. In Proceedings of the IEEE, Volume 78, Issue 9 (1990), S. 1415-1441.
- [67] Kohonen, Teuvo: Self-Organization and Associative Memory. Berlin: Springer, 1988.
- [68] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D.: Backpropagation applied to handwritten zip code recognition. In Neural computation, Volume 1, Issue 4 (1989), S. 541-551.
- [69] Levenberg, Kenneth: A method for the solution of certain problems in least squares. In Quaterla of Applied Mathematics, Volume 5 (1944), S. 164-168.

- [70] Marquardt, Donald: An algorithm for least-squares estimation of nonlinear parameters. In *SIAM Journal on Applied Mathematics*, Volume 11, Issue 2 (1963), S. 431-441.
- [71] Hao, Yu; Wilamowski, Bogdan M.: Levenberg-marquardt training. In *Industrial electronics handbook*, Volume 5, Issue 12 (2011), S. 12-1 - 12-16.
- [72] Moller, Martin Fodslette: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. In *Neural Networks*, Volume 6, Issue 4 (1993), S. 525-533.
- [73] Johansson Erik M.; Dowla, Farid U.; Goodman, Dennis M.: Backpropagation Learning for Multilayer feed-forward Neural Networks using the Conjugate Gradient Method. In *International Journal of Neural Systems*, Volume 2, Issue 4 (1991), S. 291-301.
- [74] Hagan, Martin T.; Demuth, Howard B.; Beale, Mark Hudson; De Jesus, Orlando: *Neural Network Design*. eBook: Martin Hagan, 2014.
- [75] Beale, Mark Hudson; Hagan, Martin T.; Demuth, Howard B.: *Neural Network Toolbox 7 Users's Guide*. MathWorks Inc., 2010.
- [76] Ruder, Sebastian: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016.
- [77] Livingstone, David J.; Manallack, David T.; Tetko, Igor V.: Data modelling with neural networks: Advantages and limitations. In *Journal of Computer-Aided Molecular Design*, Volume 11, Issue 2 (1997), S. 135-142.
- [78] Kruse, Rudolf; Borgelt, Christian; Braune, Christian; Klawonn, Frank; Moewes, Christian; Steinbrecher, Matthias: *Computational Intelligence: Eine Methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Wiesbaden: Springer Vieweg, 2015.
- [79] Aggarwal, Charu C.: *Neural Networks and Deep Learning*. Springer Nature Switzerland, 2018.
- [80] Garro, Beatriz, A.; Vazquez, Roberto, A.: A New Generalized Neuron Model Applied to DNA Microarray Classification. In *International Conference on Engineering Applications of Neural Networks*, Springer Nature Switzerland. 2019, S. 125-136.

- [81] Abadi, Martín; Barham, Paul; Chen, Jianmin; Chen, Zhifeng; Davis, Andy; Dean, Jeffrey; Devin, Matthieu; Ghemawat, Sanjay; Irving, Geoffrey; Isard, Michael; Kudlur, Manjunath; Levenberg, Josh; Monga, Rajat; Moore, Sherry; Murray, Derek G.; Steiner, Benoit; Tucker, Paul; Vasudevan, Vijay; Warden, Pete; Wicke, Martin; Yu, Yuan; Zheng, Xiaoqiang: Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (2016), S. 265-283.
- [82] Chollet, Francois: Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. mitp Professional. 2018.
- [83] Frank, Seide; Agarwal, Amit: CNTK: Microsoft's open-source deep-learning toolkit. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016), S. 2135-2135.
- [84] Jia, Yangqing; Shelhamer, Evan; Donahue, Jeff; Karayev, Sergey; Long, Jonathan; Girshick, Ross; Guadarrama, Sergio; Darrell, Trevor: Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM international conference on Multimedia (2014), S. 675-678.
- [85] Bergstra, James; Breuleux, Olivier; Bastien, Frederic; Lamblin, Pascal; Pascanu, Razvan; Desjardins, Guillaume; Turian, Joseph; Bengio, Yoshua: Theano: A CPU and GPU Math Expression Compiler. In Proceedings of the Python for scientific computing conference (SciPy), Volume 4, Number 3. Oral Presentation.
- [86] Pedregosa, Fabian; Varoquaux, Gael; Gramfort, Alexandre; Michel, Vincent; Thirion, Bertrand; Grisel, Olivier; Blondel, Mathieu; Prettenhofer, Peter; Weiss, Ron; Dubourg, Vincent; Vanderplas, Jake; Passos, Alexandre; Cournapeau, David; Brucher, Matthieu; Perrot, Matthieu; Duchesnay, Édouard: Scikit-learn: Machine learning in Python. Journal of machine learning research, Volume 12 (2011), S. 2825-2830.
- [87] Kirk, David: NVIDIA CUDA software and GPU parallel computing architecture. In ISMM, Volume 7 (2007), S. 103-104.
- [88] The NVIDIA CUDA® Deep Neural Network library (cuDNN): <https://developer.nvidia.com/cudnn> [Stand 20.12.2019, 11:30]

- [89] LeCunn, Yann; Bengio, Yoshua; Hinton, Geoffrey: Deep learning. In *nature*, Volume 521, Issue 7553 (2015), S. 436-444.
- [90] Goodfellow, Ian; Bengio Yoshua; Courville, Aaron: Deep Learning. MIT Press. online: <http://www.deeplearningbook.org>. 2016.
- [91] Bellman, Richard E.: *An Introduction to Artificial Intelligence: Can Computers Think?* Boyd & Fraser Publishing, San Francisco. 1978.
- [92] Charniak, Eugene; McDermott, Drew: *Introduction to Artificial Intelligence*. Addison-Wesley, Reading. 1985.
- [93] Haugeland, John: *Artificial Intelligence: The Very Idea*. MIT Press, Cambridge. 1985.
- [94] Kurzweil, Ray: *The Age of Intelligent Machines*. MIT Press, Cambridge. 1990.
- [95] Luger, George F.; Stubblefield, William. A.: *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Benjamin/Cummings, Redwood City. 1993.
- [96] Rich, Elaine; Knight, Kevin: *Artificial Intelligence*. McGraw-Hill, New York. 1991.
- [97] Schalkoff, Robert J.: *Artificial Intelligence: An Engineering Approach*. McGraw-Hill, New York. 1990.
- [98] Winston, Patrick Henry: *Artificial Intelligence*. Addison-Wesley Longman, Amsterdam. 1993.
- [99] Russel, Stuart J.; Norvig, Peter: *Artificial Intelligence - A Modern Approach*. Prentice Hill, New Jersey. 1995.
- [100] Aizenberg, Igor N.; Aizenberg, Naum N.; Vanderwalle, Joos: *Multi-Valued and Universal Binary Neurons - Theory, Learning and Applications*. Springer Science+Business Media, Dordrecht. 2000.
- [101] Mhaskar, Hrushikesh; Liao, Qianli; Poggio, Tomaso: *When and Why Are Deep Networks Better than Shallow Ones?*. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (2017)*, S. 2343-2349.
- [102] Hubel, David H.; Wiesel, Torsten N.: *Receptive fields of single neurones in the cat's striate cortex*. In *Journal of Physiology*, Volume 148, Issue 3 (1959), S. 574-591.

- [103] Hubel, David H.; Wiesel, Torsten N.: Receptive fields and functional architecture of monkey striate cortex. In *Journal of Physiology*, Volume 195, Issue 1 (1968), S. 215-243.
- [104] Hubel, David H.; Wiesel, Torsten N.: Receptive fields, binocular interaction and functional architecture in cat's visual cortex. In *Journal of Physiology*, Volume 160, Issue 1 (1962), S. 106-154.
- [105] Hubel, David H.; Wiesel, Torsten N.: Receptive fields and functional architecture in two nonstriate visual area (18 and 19) of the cat. In *Journal of Neurophysiology*, Volume 28, Issue 2 (1965), S. 229-289.
- [106] Fukushima, Kunihiko: Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. In *Journal Biological Cybernetics*, Volume 36, Issue 4 (1980), S. 193-202.
- [107] Little, William A: The existence of persistent states in the brain. In *From High-Temperature Superconductivity to Microminiature Refrigeration*. Springer, Boston. 1974, S. 145-164.
- [108] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J.: Learning representations by back-propagating errors. In *nature*, Volume 323, Issue 6088 (1986), S. 533-536.
- [109] Hopfield, John J.: Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the national academy of sciences* , Volume 79, Issue 8 (1982). S. 2554-2558.
- [110] Salvaris, Mathew; Dean, Danielle; Tok, Wee Hyong: *Deep Learning with Azure - Building and Deploying Artificial Intelligence Solutions on the Microsoft AI Platform*. Apress, Berkeley. 2018.
- [111] Miljanovic, Milos: Comparative analysis of recurrent and finite impulse response neural networks in time series prediction. In *Indian Journal of Computer Science and Engineering*, Volume 3, Issue 1 (2012), S. 180-191.
- [112] Hochreiter, Sepp; Schmidhuber, Jürgen: Long short-term memory. In *Neural Computation*, Volume 9, Issue 8, S. 1735-1780.
- [113] Gers, Felix A.; Schmidhuber, Jürgen; Cummins, Fred: Learning to forget: Continual prediction with LSTM. In *Proceedings of 9th International Conference on Artificial Neural Networks* (1999), Edinburgh, S. 850-855.

- [114] Gers, Felix A.; Schmidhuber, Jürgen; Cummins, Fred: Learning to Forget: Continual Prediction with LSTM. In *Journal Neural Computation*, Volume 12, Issue 10 (2000), S. 2451-2471.
- [115] Cho, Kyunghyun; Van Merriënboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua: Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv* (2014): S. 1406.1078.
- [116] Britz, Denny: Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano. *WILDML: Artificial Intelligence, Deep Learning and NLP*: <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/> [Stand 09.02.2020, 14:50].
- [117] Islam, Md Matriqul; Lam, Antony; Fukuda, Hisato; Kobayashi, Yoshinori; Kuno, Yoshinori: An intelligent shopping support robot: understanding shopping behaviour from 2D skeleton data using GRU network. In *Robomech Journal*, Volume 6, Issue 18 (2019). Open Access <https://doi.org/10.1186/s40648-019-0150-1>.
- [118] Chung, Junyoung; Gulcehre, Caglar; Cho, KyungHyun; Bengio, Yoshua: Empirical evaluation of gated re-current neural networks on sequence modeling. *arXiv preprint arXiv* (2014): S. 1412.3555.
- [119] Jozefowicz, Rafal; Zaremba, Wojciech; Sutskever, Ilya: An empirical exploration of recurrent network architectures. In *International conference on machine learning* (2015), S. 2342-2350.
- [120] Ayyadevara, V. Kishore: *Pro Machine Learning Algorithms - A Hands-On Approach to Implementing Algorithms in Python and R*. Springer Science+Business Media, New York, 2018.
- [121] Aghdam, Hamed Habibi; Heravi, Elnaz Jahani: *Guide to Convolutional Neural Networks - A Practical Application to Traffic-Sign Detection and Classification*, Springer Nature, Cham, 2017.
- [122] Zeiler, Matthew D.; Fergus, Rob: Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv* (2013): S. 1301.3557.

- [123] Yu, Dingjun; Wang, Hanli; Chen, Peiqiu; Wei Zhihua: Mixed Pooling for Convolutional Neural Networks. In Proceedings 9th International Conference Rough Sets and Knowledge Technology (2014), Shanghai, S. 364-375.
- [124] Aghdam, Hamed H.; Heravi, Elnaz J.; Puig, Domenec: Fusing Convolutional Neural Networks with a Restoration Network for Increasing Accuracy and Stability. In G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part I, LNCS 9913 (2016), S. 178-191.
- [125] Dong, Chao; Loy, Chen Change; He, Kaiming; Tang, Xiaoou: Image Super-Resolution Using Deep Convolutional Networks. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 38, Number 2 (2016), S. 295-307.
- [126] Xu, Li; Ren, Jimmy SJ.; Liu, Ce; Jia, Jiaya: Deep convolutional neural network for image deconvolution. In Advances in neural information processing systems, 2014, S. 1790-1798.
- [127] Lawrence, Steve; Giles, C. Lee; Tsoi, Ah Chung; Back, Andrew D.: Face Recognition: A Convolutional Neural-Network Approach. In IEEE Transactions on Neural Networks, Volume 8, Number 1 (1997), S. 98-112.
- [128] Li, Haoxiang; Lin, Zhe; Shen, Xiaohui; Brandt, Jonathan; Hua, Gang: A Convolutional Neural Network Cascade for Face Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (2015), S. 5325-5334.
- [129] Matsugu; Masakazu; Mori, Katsuhiko; Mitari, Yusuke; Kaneda, Yuji: Subject independent facial expression recognition with robust face detection using convolutional neural network. In Neural Networks Journal, Volume 16, Issue 5-6 (2003), S. 555-559.
- [130] Liang, Ming; Hu, Xiaolin: Recurrent Convolutional Neural Network for Object Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (2015), S. 3367-3375.
- [131] Ciresan, Dan Claudiu; Meier, Ueli; Gambardella, Luca Maria; Schmidhuber, Jürgen: Convolutional Neural Networks Committees For Handwritten Character Classification. In IEEE International Conference on Document Analysis and Recognition (2011), S. 1135-1139.

- [132] Maitra, Durjoy Sen; Bhattacharya, Ujjwal; Parui, Swapan K.: CNN Based Common Approach to Handwritten Character Recognition of Multiple Scripts. In IEEE 13th International Conference on Document Analysis and Recognition (ICDAR) (2015), S. 1021-1025.
- [133] Zbontar, Jure; LeCunn, Yann: Computing the Stereo Matching Cost with a Convolutional Neural Network. In Proceedings of the IEEE conference on computer vision and pattern recognition (2015), S. 1592-1599.
- [134] Anthimopoulos, Marios; Christodoulidis, Stergios; Ebner, Lukas; Christe, Andreas; Mougiakakou, Stavroula: Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network. In IEEE transactions on medical imaging, Volume 35, Issue 5 (2016), S. 1207-1216.
- [135] Ji, Shuiwang; Xu, Wei; Yang, Ming; Yu Kai: 3D Convolutional Neural Networks for Human Action Recognition. In IEEE transactions on pattern analysis and machine intelligence, Volume 35m, Issue 1 (2012), S. 221-231.
- [136] Prasoon, Adhish; Petersen, Kersten; Igel, Christian; Lauze, Francois; Dam, Erik; Nielsen, Mads: Deep Feature Learning for Knee Cartilage Segmentation Using a Triplanar Convolutional Neural Network. In Proceedings International conference on medical image computing and computer-assisted intervention (2013), Springer, Berlin Heidelberg, S. 246-253.
- [137] Hu Baotian; Lu, Zhengdong; Li, Hang; Chen, Qingcai: Convolutional Neural Network Architectures for Matching Natural Language Sentences. In Advances in neural information processing systems (2014), S. 2042-2050.
- [138] Kim, Yoon: Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv (2014), S. 1408.5882.
- [139] Karpathy, Andrej; Toderici, George; Shetty, Sanketh; Leung, Thomas; Sukthankar, Rahul; Fei-Fei, Li: Large-scale Video Classification with Convolutional Neural Networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2014), S. 1725-1732.
- [140] Simard, Patrice Y.; Steinkraus, Dave; Platt, John C.: Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In Icdar (2003), Volume 3, Number 2003.

- [141] Abdel-Hamid, Ossama; Deng, Li; Yu, Dong: Exploring Convolutional Neural Networks Structures and Optimization Techniques for Speech Recognition. In Proceedings Interspeech (2013), S. 3366-3370.
- [142] Smith, Samuel L.; Kindermans, Pieter-Jan; Ying, Chris; Le, Quoc V: Don't decay the learning rate, increase the batch size. arXiv preprint arXiv (2017), S. 1711.00489.
- [143] Pan, Sinno Jialin; Yang, Qiang: A Survey on Transfer Learning. In IEEE Transactions on Knowledge and Data Engineering, Volume 22, Number 10 (2010), S. 1345-1359.
- [144] Torrey, Lisa; Shavlik, Jude: Transfer learning. In (Olivas, Emilio Soria; Guerrero, Jose David Martin; Sober, Marcelino Martinez, Benedito, Jose Rafael Magdalena; Lopez, Antonio Jose Serrano eds.) Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI Publishing, Hersey, 2010, S. 242-264.
- [145] Fawzi, Alhussein; Moosavi-Dezfooli, Seyed-Mohsen; Frossard, Pascal: The robustness of deep networks: A geometrical perspective. In IEEE Signal Processing Magazine, Volume 34, Issue 6 (2017), S. 50-62.
- [146] Yu, Dong; Deng, Li: Deep learning and its applications to signal and information processing [exploratory dsp]. In IEEE Signal Processing Magazine, Volume 28, Issue 1 (2010), S. 145-154.
- [147] Litjens, Geert; Kooi, Thijs; Bejnordi, Babak Ehteshami; Setio, Arnaud Araindra Adiyoso; Ciompi, Francesco; Ghahfarokan, Mohsen; van der Laak, Jeroen A.W.M.; van Ginneken, Bram; Sánchez, Clara I.: A survey on deep learning in medical image analysis. In Medical image analysis, Volume 42 (2017), S. 60-88.
- [148] Zhang, Zhenhua; He, Qing; Gao, Jing; Ni, Ming: A deep learning approach for detecting traffic accidents from social media data. In Transportation research part C: emerging technologies, Volume 86 (2018), S. 580-596.
- [149] Chetlur, Sharan; Woolley, Cliff; Vandermersch, Philippe; Cohen, Jonathan; Tran, John; Catanzaro, Bryan; Shelhamer, Evan: cuDNN: Efficient Primitives for Deep Learning. arXiv preprint arXiv (2014): S. 1410.0759.

- [150] Jia, Xianyan; Song, Shutao; He, Wei; Wang, Yangzihao; Rong, Haidong; Zhou, Feihu; Xie, Liqiang; Guo, Zhenyu; Yang, Yuanzhou; Yu, Liwei; Chen, Tiegang; Hu, Guangxiao; Shi, Shaohuai; Chu, Xiaowen: Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. arXiv preprint arXiv (2018): S. 1807.11205.
- [151] Muresan, Horea; Oltean, Mihai: Fruit recognition from images using deep learning. In Acta Universitatis Sapientiae, Informatica, Volume 10, Issue 1 (2018), S. 26-42.
- [152] Najafabadi, Maryam M.; Villanustre, Flavio; Khoshgoftaar, Taghi M.; Seliya, Naeem; Wald, Randall; Muharemagic, Edin: Deep learning applications and challenges in big data analytics. Journal of Big Data, Volume 2, Issue 1 (2015), S. 1-21.
- [153] Caruana, Rich; Niculescu-Mizil, Alexandru: An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics. In Proceedings of the 23rd international conference on Machine learning (2006). S. 161-168.
- [154] Yu, Fisher; Seff, Ari; Zhang, Yinda; Song, Shuran; Funkhouser, Thomas; Xiao, Jianxiong: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv (2015): S. 1506.03365.
- [155] Kamilaris, Andreas; Prenafeta-Boldú, Francesc X.: Deep learning in agriculture: A survey. In Computers and electronics in agriculture, Volume 147 (2018), S. 70-90.
- [156] Gu, Juncheng; Chowdhury, Mosharaf; Shin, Kang G.; Zhu, Yibo; Jeon, Myeongjae; Qian, Junjie; Liu, Hongqiang; Guo, Chuanxiong: Tiresias: A GPU Cluster Manager for Distributed Deep Learning. In Proceedings of 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19), 2019, S. 485-500.
- [157] Advantages of Deep Learning | disadvantages of Deep Learning: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Deep-Learning.html> [Stand 07.03.2020, 12:00].
- [158] Zhang, Chiyuan; Bengio, Samy; Hardt, Samy; Recht, Benjamin; Vinyals, Oriol: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv (2017): S. 1611.03530.

- [159] Saxe, Andrew M.; Bansal, Yamini; Dapello, Joel; Advani, Madhu; Kolchinsky, Artemy; Tracey, Brendan D.; Cox, David D.: On the information bottleneck theory of deep learning. In *Journal of Statistical Mechanics: Theory and Experiment*, Volume 12 (2019), S.124020.
- [160] Ikeuchi, Katsushi: Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-Picking Tasks. In *International Journal of Computer Vision*, Volume 1, Issue 2 (1987), S. 145-165.
- [161] Düsing, M.; Sadiki, A.; Janicka, J.: Towards a classification of models for the numerical simulation of premixed combustion based on a generalized regime diagram. In *Combustion Theory and Modelling*, Volume 10, Issue 1 (2006), S. 105-132.
- [162] Bosche, F.; Haas, Carl T.: Automated retrieval of 3D CAD model objects in construction range images. In *Automation in Construction*, Volume 17, Issue 4 (2008), S. 499-512.
- [163] Aldoma, Aitor; Vincze, Markus; Blodow, Nico.; Gassow, David; Gedikli, Suat; Rusu, Radu Bogdan; Bradski, Gary: CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues. In *IEEE international conference on computer vision workshops (ICCV workshops)* (2011), S. 585-592.
- [164] Verschoof-van der Vaart, Wouter B.; Lambers, Karsten: Learning to Look at LiDAR: The Use of R-CNN in the Automated Detection of Archaeological Objects in LiDAR Data from the Netherlands. *Journal of Computer Applications in Archaeology*, Volume 2, Issue 1 (2019), S.31-40.
- [165] Antonarakis, A.S.; Richards, K.S.; Brasington, J.: Object-based land cover classification using airborne LiDAR. In *Remote Sensing Environment*, Volume 112, Issue 6 (2008), S. 2988-2998.
- [166] Ogawa, Takashi; Sakai, Hiroshi; Suzuki, Yasuhiro: Pedestrian Detection and Tracking using in-vehicle Lidar for Automotive Application. In *IEEE Intelligent Vehicles Symposium (IV)*, Baden Baden, 2011, S. 734-739.
- [167] Ireland, Jack; Walsh, Robert W.; Galsgaard, Klaus: Visualization of Three-Dimensional Datasets. In *Solar Physics*, Volume 181, Issue 1 (1998), S.87-90.

- [168] Fang, Yi; Xie, Jin; Dai, Guoxian; Wang, Meng; Zhu, Fan; Xu, Tiantian; Wong, Edward: 3D Deep Shape Descriptor. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, S. 2319-2328.
- [169] Albarelli, Andrea; Rodolà, Emanuele; Bergamasco, Filippo; Torsello, Andrea: A Non-Cooperative Game for 3D Object Recognition in Cluttered Scenes. In IEEE International conference on 3D imaging, modeling, processing, visualization and transmission, 2011, S. 252-259.
- [170] Bronstein, Alexander M.; Bronstein, Michael M.; Kimmel, Ron: Efficient computation of isometry-invariant distances between surfaces. In SIAM Journal on Scientific Computing, Volume 28, Issue 5 (2006), S. 1812-1836.
- [171] Chen, Xiaobai; Golovinskiy, Aleksey; Funkhouser, Thomas: A Benchmark for 3D Mesh Segmentation. In AcM transactions on graphics, Volume 28, Issue 3, S. 73-84.
- [172] De Goes, Fernando; Goldenstein, Siome; Velho, Luiz: A Hierarchical Segmentation of Articulated Bodies. In Computer graphics forum 2008, Volume 27, Issue 5, S. 1349-1356.
- [173] Katz, Sagi; Leifman, George; Tal, Ayellet: Mesh segmentation using feature point and core extraction. In The Visual Computer, Volume 21, Issue 8-10, S. 649-658.
- [174] Osada, Robert; Funkhouser, Thomas; Chazelle, Bernard; Dobkin, David: Shape Distributions. In ACM Transactions on Graphics (TOG), Volume 21, Issue 4 (2002), S. 807-832.
- [175] Rodolà, Emanuele; Bronstein, Alex M.; Albarelli, Andrea; Bergamasco, Filippo; Torsello, Andrea: A Game-Theoretic Approach to Deformable Shape Matching. In IEEE Conference on Computer Vision and Pattern Recognition, 2012, S. 182-189.
- [176] Rustamov, Raif M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In Proceedings of the fifth Eurographics symposium on Geometry processing (SGP), 2007, S. 225-233.
- [177] Qi, Charles R., Su, Hao; Mo, Kaichun; Guibas, Leonidas J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, S. 652-660.

- [178] Maturana, Daniel; Scherer, Sebastian: Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, S. 922-928.
- [179] Wu, Zhirong; Song, Shuran; Khosla, Aditya; Yu, Fisher; Zhang, Linguang; Tang, Xiaoou; Xiao, Jianxiong: 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, S. 1912-1920.
- [180] Qi, Charles R.; Su, Hao; Nießner, Matthias; Dai, Angela; Yan, Mengyuan; Guibas, Leonidas J.: Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, S. 5648-5656.
- [181] Su, Hang; Maji, Subhransa; Kalogerakis, Evangelos; Learned-Miller, Erik: Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision, 2015, S. 945-953.
- [182] Phong, Bui Tuong: Illumination for computer generated pictures. In Communications of the ACM, Volume 18, Issue 6 (1975), S. 311-317.
- [183] Ge, Liuhao; Liang, Hui; Yuan, Junsong; Thalmann, Daniel: Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, S. 3593-3601.
- [184] Bronstein, Alexander M.; Bronstein, Michael M.; Kimmel, Ron; Mahmoudi, Mona; Sapiro, Guillermo: A Gromov-Hausdorff Framework with Diffusion Geometry for Topologically-Robust Non-rigid Shape Matching. In International Journal of Computer Vision, Volume 89, Issue 2-3 (2010), S. 266-286.
- [185] Tabia, Hedi; Laga, Hamid; Picard, David; Gosselin, Philippe-Henri: Covariance Descriptors for 3D Shape Matching and Retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, S. 4185-4192.
- [186] Fang, Yi; Sun, Mengtian; Ramani, Karthik: Temperature Distribution Descriptor for Robust 3D Shape Retrieval. In IEEE CVPR 2011 WORKSHOPS, Colorado Springs, S. 9-16.

- [187] Guo, Kan; Zou, Dongqing; Chen, Xiaowu: 3D Mesh Labeling via Deep Convolutional Neural Networks. In *ACM Transactions on Graphics (TOG)*, Volume 35, Issue 1 (2015), S. 3:1-3:12.
- [188] Gal, Ran; Cohen-Or, Daniel: Salient Geometric Features for Partial Shape Matching and Similarity. In *ACM Transactions on Graphics (TOG)*, Volume 25, Issue 1 (2006), S. 130-150.
- [189] Kalogerakis, Evangelos; Hertzmann, Aaron; Singh, Karan: Learning 3D Mesh Segmentation and Labeling. In *ACM Transactions on Graphics (TOG)*, Volume 29, Issue 4 (2010), S. 102:1-102:12.
- [190] Shapira L.; Shalom, S.; Shamir, A.; Cohen-Or, D.; Zhang, H.: Contextual Part Analogies in 3D Objects. In *International Journal of Computer Vision*, Volume 89, Issue 2-3 (2010), S. 309-326.
- [191] Liu, Rong; Zhang, Hao; Shamir, Ariel; Cohen-Or, Daniel: A Part-aware Surface Metric for Shape Analysis. In *Computer Graphics Forum*, Volume 28, Issue 2 (2009), S. 397-406.
- [192] Hilaga, Masaki; Shinagawa, Yoshihisa; Kohmura, Taku; Kunii, Toshiyasu L.: Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, S. 203-212.
- [193] Belongie, Serge; Malik, Jitendra; Puzicha, Jan: Shape Matching and Object Recognition Using Shape Contexts. In *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Volume 4 (2002), S. 509-522.
- [194] Johnson, Andrew E.; Hebert, Martial: Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. In *IEEE Transactions on pattern analysis and machine intelligence*, Volume 21, Issue 5 (1999), S. 433-449.
- [195] Ovsjanikov, Maks; Bronstein, Alexander M.; Bronstein, Micheal M.; Guibas, Leonidas J.: Shape Google: a computer vision approach to invariant shape retrieval. In *IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, S. 320-327.
- [196] Bronstein, Alexander M.; Bronstein Michael M.; Guibas, Leonidas J.; Ovsjanikov, Maks: Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval. In *ACM Transactions on Graphics (TOG)*, Volume 30, Issue 1 (2011), S. 1.1-1.20.

- [197] Sun, Jian; Ovsjanikov, Maks; Guibas, Leonidas: A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings Computer graphics forum*, Volume 28, Number 5 (2009), S. 1383-1392.
- [198] Coifman, Ronald R.; Lafon Stéphane: Diffusion Maps. In *Journal of Applied and Computational Harmonic Analysis*, Volume 21, Issue 1 (2006), S. 5-30.
- [199] Reuter, Martin; Biasotti, Silvia; Giorgi, Daniela; Patanè, Guiseppe; Spagnuolo, Michaela: Discrete Laplace-Beltrami Operators for Shape Analysis and Segmentation. In *Journal of Computers & Graphics*, Volume 33, Issue 3 (2009), S. 381-390.
- [200] Reuter, Martin; Wolter, Franz-Erich; Shenton, Martha; Niethammer, Marc: Laplace-Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. In *Journal of Computer-Aided Design*, Volume 41, Issue 10 (2009), S. 739-755.
- [201] Reuter, Martin; Wolter, Franz-Erich; Peinecke, Niklas: Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids. In *Journal of Computer-Aided Design*, Volume 38, Issue 4 (2006), S. 342-366.
- [202] Masci, Jonathan; Boscaini, Davide; Bronstein, Micheal M.; Vandergheynst, Pierre: Geodesic Convolutional Neural Networks on Riemannian Manifolds. In *Proceedings of the IEEE international conference on computer vision workshop* (2015), S. 37-45.
- [203] Levy, Bruno: Laplace-beltrami eigenfunctions towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications SMI06* (2006), S. 13-13.
- [204] Aubry, Mathieu; Schlickewei, Ulrich; Cremers, Daniel: The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE international conference on computer vision workshops (ICCV workshops)* (2011), S. 1626-1633.
- [205] Bruna, Joan; Zaremba, Wojciech; Szlam, Arthur; LeCunn, Yann: Spectral Networks and Deep Locally Connected NETworks on Graphs. In *arXiv preprint arXiv* (2013), S. 1312.6203.
- [206] Qi, Charles R.; Yi, Li; Su, Hao; Guibas, Leonidas J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proceedings of Advances in neural information processing systems* (2017), S. 5099-5108.

- [207] American Society of Mechanical Engineers. V&V10: Guide for verification and validation in computational solid mechanics. American Society of Mechanical Engineers, 2006. ASME V&V 10-2006.
- [208] DIN EN ISO 9000: Qualitätsmanagementsysteme - Grundlagen und Begriffe. Ausgabe November 2015. Hrsg. DIN Deutsches Institut für Normung. Berlin: Beuth-Verlag 2015.
- [209] DIN 25478: Einsatz von Berechnungssystemen beim Nachweis der Kritikalitätssicherheit. Ausgabe Juni 2014. Hrsg. DIN Deutsches Institut für Normung. Berlin: Beuth-Verlag 2014.
- [210] Böhnert, M.; Reszke, P.: Linguistisch-philosophische Untersuchungen zu Plausibilität: über kommunikative Grundmuster bei der Entstehung von wissenschaftlichen Tatsachen. In: Engelschalt, Julia (Ed.) ; Maibaum, Arne(Ed.): Auf der Suche nach den Tatsachen: Proceedings der 1. Tagung des Nachwuchsnetzwerks "INSIST", 22.-23. Oktober 2014, Berlin. 2015. S. 40-67. URL: <http://nbn-resolving.de/urn:nbn:de:0168-ssoar-455901>.
- [211] VDI-Richtlinie 3843 Blatt 1:Modellbildung schwingungsfähiger Systeme. Berlin: Beuth-Verlag, 2018.
- [212] Sedaghat, Nima; Zolfaghari, Mohammadreza; Amiri, Ehsan; Brox, Thomas: Orientation-boosted voxel nets for 3d object recognition. 2016, arXiv preprint arXiv:1604.03351.
- [213] García, Salvador; Luengo, Julián; Herrera, Francisco: Data preprocessing in data mining (pp. 59-139). New York: Springer, 2015.
- [214] Kriegel, Hans-Peter; Borgwardt, Karsten M.; Kröger, Peer; Pryakhin, Alexey; Schubert, Matthias; Zimek, Arthur: Future trends in data mining. In Data Mining and Knowledge Discovery, Volume 15, Issue 1 (2007), S. 87-97.
- [215] Suresh, R. M.; Padmajavalli, R.: An overview of data preprocessing in data and web usage mining. In 2006 1st IEEE International Conference on Digital Information Management, 2006, S. 193-198.
- [216] Kotsiantis, S. B.; Kanellopoulos, D.; Pintelas, P. E.: Data preprocessing for supervised learning. In International Journal of Computer Science, Volume 1, Issue 2 (2006), S. 111-117.

- [217] Jolliffe, Ian: Principal component analysis. Berlin, Heidelberg: Springer, 1986.
- [218] Gray, Frank: Pulse code communication. US Patent US2632058 (A) — 1953-03-17. Bell Telephone Labor Inc. 1953.
- [219] Masters, Timothy: Practical Neural Network Recipes in C++. Academic Press Inc, 1993.
- [220] Benker, Rita; Heinrich, Kai; Brüggeman, Gert-Peter: Quantifizierung und Bewertung von Belastungen bei der Kabelbaummontage und Simulation einer Entlastung durch ein Unterstützungssystem. In 2. transdisziplinäre Konferenz «Technische Unterstützungssysteme, die die Menschen wirklich wollen», At Hamburg, 2016, S. 21-30.
- [221] <https://www.abenteuer-landy.de/defender-%C3%B6l-im-kabelbaum/> [Stand 10.10.2019, 09:00].
- [222] Keferstein, Claus P.; Marxer, Michael: Fertigungsmesstechnik. Springer Vieweg, Wiesbaden. 2015.
- [223] Wang, Zhou; Bovik, Alan C.; Sheikh, Hamid R.; Simoncelli, Eero P.: Image Quality Assessment: From Error Visibility to Structural Similarity. In IEEE Transactions on Image Processing, Volume 13, Issue 4 (2004), S. 600–612.
- [224] MSC Software Solutions - Simulating Reality, Delivering Certainty. [https://media.mscsoftware.com/cdn/farfuture/Yba9RarVxqwDg77thbeiKEolyqPz3fBCiV5GUZZzIzo/mtime:1568831960/sites/default/files/msc\\_software\\_company\\_overview.pdf](https://media.mscsoftware.com/cdn/farfuture/Yba9RarVxqwDg77thbeiKEolyqPz3fBCiV5GUZZzIzo/mtime:1568831960/sites/default/files/msc_software_company_overview.pdf) [Stand 18.10.2019, 10:10].
- [225] DIN EN ISO 4210-6: Fahrräder - Sicherheitstechnische Anforderungen an Fahrräder - Teil 6: Prüfverfahren für Rahmen und Gabel . Ausgabe Dezember 2015. Hrsg. DIN Deutsches Institut für Normung. Berlin: Beuth-Verlag 2015.
- [226] [http://www.smolik-velotech.de/glossar/r\\_RAHMENBELASTUNG.htm](http://www.smolik-velotech.de/glossar/r_RAHMENBELASTUNG.htm) [Stand 21.10.2019, 19:10].
- [227] Basshuysen, Richard van; Schäfer, Fred; Handbuch Verbrennungsmotoren. Springer Vieweg, Wiesbaden. 2014.
- [228] Köhler, Eduard; Flierl, Rudolf: Verbrennungsmotoren - Motormechanik, Berechnung und Auslegung des Hubkolbenmotors. Springer Vieweg, Wiesbaden. 2011.

- [229] Simonyan, Karen; Zissermann, Andrew: Very deep convolutional networks for large-scale image recognition. In arXiv preprint arXiv (2014), S. 1409-1556.
- [230] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, 2015, S. 1026-1034.
- [231] Ioffe, Sergey; Szegedy, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv (2015), S. 1502.03167.
- [232] Wu, Ren; Yan, Shengen; Shan, Yi; Dang, Qingqing; Sun, Gang: Deep image: Scaling up image recognition. arXiv preprint arXiv (2015), S. 1501.02876.
- [233] Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean; Huang, Zhiheng; Karpathy, Andrej; Khosla, Aditya; Bernstein, Michael; Berg, Alexander C.; Fei-Fei, Li: ImageNet Large Scale Visual Recognition Challenge. In International Journal of Computer Vision, Volume 115, Issue 3 (2015), S. 211-252.

## Verzeichnis promotionsbezogener, eigener Publikationen

- [P1] Walter, Michael; Spruegel, Tobias; Wartzack, Sandro: *Tolerance analysis of mechanism taking into account the interactions between deviations using meta-models*. In Kyvsgaard Hansen, Poul H.; Rasmussen, John; Jorgensen, Kaj Asbjorn; Tollestrup, Christian (Hrsg.), Proceedings of the 9th Norddesign (2012). S. 587-594.
- [P2] Sprügel, Tobias; Walter, Michael: *Integration elastischer Deformationen in die statistische Toleranzanalyse bewegter Systeme mittels Metamodellen*. In Wartzack, Sandro (Hrsg.), Summer School Toleranzmanagement: Beiträge zur 1. Summer School (2013), S. 46-49.
- [P3] Walter, Michael; Spruegel, Tobias; Wartzack, Sandro: *Tolerance analysis of systems in motion taking into account interactions between deviations*. Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture, Volume 207, Issue 5 (2013), S. 709-719.
- [P4] Spruegel, Tobias C.; Wartzack, Sandro: *Konzept zur automatischen Bauteilerkennung innerhalb der FE-Software-Umgebung mittels Künstlichen Neuronalen Netzen*. In Krause, Dieter; Paetzold, Kristin; Wartzack, Sandro (Hrsg.), Design for X. Beiträge zum 25. DfX-Symposium (2014), S. 47-56.
- [P5] Spruegel, Tobias C.; Walter, Michael; Wartzack, Sandro: *Robust Tolerance Design of Systems with varying ambient Temperature influence due to worldwide manufacturing and operation*. In Marjanovic, Dorian; Culley, Sue; Lindemann, Udo; McAlone, Tim C.; Weber, Christian (Hrsg.), International Design Conference - DESIGN (2014), S. 1189-1198.
- [P6] Walter, Michael; Spruegel, Tobias C.; Weikert, Tim; Mann, Vincent; Romeis, Markus; Wartzack, Sandro: *Toleranz-Kosten-Optimierung bewegter Systeme mittels Partikelschwarm-Optimierung*. In Krause, Dieter; Paetzold, Kristin; Wartzack, Sandro

- (Hrsg.), Design for X. Beiträge zum 25. DfX-Symposium (2014), S. 181-192.
- [P7] Breitsprecher, Thilo; Kestel, Philipp; Küstner, Christoph; Sprügel, Tobias; Wartzack, Sandro: *Einsatz von Data-Mining in modernen Produktentstehungsprozessen*. In ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb, Volume 110, Issue 11 (2015), S. 744-750.
- [P8] Hautsch, Stefan; Katona, Sebastian; Sprügel, Tobias; Koch, Michael; Rieg, Frank; Wartzack, Sandro: *Validierung von prozessgerecht strukturoptimierten Bauteilentwürfen mittels integrierter FEM-Realgeometrieanalyse*. In Krause, Dieter; Paetzold, Kristin; Wartzack, Sandro (Hrsg.), Design for X. Beiträge zum 26. DfX-Symposium (2015). S. 182-192.
- [P9] Katona, Sebastian; Sprügel, Tobias; Koch, Michael; Wartzack, Sandro: *Adapting FE-Meshes to real, 3D surface detected geometry data to improve FE-simulation results*. In Summary of Proceedings NAFEMS World Congress (2015). San Diego.
- [P10] Katona, Sebastian; Sprügel, Tobias; Koch, Michael; Wartzack, Sandro: *Structural Mechanics Analysis Using an FE-Mesh Adaption to Real, 3D Surface Detected Geometry Data*. In Journal of Mechanics Engineering and Automation, Volume 5, Issue 7 (2015), S. 387-394.
- [P11] Kestel, Philipp; Spruegel, Tobias C.; Katona, Sebastian; Wartzack, Sandro: *Konzept zur Umsetzung einer zentralen Wissensbasis für Simulationswissen im ANSYS Engineering Knowledge Manager*. In ANSYS Conference & 33th CADFEM Users' Meeting (2015). Bremen.
- [P12] Kestel, Philipp; Spruegel, Tobias C.; Katona, Sebastian; Lehnhäuser, Thomas; Wartzack, Sandro: *Concept and implementation of a central knowledge framework for simulation knowledge*. In NAFEMS European Conference: Simulation Process and Data Management (2015). München.

- [P13] Spruegel, Tobias C.; Hallmann, Martin; Wartzack, Sandro: *A concept for FE plausibility checks in structural mechanics*. In Summary of Proceedings NAFEMS World Congress (2015). San Diego.
- [P14] Spruegel, Tobias C.; Wartzack, Sandro: *Concept and application of automatic part-recognition with artificial neural networks for FE simulations*. In Weber, Christian; Husung, Stephan; Cantamessa, Marco.; Cascini, Gaetano; Marjanovic, Dorian; Graziosi, Serena. (Hrsg.), Proceedings of the 20th International Conference on Engineering Design (ICED15), Volume 2 (2015), S. 183-193.
- [P15] Spruegel, Tobias C.; Wartzack, Sandro: *Methodik zur Validierung, Verifikation und Auswahl mathematischer Metamodelle im Rahmen der virtuellen Produktentwicklung*. In Krause, Dieter; Paetzold, Kristin; Wartzack, Sandro (Hrsg.), Design for X. Beiträge zum 26. DfX-Symposium (2015). S. 279-288.
- [P16] Walter, Michael; Pribek, Michael; Spruegel, Tobias C.; Wartzack, Sandro: *Result Visualization and Documentation of Tolerance Simulations of Mechanisms*. In Weber, Christian; Husung, Stephan; Cantamessa, Marco.; Cascini, Gaetano; Marjanovic, Dorian; Graziosi, Serena. (Hrsg.), Proceedings of the 20th International Conference on Engineering Design (ICED15), Volume 4 (2015), S. 11-20.
- [P17] Walter, Michael S.J.; Spruegel, Tobias C.; Wartzack, Sandro: *Least Cost Tolerance Allocation for Systems with Time-variant Deviations*. In Procedia CIRP, Volume 27 (2015), S. 1-9.
- [P18] Walter, Michael; Spruegel, Tobias C.; Ziegler, Philipp; Wartzack, Sandro: *Berücksichtigung von Wechselwirkungen zwischen Abweichungen in der statistischen Toleranzanalyse*. In Konstruktion, Volume 10 (2015), S. 88-92.

- [P19] Kestel, Philipp; Sprügel, Tobias; Wartzack, Sandro: *Feature-basierte Modellierung und Bauteilerkennung in automatisierten, konstruktionsbegleitenden Finite-Elemente-Analysen*. In 34. CADFEM ANSYS Simulation Conference (2016). Nürnberg.
- [P20] Kestel, Philipp; Spruegel, Tobias C.; Wartzack, Sandro: *Assistance system for automated and knowledge-based setup and evaluation of design-accompanying Finite Element Analyses*. In Automotive Simulation World Congress 2016. München.
- [P21] Spruegel, Tobias S.; Kestel, Philipp; Wartzack, Sandro: *FEA-Assistenzsystem – Plausibilitätsprüfung für Finite-Elemente-Simulationen mittels sphärischer Detektorflächen*. In Krause, Dieter; Paetzold, Kristin; Wartzack, Sandro (Hrsg.), Beiträge zum 27. DfX-Symposium (2016). Jesteburg.
- [P22] Spruegel, Tobias C.; Wartzack, Sandro: *Das FEA-Assistenzsystem - Analyseteil FEdeLM*. In Ralph Stelzer (Hrsg.), Beiträge zur virtuellen Produktentwicklung und Konstruktionstechnik (2016). Dresden. S. 463-474.
- [P23] Spruegel, Tobias C.; Schröppel, Tina; Wartzack, Sandro: *Generic Approach to Plausibility Checks for Structural Mechanics with Deep Learning*. In Proceedings of the 21st International Conference on Engineering Design ICED17 (2017). Vancouver, S. 299 - 308.
- [P24] Hallmann, Martin; Sprügel, Tobias; Kelber, Vincent; Schleich, Benjamin; Wartzack, Sandro: *Steigerung der Bauteilqualität durch Erhöhung der Anlagensteifigkeit eines FDM-Druckers*. Konstruktion, Volume 70, Issue 7/8 (2018), S. 73-76.
- [P25] Spruegel, Tobias C.; Rothfelder, Richard; Bickel, Sebastian; Grauf, Andreas; Sauer, Christopher; Schleich, Benjamin; Wartzack, Sandro: *Methodology for plausibility checking of structural mechanics simulations using Deep Learning on existing simulation data*. In Proceedings of NordDesign (2018). Linköping.

- [P26] Spruegel, Tobias C.; Schleich, Benjamin; Wartzack, Sandro: *Simulations-DNA: Datengetriebener Ansatz zur automatischen Plausibilitätsprüfung strukturmechanischer FE-Simulationen mittels Deep Learning*. In VDI Wissensforum GmbH (Hrsg.), VDI Berichte 2333: SIMVEC - Simulation und Erprobung in der Fahrzeugentwicklung (2018), S. 29 - 38.
- [P27] Bickel, Sebastian; Spruegel, Tobias C.; Schleich, Benjamin; Wartzack, Sandro: *How Do Digital Engineering and Included AI Based Assistance Tools Change the Product Development Process and the Involved Engineers*. In Proceedings of the Design Society: International Conference on Engineering Design 2019, Volume 1, Issue 1 (2019), S. 2567-2576.
- [P28] Spruegel, Tobias C.; Bickel, Sebastian; Schleich, Benjamin; Wartzack, Sandro: *Approach and application to transfer heterogeneous simulation data from finite element analysis to neural networks*, Journal of Computational Design and Engineering, qwaa079, <https://doi.org/10.1093/jcde/qwaa079> (veröffentlicht 14. Dezember 2020).

**Verzeichnis promotionsbezogener, studentischer Arbeiten**

- [S1] Schröppel, Tina.: *Plausibilitätsprüfung für FE-Berechnungen mittels Künstlicher Neuronaler Netze*. Projektarbeit (2016), Erlangen.
- [S2] Schwarz, Sebastian: *Plausibilitätsprüfung von Finite-Elemente-Simulationen mittels Convolutional Neural Networks*. Masterarbeit (2017), Erlangen.
- [S3] Stadler, Manuel: *Generierung einer Datenbasis zur Plausibilitätsprüfung von FE-Simulationen mittels Deep Learning*. Bachelorarbeit (2017), Erlangen.
- [S4] Grauf, Andreas: *Entwicklung einer Methodik zur einheitlichen Orientierung dreidimensionaler Punktwolken*. Projektarbeit (2017), Erlangen.
- [S5] Blank, Miriam: *Plausibilitätsprüfung von Finite-Elemente-Simulationen mittels Deep Learning*. Bachelorarbeit (2017), Erlangen.
- [S6] Bickel, Sebastian: *Entwicklung von Methoden zur Dimensionsreduktion von Ergebnissen aus Finite-Elemente-Simulationen*. Projektarbeit (2017), Erlangen.
- [S7] Tumpati, Ramesh: *Training of Convolutional Neural Networks with high prediction quality for the evaluation of simulation results*. Masterthesis (2018), Erlangen.
- [S8] Tiebl, Daniel: *Training und Anpassung von Convolutional Neural Networks zur Plausibilitätsprüfung strukturmechanischer FE-Berechnungsergebnisse*. Masterarbeit (2018), Erlangen.
- [S9] Wackerbauer, Veith: *Analytische Untersuchung der Verformungen eines Kurbeltriebs zur Einbringung in die Toleranzanalyse*. Studienarbeit (2009), Erlangen.



# Anhang

Im Folgenden finden sich einige Tabellen und Bilder, welche für das bessere Verständnis hilfreich sind. Diese Tabellen und Bilder sind an den relevanten Stellen dieser Dissertation entsprechend erwähnt.

Tabelle 26: Verwendete Normteile als Basis des Datensatzes zur Bauteilerkennung, Teil 1

Nr.	Bauteil	Normbezeichnung
B1	Zylinderkopfschraube	ISO 4762 – M6 x 20
B2		ISO 4762 – M6 x 25
B3		ISO 4762 – M6 x 30
B4		ISO 4762 – M6 x 40
B5		ISO 4762 – M8 x 20
B6		ISO 4762 – M8 x 30
B7		ISO 4762 – M8 x 40
B8	Zylinderschraube mit Schlitz	ISO 1207 – M6 x 20
B9		ISO 1207 – M6 x 25
B10		ISO 1207 – M6 x 30
B11		ISO 1207 – M6 x 40
B12		ISO 1207 – M8 x 20
B13		ISO 1207 – M8 x 30
B14		ISO 1207 – M8 x 40
B15	Sechskantschraube	ISO 4017 – M6 x 20
B16		ISO 4017 – M6 x 25
B17		ISO 4017 – M6 x 30
B18		ISO 4017 – M6 x 40
B19		ISO 4017 – M8 x 20
B20		ISO 4017 – M8 x 30
B21		ISO 4017 – M8 x 40

Tabelle 27: Verwendete Normteile als Basis des Datensatzes zur Bauteilerkennung, Teil 2

Nr.	Bauteil	Normbezeichnung
B22	Sechskantschraube mit Schaft	ISO 4015 – M6 x 25
B23		ISO 4015 – M6 x 30
B24		ISO 4015 – M6 x 40
B25		ISO 4015 – M8 x 30
B26		ISO 4015 – M8 x 40
B27	Sechskantschraube mit Flansch	ISO 4162 – M6 x 30
B28		ISO 4162 – M6 x 40
B29		ISO 4162 – M8 x 40
B30	Spannstift	ISO 13337 – 6 x 20
B31		ISO 13337 – 6 x 26
B32		ISO 13337 – 6 x 30
B33		ISO 13337 – 6 x 40
B34		ISO 13337 – 8 x 20
B35		ISO 13337 – 8 x 30
B36		ISO 13337 – 8 x 40
B37	Zylinderstift	ISO 2338 – 6 x 20
B38		ISO 2338 – 6 x 26
B39		ISO 2338 – 6 x 30
B40		ISO 2338 – 6 x 40
B41		ISO 2338 – 8 x 20
B42		ISO 2338 – 8 x 30
B43		ISO 2338 – 8 x 40
B44	Sicherungsringe für Wellen	DIN 471 – 20 x 1,75
B45		DIN 471 – 25 x 2
B46		DIN 471 – 30 x 2
B47		DIN 471 – 35 x 2,5
B48		DIN 471 – 40 x 2,5

Tabelle 28: Verwendete Normteile als Basis des Datensatzes zur Bauteilerkennung, Teil 3

Nr.	Bauteil	Normbezeichnung
B49	Sicherungsringe für Wellen	DIN 471 – 50 x 2
B50		DIN 471 – 60 x 3
B51	Sicherungsringe für Bohrungen	DIN 472 – 20 x 1,5
B52		DIN 472 – 25 x 1,5
B53		DIN 472 – 30 x 1,5
B54		DIN 472 – 35 x 1,75
B55		DIN 472 – 40 x 2
B56		DIN 472 – 50 x 2
B57		DIN 472 – 60 x 3
B58	Sechskantmutter	ISO 4032 – M6
B59		ISO 4032 – M8
B60		ISO 4032 – M10
B61		ISO 4032 – M12
B62	Sechskantmutter ISO Typ 2	ISO 4033 – M6
B63		ISO 4033 – M8
B64		ISO 4033 – M10
B65		ISO 4033 – M12
B66	Sechskantmutter ISO Typ 1	ISO 4034 – M6
B67		ISO 4034 – M8
B68		ISO 4034 – M10
B69		ISO 4034 – M12
B70	Sechskantmutter niedrige Form	ISO 4035 – M6
B71		ISO 4035 – M8
B72		ISO 4035 – M10
B73		ISO 4035 – M12

Tabelle 29: Kennwerte zur Klassifikation von Bauteilen des 1. Anwendungsbeispiels (Werte  $\leq 1 \cdot 10^{-30}$ : bekannte Bauteile des Trainingsdatensatzes, Werte  $> 1 \cdot 10^{-30}$ : unbekannte Bauteile welche nicht im Testdatensatz enthalten waren)

Bauteilbezeichnung	Kennwert	bekannt / unbekannt
Trägerbauteil	0,0130	unbekannt
Lasche Nr. 1	0,0077	unbekannt
Lasche Nr. 2	0,0397	unbekannt
Welle Nr. 1	0,0016	unbekannt
Welle Nr. 2	0,0012	unbekannt
Welle Nr. 3	0,0020	unbekannt
Welle Nr. 4	0,006	unbekannt
Welle Nr. 5	0,0219	unbekannt
16x ISO 4035 – M12	$9,836 \cdot 10^{-33}$	bekannt
alle anderen Bauteile	0	bekannt

Tabelle 30: Erzeugbare Matrizen aus strukturmechanischen FE-Simulationen, Teil 1

Nr.	abzubildende FE-Größe	Betrachtung von Zwischenknoten	Wertebereich
1	Netz	Ja	Für jeden zusätzlich projizierten Knoten wird die Pixelsumme im jeweiligen Pixel um 1 erhöht
2	gesperrte Translation um X	Ja	Jeder projizierte und gesperrte Knoten um X erhöht die Pixelsumme im jeweiligen Pixel um 1
3	gesperrte Translation um Y	Ja	Jeder projizierte und gesperrte Knoten um Y erhöht die Pixelsumme im jeweiligen Pixel um 1
4	gesperrte Translation um Z	Ja	Jeder projizierte und gesperrte Knoten um Z erhöht die Pixelsumme im jeweiligen Pixel um 1
5	gesperrte Rotation um X	Ja	Jeder projizierte und gesperrte Knoten um X erhöht die Pixelsumme im jeweiligen Pixel um 1
6	gesperrte Rotation um Y	Ja	Jeder projizierte und gesperrte Knoten um Y erhöht die Pixelsumme im jeweiligen Pixel um 1
7	gesperrte Rotation um Z	Ja	Jeder projizierte und gesperrte Knoten um Z erhöht die Pixelsumme im jeweiligen Pixel um 1

Tabelle 31: Erzeugbare Matrizen aus strukturmechanischen FE-Simulationen, Teil 2

Nr.	abzubildende FE-Größe	Betrachtung von Zwischenknoten	Wertebereich
8	Externe positive Verschiebung in X	Ja	Die positiven externen Verschiebungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
9	Externe positive Verschiebung in Y	Ja	Die positiven externen Verschiebungen der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert
10	Externe positive Verschiebung in Z	Ja	Die positiven externen Verschiebungen der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert
11	Externe negative Verschiebung in X	Ja	Die negativen externen Verschiebungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
12	Externe negative Verschiebung in Y	Ja	Die negativen externen Verschiebungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
13	Externe negative Verschiebung in Z	Ja	Die negativen externen Verschiebungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
14	positive Kraft in X	Ja	Die angreifenden positiven Kräfte der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
15	positive Kraft in Y	Ja	Die angreifenden positiven Kräfte der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert
16	positive Kraft in Z	Ja	Die angreifenden positiven Kräfte der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert
17	negative Kraft in X	Ja	Die angreifenden negativen Kräfte der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
18	negative Kraft in Y	Ja	Die angreifenden negativen Kräfte der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert
19	negative Kraft in Z	Ja	Die angreifenden negativen Kräfte der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert

Tabelle 32: Erzeugbare Matrizen aus strukturmechanischen FE-Simulationen, Teil 3

Nr.	abzubildende FE-Größe	Betrachtung von Zwischenknoten	Wertebereich
20	positives Moment um X	Ja	Die umgerechneten Momente werden als positive Kräfte des projizierten Knotens in X-Richtung im Pixel aufsummiert
21	positives Moment um Y	Ja	Die umgerechneten Momente werden als positive Kräfte des projizierten Knotens in Y-Richtung im Pixel aufsummiert
22	positives Moment um Z	Ja	Die umgerechneten Momente werden als positive Kräfte des projizierten Knotens in Z-Richtung im Pixel aufsummiert
23	negatives Moment um X	Ja	Die umgerechneten Momente werden als negative Kräfte des projizierten Knotens in X-Richtung im Pixel aufsummiert
24	negatives Moment um Y	Ja	Die umgerechneten Momente werden als negative Kräfte des projizierten Knotens in Y-Richtung im Pixel aufsummiert
25	negatives Moment um Z	Ja	Die umgerechneten Momente werden als negative Kräfte des projizierten Knotens in Z-Richtung im Pixel aufsummiert
26	positiver Druck (normal)	Ja	Der umgerechnete positive Druck (normal) des projizierten Knotens wird im Pixel aufsummiert
27	negativer Druck (normal)	Ja	Der umgerechnete negative Druck (normal) des projizierten Knotens wird im Pixel aufsummiert
28	positive Verschiebung in X	Ja	Die positiven Verschiebungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
29	positive Verschiebung in Y	Ja	Die positiven Verschiebungen der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert
30	positive Verschiebung in Z	Ja	Die positiven Verschiebungen der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert
31	negative Verschiebung in X	Ja	Die negativen Verschiebungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
32	negative Verschiebung in Y	Ja	Die negativen Verschiebungen der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert

Tabelle 33: Erzeugbare Matrizen aus strukturmechanischen FE-Simulationen, Teil 4

Nr.	abzubildende FE-Größe	Betrachtung von Zwischenknoten	Wertebereich
33	negative Verschiebung in Z	Ja	Die negativen Verschiebungen der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert
34	positive Normalspannung in X	Ja	Die positiven Normalspannungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
35	positive Normalspannung in Y	Ja	Die positiven Normalspannungen der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert
36	positive Normalspannung in Z	Ja	Die positiven Normalspannungen der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert
37	negative Normalspannung in X	Ja	Die negativen Normalspannungen der projizierten Knoten in X-Richtung werden im Pixel aufsummiert
38	negative Normalspannung in Y	Ja	Die negativen Normalspannungen der projizierten Knoten in Y-Richtung werden im Pixel aufsummiert
39	negative Normalspannung in Z	Ja	Die negativen Normalspannungen der projizierten Knoten in Z-Richtung werden im Pixel aufsummiert
40	positive maximale Hauptspannung	Ja	Die positiven maximalen Hauptspannungen der projizierten Knoten werden im Pixel aufsummiert
41	negative maximale Hauptspannung	Ja	Die negativen maximalen Hauptspannungen der projizierten Knoten werden im Pixel aufsummiert
42	positive minimale Hauptspannung	Ja	Die positiven minimalen Hauptspannungen der projizierten Knoten werden im Pixel aufsummiert
43	negative minimale Hauptspannung	Ja	Die negativen minimalen Hauptspannungen der projizierten Knoten werden im Pixel aufsummiert
44	von Mises Vergleichsspannung	Ja	Die Werte der von Mises Vergleichsspannungen der projizierten Knoten werden im Pixel aufsummiert

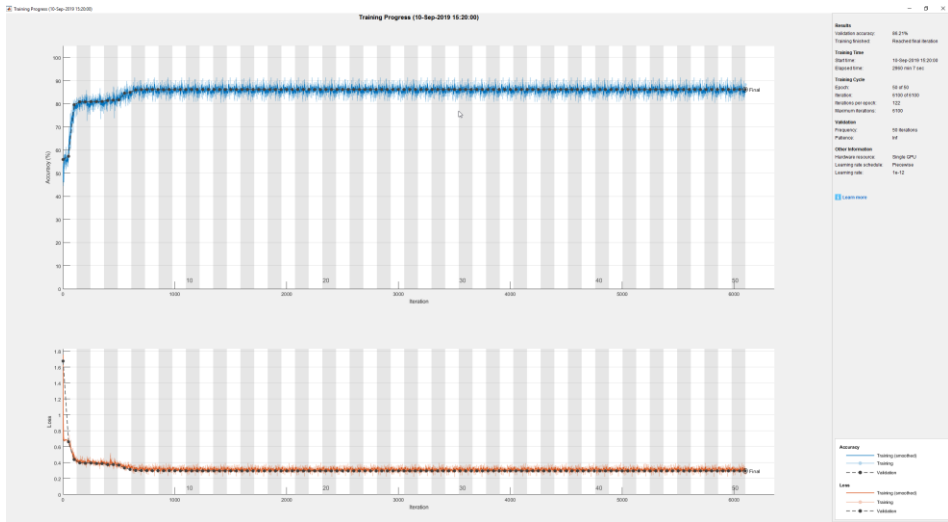


Bild 69: Trainingsprozess CNN 100x100 Normierungsstrategie 1

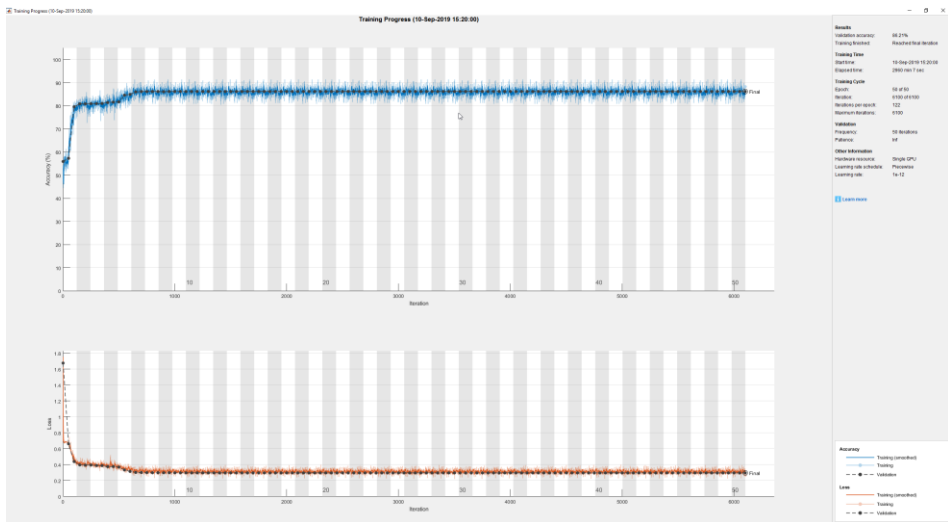


Bild 70: Trainingsprozess CNN 100x100 Normierungsstrategie 2

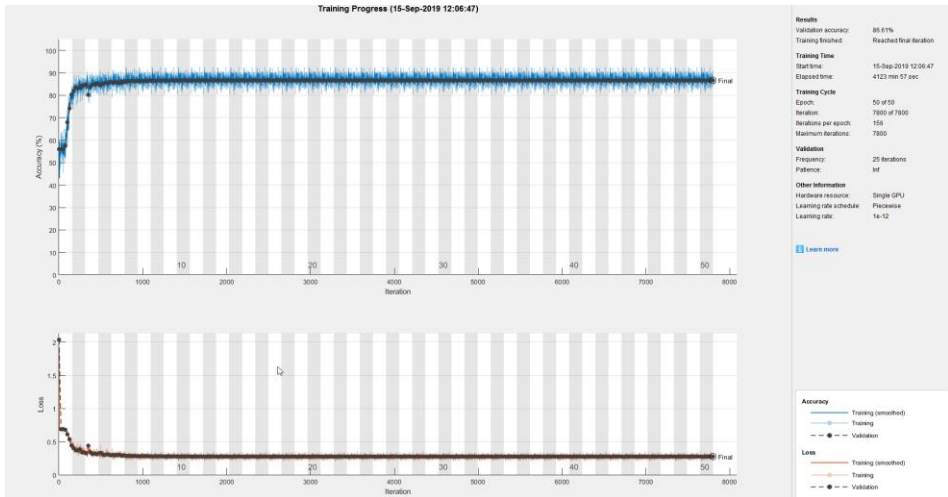


Bild 71: Trainingsprozess CNN 11x112 Normierungsstrategie 1

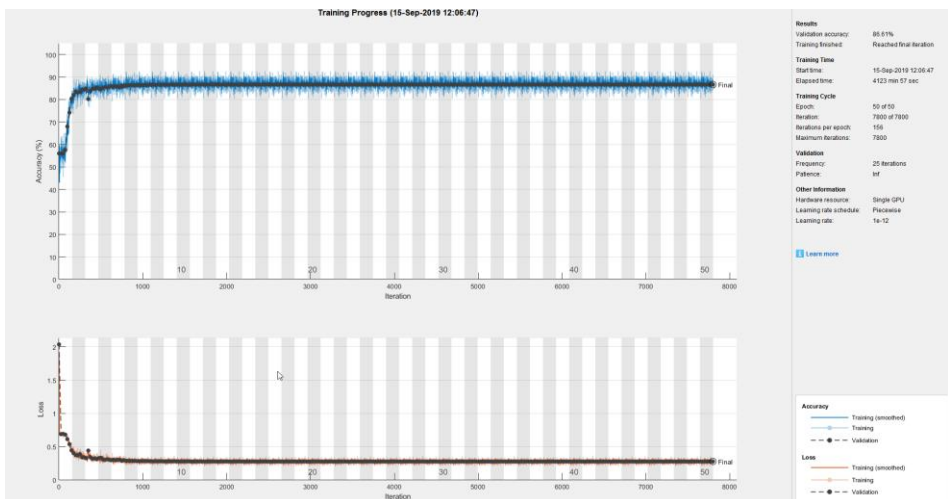


Bild 72: Trainingsprozess CNN 11x112 Normierungsstrategie 2

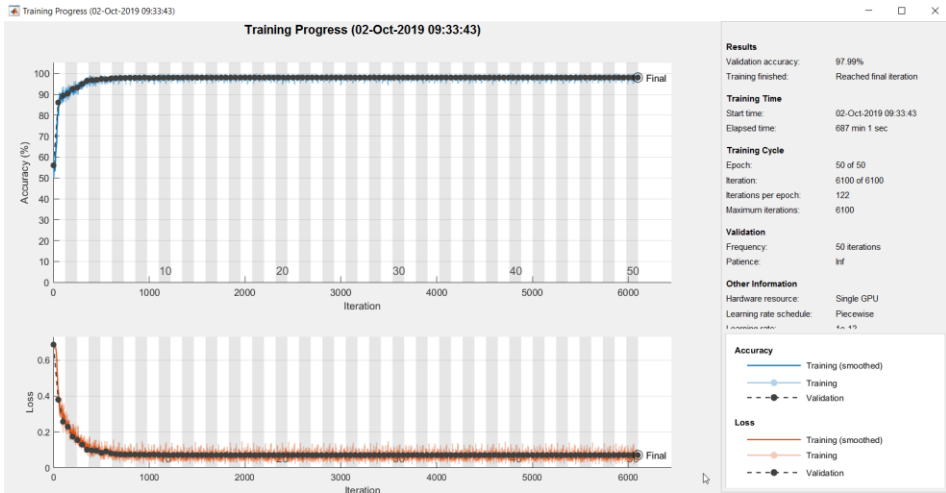


Bild 73: Trainingsprozess CNN<sub>2</sub> 2.000x100

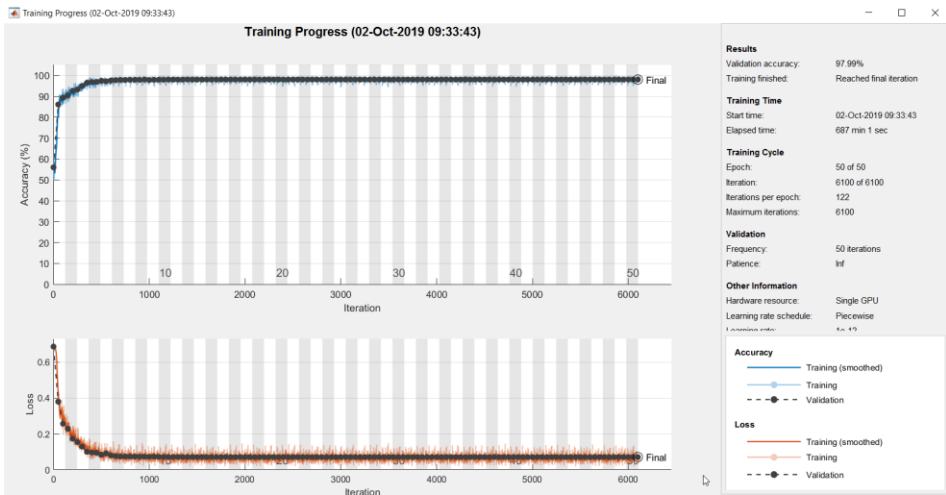


Bild 74: Trainingsprozess CNN<sub>3</sub> 2.000x100

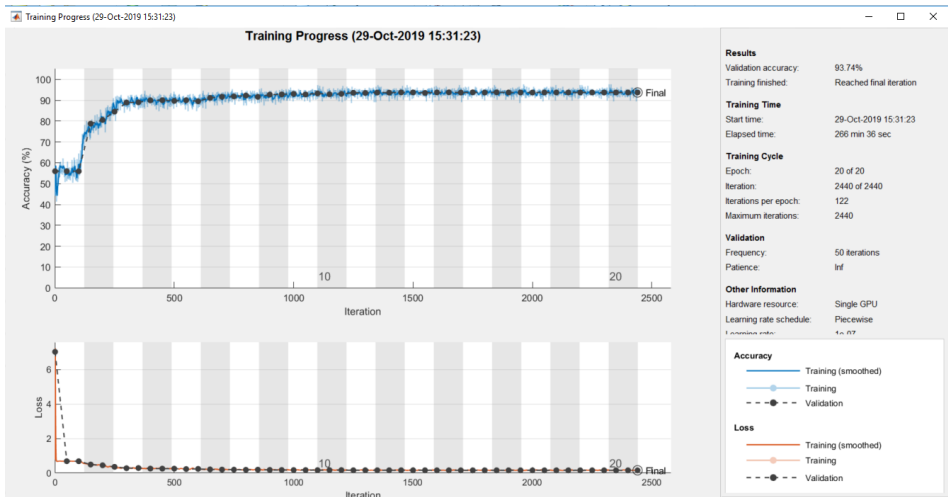


Bild 75: Trainingsprozess AlexNet Transfer Learning (100 % Trainingsdaten)

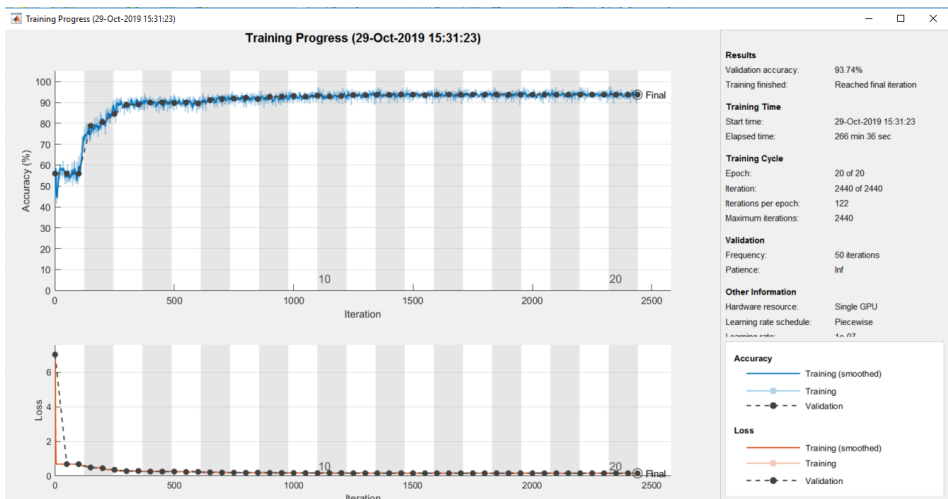


Bild 76: Trainingsprozess AlexNet Transfer Learning (90 % Trainingsdaten)

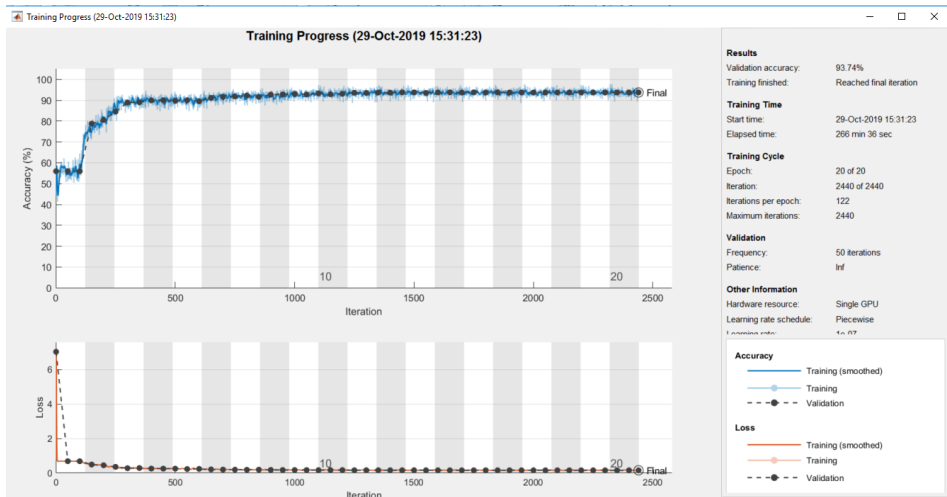


Bild 77: Trainingsprozess AlexNet Transfer Learning (80 % Trainingsdaten)

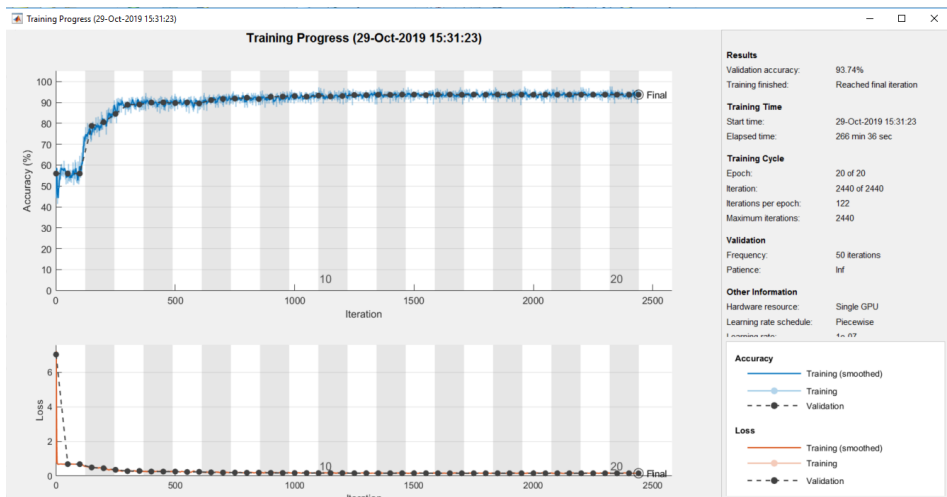


Bild 78: Trainingsprozess AlexNet Transfer Learning (70 % Trainingsdaten)

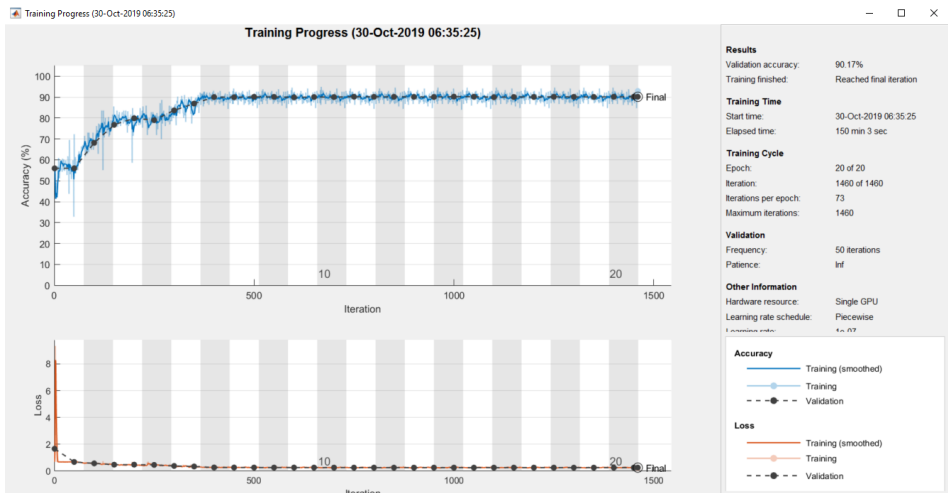


Bild 79: Trainingsprozess AlexNet Transfer Learning (60 % Trainingsdaten)

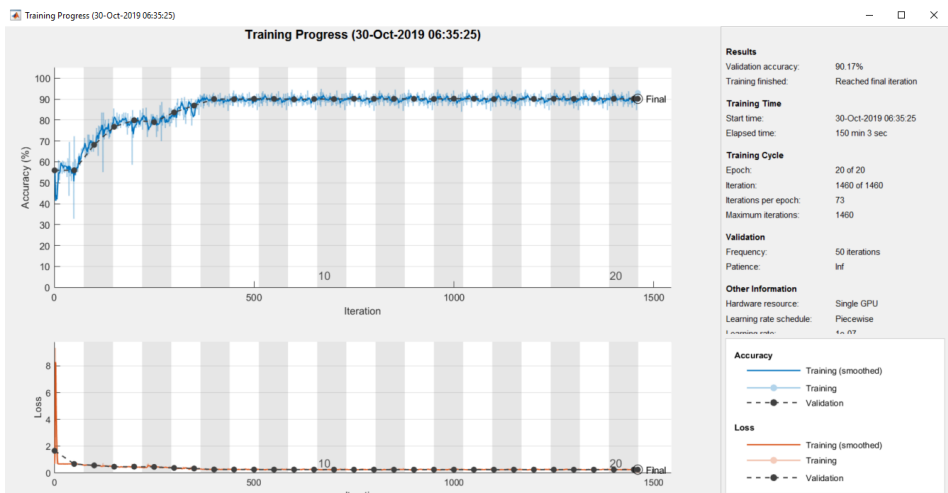


Bild 80: Trainingsprozess AlexNet Transfer Learning (50 % Trainingsdaten)

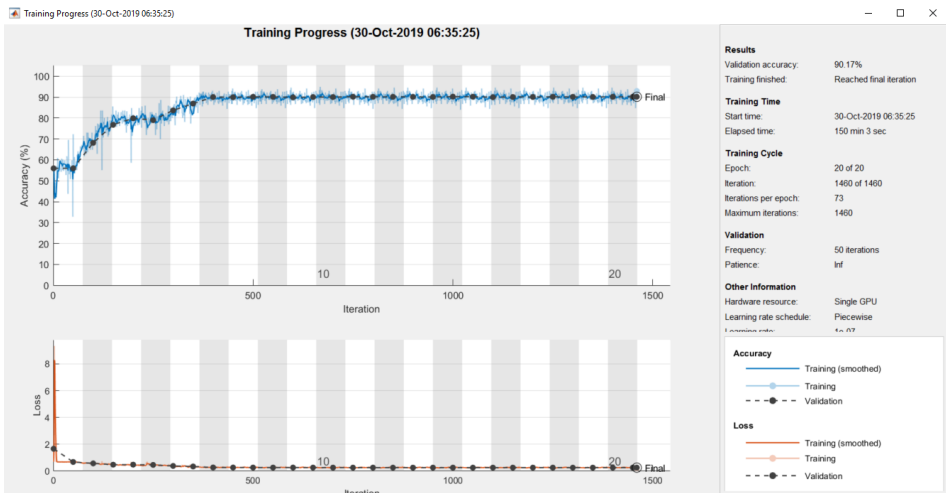


Bild 81: Trainingsprozess AlexNet Transfer Learning (40 % Trainingsdaten)

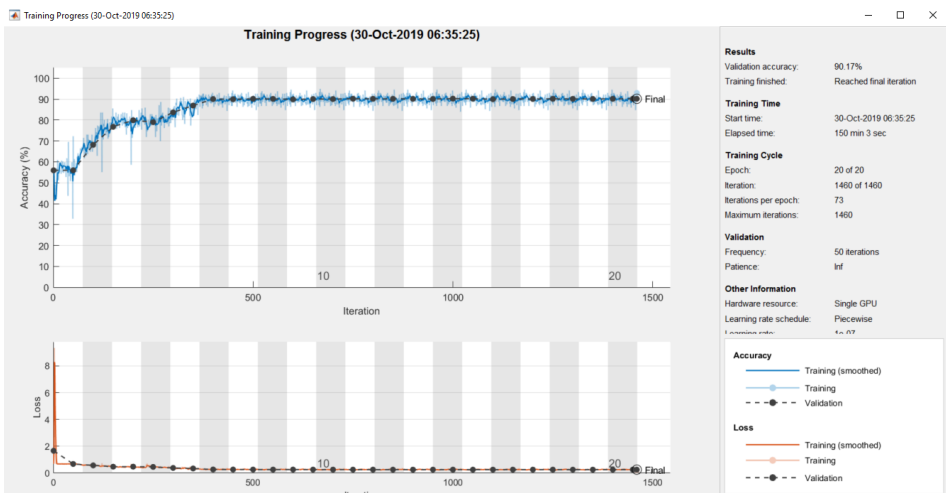


Bild 82: Trainingsprozess AlexNet Transfer Learning (30 % Trainingsdaten)

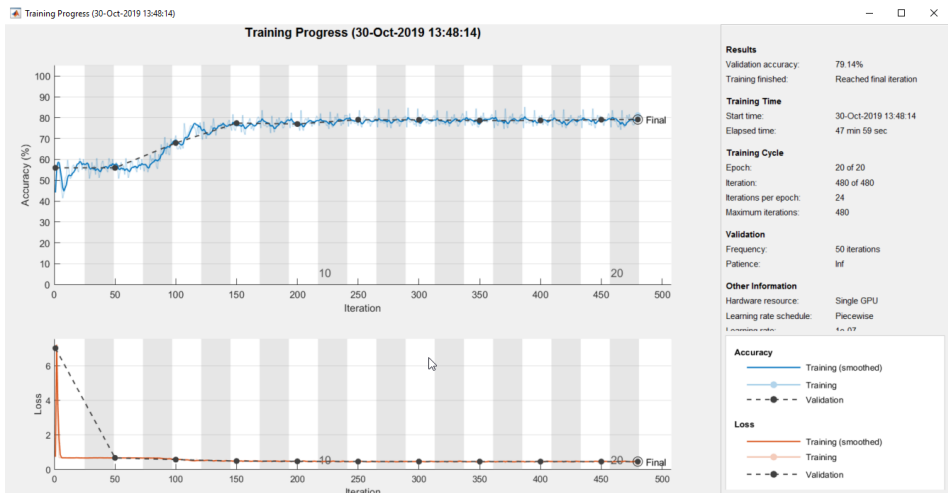


Bild 83: Trainingsprozess AlexNet Transfer Learning (20 % Trainingsdaten)

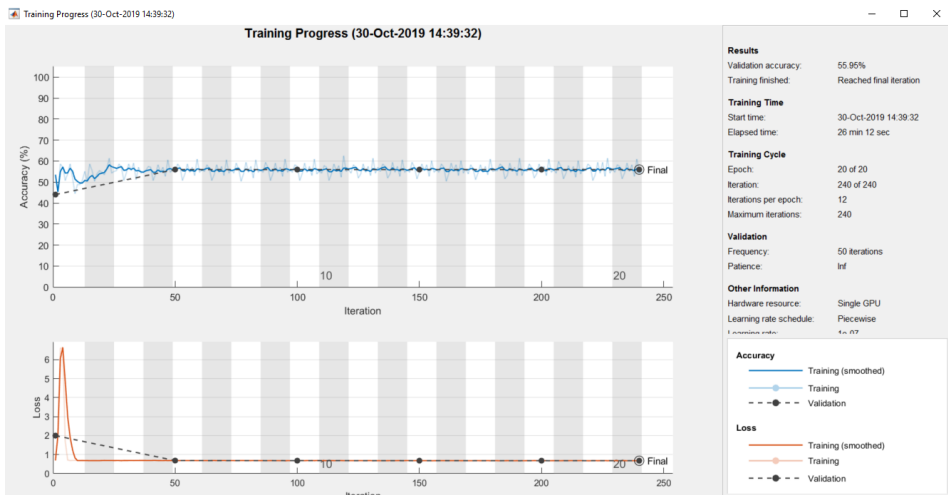


Bild 84: Trainingsprozess AlexNet Transfer Learning (10 % Trainingsdaten)



## Reihenübersicht

Koordination der Reihe (Stand 2021):  
Geschäftsstelle Maschinenbau, Dr.-Ing. Oliver Kreis, [www.mb.fau.de/diss/](http://www.mb.fau.de/diss/)

Im Rahmen der Reihe sind bisher die nachfolgenden Bände erschienen.

Band 1 – 52  
Fertigungstechnik – Erlangen  
ISSN 1431-6226  
Carl Hanser Verlag, München

Band 53 – 307  
Fertigungstechnik – Erlangen  
ISSN 1431-6226  
Meisenbach Verlag, Bamberg

ab Band 308  
FAU Studien aus dem Maschinenbau  
ISSN 2625-9974  
FAU University Press, Erlangen

Die Zugehörigkeit zu den jeweiligen Lehrstühlen ist wie folgt gekennzeichnet:

Lehrstühle:

<b>FAPS</b>	Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik
<b>FMT</b>	Lehrstuhl für Fertigungsmesstechnik
<b>KTmfk</b>	Lehrstuhl für Konstruktionstechnik
<b>LFT</b>	Lehrstuhl für Fertigungstechnologie
<b>LPT</b>	Lehrstuhl für Photonische Technologien
<b>REP</b>	Lehrstuhl für Ressourcen- und Energieeffiziente Produktionsmaschinen

Band 1: Andreas Hemberger  
Innovationspotentiale in der  
rechnerintegrierten Produktion durch  
wissensbasierte Systeme  
FAPS, 208 Seiten, 107 Bilder. 1988.  
ISBN 3-446-15234-2.

Band 2: Detlef Classe  
Beitrag zur Steigerung der Flexibilität  
automatisierter Montagesysteme  
durch Sensorintegration und erweiterte  
Steuerungskonzepte  
FAPS, 194 Seiten, 70 Bilder. 1988.  
ISBN 3-446-15529-5.

Band 3: Friedrich-Wilhelm Nolting  
Projektierung von Montagesystemen  
FAPS, 201 Seiten, 107 Bilder, 1 Tab. 1989.  
ISBN 3-446-15541-4.

Band 4: Karsten Schlüter  
Nutzungsgradsteigerung von  
Montagesystemen durch den Einsatz  
der Simulationstechnik  
FAPS, 177 Seiten, 97 Bilder. 1989.  
ISBN 3-446-15542-2.

Band 5: Shir-Kuan Lin  
Aufbau von Modellen zur Lageregelung  
von Industrierobotern  
FAPS, 168 Seiten, 46 Bilder. 1989.  
ISBN 3-446-15546-5.

Band 6: Rudolf Nuss  
Untersuchungen zur Bearbeitungsquali-  
tät im Fertigungssystem Laserstrahl-  
schneiden  
LFT, 206 Seiten, 115 Bilder, 6 Tab. 1989.  
ISBN 3-446-15783-2.

Band 7: Wolfgang Scholz  
Modell zur datenbankgestützten Planung  
automatisierter Montageanlagen  
FAPS, 194 Seiten, 89 Bilder. 1989.  
ISBN 3-446-15825-1.

Band 8: Hans-Jürgen Wißmeier  
Beitrag zur Beurteilung des Bruchverhal-  
tens von Hartmetall-Fließpreßmatrizen  
LFT, 179 Seiten, 99 Bilder, 9 Tab. 1989.  
ISBN 3-446-15921-5.

Band 9: Rainer Eisele  
Konzeption und Wirtschaftlichkeit von  
Planungssystemen in der Produktion  
FAPS, 183 Seiten, 86 Bilder. 1990.  
ISBN 3-446-16107-4.

Band 10: Rolf Pfeiffer  
Technologisch orientierte  
Montageplanung am Beispiel der  
Schraubtechnik  
FAPS, 216 Seiten, 102 Bilder, 16 Tab. 1990.  
ISBN 3-446-16161-9.

Band 11: Herbert Fischer  
Verteilte Planungssysteme zur  
Flexibilitätssteigerung der  
rechnerintegrierten Teilefertigung  
FAPS, 201 Seiten, 82 Bilder. 1990.  
ISBN 3-446-16105-8.

Band 12: Gerhard Kleineidam  
CAD/CAP: Rechnergestützte Montage-  
feinplanung  
FAPS, 203 Seiten, 107 Bilder. 1990.  
ISBN 3-446-16112-0.

Band 13: Frank Vollertsen  
Pulvermetallurgische Verarbeitung eines  
übereutektoiden verschleißfesten Stahls  
LFT, XIII u. 217 Seiten, 67 Bilder, 34 Tab.  
1990. ISBN 3-446-16133-3.

Band 14: Stephan Biermann  
Untersuchungen zur Anlagen- und  
Prozeßdiagnostik für das Schneiden  
mit CO<sub>2</sub>-Hochleistungslasern  
LFT, VIII u. 170 Seiten, 93 Bilder, 4 Tab.  
1991. ISBN 3-446-16269-0.

Band 15: Uwe Geißler  
Material- und Datenfluß in einer flexiblen  
Blechbearbeitungszelle  
LFT, 124 Seiten, 41 Bilder, 7 Tab. 1991.  
ISBN 3-446-16358-1.

Band 16: Frank Oswald Hake  
Entwicklung eines rechnergestützten  
Diagnosesystems für automatisierte  
Montagezellen  
FAPS, XIV u. 166 Seiten, 77 Bilder. 1991.  
ISBN 3-446-16428-6.

Band 17: Herbert Reichel  
Optimierung der Werkzeugbereitstellung  
durch rechnergestützte  
Arbeitsfolgenbestimmung  
FAPS, 198 Seiten, 73 Bilder, 2 Tab. 1991.  
ISBN 3-446-16453-7.

Band 18: Josef Scheller  
Modellierung und Einsatz von  
Softwaresystemen für rechnergeführte  
Montagezellen  
FAPS, 198 Seiten, 65 Bilder. 1991.  
ISBN 3-446-16454-5.

Band 19: Arnold vom Ende  
Untersuchungen zum Biegeumforme mit  
elastischer Matrize  
LFT, 166 Seiten, 55 Bilder, 13 Tab. 1991.  
ISBN 3-446-16493-6.

Band 20: Joachim Schmid  
Beitrag zum automatisierten Bearbeiten  
von Keramikguß mit Industrierobotern  
FAPS, XIV u. 176 Seiten, 111 Bilder, 6 Tab.  
1991. ISBN 3-446-16560-6.

Band 21: Egon Sommer  
Multiprozessorsteuerung für  
kooperierende Industrieroboter in  
Montagezellen  
FAPS, 188 Seiten, 102 Bilder. 1991.  
ISBN 3-446-17062-6.

Band 22: Georg Geyer  
Entwicklung problemspezifischer  
Verfahrensketten in der Montage  
FAPS, 192 Seiten, 112 Bilder. 1991.  
ISBN 3-446-16552-5.

Band 23: Rainer Flohr  
Beitrag zur optimalen  
Verbindungstechnik in der  
Oberflächenmontage (SMT)  
FAPS, 186 Seiten, 79 Bilder. 1991.  
ISBN 3-446-16568-1.

Band 24: Alfons Rief  
Untersuchungen zur Verfahrensfolge  
Laserstrahlschneiden und -schweißen  
in der Rohkarosseriefertigung  
LFT, VI u. 145 Seiten, 58 Bilder, 5 Tab.  
1991. ISBN 3-446-16593-2.

Band 25: Christoph Thim  
Rechnerunterstützte Optimierung  
von Materialflußstrukturen in der  
Elektronikmontage durch Simulation  
FAPS, 188 Seiten, 74 Bilder. 1992.  
ISBN 3-446-17118-5.

Band 26: Roland Müller  
CO<sub>2</sub>-Laserstrahlschneiden von  
kurzglasverstärkten Verbundwerkstoffen  
LFT, 141 Seiten, 107 Bilder, 4 Tab. 1992.  
ISBN 3-446-17104-5.

Band 27: Günther Schäfer  
Integrierte Informationsverarbeitung  
bei der Montageplanung  
FAPS, 195 Seiten, 76 Bilder. 1992.  
ISBN 3-446-17117-7.

Band 28: Martin Hoffmann  
Entwicklung einer  
CAD/CAM-Prozeßkette für die  
Herstellung von Blechbiegeteilen  
LFT, 149 Seiten, 89 Bilder. 1992.  
ISBN 3-446-17154-1.

Band 29: Peter Hoffmann  
Verfahrensfolge Laserstrahlschneiden  
und -schweißen: Prozeßführung und  
Systemtechnik in der 3D-Laserstrahlbear-  
beitung von Blechformteilen  
LFT, 186 Seiten, 92 Bilder, 10 Tab. 1992.  
ISBN 3-446-17153-3.

Band 30: Olaf Schrödel  
Flexible Werkstattsteuerung mit  
objektorientierten Softwarestrukturen  
FAPS, 180 Seiten, 84 Bilder. 1992.  
ISBN 3-446-17242-4.

Band 31: Hubert Reinisch  
Planungs- und Steuerungswerkzeuge zur  
impliziten Geräteprogrammierung  
in Roboterzellen  
FAPS, XI u. 212 Seiten, 112 Bilder. 1992.  
ISBN 3-446-17380-3.

Band 32: Brigitte Bärnreuther  
Ein Beitrag zur Bewertung des Kommuni-  
kationsverhaltens von Automatisierungs-  
geräten in flexiblen Produktionszellen  
FAPS, XI u. 179 Seiten, 71 Bilder. 1992.  
ISBN 3-446-17451-6.

Band 33: Joachim Hutfless  
Laserstrahlregelung und Optikdiagnostik  
in der Strahlführung einer  
CO<sub>2</sub>-Hochleistungslaseranlage  
LFT, 175 Seiten, 70 Bilder, 17 Tab. 1993.  
ISBN 3-446-17532-6.

Band 34: Uwe Günzel  
Entwicklung und Einsatz eines Simula-  
tionsverfahrens für operative und  
strategische Probleme der  
Produktionsplanung und -steuerung  
FAPS, XIV u. 170 Seiten, 66 Bilder, 5 Tab.  
1993. ISBN 3-446-17604-7.

Band 35: Bertram Ehmann  
Operatives Fertigungscontrolling durch  
Optimierung auftragsbezogener Bearbei-  
tungsabläufe in der Elektronikfertigung  
FAPS, XV u. 167 Seiten, 114 Bilder. 1993.  
ISBN 3-446-17658-6.

Band 36: Harald Kolléra  
Entwicklung eines benutzerorientierten  
Werkstattprogrammiersystems für das  
Laserstrahlschneiden  
LFT, 129 Seiten, 66 Bilder, 1 Tab. 1993.  
ISBN 3-446-17719-1.

Band 37: Stephanie Abels  
Modellierung und Optimierung von  
Montageanlagen in einem integrierten  
Simulationssystem  
FAPS, 188 Seiten, 88 Bilder. 1993.  
ISBN 3-446-17731-0.

Band 38: Robert Schmidt-Hebbel  
Laserstrahlbohren durchflußbestimmen-  
der Durchgangslöcher  
LFT, 145 Seiten, 63 Bilder, 11 Tab. 1993.  
ISBN 3-446-17778-7.

Band 39: Norbert Lutz  
Oberflächenfeinbearbeitung  
keramischer Werkstoffe mit  
XeCl-Excimerlaserstrahlung  
LFT, 187 Seiten, 98 Bilder, 29 Tab. 1994.  
ISBN 3-446-17970-4.

Band 40: Konrad Grampp  
Rechnerunterstützung bei Test und  
Schulung an Steuerungssoftware von  
SMD-Bestücklinien  
FAPS, 178 Seiten, 88 Bilder. 1995.  
ISBN 3-446-18173-3.

Band 41: Martin Koch  
Wissensbasierte Unterstützung der  
Angebotsbearbeitung in der  
Investitionsgüterindustrie  
FAPS, 169 Seiten, 68 Bilder. 1995.  
ISBN 3-446-18174-1.

Band 42: Armin Gropp  
Anlagen- und Prozeßdiagnostik beim  
Schneiden mit einem gepulsten  
Nd:YAG-Laser  
LFT, 160 Seiten, 88 Bilder, 7 Tab. 1995.  
ISBN 3-446-18241-1.

Band 43: Werner Heckel  
Optische 3D-Konturerfassung und  
on-line Biegewinkelmessung mit  
dem Lichtschnittverfahren  
LFT, 149 Seiten, 43 Bilder, 11 Tab. 1995.  
ISBN 3-446-18243-8.

Band 44: Armin Rothhaupt  
Modulares Planungssystem zur  
Optimierung der Elektronikfertigung  
FAPS, 180 Seiten, 101 Bilder. 1995.  
ISBN 3-446-18307-8.

Band 45: Bernd Zöllner  
Adaptive Diagnose in der  
Elektronikproduktion  
FAPS, 195 Seiten, 74 Bilder, 3 Tab. 1995.  
ISBN 3-446-18308-6.

Band 46: Bodo Vormann  
Beitrag zur automatisierten  
Handhabungsplanung komplexer  
Blechbiegeteile  
LFT, 126 Seiten, 89 Bilder, 3 Tab. 1995.  
ISBN 3-446-18345-0.

Band 47: Peter Schnepf  
Zielkostenorientierte Montageplanung  
FAPS, 144 Seiten, 75 Bilder. 1995.  
ISBN 3-446-18397-3.

Band 48: Rainer Klotzbücher  
Konzept zur rechnerintegrierten  
Materialversorgung in flexiblen  
Fertigungssystemen  
FAPS, 156 Seiten, 62 Bilder. 1995.  
ISBN 3-446-18412-0.

Band 49: Wolfgang Greska  
Wissensbasierte Analyse und  
Klassifizierung von Blechteilen  
LFT, 144 Seiten, 96 Bilder. 1995.  
ISBN 3-446-18462-7.

Band 50: Jörg Franke  
Integrierte Entwicklung neuer  
Produkt- und Produktionstechnologien  
für räumliche spritzgegossene  
Schaltungsträger (3-D MID)  
FAPS, 196 Seiten, 86 Bilder, 4 Tab. 1995.  
ISBN 3-446-18448-1.

Band 51: Franz-Josef Zeller  
Sensorplanung und schnelle  
Sensorregelung für Industrieroboter  
FAPS, 190 Seiten, 102 Bilder, 9 Tab. 1995.  
ISBN 3-446-18601-8.

Band 52: Michael Solvie  
Zeitbehandlung und  
Multimedia-Unterstützung in  
Feldkommunikationssystemen  
FAPS, 200 Seiten, 87 Bilder, 35 Tab. 1996.  
ISBN 3-446-18607-7.

Band 53: Robert Hopperdietzel  
Reengineering in der Elektro- und  
Elektronikindustrie  
FAPS, 180 Seiten, 109 Bilder, 1 Tab. 1996.  
ISBN 3-87525-070-2.

Band 54: Thomas Rebhahn  
Beitrag zur Mikromaterialbearbeitung  
mit Excimerlasern - Systemkomponenten  
und Verfahrensoptimierungen  
LFT, 148 Seiten, 61 Bilder, 10 Tab. 1996.  
ISBN 3-87525-075-3.

Band 55: Henning Hanebuth  
Laserstrahlhartlöten mit  
Zweistrahlschweißtechnik  
LFT, 157 Seiten, 58 Bilder, 11 Tab. 1996.  
ISBN 3-87525-074-5.

Band 56: Uwe Schönherr  
Steuerung und Sensordatenintegration  
für flexible Fertigungszellen mit  
kooperierenden Robotern  
FAPS, 188 Seiten, 116 Bilder, 3 Tab. 1996.  
ISBN 3-87525-076-1.

Band 57: Stefan Holzer  
Berührungslose Formgebung mit  
Laserstrahlung  
LFT, 162 Seiten, 69 Bilder, 11 Tab. 1996.  
ISBN 3-87525-079-6.

Band 58: Markus Schultz  
Fertigungsqualität beim  
3D-Laserstrahlschweißen von  
Blechformteilen  
LFT, 165 Seiten, 88 Bilder, 9 Tab. 1997.  
ISBN 3-87525-080-X.

Band 59: Thomas Krebs  
Integration elektromechanischer  
CA-Anwendungen über einem  
STEP-Produktmodell  
FAPS, 198 Seiten, 58 Bilder, 8 Tab. 1997.  
ISBN 3-87525-081-8.

Band 60: Jürgen Sturm  
Prozeßintegrierte Qualitätssicherung  
in der Elektronikproduktion  
FAPS, 167 Seiten, 112 Bilder, 5 Tab. 1997.  
ISBN 3-87525-082-6.

Band 61: Andreas Brand  
Prozesse und Systeme zur Bestückung  
räumlicher elektronischer Baugruppen  
(3D-MID)  
FAPS, 182 Seiten, 100 Bilder. 1997.  
ISBN 3-87525-087-7.

Band 62: Michael Kauf  
Regelung der Laserstrahlleistung und  
der Fokusparameter einer  
CO<sub>2</sub>-Hochleistungslaseranlage  
LFT, 140 Seiten, 70 Bilder, 5 Tab. 1997.  
ISBN 3-87525-083-4.

Band 63: Peter Steinwasser  
Modulares Informationsmanagement  
in der integrierten Produkt- und  
Prozeßplanung  
FAPS, 190 Seiten, 87 Bilder. 1997.  
ISBN 3-87525-084-2.

Band 64: Georg Liedl  
Integriertes Automatisierungskonzept  
für den flexiblen Materialfluß in der  
Elektronikproduktion  
FAPS, 196 Seiten, 96 Bilder, 3 Tab. 1997.  
ISBN 3-87525-086-9.

Band 65: Andreas Otto  
Transiente Prozesse beim  
Laserstrahlschweißen  
LFT, 132 Seiten, 62 Bilder, 1 Tab. 1997.  
ISBN 3-87525-089-3.

Band 66: Wolfgang Blöchl  
Erweiterte Informationsbereitstellung  
an offenen CNC-Steuerungen zur  
Prozeß- und Programmoptimierung  
FAPS, 168 Seiten, 96 Bilder. 1997.  
ISBN 3-87525-091-5.

Band 67: Klaus-Uwe Wolf  
Verbesserte Prozeßführung und  
Prozeßplanung zur Leistungs- und  
Qualitätssteigerung beim  
Spulenwickeln  
FAPS, 186 Seiten, 125 Bilder. 1997.  
ISBN 3-87525-092-3.

Band 68: Frank Backes  
Technologieorientierte Bahnplanung  
für die 3D-Laserstrahlbearbeitung  
LFT, 138 Seiten, 71 Bilder, 2 Tab. 1997.  
ISBN 3-87525-093-1.

Band 69: Jürgen Kraus  
Laserstrahlumformen von Profilen  
LFT, 137 Seiten, 72 Bilder, 8 Tab. 1997.  
ISBN 3-87525-094-X.

Band 70: Norbert Neubauer  
Adaptive Strahlführungen für  
CO<sub>2</sub>-Laseranlagen  
LFT, 120 Seiten, 50 Bilder, 3 Tab. 1997.  
ISBN 3-87525-095-8.

Band 71: Michael Steber  
Prozeßoptimierter Betrieb flexibler  
Schraubstationen in der  
automatisierten Montage  
FAPS, 168 Seiten, 78 Bilder, 3 Tab. 1997.  
ISBN 3-87525-096-6.

Band 72: Markus Pfestorf  
Funktionale 3D-Oberflächenkenngrößen  
in der Umformtechnik  
LFT, 162 Seiten, 84 Bilder, 15 Tab. 1997.  
ISBN 3-87525-097-4.

Band 73: Volker Franke  
Integrierte Planung und Konstruktion  
von Werkzeugen für die Biegebearbeitung  
LFT, 143 Seiten, 81 Bilder. 1998.  
ISBN 3-87525-098-2.

Band 74: Herbert Scheller  
Automatisierte Demontagesysteme  
und recyclinggerechte Produktgestaltung  
elektronischer Baugruppen  
FAPS, 184 Seiten, 104 Bilder, 17 Tab. 1998.  
ISBN 3-87525-099-0.

Band 75: Arthur Meißner  
Kaltmassivumformung metallischer  
Kleinstteile – Werkstoffverhalten,  
Wirkflächenreibung, Prozeßauslegung  
LFT, 164 Seiten, 92 Bilder, 14 Tab. 1998.  
ISBN 3-87525-100-8.

Band 76: Mathias Glasmacher  
Prozeß- und Systemtechnik zum  
Laserstrahl-Mikroschweißen  
LFT, 184 Seiten, 104 Bilder, 12 Tab. 1998.  
ISBN 3-87525-101-6.

Band 77: Michael Schwind  
Zerstörungsfreie Ermittlung mechani-  
scher Eigenschaften von Feinblechen mit  
dem Wirbelstromverfahren  
LFT, 124 Seiten, 68 Bilder, 8 Tab. 1998.  
ISBN 3-87525-102-4.

Band 78: Manfred Gerhard  
Qualitätssteigerung in der  
Elektronikproduktion durch  
Optimierung der Prozeßführung  
beim Löten komplexer Baugruppen  
FAPS, 179 Seiten, 113 Bilder, 7 Tab. 1998.  
ISBN 3-87525-103-2.

Band 79: Elke Rauh  
Methodische Einbindung der Simulation  
in die betrieblichen Planungs- und  
Entscheidungsabläufe  
FAPS, 192 Seiten, 114 Bilder, 4 Tab. 1998.  
ISBN 3-87525-104-0.

Band 80: Sorin Niederkorn  
Meßeinrichtung zur Untersuchung  
der Wirkflächenreibung bei umformtech-  
nischen Prozessen  
LFT, 99 Seiten, 46 Bilder, 6 Tab. 1998.  
ISBN 3-87525-105-9.

Band 81: Stefan Schuberth  
Regelung der Fokusslage beim Schweißen  
mit CO<sub>2</sub>-Hochleistungslasern unter  
Einsatz von adaptiven Optiken  
LFT, 140 Seiten, 64 Bilder, 3 Tab. 1998.  
ISBN 3-87525-106-7.

Band 82: Armando Walter Colombo  
Development and Implementation of  
Hierarchical Control Structures of  
Flexible Production Systems Using High  
Level Petri Nets  
FAPS, 216 Seiten, 86 Bilder. 1998.  
ISBN 3-87525-109-1.

Band 83: Otto Meedt  
Effizienzsteigerung bei Demontage  
und Recycling durch flexible  
Demontagetechnologien und optimierte  
Produktgestaltung  
FAPS, 186 Seiten, 103 Bilder. 1998.  
ISBN 3-87525-108-3.

Band 84: Knuth Götz  
Modelle und effiziente Modellbildung  
zur Qualitätssicherung in der  
Elektronikproduktion  
FAPS, 212 Seiten, 129 Bilder, 24 Tab. 1998.  
ISBN 3-87525-112-1.

Band 85: Ralf Luchs  
Einsatzmöglichkeiten leitender Klebstoffe zur zuverlässigen Kontaktierung elektronischer Bauelemente in der SMT  
FAPS, 176 Seiten, 126 Bilder, 30 Tab. 1998.  
ISBN 3-87525-113-7.

Band 86: Frank Pöhlau  
Entscheidungsgrundlagen zur Einführung räumlicher spritzgegossener Schaltungsträger (3-D MID)  
FAPS, 144 Seiten, 99 Bilder. 1999.  
ISBN 3-87525-114-8.

Band 87: Roland T. A. Kals  
Fundamentals on the miniaturization of sheet metal working processes  
LFT, 128 Seiten, 58 Bilder, 11 Tab. 1999.  
ISBN 3-87525-115-6.

Band 88: Gerhard Luhn  
Implizites Wissen und technisches Handeln am Beispiel der Elektronikproduktion  
FAPS, 252 Seiten, 61 Bilder, 1 Tab. 1999.  
ISBN 3-87525-116-4.

Band 89: Axel Sprenger  
Adaptives Streckbiegen von Aluminium-Strangpreßprofilen  
LFT, 114 Seiten, 63 Bilder, 4 Tab. 1999.  
ISBN 3-87525-117-2.

Band 90: Hans-Jörg Pucher  
Untersuchungen zur Prozeßfolge Umformen, Bestücken und Laserstrahllöten von Mikrokontakten  
LFT, 158 Seiten, 69 Bilder, 9 Tab. 1999.  
ISBN 3-87525-119-9.

Band 91: Horst Arnet  
Profilbiegen mit kinematischer Gestalterzeugung  
LFT, 128 Seiten, 67 Bilder, 7 Tab. 1999.  
ISBN 3-87525-120-2.

Band 92: Doris Schubart  
Prozeßmodellierung und Technologieentwicklung beim Abtragen mit CO<sub>2</sub>-Laserstrahlung  
LFT, 133 Seiten, 57 Bilder, 13 Tab. 1999.  
ISBN 3-87525-122-9.

Band 93: Adrianus L. P. Coremans  
Laserstrahlsintern von Metallpulver - Prozeßmodellierung, Systemtechnik, Eigenschaften laserstrahlgesinterter Metallkörper  
LFT, 184 Seiten, 108 Bilder, 12 Tab. 1999.  
ISBN 3-87525-124-5.

Band 94: Hans-Martin Biehler  
Optimierungskonzepte für Qualitätsdatenverarbeitung und Informationsbereitstellung in der Elektronikfertigung  
FAPS, 194 Seiten, 105 Bilder. 1999.  
ISBN 3-87525-126-1.

Band 95: Wolfgang Becker  
Oberflächenausbildung und tribologische Eigenschaften excimerlaserstrahlbearbeiteter Hochleistungskeramiken  
LFT, 175 Seiten, 71 Bilder, 3 Tab. 1999.  
ISBN 3-87525-127-X.

Band 96: Philipp Hein  
Innenhochdruck-Umformen von Blechpaaren: Modellierung, Prozeßauslegung und Prozeßführung  
LFT, 129 Seiten, 57 Bilder, 7 Tab. 1999.  
ISBN 3-87525-128-8.

Band 97: Gunter Beitinger  
Herstellungs- und Prüfverfahren für  
thermoplastische Schaltungsträger  
FAPS, 169 Seiten, 92 Bilder, 20 Tab. 1999.  
ISBN 3-87525-129-6.

Band 98: Jürgen Knoblach  
Beitrag zur rechnerunterstützten  
verursachungsgerechten  
Angebotskalkulation von Blechteilen  
mit Hilfe wissenschaftlicher Methoden  
LFT, 155 Seiten, 53 Bilder, 26 Tab. 1999.  
ISBN 3-87525-130-X.

Band 99: Frank Breitenbach  
Bildverarbeitungssystem zur Erfassung  
der Anschlußgeometrie elektronischer  
SMT-Bauelemente  
LFT, 147 Seiten, 92 Bilder, 12 Tab. 2000.  
ISBN 3-87525-131-8.

Band 100: Bernd Falk  
Simulationsbasierte  
Lebensdauervorhersage für Werkzeuge  
der Kaltmassivumformung  
LFT, 134 Seiten, 44 Bilder, 15 Tab. 2000.  
ISBN 3-87525-136-9.

Band 101: Wolfgang Schlögl  
Integriertes Simulationsdaten-Management  
für Maschinenentwicklung und  
Anlagenplanung  
FAPS, 169 Seiten, 101 Bilder, 20 Tab. 2000.  
ISBN 3-87525-137-7.

Band 102: Christian Hinsel  
Ermüdungsbruchversagen  
hartstoffbeschichteter Werkzeugstähle  
in der Kaltmassivumformung  
LFT, 130 Seiten, 80 Bilder, 14 Tab. 2000.  
ISBN 3-87525-138-5.

Band 103: Stefan Bobbert  
Simulationsgestützte Prozessauslegung  
für das Innenhochdruck-Umformen  
von Blechpaaren  
LFT, 123 Seiten, 77 Bilder. 2000.  
ISBN 3-87525-145-8.

Band 104: Harald Rottbauer  
Modulares Planungswerkzeug zum  
Produktionsmanagement in der  
Elektronikproduktion  
FAPS, 166 Seiten, 106 Bilder. 2001.  
ISBN 3-87525-139-3.

Band 105: Thomas Hennige  
Flexible Formgebung von Blechen  
durch Laserstrahlumformen  
LFT, 119 Seiten, 50 Bilder. 2001.  
ISBN 3-87525-140-7.

Band 106: Thomas Menzel  
Wissenschaftliche Methoden für die  
rechnergestützte Charakterisierung  
und Bewertung innovativer  
Fertigungsprozesse  
LFT, 152 Seiten, 71 Bilder. 2001.  
ISBN 3-87525-142-3.

Band 107: Thomas Stöckel  
Kommunikationstechnische Integration  
der Prozessebene in Produktionssysteme  
durch Middleware-Frameworks  
FAPS, 147 Seiten, 65 Bilder, 5 Tab. 2001.  
ISBN 3-87525-143-1.

Band 108: Frank Pitter  
Verfügbarkeitssteigerung von  
Werkzeugmaschinen durch Einsatz  
mechatronischer Sensorlösungen  
FAPS, 158 Seiten, 131 Bilder, 8 Tab. 2001.  
ISBN 3-87525-144-X.

Band 109: Markus Korneli  
Integration lokaler CAP-Systeme in  
einen globalen Fertigungsdatenverbund  
FAPS, 121 Seiten, 53 Bilder, 11 Tab. 2001.  
ISBN 3-87525-146-6.

Band 110: Burkhard Müller  
Laserstrahljustieren mit Excimer-Lasern -  
Prozeßparameter und Modelle zur  
Aktorkonstruktion  
LFT, 128 Seiten, 36 Bilder, 9 Tab. 2001.  
ISBN 3-87525-159-8.

Band 111: Jürgen Göhringer  
Integrierte Telediagnose via Internet  
zum effizienten Service von  
Produktionssystemen  
FAPS, 178 Seiten, 98 Bilder, 5 Tab. 2001.  
ISBN 3-87525-147-4.

Band 112: Robert Feuerstein  
Qualitäts- und kosteneffiziente Integra-  
tion neuer Bauelementetechnologien in  
die Flachbaugruppenfertigung  
FAPS, 161 Seiten, 99 Bilder, 10 Tab. 2001.  
ISBN 3-87525-151-2.

Band 113: Marcus Reichenberger  
Eigenschaften und Einsatzmöglichkeiten  
alternativer Elektroniklote in der  
Oberflächenmontage (SMT)  
FAPS, 165 Seiten, 97 Bilder, 18 Tab. 2001.  
ISBN 3-87525-152-0.

Band 114: Alexander Huber  
Justieren vormontierter Systeme mit dem  
Nd:YAG-Laser unter Einsatz von Aktoren  
LFT, 122 Seiten, 58 Bilder, 5 Tab. 2001.  
ISBN 3-87525-153-9.

Band 115: Sami Krimi  
Analyse und Optimierung von Montage-  
systemen in der Elektronikproduktion  
FAPS, 155 Seiten, 88 Bilder, 3 Tab. 2001.  
ISBN 3-87525-157-1.

Band 116: Marion Merklein  
Laserstrahlumformen von  
Aluminiumwerkstoffen - Beeinflussung  
der Mikrostruktur und der mechanischen  
Eigenschaften  
LFT, 122 Seiten, 65 Bilder, 15 Tab. 2001.  
ISBN 3-87525-156-3.

Band 117: Thomas Collisi  
Ein informationslogistisches  
Architekturkonzept zur Akquisition  
simulationsrelevanter Daten  
FAPS, 181 Seiten, 105 Bilder, 7 Tab. 2002.  
ISBN 3-87525-164-4.

Band 118: Markus Koch  
Rationalisierung und ergonomische  
Optimierung im Innenausbau durch  
den Einsatz moderner  
Automatisierungstechnik  
FAPS, 176 Seiten, 98 Bilder, 9 Tab. 2002.  
ISBN 3-87525-165-2.

Band 119: Michael Schmidt  
Prozeßregelung für das Laserstrahl-  
Punktschweißen in der Elektronikpro-  
duktion  
LFT, 152 Seiten, 71 Bilder, 3 Tab. 2002.  
ISBN 3-87525-166-0.

Band 120: Nicolas Tiesler  
Grundlegende Untersuchungen zum  
Fließpressen metallischer Kleinstteile  
LFT, 126 Seiten, 78 Bilder, 12 Tab. 2002.  
ISBN 3-87525-175-X.

Band 121: Lars Pursche  
Methoden zur technologieorientierten  
Programmierung für die  
3D-Lasermikrobearbeitung  
LFT, 111 Seiten, 39 Bilder, 0 Tab. 2002.  
ISBN 3-87525-183-0.

Band 122: Jan-Oliver Brassel  
Prozeßkontrolle beim  
Laserstrahl-Mikroschweißen  
LFT, 148 Seiten, 72 Bilder, 12 Tab. 2002.  
ISBN 3-87525-181-4.

Band 123: Mark Geisel  
Prozeßkontrolle und -steuerung beim  
Laserstrahlschweißen mit den Methoden  
der nichtlinearen Dynamik  
LFT, 135 Seiten, 46 Bilder, 2 Tab. 2002.  
ISBN 3-87525-180-6.

Band 124: Gerd Eßler  
Laserstrahlunterstützte Erzeugung  
metallischer Leiterstrukturen auf  
Thermoplastsubstraten für die  
MID-Technik  
LFT, 148 Seiten, 60 Bilder, 6 Tab. 2002.  
ISBN 3-87525-171-7.

Band 125: Marc Fleckenstein  
Qualität laserstrahl-gefügter  
Mikroverbindungen elektronischer  
Kontakte  
LFT, 159 Seiten, 77 Bilder, 7 Tab. 2002.  
ISBN 3-87525-170-9.

Band 126: Stefan Kaufmann  
Grundlegende Untersuchungen zum  
Nd:YAG- Laserstrahlfügen von Silizium  
für Komponenten der Optoelektronik  
LFT, 159 Seiten, 100 Bilder, 6 Tab. 2002.  
ISBN 3-87525-172-5.

Band 127: Thomas Fröhlich  
Simultanes Löten von Anschlußkontak-  
ten elektronischer Bauelemente mit  
Diodenlaserstrahlung  
LFT, 143 Seiten, 75 Bilder, 6 Tab. 2002.  
ISBN 3-87525-186-5.

Band 128: Achim Hofmann  
Erweiterung der Formgebungsgrenzen  
beim Umformen von  
Aluminiumwerkstoffen durch den Ein-  
satz prozessangepasster Platinen  
LFT, 113 Seiten, 58 Bilder, 4 Tab. 2002.  
ISBN 3-87525-182-2.

Band 129: Ingo Kriebitzsch  
3 - D MID Technologie in der  
Automobilelektronik  
FAPS, 129 Seiten, 102 Bilder, 10 Tab. 2002.  
ISBN 3-87525-169-5.

Band 130: Thomas Pohl  
Fertigungsqualität und Umformbarkeit  
laserstrahlgeschweißter Formplatinen  
aus Aluminiumlegierungen  
LFT, 133 Seiten, 93 Bilder, 12 Tab. 2002.  
ISBN 3-87525-173-3.

Band 131: Matthias Wenk  
Entwicklung eines konfigurierbaren  
Steuerungssystems für die flexible  
Sensorführung von Industrierobotern  
FAPS, 167 Seiten, 85 Bilder, 1 Tab. 2002.  
ISBN 3-87525-174-1.

Band 132: Matthias Nегendandck  
Neue Sensorik und Aktorik für  
Bearbeitungsköpfe zum  
Laserstrahlschweißen  
LFT, 116 Seiten, 60 Bilder, 14 Tab. 2002.  
ISBN 3-87525-184-9.

Band 133: Oliver Kreis  
Integrierte Fertigung - Verfahrensin-  
tegration durch Innenhochdruck-Umfor-  
men, Trennen und Laserstrahlschweißen  
in einem Werkzeug sowie ihre tele- und  
multimediale Präsentation  
LFT, 167 Seiten, 90 Bilder, 43 Tab. 2002.  
ISBN 3-87525-176-8.

Band 134: Stefan Trautner  
Technische Umsetzung produktbezoge-  
ner Instrumente der Umweltpolitik bei  
Elektro- und Elektronikgeräten  
FAPS, 179 Seiten, 92 Bilder, 11 Tab. 2002.  
ISBN 3-87525-177-6.

Band 135: Roland Meier  
Strategien für einen produktorientierten  
Einsatz räumlicher spritzgegossener  
Schaltungsträger (3-D MID)  
FAPS, 155 Seiten, 88 Bilder, 14 Tab. 2002.  
ISBN 3-87525-178-4.

Band 136: Jürgen Wunderlich  
Kostensimulation - Simulationsbasierte  
Wirtschaftlichkeitsregelung komplexer  
Produktionssysteme  
FAPS, 202 Seiten, 119 Bilder, 17 Tab. 2002.  
ISBN 3-87525-179-2.

Band 137: Stefan Novotny  
Innenhochdruck-Umformen von Blechen  
aus Aluminium- und Magnesiumlegie-  
rungen bei erhöhter Temperatur  
LFT, 132 Seiten, 82 Bilder, 6 Tab. 2002.  
ISBN 3-87525-185-7.

Band 138: Andreas Licha  
Flexible Montageautomatisierung zur  
Komplettmontage flächenhafter Produkt-  
strukturen durch kooperierende  
Industrieroboter  
FAPS, 158 Seiten, 87 Bilder, 8 Tab. 2003.  
ISBN 3-87525-189-X.

Band 139: Michael Eisenbarth  
Beitrag zur Optimierung der Aufbau- und  
Verbindungstechnik für mechatronische  
Baugruppen  
FAPS, 207 Seiten, 141 Bilder, 9 Tab. 2003.  
ISBN 3-87525-190-3.

Band 140: Frank Christoph  
Durchgängige simulationsgestützte  
Planung von Fertigungseinrichtungen der  
Elektronikproduktion  
FAPS, 187 Seiten, 107 Bilder, 9 Tab. 2003.  
ISBN 3-87525-191-1.

Band 141: Hinnerk Hagenah  
Simulationsbasierte Bestimmung der  
zu erwartenden Maßhaltigkeit für das  
Blechbiegen  
LFT, 131 Seiten, 36 Bilder, 26 Tab. 2003.  
ISBN 3-87525-192-X.

Band 142: Ralf Eckstein  
Scherschneiden und Biegen metallischer  
Kleinstteile - Materialeinfluss und  
Materialverhalten  
LFT, 148 Seiten, 71 Bilder, 19 Tab. 2003.  
ISBN 3-87525-193-8.

Band 143: Frank H. Meyer-Pittroff  
Excimerlaserstrahlbiegen dünner  
metallischer Folien mit homogener  
Lichtlinie  
LFT, 138 Seiten, 60 Bilder, 16 Tab. 2003.  
ISBN 3-87525-196-2.

Band 144: Andreas Kach  
Rechnergestützte Anpassung von  
Laserstrahlschneidbahnen  
an Bauteilabweichungen  
LFT, 139 Seiten, 69 Bilder, 11 Tab. 2004.  
ISBN 3-87525-197-0.

Band 145: Stefan Hierl  
System- und Prozesstechnik für das  
simultane Lötten mit Diodenlaserstrah-  
lung von elektronischen Bauelementen  
LFT, 124 Seiten, 66 Bilder, 4 Tab. 2004.  
ISBN 3-87525-198-9.

Band 146: Thomas Neudecker  
Tribologische Eigenschaften keramischer  
Blechumformwerkzeuge- Einfluss einer  
Oberflächenendbearbeitung mittels  
Excimerlaserstrahlung  
LFT, 166 Seiten, 75 Bilder, 26 Tab. 2004.  
ISBN 3-87525-200-4.

Band 147: Ulrich Wenger  
Prozessoptimierung in der Wickeltechnik  
durch innovative maschinenbauliche und  
regelungstechnische Ansätze  
FAPS, 132 Seiten, 88 Bilder, 0 Tab. 2004.  
ISBN 3-87525-203-9.

Band 148: Stefan Slama  
Effizienzsteigerung in der Montage durch  
marktorientierte Montagestrukturen und  
erweiterte Mitarbeiterkompetenz  
FAPS, 188 Seiten, 125 Bilder, 0 Tab. 2004.  
ISBN 3-87525-204-7.

Band 149: Thomas Wurm  
Laserstrahljustieren mittels Aktoren-Ent-  
wicklung von Konzepten und Methoden  
für die rechnerunterstützte Modellierung  
und Optimierung von komplexen  
Aktorsystemen in der Mikrotechnik  
LFT, 122 Seiten, 51 Bilder, 9 Tab. 2004.  
ISBN 3-87525-206-3.

Band 150: Martino Celeghini  
Wirkmedienbasierte Blechumformung:  
Grundlagenuntersuchungen zum Einfluss  
von Werkstoff und Bauteilgeometrie  
LFT, 146 Seiten, 77 Bilder, 6 Tab. 2004.  
ISBN 3-87525-207-1.

Band 151: Ralph Hohenstein  
Entwurf hochdynamischer Sensor- und  
Regelsysteme für die adaptive  
Laserbearbeitung  
LFT, 282 Seiten, 63 Bilder, 16 Tab. 2004.  
ISBN 3-87525-210-1.

Band 152: Angelika Hutterer  
Entwicklung prozessüberwachender  
Regelkreise für flexible  
Formgebungsprozesse  
LFT, 149 Seiten, 57 Bilder, 2 Tab. 2005.  
ISBN 3-87525-212-8.

Band 153: Emil Egerer  
Massivumformen metallischer Kleinst-  
teile bei erhöhter Prozesstemperatur  
LFT, 158 Seiten, 87 Bilder, 10 Tab. 2005.  
ISBN 3-87525-213-6.

Band 154: Rüdiger Holzmann  
Strategien zur nachhaltigen Optimierung  
von Qualität und Zuverlässigkeit in  
der Fertigung hochintegrierter  
Flachbaugruppen  
FAPS, 186 Seiten, 99 Bilder, 19 Tab. 2005.  
ISBN 3-87525-217-9.

Band 155: Marco Nock  
Biegeumformen mit  
Elastomerwerkzeugen Modellierung,  
Prozessauslegung und Abgrenzung des  
Verfahrens am Beispiel des Rohrbiegens  
LFT, 164 Seiten, 85 Bilder, 13 Tab. 2005.  
ISBN 3-87525-218-7.

Band 156: Frank Niebling  
Qualifizierung einer Prozesskette zum  
Laserstrahlsintern metallischer Bauteile  
LFT, 148 Seiten, 89 Bilder, 3 Tab. 2005.  
ISBN 3-87525-219-5.

Band 157: Markus Meiler  
Großserientauglichkeit trockenschmier-  
stoffbeschichteter Aluminiumbleche im  
Presswerk Grundlegende Untersuchun-  
gen zur Tribologie, zum Umformverhal-  
ten und Bauteilversuche  
LFT, 104 Seiten, 57 Bilder, 21 Tab. 2005.  
ISBN 3-87525-221-7.

Band 158: Agus Sutanto  
Solution Approaches for Planning of  
Assembly Systems in Three-Dimensional  
Virtual Environments  
FAPS, 169 Seiten, 98 Bilder, 3 Tab. 2005.  
ISBN 3-87525-220-9.

Band 159: Matthias Boiger  
Hochleistungssysteme für die Fertigung  
elektronischer Baugruppen auf der Basis  
flexibler Schaltungsträger  
FAPS, 175 Seiten, 111 Bilder, 8 Tab. 2005.  
ISBN 3-87525-222-5.

Band 160: Matthias Pitz  
Laserunterstütztes Biegen höchstfester  
Mehrphasenstähle  
LFT, 120 Seiten, 73 Bilder, 11 Tab. 2005.  
ISBN 3-87525-223-3.

Band 161: Meik Vahl  
Beitrag zur gezielten Beeinflussung des  
Werkstoffflusses beim Innenhochdruck-  
Umformen von Blechen  
LFT, 165 Seiten, 94 Bilder, 15 Tab. 2005.  
ISBN 3-87525-224-1.

Band 162: Peter K. Kraus  
Plattformstrategien - Realisierung  
einer varianz- und kostenoptimierten  
Wertschöpfung  
FAPS, 181 Seiten, 95 Bilder, 0 Tab. 2005.  
ISBN 3-87525-226-8.

Band 163: Adrienn Cser  
Laserstrahlschmelzabtrag - Prozessana-  
lyse und -modellierung  
LFT, 146 Seiten, 79 Bilder, 3 Tab. 2005.  
ISBN 3-87525-227-6.

Band 164: Markus C. Hahn  
Grundlegende Untersuchungen zur  
Herstellung von Leichtbauverbundstruk-  
turen mit Aluminiumschaumkern  
LFT, 143 Seiten, 60 Bilder, 16 Tab. 2005.  
ISBN 3-87525-228-4.

Band 165: Gordana Michos  
Mechatronische Ansätze zur Optimie-  
rung von Vorschubachsen  
FAPS, 146 Seiten, 87 Bilder, 17 Tab. 2005.  
ISBN 3-87525-230-6.

Band 166: Markus Stark  
Auslegung und Fertigung hochpräziser  
Faser-Kollimator-Arrays  
LFT, 158 Seiten, 115 Bilder, 11 Tab. 2005.  
ISBN 3-87525-231-4.

Band 167: Yurong Zhou  
Kollaboratives Engineering Management  
in der integrierten virtuellen Entwicklung  
der Anlagen für die Elektronikproduktion  
FAPS, 156 Seiten, 84 Bilder, 6 Tab. 2005.  
ISBN 3-87525-232-2.

Band 168: Werner Enser  
Neue Formen permanenter und lösbarer elektrischer Kontaktierungen für mechatronische Baugruppen  
FAPS, 190 Seiten, 112 Bilder, 5 Tab. 2005.  
ISBN 3-87525-233-0.

Band 169: Katrin Melzer  
Integrierte Produktpolitik bei elektrischen und elektronischen Geräten zur Optimierung des Product-Life-Cycle  
FAPS, 155 Seiten, 91 Bilder, 17 Tab. 2005.  
ISBN 3-87525-234-9.

Band 170: Alexander Putz  
Grundlegende Untersuchungen zur Erfassung der realen Vorspannung von armierten Kaltfließpresswerkzeugen mittels Ultraschall  
LFT, 137 Seiten, 71 Bilder, 15 Tab. 2006.  
ISBN 3-87525-237-3.

Band 171: Martin Prechtl  
Automatisiertes Schichtverfahren für metallische Folien - System- und Prozesstechnik  
LFT, 154 Seiten, 45 Bilder, 7 Tab. 2006.  
ISBN 3-87525-238-1.

Band 172: Markus Meidert  
Beitrag zur deterministischen Lebensdauerabschätzung von Werkzeugen der Kaltmassivumformung  
LFT, 131 Seiten, 78 Bilder, 9 Tab. 2006.  
ISBN 3-87525-239-X.

Band 173: Bernd Müller  
Robuste, automatisierte Montagesysteme durch adaptive Prozessführung und montageübergreifende Fehlerprävention am Beispiel flächiger Leichtbauteile  
FAPS, 147 Seiten, 77 Bilder, 0 Tab. 2006.  
ISBN 3-87525-240-3.

Band 174: Alexander Hofmann  
Hybrides Laserdurchstrahlschweißen von Kunststoffen  
LFT, 136 Seiten, 72 Bilder, 4 Tab. 2006.  
ISBN 978-3-87525-243-9.

Band 175: Peter Wölflick  
Innovative Substrate und Prozesse mit feinsten Strukturen für bleifreie Mechatronik-Anwendungen  
FAPS, 177 Seiten, 148 Bilder, 24 Tab. 2006.  
ISBN 978-3-87525-246-0.

Band 176: Attila Komlodi  
Detection and Prevention of Hot Cracks during Laser Welding of Aluminium Alloys Using Advanced Simulation Methods  
LFT, 155 Seiten, 89 Bilder, 14 Tab. 2006.  
ISBN 978-3-87525-248-4.

Band 177: Uwe Popp  
Grundlegende Untersuchungen zum Laserstrahlstrukturieren von Kaltmassivumformwerkzeugen  
LFT, 140 Seiten, 67 Bilder, 16 Tab. 2006.  
ISBN 978-3-87525-249-1.

Band 178: Veit Rückel  
Rechnergestützte Ablaufplanung und Bahngenerierung Für kooperierende Industrieroboter  
FAPS, 148 Seiten, 75 Bilder, 7 Tab. 2006.  
ISBN 978-3-87525-250-7.

Band 179: Manfred Dirscherl  
Nicht-thermische Mikrojustiertechnik mittels ultrakurzer Laserpulse  
LFT, 154 Seiten, 69 Bilder, 10 Tab. 2007.  
ISBN 978-3-87525-251-4.

Band 180: Yong Zhuo  
Entwurf eines rechnergestützten integrierten Systems für Konstruktion und Fertigungsplanung räumlicher spritzgegossener Schaltungsträger (3D-MID)  
FAPS, 181 Seiten, 95 Bilder, 5 Tab. 2007.  
ISBN 978-3-87525-253-8.

Band 181: Stefan Lang  
Durchgängige Mitarbeiterinformation zur Steigerung von Effizienz und Prozesssicherheit in der Produktion  
FAPS, 172 Seiten, 93 Bilder. 2007.  
ISBN 978-3-87525-257-6.

Band 182: Hans-Joachim Krauß  
Laserstrahlinduzierte Pyrolyse präkeramischer Polymere  
LFT, 171 Seiten, 100 Bilder. 2007.  
ISBN 978-3-87525-258-3.

Band 183: Stefan Junker  
Technologien und Systemlösungen für die flexibel automatisierte Bestückung permanent erregter Läufer mit oberflächenmontierten Dauermagneten  
FAPS, 173 Seiten, 75 Bilder. 2007.  
ISBN 978-3-87525-259-0.

Band 184: Rainer Kohlbauer  
Wissensbasierte Methoden für die simulationsgestützte Auslegung wirkmedienbasierter Blechumformprozesse  
LFT, 135 Seiten, 50 Bilder. 2007.  
ISBN 978-3-87525-260-6.

Band 185: Klaus Lamprecht  
Wirkmedienbasierte Umformung tiefgezogener Vorformen unter besonderer Berücksichtigung maßgeschneiderter Halbzeuge  
LFT, 137 Seiten, 81 Bilder. 2007.  
ISBN 978-3-87525-265-1.

Band 186: Bernd Zolleiß  
Optimierte Prozesse und Systeme für die Bestückung mechatronischer Baugruppen  
FAPS, 180 Seiten, 117 Bilder. 2007.  
ISBN 978-3-87525-266-8.

Band 187: Michael Kerausch  
Simulationsgestützte Prozessauslegung für das Umformen lokal wärmebehandelter Aluminiumplatten  
LFT, 146 Seiten, 76 Bilder, 7 Tab. 2007.  
ISBN 978-3-87525-267-5.

Band 188: Matthias Weber  
Unterstützung der Wandlungsfähigkeit von Produktionsanlagen durch innovative Softwaresysteme  
FAPS, 183 Seiten, 122 Bilder, 3 Tab. 2007.  
ISBN 978-3-87525-269-9.

Band 189: Thomas Frick  
Untersuchung der prozessbestimmenden Strahl-Stoff-Wechselwirkungen beim Laserstrahlschweißen von Kunststoffen  
LFT, 104 Seiten, 62 Bilder, 8 Tab. 2007.  
ISBN 978-3-87525-268-2.

Band 190: Joachim Hecht  
Werkstoffcharakterisierung und  
Prozessauslegung für die wirkmedienba-  
sierte Doppelblech-Umformung von  
Magnesiumlegierungen  
LFT, 107 Seiten, 91 Bilder, 2 Tab. 2007.  
ISBN 978-3-87525-270-5.

Band 191: Ralf Völkl  
Stochastische Simulation zur Werkzeug-  
lebensdaueroptimierung und Präzisions-  
fertigung in der Kaltmassivumformung  
LFT, 178 Seiten, 75 Bilder, 12 Tab. 2008.  
ISBN 978-3-87525-272-9.

Band 192: Massimo Tolazzi  
Innenhochdruck-Umformen verstärkter  
Blech-Rahmenstrukturen  
LFT, 164 Seiten, 85 Bilder, 7 Tab. 2008.  
ISBN 978-3-87525-273-6.

Band 193: Cornelia Hoff  
Untersuchung der Prozesseinflussgrößen  
beim Presshärten des höchstfesten  
Vergütungsstahls 22MnB5  
LFT, 133 Seiten, 92 Bilder, 5 Tab. 2008.  
ISBN 978-3-87525-275-0.

Band 194: Christian Alvarez  
Simulationsgestützte Methoden zur  
effizienten Gestaltung von Lötprozessen  
in der Elektronikproduktion  
FAPS, 149 Seiten, 86 Bilder, 8 Tab. 2008.  
ISBN 978-3-87525-277-4.

Band 195: Andreas Kunze  
Automatisierte Montage von makrome-  
chatronischen Modulen zur flexiblen  
Integration in hybride  
Pkw-Bordnetzsysteme  
FAPS, 160 Seiten, 90 Bilder, 14 Tab. 2008.  
ISBN 978-3-87525-278-1.

Band 196: Wolfgang Hußnätter  
Grundlegende Untersuchungen zur  
experimentellen Ermittlung und zur  
Modellierung von Fließortkurven bei  
erhöhten Temperaturen  
LFT, 152 Seiten, 73 Bilder, 21 Tab. 2008.  
ISBN 978-3-87525-279-8.

Band 197: Thomas Bigl  
Entwicklung, angepasste Herstellungs-  
verfahren und erweiterte Qualitätssiche-  
rung von einsetzgerechten elektroni-  
schen Baugruppen  
FAPS, 175 Seiten, 107 Bilder, 14 Tab. 2008.  
ISBN 978-3-87525-280-4.

Band 198: Stephan Roth  
Grundlegende Untersuchungen zum  
Excimerlaserstrahl-Abtragen unter  
Flüssigkeitsfilmen  
LFT, 113 Seiten, 47 Bilder, 14 Tab. 2008.  
ISBN 978-3-87525-281-1.

Band 199: Artur Giera  
Prozesstechnische Untersuchungen  
zum Rührreibschweißen metallischer  
Werkstoffe  
LFT, 179 Seiten, 104 Bilder, 36 Tab. 2008.  
ISBN 978-3-87525-282-8.

Band 200: Jürgen Lechler  
Beschreibung und Modellierung  
des Werkstoffverhaltens von  
presshärtbaren Bor-Manganstählen  
LFT, 154 Seiten, 75 Bilder, 12 Tab. 2009.  
ISBN 978-3-87525-286-6.

Band 201: Andreas Blankl  
Untersuchungen zur Erhöhung der  
Prozessrobustheit bei der Innenhoch-  
druck-Umformung von flächigen Halb-  
zeugen mit vor- bzw. nachgeschalteten  
Laserstrahlfügeoperationen  
LFT, 120 Seiten, 68 Bilder, 9 Tab. 2009.  
ISBN 978-3-87525-287-3.

Band 202: Andreas Schaller  
Modellierung eines nachfrageorientierten  
Produktionskonzeptes für mobile  
Telekommunikationsgeräte  
FAPS, 120 Seiten, 79 Bilder, 0 Tab. 2009.  
ISBN 978-3-87525-289-7.

Band 203: Claudius Schimpf  
Optimierung von Zuverlässigkeitsunter-  
suchungen, Prüfabläufen und Nachar-  
beitsprozessen in der Elektronikproduk-  
tion  
FAPS, 162 Seiten, 90 Bilder, 14 Tab. 2009.  
ISBN 978-3-87525-290-3.

Band 204: Simon Dietrich  
Sensoriken zur Schwerpunktslagebestim-  
mung der optischen Prozessemissionen  
beim Laserstrahliefschweißen  
LFT, 138 Seiten, 70 Bilder, 5 Tab. 2009.  
ISBN 978-3-87525-292-7.

Band 205: Wolfgang Wolf  
Entwicklung eines agentenbasierten  
Steuerungssystems zur  
Materialflussorganisation im  
wandelbaren Produktionsumfeld  
FAPS, 167 Seiten, 98 Bilder. 2009.  
ISBN 978-3-87525-293-4.

Band 206: Steffen Polster  
Laserdurchstrahlschweißen  
transparenter Polymerbauteile  
LFT, 160 Seiten, 92 Bilder, 13 Tab. 2009.  
ISBN 978-3-87525-294-1.

Band 207: Stephan Manuel Dörfler  
Rührreibschweißen von walzplattiertem  
Halbzeug und Aluminiumblech zur  
Herstellung flächiger Aluminiumschaum-  
Sandwich-Verbundstrukturen  
LFT, 190 Seiten, 98 Bilder, 5 Tab. 2009.  
ISBN 978-3-87525-295-8.

Band 208: Uwe Vogt  
Seriennahe Auslegung von Aluminium  
Tailored Heat Treated Blanks  
LFT, 151 Seiten, 68 Bilder, 26 Tab. 2009.  
ISBN 978-3-87525-296-5.

Band 209: Till Laumann  
Qualitative und quantitative Bewertung  
der Crashtauglichkeit von höchstfesten  
Stählen  
LFT, 117 Seiten, 69 Bilder, 7 Tab. 2009.  
ISBN 978-3-87525-299-6.

Band 210: Alexander Diehl  
Größeneffekte bei Biegeprozessen-  
Entwicklung einer Methodik zur  
Identifikation und Quantifizierung  
LFT, 180 Seiten, 92 Bilder, 12 Tab. 2010.  
ISBN 978-3-87525-302-3.

Band 211: Detlev Staud  
Effiziente Prozesskettenauslegung für das  
Umformen lokal wärmebehandelter und  
geschweißter Aluminiumbleche  
LFT, 164 Seiten, 72 Bilder, 12 Tab. 2010.  
ISBN 978-3-87525-303-0.

Band 212: Jens Ackermann  
Prozesssicherung beim Laserdurchstrahl-  
schweißen thermoplastischer Kunststoffe  
LPT, 129 Seiten, 74 Bilder, 13 Tab. 2010.  
ISBN 978-3-87525-305-4.

Band 213: Stephan Weidel  
Grundlegende Untersuchungen zum  
Kontaktzustand zwischen Werkstück  
und Werkzeug bei umformtechnischen  
Prozessen unter tribologischen  
Gesichtspunkten  
LFT, 144 Seiten, 67 Bilder, 11 Tab. 2010.  
ISBN 978-3-87525-307-8.

Band 214: Stefan Geißdörfer  
Entwicklung eines mesoskopischen Modells zur Abbildung von Größeneffekten in der Kaltmassivumformung mit Methoden der FE-Simulation  
LFT, 133 Seiten, 83 Bilder, 11 Tab. 2010.  
ISBN 978-3-87525-308-5.

Band 215: Christian Matzner  
Konzeption produktspezifischer Lösungen zur Robustheitssteigerung elektronischer Systeme gegen die Einwirkung von Betaung im Automobil  
FAPS, 165 Seiten, 93 Bilder, 14 Tab. 2010.  
ISBN 978-3-87525-309-2.

Band 216: Florian Schüssler  
Verbindungs- und Systemtechnik für thermisch hochbeanspruchte und miniaturisierte elektronische Baugruppen  
FAPS, 184 Seiten, 93 Bilder, 18 Tab. 2010.  
ISBN 978-3-87525-310-8.

Band 217: Massimo Cojutti  
Strategien zur Erweiterung der Prozessgrenzen bei der Innhochdruck-Umformung von Rohren und Blechpaaren  
LFT, 125 Seiten, 56 Bilder, 9 Tab. 2010.  
ISBN 978-3-87525-312-2.

Band 218: Raoul Plettke  
Mehrkriterielle Optimierung komplexer Aktorsysteme für das Laserstrahljustieren  
LFT, 152 Seiten, 25 Bilder, 3 Tab. 2010.  
ISBN 978-3-87525-315-3.

Band 219: Andreas Dobroschke  
Flexible Automatisierungslösungen für die Fertigung wickeltechnischer Produkte  
FAPS, 184 Seiten, 109 Bilder, 18 Tab. 2011.  
ISBN 978-3-87525-317-7.

Band 220: Azhar Zam  
Optical Tissue Differentiation for Sensor-Controlled Tissue-Specific Laser Surgery  
LPT, 99 Seiten, 45 Bilder, 8 Tab. 2011.  
ISBN 978-3-87525-318-4.

Band 221: Michael Rösch  
Potenziale und Strategien zur Optimierung des Schablonendruckprozesses in der Elektronikproduktion  
FAPS, 192 Seiten, 127 Bilder, 19 Tab. 2011.  
ISBN 978-3-87525-319-1.

Band 222: Thomas Rechtenwald  
Quasi-isothermes Laserstrahlsintern von Hochtemperatur-Thermoplasten - Eine Betrachtung werkstoff-prozessspezifischer Aspekte am Beispiel PEEK  
LPT, 150 Seiten, 62 Bilder, 8 Tab. 2011.  
ISBN 978-3-87525-320-7.

Band 223: Daniel Craiovan  
Prozesse und Systemlösungen für die SMT-Montage optischer Bauelemente auf Substrate mit integrierten Lichtwellenleitern  
FAPS, 165 Seiten, 85 Bilder, 8 Tab. 2011.  
ISBN 978-3-87525-324-5.

Band 224: Kay Wagner  
Beanspruchungsangepasste  
Kaltmassivumformwerkzeuge durch  
lokal optimierte Werkzeugoberflächen  
LFT, 147 Seiten, 103 Bilder, 17 Tab. 2011.  
ISBN 978-3-87525-325-2.

Band 225: Martin Brandhuber  
Verbesserung der Prognosegüte des Ver-  
sagens von Punktschweißverbindungen  
bei höchstfesten Stahlgüten  
LFT, 155 Seiten, 91 Bilder, 19 Tab. 2011.  
ISBN 978-3-87525-327-6.

Band 226: Peter Sebastian Feuser  
Ein Ansatz zur Herstellung von  
pressgehärteten Karosseriekomponenten  
mit maßgeschneiderten mechanischen  
Eigenschaften: Temperierte Umform-  
werkzeuge. Prozessfenster, Prozess-  
simulation und funktionale Untersuchung  
LFT, 195 Seiten, 97 Bilder, 60 Tab. 2012.  
ISBN 978-3-87525-328-3.

Band 227: Murat Arbak  
Material Adapted Design of Cold Forging  
Tools Exemplified by Powder  
Metallurgical Tool Steels and Ceramics  
LFT, 109 Seiten, 56 Bilder, 8 Tab. 2012.  
ISBN 978-3-87525-330-6.

Band 228: Indra Pitz  
Beschleunigte Simulation des  
Laserstrahlumformens von  
Aluminiumblechen  
LPT, 137 Seiten, 45 Bilder, 27 Tab. 2012.  
ISBN 978-3-87525-333-7.

Band 229: Alexander Grimm  
Prozessanalyse und -überwachung des  
Laserstrahlhartlötens mittels optischer  
Sensorik  
LPT, 125 Seiten, 61 Bilder, 5 Tab. 2012.  
ISBN 978-3-87525-334-4.

Band 230: Markus Kaupper  
Biegen von höhenfesten Stahlblechwerk-  
stoffen - Umformverhalten und Grenzen  
der Biegebarkeit  
LFT, 160 Seiten, 57 Bilder, 10 Tab. 2012.  
ISBN 978-3-87525-339-9.

Band 231: Thomas Kroiß  
Modellbasierte Prozessauslegung für  
die Kaltmassivumformung unter  
Brücksichtigung der Werkzeug- und  
Pressenauffederung  
LFT, 169 Seiten, 50 Bilder, 19 Tab. 2012.  
ISBN 978-3-87525-341-2.

Band 232: Christian Goth  
Analyse und Optimierung der Entwick-  
lung und Zuverlässigkeit räumlicher  
Schaltungsträger (3D-MID)  
FAPS, 176 Seiten, 102 Bilder, 22 Tab. 2012.  
ISBN 978-3-87525-340-5.

Band 233: Christian Ziegler  
Ganzheitliche Automatisierung  
mechatronischer Systeme in der Medizin  
am Beispiel Strahlentherapie  
FAPS, 170 Seiten, 71 Bilder, 19 Tab. 2012.  
ISBN 978-3-87525-342-9.

Band 234: Florian Albert  
Automatisiertes Laserstrahllöten  
und -reparaturlöten elektronischer  
Baugruppen  
LPT, 127 Seiten, 78 Bilder, 11 Tab. 2012.  
ISBN 978-3-87525-344-3.

Band 235: Thomas Stöhr  
Analyse und Beschreibung des  
mechanischen Werkstoffverhaltens  
von presshärtbaren Bor-Manganstählen  
LFT, 118 Seiten, 74 Bilder, 18 Tab. 2013.  
ISBN 978-3-87525-346-7.

Band 236: Christian Kägeler  
Prozessdynamik beim  
Laserstrahlschweißen verzinkter  
Stahlbleche im Überlappstoß  
LPT, 145 Seiten, 80 Bilder, 3 Tab. 2013.  
ISBN 978-3-87525-347-4.

Band 237: Andreas Sulzberger  
Seriennahe Auslegung der Prozesskette  
zur wärmeunterstützten Umformung  
von Aluminiumblechwerkstoffen  
LFT, 153 Seiten, 87 Bilder, 17 Tab. 2013.  
ISBN 978-3-87525-349-8.

Band 238: Simon Opel  
Herstellung prozessangepasster  
Halbzeuge mit variabler Blechdicke  
durch die Anwendung von Verfahren  
der Blechmassivumformung  
LFT, 165 Seiten, 108 Bilder, 27 Tab. 2013.  
ISBN 978-3-87525-350-4.

Band 239: Rajesh Kanawade  
In-vivo Monitoring of Epithelium  
Vessel and Capillary Density for the  
Application of Detection of Clinical  
Shock and Early Signs of Cancer Develop-  
ment  
LPT, 124 Seiten, 58 Bilder, 15 Tab. 2013.  
ISBN 978-3-87525-351-1.

Band 240: Stephan Busse  
Entwicklung und Qualifizierung eines  
Schneidclinchverfahrens  
LFT, 119 Seiten, 86 Bilder, 20 Tab. 2013.  
ISBN 978-3-87525-352-8.

Band 241: Karl-Heinz Leitz  
Mikro- und Nanostrukturierung mit kurz  
und ultrakurz gepulster Laserstrahlung  
LPT, 154 Seiten, 71 Bilder, 9 Tab. 2013.  
ISBN 978-3-87525-355-9.

Band 242: Markus Michl  
Webbasierte Ansätze zur ganzheitlichen  
technischen Diagnose  
FAPS, 182 Seiten, 62 Bilder, 20 Tab. 2013.  
ISBN 978-3-87525-356-6.

Band 243: Vera Sturm  
Einfluss von Chargenschwankungen  
auf die Verarbeitungsgrenzen von  
Stahlwerkstoffen  
LFT, 113 Seiten, 58 Bilder, 9 Tab. 2013.  
ISBN 978-3-87525-357-3.

Band 244: Christian Neudel  
Mikrostrukturelle und mechanisch-  
technologische Eigenschaften  
widerstandspunktgeschweißter  
Aluminium-Stahl-Verbindungen für  
den Fahrzeugbau  
LFT, 178 Seiten, 171 Bilder, 31 Tab. 2014.  
ISBN 978-3-87525-358-0.

Band 245: Anja Neumann  
Konzept zur Beherrschung der  
Prozessschwankungen im Presswerk  
LFT, 162 Seiten, 68 Bilder, 15 Tab. 2014.  
ISBN 978-3-87525-360-3.

Band 246: Ulf-Hermann Quentin  
Laserbasierte Nanostrukturierung mit  
optisch positionierten Mikrolinsen  
LPT, 137 Seiten, 89 Bilder, 6 Tab. 2014.  
ISBN 978-3-87525-361-0.

Band 247: Erik Lamprecht  
Der Einfluss der Fertigungsverfahren  
auf die Wirbelstromverluste von  
Stator-Einzelzahnblechpaketen für  
den Einsatz in Hybrid- und Elektrofahr-  
zeugen  
FAPS, 148 Seiten, 138 Bilder, 4 Tab. 2014.  
ISBN 978-3-87525-362-7.

Band 248: Sebastian Rösler  
Wirkmedienbasierte Umformung von  
Blechhalbzeugen unter Anwendung  
magnetorheologischer Flüssigkeiten als  
kombiniertes Wirk- und Dichtmedium  
LFT, 148 Seiten, 61 Bilder, 12 Tab. 2014.  
ISBN 978-3-87525-363-4.

Band 249: Paul Hippchen  
Simulative Prognose der Geometrie  
indirekt pressgehärteter Karosseriebau-  
teile für die industrielle Anwendung  
LFT, 163 Seiten, 89 Bilder, 12 Tab. 2014.  
ISBN 978-3-87525-364-1.

Band 250: Martin Zubeil  
Versagensprognose bei der Prozess  
simulation von Biegeumform- und Falz-  
verfahren  
LFT, 171 Seiten, 90 Bilder, 5 Tab. 2014.  
ISBN 978-3-87525-365-8.

Band 251: Alexander Kühl  
Flexible Automatisierung der  
Statorenmontage mit Hilfe einer  
universellen ambidexteren Kinematik  
FAPS, 142 Seiten, 60 Bilder, 26 Tab. 2014.  
ISBN 978-3-87525-367-2.

Band 252: Thomas Albrecht  
Optimierte Fertigungstechnologien  
für Rotoren getriebeintegrierter  
PM-Synchronmotoren von  
Hybridfahrzeugen  
FAPS, 198 Seiten, 130 Bilder, 38 Tab. 2014.  
ISBN 978-3-87525-368-9.

Band 253: Florian Risch  
Planning and Production Concepts for  
Contactless Power Transfer Systems for  
Electric Vehicles  
FAPS, 185 Seiten, 125 Bilder, 13 Tab. 2014.  
ISBN 978-3-87525-369-6.

Band 254: Markus Weigl  
Laserstrahlschweißen von Mischverbindungen aus austenitischen und ferritischen korrosionsbeständigen Stahlwerkstoffen  
LPT, 184 Seiten, 110 Bilder, 6 Tab. 2014.  
ISBN 978-3-87525-370-2.

Band 255: Johannes Noneder  
Beanspruchungserfassung für die Validierung von FE-Modellen zur Auslegung von Massivumformwerkzeugen  
LFT, 161 Seiten, 65 Bilder, 14 Tab. 2014.  
ISBN 978-3-87525-371-9.

Band 256: Andreas Reinhardt  
Ressourceneffiziente Prozess- und Produktionstechnologie für flexible Schaltungsträger  
FAPS, 123 Seiten, 69 Bilder, 19 Tab. 2014.  
ISBN 978-3-87525-373-3.

Band 257: Tobias Schmuck  
Ein Beitrag zur effizienten Gestaltung globaler Produktions- und Logistiknetzwerke mittels Simulation  
FAPS, 151 Seiten, 74 Bilder. 2014.  
ISBN 978-3-87525-374-0.

Band 258: Bernd Eichenhüller  
Untersuchungen der Effekte und Wechselwirkungen charakteristischer Einflussgrößen auf das Umformverhalten bei Mikroumformprozessen  
LFT, 127 Seiten, 29 Bilder, 9 Tab. 2014.  
ISBN 978-3-87525-375-7.

Band 259: Felix Lütteke  
Vielseitiges autonomes Transportsystem basierend auf Weltmodellerstellung mittels Datenfusion von Deckenkameras und Fahrzeugsensoren  
FAPS, 152 Seiten, 54 Bilder, 20 Tab. 2014.  
ISBN 978-3-87525-376-4.

Band 260: Martin Grüner  
Hochdruck-Blechumformung mit formlos festen Stoffen als Wirkmedium  
LFT, 144 Seiten, 66 Bilder, 29 Tab. 2014.  
ISBN 978-3-87525-379-5.

Band 261: Christian Brock  
Analyse und Regelung des Laserstrahliefschweißprozesses durch Detektion der Metaldampffackelposition  
LPT, 126 Seiten, 65 Bilder, 3 Tab. 2015.  
ISBN 978-3-87525-380-1.

Band 262: Peter Vatter  
Sensitivitätsanalyse des 3-Rollen-Schubbiegens auf Basis der Finite Elemente Methode  
LFT, 145 Seiten, 57 Bilder, 26 Tab. 2015.  
ISBN 978-3-87525-381-8.

Band 263: Florian Klämpfl  
Planung von Laserbestrahlungen durch simulationsbasierte Optimierung  
LPT, 169 Seiten, 78 Bilder, 32 Tab. 2015.  
ISBN 978-3-87525-384-9.

Band 264: Matthias Domke  
Transiente physikalische Mechanismen  
bei der Laserablation von dünnen  
Metallschichten  
LFT, 133 Seiten, 43 Bilder, 3 Tab. 2015.  
ISBN 978-3-87525-385-6.

Band 265: Johannes Götz  
Community-basierte Optimierung des  
Anlagenengineerings  
FAPS, 177 Seiten, 80 Bilder, 30 Tab. 2015.  
ISBN 978-3-87525-386-3.

Band 266: Hung Nguyen  
Qualifizierung des Potentials von  
Verfestigungseffekten zur Erweiterung  
des Umformvermögens aushärtbarer  
Aluminiumlegierungen  
LFT, 137 Seiten, 57 Bilder, 16 Tab. 2015.  
ISBN 978-3-87525-387-0.

Band 267: Andreas Kuppert  
Erweiterung und Verbesserung von Ver-  
suchs- und Auswertetechniken für die  
Bestimmung von Grenzformänderungs-  
kurven  
LFT, 138 Seiten, 82 Bilder, 2 Tab. 2015.  
ISBN 978-3-87525-388-7.

Band 268: Kathleen Klaus  
Erstellung eines Werkstofforientierten  
Fertigungsprozessfensters zur Steigerung  
des Formgebungsvermögens von Alumi-  
niumlegierungen unter Anwendung einer  
zwischen geschalteten Wärmebehandlung  
LFT, 154 Seiten, 70 Bilder, 8 Tab. 2015.  
ISBN 978-3-87525-391-7.

Band 269: Thomas Svec  
Untersuchungen zur Herstellung von  
funktionsoptimierten Bauteilen im  
partiellen Presshärtprozess mittels lokal  
unterschiedlich temperierter Werkzeuge  
LFT, 166 Seiten, 87 Bilder, 15 Tab. 2015.  
ISBN 978-3-87525-392-4.

Band 270: Tobias Schrader  
Grundlegende Untersuchungen zur  
Verschleißcharakterisierung beschichte-  
ter Kaltmassivumformwerkzeuge  
LFT, 164 Seiten, 55 Bilder, 11 Tab. 2015.  
ISBN 978-3-87525-393-1.

Band 271: Matthäus Brela  
Untersuchung von Magnetfeld-Messme-  
thoden zur ganzheitlichen Wertschöp-  
fungsoptimierung und Fehlerdetektion  
an magnetischen Aktoren  
FAPS, 170 Seiten, 97 Bilder, 4 Tab. 2015.  
ISBN 978-3-87525-394-8.

Band 272: Michael Wieland  
Entwicklung einer Methode zur Prognose  
adhäsiven Verschleißes an Werkzeugen  
für das direkte Presshärten  
LFT, 156 Seiten, 84 Bilder, 9 Tab. 2015.  
ISBN 978-3-87525-395-5.

Band 273: René Schramm  
Strukturierte additive Metallisierung  
durch kaltaktives  
Atmosphärendruckplasma  
FAPS, 136 Seiten, 62 Bilder, 15 Tab. 2015.  
ISBN 978-3-87525-396-2.

Band 274: Michael Lechner  
Herstellung beanspruchungsangepasster  
Aluminiumblechhalbzeuge durch  
eine maßgeschneiderte Variation der  
Abkühlgeschwindigkeit nach  
Lösungsglühen  
LFT, 136 Seiten, 62 Bilder, 15 Tab. 2015.  
ISBN 978-3-87525-397-9.

Band 275: Kolja Andreas  
Einfluss der Oberflächenbeschaffenheit  
auf das Werkzeugeinsatzverhalten beim  
Kaltfließpressen  
LFT, 169 Seiten, 76 Bilder, 4 Tab. 2015.  
ISBN 978-3-87525-398-6.

Band 276: Marcus Baum  
Laser Consolidation of ITO Nanoparticles  
for the Generation of Thin Conductive  
Layers on Transparent Substrates  
LPT, 158 Seiten, 75 Bilder, 3 Tab. 2015.  
ISBN 978-3-87525-399-3.

Band 277: Thomas Schneider  
Umformtechnische Herstellung  
dünnwandiger Funktionsbauteile  
aus Feinblech durch Verfahren der  
Blechmassivumformung  
LFT, 188 Seiten, 95 Bilder, 7 Tab. 2015.  
ISBN 978-3-87525-401-3.

Band 278: Jochen Merhof  
Sematische Modellierung automatisierter  
Produktionssysteme zur Verbesserung  
der IT-Integration zwischen Anlagen-  
Engineering und Steuerungsebene  
FAPS, 157 Seiten, 88 Bilder, 8 Tab. 2015.  
ISBN 978-3-87525-402-0.

Band 279: Fabian Zöller  
Erarbeitung von Grundlagen zur  
Abbildung des tribologischen Systems  
in der Umformsimulation  
LFT, 126 Seiten, 51 Bilder, 3 Tab. 2016.  
ISBN 978-3-87525-403-7.

Band 280: Christian Hezler  
Einsatz technologischer Versuche zur  
Erweiterung der Versagensvorhersage  
bei Karosseriebauteilen aus höchstfesten  
Stählen  
LFT, 147 Seiten, 63 Bilder, 44 Tab. 2016.  
ISBN 978-3-87525-404-4.

Band 281: Jochen Böning  
Integration des Systemverhaltens von  
Automobil-Hochvoltleitungen in die  
virtuelle Absicherung durch  
strukturmechanische Simulation  
FAPS, 177 Seiten, 107 Bilder, 17 Tab. 2016.  
ISBN 978-3-87525-405-1.

Band 282: Johannes Kohl  
Automatisierte Datenerfassung für  
diskret ereignisorientierte Simulationen  
in der energieflexiblen Fabrik  
FAPS, 160 Seiten, 80 Bilder, 27 Tab. 2016.  
ISBN 978-3-87525-406-8.

Band 283: Peter Bechtold  
Mikroschockwellenumformung mittels  
ultrakurzer Laserpulse  
LPT, 155 Seiten, 59 Bilder, 10 Tab. 2016.  
ISBN 978-3-87525-407-5.

Band 284: Stefan Berger  
Laserstrahlschweißen thermoplastischer  
Kohlenstofffaserverbundwerkstoffe mit  
spezifischem Zusatzdraht  
LPT, 118 Seiten, 68 Bilder, 9 Tab. 2016.  
ISBN 978-3-87525-408-2.

Band 285: Martin Bornschlegl  
Methods-Energy Measurement - Eine  
Methode zur Energieplanung für  
Fügeverfahren im Karosseriebau  
FAPS, 136 Seiten, 72 Bilder, 46 Tab. 2016.  
ISBN 978-3-87525-409-9.

Band 286: Tobias Rackow  
Erweiterung des Unternehmenscontrol-  
lings um die Dimension Energie  
FAPS, 164 Seiten, 82 Bilder, 29 Tab. 2016.  
ISBN 978-3-87525-410-5.

Band 287: Johannes Koch  
Grundlegende Untersuchungen zur  
Herstellung zyklisch-symmetrischer  
Bauteile mit Nebenformelementen durch  
Blechmassivumformung  
LFT, 125 Seiten, 49 Bilder, 17 Tab. 2016.  
ISBN 978-3-87525-411-2.

Band 288: Hans Ulrich Vierzigmann  
Beitrag zur Untersuchung der  
tribologischen Bedingungen in der  
Blechmassivumformung - Bereitstellung  
von tribologischen Modellversuchen und  
Realisierung von Tailored Surfaces  
LFT, 174 Seiten, 102 Bilder, 34 Tab. 2016.  
ISBN 978-3-87525-412-9.

Band 289: Thomas Senner  
Methodik zur virtuellen Absicherung  
der formgebenden Operation des  
Nasspressprozesses von  
Gelege-Mehrschichtverbunden  
LFT, 156 Seiten, 96 Bilder, 21 Tab. 2016.  
ISBN 978-3-87525-414-3.

Band 290: Sven Kreitlein  
Der grundoperationsspezifische  
Mindestenergiebedarf als Referenzwert  
zur Bewertung der Energieeffizienz in  
der Produktion  
FAPS, 185 Seiten, 64 Bilder, 30 Tab. 2016.  
ISBN 978-3-87525-415-0.

Band 291: Christian Roos  
Remote-Laserstrahlschweißen verzinkter  
Stahlbleche in Kehlnahtgeometrie  
LPT, 123 Seiten, 52 Bilder, 0 Tab. 2016.  
ISBN 978-3-87525-416-7.

Band 292: Alexander Kahrmanidis  
Thermisch unterstützte Umformung von  
Aluminiumblechen  
LFT, 165 Seiten, 103 Bilder, 18 Tab. 2016.  
ISBN 978-3-87525-417-4.

Band 293: Jan Tremel  
Flexible Systems for Permanent  
Magnet Assembly and Magnetic Rotor  
Measurement / Flexible Systeme zur  
Montage von Permanentmagneten und  
zur Messung magnetischer Rotoren  
FAPS, 152 Seiten, 91 Bilder, 12 Tab. 2016.  
ISBN 978-3-87525-419-8.

Band 294: Ioannis Tsoupis  
Schädigungs- und Versagensverhalten  
hochfester Leichtbauwerkstoffe unter  
Biegebeanspruchung  
LFT, 176 Seiten, 51 Bilder, 6 Tab. 2017.  
ISBN 978-3-87525-420-4.

Band 295: Sven Hildering  
Grundlegende Untersuchungen zum  
Prozessverhalten von Silizium als  
Werkzeugwerkstoff für das  
Mikroscherschneiden metallischer Folien  
LFT, 177 Seiten, 74 Bilder, 17 Tab. 2017.  
ISBN 978-3-87525-422-8.

Band 296: Sasia Mareike Hertweck  
Zeitliche Pulsformung in der  
Lasermikromaterialbearbeitung –  
Grundlegende Untersuchungen und  
Anwendungen  
LPT, 146 Seiten, 67 Bilder, 5 Tab. 2017.  
ISBN 978-3-87525-423-5.

Band 297: Paryanto  
Mechatronic Simulation Approach for  
the Process Planning of Energy-Efficient  
Handling Systems  
FAPS, 162 Seiten, 86 Bilder, 13 Tab. 2017.  
ISBN 978-3-87525-424-2.

Band 298: Peer Stenzel  
Großorientaugliche Nadelwickeltechnik  
für verteilte Wicklungen im  
Anwendungsfall der E-Traktionsantriebe  
FAPS, 239 Seiten, 147 Bilder, 20 Tab. 2017.  
ISBN 978-3-87525-425-9.

Band 299: Mario Lušić  
Ein Vorgehensmodell zur Erstellung  
montageführender Werkerinformations-  
systeme simultan zum  
Produktentstehungsprozess  
FAPS, 174 Seiten, 79 Bilder, 22 Tab. 2017.  
ISBN 978-3-87525-426-6.

Band 300: Arnd Buschhaus  
Hochpräzise adaptive Steuerung und  
Regelung robotergeführter Prozesse  
FAPS, 202 Seiten, 96 Bilder, 4 Tab. 2017.  
ISBN 978-3-87525-427-3.

Band 301: Tobias Laumer  
Erzeugung von thermoplastischen  
Werkstoffverbunden mittels simultanem,  
intensitätsselektivem  
Laserstrahlschmelzen  
LPT, 140 Seiten, 82 Bilder, 0 Tab. 2017.  
ISBN 978-3-87525-428-0.

Band 302: Nora Unger  
Untersuchung einer thermisch unter-  
stützten Fertigungskette zur Herstellung  
umgeformter Bauteile aus der höherfes-  
ten Aluminiumlegierung EN AW-7020  
LFT, 142 Seiten, 53 Bilder, 8 Tab. 2017.  
ISBN 978-3-87525-429-7.

Band 303: Tommaso Stellin  
Design of Manufacturing Processes for  
the Cold Bulk Forming of Small Metal  
Components from Metal Strip  
LFT, 146 Seiten, 67 Bilder, 7 Tab. 2017.  
ISBN 978-3-87525-430-3.

Band 304: Bassim Bachy  
Experimental Investigation, Modeling,  
Simulation and Optimization of Molded  
Interconnect Devices (MID) Based on  
Laser Direct Structuring (LDS) / Experi-  
mentelle Untersuchung, Modellierung,  
Simulation und Optimierung von Molded  
Interconnect Devices (MID) basierend  
auf Laser Direktstrukturierung (LDS)  
FAPS, 168 Seiten, 120 Bilder, 26 Tab. 2017.  
ISBN 978-3-87525-431-0.

Band 305: Michael Spahr  
Automatisierte Kontaktierungsverfahren  
für flachleiterbasierte  
Pkw-Bordnetzsysteme  
FAPS, 197 Seiten, 98 Bilder, 17 Tab. 2017.  
ISBN 978-3-87525-432-7.

Band 306: Sebastian Suttner  
Charakterisierung und Modellierung  
des spannungszustandsabhängigen  
Werkstoffverhaltens der Magnesium-  
legierung AZ31B für die numerische  
Prozessauslegung  
LFT, 150 Seiten, 84 Bilder, 19 Tab. 2017.  
ISBN 978-3-87525-433-4.

Band 307: Bhargav Potdar  
A reliable methodology to deduce  
thermo-mechanical flow behaviour of  
hot stamping steels  
LFT, 203 Seiten, 98 Bilder, 27 Tab. 2017.  
ISBN 978-3-87525-436-5.

Band 308: Maria Löffler  
Steuerung von Blechmassivumformpro-  
zessen durch maßgeschneiderte  
tribologische Systeme  
LFT, viii u. 166 Seiten, 90 Bilder, 5 Tab.  
2018. ISBN 978-3-96147-133-1.

Band 309: Martin Müller  
Untersuchung des kombinierten Trenn-  
und Umformprozesses beim Fügen art-  
ungleicher Werkstoffe mittels  
Schneidclinchverfahren  
LFT, xi u. 149 Seiten, 89 Bilder, 6 Tab.  
2018. ISBN: 978-3-96147-135-5.

Band 310: Christopher Kästle  
Qualifizierung der Kupfer-Drahtbond-  
technologie für integrierte Leistungs-  
module in harschen Umgebungs-  
bedingungen  
FAPS, xii u. 167 Seiten, 70 Bilder, 18 Tab.  
2018. ISBN 978-3-96147-145-4.

Band 311: Daniel Vipavc  
Eine Simulationsmethode für das  
3-Rollen-Schubbiegen  
LFT, xiii u. 121 Seiten, 56 Bilder, 17 Tab.  
2018. ISBN 978-3-96147-147-8.

Band 312: Christina Ramer  
Arbeitsraumüberwachung und autonome  
Bahnplanung für ein sicheres und  
flexibles Roboter-Assistenzsystem  
in der Fertigung  
FAPS, xiv u. 188 Seiten, 57 Bilder, 9 Tab.  
2018. ISBN 978-3-96147-153-9.

Band 313: Miriam Rauer  
Der Einfluss von Poren auf die  
Zuverlässigkeit der Lötverbindungen  
von Hochleistungs-Leuchtdioden  
FAPS, xii u. 209 Seiten, 108 Bilder, 21 Tab.  
2018. ISBN 978-3-96147-157-7.

Band 314: Felix Tenner

Kamerabasierte Untersuchungen der Schmelze und Gasströmungen beim Laserstrahlschweißen verzinkter Stahlbleche

LPT, xxiii u. 184 Seiten, 94 Bilder, 7 Tab.  
2018. ISBN 978-3-96147-160-7.

Band 315: Aarief Syed-Khaja

Diffusion Soldering for High-temperature Packaging of Power Electronics

FAPS, x u. 202 Seiten, 144 Bilder, 32 Tab.  
2018. ISBN 978-3-87525-162-1.

Band 316: Adam Schaub

Grundlagenwissenschaftliche Untersuchung der kombinierten Prozesskette aus Umformen und Additive Fertigung

LFT, xi u. 192 Seiten, 72 Bilder, 27 Tab.  
2019. ISBN 978-3-96147-166-9.

Band 317: Daniel Gröbel

Herstellung von Nebenformelementen unterschiedlicher Geometrie an Blechen mittels Fließpressverfahren der Blechmassivumformung

LFT, x u. 165 Seiten, 96 Bilder, 13 Tab.  
2019. ISBN 978-3-96147-168-3.

Band 318: Philipp Hildenbrand

Entwicklung einer Methodik zur Herstellung von Tailored Blanks mit definierten Halbzeugeigenschaften durch einen Taumelprozess

LFT, ix u. 153 Seiten, 77 Bilder, 4 Tab.  
2019. ISBN 978-3-96147-174-4.

Band 319: Tobias Konrad

Simulative Auslegung der Spann- und Fixierkonzepte im Karosserierohbau: Bewertung der Baugruppenmaßhaltigkeit unter Berücksichtigung schwankender Einflussgrößen

LFT, x u. 203 Seiten, 134 Bilder, 32 Tab.  
2019. ISBN 978-3-96147-176-8.

Band 320: David Meinel

Architektur applikationsspezifischer Multi-Physics-Simulationskonfiguratoren am Beispiel modularer Triebzüge

FAPS, xii u. 166 Seiten, 82 Bilder, 25 Tab.  
2019. ISBN 978-3-96147-184-3.

Band 321: Andrea Zimmermann

Grundlegende Untersuchungen zum Einfluss fertigungsbedingter Eigenschaften auf die Ermüdungsfestigkeit kaltmassivumgeformter Bauteile

LFT, ix u. 160 Seiten, 66 Bilder, 5 Tab.  
2019. ISBN 978-3-96147-190-4.

Band 322: Christoph Amann

Simulative Prognose der Geometrie nassgepresster Karosseriebauteile aus Gelege-Mehrschichtverbunden

LFT, xvi u. 169 Seiten, 80 Bilder, 13 Tab.  
2019. ISBN 978-3-96147-194-2.

Band 323: Jennifer Tenner

Realisierung schmierstofffreier Tiefziehprozesse durch maßgeschneiderte Werkzeuoberflächen

LFT, x u. 187 Seiten, 68 Bilder, 13 Tab.  
2019. ISBN 978-3-96147-196-6.

Band 324: Susan Zöller

Mapping Individual Subjective Values to Product Design

KTmfk, xi u. 223 Seiten, 81 Bilder, 25 Tab.  
2019. ISBN 978-3-96147-202-4.

Band 325: Stefan Lutz  
Erarbeitung einer Methodik zur semiempirischen Ermittlung der Umwandlungskinetik durchhärtender Wälzlagerstähle für die Wärmebehandlungssimulation  
LFT, xiv u. 189 Seiten, 75 Bilder, 32 Tab.  
2019. ISBN 978-3-96147-209-3.

Band 326: Tobias Gnihl  
Modellbasierte Prozesskettenabbildung rührreibgeschweißter Aluminiumhalbzeuge zur umformtechnischen Herstellung höchstfester Leichtbauteile  
LFT, xii u. 167 Seiten, 68 Bilder, 17 Tab.  
2019. ISBN 978-3-96147-217-8.

Band 327: Johannes Bürner  
Technisch-wirtschaftliche Optionen zur Lastflexibilisierung durch intelligente elektrische Wärmespeicher  
FAPS, xiv u. 233 Seiten, 89 Bilder, 27 Tab.  
2019. ISBN 978-3-96147-219-2.

Band 328: Wolfgang Böhm  
Verbesserung des Umformverhaltens von mehrlagigen Aluminiumblechwerkstoffen mit ultrafeinkörnigem Gefüge  
LFT, ix u. 160 Seiten, 88 Bilder, 14 Tab.  
2019. ISBN 978-3-96147-227-7.

Band 329: Stefan Landkammer  
Grundsatzuntersuchungen, mathematische Modellierung und Ableitung einer Auslegungsmethodik für Gelenkantriebe nach dem Spinnenbeinprinzip  
LFT, xii u. 200 Seiten, 83 Bilder, 13 Tab.  
2019. ISBN 978-3-96147-229-1.

Band 330: Stephan Rapp  
Pump-Probe-Ellipsometrie zur Messung transienter optischer Materialeigenschaften bei der Ultrakurzpuls-Lasermaterialbearbeitung  
LPT, xi u. 143 Seiten, 49 Bilder, 2 Tab.  
2019. ISBN 978-3-96147-235-2.

Band 331: Michael Scholz  
Intralogistics Execution System mit integrierten autonomen, servicebasierten Transportentitäten  
FAPS, xi u. 195 Seiten, 55 Bilder, 11 Tab.  
2019. ISBN 978-3-96147-237-6.

Band 332: Eva Bogner  
Strategien der Produktindividualisierung in der produzierenden Industrie im Kontext der Digitalisierung  
FAPS, ix u. 201 Seiten, 55 Bilder, 28 Tab.  
2019. ISBN 978-3-96147-246-8.

Band 333: Daniel Benjamin Krüger  
Ein Ansatz zur CAD-integrierten muskuloskeletalen Analyse der Mensch-Maschine-Interaktion  
KTmfk, x u. 217 Seiten, 102 Bilder, 7 Tab.  
2019. ISBN 978-3-96147-250-5.

Band 334: Thomas Kuhn  
Qualität und Zuverlässigkeit laserdirektstrukturierter mechatronisch integrierter Baugruppen (LDS-MID)  
FAPS, ix u. 152 Seiten, 69 Bilder, 12 Tab.  
2019. ISBN: 978-3-96147-252-9.

Band 335: Hans Fleischmann  
Modellbasierte Zustands- und Prozess-  
überwachung auf Basis sozio-cyber-phy-  
sischer Systeme  
FAPS, xi u. 214 Seiten, 111 Bilder, 18 Tab.  
2019. ISBN: 978-3-96147-256-7.

Band 336: Markus Michalski  
Grundlegende Untersuchungen zum  
Prozess- und Werkstoffverhalten bei  
schwingungsüberlagerter Umformung  
LFT, xii u. 197 Seiten, 93 Bilder, 11 Tab.  
2019. ISBN: 978-3-96147-270-3.

Band 337: Markus Brandmeier  
Ganzheitliches ontologiebasiertes  
Wissensmanagement im Umfeld der  
industriellen Produktion  
FAPS, xi u. 255 Seiten, 77 Bilder, 33 Tab.  
2020. ISBN: 978-3-96147-275-8.

Band 338: Stephan Purr  
Datenerfassung für die Anwendung  
lernender Algorithmen bei der Herstel-  
lung von Blechformteilen  
LFT, ix u. 165 Seiten, 48 Bilder, 4 Tab.  
2020. ISBN: 978-3-96147-281-9.

Band 339: Christoph Kiener  
Kaltfließpressen von gerad- und schräg-  
verzahnten Zahnrädern  
LFT, viii u. 151 Seiten, 81 Bilder, 3 Tab.  
2020. ISBN 978-3-96147-287-1.

Band 340: Simon Spreng  
Numerische, analytische und empirische  
Modellierung des Heißscrimpprozesses  
FAPS, xix u. 204 Seiten, 91 Bilder, 27 Tab.  
2020. ISBN 978-3-96147-293-2.

Band 341: Patrik Schwingenschlögl  
Erarbeitung eines Prozessverständnisses  
zur Verbesserung der tribologischen  
Bedingungen beim Presshärten  
LFT, x u. 177 Seiten, 81 Bilder, 8 Tab.  
2020. ISBN 978-3-96147-297-0.

Band 342: Emanuela Affronti  
Evaluation of failure behaviour  
of sheet metals  
LFT, ix u. 136 Seiten, 57 Bilder, 20 Tab.  
2020. ISBN 978-3-96147-303-8.

Band 343: Julia Degner  
Grundlegende Untersuchungen zur  
Herstellung hochfester Aluminiumblech-  
bauteile in einem kombinierten Umform-  
und Abschreckprozess  
LFT, x u. 172 Seiten, 61 Bilder, 9 Tab.  
2020. ISBN 978-3-96147-307-6.

Band 344: Maximilian Wagner  
Automatische Bahnplanung für die Auf-  
teilung von Prozessbewegungen in syn-  
chrone Werkstück- und Werkzeugbewe-  
gungen mittels Multi-Roboter-Systemen  
FAPS, xxi u. 181 Seiten, 111 Bilder, 15 Tab.  
2020. ISBN 978-3-96147-309-0.

Band 345: Stefan Härter  
Qualifizierung des Montageprozesses  
hochminiaturisierter elektronischer Bau-  
elemente  
FAPS, ix u. 194 Seiten, 97 Bilder, 28 Tab.  
2020. ISBN 978-3-96147-314-4.

Band 346: Toni Donhauser  
Ressourcenorientierte Auftragsregelung  
in einer hybriden Produktion mittels  
betriebsbegleitender Simulation  
FAPS, xix u. 242 Seiten, 97 Bilder, 17 Tab.  
2020. ISBN 978-3-96147-316-8.

Band 347: Philipp Amend  
Laserbasiertes Schmelzkleben von Thermoplasten mit Metallen  
LPT, xv u. 154 Seiten, 67 Bilder.  
2020. ISBN 978-3-96147-326-7.

Band 348: Matthias Ehlert  
Simulationsunterstützte funktionale Grenzlagenabsicherung  
KTmfk, xvi u. 300 Seiten, 101 Bilder, 73 Tab. 2020. ISBN 978-3-96147-328-1.

Band 349: Thomas Sander  
Ein Beitrag zur Charakterisierung und Auslegung des Verbundes von Kunststoffsubstraten mit harten Dünnschichten  
KTmfk, xiv u. 178 Seiten, 88 Bilder, 21 Tab. 2020. ISBN 978-3-96147-330-4.

Band 350: Florian Pilz  
Fließpressen von Verzahnungselementen an Blechen  
LFT, x u. 170 Seiten, 103 Bilder, 4 Tab. 2020. ISBN 978-3-96147-332-8.

Band 351: Sebastian Josef Katona  
Evaluation und Aufbereitung von Produktsimulationen mittels abweichungsbehafteter Geometriemodelle  
KTmfk, ix u. 147 Seiten, 73 Bilder, 11 Tab. 2020. ISBN 978-3-96147-336-6.

Band 352: Jürgen Herrmann  
Kumulatives Walzplattieren. Bewertung der Umformeigenschaften mehrlagiger Blechwerkstoffe der ausscheidungshärtbaren Legierung AA6014  
LFT, x u. 157 Seiten, 64 Bilder, 5 Tab. 2020. ISBN 978-3-96147-344-1.

Band 353: Christof Küstner  
Assistenzsystem zur Unterstützung der datengetriebenen Produktentwicklung  
KTmfk, xii u. 219 Seiten, 63 Bilder, 14 Tab. 2020. ISBN 978-3-96147-348-9.

Band 354: Tobias Gläßel  
Prozessketten zum Laserstrahlschweißen von flachleiterbasierten Formspulenwicklungen für automobile Traktionsantriebe  
FAPS, xiv u. 206 Seiten, 89 Bilder, 11 Tab. 2020. ISBN 978-3-96147-356-4.

Band 355: Andreas Meinel  
Experimentelle Untersuchung der Auswirkungen von Axialschwingungen auf Reibung und Verschleiß in Zylinderrollenlagern  
KTmfk, xii u. 162 Seiten, 56 Bilder, 7 Tab. 2020. ISBN 978-3-96147-358-8.

Band 356: Hannah Riedle  
Haptische, generische Modelle weicher anatomischer Strukturen für die chirurgische Simulation  
FAPS, xxx u. 179 Seiten, 82 Bilder, 35 Tab. 2020. ISBN 978-3-96147-367-0.

Band 357: Maximilian Landgraf  
Leistungselektronik für den Einsatz dielektrischer Elastomere in aktorischen, sensorischen und integrierten sensomotorischen Systemen  
FAPS, xxiii u. 166 Seiten, 71 Bilder, 10 Tab. 2020. ISBN 978-3-96147-380-9.

Band 358: Alireza Esfandiyari  
Multi-Objective Process Optimization for Overpressure Reflow Soldering in Electronics Production  
FAPS, xviii u. 175 Seiten, 57 Bilder, 23 Tab. 2020. ISBN 978-3-96147-382-3.

Band 359: Christian Sand  
Prozessübergreifende Analyse komplexer  
Montageprozessketten mittels  
Data Mining  
FAPS, XV u. 168 Seiten, 61 Bilder, 12 Tab.  
2021. ISBN 978-3-96147-398-4.

Band 360: Ralf Merkl  
Closed-Loop Control of a Storage-Sup-  
ported Hybrid Compensation System for  
Improving the Power Quality in Medium  
Voltage Networks  
FAPS, xxvii u. 200 Seiten, 102 Bilder, 2  
Tab. 2021. ISBN 978-3-96147-402-8.

Band 361: Thomas Reitberger  
Additive Fertigung polymerer optischer  
Wellenleiter im Aerosol-Jet-Verfahren  
FAPS, xix u. 141 Seiten, 65 Bilder, 11 Tab.  
2021. ISBN 978-3-96147-400-4.

Band 362: Marius Christian Fechter  
Modellierung von Vorentwürfen in der  
virtuellen Realität mit natürlicher  
Fingerinteraktion  
KTmfk, x u. 188 Seiten, 67 Bilder, 19 Tab.  
2021. ISBN 978-3-96147-404-2.

Band 363: Franziska Neubauer  
Oberflächenmodifizierung und Entwick-  
lung einer Auswertemethodik zur Ver-  
schleißcharakterisierung im Presshär-  
teprozess  
LFT, ix u. 177 Seiten, 42 Bilder, 6 Tab.  
2021. ISBN 978-3-96147-406-6.

Band 364: Eike Wolfram Schäffer  
Web- und wissensbasierter Engineering-  
Konfigurator für roboterzentrierte Auto-  
matisierungslösungen  
FAPS, xxiv u. 195 Seiten, 108 Bilder, 25  
Tab. 2021. ISBN 978-3-96147-410-3.

Band 365: Daniel Gross  
Untersuchungen zur kohlenstoffdioxid-  
basierten kryogenen Minimalmengen-  
schmierung  
REP, xii u. 184 Seiten, 56 Bilder, 18 Tab.  
2021. ISBN 978-3-96147-412-7.

Band 366: Daniel Junker  
Qualifizierung laser-additiv gefertigter  
Komponenten für den Einsatz im Werk-  
zeugbau der Massivumformung  
LFT, vii u. 142 Seiten, 62 Bilder, 5 Tab.  
2021. ISBN 978-3-96147-416-5.

Band 367: Tallal Javied  
Totally Integrated Ecology Management  
for Resource Efficient and Eco-Friendly  
Production  
FAPS, xv u. 160 Seiten, 60 Bilder, 13 Tab.  
2021. ISBN 978-3-96147-418-9.

Band 368: David Marco Hochrein  
Wälzlager im Beschleunigungsfeld – Eine  
Analysestrategie zur Bestimmung des  
Reibungs-, Axialschub- und Temperatur-  
verhaltens von Nadelkränzen –  
KTmfk, xiii u. 279 Seiten, 108 Bilder,  
39 Tab. 2021. ISBN 978-3-96147-420-2.

Band 369: Daniel Gräf  
Funktionalisierung technischer Oberflächen mittels prozessüberwachter aerosolbasierter Drucktechnologie  
FAPS, xxii u. 175 Seiten, 97 Bilder, 6 Tab.  
2021. ISBN 978-3-96147-433-2.

Band 370: Andreas Gröschl  
Hochfrequent fokusabstandsmodulierte Konfokalsensoren für die Nanokoordinatenmesstechnik  
FMT, x u. 144 Seiten, 98 Bilder, 6 Tab.  
2021. ISBN 978-3-96147-435-6.

Band 371: Johann Tüchsen  
Konzeption, Entwicklung und Einführung des Assistenzsystems D-DAS für die Produktentwicklung elektrischer Motoren  
KTmfk, xii u. 178 Seiten, 92 Bilder, 12 Tab.  
2021. ISBN 978-3-96147-437-0.

Band 372: Max Marian  
Numerische Auslegung von Oberflächenmikrostrukturen für geschmierte tribologische Kontakte  
KTmfk, xviii u. 276 Seiten, 85 Bilder, 45 Tab.  
2021. ISBN 978-3-96147-439-4.

Band 373: Johannes Strauß  
Die akustooptische Strahlformung in der Lasermaterialbearbeitung  
LPT, xvi u. 113 Seiten, 48 Bilder.  
2021. ISBN 978-3-96147-441-7.

Band 374: Martin Hohmann  
Machine learning and hyper spectral imaging: Multi Spectral Endoscopy in the Gastro Intestinal Tract towards Hyper Spectral Endoscopy  
LPT, x u. 137 Seiten, 62 Bilder, 29 Tab.  
2021. ISBN 978-3-96147-445-5.

Band 375: Timo Kordaß  
Lasergestütztes Verfahren zur selektiven Metallisierung von epoxidharzbasierten Duromeren zur Steigerung der Integrationsdichte für dreidimensionale mechanische Package-Baugruppen  
FAPS, xviii u. 198 Seiten, 92 Bilder, 24 Tab.  
2021. ISBN 978-3-96147-443-1.

Band 376: Philipp Kestel  
Assistenzsystem für den wissensbasierten Aufbau konstruktionsbegleitender Finite-Elemente-Analysen  
KTmfk, xviii u. 209 Seiten, 57 Bilder, 17 Tab.  
2021. ISBN 978-3-96147-457-8.

Band 377: Martin Lerchen  
Messverfahren für die pulverbettbasierte additive Fertigung zur Sicherstellung der Konformität mit geometrischen Produktspezifikationen  
FMT, x u. 150 Seiten, 60 Bilder, 9 Tab.  
2021. ISBN 978-3-96147-463-9.

Band 378: Michael Schneider  
Inline-Prüfung der Permeabilität in weichmagnetischen Komponenten  
FAPS, xxii u. 189 Seiten, 79 Bilder, 14 Tab.  
2021. ISBN 978-3-96147-465-3.

Band 379: Tobias Sprügel

Sphärische Detektorflächen als Unterstützung der Produktentwicklung zur Datenanalyse im Rahmen des Digital Engineering

KTmfk, xiii u. 213 Seiten, 84 Bilder, 33 Tab. 2021. ISBN 978-3-96147-475-2.

## Abstract

The importance of data is increasing in all areas of life. This effect can also be observed in product development. The combination of virtual product development with the continuous and holistic use of data is referred to as «digital engineering». The implementation of digital engineering results in a transformation process and is accompanied by a change in the previous roles of people involved and the tools used. The aim is to use as much data as possible and to process this data with machine learning algorithms. In product development there is numerous geometric data (e. g. CAD models or measurement data) or data which is linked to geometry (e. g. numerical simulations and their results). In this dissertation, the method of spherical detector surfaces was developed, which makes it possible to transform arbitrary geometries into a uniform numerical matrix. The developed method can also be used to convert information associated with the geometry into further uniform matrices and thus provide this information for machine learning algorithms. The developed methodology is implemented in three different application examples and all necessary substeps are described in detail. This also includes the transformation of finite element simulations to the so called «DNA of an FE simulation».



Die Bedeutung von Daten nimmt in allen Lebensbereichen eine immer größere Rolle ein. Diese Entwicklung kann ebenso in der Produktentwicklung und Produktentstehung beobachtet werden. Die Verknüpfung der virtuellen Produktentwicklung mit der durchgängigen und ganzheitlichen Datennutzung wird als «Digital Engineering» bezeichnet. Die Umsetzung des Digital Engineering geht mit einem starken Wandel und einer Veränderung der bisherigen Rollen der beteiligten Personen und der verwendeten Werkzeuge einher. Dabei gilt es, möglichst alle zur Verfügung stehenden Daten zu nutzen und diese Daten mittels Algorithmen des Maschinellen Lernens zu verarbeiten. In der Produktentstehung existieren zahlreiche Geometriedaten (z. B. CAD Modelle oder Messdaten) oder mit einer Geometrie verknüpfte Daten (z. B. numerische Simulationen und deren Ergebnisse). Im Rahmen der vorliegenden Dissertation wurde die Methode der sphärischen Detektorflächen entwickelt, welche es ermöglicht, beliebige Geometrien in eine einheitliche numerische Matrix zu überführen. Die entwickelte Methode kann ebenfalls genutzt werden, um Informationen, die mit der Geometrie verknüpft sind, in weitere dieser einheitlichen Matrizen umzuwandeln und so Algorithmen des Maschinellen Lernens zur Verfügung zu stellen. Das entwickelte Vorgehen wird anhand von drei unterschiedlichen Anwendungsbeispielen umgesetzt und es werden alle notwendigen Teilschritte detailliert beschrieben. Dies umfasst auch die Ableitung der sogenannten «DNA einer FE-Simulation».

