

TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Robotik, Künstliche Intelligenz und Echtzeitsysteme

Präskriptive Automatisierung.
Ein optisches Prüfsystem zur Klassifizierung
und automatisierten Nacharbeit mittels
Cloud-/Edge-Computing.

Johannes Maximilian Vater

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität
München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Thomas Neumann
Prüfer der Dissertation: 1. Prof. Dr.-Ing. habil. Alois Knoll
2. Prof. Dr.-Ing. Michael Zäh

Die Dissertation wurde am 22.12.2020 bei der Technischen Universität München einge-
reicht und durch die Fakultät für Informatik am 23.06.2021 angenommen.

Abstract

Due to climate change and the related CO₂ debate, the automotive industry is currently facing the transition from combustion engines to electrically powered vehicles. However, this change is associated on the one hand with a high degree of uncertainty in the planning of production systems due to the volatile market. On the other hand, it is linked to a low level of expertise in production technologies for the production of new hybrid and electric vehicles.

An opportunity to meet the challenges described above is offered especially by the increasing digitalization and networking of production systems. As a result, novel data-driven optimization approaches, such as artificial intelligence, can be utilized. This offers great potential for error detection and classification. The advantage is that dependencies and quality deviations are not modelled on the basis of expert knowledge, instead dependencies based on data are automatically learned by the classification algorithm. Such objective data-based learning is of great advantage for the production of the electric powertrain of newly developed battery electric vehicles. Therefore, in this thesis an optical in-situ inspection system is developed, which detects and classifies quality deviations by means of a Convolutional Neural Network (CNN) (predictive analysis).

In addition to the detection of quality deviations, a noticeable added value of process optimization is only achieved by generating a recommendation for action. Thus, a rework concept suitable for the quality deviation can be generated by prescriptive analytics. Modern production processes using Industrial Internet of Things (IIoT) networks offers the possibility of continuous monitoring. However, traditional and manual quality monitoring

using IIoT networks is not easily accessible and not possible in real-time due to a lack of standards. Therefore, automatic error detection in real-time and without human intervention is not feasible for IIoT networks. For this reason, the second part of this dissertation deals with the development of a cyber-physical system. This system consists of a modular cloud-/edge architecture. Among other things, it offers the functionality of predicting a quality deviation using the CNN described above. Depending on the predicted error, the edge device can perform an automated rework without human intervention by means of a closed loop (prescriptive automation).

These systems will be prototypically designed, implemented and validated with real production data in the automotive industry based on the use case of hairpin welding. In this novel technology the traditional copper windings of the stator of an electric motor are replaced by thick copper rods. These copper rods are bent into the shape of a hairpin. One of the main steps is the welding of these copper rods. However, this process step is unstable, which often leads to defects. Furthermore, this welding process cannot be monitored and the different classes of defects cannot be classified and therefore cannot be reworked directly in the production line. By means of the CNN mentioned above, an accuracy of 99,58 percent could be achieved for the monitoring of the production process. Using the cloud-/edge architecture, a direct and automated rework of a faulty weld could be accomplished by a closed loop. A process time per hairpin weld of 0,79 seconds was achieved. This corresponds to an undercutting of the required cycle time, which highlights the potential of the system. Due to its modular design and the use of standard protocols, the solution can be easily integrated into other applications.

Zusammenfassung

Aufgrund des Klimawandels und der damit verbundenen CO₂-Debatte steht die Automobilindustrie derzeit vor dem Übergang von konventionell zu elektrisch betriebenen Fahrzeugen. Dieser Wandel ist jedoch zum einen mit einer hohen Prognoseunsicherheit aufgrund des volatilen Marktes verbunden, zum anderen auch von einem geringen Know-How der Fertigungstechnologien für die Produktion neuer Elektrofahrzeuge beeinflusst.

Eine Chance, die oben beschriebenen Herausforderungen zu bewältigen, bietet vor allem die steigende Digitalisierung und Vernetzung der Produktionsanlagen. Dabei können neuartige, datengetriebene Optimierungsansätze wie die künstliche Intelligenz genutzt werden. Sie bieten ein großes Potenzial der Fehlererkennung und -klassifizierung. Der Vorteil besteht darin, dass Abhängigkeiten und Qualitätsabweichungen nicht auf der Basis von Expertenwissen modelliert werden, sondern auf der Basis von Daten automatisch durch den Klassifikationsalgorithmus gelernt werden. Ein solches objektives, datenbasiertes Lernen ist für die Produktion des elektrischen Antriebsstrangs von großem Vorteil. Aufgrund dessen wird in der vorliegenden Arbeit ein optisches In-Situ Prüfsystem entwickelt, welches mittels eines Convolutional Neural Network (CNN) Qualitätsabweichungen detektiert und klassifiziert (prädiktive Analyse).

Neben der Erkennung der Qualitätsabweichung wird erst durch die Generierung einer Handlungsempfehlung ein merklicher Mehrwert der Prozessoptimierung erzielt. So kann mittels präskriptiver Analyse ein zur Qualitätsabweichung passendes Nacharbeitskonzept generiert werden. Moderne Produktionsprozesse bieten mithilfe Industrial Internet of Things (IIoT)-Netzwerke die Möglichkeit einer kontinuierlichen Überwachung. Jedoch ist

die traditionelle und manuelle Qualitätsüberwachung mittels Netzwerken im Bereich des IIoT aufgrund fehlender Standards nicht leicht zugänglich und nicht in Echtzeit möglich. Daher ist eine automatische Fehlererkennung in Echtzeit und ohne menschliches Zutun für IIoT-Netzwerke nicht umsetzbar. Aufgrund dessen befasst sich der zweite Teil dieser Dissertation mit der Entwicklung eines cyber-physischen Systems. Dieses besteht aus einer modular aufgebauten Cloud-/Edge-Architektur. Es bietet unter anderem die Funktionalität der Prädiktion einer Qualitätsabweichung mittels des oben beschriebenen CNN. In Abhängigkeit des prädiktierten Fehlers kann durch das Edge-Gerät mittels eines geschlossenen Regelkreises eine automatisierte Nacharbeit ohne menschlichen Eingriff durchgeführt werden (präskriptive Automatisierung).

Diese Systeme werden mit realen Produktionsdaten in der Automobilindustrie anhand des Anwendungsfalls der Hairpin-Verschweißung prototypisch konzipiert, implementiert und validiert. Bei dieser neuartigen Technologie werden die traditionellen Kupferwicklungen des Stators eines elektrischen Motors durch dicke Kupferstäbe ersetzt. Diese Kupferstäbe werden in die Form einer Haarnadel (englisch Hairpin) gebogen. Einer der Hauptschritte ist das Schweißen dieser Kupferstäbe. Dieser Prozessschritt ist jedoch instabil, wodurch häufig Fehler entstehen. Außerdem kann dieser Schweißprozess nicht überwacht werden, und die unterschiedlichen Fehlerklassen können nicht klassifiziert und daher nicht direkt in der Produktionslinie nachbearbeitet werden. Mittels des oben erwähnten CNN wurde eine Genauigkeit von 99,58 Prozent zur Überwachung des Produktionsprozesses erreicht. Durch die Cloud-/Edge-Architektur konnte eine direkte und automatisierte Nacharbeit einer fehlerhaften Verschweißung durch einen geschlossenen Regelkreis realisiert werden. Dabei wurde eine Prozesszeit pro Hairpin-Verschweißung von 0,79 Sekunden erreicht. Dies entspricht einer Unterschreitung der geforderten Taktzeit, was das Potenzial des Systems unterstreicht. Durch einen modularen Aufbau und die Verwendung von Standardprotokollen kann die Lösung einfach in andere Anwendungsfälle integriert werden.

Danksagung

Meinen ausdrücklichen Dank möchte ich meinem Doktorvater Herrn Professor Dr.-Ing. habil. Alois Christian Knoll, dem Inhaber des Lehrstuhls für Robotik, Künstliche Intelligenz und Echtzeitsysteme der Technischen Universität München, für das Ermöglichen der Promotion und das uneingeschränkte Vertrauen in meine Arbeit aussprechen. Mein Dank gilt zudem Herrn Professor Dr.-Ing. Michael Zäh, Inhaber des Lehrstuhls für Werkzeugmaschinen und Fertigungstechnik, für das Übernehmen des Koreferats sowie Herrn Professor Dr.rer.nat Thomas Neumann für die Übernahme des Vorsitzes.

Weiterhin gilt mein Dank den ehemaligen Kollegen und Kolleginnen und meinen studentischen Hilfskräften für ihre Unterstützung und Diskussionsfreudigkeit. Im Besonderen möchte ich hierbei Herrn Daniel Winkle, Herrn David Pannen, Herrn Daniel Huber und die Kollegen aus dem Werk 0 hervorheben. Frau Isabel Götz, Herrn Gregor Götz und Herrn Herbert Eicker danke ich vielmals für die sorgfältige Durchsicht dieser Arbeit.

Besonderer Dank gebührt aus meinem beruflichen Umfeld einer Person und mittlerweile guten Freund. Aus diesem Grund möchte ich mich bei Dr.-Ing. Dr. phil. Markus Droste für sein immer konstruktives und kritisches Feedback herzlich bedanken. Seine Unterstützung vonseiten BMW hat zum Erfolg dieser Arbeit wesentlich beigetragen.

Zuletzt gilt mein persönlicher Dank meinen Eltern Ina und Bruno, meinem Bruder Claudius und meiner Freundin Stephanie für deren vorbehaltlose Unterstützung, ihr Verständnis und ihre Geduld auf meinem Weg zur Promotion. In Phasen der Ernüchterung habt ihr mir immer wieder Mut gemacht und mich angetrieben. Diese Dissertation gehört daher ebenso euch wie mir, denn ohne eure Unterstützung wäre es mir nicht möglich gewesen, sie zu schreiben.

Inhaltsverzeichnis

Abkürzungsverzeichnis	XI
Nomenklatur	XIII
1 Einleitung	1
1.1 Problemstellung und Motivation	1
1.2 Zielsetzung der Arbeit	3
1.3 Aufbau der Arbeit	5
2 Grundlagen	7
2.1 Elektromobilität und elektrische Antriebe	7
2.2 Grundlagen der Produktionssysteme	8
2.2.1 Kernbegriffe von Produktionssystemen	8
2.2.2 Produktionsmanagement	9
2.2.3 IT-unterstützte Produktion	10
2.2.4 Prozessoptimierung	11
2.3 Datengetriebene Optimierung	11
2.3.1 Business Intelligence	12
2.3.2 Advanced Data Analytics	13
2.4 Prozessabsicherung durch Machine Learning und Computer Vision	14
2.4.1 Machine Learning	15
2.4.2 Computer Vision	16
2.4.3 Das künstliche Neuron	16
2.4.4 Künstliche neuronale Netze	18
2.4.5 Convolutional Neural Network	20
2.4.6 Kennzahlenbasierte Evaluierung des Klassifikationsalgorithmus	23

INHALTSVERZEICHNIS

2.4.7	Visuelle Evaluierung des Klassifikationsalgorithmus	27
2.4.8	Finale Evaluierung des Modells	28
2.5	Industrie 4.0	33
2.5.1	Cyber-physische Systeme	34
2.5.2	Internet der Dinge	34
3	Stand der Technik	39
3.1	Optische Erkennung von Qualitätsabweichungen mittels CNN	39
3.1.1	Optische Erkennung von Qualitätsabweichungen in der Produktionsumgebung mittels CNN	40
3.1.2	Abgrenzung zur Literatur	43
3.2	Echtzeitfähige Edge-Computing-Lösung	45
3.2.1	Umsetzung der echtzeitfähigen Edge-Computing-Lösung in der Produktionsumgebung	45
3.2.2	Abgrenzung zur Literatur	47
4	Präskriptive Prozessoptimierung Methode	51
4.1	Vorgehensweise	51
4.1.1	Problemlösezyklus als Mikrozyklus	51
4.1.2	Das V-Modell als Makrozyklus	52
4.2	Anforderungen an die prädiktive und präskriptive Analyse	54
4.2.1	Allgemeine Anforderungen	54
4.2.2	Anforderungen an den Klassifikationsalgorithmus	56
4.2.3	Anforderungen an die IT-Architektur	57
4.3	Mechatronisches System als Grundmodell	59
5	Modellierung eines CNN zur Erkennung von Prozessabweichungen	61
5.1	Methodik	61
5.1.1	Business Understanding	62
5.1.2	Datengenerierung	62
5.1.3	Datenvorverarbeitung	64
5.1.4	Modellierung des Convolutional Neural Network	65
5.1.5	Kennzahlenbasierte Evaluierung des Klassifikationsalgorithmus	66
5.1.6	Visuelle Evaluierung des Klassifikationsalgorithmus	66

5.1.7	Finale Evaluierung des Modells	67
5.1.8	Präskriptive Analyse	67
5.2	Business Understanding	70
5.2.1	Ist-Produktionsprozess des Stators	71
5.2.2	Soll-Produktionsprozess des Stators	73
5.3	Datengenerierung	74
5.4	Datenvorverarbeitung	77
5.5	Aufbau der Struktur des Convolutional Neural Network	81
5.5.1	Architektur des Netzes	82
5.5.2	Trainingsprozess des Convolutional Neural Network	84
5.6	Validierung des Modells	87
5.6.1	Kennzahlenbasierte Validierung	87
5.6.2	Visuelle Validierung	91
5.7	Sensitivitätsanalyse durch eine Parametervariation	93
5.7.1	Parametervariation: Aufbau des Convolutional Neural Network	93
5.7.2	Parametervariation: Daten	99
5.8	Evaluierung des finalen Modells	103
5.9	Handlungsempfehlung zur Nacharbeit durch präskriptive Analytik	106
6	Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur	109
6.1	Ermitteln des groben Systementwurfes	109
6.1.1	Mechatronisches System als Grundmodell	109
6.1.2	Ableiten einer Systemarchitektur	111
6.2	Komponenten des Edge-Geräts	113
6.2.1	Kommunikation der auf dem Edge-Gerät lokalisierten Module	113
6.2.2	Module der Verarbeitungsebene	115
6.2.3	Module der cyber-physischen Schnittstelle	116
6.3	Komponenten der Cloud-Ebene	117
6.3.1	Datenspeicher	118
6.3.2	Organisationseinheit	119
6.3.3	Visualisierungsmodul	119

INHALTSVERZEICHNIS

7 Technische Umsetzung des Verfahrens	121
7.1 Voraussetzungen der Hard- und Softwarekomponenten	121
7.2 Komponenten des Edge-Gerätes	122
7.2.1 Module der Verarbeitungsebene	122
7.2.2 Module der cyber-physischen Schnittstelle	126
7.3 Komponenten der Cloud	127
7.3.1 Datenspeicher	127
7.3.2 IIoT-Verwaltung	128
7.3.3 Visualisierungsmodul	128
7.4 Ergebnisse	130
7.5 Anforderungsbezogene Bewertung	132
7.5.1 Anforderungsbezogene Bewertung: Allgemeine Anforderungen . .	132
7.5.2 Anforderungsbezogene Bewertung: Anforderungen an den Klassifikationsalgorithmus	134
7.5.3 Anforderungsbezogene Bewertung: Anforderungen an die IT-Architektur	135
8 Schlussbetrachtung	139
8.1 Zusammenfassung	139
8.2 Ausblick	141
Anhang	145
A Dokumentationen Experteninterviews	147
B Vergleich der unterschiedlichen CNN-Netzwerke	161
C Evaluierung der Netzarchitektur	163
D Visualisierung der Web-Applikation	165

Abbildungsverzeichnis

1.1	Forschungsziele (FZ _x der vorliegenden Arbeit).	4
1.2	Strukturierung und Aufbau der Arbeit.	6
2.1	Produktionssystem bzw. -prozess in Anlehnung an Steven [24].	8
2.2	Automatisierungspyramide in Anlehnung an Heinrich [26, S. 10].	10
2.3	Methoden der Prozessoptimierung nach Becker [25, S. 21].	12
2.4	Übersicht Analysehorizonte Data Analytics [40, S. 10].	13
2.5	Zusammenhang zwischen Computer Vision, Machine Learning, Human Vision und einem CNN [52, S. 7].	15
2.6	Generische Darstellung eines Neurons [59, S. 27].	17
2.7	Beispielhafte grafische Darstellung eines 2-2-1 Multi-Layer-Perceptrons [59, S. 50].	18
2.8	Typischer Aufbau der Schichten eines CNN [70, S. 336].	20
2.9	Beispielhafte Abbildung der Faltung durch die Kreuzkorrelation [70, S. 330].	22
2.10	Beispielhafte Aufteilung eines Datensatzes in Trainings-, Validierungs- und Testdatensatz.	23
2.11	Schematische Darstellung einer 5-fold Cross-Validation [55, S. 24].	24
2.12	Darstellung einer Confusion Matrix nach Krüger [74].	25
2.13	Anwendung des Grad-CAM- (links) und ISCSV-Verfahrens (rechts) auf die Klassifizierung einer handgezeichneten 5 (mitte).	27
2.14	Beispielhafte Wahrscheinlichkeitsdichte und kumulative Wahrscheinlich- keitsdichte einer Binomialverteilung [80, S. 188].	30
2.15	Fehleranteile im Produktionsprozess.	31
2.16	Darstellung des Hypothesentests und dessen vier Fälle bei der Durchfüh- rung eines Stichprobentests mit einer Annahmegrenze [83, S. 103].	32

ABBILDUNGSVERZEICHNIS

2.17	Datenaustausch von Modulen über einen Nachrichtenbroker.	37
4.1	V-Modell nach VDI 2206 [156, S. 29].	53
4.2	Anforderungen an die präskriptive Fertigungsoptimierung.	54
4.3	Grundstruktur eines mechatronischen Systems [156, S. 14].	59
5.1	Vorgehensweise zur automatisierten Fehlererkennung und Nacharbeit mit- hilfe von präskriptiver Analyse.	62
5.2	Magisches Dreieck nach Kuhlant et al. [188].	65
5.3	Komponenten der präskriptiven Fertigungsprozessanalyse eines laufen- den Prozessauftrags in Anlehnung an Gröger [31, S. 159].	68
5.4	Darstellung eines generischen Ist-Fertigungsprozesses.	69
5.5	Darstellung eines Soll-Fertigungsprozesses durch eine Kombination aus künstlicher Intelligenz und präskriptiver Fertigungsprozessanalyse.	69
5.6	Darstellung eines Hairpins nach Glässel [193, S. 2].	70
5.7	Ist-Prozess der aktuellen Statorfertigung.	71
5.8	Soll-Prozess der aktuellen Statorfertigung.	73
5.9	Mögliche resultierende Klassen bei der Verschweißung von Hairpins.	74
5.10	Versuchsträger zur Verschweißung von Hairpin-Paaren.	76
5.11	Aufnahme und Speicherung der 3D-Daten und Schwarz-Weiß-Bilder.	76
5.12	RBG-Rohdatenbild mit codierten Höheninformationen.	78
5.13	Erster Schritt der Datenvorverarbeitung: Ausschneiden und orthogonale Rotation des Ausgangsbildes.	79
5.14	Zweiter Schritt der Datenvorverarbeitung: Resultierende Höheninforma- tionen bei Umwandlung der Farbcodierung.	79
5.15	Dritter Schritt der Datenvorverarbeitung: Ausschnitt der Schweißkuppel durch Berechnung des Medians und Festlegung des darum liegenden Be- reiches.	80
5.16	Vierter Schritt der Datenvorverarbeitung: Reduktion und Skalierung des Höhenprofils der Hairpin-Verschweißung.	80
5.17	Fünfter Schritt der Datenvorverarbeitung: Konvertieren des Bildes in ein Graustufenbild.	81
5.18	Sechster Schritt der Datenvorverarbeitung: Normalisierung des Daten- satzes.	81

5.19 Aufbau des Convolutional-Blocks.	83
5.20 Entworfenes CNN-Modell für den in dieser Arbeit betrachteten Anwendungsfall.	86
5.21 Resultierender Trainings- und Validierungsverlauf der Kosten und der Genauigkeit eines Trainings der 5-fold Cross-Validation der definierten CNN-Architektur.	88
5.22 Vergleich der Anteile an falsch erkannten Prädiktionen und dem Auftreten falscher unsicherer Entscheidungen in Abhängigkeit des dafür festgelegten Schwellwerts.	90
5.23 Anwendung von Grad-CAM (links) und ISCSV (rechts) auf das vorverarbeitete Eingangsbild für das CNN für alle relevanten Klassen sowie deren Abstufungen.	92
5.24 Resultierender Verlauf der Genauigkeit der Validierungsdaten während des Trainingsprozesses bei Variation der Filtergröße.	94
5.25 Einfluss der Anzahl der Convolution-Blöcke auf die Genauigkeit und Prädiktionszeit.	95
5.26 Einfluss der Anzahl der Filter auf die Genauigkeit und Prädiktionszeit.	96
5.27 Einfluss der Größe des Trainingsdatensatzes auf die resultierende Genauigkeit des Validierungsdatensatzes.	100
5.28 Verlauf der Kosten der Trainingsdaten und der Validierungsdaten während des Trainingsprozesses des CNN für unterschiedliche Normalisierungstechniken.	101
5.29 Einfluss der Dropout-Rate auf die resultierende Genauigkeit des Validierungsdatensatzes.	102
5.30 Binomialverteilung des Modells mit der internen und externen Qualität und der dazugehörigen Fehlklassifizierungsraten und Stichprobengröße $n = 160$	104
5.31 Resultierende CM des Testdatensatzes.	105
5.32 Schematisches Vorgehen zur Generierung der Handlungsempfehlung mithilfe der präskriptiven Analyse.	106
5.33 Entscheidungsbaum mit den zugehörigen Nacharbeiten und Ereignissen.	107

ABBILDUNGSVERZEICHNIS

6.1	Grundstruktur eines mechatronischen Systems bezogen auf den Anwendungsfall.	110
6.2	Grundstruktur des mechatronischen Systems nach VDI-Richtlinie 2206 mit implementierter Edge-/Cloud-Architektur.	111
6.3	Konzeptionelle Darstellung der Komponenten der Edge-Ebene und ihrer Verbindungen.	114
6.4	Datenaustausch von Modulen über den IoT-Edge-Hub.	114
6.5	Konzeptionelle Darstellung der Komponenten der Cloud-Ebene.	118
7.1	Ablaufdiagramm des Datenvorverarbeitungsmoduls.	123
7.2	Ablaufdiagramm des Prädiktionsmoduls.	124
7.3	Ablaufdiagramm des Entscheidungsmoduls.	124
7.4	Ablaufdiagramm des Validierungsmoduls.	125
7.5	Startseite der Web-Applikation.	129
7.6	Zeitmesspunkt der jeweiligen Verarbeitungsschritte.	130
C.1	Confusion Matrix zur Evaluierung der Netz-Architektur mit 3D-Bildern als Eingangsgröße.	163
C.2	Confusion Matrix zur Evaluierung der Netz-Architektur mit Schwarz-Weiß-Bildern als Eingangsgröße.	163
D.1	Übersicht aktueller Informationen einer Verschweißung.	165
D.2	Auswahl Klasse in Ordnung und nicht in Ordnung.	166
D.3	Auswahl der Subklassen nicht in Ordnung.	166

Tabellenverzeichnis

3.1	Erfüllung der Anforderungen vergleichbarer Literatur im Bereich der Qualitätssicherung mittels eines CNN.	43
3.2	Erfüllung der Anforderungen vergleichbarer Literatur im Bereich des Edge-Computings.	48
5.1	Gesamtanzahl aufgenommener Verschweißungen pro Klasse.	77
5.2	Anzahl der originalen und resultierenden, synthetischen Trainingsdaten je Klasse durch Anwendung von DA.	78
5.3	Ergebnisse Validierung für die Läufe 1 bis 5 mit 3D-Bildern.	89
5.4	Ergebnisse Validierung für die Läufe 1 bis 5 mit Schwarz-Weiß-Bildern.	89
5.5	Beste 5 Ergebnisse bezogen auf die Genauigkeit ($Kosten = 0 * p_{PZ} + 0 * p_{BG} + 1 * p_{iQ}$).	98
5.6	Beste 5 Ergebnisse bezogen auf die Kostenfunktion: $Kosten = 0,4 * p_{PZ} + 0 * p_{BG} + 0,6 * p_{iQ}$	98
5.7	Besten 5 Ergebnisse bezogen auf die Kostenfunktion: $Kosten = 0,4 * p_{PZ} + 0,2 * p_{BG} + 0,4 * p_{iQ}$	99
5.8	Resultierende Ergebnisse basierend auf der CM in Abbildung 5.31 bei Klassifizierung des finalen Testdatensatzes.	105
7.1	Statistische Auswertung der Zeitdifferenz der Zeitstempel.	131
7.2	Gegenüberstellung der allgemeinen Anforderungen zu den Lösungen.	133
7.3	Gegenüberstellung der Anforderungen des Klassifikationsalgorithmus zu den Lösungen.	134
7.4	Gegenüberstellung der Anforderungen der IT-Architektur zu den Lösungen.	136
A.1	Leitfragenkatalog Experteninterviews.	148

TABELLENVERZEICHNIS

A.2	Dokumentation Experteninterview 1.	149
A.3	Dokumentation Experteninterview 2.	150
A.4	Dokumentation Experteninterview 3.	151
A.5	Dokumentation Experteninterview 4.	152
A.6	Dokumentation Experteninterview 5.	153
A.7	Dokumentation Experteninterview 6.	154
A.8	Dokumentation Experteninterview 7.	155
A.9	Dokumentation Experteninterview 8.	156
A.10	Dokumentation Experteninterview 9.	157
A.11	Dokumentation Experteninterview 10.	158
A.12	Dokumentation Experteninterview 11.	159
A.13	Dokumentation Experteninterview 12.	160
B.1	Vergleich der unterschiedlichen CNN-Netzwerke.	161

Abkürzungsverzeichnis

Abkürzung	Beschreibung
ADA	Advanced Data Analytics
API	Application Programming Interface
BI	Business Intelligence
BN	Batch Normalization
CI/CD	Continuous Integration und Continuous Deployment
CM	Confusion Matrix
CNN	Convolutional Neural Network
CPI	Cyber-Physisches Interface
CPPS	Cyber-physisches Produktionssystem
CPS	Cyber-Physisches System
CRISP-DM	CRoss-Industry Standard Process for Data Mining
DA	Data Augmentation
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
GD	Gradient Descent
Grad-CAM	Gradient-weighted Class Activation Mapping
HMI	Human Machine Interface
HOG	Histogram of Oriented Gradients
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IIoT	Industrial Internet of Things
iO	in Ordnung
IoT	Internet of Things
ISCSV	Image-Specific Class Saliency Visualisation
KDD	Knowledge Discovery Database
KV	Korrekte Verschweißung
KVP	Kontinuierlicher Verbesserungsprozess
Mean	Mittelwert
MES	Manufacturing Execution System
ML	Maschinelles Lernen

Abkürzungsverzeichnis

Abkürzung	Beschreibung
MLP	Multi-Layer-Perceptron
MNIST	Modified National Institute of Standards and Technology database
MQTT	Message Queuing Telemetry Transport
niO	nicht in Ordnung
NN	Neuronales Netz
OPC UA	Open Platform Communications United Architecture
PaaS	Platform as a Service
ReLU	Rectified Linear Unit
REST	Representational State Transfer
RGB	Rot, Grün und Blau
SaaS	Software as a Service
SD	Standardabweichung
SIFT	Scale-Invariant Feature Transform
SK	Schweißkrater
SPS	Speicherprogrammierbare Steuerung
SQL	Structured Query Language
SS	Schweißspritzer
SURF	Speeded Up Robust Features
TN	True Negative
TP	True Positive
UE	Unsichere Entscheidung
UV	Unzureichende Verschweißung
VDI	Verein Deutscher Ingenieure
VDMA	Verband Deutscher Maschinen- und Anlagenbau
VGG	Visual Geometry Group
ZVEI	Zentralverband Elektrotechnik- und Elektronikindustrie

Nomenklatur

Formelzeichen	Bedeutung
$a_i^{[l]}$	Gewichteter Ausgang nach Aktivierung des i -ten Neurons aus der l -ten Schicht
$accuracy$	Metrik zur Bewertung eines Klassifikators
b	Offset bzw. negativer Schwellwert eines Neurons
$b_i^{[l]}$	Schwellwert des i -ten Neurons aus der l -ten Schicht
c	Anzahl der Klassen
C_{pk}	Prozessfähigkeitsindex
F	zweidimensionaler Filter
F_β	Metrik zur Bewertung eines Klassifikators
$g(\cdot)$	Aktivierungsfunktion eines Neurons
$g^{[l]}(\cdot)$	Aktivierungsfunktion aller Neuronen aus der l -ten Schicht
acc_{gesamt}	Metrik zur Bewertung eines Klassifikators
g_i	Gewichtungen der Kostenfunktion
I	zweidimensionales Eingangsbild
k	Annahmegrenze der erlaubten Fehlklassifizierungen
n	Größe Stichprobensatz
p_{BG}	Parameter Bildgröße
p_{defekt}	Anzahl fehlerhafter Bauteile im Produktionsprozess
p_{eQ}	Fehlklassifizierungsrate des Klassifikationsalgorithmus bezogen auf die externe Qualität
p_f	relativer Anteil der Fehlklassifizierungen des Algorithmus
p_{iQ}	Fehlklassifizierungsrate des Klassifikationsalgorithmus bezogen auf die interne Qualität
p_k	kumulierte Binomialverteilung
p_{max_defekt}	maximal akzeptabler Fehlerschlupf
p_{PZ}	Parameter Prädiktionszeit
$precision$	Metrik zur Bewertung eines Klassifikators
p_x	Wahrscheinlichkeitsfunktion der Binomialverteilung
$recall$	Metrik zur Bewertung eines Klassifikators
S	Ergebnis der Faltung
S_N	Anzahl der Fertigungsschritte

Nomenklatur

Formelzeichen	Bedeutung
T_i	Kommunikationszeiten zwischen den Modulen
w	Gewichtung eines Eingangs zu einem Neuron
$w_{ij}^{[l]}$	Gewichtung des Signals vom j -ten Neuron aus der $(l-1)$ -ten Schicht zum i -ten Neuron in der l -ten Schicht
x	Eingang eines Neurons
y	Ausgang eines Neurons nach Aktivierung
z	Aufsummierte und gewichtete Eingänge eines Neurons vor Aktivierung
$z_i^{[l]}$	Aufsummierter Eingang des i -ten Neurons aus der l -ten Schicht
α	Fehler 1. Art bei einem Stichprobentest
β	Fehler 2. Art bei einem Stichprobentest
θ	Schwellwert

Kapitel 1

Einleitung

1.1 Problemstellung und Motivation

„Die Leute werden in einigen Jahren auf Benzinmotoren zurückblicken so wie wir heute auf Dampfmaschinen und sagen: Es war eine gute Zeit, aber sie ist vorbei.“ Elon Musk, CEO Tesla Motors [1]

Die traditionelle Fertigungsindustrie ist ein zentraler Bestandteil der weltweiten Wirtschaft. Der Anteil des verarbeitenden Gewerbes an der Bruttowertschöpfung in Deutschland belief sich im Jahr 2019 auf 24,2 Prozent [2]. Die deutsche Bruttowertschöpfung dieses Wirtschaftszweiges stieg von 605 im Jahr 2009 auf 921 Milliarden Euro im Jahr 2019. Der Kraftfahrzeugbau trägt dabei maßgeblich zur gesamtwirtschaftlichen Entwicklung in Deutschland bei [3]. Er hat einen Anteil von 5 Prozent an der gesamtwirtschaftlichen Bruttowertschöpfung, 3 Prozent der sozialversicherungspflichtigen Beschäftigten sind im Kraftfahrzeugbau tätig und 17 Prozent der deutschen Warenexporte entfallen auf diesen Wirtschaftszweig [4]. Seit 2013 unterzieht er sich jedoch einem starken Wandel. 2013 hat Tesla sein vollelektrisches Model S vorgestellt, welches die Langstreckentauglichkeit eines elektrisch angetriebenen Fahrzeugs bestätigte. Unter anderem führte dies zur Einführung eines Flottenverbrauches durch die Europäische Union [5]. Im September 2015 wurde öffentlich bekannt gegeben, dass die Volkswagen AG eine illegale Abschaltvorrichtung in der Motorsteuerung ihrer Diesel-Fahrzeuge verwendete, um US-amerikanische Abgasnormen zu umgehen [6]. Hinzu kommt, dass 2019, insbesondere durch den Einsatz von Greta Thunberg, die Klimaerwärmung und somit der CO₂-Ausstoß auf die Agenda vieler Länder und ihrer Staatschefs gerückt ist. Aus

1. Einleitung

diesem Grund hat die Europäische Union beschlossen, den CO₂-Ausstoß bis 2050 um 80 Prozent zu reduzieren [7]. In Folge dessen fordert die Politik, den Durchbruch der Elektromobilität zu beschleunigen.

Der Wandel vom verbrennungsmotorischen Antrieb hin zu elektrischen Antriebssystemen ist jedoch zum einen mit einer hohen Prognoseunsicherheit in der Planung von Produktionssystemen aufgrund des volatilen Marktes und zum anderen mit einem geringen Know-How der Fertigungstechnologien zur Produktion neuer Hybrid- und Elektrofahrzeuge verbunden [8]. Vor allem das geringe Expertenwissen erschwert es, Qualitätsabweichungen zu erkennen, was wiederum eine verstärkte und aktive Prozessabsicherung mittels Prüfsystemen erfordert [9]. Nicht detektierte Fehler könnten zum Kunden geraten, was zu einer Panne oder im schlimmsten Fall zur Gefahr von Mensch und Umwelt führen kann. Dies stellt die Produktion als Schlüsseltechnologie vor große Herausforderungen.

Die zukunftsweisende Produktion des elektrischen Antriebstrangs erfordert daher in den Produktionsprozess integrierte Prüfprozesse, um Qualitätsabweichungen zu erkennen. Vor allem optische Prüfsysteme haben den Vorteil, dass sie sich einfach in bestehende Prozesse integrieren lassen [10]. Bis 2023 werden 65 Prozent der von Unternehmen aufgezeichneten Videos und Bilder von Maschinen und nicht mehr von Menschen analysiert, was einen Anstieg um 70 Prozent seit 2018 darstellt [11]. Das ist vor allem dem Erfolg der künstlichen Intelligenz im Bereich von Computer Vision zu verdanken, denn Algorithmen in diesem Bereich können das menschliche Sehen besser als traditionelle Algorithmen simulieren. Darüber hinaus können dadurch Videos und Bilder schneller analysiert werden, was im Produktionsbereich wichtig ist [12]. Ein weiterer Vorteil von maschinellem Lernen ist, dass der Algorithmus Abhängigkeiten und Merkmale durch vorhandene Daten selbstständig erlernt, was einen großen Vorteil im Bereich der Produktion elektrischer Antriebssysteme ist [13]. Bei der heutigen Prozessoptimierung gilt es jedoch nicht mehr, die Prozessleistung ex post historisch zu untersuchen und daraus reaktive Anpassungen abzuleiten, sondern bereits im laufenden Produktionsprozess Prozessanomalien mittels prädiktiver Analyse zu prognostizieren und proaktiv und automatisiert Gegenmaßnahmen einzuleiten [14].

Die entsprechenden Gegenmaßnahmen müssen jedoch unter den gegebenen Echtzeitanforderungen ausgeführt werden. Neuartige Technologien des verteilten Rechnens bieten gute Voraussetzungen für eine echtzeitfähige Optimierung der Produktion. Edge-Computing ermöglicht eine Bereitstellung von Funktionalität in der Nähe des Prozesses. Eine direkte Anbindung von Komponenten der Produktionsanlage und Verarbeitung von Funktionalität in Echtzeit wird möglich. Konzepte, die die Vorhersage der Qualität produzierter Bauteile ermöglichen, bieten eine Gelegenheit zur Reduktion des Ausschusses und der Nacharbeitszeiten. Die nachträgliche Integration benötigter Hard- und Software in bereits bestehende Produktionsanlagen ist jedoch mit großem Aufwand verbunden. Aufgrund dieser Schwachstellen ergeben sich die in Abschnitt 1.2 beschriebenen Ziele.

1.2 Zielsetzung der Arbeit

Wie die in Abschnitt 1.1 skizzierte Vision der Produktion und dargelegte Motivation der vorliegenden Arbeit verdeutlicht, ist es wichtig, Qualitätsabweichungen in der Produktionslinie frühzeitig zu erkennen und Gegenmaßnahmen einzuleiten. Vor allem in der Produktion des elektrischen Antriebstrangs ist es essenziell, dafür neue, datengetriebene Modelle, wie ein Convolutional Neural Network (CNN), zu verwenden. Darüber hinaus soll in Abhängigkeit der prädiktierten Prozessabweichung eine daraus resultierende Handlungsempfehlung generiert werden. Diese Handlungsempfehlung soll im Anschluss automatisiert, das heißt ohne menschliches Eingreifen, umgesetzt werden. Ziel der vorliegenden Arbeit ist somit die Konzeption, die prototypische Implementierung und die Evaluierung eines integrierten Systems zur Optimierung eines Produktionsprozesses. Im Folgenden sind die daraus abgeleiteten drei Forschungsziele dargelegt (vgl. Abbildung 1.1):

Forschungsziel FZ₁: Optischer Prüfprozess mithilfe prädiktiver Analyse

Die optische Qualitätsüberprüfung in der Serienproduktion erfordert hohe Anforderungen an die Güte des datengetriebenen Klassifikationsalgorithmus und an dessen Geschwindigkeit. Denn durch die Qualitätssicherung darf die Taktzeit des Produktionsprozesses nicht negativ beeinflusst werden. Erstes Forschungsziel ist somit die Modellierung einer geeigneten Struktur des Klassifikationsalgorithmus für die optische Inline-Prüfung

1. Einleitung

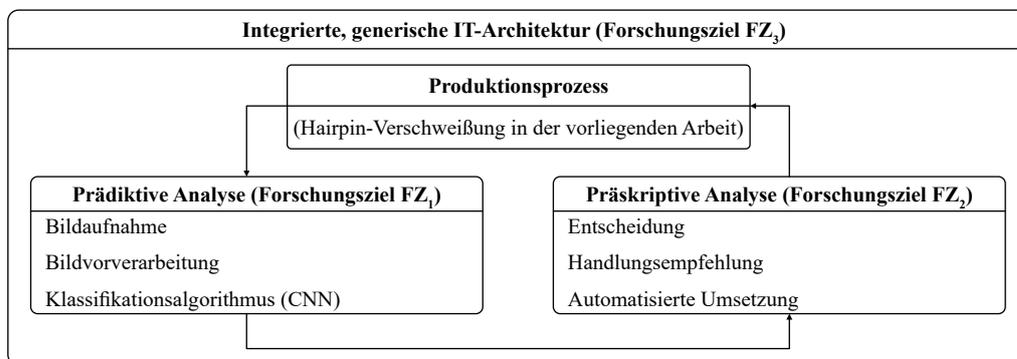


Abbildung 1.1: Forschungsziele (FZ_x der vorliegenden Arbeit).

in einer Serienfertigung. Diese soll an die entsprechenden Anforderungen adaptierbar sein. Des Weiteren muss die Güte der verwendeten Struktur des Algorithmus in Bezug auf Prüfvorschriften der Automobilindustrie validiert werden.

Forschungsziel FZ₂: Präskriptive Analyse und deren Umsetzung ohne menschliches Eingreifen

Derzeit existierende Advanced Data Analytics (ADA)-Anwendungen in der Fertigung basieren zum Großteil auf deskriptiven bzw. prädiktiven Analyseverfahren. Ein geringer Anteil beschäftigt sich mit präskriptiven Analysen. Ein starker Nachteil dabei ist zudem, dass die durch die präskriptive Analyse generierte Handlungsempfehlung durch das manuelle Eingreifen eines Menschen umgesetzt wird. Dieser muss anschließend manuell in den Prozess eingreifen. Ziel ist somit die automatisierte Umsetzung der durch die präskriptive Analyse definierten Handlungsempfehlung mithilfe eines geschlossenen Regelkreises. Wichtig dabei ist, dass diese ohne das Eingreifen von Menschen umgesetzt wird.

Forschungsziel FZ₃: Generische und integrierte IT-Architektur

Die Heterogenität der Produktionsprozesse und IT-Systeme in den verschiedenen Branchen und Unternehmen hat zur Folge, dass ADA-Anwendungen in der Produktion üblicherweise auf Einzelfallbasis realisiert werden. Die Folge ist eine hohe Umsetzungs-komplexität und ein geringer Grad der Wiederverwendung, wodurch die Implementierungskosten von ADA-Anwendungen in der Produktion steigen. Ziel ist es daher, ein ADA-System zu schaffen, das für die datengetriebene Optimierung notwendigen kon-

zeptionellen Elemente vereint und auf einem allgemeingültigen Ansatz aufsetzt. Angestrebt wird die Entwicklung allgemeiner Analyseverfahren und Schnittstellen, um die Komplexität der Implementierung im individuellen Fall zu senken. Darüber hinaus soll die im Forschungsziel FZ₂ formulierte Handlungsempfehlung durch die IT-Architektur entsprechend der geforderten Echtzeitanforderung umgesetzt werden.

1.3 Aufbau der Arbeit

Zum Erreichen der im vorigen Abschnitt festgelegten Ziele wurde für die vorliegende Arbeit eine Gliederung gewählt, die, wie in Abbildung 1.2 dargestellt, in acht inhaltliche Kapitel aufgeteilt ist.

Kapitel 2 beinhaltet die konzeptionellen Grundlagen, die für das Verständnis notwendig sind. Dies umfasst die Grundlagen eines Produktionssystems, dessen IT-unterstützte Produktion und Prozessoptimierung. Ausgehend davon werden IT-Konzepte zur datengetriebenen Optimierung im Bereich Business Intelligence und ADA vorgestellt. Ein Modell im Bereich ADA sind sogenannte neuronale Netze (NN), zu denen CNN gehören. Deren Grundlagen werden ebenfalls in Kapitel 2 erläutert. Die Ausführung von CNN im Produktionsbereich benötigt jedoch neue IT-Architekturkonzepte. Aus diesem Grund werden die Grundlagen von Industrie 4.0 beschrieben.

Ausgehend von diesen Grundlagen werden in *Kapitel 3* verwandte Arbeiten, speziell im Bereich der optischen Erkennung von Qualitätsabweichungen in der Automobilproduktion, mittels eines CNN und der Anwendung von Edge- und Cloud-Computing im produzierenden Umfeld analysiert. Anhand der identifizierten Schwachstellen wird anschließend eine wissenschaftliche Lücke abgeleitet.

Aufbauend auf diesen Schwachstellen wird in *Kapitel 4* eine Methode zur Entwicklung eines Systems vorgestellt. Durch diese können Qualitätsabweichungen erkannt und anschließend eine davon abhängige Handlungsempfehlung automatisiert generiert werden.

1. Einleitung

In *Kapitel 5* wird zuerst eine Übersicht gängiger Methoden zur Vorverarbeitung von Rohdaten, zur Modellierung eines CNN, zur Validierung dieses Modells und zur Implementierung der präskriptiven Analytik im Allgemeinen beschrieben. Anschließend wird dieser methodische Ansatz auf den in diesem Beitrag betrachteten Produktionsprozess des Hairpin-Schweißens übertragen. Dabei wird eine eigene Architektur eines CNN entwickelt und bewertet, um Qualitätsabweichungen optisch erkennen zu können.

Das Ausführen des entwickelten CNN unter harten Echtzeitanforderungen im Produktionsprozess benötigt eine spezielle IT-Architektur. Aus diesem Grund wird in *Kapitel 6* eine solche IT-Architektur konzeptioniert, welche in den Produktionsprozess integriert werden kann. Diese Architektur basiert auf dem sogenannten Cloud- und Edge-Computing.

In *Kapitel 7* werden das aus Kapitel 5 entwickelte CNN und die IT-Architektur (vgl. Kapitel 6) miteinander vereint. Dieses System wird final technisch an dem in dieser Arbeit betrachteten Anwendungsfall exemplarisch angewandt und dessen Funktionalität evaluiert.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in *Kapitel 8*.

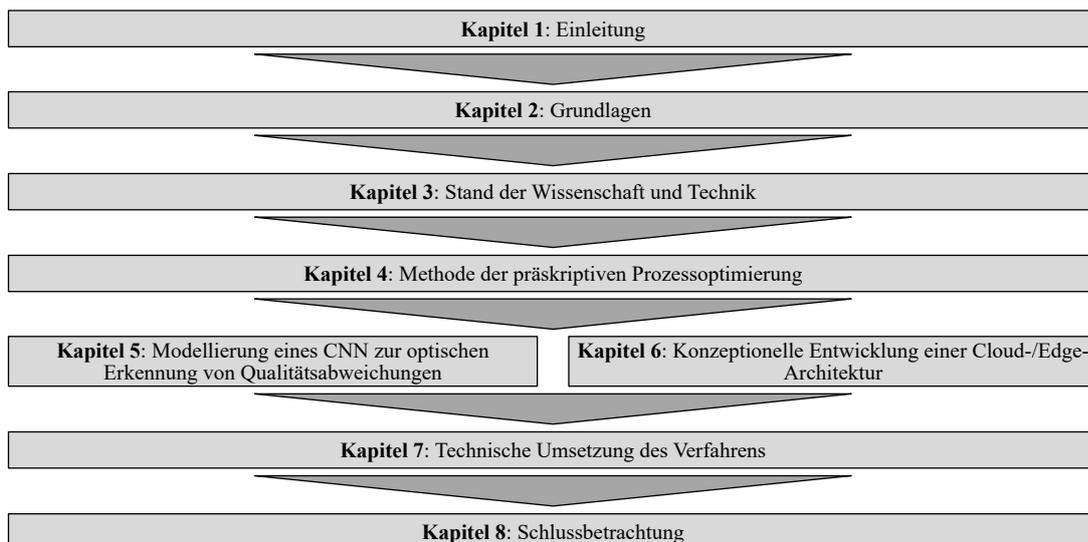


Abbildung 1.2: Strukturierung und Aufbau der Arbeit.

Kapitel 2

Grundlagen

2.1 Elektromobilität und elektrische Antriebe

Die Automobilindustrie durchläuft einen Umbruch der Antriebskonzepte und steht somit vor einer Dekade fundamentaler Veränderungen, da zukünftig die Elektromobilität im Individualverkehr eine signifikante Rolle einnehmen wird [15, S. 469]. Gründe hierfür sind verschärfte Emissionsauflagen sowie eine zunehmende gesellschaftliche Bedeutung von Umweltschutz und Energieeffizienz [16, S. 21]. Zusätzlich veranlasst die CO₂-Debatte die Automobilindustrie, die Entwicklung von elektrisch angetriebenen Fahrzeugen zu forcieren [8, S. 227], [17, S. 17–18]. Zudem schaffen das disruptive Umfeld der Elektromobilität und ihre Technologien neue Herausforderungen [18, S. 21–30]. Das Wissen auf dem Gebiet der konventionell angetriebenen Fahrzeuge wird durch die Expertise in den Bereichen der Batterieherstellung und der elektrischen Maschine substituiert. Folglich können Unternehmen anderer Branchen als neue Mitstreiter in den Automobilmarkt eintreten und den Automobilherstellern einen Teil der Wertschöpfung abnehmen. Zudem stehen die Kosten für die Produktion von Elektrofahrzeugen in direktem Wettbewerb mit denen von konventionellen Fahrzeugen [8, S. 8].

Allerdings bringt dieser Wandel eine profunde Veränderung der Fahrzeugstruktur sowie der genutzten Fahrzeugkomponenten mit sich. Diese Entwicklung hat tiefgreifende Konsequenzen für die Produktionstechnik, da sich die Bestandteile des Elektroantriebes signifikant von denen eines konventionellen Antriebssystems differenzieren. Von besonderer Relevanz für die Produktionstechnik des Elektrofahrzeugs sind der Elektromotor,

2. Grundlagen

die Traktionsbatterie als auch die Leistungselektronik, welche einen maßgeblichen Anteil an den Herstellungskosten ausmachen. Um diesen Herausforderungen gerecht zu werden, sind innovative Produktionssysteme zu entwickeln und gleichzeitig die Kosten zu senken [19, S. 139]. Dies schließt beispielsweise die Erweiterung des Verständnisses der Wertschöpfung und eine stärkere Vernetzung der Wertschöpfungsbeiträge ein [20, S. 57], [21, S. 7].

Um die Herausforderungen der Produktion des elektrischen Antriebsstranges besser zu verstehen, werden im folgenden Abschnitt die Produktion, deren Bestandteile und Optimierung erläutert. Außerdem wird dabei die in dieser Arbeit betrachtete Produktionsform und -optimierung von anderen abgegrenzt.

2.2 Grundlagen der Produktionssysteme

In Rahmen dieses Abschnittes werden zu Beginn die Grundlagen der Produktion aufgezeigt. Dies schließt die wesentlichen Begriffe der Produktion, die essenziellen Geschäftsprozesse und die produktionsunterstützenden IT-Systeme ein.

2.2.1 Kernbegriffe von Produktionssystemen

Nach Dangelmaier beschreibt der Begriff Produktion „die Transformation von Materialien, Diensten, Rechten und Informationen“ [22]. Der in diesem Zusammenhang geschilderte Transformationsprozess ist in der Literatur als ein Input-Throughput-Output-Prozess beschrieben. Entsprechend lassen sich, wie in Abbildung 2.1 dargestellt, Produktionssysteme in ausbringungsorientierte („Output“), einsatzbezogene („Input“) und prozessbezogene („Durchsatz“) Produktionssysteme unterscheiden [23, S. 23].

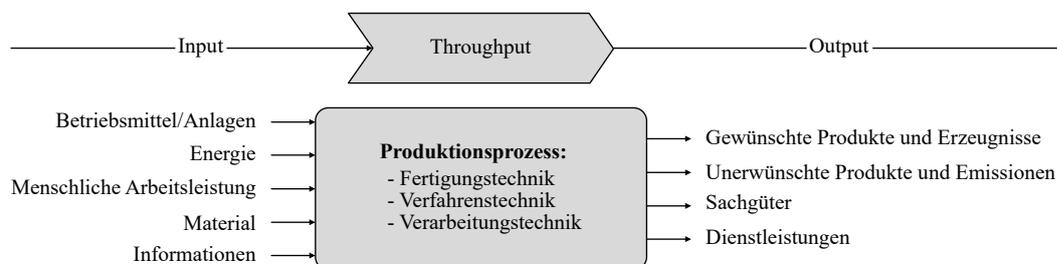


Abbildung 2.1: Produktionssystem bzw. -prozess in Anlehnung an Steven [24].

Im Rahmen der vorliegenden Dissertation sind die Ergebnisse des Prozesses keine immateriellen Güter wie Dienstleistungen, sondern nur materielle Güter [23, S. 13–16]. Abgesehen von der Herstellung des angestrebten Outputs ist das Eingangsmaterial für die Ausführung der Produktion ein weiterer wesentlicher Faktor. Die drei gängigsten Inputs sind Materialien, Equipment und die menschliche Arbeitskraft [23, S. 16–17].

In einer prozessbezogenen Produktion werden die innere Struktur eines Produktionssystems und folglich der effektive Transformationsprozess betrachtet [24, S. 16], wobei eine Nutzen- oder Wertsteigerung angestrebt wird. Der Durchsatz des Produktionssystems wird, wie in Abbildung 2.1 dargestellt, durch einen Produktionsprozess beschrieben. Der gesamtheitliche Produktionsprozess kann wiederum aus unterschiedlichen Teilprozessen aufgebaut sein. Diese Teilprozesse sind zudem aus diversen Schritten und Aktivitäten, wie beispielsweise Bohren oder Drehen, zusammengesetzt [25]. Dabei kann der Produktionsprozess unter anderem in die Fertigungstechnik, Verfahrenstechnik oder Verarbeitungstechnik unterteilt werden.

Zusammenfassend liegt der Schwerpunkt der vorliegenden Forschungsarbeit im Bereich der industriellen Fertigungstechnik zur Herstellung von Sachgütern.

2.2.2 Produktionsmanagement

Die Zielerreichung, wie beispielsweise Kosten- oder Qualitätsansprüche, erfordert eine Anpassung des Produktionssystems an die sich ändernden Rahmenbedingungen. Mittels des Produktionsmanagements werden Produktionsziele konkretisiert. Falls sich Abweichungen zu einem definierten Ziel ergeben, müssen Anpassungen durch das Produktionsmanagement vorgenommen werden, damit das im Voraus festgelegte Ziel erreicht werden kann. In Abhängigkeit des Zeithorizonts wird Produktionsmanagement in strategisches, taktisches oder operatives Produktionsmanagement unterteilt [23]. In der vorliegenden Arbeit wird ausschließlich das operative Produktionsmanagement betrachtet, welches sich mit der kurzfristigen Planung beschäftigt. Dies beinhaltet unter anderem die Überwachung und Steuerung von Fertigungsprozessen [23].

2. Grundlagen

2.2.3 IT-unterstützte Produktion

Mittels der Unterstützung von IT-Systemen können Produktionssysteme automatisiert gesteuert und optimiert werden. Ziel ist, die Produktivität der Anlage und die Qualität der produzierten Produkte zu erhöhen [26, S. 6]. Zur Erreichung dieser Ziele wird die sogenannte Prozessleittechnik benötigt welche, wie in Abbildung 2.2 visualisiert, die ganzheitliche Betrachtung der Mess-, Steuerungs- und Kommunikationstechnik beschreibt [26, S. 14].

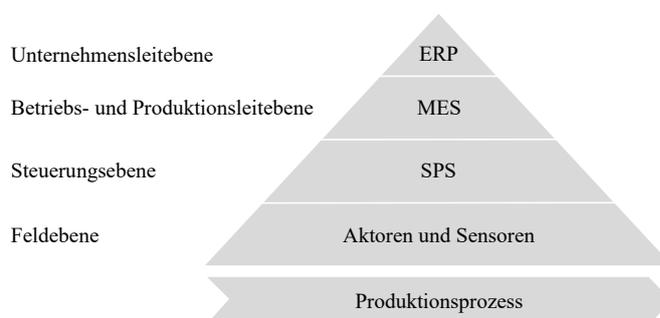


Abbildung 2.2: Automatisierungspyramide in Anlehnung an Heinrich [26, S. 10].

Die oberste Stufe der Automatisierungspyramide ist die Unternehmensleitebene, auf welcher Geschäftsprozesse des Unternehmens realisiert werden, wobei die Verarbeitungszeit eine untergeordnete Rolle spielt. Mittels der Betriebs- und Produktionsleitebene wird eine kurzfristige Produktionsplanung, wie beispielsweise die Einsatzplanung von Maschinen, sichergestellt. Dabei wird sie durch das Manufacturing Execution System (MES) unterstützt. Auf der Steuerungsebene werden durch die Verwendung einer speicherprogrammierbaren Steuerung (SPS) Maschinenaktionen koordiniert. Auf der untersten Ebene der Automatisierungspyramide sind die Sensoren und Aktoren allokiert. Auf dieser Ebene werden Daten erfasst und Reaktionen durch die Verbindung mit der Steuerung initiiert. Im Gegensatz zur Unternehmensleitebene werden hier kleine Datenmengen schnell und meist in Echtzeit verarbeitet [27, S. 5506], [28, S. 331], [26, S. 5].

Der Fokus der vorliegenden Arbeit liegt auf der Steuerungs- und Feldebene. Ziel ist die Ausdehnung des Netzwerks, um durch Vernetzung der Produktionsanlagen und einer gesteigerten Menge an Daten diese zu optimieren.

2.2.4 Prozessoptimierung

Wie bereits in Abschnitt 2.2.2 beschrieben, befasst sich das Produktionsmanagement mit der Erreichung von Produktionszielen, wozu jedoch die Optimierung des Produktionsprozesses vonnöten ist. Aus diesem Grund müssen Verbesserungspotenziale identifiziert und dazu passende Lösungen entwickelt werden [29, S. 5–9]. Herausfordernd dabei ist, dass Ziele in diverse Kriterien aufgeteilt werden können. Häufig wird dabei in der Literatur vom „magischen Dreieck“ gesprochen. Dieses Dreieck differenziert zwischen den Kategorien Kosten, Zeit und Qualität [25, S. 13].

Eine weitere Kernfrage der Prozessoptimierung ist einerseits, welche Prozesse zu optimieren sind, andererseits wie viele Änderungen zeitgleich vorgenommen werden sollen. Diese unterschiedlichen Arten sind in Abbildung 2.3 visualisiert. Da in dieser Arbeit ausschließlich eine Prozessoptimierung und ein kontinuierlicher Verbesserungsprozess (KVP) als Ansätze verwendet werden, wird im Folgenden ausschließlich auf diese eingegangen. Prozessoptimierung zielt darauf ab, einen bereits bestehenden Prozess effektiver, effizienter und flexibler zu gestalten. Die Verbesserung wird folglich auf der Grundlage eines bereits existierenden Prozesses mit einigen wesentlichen Neuerungen umgesetzt [25, S. 29–31]. Kontinuierliche Verbesserung stammt ursprünglich aus dem Japanischen und wurde erstmals unter dem japanischen Synonym „Kaizen“ (japanisch: Änderungen zum Besseren) angewandt [30, S. 260]. Verbesserungen erfolgen dabei zyklisch in kleinen Schritten. Dies hat den Vorteil, dass im Gegensatz zum Prozess-*Reengineering* beim KVP ein nachhaltiger Erfolg erzielt wird, ohne große Risiken in Kauf nehmen zu müssen (vgl. Abbildung 2.3).

2.3 Datengetriebene Optimierung

Nach Gröger gibt es drei verschiedene Ansätze zur Optimierung eines Fertigungsprozesses: management-, modell- und datengetriebene Optimierung [31, S. 33–35], [32]. Da in der vorliegenden Arbeit eine datengetriebene Optimierung verwendet wird, wird im Folgenden ausschließlich auf diesen Ansatz eingegangen.

Ein datengetriebener Ansatz behandelt die bestmögliche Nutzung und Einbeziehung von Daten zur Optimierung eines Prozesses. Die in diesem Prozess verwendeten

2. Grundlagen

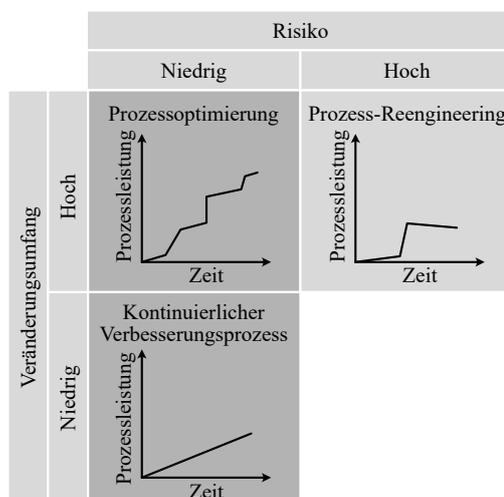


Abbildung 2.3: Methoden der Prozessoptimierung nach Becker [25, S. 21].

Daten können strukturiert oder auch unstrukturiert vorliegen. Anhand der Daten sollen automatisch Modelle erzeugt werden, mit denen sich Probleme erkennen und Optimierungspotenziale abbilden lassen. Bei einem modellgetriebenen Ansatz müssen hingegen komplexe, händische Algorithmen oder Modelle erarbeitet werden [31, S. 36].

Die Optimierung eines Produktionsprozesses mithilfe von Daten bedarf allerdings effizienter Methoden [33, S. 622], [34, S. 1]. Die Datenanalyse (Data Analytics) zählt zu den Schlüsselkompetenzen produzierender Unternehmen, um konkurrenzfähig zu bleiben, die Produktivität zu fördern und Innovationen zu entwickeln [35, S. 2], [36], [37], [38].

Data Analytics untersucht Technologien, Fähigkeiten, Anwendungen und Praktiken, um anhand der erhobenen Daten Erkenntnisse zu gewinnen. Data Analytics kann in zwei Hauptbereiche unterteilt werden: zum einen in Business Intelligence (BI) und zum zweiten in ADA [39, S. 487]. Wie in Abbildung 2.4 zu erkennen, gliedern sich diese Hauptbereiche wiederum, abhängig vom Analysehorizont und dem resultierenden Mehrwert, in vier weitere Analysestufen.

2.3.1 Business Intelligence

BI fokussiert sich auf die Messung bisheriger und aktueller Leistung eines Prozesses unter der Anwendung eines konsistenten Datensatzes. Mithilfe dieser Analyseergebnisse

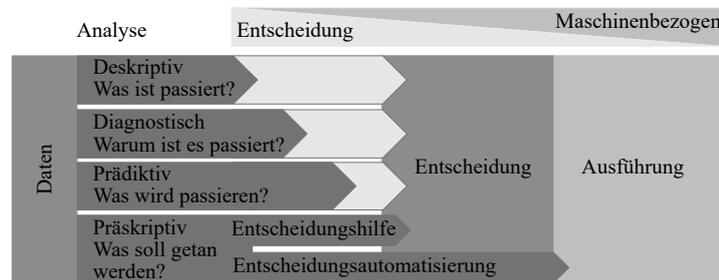


Abbildung 2.4: Übersicht Analysehorizonte Data Analytics [40, S. 10].

werden Daten zum Beispiel für Berichte oder Visualisierungen aufbereitet. Abhängig vom Analysegrad wird BI, wie in Abbildung 2.4 dargestellt, in deskriptive und diagnostische Analyse eingeteilt [41, S. 337], [42, S. 374ff.].

Deskriptive Analyse: Sie ist die einfachste und am häufigsten verwendete Art der Datenanalyse [35, S. 3]. Sie spiegelt eine Bestandsaufnahme historischer Daten wider und beschäftigt sich mit der Frage: „*Was ist passiert?*“ [41, S. 338]. Beispielsweise werden damit Ex-post-Analysen von Prozessen durchgeführt, wodurch Berichtsfunktionalitäten ermöglicht werden [43, S. 267f.]. Deskriptive Analysen helfen somit die Leistung eines Prozesses zu verstehen, um damit die vergangene Funktionalität eines Prozesses einschätzen zu können [44, S. 247].

Diagnostische Analyse: Die diagnostische Analyse ist eine Steigerung der zuvor erläuterten deskriptiven Analyse. Sie befasst sich mit der Frage: „*Warum ist es passiert?*“ [41, S. 338]. Dadurch können unerwünschte Abweichungen vom normalen Prozess erkannt werden. Dies hilft die Gründe der vergangenen Abweichung zu verstehen [44, S. 247]. Somit umfasst sie beispielsweise die kontinuierliche Überwachung und Prüfung signifikanter statistischer Differenzen zwischen Soll- und Ist-Werten bzw. Prozessen [41, S. 1418].

2.3.2 Advanced Data Analytics

ADA lässt sich klar vom klassischen Konzept der BI abgrenzen. Wie oben beschrieben befasst sich BI mit einfachen, vergangenen Ereignissen, wohingegen ADA den Fokus auf die Aufdeckung von schwer zu erkennenden Mustern und zukünftigen Ereignissen setzt. ADA versucht somit ein zukünftiges Ereignis vorherzusagen, zu korrelieren und zu optimieren. Dabei ist zwischen prädiktiver und präskriptiver Analyse zu unterscheiden [39, S. 448].

2. Grundlagen

Prädiktive Analyse: Die erste Stufe in ADA hat das Ziel, zukünftige Prozessbedingungen und eventuelle Abweichungen vorherzusagen. Dabei beantwortet sie die Frage: „*Was wird passieren?*“ [43, S. 267]. Es werden statistische Modelle verwendet, wobei angenommen wird, dass das Vergangene auch in Zukunft auftreten wird [35, S. 3]. Algorithmen im Bereich maschinelles Lernen (ML) oder Deep Learning helfen, um Verhaltensmuster in Daten zu erkennen und somit Vorhersagen treffen zu können [45, S. 1]. Gebräuchliche prädiktive Methoden sind beispielsweise Entscheidungsbäume, Bayes'sche Statistik oder neuronale Netze, welche im folgenden Abschnitt genauer erläutert werden [46, S. 678]. Ein Beispiel hierfür ist die Vorhersage eines Maschinenausfalls, was in der Literatur meist als Predictive Maintenance bezeichnet wird [47, S. 171].

Präskriptive Analyse: Die präskriptive Analyse dient der Bestimmung von Handlungsempfehlungen, um ein gewünschtes Ergebnis zu erzielen. Sie befasst sich mit der Frage: „*Was muss ich tun, um ein gewünschtes Ziel zu erreichen?*“ [48, S. 57]. Die präskriptive Analyse bietet somit eine automatisierte, adaptive und zeitabhängige Entscheidung und empfiehlt die beste Vorgehensweise zur Erreichung bestimmter Leistungskennzahlen [49]. Im Allgemeinen ist die präskriptive Analytik eine Erweiterung der prädiktiven Analyse, welche abhängig von der ermittelten Vorhersage eine Handlungsoption vorschlägt [44, S. 247]. Präskriptive Fähigkeiten werden durch Optimierungstechniken, wie beispielsweise mathematische Programmierung, Constrain-Programmierung, Entscheidungsbäume und Simulationsmodellierung umgesetzt [41, S. 339].

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

Die Produktionsautomatisierung ist angesichts der zunehmenden Forderungen nach Rentabilität und qualitativ hochwertigen Produkten ein bedeutender Bestandteil produzierender Unternehmen [50, S. 7270]. Aufgrund dessen ist es essenziell, Qualitätsabweichungen frühzeitig, zuverlässig und effizient zu erkennen. Früher hat der Mensch diese Abweichungen detektiert, allerdings wird dieser schon jetzt zunehmend durch computergestützte Programme ersetzt. Hintergrund dafür ist, dass Maschinen und Computer zuverlässiger und wirtschaftlicher sind. Zunehmend an Bedeutung gewinnt die kontaktlose Inspektion von Produktoberflächen mithilfe einer optischen Prozessabsicherung,

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

was bedeutet, dass die Fehlerfreiheit des Herstellungsprozesses anhand einer Kombination von ML und Computer Vision erfasst wird. Dies bietet den Vorteil, dass der Prüfprozess in Bezug auf elektromagnetische Felder, Temperatur und Oberflächenbeschaffenheit unabhängig ist [51, S. 9]. In Abbildung 2.5 sind die Abhängigkeiten von ML, Computer Vision und Human Vision dargestellt.

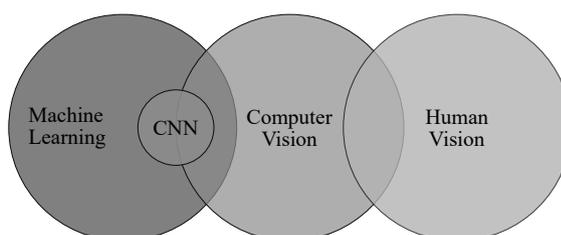


Abbildung 2.5: Zusammenhang zwischen Computer Vision, Machine Learning, Human Vision und einem CNN [52, S. 7].

ML weist den maßgeblichen Vorteil auf, dass Eigenschaften und deren Verknüpfung nicht durch spezielles Expertenwissen, sondern durch Lernen aus vorhandenen Daten erreicht wird [52, S. 5]. Nachteil dabei ist jedoch, dass Merkmale händisch ausgewählt werden müssen. Wie in Abschnitt 2.4.5 detailliert beschrieben, besitzen sogenannte CNN diesen Nachteil nicht.

2.4.1 Machine Learning

Wie aufgezeigt, gewinnen ML-Algorithmen im Bereich der Computer Vision an Wichtigkeit. Speziell die Kombination von Computer Vision und ML hat dazu beigetragen, Algorithmen im Bereich der Computer Vision flexibel und robust zu gestalten. Zielsetzung dabei ist es, Aufgaben anhand von Daten und nicht durch manuelle Programmierung von Regeln zu erlernen. Die zu diesem Zweck herangezogenen Daten werden als Trainingsdaten bezeichnet. Sie dienen dazu, während der sogenannten Trainingsphase gewünschte Regeln oder Vorschriften zu erlernen. Dabei wird zwischen überwachtem und unüberwachtem Lernen unterschieden [53], [54]. Da in der vorliegenden Arbeit ausschließlich überwachtes Lernen angewandt wird, wird im Folgenden nur auf diese Form des maschinellen Lernens eingegangen. Generell wird überwachtes Lernen angewandt, wenn einem System Eingangs- und Ausgangsvariablen zugewiesen werden mit der Intention, dass das System lernt, wie diese Variablen zueinander gehören. Angestrebt wird die Erzeugung einer präzisen Abbildungsfunktion, mit der der Algorithmus den Ausgang

2. Grundlagen

vorhersagen kann, wenn ein neuer Eingang gegeben wird. Die dafür nötigen Algorithmen werden in flache und tiefe Modelle eingeteilt. Beispiele flacher, überwachter Lernverfahren sind Random Decision Trees, Logistic Regression oder Support Vector Machines. Beispiele tiefer Netze hingegen sind NN, Kernel Machines oder Bayesian Classifier. [52, S. 5], [55, S. 2–3, 13]

2.4.2 Computer Vision

Um dreidimensionale Objekte erkennen und identifizieren zu können, verwenden Menschen ihre Augen und das Gehirn. Computer Vision verfolgt das Ziel, das menschliche Sehsystem nachzuahmen und diese Fähigkeit auf Computer zu übertragen [52, S. 1]. Merkmalsextraktion und Klassifikation sind zwei Kernkomponenten von Computer Vision-Systemen. Bei der Merkmalsextraktion wird nach charakteristischen Merkmalen gesucht, die für die Klassifikation notwendig sind. Die Kombination von n Merkmalen wird in einem n -dimensionalen Vektor, dem sogenannten Merkmalsvektor, zusammengefasst. [52, S. 1], [56, S. 61], [57, S. 172]

Klassische Algorithmen, um relevante Merkmale zu erkennen bzw. zu extrahieren, sind beispielsweise Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT) und Speeded Up Robust Features (SURF). Mithilfe dieser Algorithmen können zum Beispiel Ecken, Kanten oder weitere Merkmale in einem Bild erkannt und extrahiert werden. Nachteile dieser Algorithmen sind der Zeitaufwand und erforderliche Expertenkenntnisse, die speziell für einen Anwendungsfall nötig sind [52, S. 11–14], [58, S. 221]. Die oben erwähnten Nachteile können durch das Verwenden einer automatischen Merkmalsextraktion beseitigt werden. Ein Beispiel hierfür ist ein sogenanntes CNN [52, S. 21]. Auf dessen Bedeutung wird in Abschnitt 2.4.5 näher eingegangen. Klassifikatoren nutzen den Merkmalsvektor, um beispielsweise ein Bild einer Kategorie zuzuordnen. Modelle des maschinellen Lernens können, wie in Abschnitt 2.4.1 beschrieben, in flache und tiefe Modelle unterteilt werden. Um tiefe Modelle besser verstehen zu können, werden in dem folgenden Abschnitt NN näher erläutert.

2.4.3 Das künstliche Neuron

Das künstliche NN, ist ein informationsverarbeitendes System, das in seiner Struktur und Funktion dem Gehirn eines Menschen oder Tieres ähnelt. Ein derartiges Netzwerk

ist aus einer Vielzahl von Neuronen aufgebaut, die untereinander vernetzt sind und parallel zueinander arbeiten. Ein solches Neuron ist in Abbildung 2.6 dargestellt [59, S. 9],[60, S. 223].

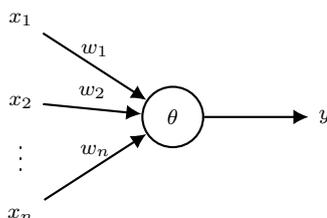


Abbildung 2.6: Generische Darstellung eines Neurons [59, S. 27].

Mathematisch ist dieses Neuron durch Gleichung 2.1 definiert.

$$y = \begin{cases} 0 & \sum_{i=1}^n w_i \cdot x_i \leq \theta \\ 1 & \sum_{i=1}^n w_i \cdot x_i > \theta \end{cases} \quad (2.1)$$

θ steht dabei für den Schwellwert, bei welchem das künstliche Neuron aktiviert wird. n gibt die Anzahl der Eingänge des Neurons und x_i den Eingang mit der dazugehörigen Gewichtung w_i an. y beschreibt den Ausgang des Neurons. Falls der Term $\sum_{i=1}^n w_i \cdot x_i$ einen Schwellwert θ überschreitet, wird das Neuron aktiviert und folglich der Ausgang y von 0 auf 1 gesetzt [59, S. 15], [60, S. 233], [61, S. 48–49]. Durch das Definieren von b als negativen Schwellwert $-\theta$ und $g(z)$ als Aktivierungsfunktion kann aus Gleichung 2.1 Gleichung 2.2 abgeleitet werden.

$$y = g\left(\sum_{i=1}^n w_i \cdot x_i + b\right) = g(z) \quad \text{mit } z = \sum_{i=1}^n w_i \cdot x_i + b \quad (2.2)$$

$g(\cdot)$ beschreibt die Aktivierungsfunktion und z die Summe der gewichteten Eingänge mit dem negativen Schwellwert. Aktivierungsfunktionen sind für das Schalten bei ausreichend großer Anregung des Neurons verantwortlich (vgl. Gleichung 2.3). [59, S. 27], [61, S. 48–49], [62, S. 37–38]

$$y = g(z) = \begin{cases} 0 & z \leq 0 \\ 1 & z > 0 \end{cases} \quad (2.3)$$

Die Aktivierungsfunktion wird in Abschnitt 2.4.5.2 näher erläutert.

2. Grundlagen

2.4.4 Künstliche neuronale Netze

Ein NN ist eine Verkettung der im Abschnitt 2.4.3 beschriebenen künstlichen Neuronen. Durch Vernetzung und Nutzung der diversen Aktivierungsfunktionen der Neuronen lässt sich ein nichtlineares Verhältnis zwischen den Ein- und Ausgängen abbilden. Im Folgenden wird ein solches Multi-Layer-Perceptron (MLP) erläutert. Die Grundstruktur des MLP besteht aus einer Eingangs- und Ausgangsschicht sowie mindestens einer verdeckten Schicht. Diese Schichten bestehen aus Neuronen, welche mit allen Neuronen der Folgeschicht verbunden sind. Zur Verbesserung der Lernfähigkeit kann die Anzahl der Neuronen pro Schicht und die Anzahl der verdeckten Schichten nach Belieben angepasst werden [59, S. 47]. In Abbildung 2.7 ist beispielhaft ein 2-2-1 MLP dargestellt. Dabei steht die jeweilige Zahl für die Anzahl der Neuronen pro Schicht. Somit steht die erste Zahl für die Anzahl der Eingänge und die letzte für die Anzahl der Ausgänge.

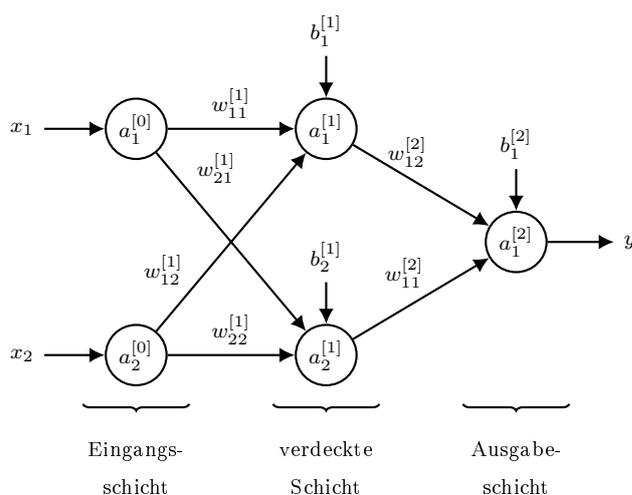


Abbildung 2.7: Beispielhafte grafische Darstellung eines 2-2-1 Multi-Layer-Perceptrons [59, S. 50].

Für den weiteren Verlauf der Arbeit werden die in dieser Dissertation benutzten Parameter wie folgt definiert [60, S. 234–235]:

- $w_{ij}^{[l]}$ definiert die Gewichtung des Signals vom j -ten Neuron aus der $(l - 1)$ -ten Schicht zum i -ten Neuron in der l -ten Schicht.
- Der Schwellwert des i -ten Neurons der l -ten Schicht ist durch die Beziehung $b_i^{[l]}$ beschrieben.

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

- $z_i^{[l]}$ definiert den aufsummierten Eingang des i -ten Neurons aus der l -ten Schicht.
- $g^{[l]}(\cdot)$ repräsentiert die Aktivierungsfunktion sämtlicher Neuronen in der l -ten Schicht. Dabei wird vorausgesetzt, dass alle Neuronen einer Schicht grundsätzlich die gleiche Aktivierungsfunktion aufweisen.
- Der Ausgang des i -ten Neurons aus der l -ten Schicht ist durch $a_i^{[l]}$ definiert. Folglich gilt $a_i^{[l]} = g^{[l]}(z_i^{[l]})$.

Aus Gründen der besseren Übersichtlichkeit lassen sich die oben festgelegten Parameter in einer Matrix-Vektor-Notation zusammenfassen. Die Gewichte des Ein- bzw. Ausgangs sind durch Gleichung 2.4 beschrieben [62, S. 2.10].

$$\mathbf{W}^{[1]} = \begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} \end{bmatrix}, \quad \mathbf{W}^{[2]} = \begin{bmatrix} w_{11}^{[2]} & w_{12}^{[2]} \end{bmatrix} \quad (2.4)$$

Durch den Vektor aus Gleichung 2.5 können die Schwellwerte zusammengefasst werden [62, S. 2.10].

$$\mathbf{b}^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \end{bmatrix}, \quad \mathbf{b}^{[2]} = \begin{bmatrix} b_1^{[2]} \end{bmatrix} \quad (2.5)$$

Somit sind die Ausgänge der Schichten eines MLP mit insgesamt L Schichten durch

$$\mathbf{a}^{[l+1]} = \mathbf{g}^{[l+1]}(\mathbf{W}^{[l+1]} \cdot \mathbf{a}^{[l]} + \mathbf{b}^{[l+1]}) \quad \text{mit } l = 0, 1, \dots, L-1 \quad (2.6)$$

beschrieben. Wie in Gleichung 2.6 zu erkennen, wurde die Aktivierungsfunktion mehrdimensional auf $\mathbf{g}^{[l+1]}(\cdot)$ erweitert. Durch iteratives Anwenden der Formel 2.6 lässt sich die Ausgabe des in Abbildung 2.7 dargestellten MLP durch Gleichung 2.7 kompakt formulieren. Die Eingänge des Netzes werden dabei mithilfe des Vektors $\mathbf{x} = [x_1, x_2]^T$ zusammengefasst. [62, S. 2.10–2.12]

$$y = a^{[2]} = \mathbf{g}^{[2]}(\underbrace{\mathbf{W}^{[2]} \cdot \mathbf{g}^{[1]}(\underbrace{\mathbf{W}^{[1]} \cdot \mathbf{x} + \mathbf{b}^{[1]}}_{z^{[1]}})}_{\mathbf{a}^{[1]}} + \mathbf{b}^{[2]}) \quad (2.7)$$

Bedingt durch die zahlreichen Parameter, die im Laufe des Trainingsprozesses eines NN erlernt werden, ist es möglich, dass das Modell zu stark an die verwendeten Trainingsdaten angepasst wird, jedoch keine gute Generalisierung aufweist. Diese Überanpassung der Parameter während des Trainingsprozesses wird als Overfitting bezeichnet.

2. Grundlagen

Zur Vermeidung von Overfitting können sogenannte Regularisierungstechniken angewandt werden [63, S. 115–117]. Dazu gehören beispielsweise das Einfügen einer Batch-Normalization oder einer Dropout-Schicht. [52, S. 73], [63, S. 115–117], [64, S. 1929]

2.4.5 Convolutional Neural Network

Wie dargestellt, lassen sich mit den vorgestellten NN ohne Expertenwissen bestimmte Ein- und Ausgabeverhältnisse von Merkmalen oder individuellen Daten erlernen. Allerdings weisen NN bei der Erkennung eines Objekts auf einem Bild deutliche Nachteile auf. Ein Beispiel hierfür ist die große Anzahl zu trainierender Parameter. Außerdem können Abhängigkeiten zwischen einzelnen Pixeln und somit wichtige Informationen verloren gehen [63, S. 85], [65, S. 89-90]. Aufgrund dieser Nachteile sind klassische NN nicht für die Klassifizierung von Merkmalen oder Objekten in einem Bild geeignet.

CNN sind eine Erweiterung klassischer NN. Diese sind speziell für das Erkennen und Klassifizieren von Merkmalen mehrdimensionaler Daten, wie beispielsweise Tonspuren (1D), Bildern (2D) und Videos (3D) entwickelt. Außerdem zeigen Studien, dass die Objekterkennung mithilfe eines CNN die Leistungen klassischer, händischer Methoden wie beispielsweise HOG, SIFT oder Farbhistogrammen übertrifft [66, S. 161], [67, S. 6232], [68, S. 4303], [69, S. 4404]. Die typische Schicht eines CNN besteht dabei, wie in Abbildung 2.8 dargestellt, aus drei Stufen: einer sogenannten faltenden Schicht, einer Aktivierungsschicht sowie einer bündelnden (Pooling-) Schicht.

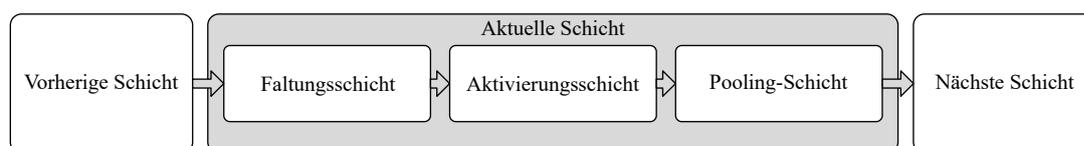


Abbildung 2.8: Typischer Aufbau der Schichten eines CNN [70, S. 336].

2.4.5.1 Faltungsschicht

Die Faltungsschicht bildet die wesentliche Komponente eines CNN. Diese Schicht umfasst eine definierte Anzahl an Filtern, die als Faltungskern bezeichnet werden. Durch die Faltung der Eingabe können Merkmale der Ausgabe erzeugt werden. Dabei entsprechen die Filter einer Matrix diskreten Werten, die die Gewichte der Filter repräsentieren. Sie

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

werden während des Trainings des CNN erlernt [52, S. 46]. Die Faltungsschicht basiert dabei auf dem mathematischen Operator der Faltung $*$. Das Ergebnis S der diskreten Faltung ist durch Gleichung 2.8 definiert.

$$S(i, j) = (F * I)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot F(m, n) \quad (2.8)$$

Dabei beschreibt I das zweidimensionale Bild und F den zweidimensionalen Filter. Durch die Faltung wird jedoch der Filter entlang der vertikalen und horizontalen Achse gespiegelt [70, S. 328]. In der Signalverarbeitung wird diese mathematische Operation als Faltung bezeichnet. Die Faltungsschicht verwendet eine ähnliche mathematische Operation, die sogenannte Kreuzkorrelation (engl. cross correlation). Die Kreuzkorrelation entspricht ebenfalls einer Faltung, jedoch ohne Spiegelung des Filters. Sie wird ebenfalls in der Literatur als Faltung beschrieben. Aus diesem Grund wird im weiteren Verlauf dieser Arbeit, falls nicht weiter angegeben, der mathematische Operator $*$ als Kreuzkorrelation definiert. Die mathematische Beschreibung der Kreuzkorrelation ist in Gleichung 2.9 wiedergegeben.

$$S(i, j) = (F * I)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot F(m, n) \quad (2.9)$$

In Abbildung 2.9 ist beispielhaft die Anwendung der Kreuzkorrelation aus Gleichung 2.9 auf ein Bild I der Größe 3×4 und einem Filter F der Größe 2×2 dargestellt. Wie zu erkennen, reduziert sich die Größe des resultierenden Bildes S auf die Größe 2×3 . Es ist jedoch darauf zu achten, dass bei mehrfacher Faltung die Größe des Bildes zu stark reduziert wird, wodurch relevante Informationen verloren gehen können [52, S. 46]. Durch das Hinzufügen von Spalten oder Zeilen mit Nullen kann die aus der Faltung resultierende Dimension des Bildes angepasst werden. Diese Operation wird Zero-Padding (dt. Null-Polsterung) genannt. Entspricht die Ausgangsgröße der Filterung der Eingangsgröße des Bildes, wird von einer Same Convolution gesprochen [52, S. 48–49].

Ein Vorteil der Faltungsschicht ist das Parameter-Sharing. Dadurch sind die Parameter der Filtermatrix F aller Eingänge einer Schicht dieselben. Dies führt dazu, dass dieselben Merkmale, unabhängig von der Position der Pixel des Eingangsbildes, extrahiert und relevante Objekte erkannt werden können. Vor allem in der Produktion ist es essenziell, dass das CNN Fehlermerkmale an unterschiedlichen Positionen erkennen

2. Grundlagen

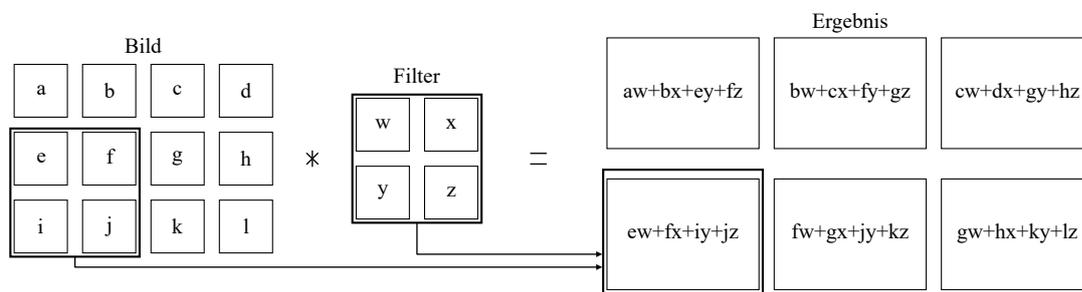


Abbildung 2.9: Beispielhafte Abbildung der Faltung durch die Kreuzkorrelation [70, S. 330].

kann, da identische Fehlerbilder an unterschiedlichen Positionen auftreten können. Ein weiterer Vorteil ist die Reduktion der Anzahl an Parametern. Durch die Faltung können bestimmte Eigenschaften wie Kanten des Bildes durch die Verwendung verschiedener Filter hervorgehoben werden. Die Parameter des Filters werden iterativ durch den Backpropagation-Algorithmus und Optimierung der Kostenfunktion angeleert [63, S. 11–21], [63, S. 90–94], [70, S. 333–335].

2.4.5.2 Aktivierungsschicht

Eine weitere Schicht des Convolution-Blocks ist die Aktivierungsschicht. Durch sie wird das Ergebnis der Faltungsschicht mithilfe einer Aktivierungsfunktion, einer nichtlinearen Funktion, aktiviert. Dieses Vorgehen ist essenziell, da dadurch das CNN nichtlineare Verbindungen erlernen kann. Die sogenannte Rectified Linear Unit (ReLU)-Funktion ist die gebräuchlichste Aktivierungsfunktion bei tiefen Netzwerken wie einem CNN [71, S. 134]. Dabei werden Eingänge kleiner als 0 auf 0 abgebildet. Für alle anderen Werte wird der positive Eingang ausgegeben. Mathematisch ist sie durch die Gleichung 2.10 definiert. [52, S. 54], [70, S. 335]

$$g_{relu}(z) = \max(0, z) \quad (2.10)$$

2.4.5.3 Pooling-Schicht

In dieser Schicht werden Blöcke des Eingabebildes zu einer bestimmten Größe kombiniert und zu einem einzigen Wert zusammengeführt. Dabei wird zwischen dem sogenannten Average- und dem Max-Pooling unterschieden. Die gängigste Art des Poolings

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

ist das Max-Pooling. Hierbei wird der Maximalwert des definierten Bereiches als Ausgabe gewählt. Bei der Verwendung von Average-Pooling wird der definierte Bereich auf den Mittelwert von diesem reduziert. Vorteil des Poolings ist die Reduktion der Dimension des Bildes, was wiederum zu einem geringeren Speicherverbrauch und einer geringeren Berechnungszeit der Faltungen in den nächsten Schichten führt. Dies ist ebenfalls für die Produktion wichtig, da dadurch Rechenressourcen und somit finanzielle Aufwände gespart werden können. Zudem stabilisiert das Pooling das CNN gegenüber kleinen Änderungen der Eingabe. Folglich ändert beispielsweise die Verschiebung eines Objektes in einem Bild nicht die Ausgabe der Pooling-Schicht, was für eine Klassifikationsaufgabe in der Produktion essenziell ist. Durch die Reduktion der Dimension ist es möglich, die Datenmenge eines Bildes zu verringern und dennoch wichtige Merkmale, wie zum Beispiel die maximalen Pixelwerte, beizubehalten. [52, S. 53], [70, S. 335]

2.4.6 Kennzahlenbasierte Evaluierung des Klassifikationsalgorithmus

Ziel beim maschinellen Lernen ist die Entwicklung eines Modells, welches sich auf unbekannte Daten anwenden lässt. Um die Güte des Modells zu evaluieren, werden im Folgenden verschiedene Möglichkeiten vorgestellt [72, S. 97]. Für die Bewertung des Modells werden die vorhandenen Daten, wie in Abbildung 2.10 visualisiert, in drei Gruppen aufgeteilt: Trainings-, Validierungs- und Testdaten. Für die Teilung des Datensatzes in Trainings- und Validierungsdaten wird ein Teilungsverhältnis von 0,8/0,2 angewandt.

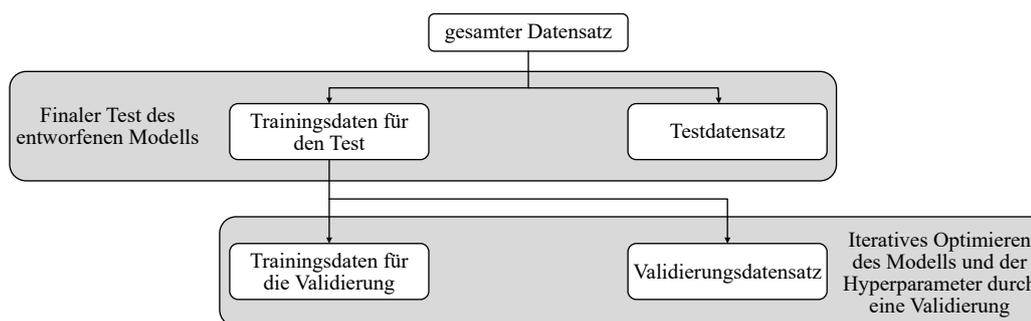


Abbildung 2.10: Beispielhafte Aufteilung eines Datensatzes in Trainings-, Validierungs- und Testdatensatz.

Wie der Name impliziert, werden die Trainingsdaten für das Training des Modells verwendet. Die Validierungsdaten hingegen werden für die Anpassung sogenannter Hyperparameter genutzt. Hyperparameter sind Parameter, die bereits vor dem Trainings-

2. Grundlagen

prozess festgelegt und an das Modell übermittelt werden. Es handelt sich dabei zum Beispiel um die Anzahl der verdeckten Schichten in einem neuronalen Netz oder unterschiedliche Aktivierungsfunktionen. Sie sind somit keine Parameter, die während des Trainingsprozesses des Modells erlernt und optimiert werden. Die Testdaten werden verwendet, um die allgemeine Gültigkeit des Netzes zu prüfen. Für diesen Test wird das Modell mit allen Trainings- und Validierungsdaten antrainiert. Anschließend wird das Netz mit dem Testdatensatz überprüft. Wichtig dabei ist, dass diese Daten nicht für den Trainingsprozess des Netzes verwendet werden. Die Bestimmung der Stichprobengröße wird in Abschnitt 2.4.8 erläutert. Nach der Auswertung des Modells mit dem Testdatensatz dürfen die Hyperparameter des Modells nicht mehr angepasst werden. [70, S.118–119]

2.4.6.1 k-fold Cross-Validation

Bei k-fold Cross-Validation verwendet wird der Datensatz in k gleich große Partitionen aufgeteilt. Jede Partition k wird einmal als Validierungssatz verwendet, während die restlichen $k - 1$ -Partitionen den Trainingssatz bilden [72, S. 99]. Bei einer beispielhaften 5-fold Cross-Validation wird der Datensatz, wie in Abbildung 2.11 dargestellt, in fünf Partitionen aufgeteilt.

Validierung	Training	Training	Training	Training	Lauf 1
Training	Validierung	Training	Training	Training	Lauf 2
Training	Training	Validierung	Training	Training	Lauf 3
Training	Training	Training	Validierung	Training	Lauf 4
Training	Training	Training	Training	Validierung	Lauf 5

Abbildung 2.11: Schematische Darstellung einer 5-fold Cross-Validation [55, S. 24].

Wie zu erkennen ist, besteht der Datensatz somit zu 80% aus Trainingsdaten und zu 20% aus Validierungsdaten, wobei die Trainings- und Validierungsdaten abhängig vom jeweiligen Lauf durchgewechselt werden.

2.4.6.2 Confusion Matrix

Die Confusion Matrix (CM) ist ein wichtiges Werkzeug zur Bewertung und übersichtlichen Darstellung der Ergebnisse eines Klassifikationsalgorithmus. Die einzelnen Einträge in einer CM geben die Anzahl der vom Modell getroffenen Prädiktionen an. In Abbildung 2.12a ist die CM einer binären Klassifikation dargestellt. Wie zu erkennen, besteht ein binäres Klassifizierungsproblem nur aus zwei Klassen, einer positiven und einer negativen Klasse. Die Einträge der Matrix lauten [73, S. 428–429]:

- True Positive (TP): Sie beschreibt die Anzahl der fehlerfreien Klassen, die durch den Klassifikationsalgorithmus richtig als fehlerfreie Klasse klassifiziert wurden.
- True Negative (TN): Sie beschreibt die Anzahl der fehlerhaften Klassen, die durch den Klassifikationsalgorithmus richtig als fehlerhafte Klasse klassifiziert wurden.
- False Positive (FP): Sie beschreibt die Anzahl der fehlerhaften Klassen, die durch den Klassifikationsalgorithmus fälschlicherweise als fehlerfreie Klasse klassifiziert wurden.
- False Negative (FN): Sie beschreibt die Anzahl der fehlerfreien Klassen, die durch den Klassifikationsalgorithmus fälschlicherweise als fehlerhafte Klasse klassifiziert wurden.

		vorhergesagte Klasse	
		kein Fehler	Fehler
echte Klasse	kein Fehler	TP	FN
	Fehler	FP	TN

(a) Confusion Matrix einer binären Klassifikation.

		vorhergesagte Klasse			
		kF	FK1	FK2	FK3
echte Klasse	kein Fehler (kF)	TP	FN	FN	FN
	Fehlerklasse 1 (FK1)	FP	TN	TN	TN
	Fehlerklasse 2 (FK2)	FP	TN	TN	TN
	Fehlerklasse 3 (FK3)	FP	TN	TN	TN

(b) Confusion Matrix einer Mehrklassen-Klassifizierung.

Abbildung 2.12: Darstellung einer Confusion Matrix nach Krüger [74].

Bei einem Klassifizierungsproblem mit mehreren Klassen besitzt die CM die Größe $c \cdot c$, wobei c der Anzahl der Klassen entspricht [75, S. 231]. In Abbildung 2.12b ist eine CM beispielhaft mit vier Klassen dargestellt.

2. Grundlagen

2.4.6.3 Bewertungsmetriken

Die Einträge der oben beschriebenen CM sind Absolutwerte. Dies erschwert es jedoch, Resultate verschiedener Modelle miteinander zu vergleichen, da eine CM mit vielen Klassen schnell unübersichtlich werden kann [73, S. 428–429]. Um dennoch Ergebnisse eines Klassifikationsproblems miteinander vergleichen zu können, werden im Folgenden vier wichtige Bewertungsmetriken für mehrklassige Entscheidungsprobleme vorgestellt.

Genauigkeit:

Die Genauigkeit eines Klassifikationsalgorithmus ist eine sehr intuitive Bewertungsmetrik, welche durch die Gleichung 2.11 für ein binäres Klassifikationsproblem definiert ist.

$$accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (2.11)$$

Bei einem Klassifikationsproblem mit mehr als zwei Klassen kann die Genauigkeit des Klassifikationsalgorithmus durch Gleichung 2.12 bestimmt werden.

$$acc_{gesamt} = \frac{\sum_{i=1}^c \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i}}{c} \quad (2.12)$$

Dabei steht c für die Anzahl der entsprechenden Klassen $i = 1, \dots, c$ [73, S. 430]. Die Klasse $i = 1$ wäre gemäß Abbildung 2.12b *kein Fehler*. Da es sich in der vorliegenden Arbeit um eine Mehrklassen-Klassifizierung handelt, werden für die folgenden Bewertungsmetriken ausschließlich die Gleichungen für ein Mehrklassenproblem wiedergegeben.

Precision:

Die Metrik *Precision* gibt die Anzahl der korrekt klassifizierten positiven Klassen geteilt durch die Anzahl der vom Klassifikationsalgorithmus fälschlicherweise als positiv gekennzeichneten Klassen wieder. Sie wird bei einem mehrklassigen Entscheidungsproblem beschrieben durch [63, S. 110–111]:

$$precision = \frac{\sum_{i=1}^c \frac{(TP_i)}{(TP_i + FP_i)}}{c}. \quad (2.13)$$

Für die Berechnung der Metrik *Precision* wurde Makro-Mittelung angewandt. Diese Berechnungsmethode wird bei ausgeglichenen Datensätzen verwendet [73, S. 430]. Wie in Abbildung 2.12 dargestellt, beschreibt Klasse 1 ein hergestelltes Produkt, welches keine Fehler aufweist, wohingegen die restlichen Klassen fehlerhaften Produkten entsprechen.

Für einen Fertigungsprozess wäre es ungünstig, wenn fehlerhafte Produkte als fehlerfreie klassifiziert würden (vgl. Abbildung 2.12). Folglich spielt diese Bewertungsmetrik in der Produktion eine essenzielle Rolle.

Recall:

Der *Recall* gibt Auskunft darüber, wie sensibel der Klassifikationsalgorithmus eine negative Klasse bestimmen kann. Oft wird diese Bewertungsmetrik auch Sensitivität genannt und ist definiert durch die Gleichung 2.14 [63, S. 110–111]:

$$recall = \frac{\sum_{i=1}^c \frac{(TP_i)}{(TP_i + FN_i)}}{c}. \quad (2.14)$$

Entsprechend der Metrik *Precision* wurde eine Makro-Mittelung angewandt, wodurch alle Klassen gleichermaßen berücksichtigt werden.

F_β – Score:

Die letzte in dieser Arbeit verwendete Bewertungsmetrik ist der F_β – Score. Dieser ist eine Kombination aus *Precision* und *Recall*, welcher für die Performance eines Klassifikators verwendet wird und ist definiert durch [73, S. 430]:

$$F_\beta = (1 + \beta^2) \cdot Precision \cdot \frac{Recall}{(\beta^2 \cdot Precision + Recall)}. \quad (2.15)$$

Diese Metrik ist hilfreich, wenn sowohl *Precision* als auch *Recall* von Bedeutung sind. Entspricht $\beta > 1$, ist die Bedeutung des *Recall* größer. Ist $\beta = 1$, werden *Precision* und *Recall* als gleich wichtig eingestuft. [63, S. 111]

2.4.7 Visuelle Evaluierung des Klassifikationsalgorithmus

Neben der klassischen, kennzahlenbasierten Evaluierung können NN auch visuell evaluiert werden. Die in dieser Arbeit verwendeten Ansätze sind beispielhaft in Abbildung 2.13 visualisiert und werden in den folgenden Abschnitten erläutert.

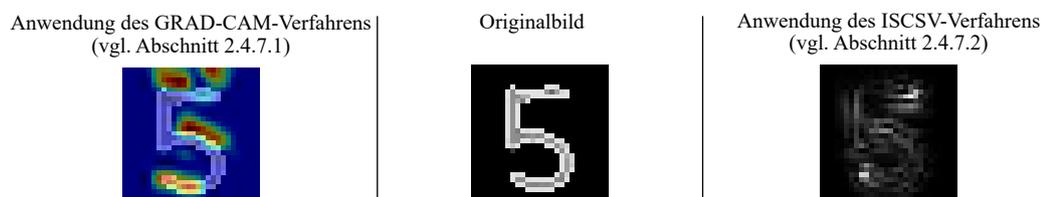


Abbildung 2.13: Anwendung des Grad-CAM- (links) und ISCSV-Verfahrens (rechts) auf die Klassifizierung einer handgezeichneten 5 (mitte).

2. Grundlagen

2.4.7.1 Class Activation Mapping

Der erste in dieser Arbeit verwendete Ansatz ist das sogenannte Gradient-weighted Class Activation Mapping (Grad-CAM)-Verfahren, welches in Abbildung 2.13 links dargestellt ist. Mithilfe dieses Ansatzes werden Bereiche im Bild hervorgehoben, welche für den Entscheidungsprozess eines NN wichtig sind. Analog zu einer Heatmap werden relevante Bereiche mit einer warmen Farbe, wie beispielsweise rot, und nicht relevante Bereiche mit einer kälteren Farbe, wie blau, eingefärbt. Das Eingangsbild des CNN wird anschließend mit dieser Heatmap überlagert, sodass das Merkmal, welches für die Klassifizierung verantwortlich ist, auch für einen Menschen lesbar bzw. erkennbar ist. [52, S. 94–95], [76, S. 618–620]

2.4.7.2 Class Saliency Visualisation

Eine weitere Darstellungsmethode ist die sogenannte Saliency Map, welche auch Image-Specific Class Saliency Visualisation (ISCSV) genannt wird. Abgeleitet wird diese Methode aus dem Konzept der Salienz in Bildern, wobei Auffälligkeiten hervorgehoben werden. Diese besonderen Merkmale kennzeichnen die visuell reizvollen Stellen in einem Bild. Die Saliency Map ist somit eine topografische Darstellung dieser Merkmale. Im Gegensatz zum Grad-CAM Verfahren werden ausschließlich die Pixel hell hervorgehoben, welche für den Entscheidungsprozess des CNN verantwortlich sind (vgl. Abbildung 2.13). Alle weiteren Pixel des Eingangsbildes werden dunkel eingefärbt. [77, S. 1–2]

2.4.8 Finale Evaluierung des Modells

Um das entworfene Modell final zu evaluieren, muss dieses anhand eines Testdatensatzes untersucht werden. Besonders wichtig ist hier die Verwendung einer hinreichend großen Anzahl von Testdaten. Für merkmalsbasierte Klassifikationsaufgaben wird im VDA Band 5 ein Stichprobenumfang von mindestens 30 Messungen gefordert. Im Verhältnis zur Größe eines gemeinsamen Trainings- sowie Validierungsdatensatzes ist diese Größe allerdings sehr klein. Da die Stichprobengröße für die meisten Anwendungen aus diesem Grund als zu klein angesehen wird, wird nachfolgend eine Methode zur Berechnung der Stichprobengröße dargestellt. [78, S. 64–66]

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

Grundlage eines qualifizierten Eignungsnachweises bilden statistische Methoden zum Nachweis der Leistungsfähigkeit eines Bildverarbeitungssystems in einem Prüfprozess. Hierfür gibt es nach der Richtlinie-VDI/VDE/VDMA 2632 Blatt 3.1 zwei grundsätzlich verschiedene Verfahren [79]:

- Sind die internen Berechnungsgrößen, die das bildverarbeitende System zur Klassifizierung heranzieht, zugänglich, kann das Verfahren „Analyse der Merkmalsverteilungen“ verwendet werden.
- Falls es sich jedoch um ein „Blackbox“-System handelt, wird der „Hypothesentest“ mittels Binomialtest angewandt. Vorteil davon ist, dass dies ein aus der Statistik bekanntes Verfahren ist, welches sich dadurch mathematisch interpretieren lässt. Da es sich bei neuronalen Netzen um Blackboxen handelt, wird diese Methode verwendet.

Mithilfe der Binomialverteilung wird die statistische Auftrittswahrscheinlichkeit für die Anzahl fehlklassifizierter Einheiten bzw. Klassen in einem Stichprobentest beschrieben. Gleichung 2.16 beschreibt die Wahrscheinlichkeitsfunktion der Binomialverteilung. Dabei steht $p_x(x)$ für die Wahrscheinlichkeit x falsch prädiktierter Klassen des Stichprobendatensatzes der Größe n und einer Fehlklassifizierungsrate p_f des Klassifikationsalgorithmus.

$$p_x(x) = \binom{n}{x} \cdot p_f^x (1 - p_f)^{n-x} \quad \text{mit} \quad \binom{n}{x} = \frac{n!}{x!(n-x)!} \quad (2.16)$$

Die kumulative Wahrscheinlichkeitsdichte ist durch Gleichung 2.17 definiert.

$$p_k(x \leq k) = \sum_{x=0}^k \binom{n}{x} \cdot p_f^x (1 - p_f)^{n-x} \quad (2.17)$$

Sie beschreibt die aufsummierte Wahrscheinlichkeit für k oder weniger falsch prädiktierter Klassen. Eine beispielhafte Darstellung der Wahrscheinlichkeitsdichte als auch der kumulativen Wahrscheinlichkeitsdichte ist in Abbildung 2.14 dargestellt.

Exemplarisch sind dabei eine Stichprobengröße $n = 60$ und eine Fehlklassifizierungsrate $p_f = 0,2$ visualisiert. Die kumulative Wahrscheinlichkeitsdichte ist in Abbildung 2.14b zu erkennen. Soll beispielsweise die größtmögliche Anzahl an Fehlklassifizierungen bei einer Überschreitung der Wahrscheinlichkeit von $p_k = 0,65$ bestimmt

2. Grundlagen

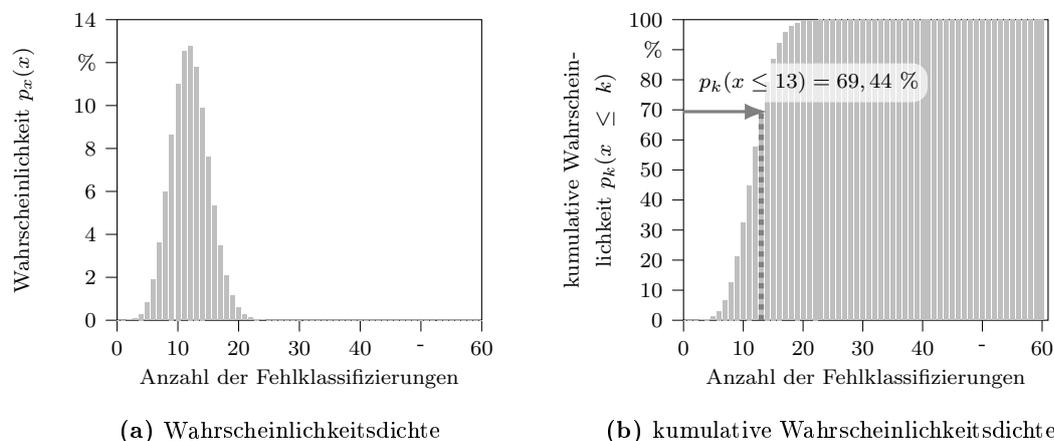


Abbildung 2.14: Beispielhafte Wahrscheinlichkeitsdichte und kumulative Wahrscheinlichkeitsdichte einer Binomialverteilung [80, S. 188].

werden, ergibt sich aus Abbildung 2.14b die Annahmegrenze von $k = 13$ Fehlklassifizierungen. [80, S. 109–112, 188], [81, S. 1–10]

Jedoch berücksichtigt diese Betrachtungsweise nicht, dass Auftragnehmer und Auftraggeber entgegengesetzte Interessen verfolgen. In der vorliegenden Arbeit ist die Auftragnehmersicht der aktuelle Produktionsprozess, in welchen das Prüfsystem integriert werden soll, und der Auftraggeber ist zum Beispiel ein Kunde oder ein nachgelagerter Produktionsprozess (vgl. Abbildung 2.15) [79, S. 13]. Um folglich eine erhöhte Fehlerrate und dadurch eventuelle Reklamationen zu vermeiden, sollten die vom Kunden, Nachfolgeprozess oder Nachfolgewerk (externe Qualität) vorgegebenen Anforderungen der Fehlklassifizierungsrate $p_{\text{externeQualität}}$ (p_{eQ}) größer als die technisch realisierbare Fehlklassifizierungsrate des Klassifikationsalgorithmus $p_{\text{interneQualität}}$ (p_{iQ}) sein [82]. Aufgrund dessen beschreibt im Folgenden p_{iQ} die Fehlklassifizierungsrate des Modells des Prüfsystems für den aktuellen Prozessschritt (interne Qualität) und p_{eQ} die maximale Fehlklassifizierungsrate des nachgelagerten Prozessschrittes oder des Kunden (externe Qualität). Wie in Abbildung 2.15 dargestellt, ergibt sich p_{eQ} aus Gleichung 2.18.

Die Anzahl fehlerhafter Bauteile im Produktionsprozess p_{defekt} kann durch Erfahrungswerte oder Experimente bestimmt werden. Der maximale Fehlerschlupf kann mithilfe des Prozessfähigkeitsindex bestimmt werden. Dieser beschreibt das Verhältnis

2.4 Prozessabsicherung durch Machine Learning und Computer Vision

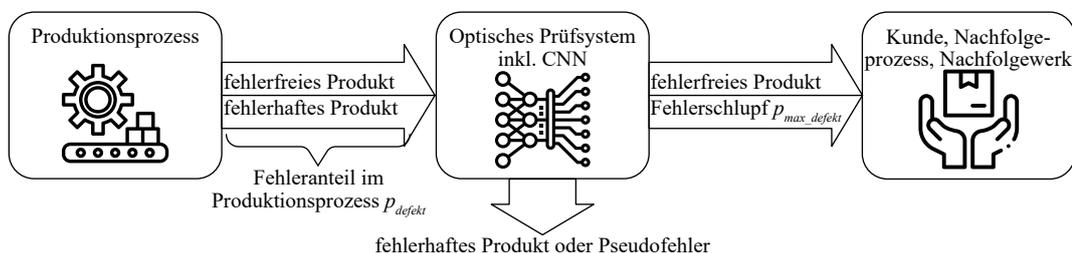


Abbildung 2.15: Fehleranteile im Produktionsprozess.

zwischen der Toleranz und definierter Prozessstreuung. Bei industriellen Fertigungsprozessen ist die Standardanforderung des Prozessfähigkeitsindex $C_{pk} = 1,33$ [83, S. 11], [84, S. 27]. Der maximal akzeptable Fehlerschlupfwert des Gesamtprozesses, dessen Prozessfähigkeitsindex $C_{pk} = 1,33$ beträgt, ist somit $p_{max_defekt} = 32 \text{ ppm} * 2 = 64 \text{ ppm}$. Bei kritischen Merkmalen kann auch ein Prozessfähigkeitsindex C_{pk} von 1,64 angenommen werden. Dieser Wert ist in der statistischen Fachliteratur zu finden [83, S. 11]. Somit ergibt sich als maximal akzeptable Fehlklassifizierungsrate des Modells der externen Qualität:

$$p_{eQ} = \frac{p_{max_defekt}}{p_{defekt}} \quad (2.18)$$

Die technisch erreichbare maximale Fehlklassifizierungsrate p_{iQ} kann anhand der in Gleichung 2.12 definierten Genauigkeit berechnet werden. Somit ergibt sich

$$p_{iQ} = (1 - acc_{gesamt}) \quad (2.19)$$

als Zielwert der Fehlklassifizierungsrate aus der Betrachtungsweise der internen Qualität.

Bei der Bestimmung des Stichprobenumfangs und der Annahmegrenze müssen jedoch die beiden Fehlklassifizierungsraten p_{iQ} und p_{eQ} berücksichtigt werden. Diese Anforderung erschwert die Bestimmung einer geeigneten Stichprobengröße und Annahmegrenze. Mithilfe des statistischen Hypothesentests kann „anhand einer Stichprobe eine Aussage auf die Datenkennwerte der Grundgesamtheit gemacht werden“, wobei beide Modelle betrachtet werden. Dabei werden, wie in Abbildung 2.16 dargestellt, die folgenden zwei Hypothesen getroffen [83, S. 103]:

2. Grundlagen

1. Hypothese H_0 : Fehlklassifizierungsrate der externen Qualität ist gleich der Fehlklassifizierungsrate der internen Qualität.
2. Hypothese H_1 : Fehlklassifizierungsrate der externen Qualität ist größer als die Fehlklassifizierungsrate der internen Qualität (Betrachtungsweise der externen Qualität).

		Realität	
		H_0 ist wahr	H_1 ist wahr
Ergebnis Stichprobentest	H_0 : Fehlklassifizierungsrate des Kunden ist gleich der Fehlklassifizierungsrate des Lieferanten (interne Qualität): $p_{eQ} = p_{iQ}$	Vertrauensniveau $1-\alpha$ Sicherheit, dass der Algorithmus mit einer guten Fehlklassifizierungsrate p_{iQ} richtigerweise angenommen wird.	Fehler 2. Art β Risiko, dass der Algorithmus mit einer schlechten Fehlklassifizierungsrate p_{eQ} fälschlicherweise angenommen wird.
	H_1 : Fehlklassifizierungsrate des Kunden ist größer als der Fehlklassifizierungsrate des Lieferanten (externe Qualität): $p_{eQ} > p_{iQ}$	Fehler 1. Art α Risiko, dass der Algorithmus mit einer guten Fehlklassifizierungsrate p_{iQ} fälschlicherweise abgelehnt wird.	Teststärke/Power $1-\beta$ Sicherheit, dass der Algorithmus mit einer schlechten Fehlklassifizierungsrate p_{eQ} richtigerweise abgelehnt wird.

Abbildung 2.16: Darstellung des Hypothesentests und dessen vier Fälle bei der Durchführung eines Stichprobentests mit einer Annahmegrenze [83, S. 103].

Für diese Hypothesen wird unter anderem die Irrtumswahrscheinlichkeit berechnet. Sie gibt an, mit welcher Wahrscheinlichkeit der Irrtum auftreten kann, sodass die wahre Nullhypothese H_0 fälschlicherweise abgelehnt wird (Fehler erster Art). In der vorliegenden Arbeit gibt diese das Risiko an, wodurch der Algorithmus mit der internen Fehlklassifizierungsrate p_{iQ} fälschlicherweise abgelehnt wird. Dahingegen beschreibt das Vertrauensniveau $(1 - \alpha)$ die Wahrscheinlichkeit, dass die wahre Nullhypothese H_0 richtigerweise angenommen wird. Sie beschreibt somit die Sicherheit, mit welcher der Algorithmus mit einer internen Fehlklassifizierungsrate p_{iQ} richtigerweise angenommen wird. [83, S.103] [81, S. 1–10]

Wird hingegen die Nullhypothese H_0 angenommen, obwohl sie in Wirklichkeit falsch ist, wird vom β -Fehler bzw. Fehler zweiter Art gesprochen. Dieser Fehler hätte sehr kritische Folgen, da somit ein Klassifikationsalgorithmus angenommen wird, der jedoch nicht den Anforderungen entspricht. In der vorliegenden Arbeit bestünde somit das Risiko, dass ein Algorithmus mit der externen Fehlerklassifizierungsrate p_{eQ} fälschlicherweise angenommen wird. Durch die Power bzw. Teststärke $(1 - \beta)$ ist die Wahrscheinlichkeit

definiert, dass eine de facto falsche Nullhypothese H_1 richtigerweise abgelehnt wird, wodurch kein β -Fehler entstehen würde. Bei der Teststärke wird somit die Sicherheit beschrieben, dass der Algorithmus mit der externen Fehlklassifizierungsrate p_{eQ} richtigerweise abgelehnt wird. [83, S.103] [81, S. 1–10]

Zusammenfassend muss somit ein Vertrauensniveau gewählt werden, wobei der Klassifikationsalgorithmus mit der Fehlklassifizierungsrate p_{iQ} angenommen wird, und gleichzeitig die Teststärke/Power, bei welcher der Klassifikationsalgorithmus mit der Klassifizierungsrate p_{eQ} abgelehnt wird. Dabei wird für das Vertrauensniveau $(1 - \alpha) = 80\%$ und für die Teststärke/Power $(1 - \beta) = 95\%$ gefordert. [81, S. 1–10]

Durch die Kenntnis der oben beschriebenen und definierten Größen lässt sich mithilfe der Binomialverteilung die Teststärke berechnen und überprüfen, ob die definierten Vorgaben erfüllt sind. Ziel des Verfahrens ist somit die Ermittlung der Stichprobengröße und der dazugehörigen Anzahl an Fehlklassifizierungen (Annahmegrenze), bei welcher ein Klassifikationsalgorithmus mit einer Fehlerklassifizierungsrate p_{iQ} durch das entsprechende Vertrauensniveau angenommen wird. Zusätzlich soll aber ein Klassifikationsalgorithmus mit einer Fehlerklassifizierungsrate p_{eQ} mit der vorgegebenen Teststärke abgelehnt werden.

2.5 Industrie 4.0

Automatisierungssysteme besitzen nur begrenzte Möglichkeiten, große Mengen an Daten in verschiedenen Umgebungen zu erheben, zu speichern und in Echtzeit zu analysieren [85, S. 27]. Für die Verwendung präskriptiver Analysen im industriellen Umfeld werden aufgrund dessen leistungsstarke Rechenressourcen benötigt [86, S. 145].

Der Begriff Industrie 4.0 beschreibt die vierte Stufe der industriellen Revolution. Er wurde unter anderem von diversen Verbänden wie dem Verband Deutscher Maschinen- und Anlagenbau (VDMA), Zentralverband Elektrotechnik- und Elektronikindustrie (ZVEI) und renommierten Forschungseinrichtungen, Unternehmen und Firmen definiert. Auf Basis internetbasierter Technologien kann eine gewinnbringende Verbindung zwischen IT, Mensch und Maschine geschaffen werden [87, S. 6], [88, S. 4]. Die

2. Grundlagen

technischen Hauptkomponenten der Industrie 4.0 sind cyber-physische Systeme (CPS), das Internet of Things (IoT) sowie Big Data und die in Abschnitt 2.3 beschriebenen Datenanalysen [89].

2.5.1 Cyber-physische Systeme

Die Verknüpfung informationsverarbeitender Elemente mit physikalischen Objekten und Prozessen existiert in Automatisierungssystemen bereits seit den 1970er Jahren. Damals wie heute waren auch diese vernetzt und kommunizierten untereinander. Der wesentliche Unterschied zu CPS ist, dass bei diesen eine Vernetzung über offene und globale Informationsnetze hinzugefügt wird [90, S. 2]. CPS sind eine neue Generation von Systemen, in welchen rechnerische und physische Fähigkeiten integriert sind. Dabei überwachen und steuern eingebettete Computer und Netzwerke beispielsweise die physikalischen Produktionsprozesse [91, S. 21–22]. CPS zeichnen sich durch die Verbindung der physisch-analogen mit der informationstechnischen Welt aus [91, S. 24–25].

CPS werden im industriellen Produktionskontext unter dem Begriff Industrie 4.0 als cyber-physisches Produktionssystem (CPPS) bezeichnet. Aufgrund einer hohen Datenverfügbarkeit entsteht dadurch die Vision einer sich selbst konfigurierenden Produktionsanlage. Die Speicherung der Dienste, Funktionen und Daten erfolgt bei CPPS direkt an dem Ort, an welchem sie abgerufen und ausgeführt wurden. Dies führt zu einer kostenoptimierteren und effizienteren Produktion als auch zu einer Auflösung der klassischen Automatisierungspyramide. Außerdem können durch die lokale Verarbeitung der Daten echtzeitkritische Steuerungen und Regelungen vorgenommen werden. Da die klassische Ebene der Automatisierungspyramide durch dezentrale Dienste in diversen Knoten abstrahiert wird, sind für die Verwendung von CPPS eine effiziente Datenspeicherung und robuste Kommunikationsstrukturen vonnöten [90, S. 3–4].

2.5.2 Internet der Dinge

Eine weitere Kerntechnologie von Industrie 4.0 ist das IoT. Aufgrund der Diversität verschiedener Produktions- und IT-Systeme sowie Anforderungen an die Kommunikation ist eine Standardisierung substanziell. Ziel ist deswegen eine vernetzte Kommunikation zwischen diversen wertschöpfungsorientierten Systemen, unter anderem im produzierenden Gewerbe [92, S. 24]. Das IoT stellt hierfür diverse Interaktionsmechanismen und

Kommunikationsmodule bereit [93, S. 7]. Zur Erfüllung der Anforderungen an die Kommunikation werden Systeme zur Übermittlung, Verarbeitung und Abfragen von Daten gefordert. Eine dabei verwendete IT-Architektur ist sowohl das Edge- als auch das Cloud-Computing [94, S. 13]. Sie ermöglichen die Verarbeitung relevanter Daten, um beispielsweise einen Prozesses mithilfe neuronaler Netze zu optimieren [95, S. 376–377].

2.5.2.1 Cloud-Computing

Ein verbreiteter Ansatz zur Verarbeitung großer Datenmengen ist das sogenannte Cloud-Computing. Dabei werden die gesammelten Daten auf einer zentralen Plattform, welche mit leistungsstarken Ressourcen ausgestattet ist, verarbeitet. Cloud-Computing bietet einen On-Demand-Netzwerkzugriff auf einen Pool von Verarbeitungsressourcen. Diese umfassen beispielsweise Rechenleistung, Speicher oder Dienste, welche mit geringem Aufwand abgerufen werden können [96, S. 1], [97, S. 2]. Diese Ressourcen können über drei verschiedene Servicemodelle beschrieben werden. Infrastructure as a Service (IaaS) bietet verschiedene Computing-Ressourcen an, die nach Belieben gebucht und genutzt werden können. Der Nutzer erhält dadurch die komplette Kontrolle über die Ressourcen, muss sich jedoch ebenfalls um die Einrichtung und Verwaltung kümmern. Ein Beispiel hierfür ist das Hosten einer Webseite auf gebuchten und konfigurierten Ressourcen. Bei Platform as a Service (PaaS) handelt es sich um die Bereitstellung von Anwendungen der Cloud-Anbieter. Diese stellen die dafür benötigten Ressourcen zur Verfügung und verwalten diese im Gegensatz zu IaaS selbst. Dadurch können selbst entwickelte Anwendungen bereitgestellt und genutzt werden. Beispielsweise können selbst erstellte ML-Anwendungen auf einer dafür vom Cloud-Anbieter vorkonfigurierten Infrastruktur bereitgestellt werden. Software as a Service (SaaS) beschreibt ein Dienstleistungsmodell, bei welchem der Nutzer eine vom Cloud-Anbieter zur Verfügung gestellte Softwareanwendung nutzen kann. Der Anbieter verwaltet den gesamten Soft- und Hardware-Stack, der für den Betrieb der Anwendung notwendig ist. Der Nutzer kann die Anwendung lediglich verwenden. Ein Beispiel hierfür ist das Web-Interface eines E-Mail-Clients [98, S. 378], [28, S. 140].

Außerdem wird Cloud-Computing in vier verschiedene Bereitstellungsmodelle unterteilt [99, S. 5349], [97, S. 3]:

2. Grundlagen

- Öffentliche Cloud: Hierbei kann jede Person die Ressourcen des Cloud-Anbieters nutzen.
- Gemeinschaftliche Cloud: Bei dieser Art des Bereitstellungsmodells ist der Zugriff nur für Personen möglich, die beispielsweise bestimmte Interessen verfolgen oder dem selben Projekt angehören.
- Private Cloud: Auf diese Art der Cloud-Technologie können ausschließlich Personen einer bestimmten Organisation zugreifen.
- Hybride Cloud: Sie ist eine Kombination aus öffentlicher, privater und gemeinschaftlicher Cloud.

2.5.2.2 Edge-Computing

Der Datenverkehr zwischen einer Produktionsanlage und der Cloud stellt bei einer großen Datenmenge eine große Herausforderung dar. Aufgrund dessen werden alternative Konzepte benötigt. Im Gegensatz zum Cloud-Computing wird die Erfassung und Verarbeitung der Daten durch Edge-Computing zu einer anlagennahen Ebene allokiert. Dabei wird Edge-Computing als Datenverarbeitung und -transport zwischen dem Datenproduzenten und dem Cloud-Datencenter definiert [100, S. 637–638]. Vorteile hierbei sind eine deutlich kürzere Latenzzeit, wodurch schneller auf Ereignisse reagiert werden kann. Bei Anwendungen mit echtzeitkritischen Anforderungen, wie beispielsweise die Erkennung und Behebung von Qualitätsabweichungen in der Produktion, kann die Verlagerung der Datenverarbeitung zu einem reduzierten Ausschuss führen [94, S. 13], [101, S. 2f.], [102, S. 37]. Außerdem führt Edge-Computing zu einem geringeren Datenverlust und einer Verbesserung der Datensicherheit und Privatsphäre der Daten, da zum Beispiel personenbezogene Daten direkt nach der Generierung verarbeitet und anonymisiert werden können [100, S. S. 637–638].

In dieser Arbeit wird eine Kombination aus Cloud- und Edge-Computing verwendet. Dabei kommt eine öffentliche Cloud zum Einsatz, wobei sicherheitsspezifische Daten auf dem Edge-Gerät vorverarbeitet werden. Über eine Wide Area Network-Technologie besteht eine direkte Verbindung zum Datencenter des Cloud-Anbieters, wodurch eine hohe Verfügbarkeit der Ressourcen sichergestellt werden kann. Außerdem wird eine Kombination aus allen drei Servicemodellen verwendet.

2.5.2.3 Microsoft Azure

Der in dieser Arbeit verwendete Cloud-Anbieter ist Microsoft Azure. Dieser stellt eine wie in Abschnitt 2.5.2.1 beschriebene öffentliche Cloud-Plattform bereit, wodurch diverse Cloud-Dienste genutzt werden können. Neben den von Microsoft angebotenen Leistungen können auch selbst entwickelte Dienste, welche im Folgenden als Module bezeichnet werden, und Anwendungen von Drittanbietern genutzt werden [103]. Die IoT-Edge-Laufzeit ist eine von Microsoft zur Verfügung gestellte Plattform. Durch diese wird die Anwendung von Cloud-Funktionalitäten ermöglicht. Außerdem können dadurch eigene Module auf einem Edge-Gerät bereitgestellt und ausgeführt werden. Die IoT-Edge-Laufzeit besteht aus den folgenden Hauptkomponenten [104]:

- IoT-Edge-Hub: Dieser dient als zentrale Kommunikationsschnittstelle zwischen den unterschiedlichen Modulen des Edge-Gerätes. Die Kommunikation der voneinander getrennten Module wird durch einen Publish-Subscribe-Mechanismus mittels Message Queuing Telemetry Transport (MQTT) ermöglicht. MQTT ist ein offenes Netzwerkprotokoll, welches für eine Machine-to-Machine-Kommunikation im Bereich des IoT Standard ist. Außerdem beruht es auf dem Broker-Client-Prinzip. Die Module fungieren dabei als Client und der IoT-Edge-Hub, welcher den Nachrichtenbroker darstellt, als Server. Jedes Modul verfügt über die Option, Daten an beliebig viele Module zu distribuieren. Dies wird, wie in Abbildung 2.17 dargestellt, durch sogenannte Topics ermöglicht. Auf diese Topics können verarbeitete Daten und Informationen geschrieben werden. Mittels eines Subscription-Mechanismus ist jedes Modul in der Lage, diesen Datenfluss abzuhören und die darauf geschriebenen Daten zu empfangen.

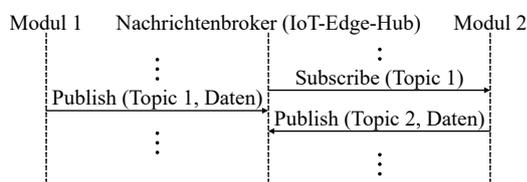


Abbildung 2.17: Datenaustausch von Modulen über einen Nachrichtenbroker.

- IoT-Edge-Agent: Dieser stellt die Laufzeit und Aktualität der Module sicher und gewährleistet die Sicherheit des Systems.

2. Grundlagen

Azure Pipelines ist ein weiterer von Microsoft zur Verfügung gestellter Cloud-Service. Mithilfe dieses Services kann ein Code, wie z. B. die Vorverarbeitung der Daten, automatisch erstellt, getestet und auf dem Edge-Gerät bereitgestellt werden. Dabei wird ein Continuous Integration und Continuous Deployment (CI/CD) verwendet, um den Code fortlaufend und konsistent zu prüfen. Anschließend wird eine Container-Instanz erstellt und diese auf dem Edge-Gerät bereitgestellt [105].

Der letzte in dieser Arbeit von Microsoft verwendete Dienst sind die sogenannten Microsoft Functions. Mit diesem Dienst kann ein selbst entwickelter Code in einer serverlosen Umgebung ausgeführt werden. Vorteil dabei ist, dass dafür kein virtueller Computer erstellt oder eine Web-Applikation veröffentlicht werden muss. Dies bietet eine gute Grundlage zur einfachen Erstellung einer Application Programming Interface (API) [106].

Wie oben erläutert, können Anwendungen von Drittanbietern integriert werden. Für die Speicherung und Verwaltung der Daten wird hierfür in dieser Arbeit eine SQL-Datenbank verwendet. Diese eignen sich für die Verwaltung alphanumerischen Zeichen, wie beispielsweise Buchstaben oder Zahlen, komplexer Datentypen oder multimediale Inhalte, wohingegen herkömmliche relationale Datenbankmanagementsysteme schnell an ihre Leistungsgrenzen stoßen. Objektrelationale Datenbanken oder Datenbankmanagementsysteme ermöglichen die Speicherung von Objekten in relationalen Systemen, die auf objektorientierten Programmiersprachen basieren. Eine der populärsten und zugleich ältesten Lösungsansätze in diesem Kontext ist die Open-Source-Datenbank PostgreSQL [107, S. 3], [108].

Kapitel 3

Stand der Technik

3.1 Optische Erkennung von Qualitätsabweichungen mittels CNN

Wie bereits in Abschnitt 2.4.1 dargestellt, gibt es diverse Algorithmen im Bereich von ML. Aufgrund der in Abschnitt 2.4.5 beschriebenen Vorteile eines CNN in der optischen Qualitätssicherung wird im Folgenden ausschließlich auf deren Anwendung eingegangen. Beispiele hierfür sind die optische Erkennung von Kratzern auf unterschiedlichsten Materialien wie beispielsweise Holz [109], Fasern [110] Stahl [111, 112], Siliziumscheiben, Solarzellen [113] oder auch Schienen-Laufflächen [114, 115]. Sie belegen, dass mit diesem Ansatz für alle betrachteten Anwendungsfälle auch bei sehr kleinen Fehlerbildern eine hohe Genauigkeit zur Detektion von Fehlern erreicht werden kann. Außerdem wird in diesen Veröffentlichungen nachgewiesen, dass ein CNN aufgrund der automatisierten Extraktion von Merkmalen bessere Ergebnisse als modellbasierte oder klassische Algorithmen erzielt. Ein weiterer Vorteil ist, dass durch die Verwendung des CNN kein Expertenwissen nötig ist, da anders als bei klassischen oder händischen Ansätzen aufgrund von fehlendem Expertenwissen keine relevanten Informationen bei der Merkmalsextraktion verloren gehen [51, 116, 117]. Wie in Abschnitt 2.1 beschrieben, ist dies ein großer Vorteil für die Anwendung eines CNN zur optischen Überprüfung von Qualitätsabweichungen in der Produktion des elektrischen Antriebsstranges.

Auch im Bereich der Schweißtechnik existieren Ansätze zur Erkennung von Qualitätsabweichungen mithilfe eines CNN. Hartl, Vieltorf und Zäh befassten sich etwa mit

3. Stand der Technik

den Charakteristika und Abhängigkeiten des Rührreibschweißens. Hierbei untersuchten sie zum einen Zusammenhänge zwischen der Topografie der Schweißnaht und ihren mechanischen Eigenschaften wie beispielsweise die Zugfestigkeit [118]. Zum anderen beschäftigten sich die Autoren mit der Ableitung eindimensionaler Schlüsselindikatoren zur Erkennung und Quantifizierung von Oberflächenfehlern und -merkmalen. Dazu wurden Schweißnähte visuell inspiziert und die Oberflächentopografie mittels Strukturlichtprojektion erfasst. Aus der resultierenden dreidimensionalen Punktwolke konnten eindimensionale Schlüsselindikatoren zur Beurteilung der aufgetretenen Oberflächendefekte und -merkmale abgeleitet werden [119]. Die Autoren nutzten Algorithmen des maschinellen Lernens, um diese Abhängigkeiten zu modellieren. Bei Verwendung der Topographie der Schweißnaht als Eingangsvariable des CNN (DenseNet-121) erzielten die Autoren eine Genauigkeit von 92,1%. Mit dem Spektrogramm erreichten die Autoren eine Genauigkeit von 99,1% [120, 121].

Bei Anwendungen im Produktionsbereich gibt es jedoch im Vergleich zu den oben genannten Veröffentlichungen andere Anforderungen an das CNN. Ein Beispiel hierfür sind zeitliche Anforderungen an die Geschwindigkeit des Klassifikationsalgorithmus. Aufgrund dessen wird im Folgenden auf Veröffentlichungen der optischen Erkennung von Qualitätsabweichungen in der Produktionsumgebung mittels eines CNN eingegangen.

3.1.1 Optische Erkennung von Qualitätsabweichungen in der Produktionsumgebung mittels CNN

Birlutiu et al., Lin et al. und Kang et al. publizierten jeweils CNN Lösungsansätze, um in einer realitätsnahen Produktionsumgebung Qualitätsabweichungen optisch zu erkennen. Diese Ansätze wurden beispielsweise in einer Porzellanfabrik, in einer Fertigung für LEDs sowie zur Überprüfung einer Schweißnaht angewandt. Die Ergebnisse dieser Veröffentlichungen belegen, dass mittels eines CNN unterschiedliche Fehler auf Bildern in verschiedensten Anwendungsfällen mit einer hohen Genauigkeit detektiert werden können [122, 123, 124, 125]. Im Bereich der Automobilproduktion gibt es derzeit nur wenige Ansätze zur Erkennung von Qualitätsabweichungen mithilfe eines CNN. Diese Publikationen sind nachfolgend beschrieben.

3.1 Optische Erkennung von Qualitätsabweichungen mittels CNN

Leichtmetallgussteile wie zum Beispiel Felgen, Achsschenkel und Lenkgetriebe sind wichtige Komponenten eines Automobils. Während des Produktionsprozesses können jedoch inhomogene Bereiche wie beispielsweise blasenförmige Hohlräume, Brüche oder Einbrüche entstehen. Um die Sicherheit gewährleisten zu können, werden alle Bauteile geprüft. Mery stellt hierfür ein CNN vor, mit welchem fehlerhafte Gussteile erkannt werden können. In einem ersten Versuch wurde eine optimale Struktur des CNN gesucht. Um einen ausgeglichenen und großen Datensatz zu erhalten, verwendet der Autor Data Augmentation (DA). Das antrainierte CNN unterscheidet dabei zwischen zwei Klassen und besitzt eine Genauigkeit von 96,9% im Testdatensatz. [126]

In ihrer Veröffentlichung stellen Mo und Chen ein Verfahren vor, um Schweißstellen an Seitentüren eines Automobils erkennen und klassifizieren zu können. Aufgrund komplexer Betriebsbedingungen ändert sich kontinuierlich die Lage der Schweißpunkte. Falls die Position der Lötstelle nicht automatisch und genau identifiziert werden kann, ist es notwendig, den Laufweg des Schweißroboters manuell einzustellen, was zu einer geringen Automatisierung und Intelligenz der Produktionslinie führt. Um diese Nachteile zu beheben, verwenden die Autoren das YOLOv3 Netzwerk [128] und das Faster-RCNN [129]. Das YOLOv3 Netzwerk erreicht eine höhere Genauigkeit und kann die Schweißpunkte mit einer durchschnittlichen Genauigkeit von 85,2% erkennen und klassifizieren. Außerdem konnte die Zeit im Vergleich zur maschinellen Bildverarbeitung um das Fünzigfache gesenkt werden. [127]

Ooi, Tay und Lai stellen in deren Veröffentlichung ein CNN vor, um Defekte wie Beulen, Dellen und Kratzer auf Scheibenwischern zu erkennen. Da die produzierte Menge hoch ist, ist es sehr schwierig, die Qualität manuell zu überwachen. Dennoch ist es derzeit üblich, manuelle Inspektionen durchzuführen, was aber zu hohen Produktionskosten und Qualitätsproblemen führt. Um diesem Nachteil entgegenzuwirken, entwickelten die Autoren ein automatisches Fehlerprüfsystem, das Defekte in der Produktion erkennen und klassifizieren kann. Dabei erreichten die Autoren eine Genauigkeit von 89%. [130]

Suescún et al. befassen sich in ihrer Arbeit mit dem Erkennen von Kratzern im Autolack. Durch das R-CNN wird zunächst das Automobil erkannt und anschließend wird

3. Stand der Technik

das Bild mithilfe des CNN auf Kratzer untersucht. Die Objektdetektion mithilfe des R-CNN besitzt eine Genauigkeit von 98,3%. Zur Erkennung der Kratzer verwendeten die Autoren insgesamt 15.510 Bilder. Die Genauigkeit zur Erkennung der Kratzer mithilfe des von den Autoren designten CNN betrug 69,89%. Die Zeit, um alle Kratzer auf dem Auto zu detektieren, beträgt dabei 2,92 Sekunden. [131]

Aufgrund der steigenden Variantenvielfalt in der Automobilindustrie steigt die Komplexität in der Logistik. Dies kann dazu führen, dass falsche Komponenten am Fahrzeug angebracht werden [132]. Die Autoren stellen eine Methode vor, wodurch verschiedene Arten von Felgen klassifiziert werden können. Dafür erstellen sie einen Datensatz mit insgesamt 60.000 Bildern. Diese Bilder wurden zuerst vorverarbeitet und anschließend ein von den Autoren entwickeltes CNN mit diesen Bildern antrainiert. Die Genauigkeit des Netzes beträgt dabei 95,6%. Außerdem ist die Größe des Netzes klein gehalten, sodass die Klasse der Felge in Echtzeit prädiziert werden konnte. [133]

Wie bereits in Abschnitt 2.1 beschrieben, ergibt sich aus der Veränderung vom Verbrennungs- zum Elektromotor eine Verschiebung der Bedeutung von Fertigungstechnologien. Dies wird vor allem zerspanende Fertigungsverfahren betreffen, welche deutlich weniger Anwendung finden [134, S. 4]. Dagegen steigt der Anteil umformender, trennender und fügender Fertigungsverfahren, wie Pressen, Ziehen, Biegen, Stanzen und Schweißen [135]. Zur Erkennung von Qualitätsabweichungen eignen sich ML-Techniken, insbesondere ein CNN, da mit deren Hilfe Qualitätsabweichungen erkannt werden können, ohne dass ein spezielles Expertenwissen vorhanden sein muss [136, S. 5], [137].

Innerhalb der Produktion von Elektroantrieben stellen die Autoren Weigelt et al. ein CNN vor, um die Ultraschall-Crimp Verbindung eines Stators zu überprüfen. Neben der Montage der Wicklung bietet der Prozess der Kontaktierung der Drahtenden mit Kabelschuhen ein großes Optimierungspotenzial. Jedoch ist das Kontaktierungsverfahren Ultraschall-Crimpen aufgrund der Vielzahl der Einflussfaktoren schwer beherrschbar. Zur optischen Überprüfung der Verbindung stellen die Autoren ein bildbasiertes CNN vor, mit welchem die Vorhersage der Qualität der Verbindung möglich ist. Mithilfe des CNN erreichen sie hierbei eine Genauigkeit von 91%. [138]

3.1 Optische Erkennung von Qualitätsabweichungen mittels CNN

Die Autoren Mayr et al. befassen sich mit der Erkennung von Schweißfehlern von Hairpin-Paaren. Zunächst befassen sie sich mit der Vorhersage der Schweißqualität auf Basis der Maschinenparameter. Dazu vergleichen sie verschiedene Algorithmen wie Support Vector Machine (dt. Stützvektormaschine) [139], Random Forest (dt. Zufallswald) [140], k-Nearest-Neighbor (dt. k-nächste-Nachbarn) [141] oder künstliche neuronale Netze [142]. Neben der Vorhersage der Schweißnahtqualität auf der Grundlage von Maschinenparametern untersuchten die Autoren eine optische Erkennung des Schweißprozesses mithilfe einer Kamera und eines CNN. In Abhängigkeit des Fehlers konnte eine Genauigkeit zwischen 61% und 91% erreicht werden. [143]

3.1.2 Abgrenzung zur Literatur

In Abschnitt 3.1.1 wurden relevante Vorarbeiten im Hinblick auf die Zielsetzung dieser Arbeit betrachtet. In diesem Rahmen wurde zunächst der Stand der Technik und Forschung in Bezug auf die optische Qualitätsüberwachung durch ein CNN in der Produktion dargestellt. Wie zu erkennen ist, gibt es derzeit wenige relevante Vorarbeiten, welche sich mit dem Thema der optischen Qualitätsüberwachung durch ein CNN in der Produktion, vor allem in der Automobilproduktion, sowie der Produktion elektrischer Antriebssysteme befassen. In Tabelle 3.1 sind die beschriebenen Arbeiten im Hinblick auf die Erfüllung der in Abschnitt 4.2 beschriebenen Anforderungen aufgelistet.

Tabelle 3.1: Erfüllung der Anforderungen vergleichbarer Literatur im Bereich der Qualitätssicherung mittels eines CNN.

Anforderungen	Publikationen									
	[122]	[123]	[125]	[126]	[127]	[130]	[131]	[133]	[138]	[143]
Genauigkeit >99%	○	○	●	○	○	○	○	○	○	○
Kurze Prädiktionszeit	●	○	○	●	○	○	○	●	●	○
Mehrklassen-Klassifizierung	●	○	○	○	●	●	○	●	○	●
Präskriptive Analyse	○	○	○	○	○	○	○	○	○	○

○ Anforderungen nicht erfüllt ● Anforderungen erfüllt

3. Stand der Technik

Es ist zu erkennen, dass nur einer der Autoren eine Genauigkeit bei der Fehlererkennung größer als 99% erreichen konnte. Für eine erfolgreiche Umsetzung einer Qualitätsüberwachung in der Serienproduktion ist dies jedoch essenziell. Denn ohne eine hohe Genauigkeit können Bauteile, welche von den Qualitätsanforderungen abweichen, nicht detektiert werden. Falls diese zum Kunden gelangen, könnte dies zu einer schlechten Reputation des Unternehmens oder sogar zur Gefährdung der Sicherheit von Mensch, Tier oder Umwelt führen. Das Ziel der vorliegenden Arbeit ist somit das Erreichen einer Genauigkeit von mindestens 99%, was im Vergleich zum bisherigen Stand der Technik eine deutliche Verbesserung wäre.

Ein weiteres Defizit der oben beschriebenen Arbeiten ist die bereits angedeutete Dauer der Prädiktionszeit, welche durch die Größe des CNN beeinflusst wird. Eine kurze Prädiktionszeit wird jedoch bei produktionstaktkritischen Anwendungen benötigt, damit die Taktzeit aufgrund der Qualitätssicherung nicht negativ beeinflusst wird. Wie erläutert erreichten ausschließlich Park, An und Kang eine Genauigkeit über 99%. Jedoch besitzt das dabei verwendete CNN insgesamt 1.036.066 Parameter. Mery verwendet ein noch größeres CNN mit insgesamt 1,3 Millionen Parametern. Die weiteren Arbeiten verwenden ein möglichst kleines Netz. Jedoch beträgt deren Genauigkeit lediglich zwischen 85,2% und 95,6%. Ziel dieser Arbeit ist somit das Erstellen eines kleinen CNN bei gleichzeitig hoher Genauigkeit. [127, 133, 138]

Ein weiteres Ziel ist die Entwicklung eines Netzes, welches eine Mehrklassen-Klassifikation vornehmen kann, da meist die Nacharbeitsstrategien vom Fehlerfall abhängig sind. Bei einer binären Klassifikation wird beispielsweise nur zwischen einem guten und schlechten Bauteil unterschieden. Nur die Hälfte der Ansätze, welche sich mit der optischen Qualitätskontrolle mithilfe eines CNN befassen, haben diese Option berücksichtigt.

Das letzte Ziel der vorliegenden Arbeit ist die Integration einer präskriptiven Analyse. Wie in Abschnitt 1.1 dargelegt, ist es im Rahmen der heutigen Prozessoptimierung essenziell, durch prädiktive Analyse Prozessanomalien bereits im laufenden Produktionsprozess vorherzusagen und darüber hinaus durch präskriptive Analyse proaktiv Gegenmaßnahmen einzuleiten. Wie jedoch in Tabelle 3.1 zu erkennen, gibt es derzeit

keine Publikation, welche die prädiktive Analyse eines CNN mit der in Abschnitt 2.3.2 beschriebenen präskriptiven Analyse verbindet. Um einen möglichst großen Mehrwert der Prozessoptimierung zu erreichen, wird aufgrund dessen dieses weitere Ziel in der vorliegenden Arbeit angestrebt.

3.2 Echtzeitfähige Edge-Computing-Lösung

In diesem Abschnitt wird auf den Stand der Technik in Bezug auf Edge-Computing in Verbindung mit Techniken des maschinellen Lernens im produzierenden Umfeld eingegangen. Es gibt derzeit viele Veröffentlichungen der Anwendung von Edge-Computing im Bereich Gesundheitswesen, Smart Home und der Landwirtschaft. Im Vergleich zum produzierenden Gewerbe besitzen diese jedoch andere Randbedingungen. Außerdem ist anhand der veröffentlichten Arbeiten zu erkennen, dass erst in den letzten drei Jahren die Integration von Edge-Computing im produzierenden Umfeld Anwendung fand. Problematisch ist darüber hinaus, dass es den oben genannten Systemen im Allgemeinen an einer Validierung auf der Grundlage komplexer industrieller Szenarien mangelt. Relevante Analysen und Ergebnisse basieren hauptsächlich auf Simulationen [144], [145], [146].

Um einen Überblick über diese Ansätze zu erhalten, werden in Abschnitt 3.2.1 die für die vorliegende Dissertation relevantesten Veröffentlichungen aufgelistet. In Abschnitt 3.2.2 wird aus diesem Stand der Technik ein Handlungsbedarf abgeleitet.

3.2.1 Umsetzung der echtzeitfähigen Edge-Computing-Lösung in der Produktionsumgebung

In ihrer Veröffentlichung beschreiben Wang et al. ein intelligentes Sortiersystem. Mithilfe eines PCs, welcher als Edge-Gerät fungiert, wurden Bilder aus der Fertigungslinie gezogen, um ähnliche Teile zu detektieren und zu klassifizieren. Dieser PC ist zudem mit der Cloud verbunden, worauf der Klassifikationsalgorithmus antrainiert wird. Der Algorithmus basiert auf einem CNN. Wurde ein nicht passendes Paar in der Produktion erkannt, erhielt der Anwender über die Cloud eine Nachricht. Als Anwendungsfall wurde dabei die Herstellung von Felgen betrachtet. [147]

3. Stand der Technik

Hou et al. thematisieren den Einsatz von Edge-Computing und maschinellem Lernen zur frühzeitigen Erkennung von Prozessproblemen innerhalb der Halbleitertechnik. Bisher wurden Fehler von Experten unter der Verwendung von digitalen Mikroskopen identifiziert. Zur Überprüfung der Qualität wird ein NN verwendet. Die entwickelte Lösung besteht unter anderem aus einem Single Board Computer (SBC), welcher in diesem Fall das Edge-Gerät darstellt, und einem digitalen Mikroskop, das über USB mit dem Edge-Gerät verbunden wurde. Die Qualitätsüberwachung findet stichprobenartig statt. Dazu werden Wafer aus der Fertigung genommen und überprüft. [148]

Trinks und Felden beschreiben in ihrer Arbeit ein echtzeitfähiges Qualitätsmanagementsystem zur Steigerung der Effizienz der Produktionsprozesse. Zur Verringerung der Latenz erfolgt eine Verlagerung der Hard- und Softwarekomponenten in die Nähe des Prozesses. Dabei wird der Druckprozess mithilfe eines Bildsensors und eines ML-Algorithmus überwacht und durch das entwickelte System validiert. Kann ein Fehler detektiert werden, wird der jeweilige Druckprozess angehalten und eine Nachricht an den zuständigen Arbeiter versendet. Sollte sich das System geirrt haben, kann der Prozess fortgesetzt werden. Handelt es sich korrekterweise um einen Fehler, kann ein Roboterarm das fehlerhafte Teil entnehmen, wodurch im Anschluss der Prozess neu gestartet wird. Es handelt sich dabei jedoch nicht um eine Serienproduktion mit hoher Taktzeitanforderung. [149]

Lin et al. beschäftigen sich mit der Planung komplexer Halbleiter-Fertigungssysteme und der damit verbundenen Aufgabenverteilung. Zur Verringerung der Antwortzeiten von Entscheidungen werden Verarbeitungsschritte durch Edge-Computing an den Rand des Netzwerkes verlagert. Zur Planung der Fertigungsschritte wird eine Kombination aus Deep Learning und Reinforcement Learning eingesetzt. Informationen der Maschine werden gesammelt, verarbeitet und an eine zentrale Einheit, ein Cloud-Datencenter, übermittelt. In der Cloud wird die Vorhersage der Aufgabenplanung generiert und an die jeweiligen Edge-Geräte gesendet. [150]

Luckow et al. evaluieren in ihrer Veröffentlichung verschiedene Einsatzmöglichkeiten von Deep Learning im logistischen Bereich einer Automobilfertigung. Eine Verbesserung der Qualität und Effizienz der Prozesse durch die Verwendung von Computer

Vision wurde angestrebt. Die untersuchten ML-Modelle wurden auf einem Edge-Gerät bereitgestellt. Das Edge-Gerät ist hierbei ein Smartphone des Personals. Da Daten verschiedener Quellen zusammengeführt werden mussten, wurden die Modelle auf einer Cloud-Plattform bereitgestellt. [151]

Queiroz et al. beschreiben in ihrer Arbeit einen agentenbasierten Edge-Computing-Ansatz zur Detektion von Fehlerfällen in der Automobilproduktion. Es werden Daten an zwei Stationen (Framing und Finish Line) der Produktionslinie erfasst. Dabei werden Spaltmaße im Heckbereich auf Qualitätsmängel untersucht. Mangels einer nicht ausreichenden Datenmenge wurde eine Anomalieerkennung mittels Unsupervised Learning-Algorithmen eingesetzt, um Abweichungen zu erfassen und so früh wie möglich eine Nacharbeit einsteuern zu können. [152]

Syafrudin et al. beschreiben in ihrer Arbeit ein kostengünstiges Frühwarnsystem basierend auf einer Edge-Computing-Lösung. Zur Detektion der Fehler wird ein hybrides Vorhersagemodell, bestehend aus verschiedenen ML-Algorithmen, eingesetzt. Um echtzeitnah handeln zu können, wurden die insgesamt vier Edge-Geräte in der Nähe des Produktionsprozesses installiert. Der Fokus liegt hierbei auf der Überwachung und Erkennung von Fehlern. Niedrige Systemkosten konnten durch die Nutzung von Open Source-Software in Kombination mit einem leistungsschwächeren SBC erreicht werden. Im Falle eines Fehlers wird über die Push-Benachrichtigung eines Smartphones das Personal informiert. [153]

3.2.2 Abgrenzung zur Literatur

Der Einsatz von Edge-Computing bietet sowohl das Potenzial der Optimierung von Produktionssystemen durch eine erhöhte Reaktionsgeschwindigkeit als auch die Verarbeitung großer Datenmengen. Wie in Abschnitt 3.2.1 beschrieben, mangelt es jedoch derzeit an Lösungen, die dieses Potenzial nutzen, sowie an der fehlenden Verbreitung dieser Prinzipien im produzierenden Umfeld [154, 144]. In Tabelle 3.2 sind die oben beschriebenen Veröffentlichungen aufgelistet. Diese werden mit den in Abschnitt 4.2 erläuterten Anforderungen an eine Cloud- und Edge-Computing-Lösung in der Produktionsumgebung verglichen.

3. Stand der Technik

Tabelle 3.2: Erfüllung der Anforderungen vergleichbarer Literatur im Bereich des Edge-Computings.

Anforderungen	Publikationen						
	[147]	[148]	[149]	[150]	[151]	[152]	[153]
Kritische Prozesszeit	○	○	○	◐	○	◐	◐
Erkennung von Qualitätsabweichungen	●	●	●	○	○	●	●
Präskriptive Analyse	○	○	○	●	○	◐	○
Geschlossener Regelkreis	○	○	○	●	○	○	○
Modularer Aufbau	○	○	○	○	○	○	○
Zentrale Verwaltung	○	○	○	○	○	○	○

○ nicht angewandt ◐ angesprochen ● angewandt

Wie in Tabelle 3.2 zu erkennen, befinden sich die derzeitigen Anwendungen nicht im Bereich einer Produktion mit hohen Taktzeiten bzw. kurzen Prozesszeiten. Ein weiteres Defizit ist, dass Edge-Computing in der Produktionsumgebung zum Großteil in Kombination mit prädiktiven Analyseverfahren untersucht wird. Es gibt jedoch nur wenige Publikationen, die Edge-Computing im Kontext von präskriptiver Analyse betrachten. Darüber hinaus wird die generierte Handlungsempfehlung meist durch das manuelle Eingreifen eines Menschen und nicht mittels eines geschlossenen Regelkreises umgesetzt, was den Forschungsbedarf unterstreicht. Ausschließlich Lin et al. beschäftigen sich in deren Arbeit mit Planung und Umsetzung von Fertigungsschritten mittels Edge-Computing. Nachteil hierbei ist jedoch, dass die Planung in der Cloud lokalisiert ist und ebenfalls durch diese gesteuert wird; somit sind harte Echtzeitanforderungen nicht einzuhalten. Das Edge-Gerät übernimmt ausschließlich das Sammeln der relevanten Daten. Somit gibt es derzeit keine Veröffentlichung, welche die Erkennung von Qualitätsabweichungen mittels maschinellen Lernens und einer automatisierten Generierung einer Handlungsempfehlung, zum Beispiel in Form einer Nacharbeit, durch ein Edge-Gerät in der Produktion umsetzt. Ein weiterer Nachteil der Publikationen ist, dass kein modularer Aufbau der Systeme gegeben ist. Somit können diese Systeme nicht ohne starke Anpassungen bei weiteren Anwendungen verwendet werden. Außerdem gibt

3.2 Echtzeitfähige Edge-Computing-Lösung

es keine zentrale Verwaltung. Somit müssen zum Beispiel Anpassungen der Software physisch vor Ort und nicht zentral durchgeführt werden.

Darüber hinaus gibt es derzeit nur eine Anwendung, bei der Cloud-Computing bei der Herstellung von Elektromotoren eingesetzt wird. Zusätzlich sind in diesem Bereich keine Edge-Computing-Anwendungen bekannt [155]. Daher erweitert diese Arbeit das Anwendungsspektrum von Edge-Computing und kombiniert zwei innovative Technologien: die Herstellung des elektrischen Antriebsstrangs eines Automobils mit Cloud- und Edge-Computing.

Kapitel 4

Präskriptive Prozessoptimierung Methode

4.1 Vorgehensweise

In diesem Kapitel wird der in Kapitel 3 abgeleitete Handlungsbedarf für eine vollständige Inline-Prüfung durch ein CNN und eine automatisierte Anpassung der Prozessparameter beschrieben. Dafür wird im Folgenden eine geeignete Vorgehensweise erläutert.

Da es sich um ein System handelt, das sowohl mechanische, elektronische als auch informationstechnische Komponenten vereint, wird ein Vorgehensmodell benötigt, welches die domänenübergreifenden Herausforderungen im Entwicklungsprozess berücksichtigt. Die Verein Deutscher Ingenieure (VDI)-Richtlinie 2206, welche die Entwicklungsmethode für mechatronische Systeme beschreibt, stellt aufgrund dessen ein gutes Vorgehensmodell dar [156, S. 3–4]. Dabei wird zwischen der Vorgehensweise auf der Mikro- bzw. Makroebene unterschieden. Diese beiden Vorgehensweisen werden im Folgenden erläutert.

4.1.1 Problemlösezyklus als Mikrozyklus

Der in der VDI 2206 vorgeschlagene Mikrozyklus zur Entwicklung von mechatronischen Systemen basiert auf dem Problemlösezyklus. Er stammt aus dem Systems Engineering und beschreibt ein Vorgehen zur Handlungsorganisation. Diese besteht aus folgenden vier Phasen [156, S. 27–29]:

4. Präskriptive Prozessoptimierung Methode

1. **Situationsanalyse:** Dabei kann entweder ein gegebenes Ziel (Soll-Zustand-orientiertes Vorgehen) angenommen oder aus einer unklaren Situation ein Ziel formuliert werden (Ist-Zustand-orientiertes Vorgehen).
2. **Analyse und Synthese:** Diese Phase beschreibt die Suche nach Lösungen für ein gegebenes Problem, um das definierte Ziel zu erreichen. In der Praxis wird dies als ein Wechselspiel zwischen Analyse und Synthese dargestellt.
3. **Analyse und Bewertung:** Die Analyse und Bewertung der konkretisierten Lösungsvarianten findet in der dritten Phase statt. Eine detaillierte Evaluation, zum Beispiel durch Berechnung, Simulation oder Versuche, gibt Aufschluss über die Realisierbarkeit der Lösungsvorschläge.
4. **Entscheidung:** Die letzte Phase soll Aufschluss darüber geben, ob die erarbeiteten und bewerteten Lösungsvorschläge zum gewünschten Ergebnis führen. Falls dies nicht der Fall sein sollte, muss mit der ersten Phase erneut begonnen werden.

4.1.2 Das V-Modell als Makrozyklus

Die Basis des dabei beschriebenen Vorgehens für die Entwicklung mechatronischer Systeme ist das aus der Softwareentwicklung bekannte V-Modell, welches jedoch an die Anforderungen der Mechatronik angepasst wurde [156, S. 26], [157, S. 58], [158, S. 168]. Dabei wird, wie in Abbildung 4.1 dargestellt, zwischen den drei Phasen Systementwurf, domänenspezifischer Entwurf und Systemintegration unterschieden [159, S. 271–275].

1. Die Ausgangsposition des zu entwickelnden Systems bildet ein konkreter Anforderungskatalog. Dieser muss die gegebene Aufgabenstellung zufriedenstellend beschreiben. Die Anforderungen sind in Abschnitt 4.2 aufgelistet.
2. In der zweiten Phase, der Systementwurfsphase, wird ein lösungsorientiertes übergreifendes Konzept entwickelt. Dafür werden die Gesamtfunktionen in wesentliche Teilfunktionen unterteilt. Bei mechatronischen Systemen sind diese Teilfunktionen beispielsweise das Messen, Steuern und Regeln oder Übertragen. Die Gesamtfunktion vereint anschließend alle Teilfunktionen und stellt somit die Funktionsweise des gesamten Systems dar.

3. Ziel des domänenspezifischen Entwurfs ist, das entwickelte Konzepte zu konkretisieren und in den jeweiligen Domänen auszuarbeiten. Wie in Abbildung 4.1 zu erkennen, sind die drei Domänen eines mechatronischen Systems der Maschinenbau, die Elektro- und Informationstechnik. Der Entwurf des Klassifikationsalgorithmus ist in Kapitel 5 definiert und das Konzept der IT-Architektur ist in Kapitel 6 beschrieben.
4. In der letzten Phase, der Systemintegration, werden die erarbeiteten Lösungen der einzelnen Domänen in einem Grundkonzept vereinigt. Die technische Umsetzung ist in Kapitel 7 definiert.

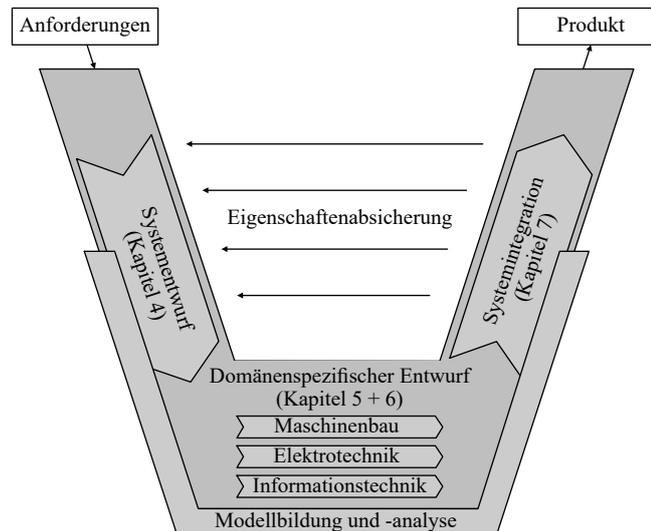


Abbildung 4.1: V-Modell nach VDI 2206 [156, S. 29].

Wie in Abbildung 4.1 zu erkennen, wird der Systementwurf in Kapitel 4 definiert. Der domänenspezifische Entwurf der beiden Teilsysteme wird in den Kapiteln 5 und 6 beschrieben und die finale Systemintegration wird in Kapitel 7 erläutert. Diesen Phasen geht die Definition der Anforderungen voraus. Sie ist die Grundlage für die Entwicklung des mechatronischen Systems, welches in Abbildung 4.3 dargestellt ist. Diese Anforderungen werden im folgenden Abschnitt definiert.

4.2 Anforderungen an die prädiktive und präskriptive Analyse

Wie im vorherigen Abschnitt beschrieben, stellen die Anforderungen den Ausgangspunkt der Entwicklung des Systems dar. Diese wurden sowohl abgeleitet von der VDI-Richtlinie 2870, welche die acht Gestaltungsprinzipien eines ganzheitlichen Produktionssystemes beschreibt [160], als auch von zwölf teilstrukturierten Interviews mit Experten im Bereich Datenanalyse, Steuerungstechnik, IoT und Produktionsplanung. Die Dokumentation der Experteninterviews ist im Anhang A zu finden. Diese Anforderungen sind, wie in Abbildung 4.2 dargestellt, in die folgenden drei Bereiche unterteilt:

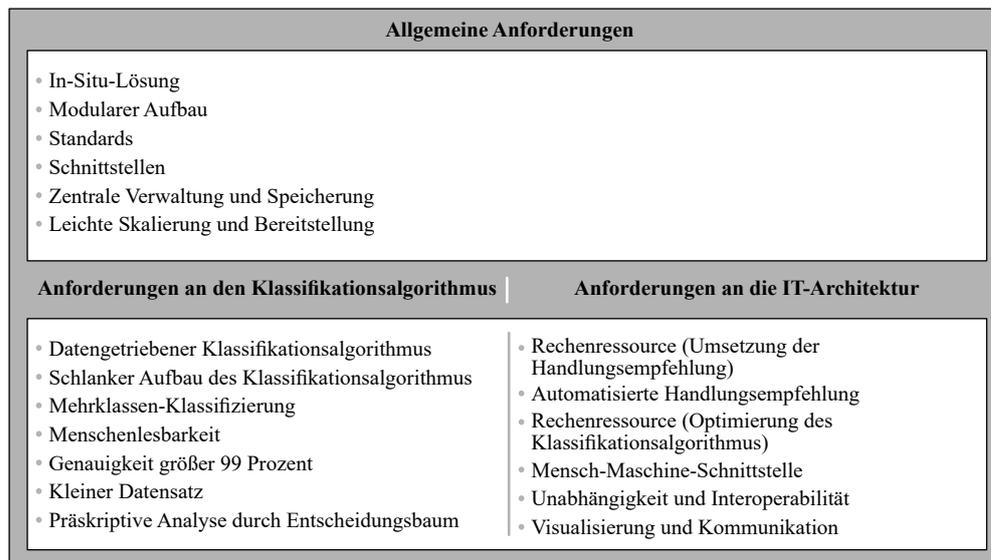


Abbildung 4.2: Anforderungen an die präskriptive Fertigungsoptimierung.

4.2.1 Allgemeine Anforderungen

Wie in Abbildung 4.2 zu erkennen, umfassen die allgemeinen Anforderungen die folgenden Punkte:

- Das in dieser Arbeit entwickelte Prüfsystem soll im Einklang mit dem Null-Fehler-Prinzip, was eine fehlerfreie Produktion voraussetzt, eine 100-Prozent-Prüfung umfassen. Da jedoch die Ursachen eines Fehlers vielfältig und oft unvorhersehbar

4.2 Anforderungen an die prädiktive und präskriptive Analyse

sind, lässt sich dieses Idealziel nicht erreichen [160]. Folglich ist es notwendig, Daten von jedem produzierten Bauteil zu erfassen und aus den auftretenden Fehlern zu lernen. Da es sich um eine Serienproduktion handelt, unterliegt das Prüfsystem definierten Taktzeitanforderungen. Bei dem in dieser Arbeit betrachteten Anwendungsfall beträgt die Taktzeit 30 Sekunden. In Kombination mit der benötigten 100-Prozent-Prüfung impliziert dies eine **In-Situ Lösung**. Das Prüfsystem muss also in die Linie und somit in den Produktionsablauf integriert werden [161]. Dabei sind die Datenaufnahme, -verarbeitung und -umsetzung inkludiert.

- Eine Standardisierung ist ein essenzieller Bestandteil eines ganzheitlichen Produktionssystems [162]. Aus diesem Grund sind für einen breiten Einsatz des Systems ein **modularer Aufbau** und die Verwendung von **Industriestandards** anzustreben. Dadurch entstehen transparente Abläufe und Schnittstellen, welche einen breiten Einsatzbereich unterstützen [160, S. 13].
- Ein Teilgebiet der Qualitätssicherung ist die Prüfdatenauswertung. Durch eine Auswertung der Prozessdaten können Aussagen bezüglich der Prozessfähigkeit gewonnen werden. Die in der vorliegenden Arbeit genutzte Prüfstrategie ist, wie in Abschnitt 4.2.2 erläutert, eine attributive Klassifizierung. Dabei werden Produkte anhand von Merkmalen einer Klasse zugewiesen. Oft wird zwischen fehlerhaften und fehlerfreien Produkten unterschieden [163], [164, S. 85]. Aufgrund dessen muss das System für den beschriebenen Prüfprozess eine **Schnittstelle** zur Sensorik besitzen. Das System soll somit in der Lage sein, Daten von einem Sensor zu erfassen und mithilfe des Algorithmus zu verarbeiten und auszuwerten.
- Für die Nachverfolgung und Evaluierung der Messergebnisse und erhobenen Daten wird ein **zentraler Speicherort** benötigt. Weil die Daten durch die oben geforderte Echtzeitfähigkeit (Velocity) als auch Größe (Volume) als Big Data angesehen werden, ist eine Cloud-Plattform notwendig [165, S. 174].
- Die Vermeidung von Verschwendung gilt nach VDI 2870 als Fundament der acht Gestaltungsprinzipien eines ganzheitlichen Produktionssystems [160, S. 15], [166]. Infolgedessen sollte das Konzept leicht an den erforderlichen Grad **skalierbar** sein. Dafür ist das System nach dem in Abschnitt 2.5.2 beschriebenen Ansatz der

4. Präskriptive Prozessoptimierung Methode

IoT zu konzipieren. Deswegen erfordert das System eine zentrale Steuereinheit, durch welche die einzelnen anlagennahen Geräte verwaltet werden können.

4.2.2 Anforderungen an den Klassifikationsalgorithmus

In der Produktion ist es essenziell, Fehler frühzeitig zu erkennen, damit diese in der Produktionslinie behoben werden können. Für die Erkennung und Klassifizierung der Qualitätsabweichungen soll der Algorithmus zwischen prädiktiver und präskriptiver Analytik aufgeteilt werden. Durch einen modularen Aufbau des Algorithmus kann die resultierende Handlungsempfehlung besser verstanden und angepasst werden. Im Folgenden sind die Anforderungen an den Algorithmus aufgelistet:

- Bei dem zu entwickelnden System soll eine 100-Prozent-Prüfung zur Detektion von Qualitätsabweichungen erfolgen. Dies erfordert eine Prüfung, welche den Takt des Produktionsprozesses nicht negativ beeinflusst. Inkludiert sind dabei sowohl die Datenaufnahme als auch die Datenverarbeitung. Aufgrund dessen muss die Prädiktionszeit des Klassifikationsalgorithmus gering sein. Dies bedeutet, dass der verwendete Klassifikationsalgorithmus eine **schlanke Architektur** besitzen muss. Für die Erfassung und Verarbeitung der Daten ist für den in dieser Arbeit verwendeten Anwendungsfall eine maximale Bearbeitungszeit von 30 Sekunden zulässig. Die hier zu entwickelnde Lösung sollte daher eine Verarbeitungszeit eines Stators von 30 Sekunden nicht überschreiten.
- Aufgrund des geringen Expertenwissens in der Produktion elektrischer Antriebssysteme soll ein Klassifikationsalgorithmus aus dem Bereich des überwachten Lernens verwendet werden. Der Algorithmus soll dabei Zusammenhänge und Merkmale selbstständig erlernen können und nicht durch explizites Expertenwissen programmiert werden. Aufgrund dessen soll ein **datengetriebener Klassifikationsalgorithmus** verwendet werden.
- Bei einem Produktionsprozess tritt meist eine Vielzahl unterschiedlicher Klassen von Qualitätsabweichungen auf. Daher muss der Klassifikationsalgorithmus die Fähigkeit besitzen, zwischen mehr als zwei Klassen unterscheiden zu können. Somit muss er eine **Mehrklassen-Klassifizierung** aufweisen.

4.2 Anforderungen an die prädiktive und präskriptive Analyse

- Wie in Abschnitt 2.4.1 erläutert, ist die Klassifikation ein Teilgebiet des überwachten Lernens. Das Labeling und somit die Zuordnung einer Qualitätsabweichung zu der entsprechenden Klasse muss durch Expertenwissen erfolgen. Für eine Plausibilitätskontrolle ist es nötig, eine **Menschenlesbarkeit** des Modells zu gewährleisten.
- Die Komplexität sowie die Quantität der Prüfdaten in einer Serienproduktion setzen eine hohe Robustheit des Klassifikationsalgorithmus voraus. Vor allem ist es essenziell, dass keine fehlerhaften Bauteile übersehen werden. Deswegen wird eine **Genauigkeit** des Algorithmus größer als 99% gefordert.
- Die Güte des Klassifikationsmodells hängt von der Qualität, Quantität und Ausgeglichenheit des Datensatzes ab. Im produzierenden Umfeld ist es jedoch schwer, einen großen und ausgeglichenen Datensatz zu erstellen, da weniger fehlerhafte Bauteile als fehlerfreie entstehen. Aufgrund dessen muss ein **kleiner Datensatz** ausreichen.
- Zur Generierung einer Handlungsempfehlung in Abhängigkeit der Fehlerklasse wird eine präskriptive Analyse gewählt. Durch einen **Entscheidungsbaum** wird eine solche Handlungsempfehlung modelliert. Dabei soll die Struktur des Entscheidungsbaumes durch Expertenwissen und Versuche abgeleitet werden.

4.2.3 Anforderungen an die IT-Architektur

Für die in Abschnitt 4.2.2 gelisteten Anforderungen an den Klassifikationsalgorithmus, wie beispielsweise NN, werden leistungsstarke Rechenressourcen benötigt. Aufgrund dessen sind im Folgenden die Anforderungen an die IT-Architektur aufgelistet:

- Aufgrund des hohen Produktionstaktes stellt die Signalverarbeitung eine hohe Anforderung dar. Wichtig sind das Erkennen einer Qualitätsabweichung und die proaktive Bereitstellung einer Handlungsempfehlung, wie beispielsweise eine Nacharbeitsstrategie, um auf aktuelle Bedingungen und Prozessabweichungen reagieren zu können. Dies muss unter Echtzeitanforderungen gewährleistet sein, um Zykluszeitverluste zu vermeiden. Dabei darf die Taktzeit nicht überschritten werden. Deshalb wird eine geeignete **Rechenressource** benötigt.

4. Präskriptive Prozessoptimierung Methode

- Das ausgewertete Ergebnis durch den Algorithmus soll in eine Umsetzungsstrategie umgewandelt werden. In Abhängigkeit der bestimmten Klasse soll eine **automatisierte Handlungsempfehlung** getätigt werden. Diese soll über Aktoren in der Produktionslinie umgesetzt werden.
- Die Produktionsbedingungen und deren Umgebung ändern sich in der Regel im Laufe der Zeit. Dies hat, wie in Abbildung 4.3 visualisiert, Auswirkungen auf die Sensorik. Ein Beispiel dafür ist, dass sich im Laufe der Jahreszeiten die Lichtverhältnisse in der Produktion ändern können, was sich negativ auf die optische Qualitätsüberwachung auswirken kann. Wie in der vorherigen Anforderung beschrieben, sollte ein Klassifikationsalgorithmus aus dem Bereich des überwachten Lernens verwendet werden. Das bedeutet, dass der Algorithmus mit aktualisierten Daten neu antrainiert und an die veränderten Bedingungen angepasst werden muss. Dazu ist eine geeignete **Rechenressource** erforderlich.
- Bei maschinellen Lernalgorithmen können bei der Klassifikation Fehler auftreten. So kann z. B. ein fehlerhaftes Bauteil als gutes Bauteil klassifiziert werden. Wie oben beschrieben, werden die Daten und ihre Labels verwendet, um den Algorithmus bei sich ändernden Bedingungen neu anzutrainieren. Es ist somit essenziell, dass die Labels korrekt sind. Daher muss, wie in Abbildung 4.3 dargestellt, eine **Mensch-Maschine-Schnittstelle** vorhanden sein, um Labels anpassen zu können. Dadurch können Fehler durch den Klassifikationsalgorithmus im Nachhinein korrigiert und somit eine höchstmögliche Qualität der Daten erreicht werden.
- Die **Unabhängigkeit und Interoperabilität** der Informationsverarbeitung ist wichtig. Dadurch wird sichergestellt, dass das Klassifikationsmodell zur Erkennung von Qualitätsabweichungen und zur Generierung von Handlungsempfehlungen nicht von einer bestimmten Hardware abhängig ist.
- Die **Visualisierung und Kommunikation** der implementierten Optimierungen ist wichtig für die Benachrichtigung des Personals und kann elementar für die Akzeptanz eines Systems beim Personal sein. Dadurch können wichtige Kennzahlen visualisiert werden.

4.3 Mechatronisches System als Grundmodell

Wie in den Anforderungen zu erkennen, wird ein System suggeriert, das aus mechanischen, elektronischen und informationstechnischen Komponenten besteht. Aus diesem Grund ist ein Ausgangsmodell erforderlich, welches diese drei Domänen kombiniert. Die in der VDI 2206 vorgeschlagene Struktur mechatronischer Systeme deckt diese Anforderungen ab [156]. Ein solches System ist in Abbildung 4.3 dargestellt.

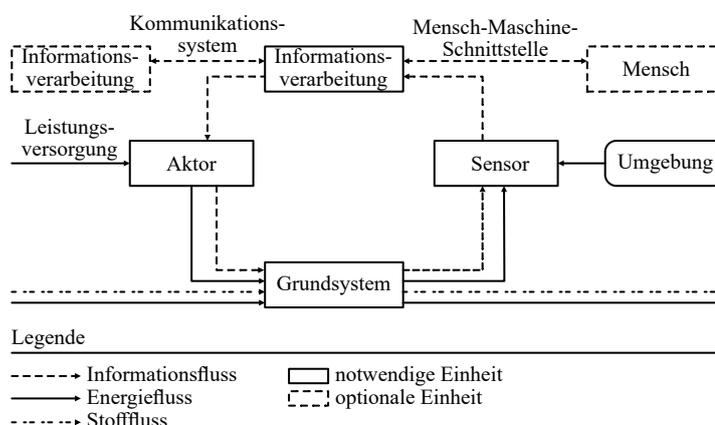


Abbildung 4.3: Grundstruktur eines mechatronischen Systems [156, S. 14].

Wie zu erkennen, besteht ein mechatronisches System aus vier Elementen, welche den systeminternen Regelkreis bilden [167, S. 27]:

- Das **Grundsystem** besitzt eine mechanische, elektromechanische, hydraulische oder pneumatische Struktur, welche auch miteinander kombiniert werden können. In der vorliegenden Arbeit ist dies eine Produktionsanlage. Das Grundsystem bildet das Kernelement der physischen Ebene.
- Durch den **Sensor** können Messgrößen des Grundsystems erfasst werden. Wie zu erkennen, können Sensoren durch die Umgebung beeinflusst werden. Die gemessenen Werte werden an das Informationssystem übermittelt. Der in dieser Arbeit verwendete Sensor zur Prüfung der Qualität der produzierten Bauteile soll ein optisches Messprinzip verwenden, was ebenfalls als Sichtprüfung bezeichnet wird [168, S. 5]. Nach DIN EN 1330, Teil 10 kann dabei wiederum zwischen einer direkten Sichtprüfung mit einem nicht unterbrochenen Strahlengang und einer indirekten Sichtprüfung mit einem unterbrochenen Strahlengang unterschieden

4. Präskriptive Prozessoptimierung Methode

werden. Zur indirekten Sichtprüfung zählt die Anwendung von Foto- und Videotechnik [169]. Vorteile eines optischen Prüfverfahrens sind die Berührungsfreiheit, die Geschwindigkeit, die Integrierbarkeit und die geringen Investitionskosten [170, S. 425].

- Die **Informationsverarbeitung** ist der zentrale Baustein der logischen Ebene. Mithilfe der durch die Sensoren erhobenen Messwerte ermittelt die Informationsverarbeitung die notwendige Einwirkung, um die Zustandsgrößen des Grundsystems zu beeinflussen. Die Informationsverarbeitung kann ebenfalls als Mensch-Maschine-Schnittstelle durch einen Anwender beeinflusst werden.
- Die **Aktorik** ist zuständig, die vom Informationssystem ermittelte und notwendige Einwirkung auszuführen.

Diese Elemente sind durch Stoff-, Energie- und Informationsflüsse miteinander verbunden. Das Hauptaugenmerk dieser Arbeit liegt auf der Einheit Informationsverarbeitung (Klassifikationsalgorithmus und verarbeitende IT-Architektur) und der Sensorik, welche durch das Umfeld beeinflusst werden kann.

Auf Basis dieses Vorgehens wird in Kapitel 5 das zur optischen Überprüfung notwendige CNN modelliert. In Kapitel 6 wird anschließend eine IT-Architektur konzeptioniert, sodass dieses CNN in der Produktion anwendbar ist.

Kapitel 5

Modellierung eines CNN zur Erkennung von Prozessabweichungen

5.1 Methodik

Wie in Kapitel 4 beschrieben, liegt der Schwerpunkt dieser Arbeit auf der Einheit der **Informationsverarbeitung** des mechatronischen Systems. Wie erläutert, ist der zweite Schritt des V-Modells, welches nach der VDI-Richtlinie 2206 als Leitfaden für die Entwicklung eines mechatronischen Systems genannt wird, der domänenspezifische Entwurf. Aus diesem Grund wird in diesem Kapitel auf den ersten Teil, die Erstellung des Klassifikationsalgorithmus, eingegangen. Da es sich um einen datengetriebenen Algorithmus handelt, eignet sich für die Entwicklung des Klassifikationsalgorithmus das in Abbildung 5.1 dargestellte Vorgehen, welches auf dem Cross-Industry Standard Process for Data Mining (CRISP-DM) und Knowledge Discovery Database (KDD) Modell basiert [171, 172, 173, 174, 175]. Neben der Klassifikation ist eine automatisierte Generierung einer Handlungsempfehlung in der vorliegenden Arbeit von wichtiger Bedeutung. Aus diesem Grund ist das CRISP-DM Vorgehen um den Punkt „Modellierung präskriptive Analyse“ erweitert. Des Weiteren wurden die grau hinterlegten Schritte im Bezug auf die optische Erkennung von Qualitätsabweichungen mithilfe von Data Analytics in der Automobilproduktion angepasst. Die Bestandteile des Ablaufmodells werden im Anschluss vorgestellt.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

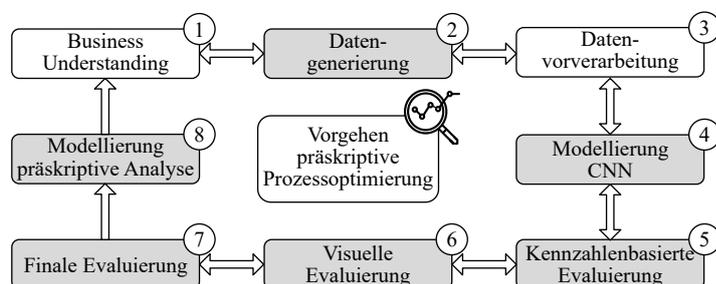


Abbildung 5.1: Vorgehensweise zur automatisierten Fehlererkennung und Nacharbeit mithilfe von präskriptiver Analyse.

5.1.1 Business Understanding

Im ersten Schritt des Ablaufmodells geht es um das Verstehen des Anwendungsfalls. Hierfür werden die in Abschnitt 4.2.2 erläuterten Anforderungen und Ziele detailliert analysiert. Ein weiterer wichtiger Bestandteil ist die Planung und das Vorgehen der Datengenerierung und Definition der relevanten Fehlerklassen mithilfe von Experten. Zusätzlich wird in diesem Schritt die Umsetzung der Handlungsempfehlung mittels präskriptiver Analytik definiert.

5.1.2 Datengenerierung

Im zweiten Schritt werden Daten generiert, welche für das Training des Klassifikationsalgorithmus relevant sind. Diese werden, wie im Abschnitt 2.4.1 vorgestellt, für die Verwendung eines Algorithmus aus dem Bereich überwachtes Lernen benötigt. Die Qualität und Quantität hat dabei starken Einfluss auf den Trainingsprozess und somit auf die Güte des Klassifikationsalgorithmus. Nach der VDI/VDE/VDMA-Richtlinie 2632 Blatt 3 ist aufgrund dessen die Erstellung von Referenzteilen essenziell. Dabei kann es sich sowohl um reale Bauteile mit entsprechender Bewertung durch Referenzsysteme oder Experten als auch um Prüfobjekte mit künstlich hergestellten Eigenschaften handeln [176, S. 14]. Aufgrund dessen ist für die Erstellung der Referenzteile für den Trainingsprozess die Einbeziehung von Expertenwissen in den Prozess des Labelings von großer Bedeutung. Bei der Verwendung von Bildern kann die Anzahl der Referenzteile in der Regel durch zusätzliche Bilder erweitert werden, indem die Originalbilder nachträglich bearbeitet werden. Auf diese Weise können z. B. der Kontrast, die Schärfe oder die Position bzw. Perspektive der Kamera in Relation zum Prüfobjekt negativ beeinflusst

werden [177, S. 2–3]. Dies ermöglicht den Einsatz von DA. Darüber hinaus müssen in diesem Schritt nach der VDI/VDE/VDMA-Richtlinie 2632 Blatt 3 durch einen Experten Merkmale definiert werden, welche für die Zuordnung zur entsprechenden Klasse verantwortlich sind [176, S. 14], [177, S. 2–3].

Die Erstellung eines umfangreichen Datensatzes ist jedoch in der Produktion mit hohem Aufwand verbunden, da Produktionsanlagen voll ausgelastet sind und daher deren Verfügbarkeit für Experimente sehr begrenzt ist. Zudem ist die Quote fehlerhafter Komponenten sehr niedrig. Aus diesem Grund kann lediglich ein kleiner, ausgewogener Datensatz generiert werden. Dadurch unterscheidet sich ein in der Produktion erstellter Datensatz hinsichtlich der verfügbaren Daten von klassischen Datensätzen wie ImageNet¹. Dies führt zu der Notwendigkeit, Daten synthetisch mithilfe von DA zu vermehren. Dabei ist jedoch zu beachten, dass die synthetisch vervielfältigten Daten der Realität entsprechen. Aufgrund dessen werden die folgenden drei Methoden der Datenvervielfältigung verwendet [180, S. 2–3], [181, S. 1–2]:

1. Durch das horizontale bzw. vertikale Spiegeln eines Bildes können aus einem Bild drei weitere erzeugt werden. Beispielsweise verbessert dies die Detektion und Klassifikation von Objekten, die verkehrt auf einem Bild abgebildet sind.
2. Durch das Anwenden einer Drehmatrix lassen sich Bilder um einen definierten Winkel drehen. Dadurch kann beispielsweise eine Verdrehung der Kamera oder des Prüfobjekts im Produktionsprozess simuliert werden.
3. Zuletzt kann durch die pixelweise Verschiebung des gesamten Bildes der Datensatz synthetisch vergrößert werden. Dadurch kann sichergestellt werden, dass Merkmale eines Prüfobjektes unabhängig von dessen Position detektiert werden können.

Durch die Kombination dieser drei Verfahren kann der Datensatz deutlich vergrößert werden. Zu beachten ist jedoch, dass ausschließlich Daten des Trainingsdatensatzes synthetisch vervielfältigt werden. Synthetische Daten im Test- oder Validierungsdatensatz zu verwenden, würde das Ergebnis des Klassifikationsalgorithmus verfälschen.

¹ImageNet ist eine Bilddatenbank mit insgesamt 14 Millionen Bildern und 22.000 Bildkategorien. Diese Datenbank bietet einen öffentlichen und freien Zugang für Forschung und Lehre. Sie hilft bei der Forschung von Algorithmen im Bereich von Computer Vision und unterstützt dabei, Deep Learning-Algorithmen zu verbessern [178, S. 248], [179, S. 1].

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

5.1.3 Datenvorverarbeitung

Die im vorherigen Schritt generierten Daten müssen im Anschluss vorverarbeitet werden. Die Vorverarbeitung hat starken Einfluss auf den Trainingsverlauf und bestimmt, wie schnell der Trainingsprozess konvergiert [62, S. 22-5]. Die Grundidee der in dieser Arbeit verwendeten Vorverarbeitungskette wird aus der Datenvorverarbeitung der bekannten Datensätze Modified National Institute of Standards and Technology database (MNIST) und Fashion-MNIST abgeleitet. MNIST ist eine Datenbank mit handgeschriebenen Ziffern und Fashion-MNIST ist eine Datenbank von Zalandos Artikelbildern, die aus einem Trainingssatz mit 60.000 Beispielen und einem Testsatz mit 10.000 Beispielen zusammengesetzt ist [182, S. 2], [183, S. 10]. Dabei werden die Daten wie folgt vorverarbeitet:

1. Im ersten Schritt wird das Objekt aus dem Bild eng ausgeschnitten.
2. Anschließend werden die Seiten des ausgeschnittenen Bildes auf eine bestimmte Länge mithilfe von Subsampling angepasst.
3. Im dritten Schritt wird das Rot, Grün und Blau (RGB)-Bild in ein Graustufenbild umgewandelt.
4. Ein weiterer wichtiger Vorverarbeitungsschritt ist die Normalisierung der Daten. Hierfür werden in dieser Arbeit drei Techniken miteinander verglichen [52, S. 44–45], [62, S. 22 5–6]:
 - (a) Für einen besseren Vergleich wird zu Beginn der Trainingsprozess ohne Normalisierungstechnik betrachtet.
 - (b) Bei der zweiten Technik wird der Wertebereich der Pixel zwischen -1 und 1 aller Pixel des Trainingsdatensatzes normalisiert.
 - (c) Bei der dritten Normalisierungstechnik werden die Pixel der Bilder des Trainingsdatensatzes auf einen Mittelwert (Mean) und einer Standardabweichung (SD) normiert. Gleichung 5.1 beschreibt den mathematischen Zusammenhang.

$$\mathbf{X}^n = (\mathbf{X} - \mathbf{X}^M) ./ \mathbf{X}^{SD} \quad (5.1)$$

\mathbf{X}^n steht dabei für die normalisierten Bilder und \mathbf{X} für die Bilder aus dem Trainingsdatensatz. Der Operator $./$ beschreibt die elementweise Division der Matrixeinträge und \mathbf{X}^M und \mathbf{X}^{SD} stehen für den Mittelwert beziehungsweise die Standardabweichung der jeweiligen Pixelwerte im Bezug auf die Pixel aller Bilder.

5.1.4 Modellierung des Convolutional Neural Network

Aufgrund der in Kapitel 4 beschriebenen Vorteile wird in der vorliegenden Arbeit ein optischer Sensor mit einer indirekten Sichtprüfung verwendet. Dabei haben sich vor allem CNN etabliert, Merkmale in Bildern zu erkennen und diese zu klassifizieren. Deren Vorteile im Vergleich zu gewöhnlichen NN sind, dass CNN relevante Merkmale aus einem Bild erfassen und erlernen können. Darüber hinaus wird die Anzahl an trainierbaren Parametern durch Parameter-Sharing deutlich reduziert. In Kontrast zu NN können dadurch Rechenressourcen und Speicherplatz gespart werden [184, S. 1904], [185, S. 175], [186, S. 1717–1718], [187, S. 1285]. Aus diesem Grund wird im vierten Schritt (vgl. Abbildung 5.1) ausschließlich auf die Modellierung eines CNN eingegangen.

Bei der Auslegung des CNN ist auf diverse Anforderungen zu achten, welche in der Produktion oft konträr zueinander stehen. Diese Anforderungen besitzen eine Analogie zum magischen Dreieck des Projektmanagements, welches auch häufig in der Produktion eine Anwendung findet. In Abbildung 5.2 ist das magische Dreieck visualisiert.

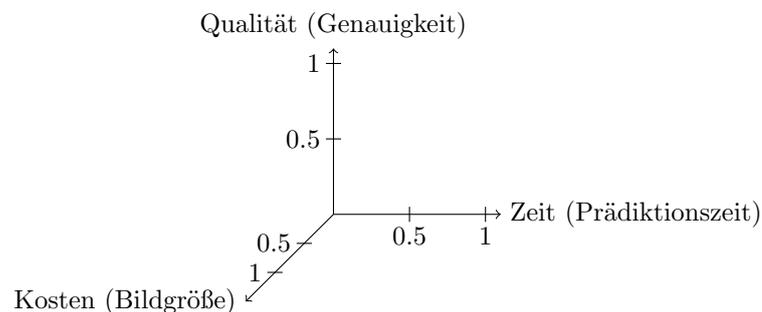


Abbildung 5.2: Magisches Dreieck nach Kuhlang et al. [188].

Es veranschaulicht die wechselseitigen Zusammenhänge der Achsen Qualität, Zeit und Kosten. Die Optimierung einer Dimension kann eine oder zwei andere Dimensionen

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

negativ beeinflussen. Dies bedeutet, dass alle drei Zielgrößen miteinander in Relation stehen. Somit müssen bei der Änderung einer Dimension Auswirkungen auf die anderen Dimensionen berücksichtigt werden [189, S. 11]. Ziel ist es, dass das magische Dreieck möglichst in allen drei Dimensionen stark ausgeprägt ist [190, S. 31]. Ähnlich wie beim Konflikt der Zieldimensionen Qualität, Zeit und Kosten in der Produktion stehen sich die Größen Genauigkeit des CNN, Anzahl der Parameter des CNN und Speichergröße des verwendeten Bildes gegenüber. Die Genauigkeit des Netzes besitzt eine starke Analogie zur Qualität. Denn bei einer geringen Genauigkeit können fehlerhafte Bauteile nicht erkannt werden. Der Faktor Zeit geht mit der Größe des Netzwerkes einher, da die Anzahl der Parameter die Prädiktionszeit beeinflusst. Der Faktor Kosten besitzt eine Analogie mit der Größe der verwendeten Bilder. Denn je größer das verwendete Bild ist, desto größer sind die Kosten für eine Langzeitarchivierung (vgl. Abbildung 5.2). Diese Zielgrößen werden in dieser Arbeit bei der Auslegung des CNN miteinander verglichen. Dieser Vergleich kann bei der zukünftigen Auslegung eines CNN im Bereich der Produktion helfen.

5.1.5 Kennzahlenbasierte Evaluierung des Klassifikationsalgorithmus

Nachdem die Architektur des CNN entworfen wurde, muss diese evaluiert werden. Standardmäßig werden hierfür, wie in Abschnitt 2.4.6 erläutert, die Daten in Trainings-, Validierungs-, und Testdaten aufgeteilt. Für die klassische, kennzahlenbasierte Evaluierung der Architektur des CNN werden hierfür die im Abschnitt 2.4.6 definierten Metriken herangezogen.

5.1.6 Visuelle Evaluierung des Klassifikationsalgorithmus

Wie in Kapitel 4 beschrieben, ist für einen Plausibilitätscheck eine der Anforderungen an das CNN die Menschenlesbarkeit des Modells. Problematisch ist, dass ausschließlich bei einem Entscheidungsbaum eine Menschenlesbarkeit durch Wenn-Dann-Regeln möglich ist [10, S. 89]. Ein CNN hat den Nachteil, dass das während des Trainingsprozesses gelernte Wissen meist schwer nachzuvollziehen ist und es dadurch als eine Art Black-box erscheint [191, S. 77]. Vor allem in der Produktion ist es jedoch essenziell, dass die durch den Experten definierten Merkmale für den Entscheidungsprozess des CNN

ausschlaggebend sind. Um dennoch ein CNN und dessen Vorteile nutzen und gleichzeitig dessen Entscheidungsprozess verstehen zu können, werden die im Abschnitt 2.4.7 beschriebenen Ansätze herangezogen.

5.1.7 Finale Evaluierung des Modells

Wie in Abschnitt 2.4.6 beschrieben, wird das finale Modell anhand eines Testdatensatzes beurteilt, mit dessen Hilfe die Generalisierung des Netzes bewertet werden kann. Die Anzahl an Fehlklassifizierungen muss jedoch an einem ausreichend großen Testdatensatz evaluiert werden. Nach VDA Band 5 wird für eine attributive Prüfung eine Stichprobengröße von mindestens 30 Messdaten vorgegeben [78, S. 64–66], [81, S. 1–10]. Diese ist jedoch im Vergleich zum Trainings- und Validierungsdatensatz für die meisten Anwendungsfälle sehr gering. Aufgrund dessen wurde im Abschnitt 2.4.8 ein Verfahren zur Berechnung der geeigneten Stichprobengröße und maximalen Anzahl an Fehlklassifizierungen für eine attributive Kameraprüfung vorgestellt. Dieses Verfahren wird auf die Evaluierung eines Kamerasystems in Kombination mit einem NN übertragen.

5.1.8 Präskriptive Analyse

Die präskriptive Analyse nutzt die aus dem Produktionsprozess gewonnenen Daten, um daraus Handlungsvorschläge für einen nachfolgenden Prozessschritt abzuleiten [28]. Das in Kapitel 5.1.4 erläuterte CNN dient der Qualitätssicherung, die eine Handlungsempfehlung durch eine echtzeitnahe Anpassung und Optimierung des Prozesses anhand der Zustandsparameter ermöglicht. Ein Beispiel hierfür ist im Produktionsumfeld die Ableitung eines passenden Nacharbeitskonzeptes oder die Anpassung der Prozessschritte in Abhängigkeit der detektierten Qualitätsabweichung. Durch eine passende Wenn-Dann-Beziehung zwischen eingetretenem Ereignis und passender Handlungsempfehlung kann die präskriptive Analyse durch einen Entscheidungsbaum modelliert werden. Die Definition der Rahmenbedingungen und Anforderungen der präskriptiven Fertigungsanalyse für eine intelligente Steuerung der Fertigung wird in Anlehnung an Gröger im Folgenden vorgestellt [31, S. 157ff.].

Die Grundlage der präskriptiven Fertigungsprozessanalyse stellt die zu optimierende prozessbezogene Kennzahl dar. In der vorliegenden Arbeit ist dies die Qualität eines

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

Bauteils. Der zu optimierende Fertigungsprozess P ist in N einzelne Prozessschritte S_N aufgeteilt:

$$P = \{S_1, \dots, S_K\} \text{ mit } 1 \leq N \leq K \quad (5.2)$$

Die aufeinander aufbauenden Prozessschritte können sowohl parallel als auch sequenziell gestaltet werden. Der Übersichtlichkeit halber wird nachstehend von einem rein sequenziellen Prozessablauf ausgegangen. Wie in Abbildung 5.3 ersichtlich, gliedert sich die präskriptive Prozessoptimierung in die beiden folgenden Bereiche [31, S. 157ff.]:

1. Nach Abschluss jedes Prozessschrittes S_L wird die **Laufzeitprognostik** ausgeführt. Dabei wird mithilfe des CNN die Qualität des zu produzierenden Bauteils prognostiziert.
2. Wird aufgrund der Laufzeitprognose eine Qualitätsabweichung des Bauteils festgestellt, ist proaktiv und in Echtzeit eine **automatisierte Nacharbeit** zu generieren. Für diesen Fall wird für den folgenden Prozessschritt S_E mit $E = (L+1) \leq K$ bzw. für eine Inline-Nacharbeit zwischen dem derzeitigen Prozessschritt und dem folgenden Prozessschritt eine Handlungsempfehlung abgeleitet.

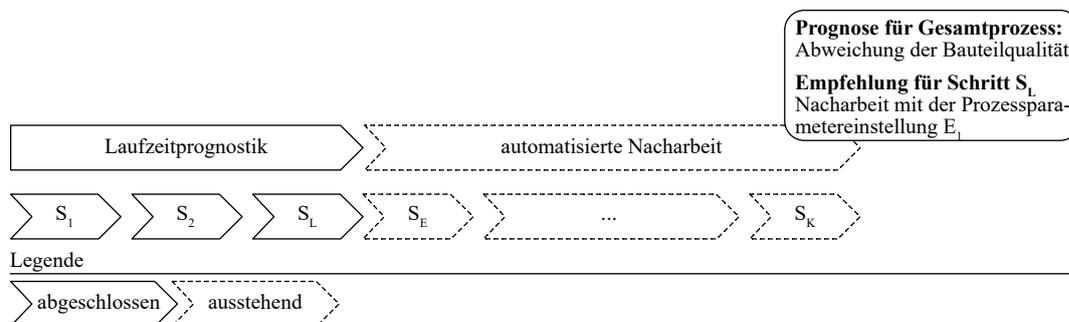


Abbildung 5.3: Komponenten der präskriptiven Fertigungsprozessanalyse eines laufenden Prozessauftrags in Anlehnung an Gröger [31, S. 159].

Im Folgenden wird dieses Problem an einem generischen Beispiel erläutert. Wie in Abbildung 5.4 zu erkennen ist, erfolgt eine Qualitätsabweichung bereits im Prozessschritt S_3 . Diese lässt sich jedoch aufgrund fehlender oder technisch nicht realisierbarer Prüfprozesse nicht erfassen. Folglich wird erst im letzten Prozessschritt S_K des Gesamtprozesses durch eine End-of-Line-Prüfung eine Qualitätsabweichung detektiert. Als

Folge muss das Bauteil zum Prozessschritt P_3 demontiert und visuell vom Personal begutachtet werden. Anschließend kann das Personal den Fehlerfall ermitteln und manuell nachbearbeiten. Dies beansprucht jedoch Zeit und somit finanzielle Mittel.

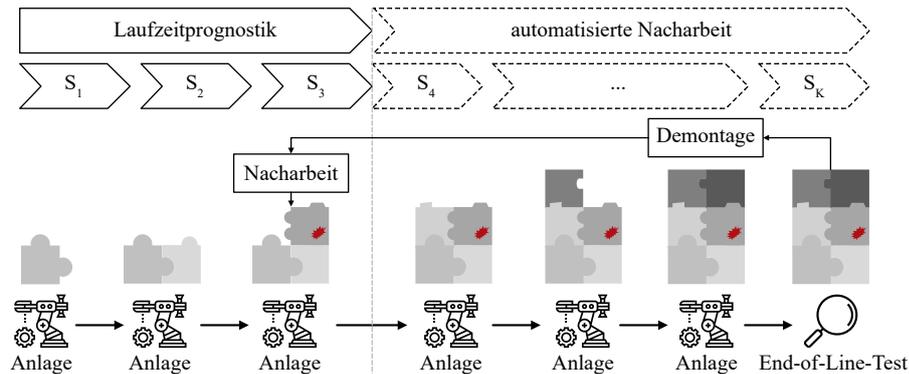


Abbildung 5.4: Darstellung eines generischen Ist-Fertigungsprozesses.

Um diesem Defizit entgegenzuwirken, ist wie oben beschrieben eine präskriptive Fertigungsprozessanalyse anzuwenden:

1. **Laufzeitprognostik:** Wie in Abbildung 5.5 dargestellt, findet im Prozessschritt S_3 eine Qualitätsabweichung statt. Diese Qualitätsabweichung kann jedoch mithilfe des vorgestellten CNN detektiert und somit eine unerwünschte Kennzahlenabweichung des Gesamtprozesses prognostiziert werden.

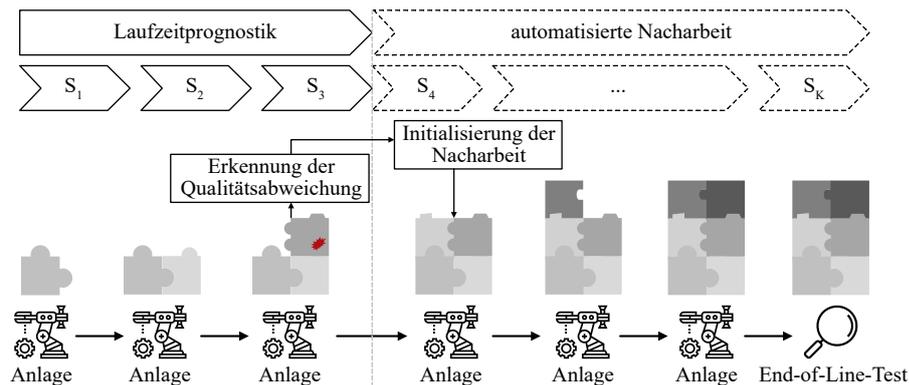


Abbildung 5.5: Darstellung eines Soll-Fertigungsprozesses durch eine Kombination aus künstlicher Intelligenz und präskriptiver Fertigungsprozessanalyse.

2. **Automatisierte Nacharbeit:** Auf der Grundlage der prognostizierten Kennzahlenabweichung wird automatisch und in Echtzeit eine Handlungsempfehlung

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

erarbeitet. Hierzu werden die für Prozessschritt S_4 zu verwendenden Prozessparameter empfohlen und in einem geschlossenen Regelkreis, z. B für eine Nacharbeit, automatisch an die Produktionsanlage übergeben.

5.2 Business Understanding

In diesem Abschnitt wird der eben dargelegte methodische Ansatz auf den Produktionsprozess des Hairpin-Verschweißens übertragen. Dadurch soll wie beschrieben eine frühzeitige Erkennung von Qualitätsabweichungen und eine daraus abgeleitete Handlungsempfehlung erfolgen.

Wie in Abschnitt 2.1 beschrieben, gehört die elektrische Maschine zu den Hauptkomponenten des elektrischen Antriebsstrangs. Im Vergleich zu normalen industriellen Elektromotoren stellt der Motor in einem Automobil besondere Anforderungen. Dazu gehören beispielsweise die Steigerung der Leistungsdichte und Effizienz bei gleichzeitiger Reduzierung von Gewicht, Kosten und Dauer der Produktion. Bezogen auf die Gesamtkosten des Motors trägt der Stator mit einem Anteil von 35% maßgeblich zu den Kosten in der Produktion des elektrischen Antriebs bei [8, S. 147]. Die Umsetzung dieser Anforderungen erfordert den Einsatz innovativer Technologien, wie beispielsweise die Hairpin-Technologie [19, S. 139]. Das Besondere dieser Technologie ist, dass die klassischen Kupferwicklungen des Stators durch gebogene Kupferstäbe, welche Haarklammern ähneln, ersetzt werden. Ein solches Hairpin ist beispielhaft in Abbildung 5.6 dargestellt. [192, S. 2471–2472], [193, S. 1]



Abbildung 5.6: Darstellung eines Hairpins nach Glässel [193, S. 2].

Die Vorzüge dieser Technologie liegen zum einen darin, dass gegenüber einem Elektromotor mit Drahtwicklungen eine vergleichsweise höhere Leistungsdichte erreicht wird. Zum anderen können die Kühleigenschaften des Motors dank der verbesserten Wärmeleiteigenschaften der Kupferstäbe gesteigert werden [194, S. 1], [195, S. 273–276]. Neben den verbesserten Produkteigenschaften kann auch der Herstellungs- und Montageprozess vereinfacht werden. Aufgrund des hohen Automatisierungspotenzials der nachfolgend beschriebenen Prozessschritte lässt sich der Stator effizient in großen Stückzahlen fertigen [192, S. 2471–2472], [193, S. 1–2]. Im Folgenden wird detailliert der aktuelle und gewünschte Produktionsprozess beschrieben.

5.2.1 Ist-Produktionsprozess des Stators

In Abbildung 5.7 sind die wichtigsten Prozessschritte zur Herstellung eines Stators mit Hairpins visualisiert. Wie zu erkennen, besteht der aktuelle Produktionsprozess aus sechs Schritten.

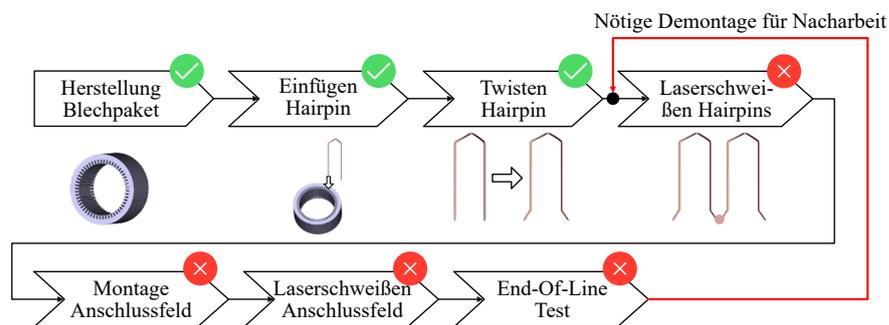


Abbildung 5.7: Ist-Prozess der aktuellen Statorfertigung.

Im ersten Schritt des Produktionsprozesses wird, wie in Abbildung 5.7 zu erkennen, das Blechpaket hergestellt. In einem parallelen Produktionsschritt wird ein rechteckiger und isolierter Kupferstab in der einer Haarklammer sich ähnelnden Form gebogen und anschließend werden die beiden Enden des gebogenen Kupferstabes abisoliert. Der Begriff Hairpin bezeichnet einen U-förmig vorgeformten Stromleiter, bestehend aus einem isolierten, rechteckigen Kupferstab. Diese Hairpins werden im zweiten Schritt in die im Kreis angeordneten Nuten des Blechpaketes vollautomatisiert eingefügt. Der dritte Prozessschritt ist das sogenannte Twisten. Bei diesem Schritt werden die freien Hairpin-Enden so zurechtgebogen, sodass sie sich paarweise berühren. Anschließend werden die

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

sich berührenden und abisolierten Hairpin-Paare sowohl elektrisch als auch mechanisch miteinander verbunden. Das dabei am häufigsten verwendete Verfahren ist das sogenannte automatisierte Laserschweißverfahren. Nachteil des Verfahrens ist jedoch, dass die automatisierte Verschweißung von Kupfer aufgrund des komplexen Prozesses zu unterschiedlichen Schweißfehlern führen kann. Die Ursache dieser Fehler und die Komplexität des Laserschweißverfahrens von Kupfer werden im Folgenden genauer erläutert. [192, S. 2471–2472], [193, S. 1–2]

Das Laserstrahlschweißen, oder Laserschweißen, ist ein weit verbreitetes Schweißverfahren in der Industrie und gehört zu der Gruppe des Schmelz-Verbindungsschweißens. Mittels elektromagnetischer Wellen wird ohne mechanischen oder elektrischen Kontakt Energie durch eine Linse auf den Fügepartner fokussiert. Die Fügepartner absorbieren diese Strahlung und erhitzen sich dabei. Bei ausreichender Intensität beginnen die Fügepartner im Fügespalt zu schmelzen und vermischen sich aufgrund der geringen Oberflächenspannung der niederviskosen Schmelzanteile. Aufgrund der hohen elektrischen und thermischen Leitfähigkeit spielt Kupfer für viele Anwendungen, wie zum Beispiel für erneuerbare Energien oder für die Elektromobilität, eine wichtige Rolle [194, S. 1], [195, S. 273–276]. Ein Vorteil ist, dass die elektrische und thermische Leitfähigkeit der beiden Materialien aus Kupfer nach dem Fügen an der Schweißnaht erhalten bleibt. Ein großer Nachteil ist jedoch, dass Kupfer bei Wellenlängen der Größenordnung 1000 nm die Laserstrahlung weniger als 5% absorbiert und somit starke reflektierende Eigenschaften aufweist. Aufgrund dessen erfordert das Laserstrahlschweißen eine sehr intensive Leistungsdichte [196, S. 494]. Dies führt jedoch zu Fehlerbildern, wie beispielsweise Schweißspritzer, oder unregelmäßigen Schweißnähten [197, S. 85–86], [198, S. 21]. Durch eine fehlerhafte Schweißnaht kann sich die elektrische Eigenschaft verschlechtern, wodurch höhere Temperaturen aufgrund des Stromflusses entstehen, was im schlechtesten Fall zum Ausfall des elektrischen Motors führen kann [199, S. 34], [200, S. 339].

Wie in Abbildung 5.7 im dritten Schritt dargestellt, gibt es derzeit kein automatisiertes Fehlerklassifizierungssystem. Daher durchläuft der Stator alle weiteren Verarbeitungsschritte bis zu seiner Kontrolle am Ende der Produktionslinie. Im Falle einer fehlerhaften Schweißnaht, hier als X bezeichnet, muss der Stator aus der Fertigungslinie ausgeschleust, das Schweißwerkzeug demontiert, der Fehler optisch befundet, der

Stator manuell ausgerichtet, das Schweißwerkzeug wieder montiert, der Stator wieder eingeschleust und die fehlerhaften Verschweißungen nachgeschweißt werden. Für jeden fehlerhaften Stator ist somit eine Nacharbeitszeit von 18 Minuten nötig. Dieser Prozess ist zeitaufwendig und in einer Serienproduktion damit teuer. Um einen effizienteren Prozess zu etablieren, müssen Schweißfehler direkt in der Station des Hairpin-Verschweißens **erkannt** und **klassifiziert** werden, um dadurch ein gezieltes und effizientes Nachschweißen in der Großserienfertigung zu ermöglichen.

5.2.2 Soll-Produktionsprozess des Stators

Ein solches Zielsystem ist in Abbildung 5.8 dargestellt. Wie beschrieben, können fehlerhafte Schweißungen im dritten Prozessschritt entstehen.

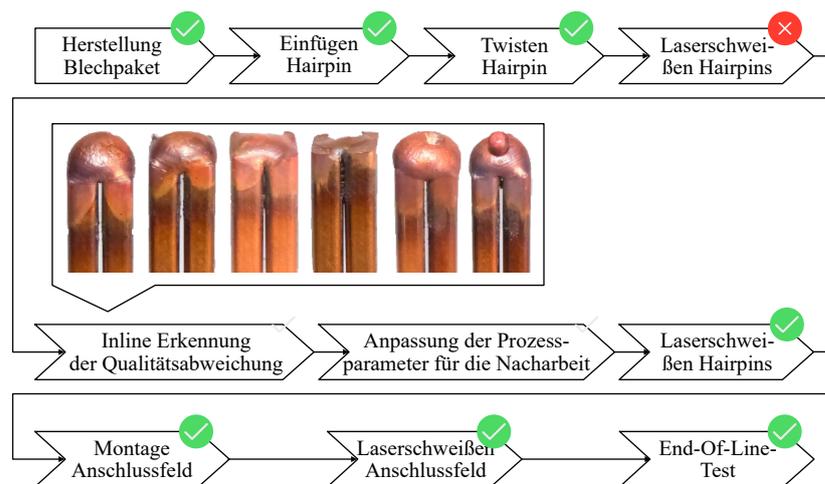


Abbildung 5.8: Soll-Prozess der aktuellen Statorfertigung.

Die in dieser Arbeit betrachteten Fehler, deren Merkmale und Entstehung wurden zusammen mit einem Experten für dieses Fachgebiet erarbeitet. Bei der Verschweißung der Hairpins in der Serienproduktion können die Fehlerfälle unzureichende Verschweißung (UV), Schweißspritzer (SS) und Schweißkrater (SK) auftreten. In Abbildung 5.9 sind diese Fehlerbilder inklusive der Klasse korrekte Verschweißung (KV) dargestellt. Außerdem wurde, wie in Abbildung 5.9 zu erkennen, die Schwere des Fehlers in zwei weitere Klassen unterteilt. Diese weitere Unterteilung entspricht dem jeweiligen Grenzmuster zwischen den jeweiligen Klassen.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

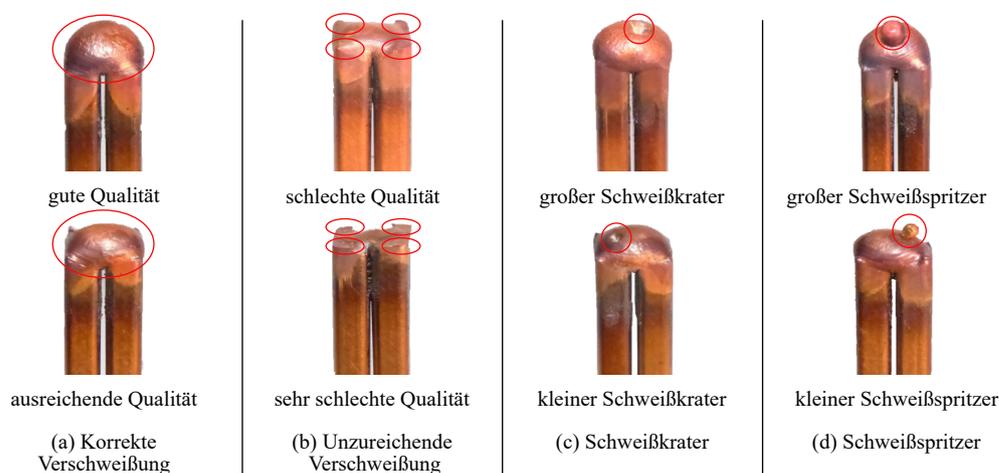


Abbildung 5.9: Mögliche resultierende Klassen bei der Verschweißung von Hairpins.

Wie in Abbildung 5.8 zu erkennen, durchläuft der Stator im Anschluss die restlichen Produktionsschritte und entspricht beim Test am Ende der Linie den geforderten Qualitätsanforderungen. Zusammengefasst und wie oben verdeutlicht, können aufgrund des instabilen Laserschweißprozesses Fehler in der Produktion des Stators entstehen. Aus diesem Grund ist es essenziell, Fehler frühzeitig zu erkennen, damit diese bereits innerhalb des Produktionsschrittes behoben werden können.

5.3 Datengenerierung

Wie bereits in Abschnitt 5.1.2 beschrieben, wirken sich die Quantität, Qualität sowie die Verteilung der Klassen des Datensatzes stark auf Resultate des Trainingsprozesses des Klassifikationsalgorithmus aus. Aufgrund dessen ist es essenziell, bei der Generierung der Daten strukturiert vorzugehen.

Nach der VDI/VDE/VDMA Richtlinie 2632 müssen für die Generierung der Daten und die anschließende Verifizierung des Klassifikationsalgorithmus Referenzteile definiert werden. Dabei sind die Definition sowie die Verifizierung dieser Referenzteile mithilfe von Expertenwissen durchzuführen. Denn dem korrekten Labeling der Referenzteile kommt eine maßgebliche Rolle bei der Bewertung der Eignung des Klassifikationsalgorithmus zu [176, S. 14], [177, S. 2–3]. Die Merkmalsausprägungen der Referenzteile der

verschiedenen Klassen, welche in Abbildung 5.9 rot markiert sind, werden im Folgenden beschrieben:

- Korrekte Verschweißung (KV): Diese Klasse hat das Merkmal einer gleichmäßig aufgeschmolzenen runden Kuppel.
- Unzureichende Verschweißung (UV): Diese Art des Fehlers tritt auf, wenn das Hairpin-Paar mit einer zu geringen Leistung miteinander verschweißt wird. Dies kann beispielsweise durch ein verschmutztes Schutzglas verursacht werden. Ein Merkmal ist die in Abbildung 5.9b deutlich erkennbare eckige Form der Verschweißung im Vergleich zu einer KV, welche in Abbildung 5.9a zu erkennen ist. Durch das geringe Aufschmelzen der Hairpin-Paare zeichnet sich eine UV des Weiteren durch eine Kronenbildung in den Ecken der Pin-Paare aus. Diese sind an den zurückgebliebenen Spitzen zu erkennen.
- Schweißkrater (SK): Während des Schweißprozesses können Staub- oder Partikelreste, welche sich auf den Hairpin-Paaren befinden, verbrennen, wodurch ein Krater an der Oberfläche der Schweißperle entsteht. Außerdem kann ein Krater durch den Einschluss von Prozessgas während der Verschweißung entstehen. Befindet sich diese Pore in der Nähe der Oberfläche, kann dies auch zu einem SK führen. Das optische Merkmal ist ein Krater inmitten der runden Schweißperle.
- Schweißspritzer (SS): Ein SS entsteht, wenn eine durch das Prozessgas oder Fremdpartikel an der Oberfläche entstandene Pore auf ein anderes Pin-Paar spritzt und dort haften bleibt. Erkennbar ist dies durch einen zusätzlich aufgeschmolzenen, haftenden „Kupfertropfen“ an der Oberfläche der Schweißkuppel eines Hairpin-Paares.

Zur Generierung qualitativ hochwertiger und realitätsnaher Referenzteile wurde ein spezieller Versuchsaufbau konstruiert und hergestellt. Da lediglich der obere Teil des Hairpins, also die Schweißkuppel, von Bedeutung ist, wird anstelle eines ganzen Hairpins lediglich ein oben abisoliertes Kupferstück der Länge von 100 *mm* verwendet. Geometrische Ausprägung, Material etc. sind identisch. Diese Kupferstücke werden in einen speziell dafür entworfenen Versuchsträger eingesetzt, wodurch die oberen Enden eines Hairpin-Paares miteinander verschweißt werden können. In Abbildung 5.10 ist dieser Datenträger dargestellt.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen



Abbildung 5.10: Versuchsträger zur Verschweißung von Hairpin-Paaren.

Mit diesem Datenträger lassen sich die Hairpin-Paare wie in einem echten Stator anordnen und in derselben Schweißanlage miteinander verschweißen. Diese Vorgehensweise wurde von einem Schweißexperten verifiziert. Außerdem wurden dieselben Parameter der Anlage wie bei der Verschweißung von Hairpin-Paaren eines echten Stators der Serienproduktion verwendet. Mithilfe der in Abbildung 5.11 symbolisierten 3D-Kamera XR-HT40M der Firma Keyence wurden die verschweißten Hairpin-Paare aus einer Vogelperspektive aufgenommen und digitalisiert. Diese Bilder sind zum einen 3D-Daten, welche die Form der Schweißkuppel abbilden, als auch gewöhnliche Schwarz-Weiß-Bilder. Der Aufnahmebereich der Kamera, welcher 16 mm beträgt, wurde so eingestellt, dass sich lediglich die Schweißkuppel im Aufnahmebereich befindet. Abbildung 5.11 visualisiert schematisch das Vorgehen.

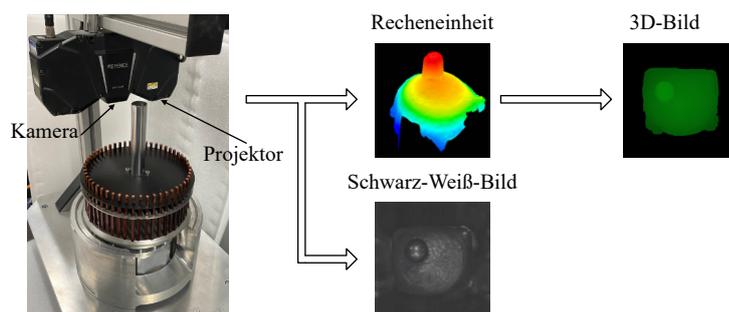


Abbildung 5.11: Aufnahme und Speicherung der 3D-Daten und Schwarz-Weiß-Bilder.

Wie beschrieben, wurden neben den Schwarz-Weiß-Bildern auch 3D-Daten erzeugt. Zur Erzeugung dieser 3D-Daten berechnet die Recheneinheit des Kamerasystems die Höheninformationen aus einem RGB-Bild, in welchem die Höheninformationen codiert

sind. Des Weiteren sind beispielhaft in Abbildung 5.11 das Schwarz-Weiß-Bild und das resultierende RGB-Bild mit den Höheninformationen einer Verschweißung der Klasse SS visualisiert. Durch dieses Vorgehen wurden zwischen 550 und 560 Schwarz-Weiß-Bilder als auch RGB-Bilder der Verschweißungen pro Klasse angefertigt. In Tabelle 5.1 ist die genaue Anzahl der Aufnahmen jeder Klasse aufgelistet.

Tabelle 5.1: Gesamtanzahl aufgenommener Verschweißungen pro Klasse.

Fehlerklasse	Anzahl Aufnahmen
Korrekte Verschweißung	604
Unzureichende Verschweißung	560
Schweißkrater	563
Schweißspritzer	557
Summe	2.284

Wie in Abschnitt 5.1.2 bereits erläutert, ist die Güte des Klassifikationsalgorithmus unter anderem von der Quantität der verwendeten Daten abhängig. Die Erstellung eines großen und ausgeglichenen Datensatzes ist jedoch in der Produktion mit hohem Aufwand verbunden. Um dennoch einen größeren Datensatz zu generieren, wurden die originalen Bilder mithilfe von DA vervielfältigt. Dadurch lassen sich durch Kombination von Rotation, Verschiebung als auch Spiegelung weitere synthetische Bilder erzeugen. Wichtig dabei ist, dass diese synthetisch erzeugten Bilder ausschließlich im Trainingsdatensatz verortet sind. Aufgrund dessen wurden die generierten originalen Daten aus Tabelle 5.1 mit einem Verhältnis von 80% Trainingsdaten zu 20% Validierungsdaten aufgeteilt und DA ausschließlich auf den Trainingsdatensatz angewandt. Bei der beispielhaften Verwendung eines Faktors von 49 konnte durch Spiegelung, Rotation und Verschiebung der Trainingsdatensatz von ca. 1.800 auf 90.000 Bilder erweitert werden. Die dabei resultierende Größe des Trainings- und Validierungsdatensatzes ist in Tabelle 5.2 aufgelistet.

5.4 Datenvorverarbeitung

Um eine möglichst hohe Güte des CNN zu erreichen, ist es notwendig, die verwendeten Daten vorzuvorbereiten. Neben der Vorverarbeitung ist es oft nötig, diese Daten in eine für den Eingang des CNN sinnvolle Struktur umzuwandeln. Wie in Abschnitt 5.1.3

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

Tabelle 5.2: Anzahl der originalen und resultierenden, synthetischen Trainingsdaten je Klasse durch Anwendung von DA.

Klasse	Trainingsdaten	DA-Trainingsdaten	Validierungsdaten
Korrekte Verschweißung	478	23.900	126
Unzureichende Verschweißung	456	22.800	104
Schweißkrater	438	21.900	125
Schweißspritzer	455	22.750	102
Summe	1.827	91.350	457

beschrieben, wird dabei das Vorgehen von Xiao et al. mit kleinen Änderungen angewandt [182, S. 2]. Da in der Serienproduktion viele Bilder generiert werden, soll neben der Verbesserung der Güte des CNN die Größe des Bildes reduziert werden, wodurch notwendige Speicherressourcen minimiert werden können. Da sich die Vorverarbeitung der Schwarz-Weiß-Bilder nur minimal von der der RGB-Bilder unterscheidet, wird im Folgenden ausschließlich auf die Vorverarbeitungsschritte der RGB-Bilder eingegangen. Dabei ist auf der linken Seite der Abbildungen 5.13 bis 5.18 beispielhaft jeweils ein SS und auf der rechten ein SK dargestellt.

1. Wie bereits beschrieben, ist ein Ausgang der 3D-Kamera ein RGB-Bild, in welchem die Höheninformationen der Hairpin-Verschweißung codiert sind. Wie in Abbildung 5.12 dargestellt, befinden sich insgesamt vier Hairpins auf dem Bild. Die zwei äußeren sind teilweise abgeschnitten bzw. besitzen eine schlechte Aufnahmequalität.

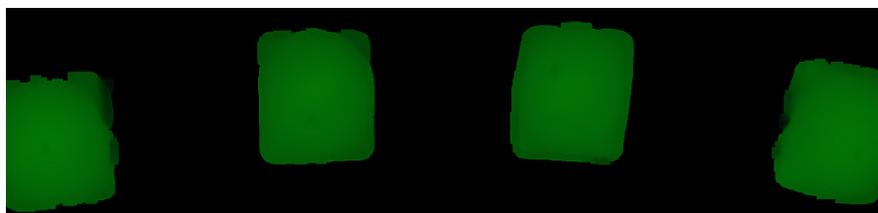


Abbildung 5.12: RGB-Rohdatenbild mit codierten Höheninformationen.

2. Das in Abbildung 5.12 dargestellt Bild ist das Ausgangsbild der Vorverarbeitungspipeline. Da sich die 3D-Kamera immer auf der selben Aufnahmehöhe befindet und somit die Verschweißung eines Hairpin-Paares immer derselben Pixelgröße

entspricht, werden jeweils die zwei mittleren Verschweißungen mit einer Größe von 450×450 Pixel um deren Schwerpunkt ausgeschnitten. Wie in Abbildung 5.12 jedoch zu erkennen ist, sind diese Verschweißungen kreisrund im Stator angeordnet. Somit sind diese Verschweißungen verdreht zueinander angebracht. Um diese Rotation zu beheben, werden die ausgeschnittenen Hairpins zurückgedreht. In Abbildung 5.13 ist beispielhaft eine ausgeschnittene und zurechtgedrehte Verschweißung dargestellt.



Abbildung 5.13: Erster Schritt der Datenvorverarbeitung: Ausschneiden und orthogonale Rotation des Ausgangsbildes.

3. Wie bereits erläutert, sind in den Ausgangsbildern der Keyence-Kamera die Höheninformationen codiert. Aufgrund dessen wird im zweiten Schritt der Vorverarbeitung diese Farbcodierung in Höheninformationen umgerechnet. Wie in Abbildung 5.14 zu erkennen, liegt die Höhe in einem Bereich von 0 bis 16 mm. Dies liegt daran, dass die verwendete Kamera einen Aufnahmebereich von 16 mm besitzt. Aus der in Abbildung 5.13 visualisierten Aufnahme ergibt sich die in Abbildung 5.14 dargestellte 3D-Darstellung der Verschweißung der Fehlerklassen SS (links) und SK (rechts).

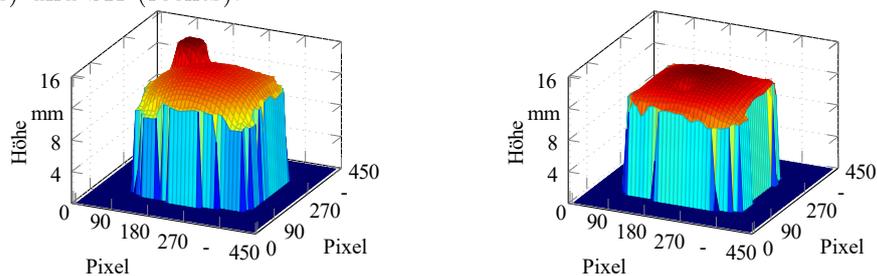


Abbildung 5.14: Zweiter Schritt der Datenvorverarbeitung: Resultierende Höheninformationen bei Umwandlung der Farbcodierung.

4. Da ausschließlich die Kuppel der Hairpin-Verschweißung von Interesse ist, wird im folgenden Vorverarbeitungsschritt die Kuppel in einem Bereich von 5 mm sowohl in positiver als auch negativer z -Richtung um den Median des Hairpins

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

ausgeschnitten. Das Ergebnis dieses Vorverarbeitungsschrittes ist beispielhaft in Abbildung 5.15 visualisiert.

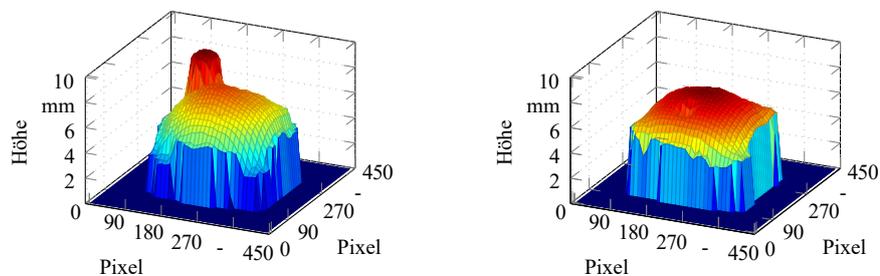


Abbildung 5.15: Dritter Schritt der Datenvorverarbeitung: Ausschnitt der Schweißkuppel durch Berechnung des Medians und Festlegung des darum liegenden Bereiches.

5. Aufgrund des hohen Speichervolumens der erzeugten Bilder, welche durch die Serienproduktion entstehen, ist es essenziell, die Größe der Aufnahme zu komprimieren, um dadurch den Speicherbedarf zu reduzieren. In Abbildung 5.16 ist beispielhaft die Verschweißung auf einen Größenbereich von 30×30 Pixel in x - und y -Richtung komprimiert. Diese Reduktion wird mithilfe eines Average-Pooling erreicht. Dadurch werden die Höheninformationen zusätzlich geglättet. Darüber hinaus werden diese in einen Bereich zwischen 0 und 255 skaliert. Diese beiden Schritte sind in Abbildung 5.16 visualisiert.

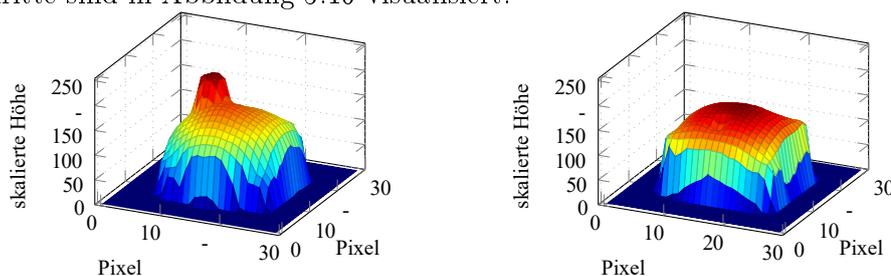


Abbildung 5.16: Vierter Schritt der Datenvorverarbeitung: Reduktion und Skalierung des Höhenprofils der Hairpin-Verschweißung.

6. Im fünften Schritt der Datenvorverarbeitung werden die aus den Schritten eins bis vier vorverarbeiteten Daten in ein Graustufenbild konvertiert. Dadurch kann das CNN das resultierende Bild einfacher optisch verarbeiten und die Fehlerklasse bestimmen. In Abbildung 5.17 ist das konvertierte Graustufenbild dargestellt.



Abbildung 5.17: Fünfter Schritt der Datenvorverarbeitung: Konvertieren des Bildes in ein Graustufenbild.

7. Im letzten Vorverarbeitungsschritt werden die Daten normalisiert. Dies hat zur Folge, dass alle Daten aufeinander abgestimmt sind und, wie in Abschnitt 5.7.2.2 beschrieben, die Güte des CNN somit deutlich höher ist. Außerdem können durch geringeren Trainingsaufwand bessere Ergebnisse erreicht werden. Wie in Abbildung 5.18 dargestellt, ergeben sich daraus die folgenden Ergebnisse für einen beispielhaften SS (vgl. Abbildung 5.18 links) und einen SK (vgl. Abbildung 5.18 rechts). Es ist gut mit bloßem Auge zu erkennen, dass ein zuvor kaum erkennbarer Krater durch die Normalisierung deutlich ausgeprägter ist. Ebenfalls ist zu sehen, dass die Normalisierung einen positiven Einfluss auf die Erkennung eines SS hat.



Abbildung 5.18: Sechster Schritt der Datenvorverarbeitung: Normalisierung des Datensatzes.

Bei der Vorverarbeitung der Schwarz-Weiß-Bilder werden ausschließlich die Schritte eins, zwei, fünf, sechs und sieben benötigt.

5.5 Aufbau der Struktur des Convolutional Neural Network

Wie in Abschnitt 5.1.4 beschrieben, gibt es drei Anforderungen an die Modellierung und Verwendung eines CNN in der Produktion. Jedoch ist es nicht möglich, alle Anforderungen komplett zu erfüllen. Aus diesem Grund wurde in Abschnitt 5.1.4 das magische Dreieck eingeführt. Für die Entwicklung des Modells wurden die folgenden Anforde-

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

rungen initial festgelegt. Diese Gewichtungen repräsentieren einen initialen Schritt und geben somit eine reine Tendenz an.

- Genauigkeit = 60%
- Größe des Netzwerks = 30%
- Größe des Eingangsbildes = 10%

Gemäß der Definition in Kapitel 4 wird eine Genauigkeit des CNN von mindestens 99% gefordert. Dies ist essenziell, da Qualitätsabweichungen erkannt werden müssen, sodass beispielsweise fehlerhafte Produkte nicht dem Kunden ausgeliefert werden. Aufgrund dessen ist Gewichtung der Genauigkeit mit 60% die höchste. Außerdem soll durch die Erkennung von Qualitätsabweichungen nicht die Taktzeit durch das CNN negativ beeinflusst werden. Die Einhaltung der Taktzeit ist ebenfalls wichtig, da bei steigender Taktzeit eine Produktionslinie weniger Ausbringung und somit das Unternehmen finanzielle Einbußen hat. Aus diesem Grund besitzt die Größe des Netzwerkes die zweithöchste Gewichtung. Die Größe des Eingangsbildes wird zunächst als eher unwichtig erachtet, da Speicherkosten über die letzten Jahre stark gesunken sind [201, 202].

5.5.1 Architektur des Netzes

Herausforderungen im Bereich der Bildklassifikation, wie beispielsweise ImageNet (vgl. Abschnitt 5.1.2), haben sich stark auf die Entwicklung von Klassifikatoren ausgewirkt. Support-Vektormaschinen oder Nearest-Neighbor-Klassifikationen waren die bevorzugten Methoden, bevor neuronale Netze große Anwendung fanden. Seitdem sind viele Netzwerkarchitekturen entwickelt worden, wie z.B. AlexNet [203], Visual Geometry Group (VGG) [204], Inception [205] oder ResNet [206]. Dabei verwenden alle Netzwerke Faltungen, um relevante Merkmale zu extrahieren. Die Größe der verwendeten Kernel variiert bei diesen Netzwerken. AlexNet verwendet unterschiedlich große Kernel, während VGG nur 3×3 -Kernel verwendet. Da, wie im Anhang B in Tabelle B.1 zu erkennen, die höchste Genauigkeit, Precision und die geringste Anzahl an trainierbaren Parametern mit dem VGG erzielt wurde, wird im Folgenden ausschließlich dieses Netzwerk in Betracht gezogen, welches aus sogenannten Convolution-Blöcken aufgebaut ist [204]. Wie in Abschnitt 2.4.5 beschrieben, bestehen diese Blöcke aus zwei oder mehreren Filtern mit dahinter geschalteter Aktivierungsschicht. Im Anschluss an die

5.5 Aufbau der Struktur des Convolutional Neural Network

Aktivierungsschicht folgt eine Pooling-Schicht. Dieser Aufbau ist in Abbildung 5.19 dargestellt.

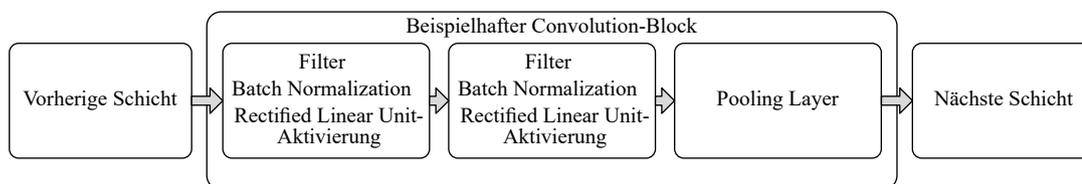


Abbildung 5.19: Aufbau des Convolutional-Blocks.

Die Besonderheit des in dieser Arbeit entworfenen CNN ist im Vergleich zum klassischen VGG-Netzwerk, dass, wie in Abbildung 5.19 zu erkennen, ergänzend vor jeder Aktivierungsschicht eine Batch Normalization (BN) eingefügt ist. Dadurch können eine bessere Generalisierung, ein schnellerer Trainingsprozess und eine Reduzierung des Overfittings erreicht werden (vgl. Abschnitt 2.4.4).

Abbildung 5.20 stellt das im Rahmen dieser Arbeit entworfene CNN dar. Die Grundstruktur des CNN basiert auf insgesamt vier Convolution-Blöcken. Wie in Abbildung 5.19 dargestellt, ist das CNN wie folgt aufgebaut:

1. Convolution-Blöcke eins bis drei:
 - (a) 2^{n+2} Filter der Größe 3×3 Pixel mit einem Stride=1. n beschreibt dabei die Nummerierung des Convolution-Blocks
 - (b) Batch Normalization
 - (c) ReLU-Aktivierungsfunktion
 - (d) 2^{n+2} Filter der Größe 3×3 Pixel.
 - (e) Batch Normalization
 - (f) ReLU-Aktivierungsfunktion
 - (g) Max-Pooling der Größe 2×2 Pixel mit einem Stride=2
2. Convolution-Block vier:
 - (a) 2^{n+2} Filter der Größe 3×3 Pixel mit einem Stride=1. n beschreibt dabei die Nummerierung des Convolution-Blocks

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

- (b) Batch Normalization
- (c) ReLU-Aktivierungsfunktion
- (d) 2^{n+2} Filter der Größe 3×3 Pixel.
- (e) Batch Normalization
- (f) ReLU-Aktivierungsfunktion
- (g) Global-Average-Pooling der Größe 2×2 Pixel mit einem Stride=2

Wie in Gleichung 5.3 zu erkennen, verdoppelt sich die Anzahl der Filter von dem einen auf den folgenden Convolution-Block.

$$Filter_n = 2^{n+2} \quad (5.3)$$

n definiert die Nummerierung des Convolution-Blocks. Somit werden in der ersten Schicht 8 Filter der Größe 3×3 verwendet. Der zweite Block besitzt 16 Filter, der dritte 32 Filter und der vierte 64 Filter. Am Ende des ersten bis dritten Blocks wird ein Maximal-Pooling verwendet. Für den letzten Block wird anstelle des Maximal-Poolings ein Global-Average-Pooling angewandt. Dadurch ergibt sich für alle vorhandenen Filter im letzten Convolution-Block ein eindimensionaler Merkmalsvektor mit 64 Einträgen. Anschließend folgen eine vollkommen vernetzte Schicht mit 32 Neuronen, erneut eine BN-Schicht, eine ReLU-Aktivierungsfunktion und eine Dropout-Schicht mit einer Dropout-Rate von 0,5. Abschließend folgt in dieser Struktur eine Ausgabeschicht mit einem Neuron für jede Klasse und einer Softmax-Aktivierungsfunktion, sodass als Ausgabe eine Wahrscheinlichkeit für jede Klasse erzeugt wird.

Die dabei verwendeten Regularisierungstechniken Global-Average-Pooling, BN und Dropout ermöglichen eine bessere Generalisierung sowie die Reduktion von Overfitting. Wie in den Anforderungen beschrieben, wurde das NN möglichst schlank entworfen. Insgesamt besitzt es damit nur 76.220 trainierbare Parameter, wodurch sich die Zeit des Trainingsprozesses und darüber hinaus die Prädiktionszeit reduziert.

5.5.2 Trainingsprozess des Convolutional Neural Network

Um die in Abschnitt 5.5.1 vorgestellte Architektur des CNN antrainieren zu können, müssen das Optimierungsverfahren, die Lernrate und die Kostenfunktion ausgewählt werden.

5.5.2.1 Optimierungsverfahren

Das sogenannte Gradient Descent-Verfahren, oder auch Delta-Regel genannt, ist eine der meist genutzten Methoden zur Optimierung eines NN während des Trainingsprozesses. Dabei gibt es unterschiedliche Varianten des Gradient-Descent: der Batch-Gradient Descent (GD)-, der Stochastic-GD- und Mini-Batch-GD-Optimierer. Sie sind eine Erweiterung des Gradient Descent-Verfahrens. Um die optimalen Gewichtungen des CNN zu finden, wurde in der vorliegenden Arbeit der sogenannte Mini-Batch-Gradient-Descent-Optimierer verwendet. Zur schnelleren Findung des Optimums wurde eine adaptive Lernrate von 0,001 verwendet. Diese reduziert sich alle fünf Trainingsepochen um das 0,9-Fache. Nachteil dieser Methoden ist jedoch, dass der Trainingsprozess stark von den initialisierten Gewichtungen abhängt. Darüber hinaus kann das Optimierverfahren in einem lokalen Minimum enden. Um diesen Nachteilen entgegenzuwirken, wird in der vorliegenden Arbeit das Nesterov Momentum mit einem Wert von 0,9 angewandt.¹

5.5.2.2 Kostenfunktion

Die in der vorliegenden Arbeit verwendete Kostenfunktion, mit welcher die Gradienten zum Trainieren des CNN kalkuliert werden, basiert auf der Kreuzentropie. Sie ist durch die Gleichung 5.4 definiert.

$$E = - \sum_{c=1}^M w_c \cdot l_c \cdot \ln(y_c) \quad (5.4)$$

Dabei steht E für die Kosten der Prädiktion, welche durch die Summe der prädiktierten Wahrscheinlichkeiten y_c für alle M Klassen entsteht. Der Indikator l_c ist 1, falls das CNN der richtigen Klasse entsprechend dem Eingang zugeordnet hat. Ansonsten ist l_c gleich 0. Wird beispielsweise der Eingang des Netztes mit einer Wahrscheinlichkeit von 100% der dazugehörigen richtigen Klasse zugeordnet, wird der Fehler durch den negativen natürlichen Logarithmus gleich 0. w_c steht für die Gewichtung der jeweiligen Klasse (vgl. Abschnitt 5.5.2.3).

¹Aufgrund der weiten Verbreitung dieser Methoden wird auf diese nicht tiefer eingegangen. Diese sind jedoch in der folgenden Standardliteratur detailliert beschrieben [207, S. 41–54], [208, S. 2–3], [52, S. 80–81].

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

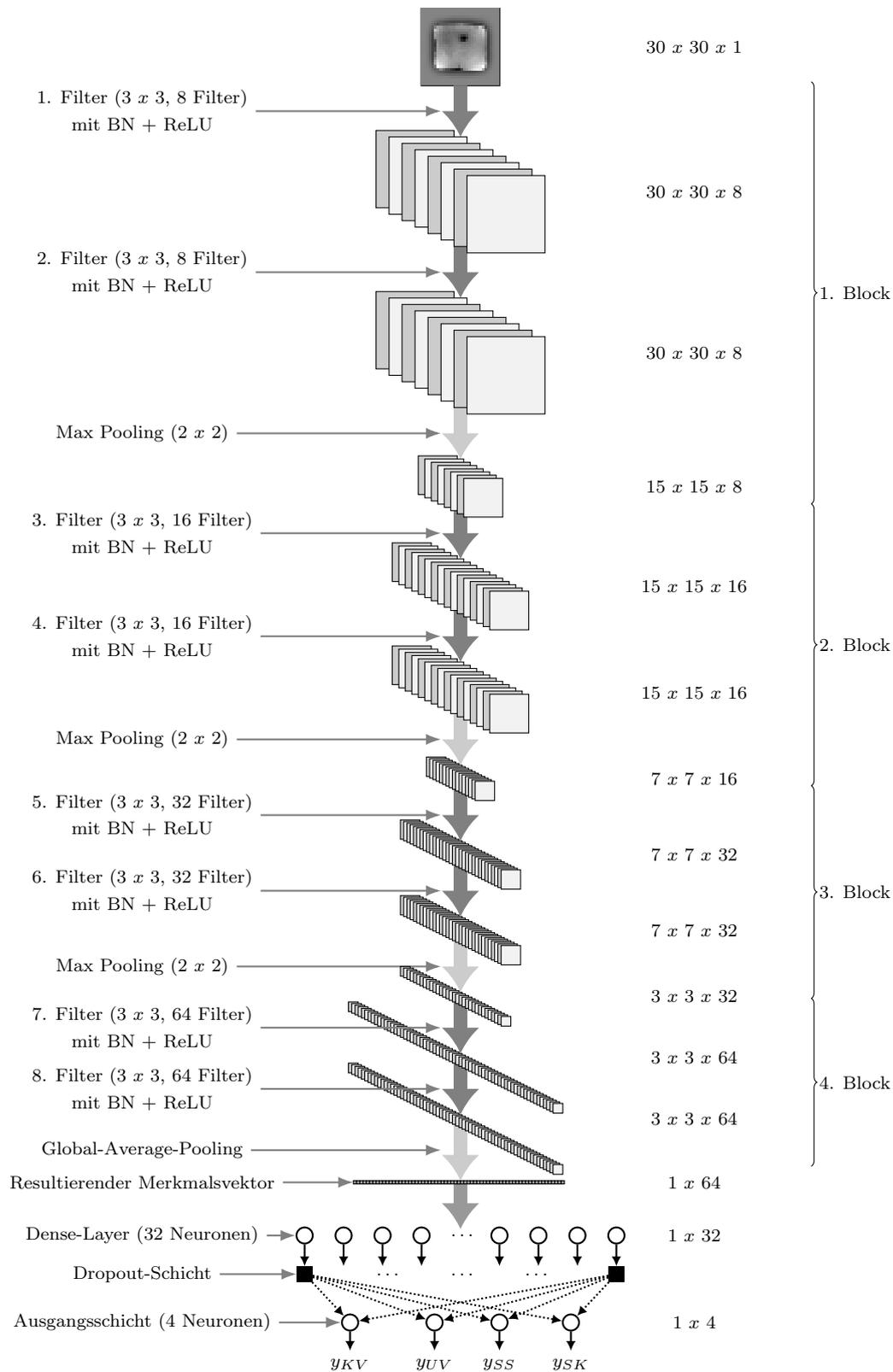


Abbildung 5.20: Entworfenes CNN-Modell für den in dieser Arbeit betrachteten Anwendungsfall.

5.5.2.3 Klassengewichtung

Wie bereits beschrieben, ist es essenziell, dass fehlerhafte Verschweißungen nicht als gute Verschweißungen klassifiziert werden. Um diese Wichtigkeit beim Antrainieren des Modells zu berücksichtigen, werden Klassengewichtungen in die in Abschnitt 5.5.2.2 definierte Kostenfunktion 5.4 integriert. Dadurch werden bei der falschen Klassifizierung einer Hairpin-Verschweißung die Kosten höher gewertet und dadurch der Gradient stärker beeinflusst. Beim vorliegenden Anwendungsfall werden daher die Klassen wie folgt gewichtet:

$$[w_{KV}, w_{UV}, w_{SS}, w_{SK}] = [1, 10, 10, 10] \quad (5.5)$$

Dabei steht w_i mit $i \in \{KV, UV, SS, SK\}$ für die Gewichtung der entsprechenden Klasse. Dies führt dazu, dass fehlerhafte Verschweißungen (UV, SS, SK) mit dem Faktor 10 in den Kosten höher gewichtet werden als eine falsche Klassifizierung der korrekten Verschweißung.

5.6 Validierung des Modells

Im Folgenden wird die definierte Architektur des CNN zunächst durch die Verwendung der 5-fold Cross-Validation klassisch und anschließend visuell überprüft.

5.6.1 Kennzahlenbasierte Validierung

Abbildung 5.21 stellt einen Trainingsverlauf der 5-fold Cross-Validation sowohl für die Schwarz-Weiß- als auch die 3D-Bilder dar. Dabei sind die Verläufe der Genauigkeit über die Trainingsepochen in Abbildung 5.21a zum einen für die Trainingsdaten als auch die Validierungsdaten dargestellt. Zum anderen sind ebenfalls die Verläufe der Kosten über die Epochen für Trainings- und Validierungsdaten in Abbildung 5.21b visualisiert. Beide Verläufe der 3D-Bilder zeigen, dass der Trainingsprozess mit hoher Genauigkeit und geringen Kosten sehr stabil ist. Hervorzuheben ist, dass keine Schwankungen im Verlauf zu erkennen sind und darüber hinaus durch die in das Netz eingebauten Regularisierungstechniken kein Over- oder Underfitting festzustellen ist. Die Trainingsverläufe mit den Schwarz-Weiß-Bildern weisen jedoch starke Schwankungen auf. Des Weiteren kann nur eine deutlich geringere Genauigkeit erreicht werden.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

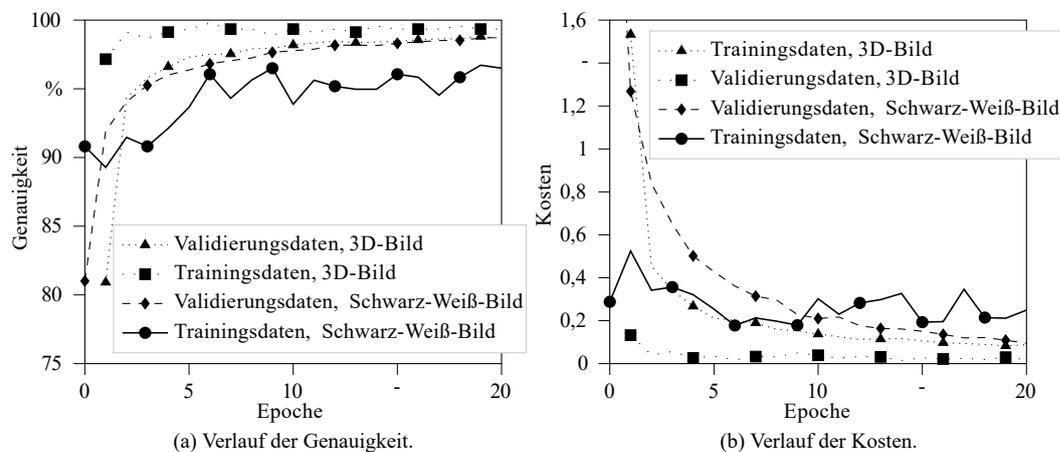


Abbildung 5.21: Resultierender Trainings- und Validierungsverlauf der Kosten und der Genauigkeit eines Trainings der 5-fold Cross-Validation der definierten CNN-Architektur.

In Abbildung C.1 im Anhang C sind die CM der 5-fold Cross-Validation mit 3D-Bildern als Eingangsgröße dargestellt. Aufgrund der hohen Einträge auf den Diagonalen ist gut zu erkennen, dass das erstellte Modell mit einer hohen Genauigkeit die unterschiedlichen Klassen voneinander abgrenzen kann. Einträge, welche nicht auf den Diagonalen liegen, sind vom Netz falsch klassifiziert. Aus den im Anhang C dargestellten CM werden im Folgenden verschiedene Bewertungsmetriken berechnet. Dadurch kann eine direkte Aussage über die Güte des entworfenen Netzwerks erstellt werden. In Tabelle 5.3 sind die Ergebnisse der 5-fold Cross-Validation mit den 3D-Bildern als Eingangsgröße aufgelistet. Anzumerken dabei ist, dass die Metriken *Precision* und *Recall* sich ausschließlich auf die Klasse KV beziehen.

Es ist zu erkennen, dass eine Genauigkeit von 99,56% erreicht wurde. Außerdem zeigt die Metrik *Precision* mit einem durchschnittlichen Wert von 99,15%, dass das Modell gute und zuverlässige Ergebnisse für die korrekte Erkennung und Klassifizierung einer KV liefert. Des Weiteren ist der *Recall* und somit auch der F_1 -Score über 99%.

Außerdem sind in Tabelle 5.4 die Ergebnisse der 5-fold Cross-Validation mit den Schwarz-Weiß-Bildern als Eingangsgröße dargestellt. Die erreichte Genauigkeit erzielt

Tabelle 5.3: Ergebnisse Validierung für die Läufe 1 bis 5 mit 3D-Bildern.

Lauf	Genauigkeit (%)	KV Precision (%)	KV Recall (%)	KV F_1-Score (%)
1	100,0	100,0	100,0	100,0
2	99,34	100,0	97,83	98,90
3	99,34	98,31	99,19	98,72
4	99,34	98,39	99,15	98,79
5	99,78	99,05	100,0	99,52
Durchschnitt	99,56	99,15	99,23	99,19

nur einen Wert von 95,01%. Wie bereits erläutert, ist es besonders in der Produktion essenziell, dass fehlerhaft verschweißte Hairpins nicht als korrekte Verschweißung klassifiziert werden. Diese Fälle befinden sich in der in Abbildung C.2 dargestellten CM in der ersten Spalte und sind durch einen hellgrauen Hintergrund hervorgehoben. Die Ergebnisse zeigen, dass im Durchschnitt pro Durchlauf 1,8 Hairpin-Verschweißungen der nicht zutreffenden Klasse KV zugeordnet wurden. Aus diesem Grund weist die Precision bei der Verwendung von Schwarz-Weiß-Bildern keine guten Werte auf. Darüber hinaus erreicht der Recall einen geringen Wert (93,8%). Dies würde bedeuten, dass fälschlicherweise viele Verschweißungen durch das Personal geprüft werden müssten, obwohl diese korrekt verschweißte wurden. Da dies nicht den in Kapitel 4 vorgegebenen Anforderungen entspricht, wird im weiteren Verlauf der Arbeit nur die Verwendung von 3D-Bildern betrachtet.

Tabelle 5.4: Ergebnisse Validierung für die Läufe 1 bis 5 mit Schwarz-Weiß-Bildern.

Lauf	Genauigkeit (%)	KV Precision (%)	KV Recall (%)	KV F_1-Score (%)
1	96,06	99,15	95,08	97,07
2	93,22	97,69	92,03	94,78
3	94,31	97,35	94,02	95,65
4	95,62	98,28	92,68	95,40
5	95,83	100,0	95,19	97,54
Durchschnitt	95,01	98,49	93,8	96,09

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

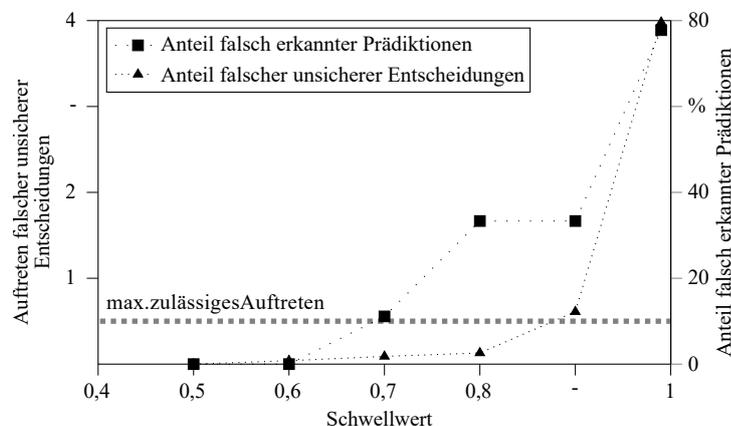


Abbildung 5.22: Vergleich der Anteile an falsch erkannten Prädiktionen und dem Auftreten falscher unsicherer Entscheidungen in Abhängigkeit des dafür festgelegten Schwellwerts.

Um die Anzahl fälschlicherweise fehlerfrei klassifizierter Verschweißungen zu reduzieren, wurde die Klasse unsichere Entscheidung (UE) eingeführt. Wie bereits beschrieben, gibt die Softmax-Funktion am Ausgang des CNN die Wahrscheinlichkeit der Klassenzugehörigkeit an. Dieser Wert kann mit einem Schwellwert abgeglichen werden, sodass bei Unterschreitung des Schwellwerts die Entscheidung des Netzes als UE gewertet wird. Als Folge dessen muss das Personal an der Produktionslinie die Verschweißung optisch überprüfen, um sicherzugehen, dass keine fehlerhafte Verschweißung die Produktionslinie verlässt. Bei der Festlegung des Schwellwerts ist es deshalb wichtig, dass möglichst viele fehlerhafte Verschweißungen erkannt werden, welche fälschlicherweise durch das CNN als korrekte Verschweißung klassifiziert wurden. Gleichzeitig darf jedoch eine UE nicht zu häufig auftreten, denn dadurch müsste das Personal zu oft Verschweißungen manuell überprüfen. Abbildung 5.22 verdeutlicht dabei das Vorgehen zur Findung des Schwellwerts. Um den manuellen Aufwand des Personals zu reduzieren, darf eine fehlerhaft zugeordnete UE bei maximal 0,5 % aller Verschweißungen auftreten (vgl. Abbildungen 5.22). Wie zu erkennen, gibt es zwischen den Schwellwerten 0,8 und 0,9 keine Verbesserung zur Klassifizierung einer falschen KV. Aufgrund dessen wurde ein Schwellwert von 0,8 gewählt. Wie in Abbildung 5.22 dargestellt, liegt die dabei maximal zulässige Anzahl falscher UE deutlich unter 0,5%.

5.6.2 Visuelle Validierung

Wie bereits beschrieben, basiert das Antrainieren eines NN nicht auf Expertenwissen, sondern auf großen Mengen an Trainingsdaten. Die in Abschnitt 5.6 verwendeten Metriken ermöglichen zwar die Bewertung der Güte eines Klassifikators, es kann jedoch nicht bewertet werden, aufgrund welcher Merkmale das Modell eine Klasse wählt. Wie in Kapitel 4 erläutert, ist jedoch für die Anwendung des CNN in der Produktion eine Anforderung, dass der Klassifikationsalgorithmus menschenlesbar ist. Es kann beispielsweise vorkommen, dass das Netz unwichtige Merkmale, die sich im Hintergrund eines Bildes befinden, erlernt und somit aufgrund eines falschen Merkmals Entscheidungen trifft. Um dies ausschließen und somit den Entscheidungsprozess des CNN besser nachvollziehen und bewerten zu können, werden andere Verfahren wie beispielsweise Grad-CAM [76, S. 618–621] und ISCSV [77, S. 1–2] benötigt. Wie in Abschnitt 5.1.6 beschrieben, kann dadurch visuell veranschaulicht werden, welche Bereiche des Eingangsbildes für die Entscheidung des CNN relevant waren. Abbildung 5.23 stellt auf der linken Hälfte die visuelle Evaluation mithilfe der Grad-CAM und auf der rechten Hälfte die visuelle Evaluierung durch die Anwendung der ISCSV-Methode dar. In der Mitte ist das vorverarbeitete Eingangsbild des CNN der vier Fehlerbilder visualisiert. Analog Abschnitt 5.2 sind in Abbildung 5.23 die Grenzmuster dargestellt.

In Abbildung 5.23a und 5.23b ist die visuelle Validierung der Klasse KV dargestellt. Dabei ist deutlich durch die Anwendung der Grad-CAM entstandene Heatmap zu erkennen, dass die gleichmäßig aufgeschmolzene Kuppel, welche in Abschnitt 5.3 durch den Experten als ausschlaggebendes Merkmal definiert wurde, hervorgehoben wird. Die Saliency Maps des ISCSV-Verfahrens, welche jeweils auf der rechten Seite der Abbildung dargestellt sind, heben ebenfalls die Kanten der Pins hervor.

Bei der Fehlerklasse UV mit der Unterteilung schlechter und sehr schlechter Schweißqualität sind bei der Heatmap der sich in der Mitte befindende unzureichende Schweißpunkt und zum anderen die Kronen an den Ecken der Hairpin-Paare bei der Saliency Map visuell hervorgehoben (vgl. Abbildung 5.23c und 5.23d). Diese Merkmale wurden, wie in Abschnitt 5.2 beschrieben, von einem Experten genannt. Somit kann eine korrekte Entscheidung des CNN bestätigt werden.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

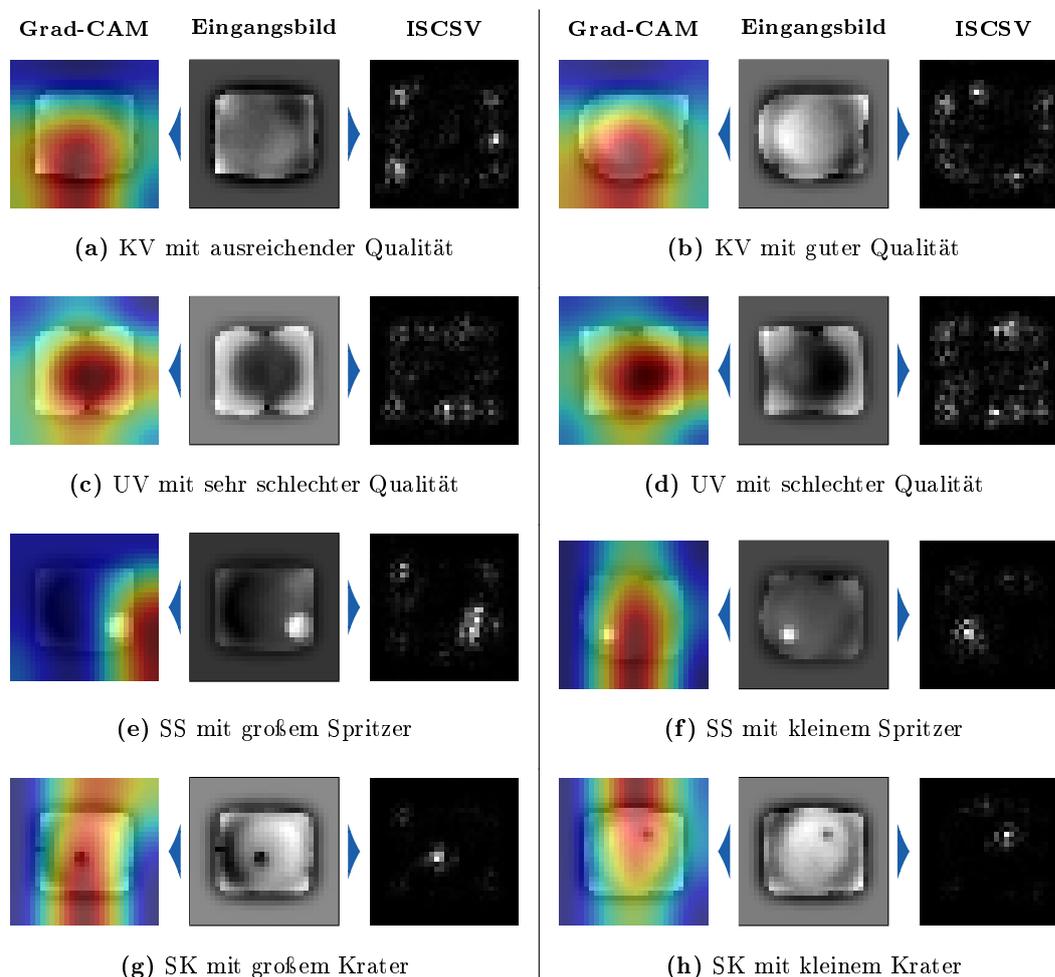


Abbildung 5.23: Anwendung von Grad-CAM (links) und ISCSV (rechts) auf das vorverarbeitete Eingangsbild für das CNN für alle relevanten Klassen sowie deren Abstufungen.

Die Hervorhebung der Merkmale der letzten Klassen SK und SS ist in den Abbildungen 5.23e bis 5.23h dargestellt. Es ist deutlich zu erkennen, dass durch die Heatmap der Bereich eingefärbt wird, in welchem sich der Spritzer bzw. der Krater befindet. Durch die Saliency Map wird eindeutig der Spritzer bzw. der Krater hervorgehoben. Somit entscheidet auch bei diesen beiden Fehlerbildern das CNN aufgrund der richtigen Merkmale.

5.7 Sensitivitätsanalyse durch eine Parametervariation

Es kann somit festgehalten werden, dass mithilfe von Grad-CAM und ISCSV visuell das Modell evaluiert werden kann. Es ist zu erkennen, dass das in dieser Arbeit entwickelte CNN korrekt die durch den Experten definierten Merkmale der jeweiligen Fehlerklasse erkennt und interpretiert. Ebenfalls wird die Güte der entwickelten Netzarchitektur durch die oben beschriebene kennzahlenbasierte Evaluation bestätigt. Dadurch ist eine korrekte Klassifizierung der Hairpin-Verschweißungen und die Anforderung an Menschenlesbarkeit sichergestellt.

5.7 Sensitivitätsanalyse durch eine Parametervariation

Wie in Abschnitt 5.6 dargestellt, sind die Anforderungen an die Genauigkeit des entwickelten CNN erfüllt. In diesem Abschnitt wird eine Sensitivitätsanalyse der Hyperparameter durchgeführt, um deren Auswirkungen auf die Anforderungen des magischen Dreiecks zu vergleichen. Die zu variierenden Parameter wurden dafür in zwei Gruppen aufgeteilt. In der ersten Gruppe befinden sich Parameter, welche den Aufbau des CNN bestimmen. In der zweiten Gruppe sind Parameter verortet, welche die verwendeten Eingangsdaten beeinflussen. Für diese Sensitivitätsanalyse wurden Trainings- und Validierungsdaten in einem Verhältnis von 0,8 zu 0,2 aufgeteilt. Außerdem wurde wie zuvor eine 5-fold Cross-Validation angewandt.

5.7.1 Parametervariation: Aufbau des Convolutional Neural Network

Wie erwähnt, werden in der ersten Gruppe die Hyperparameter, welche den Netzaufbau beeinflussen, variiert. Diese Parameter wurden zusätzlich mit der verwendeten Bildgröße verglichen. Im Folgenden sind diese Parameter und der Variationsbereich aufgelistet.

- Filtergröße = 2; 4; 6; 8; 10; 12; 14; 18; 20
- Anzahl der Convolution-Blöcke = 1; 2; 3; 4; 5; 6; 7
- Anzahl der Filter = 4; 8; 32; 64
- Bildgröße = 15; 30; 45; 65; 90; 150

Außerdem ist es wichtig zu beachten, dass keine DA angewandt wurde, denn dies hätte den Trainingsaufwand deutlich erhöht.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

5.7.1.1 Variation der Filtergröße

Zu Beginn der Parametervariation wird die Filtergröße variabel gewählt. Abbildung 5.24 stellt die Genauigkeit der Validierungsdaten des resultierenden Netzes über die Filtergröße dar. Des Weiteren wurde die verwendete Bildgröße variiert. Bei der Variation betrug die Anzahl der Convolution-Blöcke 4 und die Anzahl der Filter des ersten Convolution-Blocks 8.

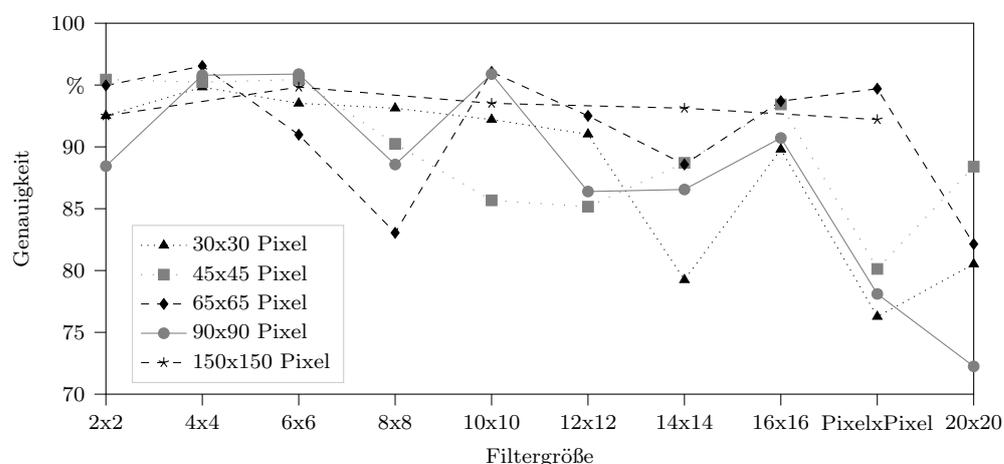


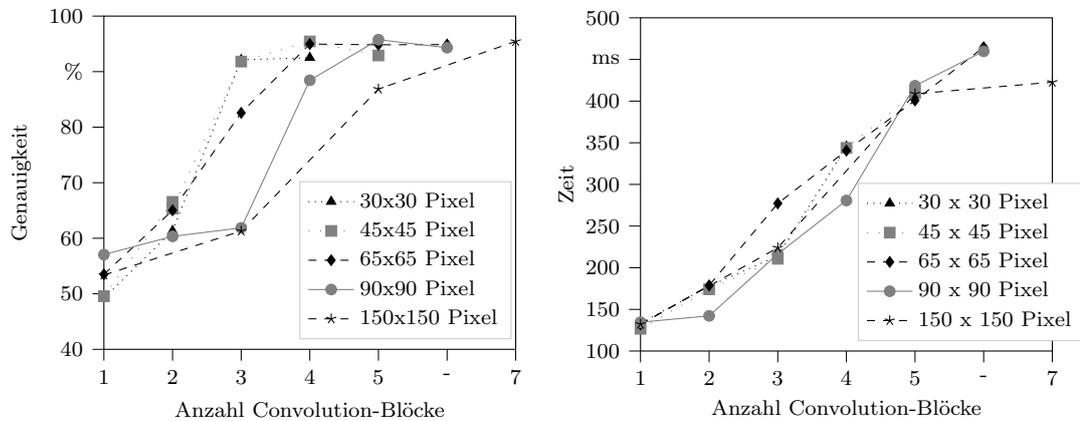
Abbildung 5.24: Resultierender Verlauf der Genauigkeit der Validierungsdaten während des Trainingsprozesses bei Variation der Filtergröße.

Es ist erkennbar, dass die Genauigkeit sehr starke Fluktuationen über die Filtergröße aufweist. Die höchste Genauigkeit konnte bei einer Bildgröße von 65×65 Pixeln und einer Filtergröße von 4×4 Pixel erzielt werden. Weitere Erkenntnisse konnten jedoch nicht aus der Variation der Filtergröße gewonnen werden.

5.7.1.2 Variation der Anzahl der Convolution-Blöcke

Ein weiterer wichtiger Aspekt beim Aufbau eines CNN ist dessen gewählte Tiefe, welche unter anderem durch die Anzahl an Convolution-Blöcken bestimmt wird. Da dieser Parameter sowohl die Genauigkeit als auch die Prädiktionszeit beeinflusst, wird in diesem Abschnitt die Anzahl der Convolution-Blöcke variiert. Bei der Variation beträgt die Filtergröße 2 und die Anzahl der Filter des ersten Convolution-Blocks 8. In Abbildung 5.25a ist die Genauigkeit der Validierungsdaten in Abhängigkeit der Anzahl der verwendeten Convolution-Blöcke dargestellt. Zusätzlich wurde die Bildgröße variiert.

5.7 Sensitivitätsanalyse durch eine Parametervariation



(a) Einfluss der Anzahl der Convolution-Blöcke auf die Genauigkeit. (b) Einfluss der Anzahl der Convolution-Blöcke auf die Prädiktionszeit.

Abbildung 5.25: Einfluss der Anzahl der Convolution-Blöcke auf die Genauigkeit und Prädiktionszeit.

Es ist zu erkennen, dass mit dem Anstieg der verwendeten Convolution-Blöcke ein Anstieg der Genauigkeit einhergeht. Vor allem beim Anstieg der Bildgröße, wie beispielsweise einer Größe von 90×90 oder 150×150 Pixeln, sind die Kurven nach hinten verschoben. Somit lässt sich schlussfolgern, dass für größere Eingangsbilder mehr Convolution-Blöcke notwendig sind.

Da neben der Genauigkeit des Netzes die Prädiktionszeit in der Serienproduktion ein wichtiger Punkt ist, wird im Folgenden die Prädiktionszeit über die Anzahl an Convolution-Blöcken analysiert. In Abbildung 5.25b ist die Prädiktionszeit über die Anzahl der Convolution-Blöcke dargestellt. Es ist deutlich zu erkennen, dass mit steigender Anzahl an Convolution-Blöcken die Prädiktionszeit steigt. Außerdem ist zu erkennen, dass die Bildgröße keinen Einfluss auf die Prädiktionszeit hat. Aufgrund der größer werdenden Prädiktionszeit bei steigender Anzahl an Convolution-Blöcken ist die Verwendung einer Bildgröße von 30×30 Pixeln bei 4 Convolution-Blöcken zu empfehlen. Würde ein größeres Eingangsbild genutzt werden, müssten mehr Convolution-Blöcke verwendet werden. Dies würde sich jedoch negativ auf die Prädiktionszeit auswirken.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

5.7.1.3 Variation der Anzahl der Filter

Neben der Tiefe des CNN wird in diesem Abschnitt die Breite des Netzes analysiert, welche ebenfalls einen Einfluss auf die Anzahl der Parameter und somit auf die Genauigkeit und Prädiktionszeit hat. Um den Einfluss auf die Genauigkeit zu untersuchen, wurde deswegen die Anzahl der Filter in den Convolution-Blöcken variiert (vgl. Abbildung 5.26a).

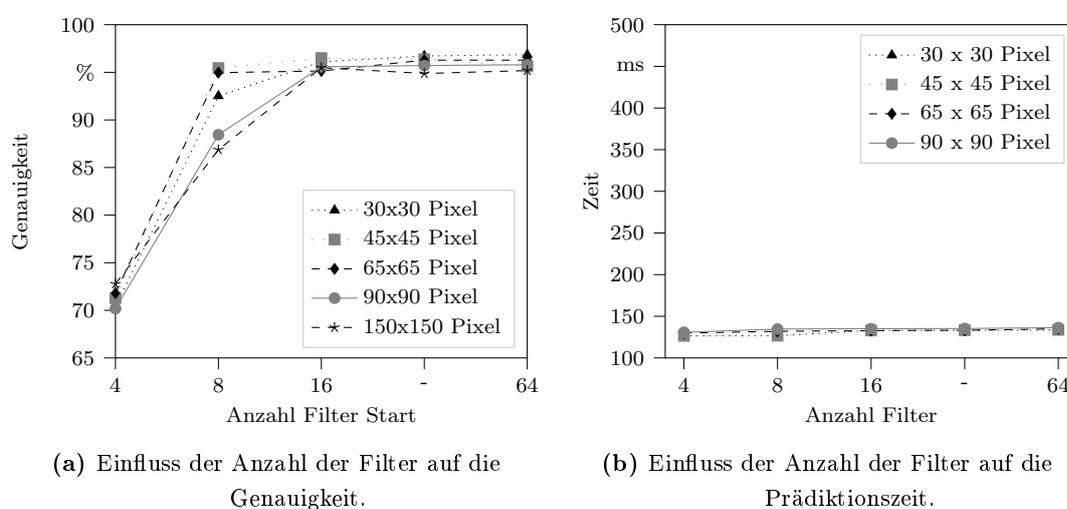


Abbildung 5.26: Einfluss der Anzahl der Filter auf die Genauigkeit und Prädiktionszeit.

Bei der Variation beträgt die Filtergröße 2×2 Pixel und die Anzahl an Convolution-Blöcken 4. Wie bereits in Abschnitt 5.5.1 beschrieben, verdoppelt sich die Anzahl der Filter von einem auf den nächsten Convolution-Block. Im Folgenden wird ausschließlich die Anzahl der Filter im ersten Block angegeben. Beträgt der angegebene Wert beispielsweise 16, besitzt der erste Convolution-Block 16 Filter, der zweite 32 Filter und so weiter. Es ist gut zu erkennen, dass sich die Genauigkeit zwischen 4 und 8 Filter im ersten Convolution-Block stark verbessert. Bei weiterer Erhöhung verbessert sich die Genauigkeit jedoch nicht. Aufgrund dessen wird eine Wahl von 8 Filtern im ersten Block empfohlen. Außerdem unterscheidet sich der Verlauf der Genauigkeit über die Anzahl der Filter bei Variation der Bildgröße ebenfalls nur marginal.

Analog der Analyse der Auswirkungen durch die Erhöhung der Convolution-Blöcke wird in Abbildung 5.26b die Auswirkung der Anzahl der Filter auf die Prädiktionszeit

5.7 Sensitivitätsanalyse durch eine Parametervariation

genauer betrachtet. Es ist zu erkennen, dass die Anzahl der Filter nur einen minimalen Einfluss auf die Prädiktionszeit hat.

5.7.1.4 Auswahl einer geeigneten Netzarchitektur

Wie oben zu erkennen, gibt es viele Einflussfaktoren, welche die Auswahl eines geeigneten Netzwerkes erschweren. Die Einführung des magischen Dreiecks, welches in Abbildung 5.2 dargestellt ist, soll dabei helfen, eine geeignete Netzwerkarchitektur in Abhängigkeit der Anforderungen auszuwählen. Dafür wurden zunächst die in Abschnitt 5.7.1 gelisteten Parameter des CNN vollfaktoriell miteinander kombiniert. Darüber hinaus wurde die folgende Kostenfunktion entwickelt:

$$Kosten = g_1 * p_{PZ} + g_2 * p_{BG} + g_3 * p_{iQ} \quad (5.6)$$

Dabei beschreibt p_{PZ} den Parameter Prädiktionszeit, p_{BG} die Bildgröße und p_{iQ} die Fehlklassifizierungsrate des CNN. Außerdem definieren die Gewichtungen $g_i \in [0, 1]$, $i \in \mathbb{N}$ und der Bedingung $\sum_{i=1}^N g_i = 1$, mit welcher Stärke ein Einflussfaktor (Prädiktionszeit, Bildgröße oder Fehlklassifizierungsrate) N gewichtet werden soll. Die Fehlerklassifikationsrate ist nach Gleichung 2.19 durch die Differenz $p_{iQ} = 1 - acc_{gesamt}$ definiert. Um die unterschiedlichen Wertebereiche auszugleichen, wurden zudem die Einflussfaktoren, wie in Gleichung 5.7 beschrieben, normalisiert.

$$\tilde{p} = \frac{p - \min(p)}{\max(p) - \min(p)} \quad (5.7)$$

Exemplarisch werden im Folgenden drei unterschiedliche Gewichtungen genommen und jeweils die besten fünf Netzwerkarchitekturen in den dazugehörigen Tabellen aufgelistet:

1. $Kosten = 0 * p_{PZ} + 0 * p_{BG} + 1 * p_{iQ}$ siehe Tabelle 5.5
2. $Kosten = 0,4 * p_{PZ} + 0 * p_{BG} + 0,6 * p_{iQ}$ siehe Tabelle 5.6
3. $Kosten = 0,3 * p_{PZ} + 0,2 * p_{BG} + 0,5 * p_{iQ}$ siehe Tabelle 5.7

In Tabelle 5.5 ist der Schwerpunkt ausschließlich auf die Genauigkeit des Netzwerkes gelegt. Dabei sind die besten Ergebnisse in Bezug auf die Genauigkeit des Validierungsdatensatzes aufgelistet. Hierbei ist zu beachten, dass kein DA verwendet wurde.

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

Tabelle 5.5: Beste 5 Ergebnisse bezogen auf die Genauigkeit ($Kosten = 0 * p_{PZ} + 0 * p_{BG} + 1 * p_{iQ}$).

#	Convolution-Blöcke	Filter	Filtergröße	Prädiktionszeit	Genauigkeit	Bildgröße	Kosten
1	5	8	6x6	319,59	0,96848	90	0
2	4	64	2x2	283,76	0,96848	30	0
3	4	32	2x2	269,57	0,96716	30	0,0024
4	3	64	2x2	192,72	0,96585	30	0,0047
5	4	16	2x2	330,37	0,96498	45	0,0063

Es ist zu erkennen, dass die ersten beiden Platzierungen eine identische Genauigkeit des Validierungsdatensatzes von 96,85% erreichen konnten. Interessant ist jedoch, dass die Prädiktionszeit der ersten Platzierung ca. 11% größer ist. In der Serienproduktion, in der möglichst viele Produkte hergestellt werden, könnte sich eine längere Prädiktionszeit negativ auf die Taktzeit auswirken. Aus diesem Grund wird im Folgenden in der Kostenfunktion die Prädiktionszeit ebenfalls mit einem Faktor von 0,4 gewichtet. Die Ergebnisse sind in Tabelle 5.6 aufgelistet.

Tabelle 5.6: Beste 5 Ergebnisse bezogen auf die Kostenfunktion: $Kosten = 0,4 * p_{PZ} + 0 * p_{BG} + 0,6 * p_{iQ}$.

#	Convolution-Blöcke	Filter	Filtergröße	Prädiktionszeit	Genauigkeit	Bildgröße	Kosten
1	3	64	2x2	192,72	0,96585	30	0,0127
2	3	16	6x6	187,92	0,96323	45	0,0149
3	4	32	2x2	269,57	0,96716	30	0,0197
4	4	64	2x2	283,76	0,96848	30	0,0198
5	3	64	6x6	221,14	0,96148	90	0,0205

Erkennbar ist, dass nun neben der hohen Genauigkeit des Validierungsdatensatzes die Prädiktionszeit eine wichtige Rolle spielt. Für nachgelagerte Analysen ist es essenziell, dass die erzeugten Bilder der Verschweißung archiviert werden. Der dafür nötige Speicherplatz verursacht jedoch ebenfalls Kosten, welche maßgeblich durch die Größe des Bildes beeinflusst werden. Aufgrund dessen sind in Tabelle 5.7 die Ergebnisse dar-

5.7 Sensitivitätsanalyse durch eine Parametervariation

gestellt, bei welchen die Bildgröße mit dem Faktor 0,2, die Prädiktionszeit mit 0,4 und die Genauigkeit ebenfalls mit 0,4 gewichtet wurde.

Tabelle 5.7: Besten 5 Ergebnisse bezogen auf die Kostenfunktion: $Kosten = 0,4 * p_{PZ} + 0,2 * p_{BG} + 0,4 * p_{iQ}$.

#	Convolution-Blöcke	Filter	Filtergröße	Prädiktionszeit	Genauigkeit	Bildgröße	Kosten
1	3	64	2x2	192,72	0,96585	30	0,0117
2	4	32	2x2	269,57	0,96717	30	0,0192
3	3	32	2x2	216,11	0,95841	30	0,0196
4	4	64	2x2	283,76	0,96848	30	0,0198
5	3	16	2x2	215,64	0,95579	30	0,0214

Tabelle 5.7 spiegelt die Ergebnisse der Parametervariation wider. Es ist zu erkennen, dass bei einem Netzwerk mit wenigen Convolution-Blöcken kleine Bilder verwendet werden können (vgl. Abbildung 5.25a), wobei trotzdem eine hohe Genauigkeit erreicht werden kann. Des Weiteren hat Abbildung 5.25b gezeigt, dass bei wenigen Convolution-Blöcken die Prädiktionszeit deutlich geringer ist. Die Anzahl der Filter, wie in Abbildung 5.26b dargestellt, hat kaum Auswirkungen auf die Prädiktionszeit.

Wie erwähnt, gibt es viele Faktoren, welche die Wahl einer geeigneten Architektur beeinflussen. Aufgrund dessen wurde in der vorliegenden Dissertation eine Kostenfunktion eingeführt, wodurch der Einfluss der Netzwerkarchitektur auf die Genauigkeit, die Prädiktionszeit und die benötigten Speicherressourcen bestimmt werden kann. Diese Ausprägungen orientieren sich an dem in Abschnitt 5.1.4 beschriebenen magischen Dreieck. Mithilfe dieses Vorgehens kann in Abhängigkeit der definierten Anforderungen der optimale Aufbau des CNN für die optische Erkennung von Qualitätsabweichungen in der Produktion entwickelt werden.

5.7.2 Parametervariation: Daten

Analog zu Abschnitt 5.7.1 werden in den folgenden Abschnitten diejenigen Parameter variiert, welche die vorhandenen Daten und deren Verarbeitung beeinflussen. Die dabei betrachteten Parameter und deren zu variierender Bereich sind wie folgt:

- Anzahl Datenvervielfältigung = 1; 2; 4; 8; 16; 32; 64; 128

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

- Normalisierungstechnik = *keine Normalisierung*; -1 bis 1 ; *Mean/SD*
- Dropout-Rate = $0,25$; $0,5$; $0,75$

Um die Auswirkungen der Parameter auf die Güte des CNN miteinander vergleichen zu können, wurde einer der oben gelisteten Parameter variiert. Die restlichen Parameter wurden analog zu der in Abschnitt 5.5.1 beschriebenen Architektur (vgl. Abbildung 5.20) gewählt.

5.7.2.1 Variation der Größe des Trainingsdatensatzes

Zu Beginn wird der Einfluss der Datenvervielfältigung auf die Genauigkeit des Trainingsprozesses evaluiert. Hierbei wird, wie in Abbildung 5.27 dargestellt, die Genauigkeit über die Anzahl der Datenvervielfältigungen visualisiert. Der dabei verwendete Bereich variiert von einer geringen Datenvervielfältigung bis hin zu einer Vervielfältigung von 128.

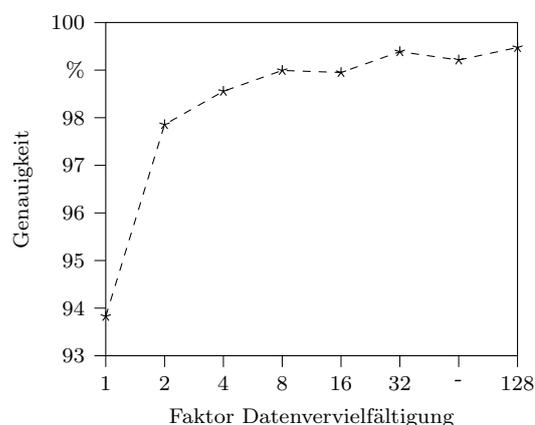


Abbildung 5.27: Einfluss der Größe des Trainingsdatensatzes auf die resultierende Genauigkeit des Validierungsdatensatzes.

Es ist sehr gut zu erkennen, dass zwischen keiner Datenvervielfältigung und einer Vervielfältigung von 2 die größtmögliche Verbesserung der Genauigkeit erreicht wurde. Ab einer Vervielfältigung von 32 ist jedoch nur noch eine minimale Verbesserung zu verzeichnen. Da eine weitere Vergrößerung des Trainingsdatensatzes durch synthetisch hergestellte Bilder die Genauigkeit jedoch nur marginal verbessern würde, sich die Trainingszeit des Netzes erhöht und darüber hinaus ein größerer Speicherbedarf entsteht, ist eine Verwendung einer Datenvervielfältigung mit dem Faktor 32 eine gute Wahl.

5.7.2.2 Variation der Normalisierungstechnik

Ein weiterer wichtiger Parameter der Vorverarbeitung der Daten ist, wie in Abschnitt 5.4 beschrieben, die Art der verwendeten Normalisierungstechnik. Diese hat großen Einfluss auf die Daten und somit auf den Eingang des CNN und letztendlich auf den Trainingsprozess des verwendeten Modells. Aufgrund dessen wird in Abbildung 5.28 der Verlauf der Kosten über die Epochen für drei gebräuchliche Normalisierungstechniken dargestellt.

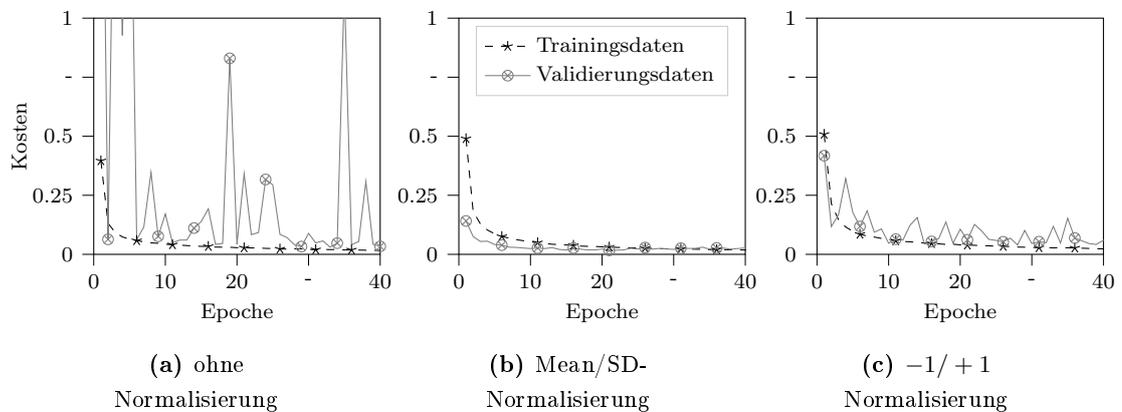


Abbildung 5.28: Verlauf der Kosten der Trainingsdaten und der Validierungsdaten während des Trainingsprozesses des CNN für unterschiedliche Normalisierungstechniken.

Abbildung 5.28a visualisiert den Verlauf der Kosten des Trainingsprozesses des CNN ohne die Verwendung einer Normalisierungstechnik. Abbildung 5.28b veranschaulicht den Trainingsverlauf der Kosten durch eine Normalisierung abhängig vom Mittelwert und der Standardabweichung, wie in Gleichung 5.1 definiert. Des Weiteren bildet die Grafik 5.28c den Trainingsverlauf unter der Verwendung einer Normalisierung der Werte zwischen -1 und $+1$ ab.

Wie deutlich in Abbildung 5.28a zu erkennen, ist der Trainingsprozess ohne eine Normalisierung der Daten sehr instabil, da dieser im Verlauf starke Schwankungen aufweist. Darüber hinaus ist die damit erzielte Genauigkeit geringer im Vergleich zu den zwei weiteren Normalisierungstechniken. Bereits eine Normalisierung der Daten in einem Bereich zwischen -1 bis $+1$ reduziert diese Schwankungen deutlich (vgl. Abbildung 5.28c). Durch die Normalisierung der Pixel basierend auf dem Mittelwert und der

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

Standardabweichung können die Schwankungen nahezu eliminiert werden. Aufgrund dessen wurde diese Art für die Normalisierung für den in dieser Arbeit verwendeten Trainingsprozess angewandt.

5.7.2.3 Variation der Dropout-Rate

Als letzter Parameter wurde die Dropout-Rate variiert. Wie bereits beschrieben, ist sie für die Generalisierung des Netzes notwendig. Neben dem Dropout, welcher wie in Abbildung 5.20 nach der vollkommen vernetzte Schicht angeordnet ist, wurde ein Spatial Dropout verwendet. Diese Art des Dropouts unterscheidet sich darin, dass nach jedem Convolutional-Block eine zusätzlich Spatial Dropout-Schicht hinzugefügt wurde. Diese Schicht wurde speziell für ein CNN konzeptioniert [209, S. 3]. In Abbildung 5.29 ist die Genauigkeit der beiden Dropout-Arten über die Dropout-Rate visualisiert.

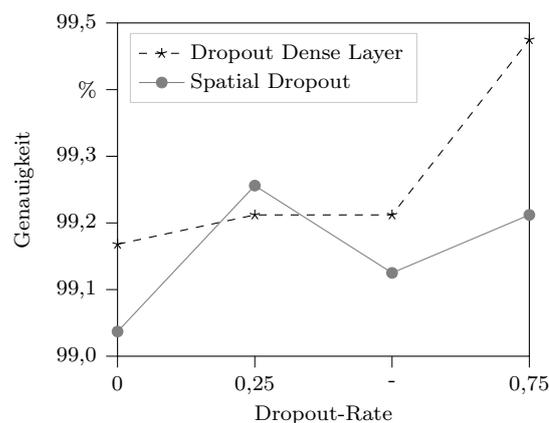


Abbildung 5.29: Einfluss der Dropout-Rate auf die resultierende Genauigkeit des Validierungsdatensatzes.

Wie in Abbildung 5.29 dargestellt, wurde die höchste Genauigkeit mithilfe des Spatial Dropouts bei einer Dropout-Rate von 0,25 erreicht. Im Vergleich zum konventionellen Dropout ist jedoch keine Verbesserung des Ergebnisses zu erkennen. Aus diesem Grund wurde diese Art des Dropouts verworfen. Des Weiteren gibt es eine minimale Verbesserung der Genauigkeit bei der Verwendung eines Dropouts mit einem Wert von 75%. Wie jedoch in Abbildung 5.21 dargestellt, ist der Trainingsverlauf des entwickelten Netzes nahezu optimal. Da kein Overfitting und somit eine Generalisierung des Netzwerkes

gegeben ist und die Verwendung eines größeren Dropouts zu einem Overfitting führen kann, ist die Wahl der Dropout-Rate von 50% sehr gut.

5.8 Evaluierung des finalen Modells

Wie in Abschnitt 5.1.7 erläutert, wird das CNN final an einem gesonderten Testdatensatz antrainiert und überprüft. Für das Antrainieren des finalen Modells werden die Trainings- und Validierungsdaten vereint und das in Abbildung 5.20 dargestellte CNN unter den gleichen Trainingsbedingungen antrainiert (vgl. Abschnitt 5.5.1). Dafür muss eine passende Stichprobengröße des Testdatensatzes definiert werden. Wichtig dabei ist, dass für die Evaluation eine neue Charge an Daten verwendet wird. Die Stichprobengröße wird anhand des in Abschnitt 5.1.7 definierten statistischen Verfahrens bestimmt. Wie in Abschnitt 5.1.7 erläutert, lässt sich die Fehlklassifizierungsrate p_{iQ} des Modells mit der geringeren Fehlklassifizierungsrate (interne Qualität) anhand der in Tabelle 5.3 bestimmten *Genauigkeit* berechnen. Mithilfe der Gleichung 5.8 kann die Fehlklassifizierungsrate p_{iQ} bestimmt werden.

$$p_{iQ} = (1 - acc_{gesamt}) = (1 - 0,9956) = 0,0044 \quad (5.8)$$

Dabei soll der Klassifikationsalgorithmus, welcher die berechnete Fehlklassifizierungsrate p_{iQ} erfüllt, bei einem Vertrauensniveau von mindestens 80% angenommen werden [81].

Für die Bestimmung des zulässigen Höchstwertes der Fehlklassifizierungsrate des Modells mit der höheren Fehlklassifizierungsrate (externe Qualität) wird, wie in Gleichung 2.18 definiert, der Anteil fehlerhafter Verschweißungen und der maximal erlaubte Fehlerschlupf benötigt. Die Anzahl fehlerhafter Verschweißungen wurde durch einen Experten festgelegt und beträgt im betrachteten Anwendungsfall 2175 *ppm*. Der maximale Fehlerschlupf kann mithilfe des Prozessfähigkeitsindex bestimmt werden. Wie in Abschnitt 5.1.7 beschrieben, beträgt dieser im Produktionsumfeld $p_{max_defekt} = 64$ *ppm*. Somit ergibt sich als maximal akzeptabler Fehlerschlupf des Klassifizierungsalgorithmus:

$$p_{eQ} = \frac{p_{max_defekt}}{p_{defekt}} = \frac{64}{2175} = 0,02943 \quad (5.9)$$

5. Modellierung eines CNN zur Erkennung von Prozessabweichungen

Der Klassifikationsalgorithmus mit dieser Fehlklassifizierungsrate soll wie beschrieben mit einer Wahrscheinlichkeit zu 95% (Teststärke/Power) richtigerweise abgelehnt werden. Durch iteratives Erhöhen der Stichprobengröße und die Berechnung des dazugehörigen Vertrauensniveaus und der Teststärke ergibt sich eine Stichprobengröße von $n = 160$ und eine Anzahl an Fehlklassifizierungen von *Annahmegrenze* = 1. Diese Werte führen zu der in Abbildung 5.30a visualisierten Binomialverteilung für das Modell mit einer geringeren Fehlklassifizierungsrate p_{iQ} (interne Qualität) bzw. mit der Fehlklassifizierungsrate p_{eQ} (externe Qualität) (vgl. Abbildung 5.30b).

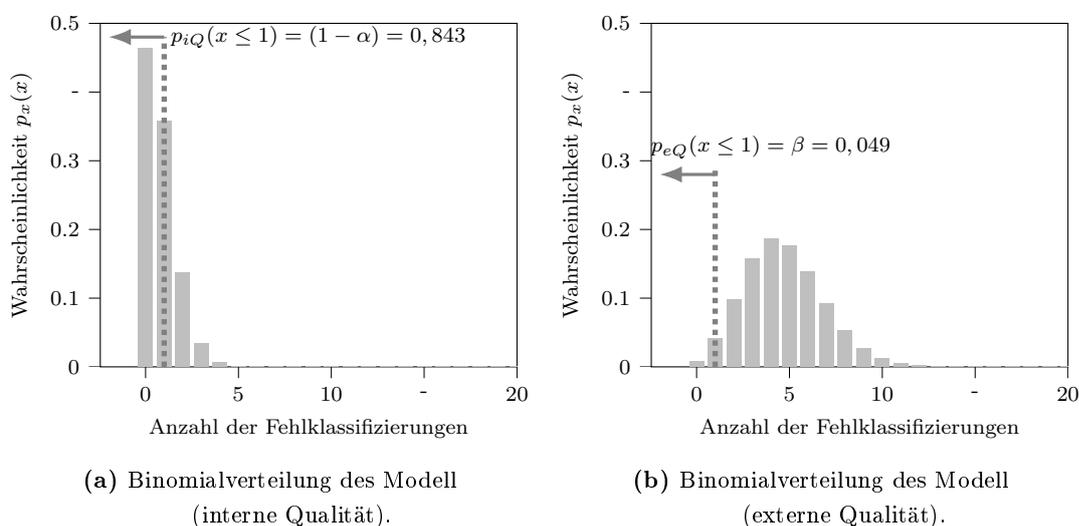


Abbildung 5.30: Binomialverteilung des Modells mit der internen und externen Qualität und der dazugehörigen Fehlklassifizierungsraten und Stichprobengröße $n = 160$.

Wie in Abbildung 5.30a zu erkennen, beträgt das Vertrauensniveau des Klassifikationsalgorithmus mit der Fehlklassifizierungsrate p_{iQ} $(1 - \alpha) = 0,843$ und die Teststärke des Klassifikationsalgorithmus mit der Fehlklassifizierungsrate p_{eQ} $(1 - \beta) = 0,951$. Bei diesen Werten sind die oben definierten Anforderungen erfüllt. Mithilfe der ermittelten Stichprobengröße $n = 160$ und der maximalen Anzahl an falsch klassifizierten Verschweißungen *Annahmegrenze* = 1 kann das CNN final überprüft werden. Die Daten für die finale Evaluierung des CNN wurden in einer neuen und unabhängigen Charge generiert und digitalisiert. Wie oben beschrieben, beträgt die Größe des Stichprobensatzes $n = 160$. Diese Anzahl wird auf die vier in Abschnitt 5.3 definierten Fehlerklassen aufgeteilt, sodass für jede Klasse und deren Abstufung insgesamt 40 Aufnahmen vorhanden

sind. Dieser Testdatensatz wurde ebenfalls von einem Experten überprüft, sodass die Zuordnung der jeweiligen Klassen stimmig ist.

Die Auswertung des Testdatensatzes des finalen CNN aus Abbildung 5.20 ist in Abbildung 5.31 dargestellt. Es ist zu erkennen, dass sich alle Einträge der CM auf der Diagonalen befinden. Dies bedeutet, dass der komplette Testdatensatz durch das CNN korrekt klassifiziert wurde. Darüber hinaus gab es kein UEs, da die Wahrscheinlichkeit des CNN immer größer als 98,5% war.

		vorhergesagte Klasse				
		KV	UV	SS	SK	UE
echte Klasse	Korrekte Verschweißung (KV)	40	0	0	0	0
	Unzureichende Verschweißung (UV)	0	40	0	0	0
	Schweißspritzer (SS)	0	0	40	0	0
	Schweißkrater (SK)	0	0	0	40	0

Abbildung 5.31: Resultierende CM des Testdatensatzes.

Aus der CM ergeben sich die in Tabelle 5.8 gelisteten Metriken. Da, wie in Abbildung 5.31 zu erkennen, das CNN alle vier Klassen korrekt prädiziert, entsprechen alle vier Metriken, welche in Abschnitt 2.4.6.3 definiert wurden, 100 Prozent. Somit wurde die Annahmegränze beim Testdatensatz nicht überschritten. Diese Ergebnisse bestätigen, dass das CNN mit einer Genauigkeit von 99,56% den Anforderungen entspricht und angenommen werden kann.

Tabelle 5.8: Resultierende Ergebnisse basierend auf der CM in Abbildung 5.31 bei Klassifizierung des finalen Testdatensatzes.

Gesamtgenauigkeit	Durchschnittliche Precision	Durchschnittlicher Recall	Durchschnittlicher F_1 -Score
100,0%	100,0%	100,0%	100,0%

5.9 Handlungsempfehlung zur Nacharbeit durch präskriptive Analyse

Mithilfe des entwickelten und validierten CNN können zuverlässig Qualitätsabweichungen in der laufenden Serienproduktion detektiert und klassifiziert werden. Wie in Abschnitt 2.3.2 beschrieben, kann in Abhängigkeit der Qualitätsabweichungen eine Handlungsempfehlung generiert werden. In Abbildung 5.32 ist das dafür entwickelte schematische Vorgehen visualisiert.

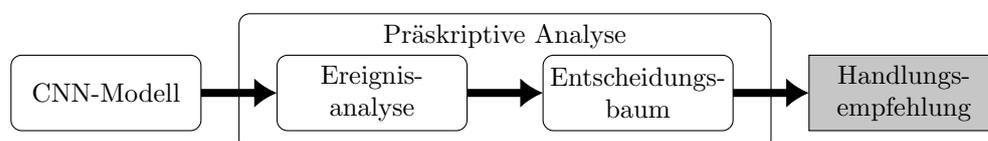


Abbildung 5.32: Schematisches Vorgehen zur Generierung der Handlungsempfehlung mithilfe der präskriptiven Analyse.

Wie zu erkennen, ist der Eingang der präskriptiven Analyse die Ausgabe des CNN. Im ersten Schritt wird eine Ereignisanalyse durchgeführt. Wie bereits beschrieben, geben die Ausgänge des CNN die Wahrscheinlichkeiten wieder, zu welcher Klasse die Verschweißung gehört. Im Modul Ereignisanalyse werden diese Wahrscheinlichkeiten mit dem in Abschnitt 5.6 eingeführten Schwellwert verglichen und bei Unterschreitung dieses Wertes die Verschweißungen als UE klassifiziert.

Im zweiten Schritt wird eine Handlungsempfehlung mithilfe eines Entscheidungsbaumes generiert. Zur einfachen Verknüpfung zwischen dem Ereignis und der Handlungsempfehlung werden bei einer präskriptiven Analyse in der Regel Entscheidungsbäume verwendet. In Abbildung 5.33 ist der hierbei modellierte Entscheidungsbaum dargestellt. Die Handlungsempfehlungen wurden basierend auf Expertenwissen und Versuchen ausgearbeitet. Wichtig bei der Gestaltung des Nacharbeitskonzeptes ist es, dass die Nacharbeit einfach und mit den verfügbaren Mitteln durchgeführt werden kann.

Wird ein Hairpin-Paar korrekt miteinander verschweißt und die Klasse KV durch das CNN prädiziert, sind keine Nacharbeiten oder Korrekturen vorzunehmen. Bei der Erkennung und Klassifizierung eines SS und eines SK kann die Verschweißung automatisch nachgearbeitet werden. Experimentell wurde zur Nacharbeit eine reduzierte

5.9 Handlungsempfehlung zur Nacharbeit durch präskriptive Analytik

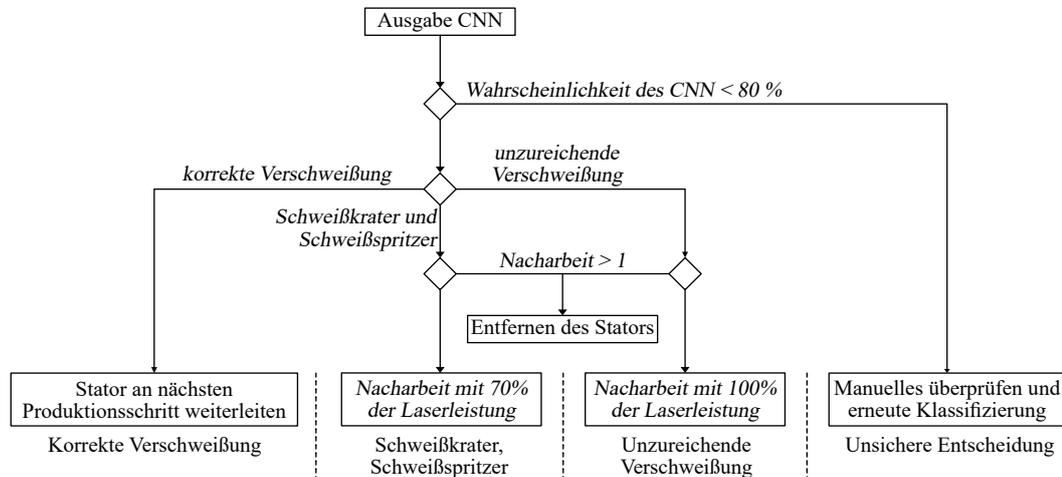


Abbildung 5.33: Entscheidungsbaum mit den zugehörigen Nacharbeiten und Ereignissen.

Laserleistung (70%) als ideal ermittelt. Bei einer Nacharbeit mit gleicher Laserleistung ist der erneute Energieeintrag zu hoch, wodurch die Isolation des Hairpins beschädigt werden kann. Wie bereits beschrieben, wird in der Ereignisanalyse geprüft, ob die Wahrscheinlichkeit des CNN unter den in Abschnitt 5.6 definierten Schwellwert von 80% fällt. Ist dies der Fall, wird die Verschweißung als UE klassifiziert. Tritt dies auf, muss der Stator automatisiert aus dem Produktionsprozess ausgeschleust und manuell vom Personal überprüft werden. Tritt während des Produktionsprozesses eine UV auf, kann das Hairpin-Paar mit voller Laserleistung nachgearbeitet werden. Außerdem ist darauf zu achten, dass keine Verschweißung häufiger als einmal nachgearbeitet wird.

Durch die frühzeitige Erkennung einer Qualitätsabweichung und anschließende automatisierte Umsetzung dieser Handlungsempfehlung, wie beispielsweise eine Nacharbeit, kann der in dieser Arbeit betrachtete Produktionsprozess deutlich optimiert werden. Zudem lässt sich dieses Vorgehen aufgrund des modularen Aufbaus auf andere Prozesse einfach übertragen. Für die Implementation des in diesem Kapitel beschriebenen CNN und der automatisierten Umsetzung der präskriptiven Analyse mittels eines geschlossenen Regelkreises wird eine dafür angepasste IT-Architektur benötigt. Diese Architektur wird im Folgenden erläutert.

Kapitel 6

Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur

6.1 Ermitteln des groben Systementwurfes

In den folgenden Abschnitten werden der Entwurf und die Umsetzung der in Abschnitt 4.2 definierten Anforderungen erläutert. Dabei werden die wichtigsten Teilsysteme und deren Kommunikationsstruktur beschrieben.

6.1.1 Mechatronisches System als Grundmodell

Wie in Abschnitt 4.3 erläutert, wird ein System benötigt, welches mechanische, elektronische und informationstechnische Teilsysteme aufweist [156]. Das mechatronische System besteht aus vier Teilsystemen (vgl. Abschnitt 4.3): einem Grundsystem, Sensoren, Aktoren und einer Informationsverarbeitung. Außerdem spielt die Umgebung, in welcher sich das mechatronische System befindet, eine entscheidende Rolle. Der Aufbau des mechatronischen Systems bezogen auf die optische Erkennung von Schweißfehlern ist in Abbildung 6.1 dargestellt.

Das Grundsystem entspricht der physikalisch existierenden Komponente. In der Regel handelt es sich dabei um ein mechanisches, elektro-mechatronisches, hydraulisches oder pneumatisches System. Im vorhandenen Anwendungsfall ist dies die Produktionsstation, in welcher die jeweiligen Hairpin-Paare verschweißt werden.

6. Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur

Die Sensoren sind zur Erfassung spezifisch ausgewählter Zustandsgrößen des Basissystems vorgesehen. Die gemessenen Werte werden durch äußere Einflüsse und Ungenauigkeiten des Messsystems verfälscht und spiegeln ein ungenaues Abbild der Realität wider. Im Falle des in dieser Arbeit verwendeten Anwendungsfalls der Hairpin-Verschweißung wird ein in Abschnitt 5.3 beschriebener 3D-Scanner als Sensor eingesetzt. Die Sensorwerte werden an das Informationsverarbeitungssystem übergeben.

Die Informationsverarbeitung stellt den Kern der Datenverarbeitung dar. Ihre Funktion liegt in der Ermittlung der notwendigen Strategien zur Beeinflussung der Zustandsgrößen des Grundsystems. Zudem bietet es Schnittstellen zum Menschen und zu weiteren Kommunikationssystemen. Dieses Teilsystem wird in Abschnitt 6.2 detailliert beschrieben.

Die aus der Informationsverarbeitung bestimmte Strategie wird mithilfe von Aktoren umgesetzt. Diese haben die Möglichkeit, das Grundsystem aktiv zu beeinflussen und damit Zustandsgrößen zu verändern. Dabei wird zwischen einer passiven Beeinflussung oder einer aktiven Beeinflussung der Zustandsgrößen unterschieden. In der vorliegenden Arbeit ist dies die Schweißanlage, durch welche fehlerhafte Verschweißungen nachgearbeitet werden können.

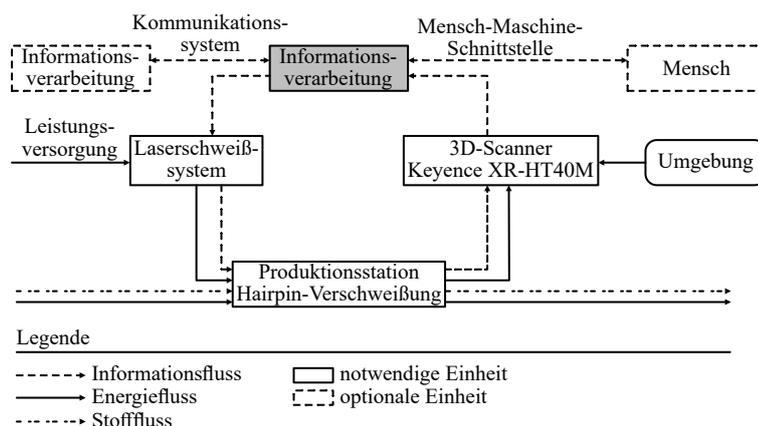


Abbildung 6.1: Grundstruktur eines mechatronischen Systems bezogen auf den Anwendungsfall.

6.1.2 Ableiten einer Systemarchitektur

Das in Abbildung 6.1 dargestellte mechatronische System bildet ein generelles Basismodell, das sich für viele unterschiedliche Anwendungsfälle eignet. Es repräsentiert jedoch nicht alle erforderlichen Voraussetzungen und dient somit als Ausgangsbasis für das im Folgenden entwickelte System. In Abbildung 6.2 ist diese Grundstruktur des mechatronischen Systems nach VDI-Richtlinie 2206 mit der implementierten Edge-/ Cloud-Architektur visualisiert. Wie zu erkennen, ist das mechatronische System in die drei Teilsysteme Produktion, CPS und Cloud unterteilt.

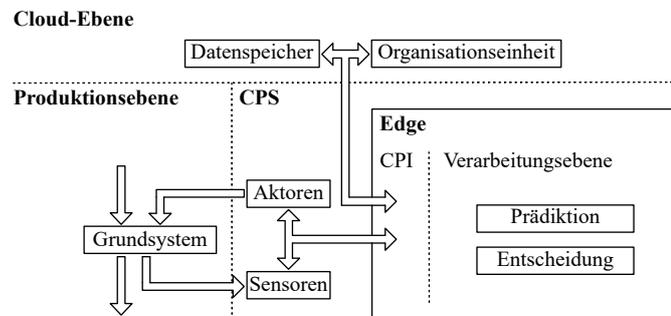


Abbildung 6.2: Grundstruktur des mechatronischen Systems nach VDI-Richtlinie 2206 mit implementierter Edge-/Cloud-Architektur.

Die Produktionsebene beinhaltet das Grundsystem und dessen Ein- und Ausgänge. Auf dieser Ebene werden die Produkte, wie in Abschnitt 2.2 beschrieben, verarbeitet und transformiert. Durch den Aktor werden die jeweiligen Hairpin-Paare miteinander verschweißt.

Das CPS bildet die Ausführungsebene. Auf dieser werden Daten über Sensoren erfasst und in einer Recheneinheit verarbeitet. Anschließend wird eine Transformation in Abhängigkeit des detektieren Fehlerfalls des jeweiligen Produkts über Aktoren durchgeführt. Die Recheneinheit ist die Hauptkomponente des CPS. Da diese, wie in Abschnitt 4.2 erläutert, harte Echtzeitanforderungen besitzt, muss die Recheneinheit in physischer Nähe zur Produktionsebene verortet sein. Aufgrund dessen ist sie als Edge-Gerät modelliert und entspricht damit den Anforderungen einer In-Situ-Lösung. Das Edge-Gerät umfasst die notwendigen Funktionsbausteine zur Verarbeitung der durch

6. Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur

Sensoren erzeugten Daten. Im weiteren Verlauf der Arbeit werden diese Funktionsbausteine als Module bezeichnet. Eine zweite Anforderung ist ein modularer Aufbau des Systems (vgl. Abschnitt 4.2). Dank einer modularen Softwarearchitektur lassen sich die Aufgaben in einzelne Module aufteilen, welche in Abschnitt 6.2 erläutert werden. Der modulare Aufbau bietet den Vorteil, dass die Modifikation und Bereitstellung der einzelnen Module vereinfacht wird, da nicht das gesamte System, sondern nur einzelne Bausteine ersetzt werden müssen. Ein möglicher Nachteil dieses modularen Aufbaus könnte der vermehrte Kommunikationsaufwand sein, der aber, wie in Abschnitt 7.4 beschrieben, zu keinem praktischen Nachteil führt. Die Module des Edge-Gerätes sind in zwei Schichten unterteilt, die Verarbeitungsebene (siehe Abschnitt 6.2.2) und das cyber-physische Interface (CPI) (siehe Abschnitt 6.2.3). Neben der Kommunikation mit den Sensoren und Aktoren beinhaltet das CPI die Kommunikationsmodule zur Cloud-Ebene. Die Verarbeitungsschicht handhabt die tatsächliche Datenverarbeitung. Die beiden Schichten sind in Abbildung 6.2 dargestellt.

Das CPI stellt, wie in Abbildung 6.2 veranschaulicht, die zentrale Kommunikationsschicht innerhalb des CPS dar. Es verbindet zum einen die Module der Verarbeitungsschicht mit den physischen Elementen, zum anderen ermöglicht es gleichzeitig die Kommunikation mit externen Komponenten wie beispielsweise Cloud-Modulen oder anderen CPS. Die auf dieser Schicht verankerten Module sind ausschließlich für die Datenweiterleitung konzipiert. Somit ist beispielsweise ein Modul für Kommunikation mit den in der Cloud befindlichen Datenbanken und ein anderes Modul für die Kommunikation mit einer SPS über das Protokoll Open Platform Communications United Architecture (OPC UA) verantwortlich. Eine genaue Beschreibung der Module ist in Abschnitt 6.2.3 erläutert. Diese Vorgehensweise hat den Vorteil einer klar festgelegten Separation zwischen Kommunikation und Verarbeitung. Dadurch kann die Anzahl komplexer Schnittstellen reduziert und die Wartungsfreundlichkeit gestärkt werden.

Die Verarbeitungsschicht ist verantwortlich für die Prozessierung dieser Kommunikationsströme. Eine detaillierte Erläuterung der in der Verarbeitungsschicht lokalisierten Module findet sich in Abschnitt 6.2.2. In dieser Schicht werden die Eingangsdaten verarbeitet, ausgewertet oder modifiziert. Die individuellen Verarbeitungsschritte sind als unabhängige Module implementiert. Sie agieren voneinander getrennt und wandeln die

Daten, die über definierte Ein- und Ausgänge weitergeleitet werden, um. Die dafür notwendige Kommunikation der Module untereinander findet bilateral statt und wird in Abschnitt 6.2.1 erläutert. Jedes Modul nimmt eine spezifische Aufgabe wahr und leitet die Ergebnisse an die entsprechenden Empfänger weiter. Auf diese Weise können Aufgaben parallelisiert werden. Im Falle einer kontinuierlichen Produktion kann bei der Bestimmung des weiteren Vorgehens zeitgleich ein Folgeprodukt klassifiziert werden.

Die übergeordnete Ebene ist die Cloud-Ebene. Sie fungiert als Datenspeicher, Industrial Internet of Things (IIoT)-Verwaltungsplattform und Recheneinheit. Wie in Abschnitt 4.2 beschrieben, wird eine zentrale Verwaltung der Edge-Geräte und Speicherung von Daten benötigt. Mittels Software für das zentrale Management mehrerer CPS kann eine unbegrenzte Anzahl von Edge-Geräten verwaltet, modifiziert und überwacht werden. Dies bietet den Vorteil eines schnellen Einsatzes und einer flexiblen Zuteilung von Produktionseinheiten. Die in der Cloud verorteten Module sind in Abschnitt 6.3 beschrieben.

6.2 Komponenten des Edge-Geräts

Wie beschrieben bildet die Prädiktion von Qualitätsabweichungen im laufenden Produktionsprozess die Basis des geforderten Systems. Neben der Erkennung und Klassifizierung der in Abschnitt 5.2 erläuterten Fehler bietet eine direkte Umsetzung der Nacharbeit einen deutlichen Mehrwert. Dabei muss sichergestellt werden, dass die in Abschnitt 4.2 definierten Echtzeitanforderungen erfüllt werden. Für die Umsetzung der gewünschten Anforderungen werden unterschiedliche Funktionen gefordert. Diese Module/Funktionen werden auf dem eben eingeführten Edge-Gerät ausgeführt. Die Module werden in Kommunikationsmodule, welche im CPI lokalisiert sind, verarbeitende Module sowie prädiktive und präskriptive Module aufgeteilt. Grafisch ist diese Aufteilung in Abbildung 6.3 visualisiert. Das Konzept der dazugehörigen Module wird im Folgenden vorgestellt.

6.2.1 Kommunikation der auf dem Edge-Gerät lokalisierten Module

Die simultane Datenverarbeitung und -versendung in den unterschiedlichen Modulen setzt eine globale Kommunikation zwischen den entsprechenden Modulen voraus. Im

6. Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur

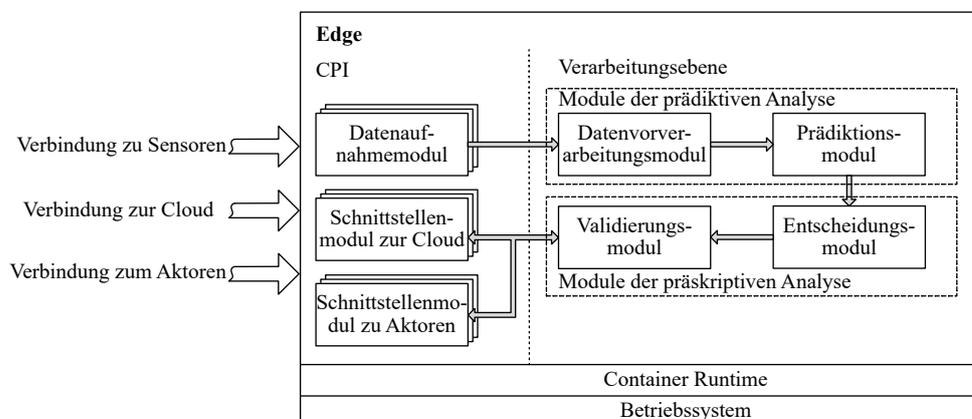


Abbildung 6.3: Konzeptionelle Darstellung der Komponenten der Edge-Ebene und ihrer Verbindungen.

Vergleich zur externen Kommunikation über das CPI kommunizieren die Module mithilfe einer bidirektionalen Nachrichtenübermittlung. Wie in Abschnitt 4.2 beschrieben, müssen standardisierte Kommunikationsprotokolle verwendet werden, weswegen MQTT verwendet wird. Wie in Abschnitt 2.5.2.3 erläutert, können mittels des Publish-Subscribe-Mechanismus Module Informationen und Daten auf einem Topic veröffentlichen und dieses abonnieren. Dies hat den Vorteil, dass die Daten nur dort ankommen, wo sie tatsächlich verarbeitet werden sollen und kein weiterer Overhead durch Vermittlung der Kommunikationspartner verursacht wird. Beispielhaft sind dafür in Abbildung 6.4 das Vorverarbeitungs-, Prädiktions- und Entscheidungsmodul dargestellt. Wie zu erkennen, verarbeitet der IoT-Edge-Hub die empfangenen Daten und leitet diese über definierte Routen weiter.

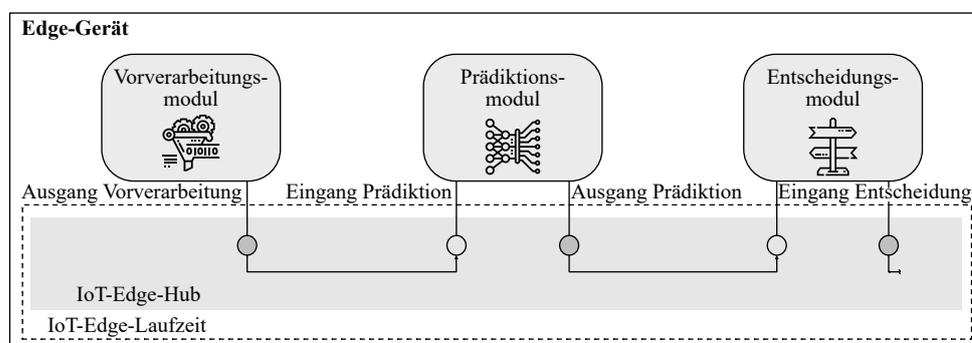


Abbildung 6.4: Datenaustausch von Modulen über den IoT-Edge-Hub.

6.2.2 Module der Verarbeitungsebene

Wie beschrieben und in Abbildung 6.3 dargestellt, sind die Module der Verarbeitungsebene in die Bereiche der prädiktiven und präskriptiven Analyse aufgeteilt. Zu der prädiktiven Analyse gehören das Datenvorverarbeitungs- und das Prädiktionsmodul. Die präskriptive Analyse umfasst das Entscheidungs- und Validierungsmodul. Das Konzept der jeweiligen Module dieser Ebene wird im Folgenden spezifiziert.

6.2.2.1 Datenvorverarbeitungsmodul

Die Datenvorverarbeitung wird den prädiktiven Modulen zugewiesen und ist das erste Modul in der Verarbeitungsebene. Es bündelt die für die Vorhersage erforderlichen unterschiedlichen Daten und bereitet sie in einem geeigneten Format auf. Es fungiert dabei wie ein Trichter, der eine Vielzahl an Daten sammelt, sie filtert und konvertiert und als geeignetes Paket für die Vorhersage bereitstellt. Dies umfasst häufig Schritte wie Normalisierung, Bereinigung, Zuschneiden oder Verkleinern der Bildgröße. Zur Erleichterung der Erkennung relevanter Merkmale müssen irrelevante Informationen, wie beispielsweise Hintergrundinformationen, entfernt werden. Außerdem ermöglicht eine Reduktion der Bildgröße eine höhere Effizienz. Die Normalisierung der Daten hilft, eine einheitliche Datengrundlage zu schaffen.

6.2.2.2 Prädiktionsmodul

Eine Vorhersage der vorliegenden Fehlerklasse im Produktionsprozess wird mithilfe des Prädiktionsmoduls getroffen. Die dafür nötigen Bilddaten werden vom Vorverarbeitungsmodul direkt über ein Topic an das Prädiktionsmodul gesendet. Es setzt sich immer aus zwei abstrakten Ebenen zusammen, zum einen aus dem eigentlichen maschinellen Lernmodell und zum anderen aus dem eingebetteten Code. Hierbei ist das Modell über eine Representational State Transfer (REST)-API in das Modul eingebunden. Auf diese Weise kann das Modell stets über eine in der Cloud befindliche Komponente aktualisiert werden, ohne dass dafür das gesamte Modul ersetzt werden muss. Die im Modell verwendeten Algorithmen sind oft neuronale Netze, Support-Vector-Machines oder logistische Regression. Da es sich jedoch um eine optische Erkennung von Qualitätsabweichungen handelt, wurde in Kapitel 5 ein passender Ansatz eines CNN vorgestellt.

6. Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur

6.2.2.3 Entscheidungsmodul

Den Ausgangspunkt der präskriptiven Analytik bildet das Entscheidungsmodul. Mittels eines Entscheidungsbaumes wird die zur prädiktierten Fehlerklasse passende Nacharbeitsstrategie bestimmt. Die jeweilige Strategie ist stark vom Prozess abhängig und muss infolgedessen durch einen Experten definiert und vorgegeben werden.

6.2.2.4 Validierungsmodul

Das Validierungsmodul gehört ebenfalls zur präskriptiven Analyse. Durch dieses Modul werden die getroffenen Handlungsempfehlungen in den aktuellen Produktionskontext eingeordnet. Das Validierungsmodul überprüft, ob die ausgewählte Nacharbeitsmaßnahme umgesetzt werden kann. Dafür benötigt das Modul Daten von Sensoren sowie der SPS. Durch eine Abfrage der Sensoren bzw. SPS wird sichergestellt, dass die durch das Entscheidungsmodul getroffene Handlungsempfehlung im aktuellen Produktionskontext umgesetzt werden kann. Beispielsweise kann damit kontrolliert werden, ob die Taktzeit eingehalten werden kann oder die aktuelle Statusmeldungen der SPS abgefragt werden.

6.2.3 Module der cyber-physischen Schnittstelle

Wie in Abschnitt 6.1.2 erläutert, dient das CPI als Kommunikationsschnittstelle. Die in diesem Bereich lokalisierten Module werden in den folgenden Abschnitten beschrieben.

6.2.3.1 Datenaufnahmemodul

Die Datenaufnahmemodule stellen die Schnittstelle zwischen Sensortechnik und Verarbeitung dar. Bedingt durch die Hardware-Nähe wird auf Schnittstellen wie Ethernet, Seriell oder BUS-Systeme zugegriffen. Die gewonnenen Daten werden in ein für Module lesbares Format konvertiert und, wie in Abschnitt 6.2.1 beschrieben, auf ein Topic geschrieben. Aus Gründen der Übersichtlichkeit sollte jeder Sensorwert ein eigenes Topic besitzen, das dem jeweiligen Sensor und Datentyp zugeordnet ist. Beispielsweise sollte je ein Topic für einen 3D-Scan und ein Schwarz-Weiß-Bild vorhanden sein. Darüber hinaus gibt es für jeden Sensor ein separates Modul für die Datenerfassung, wie in Abbildung 6.3 dargestellt. Dadurch kann die Sensortechnologie einfach gewechselt werden

und darüber hinaus ergibt sich eine geordnete Struktur sowie die Möglichkeit, auf die verschiedenen Anschlussoptionen der Sensoren einzugehen.

6.2.3.2 Schnittstellenmodul zu den Aktoren

Die Schnittstelle zu den Aktoren ist ebenfalls Bestandteil der Kommunikationsmodule und befindet sich im CPI. Es bildet die verwendeten Variablen der SPS auf der Edge-Ebene ab. Dies ermöglicht den Verarbeitungsmodulen, wie beispielsweise dem Validierungsmodul, auf Daten der SPS zuzugreifen. Die Kommunikation erfolgt bidirektional, dabei können die auf der SPS definierten Variablen sowohl gelesen als auch geschrieben werden. Analog zur Datenerfassung wird für jede Variable ein gesondertes Topic verwendet. Somit kann über diese Schnittstelle eine Nacharbeit über die SPS-Schnittstelle erfolgen.

6.2.3.3 Schnittstellenmodul zur Cloud

Ist eine Vorhersage der Fehlerklasse durchgeführt, sind alle für eine Handlungsempfehlung erforderlichen Daten verfügbar. Um in Zukunft diese Daten zur weiteren Analyse und Optimierung der Algorithmen verwenden zu können, müssen wesentliche Daten gespeichert und verwaltet werden. Die Speicherschnittstelle verwaltet die Kommunikation zwischen dem Edge-Gerät und den in der Cloud befindlichen Datenbanken. Die Kommunikation mit der Datenbank wird über eine in der Cloud angesiedelte REST-API durchgeführt. Dies führt zu einer Abstraktion der Kommunikation und Datenspeicherung und gibt der Cloud die Möglichkeit, die hinterlegte Datenbank beliebig zu skalieren oder auszuwechseln. Verarbeitungsmodule können über die Speicherschnittstelle auf diese API zugreifen und Daten empfangen oder senden. Bei den Daten handelt es sich häufig um Bilder, Vorhersagen oder Durchlaufzeiten.

6.3 Komponenten der Cloud-Ebene

Auf der Cloud-Ebene sind die Komponenten angeordnet, welche keine harten Echtzeitanforderungen besitzen. Aufgrund dessen dient die Cloud-Instanz, wie in Abbildung 6.5 dargestellt, primär dem Speichern und Verwalten von Daten, dem Verwalten

6. Konzeptionelle Entwicklung einer Cloud-/Edge-Architektur

verschiedener Edge-Geräte als auch der Visualisierung des aktuellen Produktionsprozesses und dessen Informationen. Im Gegensatz zum Edge-Gerät ist die Cloud nicht in der physischen Nähe der Produktionsebene verortet.

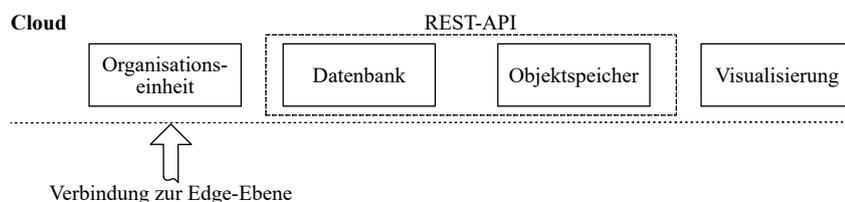


Abbildung 6.5: Konzeptionelle Darstellung der Komponenten der Cloud-Ebene.

6.3.1 Datenspeicher

Da es sich um eine Serienproduktion handelt, fallen im laufenden Produktionsbetrieb viele Daten an. Aufgrund dessen sind große Speicherressourcen nötig, welche einfach skalierbar sind. Aus diesem Grund wird die Speicherung der Daten in der Cloud angestrebt. Aus Gründen der Effizienz sollten die Bilder als Objektdaten gespeichert werden.

Wie in Abschnitt 2.5.2 beschrieben, ist eine starke Vernetzung für eine intelligente Produktion essenziell. Ein Problemfeld stellen hierbei die unterschiedlichen Protokolle verschiedener Produkte dar. Deshalb ist Fachwissen für die unterschiedlichen Plattformen notwendig, wodurch eine einfache Integration von Systemen, wie beispielsweise Datenbanken, erschwert wird. Im Gegensatz dazu sind Web-Technologien ein gut etabliertes Beispiel eines skalierbaren Computernetzes. Denn durch das Web können heterogene Soft- und Hardwareplattformen wie PC, Konsolen oder Mobiltelefone ohne Integrationsprobleme miteinander kommunizieren und arbeiten. Bei aktuellen IoT ist dies derzeit jedoch nicht flächendeckend möglich. Aufgrund dessen ist das sogenannte *Web of Things* Bestandteil aktueller Forschung. Ziel dabei ist es, vorhandene Standards, wie beispielsweise REST, in den Bereich der IoT zu integrieren. Dadurch kann das über Jahre aufgebaute Know-How auf Anwendungen im Bereich IoT übertragen werden. Dies hat den Vorteil, dass bereits etablierte Standards verwendet werden können und sich damit ein System einfacher skalieren und anpassen lässt [210, 211]. Bei Web-Technologien wird die Kommunikation zwischen Client und Server über APIs vollzogen. Aufgrund dessen fungiert das Edge-Gerät als Client und die Structured Query

Language (SQL)-Datenbank als Server, wodurch sich dieses Prinzip gut auf den Anwendungsfall übertragen lässt. Aus diesem Grund wurden um die Datenspeicher eine REST-API gelegt. REST ist ein Standard in Web-Technologien und stellt sicher, dass die Architektur eine Skalierbarkeit, eine Allgemeingültigkeit der Schnittstellen wie auch eine Unabhängigkeit der verwendeten Komponenten unterstützt [212].

6.3.2 Organisationseinheit

Wie in Abschnitt 4.2 beschrieben, ist eine einfache Skalierung und Anpassung der Module der Edge-Geräte erforderlich. Aufgrund dessen wird in der vorliegenden Arbeit die Bereitstellung von Funktionen auf dem jeweiligen Edge-Gerät über die Cloud umgesetzt. Dies wird mittels eines zentralisierten Ansatzes zur Verwaltung der Software implementiert. Mithilfe einer automatisierten Erstellung und Bereitstellung von Funktionalität auf dem Edge-Gerät wird eine einfache Skalierung der Edge-Geräte in der Produktionsebene ermöglicht.

6.3.3 Visualisierungsmodul

Zur Darstellung des aktuellen Prozesses wird ebenfalls eine Human Machine Interface (HMI) entworfen, wodurch Echtzeitdaten direkt veranschaulicht werden können. Mittels der Web-Anwendung kann über Geräte wie Laptops und Smartphones innerhalb des Produktionsnetzwerks von überall auf das System zugegriffen werden. Um Daten direkt über diese Schnittstelle zu aktualisieren, wird zudem eine bidirektionale Kommunikation realisiert. Dafür sind die folgenden drei Funktionen erforderlich, um die Anforderungen der Handlungsempfehlung abzudecken:

1. Darstellung des aktuellen Produktionsprozesses.
2. Korrektur der in Abschnitt 5.6 definierten unsicheren Entscheidung durch das CNN.
3. Manuelles Klassifizieren von Verschweißungen.

Das in Kapitel 5 vorgestellte CNN und die in diesem Kapitel konzeptionierte IT-Architektur werden im folgenden Kapitel miteinander vereint und am Beispiel der Hairpin-Verschweißung validiert.

Kapitel 7

Technische Umsetzung des Verfahrens

7.1 Voraussetzungen der Hard- und Softwarekomponenten

Basierend auf dem Entwurf des CNN in Kapitel 5 und dem Konzept der Cloud-/Edge-Architektur in Kapitel 6 wird in den folgenden Abschnitten das Verfahren zur Klassifikation und Behebung von Qualitätsabweichungen bei der Hairpin-Verschweißung technisch umgesetzt. Dafür werden im Folgenden zunächst die Voraussetzungen der Hard- und Softwarekomponenten vorgestellt. Die Grundlagen hierzu wurden in Abschnitt 2.5.2.3 beschrieben.

- Microsoft Azure: Der in dieser Arbeit verwendete Cloud-Anbieter ist Microsoft Azure. Neben den von Microsoft angebotenen Leistungen können auch selbst entwickelte Produkte genutzt werden. Diese selbst entwickelten Dienste sind die in Abschnitt 6.2 beschriebenen Module.
- Container-Technologie: Um den in Abschnitt 4.2 geforderten modularen Aufbau und die relevante Unabhängigkeit der verwendeten Hardware zu gewährleisten, wird eine weitere Technologie benötigt. Aufgrund dessen wird die sogenannte Container-Technologie verwendet. Dabei werden Funktionen einschließlich deren Anforderungen und Abhängigkeiten in voneinander entkoppelte Module verlagert, um eine geräteunabhängige Laufzeit der Funktionen zu gewährleisten [213].

7. Technische Umsetzung des Verfahrens

- PostgreSQL: Aufgrund der starken Verbreitung dieser Datenbank in der Industrie wird sie als Datenbanksystem in der Cloud verwendet. Zur Speicherung der vorverarbeiteten Bilder wird ein Blob-Speicher genutzt.
- Edge-Gerät: Das in dieser Arbeit verwendete Edge-Gerät soll die entsprechenden Ressourcen für die Erfüllung der in Abschnitt 4.2 beschriebenen Anforderungen verfügen und in der physischen Nähe des Produktionsprozesses genutzt werden. Geeignet hierfür ist ein Industrie-PC, da dieser für die Integration in die industrielle Umgebung ausgelegt ist. Außerdem sollte der verwendete Industrie-PC über mindestens zwei Netzwerkkarten besitzen, um sowohl eine Verbindung an die Produktionsanlage als auch an das Internet gewährleisten zu können. Der in dieser Arbeit verwendete Siemens SIMATIC IPC427E bietet diese Voraussetzungen und ist ein Standard-PC im Produktionsbereich. Mit einem Intel XEON E3-1505L (4 x 2,80 GHz, 8 Threads) Prozessor und einem Arbeitsspeicher von 16 GB bietet er zudem genügend Leistung für eine Prädiktion der Qualitätsabweichung [214]. Als Betriebssystem wird eine Linux-Variante verwendet.

7.2 Komponenten des Edge-Gerätes

Wie in Abschnitt 6.2 aufgezeigt, werden für die Umsetzung der Anforderungen die folgenden Module benötigt, um eine vollumfängliche Nacharbeit garantieren zu können. Dies wird an dem in Abschnitt 5.2 erläuterten Anwendungsfall der Haiprin-Verschweißung umgesetzt. Dabei wird, wie in Abschnitt 6.2 erläutert, zwischen den Modulen der Verarbeitungsebene und des CPI differenziert.

7.2.1 Module der Verarbeitungsebene

Wie in Abbildung 6.2 visualisiert, werden die Module der Verarbeitungsebene in prädiktive und präskriptive Module unterteilt. Die Module der prädiktiven Analyse sind in Datenvorverarbeitungs- und Prädiktionsmodul untergliedert. Wichtig dabei ist, dass es sich ausschließlich um die Prädiktion der vorverarbeiteten Daten handelt. Die zum Fehlerfall passende Handlungsempfehlung wird erst durch die Module der präskriptiven

Analyse generiert. Zu den Modulen der präskriptiven Analyse gehören das Entscheidungsmodul, durch welches die Nacharbeitsstrategie bestimmt wird, und das Validierungsmodul. Bei diesem wird mittels eines weiteren Entscheidungsbaumes die Durchführbarkeit geprüft.

7.2.1.1 Datenvorverarbeitungsmodul

Wie in Abschnitt 5.4 beschrieben, ist die Vorverarbeitung der Daten essenziell für eine sichere Klassifizierung durch das CNN. Aufgrund dessen bereitet das Datenvorverarbeitungsmodul die Daten auf. In Abbildung 7.1 ist der Ablauf des Vorverarbeitungsmoduls visualisiert.

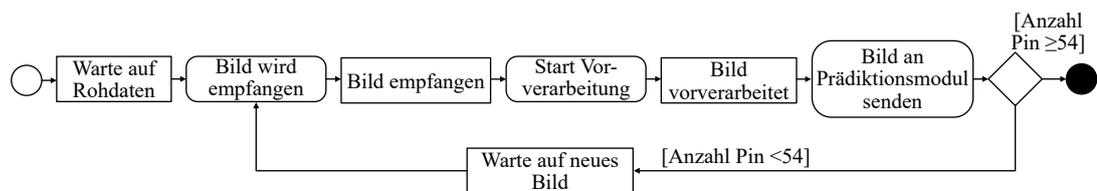


Abbildung 7.1: Ablaufdiagramm des Datenvorverarbeitungsmoduls.

Wie zu erkennen, beobachtet das Datenvorverarbeitungsmodul den Kameratopic, auf welchen das Modul die durch den 3D-Scanner erfassten Daten schreibt (vgl. Abschnitt 7.2.2.1). Bei eingehenden Daten werden diese ausgelesen und analog zu dem in Abschnitt 5.4 erläuterten Vorgehen vorverarbeitet. Nachdem das jeweilige Bild vorverarbeitet wurde, wird dieses direkt an das Prädiktionsmodul geschickt, sodass dieses parallel die Schweißklasse prädiktieren kann. Sind alle 54 Pins des Stators verarbeitet, wird das Vorverarbeitungsmodul beendet und wartet auf neue Daten, welche durch die Produktion des nächsten Stators entstehen.

7.2.1.2 Prädiktionsmodul

Mithilfe des Prädiktionsmoduls wird eine in Abbildung 5.9 dargestellte Fehlerklassen prädiktiert. Die Prädiktion wird durch das in Kapitel 5 definierte und im Vorhinein antrainierte CNN ausgeführt. Dabei ist zu beachten, dass zwischen CNN und dem ausführenden Code eine strikte Trennung existiert. Über einen sogenannten Azure Machine Learning Workspace ist das CNN in einem Docker-Container eingebettet und mit einer

7. Technische Umsetzung des Verfahrens

REST-API umschlossen. Empfängt die REST-API ein vom Prädiktionsmodul versendetes Bild, wird die Wahrscheinlichkeit für jede Fehlerklasse über die REST-API zurückgegeben. Vorteil dieser Separierung ist, dass der ausführende Code und das CNN unabhängig voneinander ersetzt werden können. Der Ablauf des Moduls ist in Abbildung 7.2 dargestellt. Das Prädiktionsmodul wartet auf die vorverarbeiteten Daten. Sobald diese vorhanden sind, empfängt es diese und normalisiert jedes Bild. Im Anschluss wird mittels des in Kapitel 5 vorgestellten CNN die Fehlerklasse prädiziert. Unter anderem wird die Klasse dem Bildnamen hinzugefügt und das fertige Objekt an das Entscheidungsmodul gesendet. Parallel dazu wird das vorverarbeitete Bild an das CPI geschickt, sodass dieses in der Cloud gespeichert werden kann.

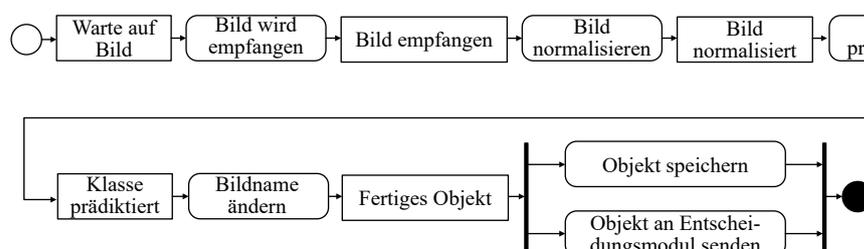


Abbildung 7.2: Ablaufdiagramm des Prädiktionsmoduls.

7.2.1.3 Entscheidungsmodul

Wie in Abschnitt 6.2 erläutert, ist das Entscheidungsmodul der Ausgangspunkt der präskriptiven Analyse. Der Ablauf des Entscheidungsmoduls ist in Abbildung 7.3 dargestellt.

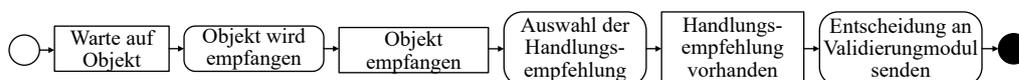


Abbildung 7.3: Ablaufdiagramm des Entscheidungsmoduls.

Das Modul empfängt das Objekt und anschließend wird mittels des in Abbildung 5.33 visualisierten Entscheidungsbaumes eine passende Nacharbeitsmethode gewählt. Zu Beginn wird die ausgegebene Wahrscheinlichkeit des CNN überprüft. Ist sie geringer als 80%, wird das Personal über die in Abschnitt 6.3.3 beschriebene Web-Applikation informiert. Anschließend muss dieser eine manuelle Klassifikation durchführen. Das Auswahl der Entscheidung ist, wie beschrieben, analog zu dem in Abbildung 5.33 illustrierten

Entscheidungsbaum. Die sich daraus ergebende Entscheidung wird in einem Integer kodiert und an das Validierungsmodul übergeben. Die als Integer gespeicherte Variable entspricht dabei der des in Abbildung 5.33 visualisierten Nacharbeitskonzeptes. Der Wert „1“ entspricht zum Beispiel dem Nacharbeitskonzept „Stator an nächsten Produktionsschritt weiterleiten“.

7.2.1.4 Validierungsmodul

Wie in Abbildung 6.2 dargestellt, ist das Validierungsmodul auf der Verarbeitungsebene lokalisiert. Die vom Entscheidungsmodul ermittelte Nacharbeitsstrategie wird in diesem Modul in den aktuellen Produktionskontext gesetzt. Dabei werden die folgenden Schritte ausgeführt (vgl. Abbildung 7.4):

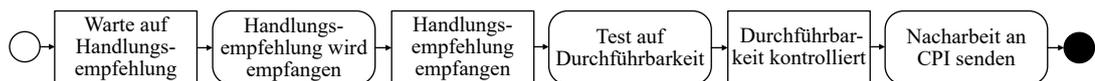


Abbildung 7.4: Ablaufdiagramm des Validierungsmoduls.

1. Das Validierungsmodul wartet zunächst im Ruhezustand, bis eine Nacharbeitsstrategie vom Entscheidungsmodul bestimmt und auf ein Topic geschrieben wurde.
2. Ist dies der Fall, wird im zweiten Schritt kontrolliert, ob eine Nacharbeit durchführbar ist. Dies beinhaltet zum einen die Einhaltung der Taktzeit von 30 Sekunden und zum anderen auch den Status der Produktionsanlage. Beim Status wird geprüft, ob die Produktionsanlage voll funktionsfähig ist oder eine Störung aufweist.
3. Im dritten Schritt wird über die SPS-Schnittstelle ein Integer auf den Wert des zur Fehlerklasse passenden Nacharbeitskonzeptes gesetzt.
4. Ist dies erfolgt, wird die SPS über eine Startvariable informiert, dass die Nacharbeit ausgeführt werden kann.
5. Ist die Nacharbeit durchgeführt und von der SPS bestätigt worden, wird das Validierungsmodul wieder auf den ersten Schritt gesetzt.

7. Technische Umsetzung des Verfahrens

7.2.2 Module der cyber-physischen Schnittstelle

Diese Module befinden sich, wie in Abbildung 6.2 visualisiert, im Bereich des CPI. Sie sind ausschließlich für die interne als auch externe Kommunikation zuständig. Sie sind unterteilt in Datenaufnahmemodul, Schnittstellenmodul zu den Aktoren bzw. der SPS und zur Cloud-Instanz.

7.2.2.1 Datenaufnahmemodul

Die Generierung der für die Prädiktion nötigen Daten findet, wie in Abschnitt 5.3 beschrieben, über eine in der Schweißanlage positionierte 3D-Kamera (Keyence XR-HT40M) statt. Das dabei generierte 3D-Bild wird nach dem in Abschnitt 6.2 erläuterten Konzept des Datenaufnahmemoduls einem dafür vorgesehenen Topic zugeordnet. Auf dem Eingangsbild sind, wie in Abbildung 5.12 zu erkennen, vier verschweißte Hairpin-Paare dargestellt. Der in der Arbeit verwendete 3D-Scanner verfügt über eine File Transfer Protocol (FTP)-Funktion. Dadurch kann die Kamera die generierten 3D-Scans automatisiert über einen FTP-Server auf das Edge-Gerät transferieren und dort speichern. Neben dem 3D-Scan werden Metadaten im Dateinamen gespeichert. Die Metadaten umfassen einen Zeitstempel, die Statornummer, die Pin-Nummer, die Anzahl der durchgeführten Nacharbeiten und die verwendete Laserleistung der Verschweißung. Das Datenaufnahmemodul beobachtet mittels eines Watchers den dafür vorgesehenen Ordner. Liegen neue Dateien vor, werden die Bilder als Byte-Nachrichten in das dafür vorgesehene Topic geschrieben und die Metadaten werden der Nachricht als benutzerdefinierte Eigenschaften beigefügt.

7.2.2.2 Schnittstellenmodul zur SPS

Wie in Abbildung 6.2 zu erkennen, ist das Schnittstellenmodul zwischen SPS und CPS im CPI lokalisiert. Es bildet die geforderten Variablen der SPS auf die von den Modulen der Verarbeitungsebene les- und schreibbaren Topics ab. Die SPS erstellt einen OPC UA-Server, auf den sich das Schnittstellenmodul der SPS einwählt. Beim gegebenen Anwendungsfall werden die Variablen „Start“ und „Nacharbeitsinteger“ auf Topics gemapped und sind somit für alle Module les- und schreibbar.

7.2.2.3 Schnittstellenmodul zur Datenspeicherung

Mithilfe dieses Moduls kann das CPS mit der in der Cloud befindlichen PostgreSQL Datenbank kommunizieren. Wie in Abschnitt 6.3.1 beschrieben, können über eine REST-API Daten sowohl geschrieben als auch gelesen werden. Über die Hypertext Transfer Protocol (HTTP)-Anfrage *POST* werden vom Schnittstellenmodul gewünschte Informationen an die API übermittelt. Die API verarbeitet diese Anfrage und schreibt die Informationen in die Datenbank. Anschließend wird eine Antwort bezüglich des Status des Prozesses rückgemeldet.

7.3 Komponenten der Cloud

Im Folgenden werden die Komponenten erläutert, welche in der Cloud lokalisiert sind. Neben dem Speichern von Prozessdaten soll die Cloud die gesamte Logik der Edge-Geräte zentral verwalten. Vorteil davon ist das Ermöglichen einer einfacheren Skalierung der Anwendung.

7.3.1 Datenspeicher

Informationen des Produktionsprozesses als auch vorverarbeitete Bilder werden in der Cloud gespeichert und archiviert. Basierend auf der PostgreSQL können Tabellen mit hoher Verfügbarkeit auf der dafür vorgesehenen Server-Instanz verwendet werden. Dabei werden folgende Informationen über eine Verschweißung gespeichert: eine eindeutige Kennzeichnung, der Produktionstag, Anzahl des aktuellen Stators bezogen auf die Tagesproduktion, aktuelle Pin-Nummer, Bildname des vorverarbeiteten 3D-Scans, ausgegebene Wahrscheinlichkeit des CNN der prädiktierten Fehlerklasse, die daraus abgeleitete prädiktierte Klasse, das passende Nacharbeitskonzept und diverse Zeitstempel.

In einem Objektspeicher werden zusätzlich zu den oben genannten Informationen die vorverarbeiteten Bilder gespeichert. Dafür wird der von Azure angebotene Blob-Speicher verwendet.

Wie in Abschnitt 6.3 erläutert, wurde eine REST-API um die Datenspeicher gelegt. Mithilfe der HTTP-Anfrage *POST* werden die oben beschriebenen Informationen vom Edge-Gerät an die REST-API gesendet, welche anschließend die benötigten Änderungen

7. Technische Umsetzung des Verfahrens

an der PostgreSQL vornimmt. Um diese Daten für weitere Clients zur Verfügung zu stellen, kann über eine HTTP-Anfrage *GET* auf diese zugegriffen werden. Dadurch kann die in dieser Arbeit konzipierte Architektur einfach um beispielsweise unterschiedliche Datenbanken erweitert werden, ohne dass der Code dezentral auf dem einzelnen Edge-Gerät angepasst werden muss.

7.3.2 IIoT-Verwaltung

Neben der Speicherung von Informationen und Daten dient, wie in Abschnitt 4.2 gefordert, die Cloud als zentrale Verwaltungseinheit. Softwarecodes, wie beispielsweise die Vorverarbeitung oder das CNN, werden in der Cloud-Instanz gehostet und auf dem Edge-Gerät bereitgestellt. Mittels der in Abschnitt 2.5.2.3 erläuterten Pipeline wird der Code automatisiert in ein Container-Abbild gepackt und in eine Speicher-Instanz geladen. Dieser Container wird anschließend auf dem Edge-Gerät bereitgestellt. Vorteil dieses Verfahrens ist, dass Anpassungen von einer zentralen Stelle getätigt werden können. Das Personal muss so keine Anpassungen vor Ort vornehmen, sondern kann diese remote ausführen, was zu einem einfachen und schnellen Prozess führt.

7.3.3 Visualisierungsmodul

Die Web-Applikation besitzt zwei Funktionen und wird über die Cloud gehostet. Wie in den Anforderungen erläutert, ist die Visualisierung wichtig für die Akzeptanz des Systems beim Personal. Außerdem können sich, wie beschrieben, die Produktionsbedingungen und deren Umgebung ändern. Aus diesem Grund muss das verwendete CNN mit neuen Daten antrainiert und an die veränderten Bedingungen angepasst werden. Wichtig dabei ist, dass die Daten und deren Labels richtig sind. Somit ist die zweite Funktion, das Anpassen der Labels in der SQL-Datenbank, zu vereinfachen. Die Anwendung soll auf unterschiedlichen Endgeräten, wie beispielsweise Laptops oder Smartphones genutzt werden. Anhand der Abbildung 7.5 wird die Funktionalität der Web-Applikation beschrieben.

Wie in der Abbildung zu erkennen, wird über ein Ampelsystem der aktuelle Produktionsprozessstatus visualisiert (siehe 1). Dabei bedeutet grün = in Ordnung (iO), gelb = UE, rot = nicht in Ordnung (niO). Darunter befinden sich drei Elemente, welche wichtige Tageskennzahlen hervorheben (siehe 2). Bei der Verschweißung neuer Hairpin-Paare

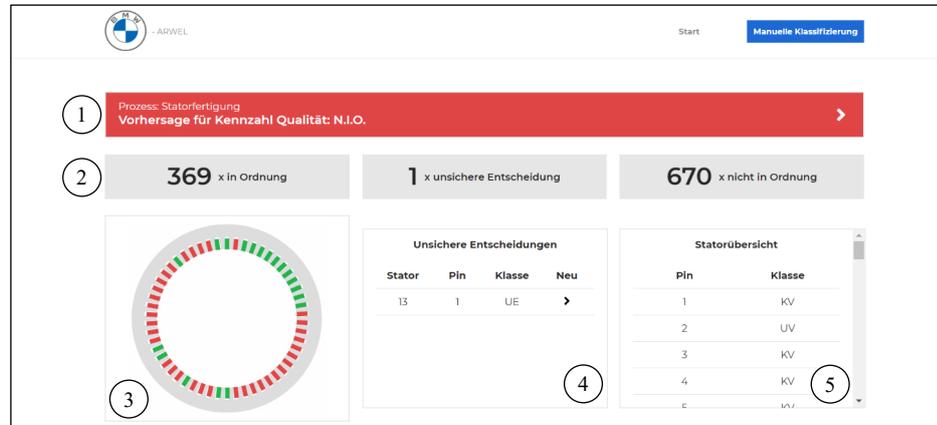


Abbildung 7.5: Startseite der Web-Applikation.

werden die Nutzen der jeweiligen Verschweißung in Abhängigkeit der prädiktierten Klasse eingefärbt. Grün bedeutet eine gute Verschweißung, rot eine fehlerhafte Verschweißung und gelb eine UE (siehe 3). Ebenfalls werden die UE in einer Liste aufgelistet (siehe 4). Dadurch wird dem Anwender eine direkte Schnittstelle über das Schnittstellenmodul zur SQL-Datenbank für eine Neuklassifizierung geboten. Außerdem werden auf der rechten Seite alle Schweißklassen des aktuellen Stators dargestellt (siehe 5).

Falls der Anwender eine durch das CNN definierte Klasse UE ändern möchte, wird er auf die in Abbildung D.1 im Anhang dargestellte Seite weitergeleitet. Auf dieser Seite werden die wichtigsten Informationen visualisiert. Möchte der Anwender eine Neuklassifizierung vornehmen, muss er dies zu Beginn bestätigen.

Im nächsten Schritt muss der Anwender die Klasse iO oder niO auswählen. Wie in Abbildung D.2 zu erkennen, sind die wichtigsten Merkmale der Klasse sowie ein Beispielfeld visualisiert. Dadurch können auch die Anwender eine Neuklassifizierung vornehmen, welche bisher nur eine geringe Expertise besitzen.

Anschließend werden die dazugehörigen Subklassen angezeigt. Die Subklassen UV, SS und SK werden in der Web-Applikation der Klasse niO zugeordnet. Diese sind in Abbildung D.3 im Anhang zu erkennen. Außerdem sind Informationen und ein Beispielfeld dargestellt, um eine Hilfe bei der Klassifizierung zu geben.

7. Technische Umsetzung des Verfahrens

Nachdem der Anwender eine Subklasse gewählt hat, werden alle Informationen der Neuklassifizierung in einer weiteren Seite visualisiert. Zum Überschreiben des alten Labels in der Datenbank muss der Anwender die Auswahl bestätigen. Anschließend wird mittels des Schnittstellenmoduls die HTTP-Anfrage *POST* ausgeführt und somit über die REST-API die Datenbank in der Cloud aktualisiert.

7.4 Ergebnisse

Um das beschriebene System zu bewerten, werden im Folgenden die Durchschnittszeiten der einzelnen Module, die Kommunikationsdauer zwischen den Modulen, die Gesamtverarbeitungszeit eines einzelnen Pins und Stators sowie die eingesparte Zeit aufgrund des modularen Aufbaus analysiert. Die jeweiligen Zeiten sind in Abbildung 7.6 grafisch dargestellt. Zu diesem Zweck wurden Messreihen an zehn Statoren mit je 54 Hairpin-Verschweißungen auf dem verwendeten Industrie-PC (Siemens SIMATIC IPC427D vgl. Abschnitt 7.1) aufgezeichnet.

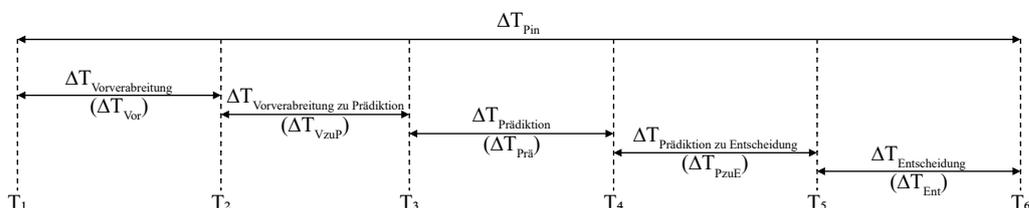


Abbildung 7.6: Zeitmesspunkt der jeweiligen Verarbeitungsschritte.

Die jeweiligen Start- und Endzeiten wurden mit Zeitstempeln in der PostgreSQL-Datenbank gespeichert und anschließend ausgewertet. Die daraus resultierenden Werte sind in Tabelle 7.1 aufgelistet. Die Differenzen der Zeitstempel T_1 bis T_6 sind in Abbildung 7.6 grafisch dargestellt.

Wie in Tabelle 7.1 zu erkennen, ist die Standardabweichung zwischen den zehn verschiedenen Statoren sehr gering. Dies bedeutet, dass die Stabilität des Systems gegeben ist. Vor allem in der Serienproduktion ist ein stabiles System besonders wichtig. Außerdem ist zu erkennen, dass das Vorverarbeitungsmodul mit einer durchschnittlichen Verarbeitungszeit von 0,7796 s pro Pin die meiste Zeit der gesamten Verarbeitungskette

Tabelle 7.1: Statistische Auswertung der Zeitdifferenz der Zeitstempel.

	ΔT_{Stator} (s)	ΔT_{Pin} (s)	ΔT_{Vor} (s)	$\Delta T_{Präd}$ (s)
Mittelwert	21,3501	0,7968	0,7796	0,0033
Standardabweichung	0,0415	0,0083	0,0070	0,0003
Min	21,3059	0,7744	0,7608	0,0026
Max	21,4231	0,8291	0,8092	0,0046

	ΔT_{Ent} (s)	ΔT_{VzuP} (s)	ΔT_{PzuE} (s)	$\Delta T_{gespart}$ (s)
Mittelwert	0,0001	0,0086	0,0052	0,0064
Standardabweichung	0,0000	0,0034	0,0013	0,0042
Min	0,0000	0,0044	0,0036	0,0023
Max	0,0005	0,0158	0,0137	0,0189

in Anspruch nimmt. Jedoch war ein solches Ergebnis zu erwarten, da die 3D-Bilder, wie in Abschnitt 5.4 erläutert, komprimiert und transformiert werden müssen. Es ist aber auch anzumerken, dass somit das größte Optimierungspotenzial bei dem Vorverarbeitungsmodul liegt.

Die Betrachtung der Kommunikationszeiten $\Delta T_{VorverarbeitungzuPrädiktion}$ und $\Delta T_{PrädiktionzuEntscheidung}$ verdeutlicht, dass durch den modularen Aufbau des Systems nur ein geringer zeitlicher Mehraufwand entsteht. Somit ist der in Abschnitt 6.1.2 dargestellte Nachteil aufgrund eines vermehrten Kommunikationsaufwands sehr begrenzt.

Die Verarbeitungszeit eines vollständigen Stators ΔT_{Stator} berechnet sich, wie in Gleichung 7.1 mathematisch dargestellt, aus der Differenz zwischen dem letzten Zeitstempel des letzten Pins und dem ersten Zeitstempel des ersten Pins.

$$\Delta T_{Stator} = T_6^{54} - T_1^1 \quad (7.1)$$

Wie in Tabelle 7.1 dargestellt, ist auch beim gesamten Stator nur eine geringe Standardabweichung zu erkennen. Analog zu den einzelnen Modulen weist dies ebenfalls auf die Stabilität des Systems hin. Die gesamte Verarbeitungszeit des Stators beträgt $T_{Stator} = 21,35$ Sekunden. Zu beachten ist jedoch dabei, dass das Vorverarbeitungsmodul zwei Hairpin-Verschweißungen auf einmal verarbeitet.

7. Technische Umsetzung des Verfahrens

Der Messwert $\Delta T_{gespart}$ gibt die durchschnittliche Zeitersparnis aufgrund der Modularisierung und der daraus resultierenden Parallelisierung der Verarbeitung einzelner Hairpins an. Wie in Gleichung 7.2 beschrieben, wird der erste Zeitstempel (T_1) des $i + 2^{ten}$ Pins vom letzten Zeitstempel (T_6) des i^{ten} Pins subtrahiert. Somit wird beispielsweise der Zeitstempel T_1 des ersten Hairpins eines Stators mit dem Zeitstempel (T_6) des dritten Hairpins verglichen. Die zweite Hairpin-Verschweißung wird übersprungen, da sie aufgrund der Parallelisierung zeitgleich mit der ersten Hairpin-Verschweißung vorverarbeitet wird. Es ist zu berücksichtigen, dass die Kommunikationszeiten zwischen den Modulen in die Berechnung einbezogen wurden und daher $\Delta T_{gespart}$ einen reinen Nettogewinn an Zeit darstellt.

$$\Delta T_{gespart} = T_6^i - T_1^{i+2} \quad (7.2)$$

Wie in Tabelle 7.1 zu erkennen, wird im Durchschnitt für einen Stator eine Verarbeitungszeit von 21,35 Sekunden benötigt. Dies übertrifft die Anforderungen einer Taktzeit von 30 Sekunden erheblich. Somit kann festgehalten werden, dass das System für die gegebenen Anforderungen geeignet ist.

7.5 Anforderungsbezogene Bewertung

In Abschnitt 4.2 wurden Anforderungen an die präskriptive Fertigungsoptimierung aus dem Stand der Technik sowie anhand teilstrukturierter Interviews abgeleitet und in Kapitel 7 am Beispiel des optischen Prüfverfahrens für Hairpin-Verschweißungen umgesetzt. Diese werden nun auf ihren Erfüllungsgrad hin bewertet.

7.5.1 Anforderungsbezogene Bewertung: Allgemeine Anforderungen

Im Handlungsrahmen der vorliegenden Forschungsarbeit wurde ein bildverarbeitendes In-Situ-Prüfsystem (vgl. Tabelle 7.2) entwickelt, wodurch Qualitätsabweichungen in der Produktionslinie erkannt und klassifiziert werden können. Dieses bildverarbeitende Prüfsystem wurde prototypisch entworfen, implementiert und mit realen Produktionsdaten auf der Grundlage des Anwendungsfalles Hairpin-Verschweißung validiert.

7.5 Anforderungsbezogene Bewertung

Tabelle 7.2: Gegenüberstellung der allgemeinen Anforderungen zu den Lösungen.

Anforderung	Lösung
Prüfsysteme	In-Situ-Lösung
Modularer Aufbau	Aufteilung zwischen CPI und Verarbeitungsebene; Publish-Subscribe-Mechanismus
Industrie-Standards	OPC UA und MQTT
Schnittstellen	CPI
Zentrale Verwaltung und Speicherung	Cloud-Instanz (SQL-Datenbank und Blob-Speicher)
Leichte Skalierung und Bereitstellung	Cloud-Instanz und CI/CD

Der modulare Aufbau des CPS erleichtert dessen Anpassung an sich ändernde Randbedingungen bzw. an einen weiteren Anwendungsfall in der industriellen Produktion. Aufgrund dessen wurde das CPS in eine CPI und eine Verarbeitungsebene unterteilt. Wie erläutert, ermöglicht die CPI die Kommunikation mit externen Komponenten, wie beispielsweise Aktoren. Die Kommunikation findet dabei über das Protokoll OPC UA statt, welches als offener Schnittstellenstandard das Edge-Gerät mit der SPS verbindet. Die Verarbeitungsschicht übernimmt die Verarbeitung dieser Kommunikationsströme. Die Module innerhalb der Verarbeitungsebene kommunizieren über einen Publish-Subscribe-Mechanismus und verwenden dabei MQTT als Standardprotokoll. Dieser Mechanismus vereinfacht zudem den modularen Aufbau, da weitere Module einfach integriert werden können, welche auf bestehende Topics schreiben und zugreifen können.

Die Nachverfolgung und Evaluierung der erhobenen Daten wird durch eine zentrale SQL-Datenbank und einen Blob-Speicher in der Cloud sichergestellt. Darüber hinaus dient die Cloud als zentrale Steuereinheit, wodurch Module der CPI und der Verarbeitungsebene verwaltet werden können. Durch den in Abschnitt 2.5.2.3 beschriebenen CI/CD-Prozess können Module auf den Edge-Geräten bereitgestellt werden.

7. Technische Umsetzung des Verfahrens

7.5.2 Anforderungsbezogene Bewertung: Anforderungen an den Klassifikationsalgorithmus

Neben den bereits vorgestellten allgemeinen Anforderungen wurden in Abschnitt 4.2 zusätzlich Anforderungen an den Klassifikationsalgorithmus bestimmt. In Tabelle 7.3 sind diese den in dieser Arbeit vorgestellten Lösungen gegenübergestellt. Diese Gegenüberstellung wird im Folgenden erläutert.

Tabelle 7.3: Gegenüberstellung der Anforderungen des Klassifikationsalgorithmus zu den Lösungen.

Anforderung	Lösung
Klassifikationsalgorithmus	Convolutional Neural Network
Schlanker Aufbau des Klassifikationsalgorithmus	76.220 trainierbare Parameter
Mehrklassen-Klassifizierung	Insgesamt vier Klassen
Kleiner Datensatz	Insgesamt 2.284 Daten
Genauigkeit größer 99 Prozent	Erreichte Genauigkeit: 99,56 Prozent
Menschenlesbarkeit	Grad-CAM- und ISCSV-Verfahren
Präskriptive Analyse	Modellierung durch Entscheidungsbaum

Aufgrund der Vorteile bei der Verarbeitung von Bildern wurde in der vorliegenden Forschungsarbeit ein CNN zur Prädiktion der Fehlerklasse verwendet. Die CNN-Architektur wurde für diesen Anwendungsfall durch die Einführung von Convolution-Blöcken und unter Verwendung der Techniken Batch-Normalization, Global-Average-Pooling und Dropout entwickelt. Dadurch konnte eine erhöhte Genauigkeit, ein stabiler Trainingsprozess sowie eine gute Generalisierung erreicht werden.

Eine weitere Anforderung an das entwickelte Modell ist eine Echtzeitfähigkeit in Bezug auf die in Betracht gezogene Anwendung der Hairpin-Verschweißung. Aus diesem Grund wurde die eben beschriebene Architektur des CNN schlank modelliert. Dadurch besitzt das Netz insgesamt nur 76.220 trainierbare Parameter.

Zudem wird eine Mehrklassen-Klassifizierung gefordert, da die zu generierende Handlungsempfehlung von dem jeweiligen Fehlerfall abhängig ist. Hierbei wurde zwischen vier

Klassen (KV, UV, SS und SK) differenziert. Durch die Anpassung der Ausgänge des CNN kann die Anzahl der Klassen beliebig angepasst werden.

Aufgrund der geringen Verfügbarkeit der Produktionsanlagen ist das Generieren eines großen ausgeglichenen Datensatzes mit hohem Aufwand verbunden. Bei der vorliegenden Arbeit konnten mit einem Trainingsdatensatz von 2.284 Bildern die im Folgenden gewünschten Anforderungen an die Güte des CNN erreicht werden.

Auf Basis der 5-fold Cross-Validation wurde die modellierte Architektur anschließend unter gängigen Bewertungsmetriken validiert. Die dabei erzielte Genauigkeit des CNN beträgt 99,56%. Zusätzlich wurde eine visuelle Validierung mithilfe der Verfahren Grad-CAM und ISCSV durchgeführt. Dadurch konnte gezeigt werden, dass das modellierte CNN relevante Merkmale aus den Bildern extrahieren kann. Darüber hinaus ist dadurch eine Menschenlesbarkeit gegeben.

Zuletzt wurde ein präskriptiver Ansatz umgesetzt. Dabei wurde ein Nacharbeitskonzept in Abhängigkeit des prädiktierten Fehlerfalls unter der Verwendung eines Entscheidungsbaumes modelliert.

7.5.3 Anforderungsbezogene Bewertung: Anforderungen an die IT-Architektur

Der zweite Teil dieser Dissertation befasst sich mit der Entwicklung eines CPS, welches für die automatisierte Umsetzung der generierten Handlungsempfehlung zuständig ist. In Tabelle 7.4 sind die Anforderungen an das CPS den Lösungen gegenübergestellt.

Für eine echtzeitfähige Generierung der Handlungsempfehlung wird, wie erläutert, eine Rechenressource benötigt, welche in der Nähe des Produktionsprozesses lokalisiert ist. Aus diesem Grund bietet Edge-Computing die optimalen Voraussetzungen, um ohne große Latenzzeiten in Echtzeit eine Handlungsempfehlung zu generieren. Durch die Verwendung der Container-Technologie wird sichergestellt, dass für das Edge-Gerät keine bestimmte Hardware notwendig ist, wodurch eine Unabhängigkeit und Interoperabilität der Informationsverarbeitung sichergestellt ist.

7. Technische Umsetzung des Verfahrens

Tabelle 7.4: Gegenüberstellung der Anforderungen der IT-Architektur zu den Lösungen.

Anforderung	Lösung
Rechenressource zur Umsetzung der Handlungsempfehlung	Edge-Gerät
Unabhängigkeit und Interoperabilität	Container-Technologie
Automatisierte Handlungsempfehlung	Schnittstelle zwischen Edge-Gerät und Aktoren
Rechenressource zur kontinuierlichen Optimierung des CNN	Cloud-Instanz
Mensch-Maschinen-Schnittstelle	Web-Applikation
Visualisierung und Kommunikation	Web-Applikation

Durch eine Schnittstelle zwischen dem Edge-Gerät und den Aktoren der Produktionsanlage kann durch einen geschlossenen Regelkreis diese generierte Handlungsempfehlung ohne menschlichen Eingriff umgesetzt werden. Die dabei erreichte Prozesszeit von 21,35 Sekunden kann die für den Anwendungsfall bestimmte Prozesszeit von 30 Sekunden unterbieten.

Ein Problem bei bildverarbeitenden Prüfsystemen ist, dass sich Produktionsumgebungen im Laufe der Zeit ändern können, was die Genauigkeit des CNN negativ beeinflussen kann. Aufgrund dessen muss das CNN an die sich ändernden Bedingungen angepasst werden können. Leistungsstarke Ressourcen der Cloud bieten dafür die optimalen Voraussetzungen. Mit den in der Cloud gespeicherten Daten kann das CNN neu antrainiert und dadurch auf dem Edge-Gerät bereitgestellt werden.

Wichtig dabei ist jedoch, dass die Qualität der dafür verwendeten Daten in der Cloud gegeben ist. Dies bedeutet, dass zu dem jeweiligen Bild das richtige Label vorhanden sein muss. Durch die beschriebene Web-Applikation ist es möglich, die generierten Labels in der Cloud einfach zu ändern. Darüber hinaus werden wichtige Kennzahlen durch die Web-Applikation dem Personal kontinuierlich bereitgestellt.

Die Ergebnisse spiegeln wider, dass mithilfe der präskriptiven Prozessoptimierung Fehler im laufenden Produktionsprozess optisch mithilfe eines CNN erkannt und mittels eines CPS, bestehend aus einer Cloud-/Edge-Architektur, automatisiert nachgearbeitet werden können. Zusammenfassend kann behauptet werden, dass alle Anforderungen an das entwickelte System umgesetzt und die dadurch definierten Ziele erreicht wurden. Es wurde somit gezeigt, dass durch die prädiktive Analyse und durch einen geschlossenen Regelkreis die Optimierung des Produktionsprozesses möglich und umsetzbar ist. Mithilfe dieses Systems konnte die in Abschnitt 5.2 erwähnte Nacharbeitszeit von 18 Minuten auf unter eine Minute pro fehlerhaftem Stator reduziert werden.

Im nachstehenden Kapitel wird die vorliegende Dissertation zusammengefasst sowie einer kritischen Bewertung unterzogen. Den Abschluss des Kapitels bildet ein Ausblick auf weitergehende Forschungsansätze, welche den Rahmen dieser Arbeit erweitern.

Kapitel 8

Schlussbetrachtung

8.1 Zusammenfassung

„Man merkt nie, was schon getan wurde, man sieht nur, was noch zu tun bleibt.“ Marie Curie, Physikerin und Nobelpreisträgerin

Der derzeitige Wandel vom konventionell zum elektrisch angetriebenen Fahrzeug ist zum einen mit einer starken Unsicherheit und zum anderen mit hohem Zeitdruck verbunden. Kombiniert mit dem im Vergleich zur Verbrennertechnologie geringen Expertenwissen stellt dies die Automobilindustrie vor große Herausforderungen. Um diesen Herausforderungen gerecht zu werden, stellen adaptive und kontinuierlich sich verbessernde Produktionsprozesse eine zentrale Rolle für die Wettbewerbsfähigkeit der Automobilindustrie dar. Der vermehrte Einsatz von Sensoren zur Datenerfassung sowie die Digitalisierung des Shopfloors führen zu steigenden Datenmengen, wie beispielsweise Daten der Qualitätsüberwachung, was ein enormes Verbesserungspotenzial der datengetriebenen Optimierung mit sich bringt. Diese Art der Optimierung bietet vor allem in der Produktion des elektrischen Antriebsstranges ein enormes Potenzial, da, wie erläutert, ein breites Expertenwissen in diesem Bereich nur eingeschränkt vorhanden ist. Allerdings nutzt die bestehende Prozessanalytik in der industriellen Praxis die in der Datenmenge steckenden Informationen kaum aus. Ein starker Nachteil vor allem ist, dass keine präskriptiven Analysen genutzt werden, um mittels eines geschlossenen Regelkreises automatisiert eine Handlungsempfehlung umzusetzen.

8. Schlussbetrachtung

Ein möglicher Ansatz zur Datenerfassung und zerstörungsfreien Prüfung ist die optische Überwachung des Produktionsprozesses. Nachteilig an den heute eingesetzten Bildverarbeitungssystemen ist unter anderem der hohe manuelle Parametrierungsaufwand des Bildverarbeitungssystems. Zudem beruht dies auf Erfahrung und erfordert zusätzlich den Einsatz von speziell geschultem Personal, was, wie beschrieben, bei der Produktion von elektrischen Antrieben problematisch ist. Vor diesem Hintergrund wurde im Rahmen der vorliegenden Dissertation ein optisches Prüfsystem entwickelt, das es erlaubt, Fehler im Produktionsprozess zu erkennen, zu klassifizieren und automatisch nachzubearbeiten.

Zu Beginn wurden zu diesem Zweck zum Verständnis der vorliegenden Forschungsarbeit wesentliche Begriffe festgelegt, sowie die Grundlagen des Produktionssystems, dessen datengetriebene Prozessoptimierung sowie die Grundlagen des maschinellen Lernens und der Industrie 4.0 erläutert. Anschließend wurde in Kapitel 3 der aktuelle Stand der Technik recherchiert und eine daraus resultierende Forschungslücke abgeleitet.

Darauf aufbauend wurde in Kapitel 4 das methodische Vorgehen zur Gestaltung des optischen Inspektionssystems beschrieben. Da dieses System aus mechanischen, elektrischen und informationstechnischen Komponenten besteht, wurde die VDI-Richtlinie 2206 als Grundlage für das Vorgehensmodell herangezogen. Darüber hinaus wurde das in dieser Richtlinie beschriebene Grundmodell des mechatronischen Systems für das in dieser Arbeit geforderte System erweitert. Grundlage für die Auslegung des Prüfsystems ist die Formulierung der Anforderungen, welche ebenfalls in Kapitel 4 definiert wurden.

Anschließend wurde in Kapitel 5 der für die Erkennung und Klassifizierung notwendige Algorithmus entworfen. Zu diesem Zweck wurde zunächst ein Vorgehensmodell entworfen, das auf dem CRISP-DM-Modell basiert. Das in dieser Arbeit beschriebene Verfahren unterscheidet sich dadurch, dass es um einen Nachweis der Leistungsfähigkeit eines Bildverarbeitungssystems in einem Prüfprozess (siehe Richtlinie VDI/VDE/VDMA 2632 Blatt 3.1) und einen präskriptiven Ansatz erweitert wurde. Dieses Verfahren wurde daraufhin am Beispiel des Hairpin-Schweißens umgesetzt. Mit dem im Verfahren entwickelten CNN konnte eine Genauigkeit von 99,56 Prozent erreicht werden.

Für eine 100-Prozent-Prüfung durch eine CNN und die automatisierte Einstellung der Prozessparameter ist eine geeignete IT-Architektur erforderlich, die in Kapitel 6 vorgestellt wurde. Da in der Serienproduktion harte Echtzeitbedingungen herrschen, besteht die Architektur aus einer Kombination von Edge- und Cloud-Computing. Der Vorteil der in dieser Arbeit entwickelten Architektur ist der modulare Aufbau des Systems. Dadurch ist gewährleistet, dass das CPS leicht an sich ändernde oder zusätzliche Anwendungen angepasst werden kann.

Schließlich wurden die entwickelte Methode und das Testverfahren in Form eines Prüfstandes am Beispiel des Hairpin-Schweißens in Kapitel 7 technisch umgesetzt. Kernelemente sind das CNN und die in Kapitel 6 definierte IT-Architektur. Die Machbarkeit des entworfenen optischen Testverfahrens wurde dadurch demonstriert. Anhand des Anwendungsfalles konnten sowohl die Erfüllung der Anforderungen als auch der Nutzen des Verfahrens nachgewiesen werden. Durch das Prüfsystem konnte die Nacharbeit von 18 Minuten (vgl. Abschnitt 5.2) auf unter eine Minute pro fehlerhaftem Stator reduziert werden.

8.2 Ausblick

Der weitere Forschungsbedarf resultiert aus der Limitation der vorliegenden Forschungsarbeit. Das Forschungsfeld dieser Dissertation bezieht sich stark auf die optische Erkennung von Qualitätsabweichungen mittels eines CNN am Beispiel der Hairpin-Verschweißung. Dabei werden ausschließlich Daten eines Produktionsschrittes (Schweißstation) verwendet. Jedoch wirken sich mehrere, nichtlineare Effekte der vorgelagerten Prozessschritte ebenfalls auf das Schweißergebnis aus. Ein Beispiel hierfür sind Lackrückstände auf einem Hairpin, welche sich aus der Veränderung des Isolationsmaterials oder einer schwankenden Stärke der Isolationsschicht ergeben. Der vorgestellte Ansatz dieser Forschungsarbeit bezieht sich daher auf einen kleinen Ausschnitt des gesamten Produktionsprozesses, ohne eine ganzheitliche Sichtweise zu verfolgen. Die Reduktion des Einflusses von Material und von vorgelagerten Prozessschritten auf das Ergebnis der Hairpin-Verschweißung wird folglich derzeit nicht betrachtet. Jedoch kann auf wichtige prozesskettenübergreifende Abhängigkeiten der Statorproduktion auf den Schweißprozess mit einer künstlichen intelligenz-basierten Regelung automatisiert reagiert wer-

8. Schlussbetrachtung

den. Auf der Grundlage der in dieser Arbeit gesammelten Erkenntnisse lassen sich die folgenden zwei weiteren Forschungsfragen ableiten:

- Welche Materialabweichungen, vorgelagerten Produktionsschritte oder das Laserschweißverfahren selbst beeinflussen die endgültige Schweißqualität maßgeblich?
- Wie kann eine intelligente Regelung des Laserschweißprozesses mittels künstlicher Intelligenz umgesetzt werden, bei der die Abweichungen der vorgelagerten Prozessschritte durch geeignete Anpassung der Schweißparameter ausgeglichen werden, wodurch eine konstant hohe Verbindungsqualität gewährleistet werden kann?

Wie in Abschnitt 5.2 beschrieben, wird über die Cloud das CNN zur Erkennung der Qualitätsabweichung bei sich ändernden Umgebungsbedingungen neu antrainiert, da sich diese negativ auf das Netz auswirken können. Diese negativen Auswirkungen werden jedoch derzeit nicht automatisiert verfolgt. Darüber hinaus muss im aktuellen Prozess die Güte des CNN manuell durch eine Person kontrolliert werden. Von Vorteil wäre jedoch eine automatisierte Beobachtung des CNN und dessen Güte, die es ermöglicht, automatisch auf veränderte Umgebungsbedingungen zu reagieren und das CNN durch erneutes Antrainieren zu verbessern. Das System sollte in der Lage sein, sich durch die in der Produktion generierten Daten selbst zu optimieren, um stets die bestmögliche Genauigkeit des CNN zu gewährleisten. Hieraus ergeben sich zwei zusätzliche Forschungsfragen:

- Durch welche Metriken kann die Güte des CNN automatisiert beobachtet werden, wodurch zum Beispiel eine Erhöhung von Pseudofehlern automatisiert erkannt wird?
- Wie kann die Güte des erneut antrainierten CNN automatisiert bewertet werden, bevor dieses Modell an der Produktionslinie bereitgestellt wird?

Außerdem wurde eine Parametervariation durchgeführt. Wie in Abschnitt 5.7 beschrieben, wurden dabei die generierten Hairpin-Aufnahmen verwendet. Die dabei erzielten Ergebnisse sollten jedoch an weiteren Datensätzen aus dem industriellen Umfeld überprüft werden. Wichtig dabei ist, dass es sich ebenfalls um Bildaufnahmen handelt. Des Weiteren wurden beim magischen Dreieck drei Dimensionen betrachtet. Das Dreieck könnte jedoch um weitere Dimensionen erweitert werden. Auf Basis dieser Erkenntnis stellen sich zwei zusätzliche Forschungsfragen:

- Wie lassen sich die Ergebnisse der Parametervariation mittels weiterer Datensätze validieren?
- Um welche Dimensionen lässt sich das magische Dreieck erweitern und welchen Einfluss hat der Aufbau des CNN auf diese?

Die Beantwortung der geschilderten, noch offenen Forschungsfragen wie auch die Erkenntnisse der vorliegenden Dissertation, vor allem das entwickelte CNN und CPS, bilden die Killerapplikation für den breiten Einsatz des bildverarbeitenden Prüfsystems in der Produktion des elektrischen Antriebsstrangs und werden der zukünftigen Technologie zum Durchbruch verhelfen.

Anhang

Anhang A

Dokumentationen Experteninterviews

A. Dokumentationen Experteninterviews

Tabelle A.1: Leitfragenkatalog Experteninterviews.

Expertise	Leitfragen
1	<ul style="list-style-type: none"> • Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen? • Welche Steuerungen sollten in der Automobilindustrie verwendet werden? • Wie ist eine dynamische Prozessanpassung umsetzbar? • Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen? • Was ist bei einer zusätzlichen Hardwareimplementierung in die Produktion zu beachten? • Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt? • Sind weitere Komponenten besonders zu empfehlen?
2	<ul style="list-style-type: none"> • Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen? • Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen? • Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen? • Wie sind zusätzliche Systeme (z. B. MES) an das Gesamtsystem anzubinden? • Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt? • Sind weitere Komponenten besonders zu empfehlen?
3	<ul style="list-style-type: none"> • Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen? • Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen? • Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen? • Wie ist die Anbindung der Datenverarbeitung auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen? • Wie sind zusätzliche Systeme (z. B. MES) an das Gesamtsystem anzubinden? • Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt? • Sind weitere Komponenten besonders zu empfehlen?

Tabelle A.2: Dokumentation Experteninterview 1.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
1	24.09.2018	Experte für Steuerungstechnik	Männlich	30
<p>Welche Steuerungen sollten in der Automobilindustrie verwendet werden? Es werden primär SPS der Siemens TIA 1500 Serie verwendet. Die Programmierung ist stark erleichtert, da unternehmensweite Standardbausteine verfügbar sind. So kann beispielsweise die Anbindung eines Lasers so in minimaler Zeit erfolgen. Weitere SPS-Hersteller werden kaum eingesetzt. Daher ist wenig Know-how über die Programmierung und Anbindung vorhanden.</p> <p>Wie ist eine dynamische Prozessanpassung umsetzbar? Der Anlagenlieferant gibt die Variablen über die SPS in den Einstellungen frei. Unterscheidung in lesbar und schreibbar. Durch schreibbare Variablen können in der SPS Parameter angepasst werden. Jedoch können Probleme mit der Gewährleistung der Anlagenlieferanten entstehen, wenn ohne Absprachen ein Schreiben von Variablen erfolgt.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen? Die Anbindung der Feld- und Steuerebene kann über ein Gateway erfolgen. Bisher wurden mit zwei Systemen Erfahrungen gesammelt. Ein Softwaregateway ist funktional herausragend. Es funktioniert sehr gut mit OPC UA und Profinet zusammen. Die Einbindung von OPC UA-Variablen ist intuitiv. Die Einbindung von Profinet-Variablen ist jedoch umständlicher über Hardware-Adressen. Der optimierte Bausteinzugriff der SPS ist bei Verwendung von einem Softwaregateway nicht möglich, wenn Profinet verwendet wird nur bei OPC UA möglich. Die zweite Variante erfolgt über ein Hardware-Gateway, jedoch ist die Einbindung der Variablen relativ umständlich und die Sinnhaftigkeit bei der Verwendung von OPC UA fraglich.</p>				

A. Dokumentationen Experteninterviews

Tabelle A.3: Dokumentation Experteninterview 2.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
2	02.10.2018	Experte für IoT	Männlich	45
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Wichtig ist es, Prozessdaten und weitere Meta-Daten für eine Verarbeitung verfügbar zu machen. Zusätzlich ist ein interoperables System wichtig, das möglichst viele Geräte anbinden kann.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Es ist empfehlenswert, die Geräte in der Produktion über ein zentrales Gateway anzubinden. Dieses kann hierbei in Software oder Hardware umgesetzt werden. Das Gateway ermöglicht den Datentransport aus dem Produktionsnetzwerk nach „oben“, kann aber auch Daten nach „unten“ über Subscription senden. OPC UA sollte als Protokoll unterstützt werden, da sich dies als Standard klar durchgesetzt hat und eine interoperable Kommunikation ermöglicht.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Edge-Computing sollte für die Verarbeitung von Steuerungsdaten als Möglichkeit miteinbezogen werden. Es existieren Lösungen wie die IoT-Edge von Microsoft. Hierdurch können neue Konfigurationen und Software über den IoT-Hub der Cloud an das Edge-Device gesendet werden. Besonders komfortabel geht das über Docker Container. Denn diese sind je nach Anforderung ‚befüllbar‘ und unabhängig ausführbar. Für aufwendige Datenverarbeitungen sollte die Cloud eingesetzt werden, da diese eine hohe und skalierbare Rechenleistung bietet. Die Cloud bietet zusätzlich den großen Vorteil, dass die Daten zentral gebündelt und so einfacher umfassende Analysen durchgeführt werden können.</p> <p>Wie ist die Anbindung der Datenverarbeitung auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Ein IoT-Hub ist die zentrale administrative Stelle für den Datentransport zur Modellaktualisierung. Dieser IoT-Hub bekommt Daten über ein Gateway im Produktionsnetz. Er speichert zudem die Daten für sieben Tage zwischen. Der IoT-Hub ist auf Werke aufgeteilt und somit getrennt. Ziel ist jedoch eine möglichst geringe Trennung, da die Aufteilung teurer ist und gesamtheitliche Analysen schwieriger sind. Als Protokolle sind MQTT und HTTP für einen bidirektionalen Datenverkehr empfehlenswert.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt?</p> <p>Aktuell sind zwei Projekte bekannt, die Data Analytics für eine aktive Reaktion einsetzen wollen: eine Montagehilfe für die Produktion, die einem Mitarbeiter in der Produktion durch ein Entscheidungsunterstützungssystem bei der Arbeit hilft. Das zweite Projekt ist eine dynamische Platinen-Beölung im Presswerk. Hierdurch soll eine proaktive Steuerung von Maschinen implementiert werden. Dieses Projekt ist jedoch noch in einer frühen Projektphase und nur teilweise implementiert (vgl. Experteninterview 3).</p>				

Tabelle A.4: Dokumentation Experteninterview 3.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
3	08.10.2018	Experte für IoT	Männlich	60
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Es müssen verschiedenste Meta-Daten und Prozessparameter gesammelt werden, um den Produktionsprozess verstehen zu können. Eine Steuerung ist jedoch nur über strukturierte Daten sinnvoll. Unstrukturierte Daten sollten für die Modellerstellung verwendet werden. Wichtig ist, dass die Architektur Feldgeräte über Setting-Daten konfigurieren kann, ohne dass ein Mensch manuell tätig werden muss. Die steuernde Einheit muss mit mehreren Netzwerken und Protokollen für Sensoren, Aktoren und Verarbeitungseinheiten interagieren können. Über entsprechende Protokolle ist die Sicherheit des Datentransportes sicherzustellen und es ist darauf zu achten, ob es sinnvoll ist, Daten bidirektional oder unidirektional zu übertragen. Eine Datenvorverarbeitung wird benötigt, um die Datenströme der Geräte zu reduzieren, z. B. durch Transformation, Aggregation und Filterung. Für die Datenspeicherung müssen zwei Arten von Speichern zur Verfügung stehen: ein Cache als heißer Speicher mit Daten über die nächsten Teile und über die letzten Teilequalitäten. Ein Langzeitspeicher wird für das Training des Modells benötigt. Dieser sollte möglichst alle Daten zu einem Prozess gebündelt speichern.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Die Anbindung kann direkt an ein Edge-Device erfolgen. Hierdurch können eine Vielzahl von Protokollen über Protokollwandler angewendet werden. Es sollte jedoch auf OPC UA und MQTT der Fokus gesetzt werden. Diese sind aktuell der industrielle Standard. Aufgrund älterer SPS in bestehenden Produktionen ist jedoch auch die Implementierung von Profinet sehr wichtig.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Ein Edge-Device ist hervorragend dazu geeignet, einen Produktionsprozess mit sehr niedrigen Latenzzeiten zu steuern, da die Verarbeitung sehr nah an den Maschinen erfolgt. Jedoch wird zusätzliche Hardware in die Produktionsumgebung gebracht. Das ist für die Instandhaltung negativ wegen des potentiellen Austauschs von defekter Hardware. Dies ist ein Spannungsfeld, weshalb ein Edge-Device nur bei absoluter technischer Notwendigkeit eingesetzt werden sollte.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt?</p> <p>Ein Projekt wird aktuell selbst initiiert für eine dynamische Prozesssteuerung in einem Presswerk. Bisher sonst keine gleichwertige Bebauung mit geschlossenem Regelkreis über ein Edge-Device bekannt. Weder bei OEMs, noch bei Anlagenbauern. Die Idee ist zu prüfen, welche Blech-Platine welche Beölung hat. Diese und andere Daten sollen in einem Cache-Speicher als Datensinke gespeichert werden. Nachfolgend soll die Verarbeitung, Berechnung und Steuerung über ein Microsoft IoT-Edge Modul nahe Echtzeit erfolgen. Die Anlernung des Machine-Learning-Modells erfolgt innerhalb der Microsoft Azure Cloud. Hier können viele vordefinierte Analysetools und auch eigene Analysen verwendet werden. In der Cloud wird die Software für die Steuerung zyklisch aktualisiert und an das Edge-Device übergeben. Dies erfolgt über Jobs aus der Cloud. Wichtig ist das Augenmerk auf die CI/CD-Kette zu legen. Über diese erfolgt die Aktualisierung der neusten Erkenntnisse auf das Edge-Device in Form eines neuen Steuerungsmodells.</p>				

A. Dokumentationen Experteninterviews

Tabelle A.5: Dokumentation Experteninterview 4.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
4	08.10.2018	Experte für Data Analytics	Männlich	60
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Ein Zwischenpuffer für wichtige Produktionsdaten sollte implementiert werden, um einen Datenverlust durch Verbindungsprobleme zu umgehen. Es müssen drei Arten von Daten erfasst und verarbeitet werden: Zeitreihendaten (aus SPS, durch Abstraten, mit Zeitstempel), Verlaufsdaten (aus Sensoren) und Large Data (oft > 100 MB). Eine möglichst freie Implementierung von Algorithmen ist gerade für weitreichende Datenanalysen von Data Analysten für die Modellaktualisierung sehr wichtig. Zusätzlich sind jedoch vorgefertigte Apps in der Cloud-Umgebung von Vorteil für schnelle Analysen. Eine möglichst freie Wahl des übermittelten Telegramms, wie z. B. JSON, bietet zudem viele Möglichkeiten Daten optimal zu transportieren.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Die Anbindung der lokalen Geräte sollte über ein Gateway mit einem lokalen Rechenzentrum oder einer Cloud erfolgen. Hierbei besteht die Möglichkeit, bei sehr großen Daten das Gateway direkt an den Langzeitspeicher anzubinden und so einen höheren Datendurchsatz zu ermöglichen.</p> <p>Wie sind zusätzliche Systeme (z. B. MES) an das Gesamtsystem anzubinden?</p> <p>Daten von MES-Systemen können zentral durch die Datenverarbeitung der Modellaktualisierung über eine ODBC-Schnittstelle abgefragt und gespiegelt werden. Diese Daten können dann über ein Gateway der Steuerung verfügbar gemacht werden. Hierdurch werden die angebundenen Datenbanksysteme entlastet. Deutlich mehr Last entsteht, wenn alle Steuerungen selbst die Daten abfragen. Hier entsteht die Gefahr, die Datenbanken durch zu viele Anfragen zu blockieren und andere produktive Systeme zu behindern.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt?</p> <p>Aktuell werden von Seiten Data Analytics nur Daten aus dem Produktionsprozess extrahiert und anschließend analysiert. Hierzu werden die Daten über ein Gateway zentral in die Cloud an ein IoT-Hub gesendet. Dieses puffert die Daten sieben Tage und ermöglicht eine Analyse in der Cloud oder die Daten direkt weiterzuleiten. Die Daten werden in der Regel unstrukturiert in der Cloud gespeichert. Über eine Standleitung ist jedoch die Weiterleitung an eine On-Premise-Cloud möglich.</p>				

Tabelle A.6: Dokumentation Experteninterview 5.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
5	21.11.2018	Experte für IoT	Männlich	60
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Das System muss interoperabel sein. So sollte das System möglichst modular hinsichtlich der Schnittstellen, Verarbeitungseinheiten und Software sein. Diese Module können als wiederverwendbare ‚Einheiten‘ den zukünftigen Entwicklungsaufwand reduzieren. Zudem ist es wichtig, sowohl kabelgebundene als auch drahtlose Geräte anzubinden. Hinsichtlich der Komponenten ist es wichtig auf Industriestandards zu setzen, um das System möglichst breitflächig einsetzen zu können. Die Vertraulichkeit und Sicherheit der Daten muss durch ein Schutzkonzept sichergestellt werden, sodass keine sensiblen Daten an Dritte gelangen können.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Die horizontale Kommunikation zwischen den Steuerungen kann direkt durch diese erfolgen oder durch eine überlagerte Schnittstelle. Diese wird in der Regel ein Gateway sein. Es ist empfehlenswert auf Feld- und Steuerungsebene Daten per Pull-Prinzip auszutauschen und auf OPC UA als Protokoll zu setzen. MQTT sollte für Peripheriegeräte mit einer geringen Leistung verwendet werden, da dieses wenige Ressourcen benötigt. Proprietäre Protokolle sollten vermieden werden. Zur Sicherstellung der Sicherheit können folgende Komponenten für den Datentransfer empfohlen werden: Zertifikate, Anonymisierung der Daten durch Vorverarbeitung, Lese-Rechte, IP-Filterung, Firewalls, Passwortmanagement, Aufteilung in interne und externe Systeme.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Es müssen drei Verarbeitungsressourcen unterschieden werden: Cloud-, Fog- und Edge-Computing. Für eine hohe zukünftige Skalierbarkeit ist die Cloud sehr gut, Fog gut und Edge eher schlecht geeignet. Dies hängt davon ab, in welcher Form eine physikalische Erweiterung durch den Bedarf an mehr Rechenleistung nötig ist. Edge-Computing ist jedoch hinsichtlich Echtzeit, Security (Verschlüsselung) und Safety (Datenverfügbarkeit, Vertraulichkeit) herausragend.</p> <p>Wie ist die Anbindung der Datenverarbeitung auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Unterschiede zwischen dem unternehmensinternen Netzwerk und dem externen Internet nehmen immer weiter ab und so sollte es in Zukunft eine geringe Bedeutung haben, über welche Netzwerkart die Daten zur Datenverarbeitung transportiert werden. Wichtig ist jedoch, dass der Datenaustausch auf dem Push-Prinzip basieren sollte, um nur relevante Daten zu übertragen. Zudem sollten eher Daten aus nachgelagerten Prozessen in die Cloud zu transferiert und von dort zielgerichtet zurück geleitet werden.</p> <p>Wie sind zusätzliche Systeme (z. B. MES) an das Gesamtsystem anzubinden?</p> <p>Die MES-Schnittstelle sollte in der zentralsten Instanz des Systems implementiert werden. Dies bietet einen Vorteil für die Skalierbarkeit bei großen Systemen mit vielen dezentralen Steuerungen. Die Daten sind hierbei über den Message-Broker Kafka entgegen zu nehmen und weiter zu verteilen. So können alle Steuerungen die Daten zentral abfragen.</p>				

A. Dokumentationen Experteninterviews

Tabelle A.7: Dokumentation Experteninterview 6.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
6	27.11.2018	Experte für Steuerungstechnik	Männlich	45
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Für eine Steuerung muss das System ein Feedback des Prozessergebnisses erfassen. Hierzu werden oft Referenzwerte verwendet, um Schlüsse über die Prozessqualität ziehen zu können. Der Datenfluss für die Steuerung muss zudem sichergestellt oder eine Notfallstrategie bei Steuerungsdatenverlust entwickelt werden. Die Softwarekomponenten und Schnittstellen sollten modularisiert sein, um hilfreich für weitere Projekte und eine zukünftige Skalierbarkeit zu sein.</p> <p>Welche Steuerungen sollten in der Automobilindustrie verwendet werden?</p> <p>In der Regel werden SPS von Siemens verwendet. Diese besitzen je nach Software eine Verarbeitungszeit von 5 bis 7ms, können aber bei aufwendigen Berechnungen auch langsamer sein. Aufwendige Berechnungen sollten daher nicht in einer SPS, sondern extern ausgeführt werden, um die Zykluszeiten und somit die Reaktionszeiten kurz zu halten.</p> <p>Wie ist eine dynamische Prozessanpassung umsetzbar?</p> <p>Die Steuerung über eine Parameteranpassung der SPS ist sinnvoller und sicherer gegenüber Störungen. So können Daten einmal pro Zyklus oder Schrittkette geliefert werden. Hierdurch kann die Verarbeitung komplett auf einmal oder teilweise nacheinander erfolgen. Durch eine teilweise Datenübertragung kann parallel zur Prozessausführung weiter verarbeitet und eine Entzerrung der Echtzeitanforderung ermöglicht werden. Wichtig ist zudem eine hohe Ausfallsicherheit sicherzustellen und eine Notfallstrategie zu implementieren. Denn es ist schlimm, wenn eine Steuerung defekt ist, aber noch schlimmer wenn die Anlage Schaden nimmt.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Weiterhin ist Profinet für die Anbindung der SPS an Aktoren sehr verbreitet und kann als Standard auf der Feldebene angesehen werden. Profi Safe wird hierbei seltener bei stark echtzeitkritischen Anwendungen implementiert. Die Anbindung der SPS erfolgt hierbei durch ein Switch an die Anlagen. IT-Systeme können auf SPS-Daten in der Regel über einen OPC-Server zugreifen. Dieser ist zumeist im Gateway zwischen der SPS und den überlagerten IT-Systemen implementiert. Die Aufrechterhaltung der Verbindung kann über ein Life-Bit oder Watch-Dog geprüft und bei Bedarf ein Reconnect initiiert werden.</p> <p>Was ist bei einer zusätzlichen Hardwareimplementierung in die Produktion zu beachten?</p> <p>Wichtig ist es, einen schnellen Hardwaretausch zu ermöglichen, denn es ist nicht üblich Redundanzen einer SPS oder eines Edge-Gerät zu implementieren. Hier ist es wichtig, die Hardware schnell zu tauschen und die Software umgehend automatisiert neu aufspielen zu können, um dadurch Ausfallzeiten und Produktionsstörungen zu minimieren.</p>				

Tabelle A.8: Dokumentation Experteninterview 7.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
7	30.11.2018	Experte für Data Analytics	Männlich	30
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Wichtig für ein solches System ist der mögliche Eingriff durch einen Mitarbeiter in der Produktion, da sonst eine sehr große Skepsis durch den Betreiber und die Mitarbeiter entstehen wird. Das System sollte eine stufenweise Integration in das Steuerungssystem eines existierenden Prozesses zulassen, da zuerst Trainingsdaten erfasst werden müssen. Mit diesen Daten können Modelle angelernet werden, die dann Handlungen vorhersagen. Zudem sollte in der Implementierungsphase das System parallel zur bisherigen Steuerung Vorhersagen treffen und diese dann auf Sinnhaftigkeit prüfen. Die Modellaktualisierung sollte eine Möglichkeit bieten, ein neues Produkt oder eine Prozessänderung sinnvoll in das Steuerungsmodell aufzunehmen.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Die Steuerungsebene ist über OPC UA anzubinden und durch ein Gateway mit einem Fog-Node oder der Cloud zu verbinden. Die Präferenz für die Anbindung von Sensoren und SPS ist Ethernet im Produktionsnetz. Diese Anbindung ist bei Sicherheitsprüfungen unkritisch als Einfallstor für Dritte. Eine drahtlose Anbindung ist sehr angenehm bei der Implementierung, jedoch schwierig bei der anschließenden Sicherung und so erzeugt diese in der Regel einen größeren Aufwand.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Eine Steuerung über eine Parameterübergabe ist sehr sinnvoll, da die Logik innerhalb der SPS teilweise sehr komplex und schwer in Steuerungsmodellen abzubilden ist. Eine direkte Steuerung von Aktoren ist nur über ein Edge-Device umsetzbar. Die Steuerung über einen Fog-Node erzeugt Verarbeitungszeiten, die für eine direkte Steuerung definitiv zu lang sind.</p> <p>Wie sind zusätzliche Systeme (z. B. MES) an das Gesamtsystem anzubinden?</p> <p>Daten sind aus Logistiksystemen, Fehlersystemen und Qualitätssystemen zu entnehmen, um eine sinnvolle Datenbasis für Analysen zu besitzen. Hierbei sind Logistik-, Qualitäts-, und Fehlerdaten vorzuhalten. Die Daten sind am sinnvollsten an einem zentralen Punkt in eine systemeigene Datenbank zu spiegeln. So können alle Komponenten des Systems auf diese Daten zugreifen.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt?</p> <p>Ein Projekt für die Erkennung der Typschilder ist bekannt. Hierbei werden über eine Kamera Bilder von Typschildern aufgenommen und anschließend analysiert. Eine Meldung, ob das Aufbringen korrekt war, wird abschließend dem Mitarbeiter in der Produktion übermittelt. Hierbei wurden die Bilddaten aufgrund der Größe zuerst durch ein Edge-Device vorverarbeitet, an eine On-Premise-Cloud gesendet, dort verarbeitet und anschließend über das Edge-Device eine Meldung abgegeben. Da dies sehr aufwendig, ist wurde das System umgestellt und die Bilddaten unverarbeitet an ein Fog-Node geleitet, dort verarbeitet und die Meldung direkt abgegeben.</p>				

A. Dokumentationen Experteninterviews

Tabelle A.9: Dokumentation Experteninterview 8.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
8	29.11.2018	Experte für Steuerungstechnik	Männlich	60
<p>Welche Steuerungen sollten in der Automobilindustrie verwendet werden? Eine direkte Steuerung über ein Edge-Device wird nicht empfohlen. Es sollte auf SPS mit ihrer sehr zuverlässigen Hard- und Software gesetzt werden.</p> <p>Wie ist eine dynamische Prozessanpassung umsetzbar? Es ist sehr sinnvoll, nicht in die Steuerung einzugreifen, sondern Prozessparameter der SPS angepasst zu übergeben. Hierdurch wird die Steuerung deutlich zuverlässiger werden. Wichtig ist jedoch, die Echtzeitanforderungen, also wie hoch die Latenz sein darf, bis eine Anpassung der Parameter erfolgt sein muss.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen? Bisher war es aufgrund von proprietären Standards sehr aufwendig Geräte anzubinden und Variablen in die SPS zu schreiben. OPC UA erleichtert dies sehr stark und ermöglicht ein standardisiertes Vorgehen. Daher ist OPC UA als Standard notwendig und sollte durch das System unterstützt werden.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt? Prescriptive Analytics ist in der Umsetzung sehr schwer. Bisher ist Predictive Analytics auch noch sehr aufwendig und steckt noch immer in den Kinderschuhen. Wichtig ist jedoch, dies jetzt schon in Steuerungskonzepten einzubeziehen. Aktuell werden die Daten über ein Gateway in eine On-Premise-Cloud geschickt, dort gespeichert und verarbeitet. Jedoch fehlt bisher die Rückspeisung der Analyseergebnisse in den Prozess.</p> <p>Sind weitere Komponenten besonders zu empfehlen? Zur Datenaufnahme und Weiterleitung, z. B. in die Cloud, ist der Message-Broker zuständig. Hierfür wird Apache Kafka als sehr zuverlässiges und schnelles System empfohlen. Durch dieses können auch die Daten des MES zentral abgefragt bzw. entgegengenommen und weitergeleitet werden.</p>				

Tabelle A.10: Dokumentation Experteninterview 9.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
9	24.09.2018	Experte für Steuerungstechnik	Männlich	30
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Es werden mehrere Datenarten für das System benötigt. Kurzzeit-Daten werden für die Steuerung benötigt und sollten in einem Cache-Speicher liegen. Historische Langzeit-Daten sollten persistent in einer NoSQL-Datenbank liegen, um zyklisch durch Machine Learning verarbeitet zu werden. Wichtig ist, die Daten nicht zu lange zu verwenden und in der Regel nach ca. fünf Monaten nicht mehr für das Training des Modells zu verwenden, da durch einen Datendrift falsche Relationen entstehen können. Steuerungsdaten ermöglichen die Prozessanpassung, jedoch ist es wichtig, bei Störungen eine Notfallsteuerung zu bedenken. Hierbei sollte eine Fallbacksteuerung auf Basis einer SPS oder ein kompletter Not-Halt in Erwägung gezogen werden. Die Architektur sollte mehrere Pufferarten zur Verfügung stellen. So können Datenpuffer auf drei Arten umgesetzt werden, über ein Edge-Device, das Gateway oder Kafka.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Die Umsetzung einer drahtlosen Verbindung über WLAN ist schwierig, da die Sicherheit problematisch ist und technische Überlagerungen/ Störungen entstehen können. Besser ist die Verwendung von LTE. Durch ein Mobiltelefon in Kombination mit einem Industrie-PC kann ein kleines zusätzliches Netzwerk, z. B. für Scanner und Sensoren, aufgebaut werden. Mit Blick in die Zukunft hinzu 5G könnte dies sehr attraktiv werden, wobei die Kosten zu beachten sind. Über einen Messagebus bzw. Protokolladapter können Protokolle übersetzt werden, da z. B. unterlagerte Sensoren zum Teil proprietäre Protokolle verwenden und so in OPC UA vereinheitlicht werden.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Die Auswahl hängt sehr stark von der gewünschten Skalierbarkeit und der Datenvertraulichkeit ab. Edge-Computing ermöglicht das Verarbeiten von kritischen Daten bei sehr niedrigen Latenzzeiten, jedoch ist es kaum skalierbar. Fog-Computing ermöglicht bei Verlust des Latenzzeitvorteils eine Skalierbarkeit im Rechenzentrum. Cloud-Computing lässt sich nahezu unbeschränkt skalieren, jedoch sind auch die Latenzzeiten sehr hoch. Wichtig bei Cloud-Computing ist eine Informationsbedarfsanalyse zur Einstufung, ob die Daten das Unternehmen verlassen dürfen.</p> <p>Wie ist die Anbindung der Datenverarbeitung auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Der Kafka-Broker kann eine Vielzahl von Verbindungen verwalten und nimmt die Daten des Gateways für die Datenverarbeitung an. Anschließend leitet er die Daten an den Zielort, wie z. B. einen Langzeitspeicher bzw. Data Lake. Hierbei entkoppelt er die Verbindungen. Kafka ist z. B. in der Lambda-Architektur ein zentrales Element und ermöglicht parallele Verarbeitungen.</p> <p>Wie sind zusätzliche Systeme (z.B. MES) an das Gesamtsystem anzubinden?</p> <p>MES-Daten können über die Cloud per HTTP-Call und Insert auf ein Datenpaket aus dem MES-System abgefragt und anschließend gespiegelt für das gesamte System zur Verfügung gestellt werden. Die Schnittstelle zur Cloud kann über einen Connector des Message-Broker Kafka erfolgen.</p>				

A. Dokumentationen Experteninterviews

Tabelle A.11: Dokumentation Experteninterview 10.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
10	03.12.2018	Experte für Steuerungstechnik	Männlich	30
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Wichtig ist eine einfache Instandhaltung der Steuerung. So sollte die Software durch einen Client automatisiert auf dem Steuerungsrechner installiert werden.</p> <p>Welche Steuerungen sollten in der Automobilindustrie verwendet werden?</p> <p>Künftig wird die Verwendung von Steuerungsrechnern zunehmen, da Zykluszeiten kürzer werden sollen. Dies wird durch performantere Prozessoren mit mehreren Kernen erreicht. So kann der erste Kern die Auswertung anhand des Steuerungsprognosemodells übernehmen und weitere Kerne die Steuerungsfunktion der Anlage.</p> <p>Wie ist eine dynamische Prozessanpassung umsetzbar?</p> <p>Übergabe von Parametern an eine SPS sollte bevorzugt werden, da die SPS für die Aktor-Steuerung prädestiniert sind und eine sehr zuverlässige Steuerung sicherstellen können.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Für den Fall, dass die Bandbreite des Produktionsnetzwerkes nicht ausreicht, kann ein zweites Netzwerk nur für dieses Gerät (zumeist Kameras) aufgebaut werden und über ein Gateway eingebunden werden. Es kann auch ein Edge-Device für eine Vorverarbeitung verwendet werden. Bei einer echtzeitkritischen Verbindung zwischen der Steuerung und z. B. einem Aktor ist OPC UA als Standardprotokoll mit TSN zu implementieren.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt?</p> <p>Bisher wurde noch keine Rückführung der Daten aus dem Prozess zu einer umfassenden Verarbeitung und wieder zurück implementiert. Nur auf lokaler Ebene wurden über eine SPS-Berechnung rudimentäre Auswertungen durchgeführt.</p>				

Tabelle A.12: Dokumentation Experteninterview 11.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
11	03.12.2018	Experte für Data Analytics	Männlich	60
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Für die Untersysteme der IT-Architektur sollte beachtet werden, wie produktionskritisch diese sind. Die Ausfallsicherheit der Steuerung ist daher sehr wichtig. Bei einer Fog-Steuerung ist diese bei 100%, bei einer Steuerung über Verbindung außerhalb des Unternehmensnetzwerks lediglich bei maximal 99% aufgrund der Komplexität der Internetverbindung. Die historischen, unstrukturierten Daten sind für die Modellaktualisierung bis zu einem Jahr vorzuhalten, jedoch ist es sinnvoll, nur die Daten der letzten fünf Monate zu verwenden, da sich Parameter immer wieder verändern. Zudem ist es wichtig zu betrachten welche Datenmengen erzeugt werden und wie lange diese Daten einen Nutzen erzeugen. Für die Steuerung sind strukturierte Daten zu verwenden, wobei auf eine ausreichende Auflösung der Zeitreihendaten zu achten ist. Da die Verbindung zur Cloud in der Regel im 100ms Takt stattfindet, müssen höhere Auflösungen in einer Datei zusammengefasst und als Batch-Datei versendet werden.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Es muss zwischen zwei grundlegenden Aufgaben unterschieden werden. Die Visualisierung durch Descriptive Analytics sollte für aktuelle Daten in einem Fog-Node und für historische Daten in der Cloud erfolgen. Die auf Machine Learning basierende Steuerung sollte auf einem Edge-Device oder einem Fog-Node und die Modellaktualisierung zentral in der Cloud erfolgen. In der Cloud ist primär die Batch-Verarbeitung bisher anzutreffen, jedoch kann durch die Lambda-Architektur dem Wunsch nach mehr Stream-Verarbeitungen entgegengekommen werden. Die richtige Reihenfolge der Verarbeitungsschritte für die Modellaktualisierung sollte hierbei wie folgt aussehen: die erfassten Daten werden durch den Message-Broker Kafka entgegengenommen, in einen Langzeitspeicher abgelegt und zyklisch über Data Analytics und Machine Learning wird Wissen generiert. Dieses Wissen wird anschließend durch die Erzeugung eines ausführbaren Modells nutzbar gemacht. Dieses Modell sollte als Docker Container der Steuerung zur Verfügung gestellt werden, denn diese werden sich in Zukunft nahezu sicher als Standard für Micro-Apps durchsetzen. Docker Container bieten den Vorteil, dass sie skalierbar und sehr flexibel in der Einsetzung sind. Jedoch sind sie nur unter Linux performant und der Container ist eine zusätzliche Fehlerquelle.</p> <p>Wie sind zusätzliche Systeme (z. B. MES) an das Gesamtsystem anzubinden?</p> <p>Der Datenaustausch zwischen MES-Systemen sollte über die Cloud gelöst werden. Dies ist hinsichtlich der Fehleranfälligkeit deutlich sicherer. Die direkte Anbindung an ein Edge-Device kann für produktive Störungen sorgen und bei zu vielen Anfragen das MES lahmlegen. Der Datenbezug erfolgt über eine Anfrage und Spiegelung der Daten. Diese gespiegelten Daten können dann zentral zur Verfügung gestellt werden.</p>				

A. Dokumentationen Experteninterviews

Tabelle A.13: Dokumentation Experteninterview 12.

Nr.	Datum	Expertise	Geschlecht	Dauer (Min.)
12	05.12.2018	Experte für IoT	Männlich	35
<p>Welche Anforderungen sind an eine IT-Architektur für Prescriptive Automation zu stellen?</p> <p>Eine einfache Integration des Systems in die Produktion soll möglich sein, wie z. B. die Anbindung von Kameras über Power over Ethernet. Neben Kurzzeit-Daten für die Steuerung wird ein Langzeitdatenspeicher für das Machine Learning benötigt. Wichtig hierbei ist, nicht nur Daten von fehlerhaften Prozessen zu erfassen, sondern auch von korrekten Prozessen, um spätere Ausfälle oder Qualitätseinbußen nachverfolgen zu können.</p> <p>Wie ist die Anbindung der Feld- und Steuerungsebene auszugestalten und welche Protokolle werden für den Datenaustausch empfohlen?</p> <p>Der Verwendung von WLAN für die Anbindung wird abgeraten, da diese sehr aufwendig in der Umsetzung ist. So werden in der Regel viele Genehmigungsläufe benötigt und aus Betreibersicht werden nur absolut notwendige WLAN-Bebauungen akzeptiert, da eine sehr begrenzte Kapazität der Frequenzbänder vorliegt. Zudem ist WLAN eine potenziell große Fehlerquelle. Auch 4G wird nicht empfohlen, da dies sehr teuer ist. In Zukunft könnten unternehmenseigene 5G Netze eine Alternative für die drahtlose Anbindung sein.</p> <p>Welche Verarbeitungsressourcen werden für die Datenverarbeitung empfohlen?</p> <p>Für die Auswahl muss entschieden werden, wie kritisch die Daten sind. Dies entscheidet, ob die Daten in der Cloud verarbeitet werden können. Wenn die Daten zu kritisch sind, ist die Frage ob die Verarbeitungen für die Modellierung nicht im Werksrechenzentrum durchgeführt werden können. Ziel sollte es sein, möglichst geringe Hardwareumfänge in die Produktion zu bringen und nach Möglichkeit auf ein Edge-Device zu verzichten. Die Empfehlung für die Datenverarbeitung lautet einen Fog-Node zu verwenden.</p> <p>Sind aktuelle Projekte in die Richtung von Prescriptive Automation bekannt und wie sind diese hinsichtlich der IT-Architektur umgesetzt?</p> <p>Die Erfassung von Fahrzeug-Features mittels eines Smartphones; die Weiterleitung der Daten über ein Gateway in eine Off-Premise-Cloud; die Verarbeitung, ob das verbaute Element dem Korrekten für dieses Fahrzeug entspricht; Meldung zurück über das Gateway an ein HMI für den Mitarbeiter, ob der Verbau korrekt. Das System benötigt nur einen geringen Datentransfer, da Einzelbilder pro Takt. Das System liefert innerhalb von Bruchteilen einer Sekunde das Ergebnis.</p>				

Anhang B

Vergleich der unterschiedlichen CNN-Netzwerke

In diesem Kapitel werden die unterschiedlichen CNN-Architekturen verglichen. Dabei werden die gängigsten Netzwerke in Betracht gezogen. Diese heißen VGG, ResNet und Inception-v3. Wie in Tabelle B.1 zu erkennen, wurden wie Metriken Genauigkeit des Trainings- und Validierungssatzes, die Precision als auch die Anzahl an Parametern miteinander verglichen. Die dargestellten Werte sind die Mittelwerte einer 5-fold Cross-Validation. Dabei wurde der in Abschnitt 5.3 beschriebene Datensatz verwendet.

Tabelle B.1: Vergleich der unterschiedlichen CNN-Netzwerke.

Architektur	Genauigkeit (Validierung)	Genauigkeit (Training)	Precision	Anzahl Parameter
VGG-10	99,56 %	98,35 %	99,15 %	76.220
ResNet-10	99,34 %	98,79 %	98,89 %	79.110
Inception-v3	99,04 %	99,31 %	98,63 %	23.870.000

Wie zu erkennen, besitzt das VGG die wenigsten Parameter. Dies ist von Vorteil in der Produktion mit einer kleinen Taktzeit, da diese dadurch nicht negativ beeinflusst wird. Darüber hinaus ist sowohl die Genauigkeit des Validierungssatzes als auch die Precision die höchste. Wie bereits in Abschnitt 2.4.6.3 beschrieben, ist die essenziell, da dadurch die Wahrscheinlichkeit, dass schlechte Bauteile als gute klassifiziert werden am geringsten ist. Die zweithöchste Genauigkeit wurde mit dem ResNet erzielt. Dieses hat jedoch eine etwas schlechtere Precision und besitzt darüber hinaus mehr zu trai-

B. Vergleich der unterschiedlichen CNN-Netzwerke

nierende Parameter. Das Inception-v3 besitzt die geringste Genauigkeit und ist zudem aus insgesamt 23.870.000 trainierbaren Parametern aufgebaut, was, wie beschrieben, ein starker Nachteil ist. Aus diesen Gründen ist das VGG zu verwenden.

Anhang C

Evaluierung der Netzarchitektur

In diesem Kapitel werden ergänzende, für die kennzahlenbasierte Evaluierung des CNN benötigten CM abgebildet.

		Lauf 1				Lauf 2				Lauf 3				Lauf 4				Lauf 5			
		vorhergesagte Klasse				vorhergesagte Klasse				vorhergesagte Klasse				vorhergesagte Klasse				vorhergesagte Klasse			
		KV	UV	SK	SS																
echte Klasse	KV	122	0	0	0	135	3	0	0	116	1	0	0	122	1	0	0	104	0	0	0
	UV	0	122	0	0	0	116	0	0	2	106	0	0	2	108	0	0	1	103	0	0
	SS	0	0	102	0	0	0	100	0	0	0	121	0	0	0	112	0	0	0	122	0
	SK	0	0	0	111	0	0	0	103	0	0	0	111	0	0	0	112	0	0	0	126

Abbildung C.1: Confusion Matrix zur Evaluierung der Netz-Architektur mit 3D-Bildern als Eingangsgröße.

		Lauf 1				Lauf 2				Lauf 3				Lauf 4				Lauf 5			
		vorhergesagte Klasse				vorhergesagte Klasse				vorhergesagte Klasse				vorhergesagte Klasse				vorhergesagte Klasse			
		KV	UV	SK	SS	KV	UV	SK	SS	KV	UV	SK	SS	KV	UV	SK	SS	KV	UV	SK	SS
echte Klasse	KV	116	2	0	4	127	5	2	4	110	1	3	3	114	3	0	6	99	4	0	1
	UV	1	121	0	0	1	113	2	0	3	105	0	0	2	108	0	0	0	104	0	0
	SS	0	0	96	6	1	1	95	3	0	2	116	3	0	0	110	2	0	0	116	6
	SK	0	1	4	106	1	2	9	91	0	3	8	100	0	0	7	105	0	2	6	118

Abbildung C.2: Confusion Matrix zur Evaluierung der Netz-Architektur mit Schwarz-Weiß-Bildern als Eingangsgröße.

Anhang D

Visualisierung der Web-Applikation

In diesem Kapitel sind weitere Screenshots der Web-Applikation dargestellt.



The screenshot shows a web application interface for manual classification. At the top left is the ARWEL logo. At the top right are the words "Start" and a blue button labeled "Manuelle Klassifizierung". The main heading is "Benötigte Neuklassifizierung." followed by the instruction "Bitte beantworten Sie die folgenden Fragen, um den aktuellen Pin neu zu klassifizieren." Below this are four input fields: "Stator-Nr.: 13", "Pin-Nr.: 1", "Prädikierte Klasse: UE", and "Prädikierte Subklasse: 40W". To the right of these fields is a dark, blurry image of a weld. At the bottom center is a grey button labeled "Weiter".

Abbildung D.1: Übersicht aktueller Informationen einer Verschweißung.

D. Visualisierung der Web-Applikation

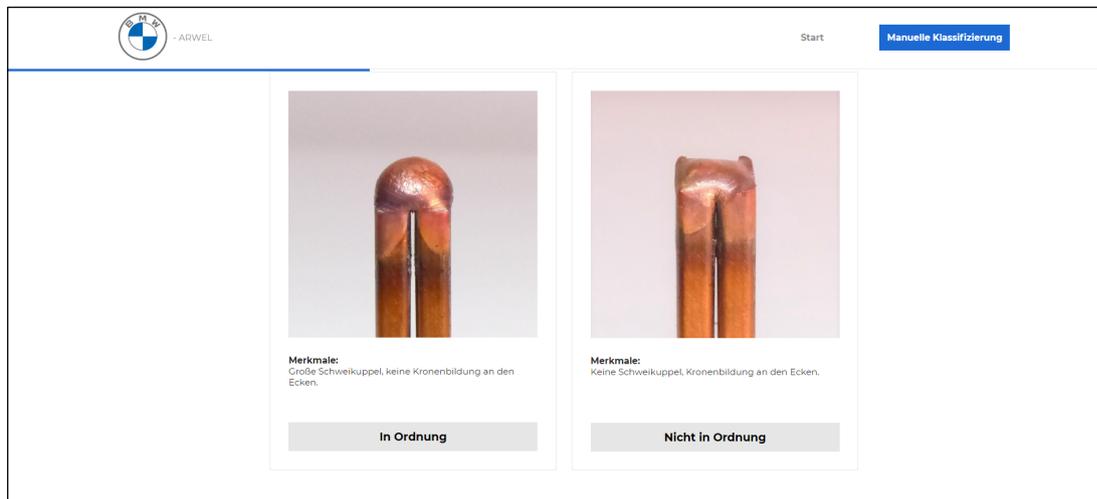


Abbildung D.2: Auswahl Klasse in Ordnung und nicht in Ordnung.

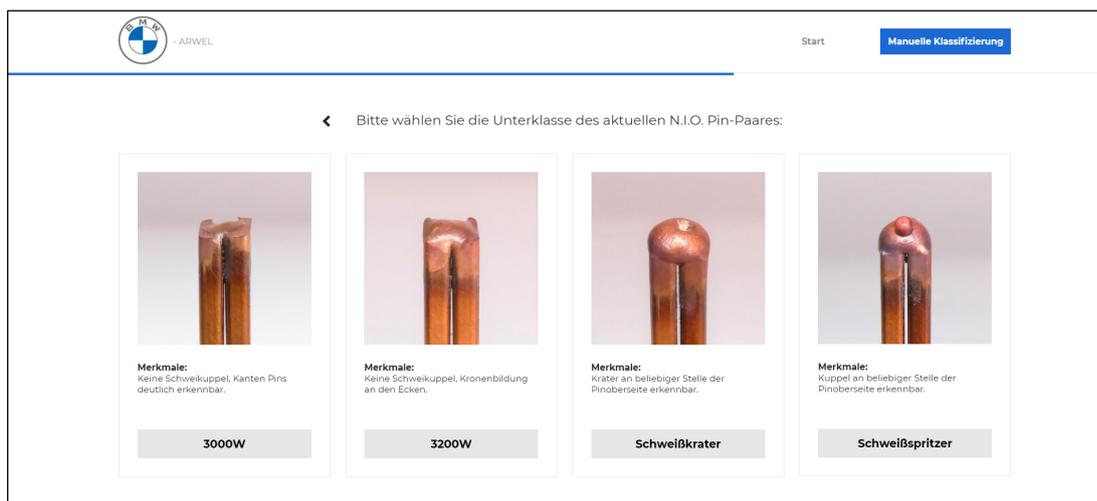


Abbildung D.3: Auswahl der Subklassen nicht in Ordnung.

Literatur

- [1] D. Hawranek. *Ständig unter Strom*. Hrsg. von Der Spiegel. 2014.
- [2] *Verteilung der Bruttowertschöpfung in Deutschland nach Wirtschaftszweigen im Jahr 2019*. URL: <https://de.statista.com/statistik/daten/studie/252123/umfrage/anteil> (besucht am 03.06.2020).
- [3] J. Leal, R. Lehmann, B. Marc, T. Wollmershäuser und P. Wozniak. “The Weakness of the German Car Industry and Its Sectoral and Global Impacts”. In: (2019). URL: https://www.econpol.eu/sites/default/files/2019-09/EconPol_Policy_Brief_18_German_Car_Industry.pdf.
- [4] T. Wollmershäuser u. a. “ifo Konjunkturprognose Winter 2019: Deutsche Konjunktur stabilisiert sich”. In: *ifo Schnelldienst* 72.24 (2019), S. 27–89. ISSN: 0018-974X. URL: <http://hdl.handle.net/10419/216115>.
- [5] M. Homm, M. Lienkamp, W. Schmid und S. Wolff. *Status Elektromobilität 2020: Das Endspiel nach der Corona-Krise*. 2020. URL: <https://www.researchgate.net/publication/341670568>.
- [6] M. Breiting. “Der Abgasskandal”. In: (2018). URL: <https://www.zeit.de/wirtschaft/diesel-skandal-volkswagen-abgase> (besucht am 03.06.2020).
- [7] *CO₂-Emission*. 2020. URL: <https://www.bundesregierung.de/breg-de/themen/energiewende/co2-kohlenstoffdioxid-oder-kohlendioxid-emission-614692> (besucht am 03.06.2020).
- [8] A Kampker. *Elektromobilproduktion*. Berlin: Springer Vieweg, 2014. ISBN: 978-3-642-42022-1.
- [9] W. Huber. “Produktion der Zukunft”. In: *Industrie 4.0 in der Automobilproduktion: Ein Praxisbuch*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016, S. 259–269. ISBN: 978-3-658-12732-9. DOI: 10.1007/978-3-658-12732-9_9.

LITERATUR

- [10] J. Huber. “Verfahren zur Klassifikation von Ungängen bei der optischen Prüfung von Batterieseparatoren”. Dissertation. München: Technische Universität München, 2017.
- [11] A. Priestley. *Market Insight: Tech CEOs Must Leverage Ecosystems and Partnerships for Business Success With AI-Enabled Vision Systems*. Hrsg. von Gartner. 2019.
- [12] Intel. *Making a Success of Computer Vision*. 2019. URL: <https://www.insight.tech/white-papers-and-other-premium-assets/making-a-success-of-computer-vision>.
- [13] N. O’Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan und J. Walsh. “Deep Learning vs. Traditional Computer Vision”. In: *Advances in Computer Vision*. Hrsg. von K. Arai und S. Kapoor. Bd. 943. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, S. 128–144. ISBN: 978-3-030-17794-2. DOI: 10.1007/978-3-030-17795-9_10.
- [14] C. Gröger, H. Schwarz und B. Mitschang. “Prescriptive Analytics for Recommendation Based Business Process Optimization”. In: *Business Information Systems*. Bd. 176. Cham: Springer International Publishing, 2014, S. 25–37. ISBN: 978-3-319-06694-3. DOI: 10.1007/978-3-319-06695-0_3.
- [15] P. Burggräf, A. Kampker, C. Deutskens und C. Niebuhr. *Competitive Strategies for Value Creation During Disruptive Innovations*. 2013. ISBN: 978-0-7972-1405-7.
- [16] H. Hanselka und M. Jöckl. “Elektromobilität - Elemente, Herausforderungen, Potenziale”. In: *Elektromobilität*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 21–38. ISBN: 978-3-642-16254-1.
- [17] B. Schuler. *Elektromobilität und Elektrofahrzeuge: Ökonomische Bewertung des Marktpotenzials im Jahr 2020*. 1. Aufl. Hamburg: Diplomica Verlag, 2011. ISBN: 978-3-842-85769-8.
- [18] A. Schneider und S. N. Grösser. “Elektromobilität: Ist das Elektrofahrzeug eine disruptive Innovation?” In: *Zeitschrift für die gesamte Wertschöpfungskette Automobilwirtschaft* 1 (2013), S. 1–14. ISSN: 1434-1808.

-
- [19] A. Kampker, K. Kreisköther, M. Hehl, S. Gillen und M. Rothe. “Discrete event simulation approach considering scalable systems and non-expert users in the early phase of production planning for electric powertrains”. In: *2017 6th International Conference on Industrial Technology and Management (ICITM)*. 2017, S. 139–144. DOI: 10.1109/ICITM.2017.7917910.
- [20] H. Heimes, A. Kampker, U. Bühner und S. Krotil. “Potenziale und Hürden von Data Analytics in der Serienfertigung”. In: *Industrie 4.0 Management 2019.1* (2019), S. 57–60. ISSN: 23649208. DOI: 10.30844/I40M_19-1_S57-60.
- [21] A. Mayr, J. Seefried, M. Ziegler, M. Masuch, A. Mahr, J. v. Lindenfels, M. Meiners, D. Kisskalt, M. Metzner und J. Franke. “Machine Learning in Electric Motor Production - Potentials, Challenges and Exemplary Applications”. In: *2019 9th International Electric Drives Production Conference (EDPC)*. IEEE, 2019, S. 1–10. ISBN: 978-1-7281-4319-4. DOI: 10.1109/EDPC48408.2019.9011861.
- [22] W. Dangelmaier. *Theorie der Produktionsplanung und -steuerung - Im Sommer keine Kirschpralinen?* 2009. Aufl. Berlin Heidelberg: Springer Science & Business Media, 2009. ISBN: 978-3-642-00632-6.
- [23] H. Dyckhoff und T. S. Spengler. *Produktionswirtschaft - Eine Einführung*. 3. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2010. ISBN: 978-3-642-13684-9.
- [24] M. Steven. *Handbuch Produktion - Theorie - Management - Logistik - Controlling*. Stuttgart: W. Kohlhammer Verlag, 2007. ISBN: 978-3-170-18312-4.
- [25] T. Becker. *Prozesse in Produktion und Supply Chain optimieren*. 3. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2018. ISBN: 978-3-662-49075-4.
- [26] B. Heinrich, P. Linke und M. Glöckler. *Grundlagen Automatisierung: Sensorik, Regelung, Steuerung*. 2., überarbeitete und erweiterte Auflage. Wiesbaden: Springer Vieweg, 2017. ISBN: 978-3-658-17582-5.
- [27] K. Barton, F. Maturana und D. Tilbury. “Closing the Loop in IoT-enabled Manufacturing Systems: Challenges and Opportunities”. In: *2018 Annual American Control Conference (ACC)*. IEEE, 2018, S. 5503–5509. ISBN: 978-1-5386-5428-6. DOI: 10.23919/ACC.2018.8431577.
- [28] T. Bauernhansl, Hrsg. *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien und Migration*. Wiesbaden: Springer Vieweg, 2014. ISBN: 978-3-658-04682-8.
- [29] K. R. Stoesser. *Prozessoptimierung für produzierende Unternehmen*. 2. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2019. ISBN: 978-3-658-25368-4.

LITERATUR

- [30] F. J. Brunner und K. W. Wagner. *Qualitätsmanagement - Leitfaden für Studium und Praxis*. 6. Aufl. Carl Hanser Verlag GmbH Co KG, 2016. ISBN: 978-3-446-44840-7.
- [31] C. Gröger. *Advanced Manufacturing Analytics - Datengetriebene Optimierung von Fertigungsprozessen*. 1. Aufl. Norderstedt: BoD – Books on Demand, 2015. ISBN: 978-3-844-10420-2.
- [32] S. Niedermann F.and Radeschütz und B. Mitschang. “Business Process Optimization Using Formalized Optimization Patterns”. In: *Business Information Systems*. Hrsg. von W. Abramowicz. Bd. 87. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, S. 123–135. ISBN: 978-3-642-21829-3. DOI: 10.1007/978-3-642-21863-7_11.
- [33] R. Y. Zhong, X. Xu, E. Klotz und S. T. Newman. “Intelligent Manufacturing in the Context of Industry 4.0: A Review”. In: *Engineering 3.5* (2017), S. 616–630. ISSN: 20958099. DOI: 10.1016/J.ENG.2017.05.015.
- [34] S. I. Chang. “Approaches to Implement Statistical Process Control for Manufacturing in Big Data Era”. In: *Volume 3: Manufacturing Equipment and Systems*. ASME, 2017, V003T04A009. ISBN: 978-0-7918-5074-9. DOI: 10.1115/MSEC2017-2840.
- [35] M. Sanjay und B. H. Alamma. “An insight into big data analytics — Methods and application”. In: *2016 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2016, S. 1–5. ISBN: 978-1-5090-1285-5. DOI: 10.1109/INVENTIVE.2016.7823269.
- [36] B. Esmaeilian, S. Behdad und B. Wang. “The evolution and future of manufacturing: A review”. In: *Journal of Manufacturing Systems 39* (2016), S. 79–100. ISSN: 02786125. DOI: 10.1016/j.jmsy.2016.03.001.
- [37] P. O’Donovan, K. Leahy, K. Bruton und D. T. J. O’Sullivan. “Big data in manufacturing: a systematic mapping study”. In: *Journal of Big Data 2.1* (2015), S. 145. DOI: 10.1186/s40537-015-0028-x.
- [38] Y. Zhang, S. Ren, Y. Liu und S. Si. “A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products”. In: *Journal of Cleaner Production 142* (2017), S. 626–641. ISSN: 09596526. DOI: 10.1016/j.jclepro.2016.07.123.

- [39] C. Wang, H. Cheng und Y. Deng. “Using Bayesian belief network and time-series model to conduct prescriptive and predictive analytics for computer industries”. In: *Computers & Industrial Engineering* 115 (2018), S. 486–494. ISSN: 03608352. DOI: 10.1016/j.cie.2017.12.003.
- [40] J. Hagerty. *2017 Planning Guide for Data and Analytics*. 2016. URL: https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/2017_planning_guide_for_data_analytics.pdf.
- [41] A. Brodsky, G. Shao, M. Krishnamoorthy, A. Narayanan, D. Menasce und R. Ak. “Analysis and optimization in smart manufacturing based on a reusable knowledge base for process performance models”. In: *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 2015, S. 1418–1427. ISBN: 978-1-4799-9926-2. DOI: 10.1007/s00170-016-8761-7.
- [42] E. Turban, R. Sharda und D. Delen. *Decision support and business intelligence systems*. 9th ed. Upper Saddle River: Pearsonal, 2011. ISBN: 978-0-13-610729-3.
- [43] J. Krumeich, D. Werth und P. Loos. “Prescriptive control of business processes”. In: *Business & Information Systems Engineering* 58.4 (2016), S. 261–280.
- [44] R. Soltanpoor und T. Sellis. “Prescriptive analytics for big data”. In: *Australasian Database Conference*. Springer. 2016, S. 245–256. DOI: https://doi.org/10.1007/978-3-319-46922-5_19.
- [45] P. Kaur, R. Kumar und M. Kumar. “A healthcare monitoring system using random forest and internet of things (IoT)”. In: *Multimedia Tools and Applications* 78.14 (2019), S. 19905–19916. ISSN: 1380-7501. DOI: 10.1007/s11042-019-7327-8.
- [46] A. Abraham, A. K. Cherukuri, P. Melin und N. Gandhi, Hrsg. *Intelligent Systems Design and Applications*. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-16659-5. DOI: 10.1007/978-3-030-16660-1.
- [47] M. Fernandes, A. Canito, J. M. Corchado und G. Marreiros. “Fault Detection Mechanism of a Predictive Maintenance System Based on Autoregressive Integrated Moving Average Models”. In: *Distributed Computing and Artificial Intelligence, 16th International Conference*. Hrsg. von F. Herrera, K. Matsui und S. Rodríguez-González. Bd. 1003. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, S. 171–180. ISBN: 978-3-030-23886-5. DOI: 10.1007/978-3-030-23887-2_20.

LITERATUR

- [48] K. Lepenioti, A. Bousdekis, D. Apostolou und G. Mentzas. “Prescriptive analytics: Literature review and research challenges”. In: *International Journal of Information Management* 50 (2020), S. 57–70. ISSN: 02684012. DOI: 10.1016/j.ijinfomgt.2019.04.003.
- [49] A. Basu. *Executive Edge: Five pillars of prescriptive analytics success*. 2013. URL: <http://analytics-magazine.org/executive-edge-five-pillars-of-prescriptive-analytics-success/>.
- [50] M. Correa, C. Bielza und J. Pamies-Teixeira. “Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process”. In: *Expert systems with applications* 36.3 (2009), S. 7270–7279. DOI: 10.1016/j.eswa.2008.09.024.
- [51] J. Park, B. Kwon, J. Park und D. Kang. “Machine learning-based imaging system for surface defect inspection”. In: *International Journal of Precision Engineering and Manufacturing-Green Technology* 3.3 (2016), S. 303–310.
- [52] S. Khan, H. Rahmani, S. A. A. Shah und M. Bennamoun. *A guide to convolutional neural networks for computer vision*. Morgan & Claypool Publishers, 2018.
- [53] B. C. Love. “Comparing supervised and unsupervised category learning”. In: *Psychonomic bulletin & review* 9.4 (2002), S. 829–835. ISSN: 1069-9384. DOI: 10.3758/BF03196342.
- [54] R. Sathya und A. Abraham. “Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification”. In: *International Journal of Advanced Research in Artificial Intelligence* 2.2 (2013). ISSN: 21654050. DOI: 10.14569/IJARAI.2013.020206.
- [55] K. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [56] R. C. Hermes. *Entwicklung flexibler Bildverarbeitungsketten zur Klassifikation und Verschleissmessung an Wendeschneidplatten*. Bd. Bd. 2007,26. Berichte aus der Produktionstechnik. Aachen: Shaker, 2007. ISBN: 978-3-8322-6610-3.
- [57] C. Demant, B. Streicher-Abel und A. Springhoff. *Industrielle Bildverarbeitung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. ISBN: 978-3-642-13096-0. DOI: 10.1007/978-3-642-13097-7.
- [58] S. Ji, M. Yang und K. Yu. “3D convolutional neural networks for human action recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2013), S. 221–231. DOI: 10.1109/TPAMI.2012.59.

-
- [59] R. Kruse. *Computational Intelligence: A Methodological Introduction*. Springer-Verlag, 2013. DOI: 10.1007/978-1-4471-5013-8.
- [60] A. Kroll. *Computational Intelligence: Probleme, Methoden und technische Anwendungen*. De Gruyter Oldenbourg, 2016. DOI: <https://doi.org/10.1515/9783110401776>.
- [61] U. L. Jürgen C. *Data Mining*. De Gruyter Oldenbourg, 2016.
- [62] M. Hagan, H. Demuth, M. Beale und O. de Jesús. *Neural network design*. Bd. 20. Pws Pub. Boston, 1996. ISBN: 0971732116.
- [63] H. H. Aghdam und E. J. Heravi. *Guide to Convolutional Neural Networks*. Bd. 10. Springer, 2017. ISBN: 978-3-319-57550-6.
- [64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever und R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), S. 1929–1958.
- [65] R. Venkatesan und B. Li. *Convolutional Neural Networks in Visual Computing: A Concise Guide*. CRC Press, 2017. ISBN: 1138747955.
- [66] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao und J. Cong. “Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks”. In: *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA '15*. Hrsg. von G. A. Constantinides und D. Chen. New York, New York, USA: ACM Press, 2015, S. 161–170. ISBN: 9781450333153. DOI: 10.1145/2684746.2689060.
- [67] Y. Chen, H. Jiang, C. Li, X. Jia und P. Ghamisi. “Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks”. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.10 (2016), S. 6232–6251. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2584107.
- [68] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim und M.-H. Yang. “Hedged Deep Tracking”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, S. 4303–4311. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.466.
- [69] H. O. Song, Y. Xiang, S. Jegelka und S. Savarese. “Deep Metric Learning via Lifted Structured Feature Embedding”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, S. 4004–4012. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.434.
- [70] I. G., Y. B. und A. C. *Deep Learning*. MIT Press, 2016.

LITERATUR

- [71] A. Zhang, Z. C. Lipton, M. Li und A. J. Smola. *Dive into Deep Learning*. <http://www.d2l.ai>. 2019.
- [72] F. Chollet. *Deep learning with Python*. Shelter Island, NY: Manning, 2018. ISBN: 978-1-61729-443-3.
- [73] M. Sokolova und G. Lapalme. “A systematic analysis of performance measures for classification tasks”. In: *Information Processing & Management* 45.4 (2009), S. 427–437. DOI: 10.1016/j.ipm.2009.03.002.
- [74] F. Krüger. “Activity, Context, and Plan Recognition with Computational Causal Behaviour Models”. Dissertation. Dez. 2016.
- [75] R. P. Anderson, D. Lew und A. T. Peterson. “Evaluating predictive models of species’ distributions: criteria for selecting optimal models”. In: *Ecological Modelling* 162.3 (2003), S. 211–232. ISSN: 03043800. DOI: 10.1016/S0304-3800(02)00349-6.
- [76] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh und D. Batra. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, S. 618–626. DOI: 10.1109/ICCV.2017.74.
- [77] K. Simonyan, A. Vedaldi und A. Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps”. In: *arXiv preprint arXiv:1312.6034* (2013).
- [78] Verband der Automobilindustrie e.V. “Qualitätsmanagement in der Automobilindustrie - Prüfprozesseignung”. In: *VDA Band 5* (2011). ISSN: 0943-9412.
- [79] V. D. Ingenieure. *Industrielle Bildverarbeitung Abnahme klassifizierender Bildverarbeitungssysteme Prüfung der Klassifikationsleistung*. 2019.
- [80] C.-U. Ronniger. *Taschenbuch der statistischen Methoden - Die wichtigsten Methoden und Verfahren für die Praxis*. 5. Aufl. 2019.
- [81] BMW. *Eignungsnachweis für klassierende Bildverarbeitungssysteme*. Internes Dokument bei der BMW AG.
- [82] M. Radeck. *Einfach- Stichprobenanweisung: Verfahren zur Ermittlung des Stichprobenumfangs auf der Grundlage der Annahmestichprobenprüfung (Einstichproben Prüfanweisung)*. Hrsg. von B. AG. 2016.
- [83] C.-U. Ronniger. *Taschenbuch der statistischen Methoden - Die wichtigsten Methoden und Verfahren für die Praxis*. 6. Aufl. 2020.

-
- [84] Verband der Automobilindustrie e.V. “Qualitätsmanagement in der Automobilindustrie - Sicherung der Qualität in der Prozesslandschaft - Methoden”. In: *VDA Band 4* (2005).
- [85] D. Wu, S. Liu, L. Zhang, J. Terpenney, R. X. Gao, T. Kurfess und J. A. Guzzo. “A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing”. In: *Journal of Manufacturing Systems* 43 (2017), S. 25–34. ISSN: 02786125. DOI: 10.1016/j.jmsy.2017.02.011.
- [86] J. Um, V. Gezer, A. Wagner und M. Ruskowski. “Edge Computing in Smart Production”. In: *Advances in Service and Industrial Robotics*. Hrsg. von K. Berns und D. Görges. Bd. 980. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, S. 144–152. ISBN: 978-3-030-19647-9. DOI: 10.1007/978-3-030-19648-6_17.
- [87] R. Obermaier, Hrsg. *Handbuch Industrie 4.0 und Digitale Transformation: Betriebswirtschaftliche, technische und rechtliche Herausforderungen*. Wiesbaden: Springer Gabler, 2019. ISBN: 978-3-658-24575-7.
- [88] T. Kaufmann. *Geschäftsmodelle in Industrie 4.0 und dem Internet der Dinge: der Weg vom Anspruch in die Wirklichkeit*. essentials. Wiesbaden: Springer Vieweg, 2015. ISBN: 978-3-658-10272-2.
- [89] P. Gölzer, P. Cato und M. Amberg. “Data processing requirements of Industry 4.0 - use cases for Big Data applications”. In: 2015. ISBN: 978-3-00-050284-2.
- [90] VDI/VDE. *Thesen und Handlungsfelder - Cyber-Physical Systeme : Chancen und Nutzen aus Sicht der Automation*. 2013.
- [91] M. Broy und D. A. der Technikwissenschaften, Hrsg. *Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme*. Berlin: Springer, 2010. ISBN: 978-3-642-14901-6.
- [92] H. Kagermann, W. Wahlster und J. Helbig. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*. Techn. Ber. Frankfurt/Main, 2013. URL: https://www.bmbf.de/files/Umsetzungsempfehlungen_Industrie4_0.pdf.
- [93] VDI/VDE. *Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*. Techn. Ber. Düsseldorf, 2015.
- [94] F. Bonomi, R. Milito, J. Zhu und S. Addepalli. “Fog computing and its role in the internet of things”. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12*. Hrsg. von M. Gerla und D. Huang. New

LITERATUR

- York, New York, USA: ACM Press, 2012, S. 13. ISBN: 9781450315197. DOI: 10.1145/2342509.2342513.
- [95] S. Salkic, B. C. Ustundag, T. Uzunovic und E. Golubovic. “Edge Computing Framework for Wearable Sensor-Based Human Activity Recognition”. In: Bd. 83. Cham: Springer International Publishing, 2020, S. 376–387. ISBN: 978-3-030-24985-4. DOI: 10.1007/978-3-030-24986-1_30.
- [96] I. Foster, Y. Zhao, I. Raicu und S. Lu. “Cloud Computing and Grid Computing 360-Degree Compared”. In: *2008 Grid Computing Environments Workshop*. IEEE, 2008, S. 1–10. ISBN: 978-1-4244-2860-1. DOI: 10.1109/GCE.2008.4738445.
- [97] P. M. Mell und T. Grance. *The NIST definition of cloud computing*. Gaithersburg, MD: National Institute of Standards and Technology, 2011. DOI: 10.6028/NIST.SP.800-145.
- [98] K. Huang und B. Shen. “Service deployment strategies for efficient execution of composite SaaS applications on cloud platform”. In: *Journal of Systems and Software* 107 (2015), S. 127–141. ISSN: 01641212. DOI: 10.1016/j.jss.2015.05.050.
- [99] D. Coupek, A. Lechler und A. Verl. “Cloud-Based Control Strategy: Downstream Defect Reduction in the Production of Electric Motors”. In: *IEEE Transactions on Industry Applications* 53.6 (2017), S. 5348–5353. ISSN: 0093-9994. DOI: 10.1109/TIA.2017.2732340.
- [100] W. Shi, J. Cao, Q. Zhang, Y. Li und L. Xu. “Edge Computing: Vision and Challenges”. In: *IEEE Internet of Things Journal* 3.5 (2016), S. 637–646. DOI: 10.1109/JIOT.2016.2579198.
- [101] V. Gazis, A. Leonardi, K. Mathioudakis, K. Sasloglou, P. Kikiras und R. Sudhaakar. “Components of fog computing in an industrial internet of things context”. In: *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking - Workshops (SECON Workshops)*. IEEE, 2015, S. 1–6. ISBN: 978-1-4673-7392-0. DOI: 10.1109/SECONW.2015.7328144.
- [102] S. Yi, C. Li und Q. Li. “A Survey of Fog Computing”. In: *Proceedings of the 2015 Workshop on Mobile Big Data - Mobidata '15*. Hrsg. von Q. Li und D. Xuan. New York, New York, USA: ACM Press, 2015, S. 37–42. ISBN: 9781450335249. DOI: 10.1145/2757384.2757397.
- [103] *IoT Edge: Microsoft Azure*. URL: <https://azure.microsoft.com/de-de/services/iot-edge/> (besucht am 24.12.2019).

-
- [104] K. Gremban. URL: <https://docs.microsoft.com/de-de/azure/iot-edge/iot-edge-runtime> (besucht am 24.12.2019).
- [105] S. Danielson. URL: <https://docs.microsoft.com/de-de/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops> (besucht am 27.05.2010).
- [106] S. Deng. URL: <https://docs.microsoft.com/de-de/azure/azure-functions/functions-overview> (besucht am 27.05.2010).
- [107] J. D. Drake und J. C. Worsley. *Practical PostgreSQL*. O'Reilly Media, 2002. ISBN: 9781449310103. URL: <https://books.google.de/books?id=52ENWgsOWLUC>.
- [108] 2020. URL: <https://www.ionos.de/digitalguide/server/knowhow/postgresql/> (besucht am 27.05.2020).
- [109] N. Rudakov, T. Eerola, L. Lensu, H. Kälviäinen und H. Haario. “Detection of mechanical damages in sawn timber using convolutional neural networks”. In: *German Conference on Pattern Recognition*. Springer. 2018, S. 115–126. DOI: 10.1007/978-3-030-12939-2_9.
- [110] J. Jing, H. Ma und H. Zhang. “Automatic fabric defect detection using a deep convolutional neural network”. In: *Coloration Technology* (2019). DOI: 10.1111/cote.12394.
- [111] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber und G. Fricout. “Steel defect classification with max-pooling convolutional neural networks”. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2012, S. 1–6. DOI: 10.1109/IJCNN.2012.6252468.
- [112] L. Yi, G. Li und M. Jiang. “An End-to-End Steel Strip Surface Defects Recognition System Based on Convolutional Neural Networks”. In: *steel research international* 88.2 (2017), S. 1600068. DOI: 10.1002/srin.201600068.
- [113] A. Bartler, L. Mauch, B. Yang, M. Reuter und L. Stoicescu. “Automated Detection of Solar Cell Defects with Deep Learning”. In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, S. 2035–2039. DOI: 10.23919/EUSIPCO.2018.8553025.
- [114] D. Soukup und R. Huber-Mörk. “Convolutional neural networks for steel surface defect detection from photometric stereo images”. In: *International Symposium on Visual Computing*. Springer. 2014, S. 668–677. DOI: 10.1007/978-3-319-14249-4_64.

LITERATUR

- [115] S. Faghieh-Roohi, S. Hajizadeh, A. Núñez, R. Babuska und B. De Schutter. “Deep convolutional neural networks for detection of rail surface defects”. In: *2016 International joint conference on neural networks (IJCNN)*. IEEE. 2016, S. 2584–2589. DOI: 10.1109/VPPC46532.2019.8952236.
- [116] X. Wu, K. Cao und X. Gu. “A surface defect detection based on convolutional neural network”. In: *International Conference on Computer Vision Systems*. Springer. 2017, S. 185–194. DOI: 10.1007/978-3-319-68345-4_17.
- [117] T. Wang, Y. Chen, M. Qiao und H. Snoussi. “A fast and robust convolutional neural network-based defect detection model in product quality control”. In: *The International Journal of Advanced Manufacturing Technology* 94.9-12 (2018), S. 3465–3471. DOI: 10.1007/s00170-017-0882-0.
- [118] R. Hartl, F. Vieltorf und M. F. Zaeh. “Correlations between the Surface Topography and Mechanical Properties of Friction Stir Welds”. In: *Metals* 10.7 (2020), S. 890. DOI: 10.3390/met10070890.
- [119] R. Hartl, A. Bachmann, S. Liebl, A. Zens und M. Zäh. “Automated surface inspection of friction stir welds by means of structured light projection”. In: *IOP Conference Series: Materials Science and Engineering* 480 (2019). DOI: 10.1088/1757-899x/480/1/012035. URL: <https://doi.org/10.1088/1757-899x/480/1/012035>.
- [120] R. Hartl, B. Praehofer und M. Zaeh. “Prediction of the surface quality of friction stir welds by the analysis of process data using Artificial Neural Networks”. In: *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications* 234.5 (2020), S. 732–751. ISSN: 1464-4207. DOI: 10.1177/1464420719899685.
- [121] R. Hartl, J. Landgraf, J. Spahl, A. Bachmann und M. Zaeh. “Automated visual inspection of friction stir welds: a deep learning approach”. In: SPIE, 2019. ISBN: 9781510627970. DOI: 10.1117/12.2525947. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11059/2525947/Automated-visual-inspection-of-friction-stir-welds--a-deep/10.1117/12.2525947.full>.
- [122] A. Birlutiu, A. Burlacu, M. Kadar und D. Onita. “Defect Detection in Porcelain Industry based on Deep Learning Techniques”. In: *2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE. 2017, S. 263–270. DOI: 10.1109/SYNASC.2017.00049.

-
- [123] H. Lin, B. Li, X. Wang, Y. Shu und S. Niu. “Automated defect inspection of LED chip using deep convolutional neural network”. In: *Journal of Intelligent Manufacturing* (2018), S. 1–10. DOI: 10.1007/s10845-018-1415-x.
- [124] B. C. F. de Oliveira, A. A. Seibert, H. B. Fröhlich, L. R. C. da Costa, L. B. Lopes, L. A. Iervolino, M. B. Demay und R. C. C. Flesch. “Defect inspection in stator windings of induction motors based on convolutional neural networks”. In: *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2018, S. 1143–1149. DOI: 10.1109/INDUSCON.2018.8627172.
- [125] J. Park, W. An und D. Kang. “Convolutional Neural Network Based Surface Inspection System for Non-patterned Welding Defects”. In: *International Journal of Precision Engineering and Manufacturing* 20.3 (2019), S. 363–374. DOI: 10.1007/s12541-019-00074-4.
- [126] D. Mery. “Aluminum Casting Inspection Using Deep Learning: A Method Based on Convolutional Neural Networks”. In: *Journal of Nondestructive Evaluation* 39.1 (2020), S. 14. ISSN: 0195-9298. DOI: 10.1007/s10921-020-0655-9.
- [127] Z. Mo und W. Chen L.and You. “Identification and Detection of Automotive Door Panel Solder Joints based on YOLO”. In: *2019 Chinese Control And Decision Conference (CCDC)*. IEEE, 3.06.2019 - 05.06.2019, S. 5956–5960. ISBN: 978-1-7281-0106-4. DOI: 10.1109/CCDC.2019.8833257.
- [128] J. Redmon und A. Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: <http://arxiv.org/abs/1804.02767>.
- [129] S. Ren, K. He, R. B. Girshick und J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015). arXiv: 1506.01497. URL: <http://arxiv.org/abs/1506.01497>.
- [130] J. Ooi, L. C. Tay und W. K. Lai. “Bottom-hat filtering for Defect Detection with CNN Classification on Car Wiper Arm”. In: *2019 IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, 2019, S. 90–95. ISBN: 978-1-5386-7563-2. DOI: 10.1109/CSPA.2019.8696080.
- [131] C. G. Pachón Suescún, J. O. Pinzón Arenas und R. Jiménez Moreno. “Detection of Scratches on Cars by Means of CNN and R-CNN”. In: *International Journal on Advanced Science, Engineering and Information Technology* 9.3 (2019), S. 745. ISSN: 2088-5334. DOI: 10.18517/ijaseit.9.3.6470.

LITERATUR

- [132] C.-A. Hoffmann. “Komplexitätsmanagement in der Automobilindustrie”. In: *Methodik zur Steuerung modularer Produktbaukästen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2018, S. 27–38. ISBN: 978-3-658-20561-4. DOI: 10.1007/978-3-658-20562-1_3.
- [133] Y. Nie, S. Xia und Y. Wu. “Wheel classification using convolutional neural networks”. In: *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2018, S. 515–520. ISBN: 978-1-5386-7255-6. DOI: 10.1109/YAC.2018.8406429.
- [134] E. Abele, J. Hohenstein, P. Pfeiffer und E. Wühl. “Wandel im PKW-Antriebsstrang: Auswirkungen auf zukünftige Produktionskonzepte”. In: *Maschinenbau und Metallbearbeitung, Kuhn Fachverlag, Villingen-Schwenningen* (2009), S. 12–16. URL: <http://tubiblio.ulb.tu-darmstadt.de/35823/>.
- [135] Fraunhofer ISI, VDW und VDMA. *Online-Umfrage "Veränderungen des Kfz-Antriebsstrangs"*. Düsseldorf, 2010.
- [136] F. J. Ordóñez und D. Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. In: *Sensors (Basel, Switzerland)* 16.1 (2016). DOI: 10.3390/s16010115.
- [137] A. Mayr u. a. “Machine Learning in Production – Potentials, Challenges and Exemplary Applications”. In: *Procedia CIRP* 86 (2019), S. 49–54. ISSN: 22128271. DOI: 10.1016/j.procir.2020.01.035.
- [138] M. Weigelt, A. Mayr, J. Seefried, P. Heisler und J. Franke. “Conceptual design of an intelligent ultrasonic crimping process using machine learning algorithms”. In: *Procedia Manufacturing* 17 (2018), S. 78–85. ISSN: 23519789. DOI: 10.1016/j.promfg.2018.10.015.
- [139] A. Shmilovici. “Support Vector Machines”. In: *Data Mining and Knowledge Discovery Handbook*. Hrsg. von O. Maimon und L. Rokach. Bd. 13. New York: Springer-Verlag, 2005, S. 257–276. ISBN: 0-387-24435-2. DOI: 10.1007/0-387-25465-X_12.
- [140] L. Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), S. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.
- [141] S. Ougiaroglou, A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos und T. Welzer-Druzovec. “Adaptive k-Nearest-Neighbor Classification Using a Dynamic Number of Nearest Neighbors”. In: *Advances in Databases and Information Systems*. Hrsg. von Y. Ioannidis, B. Novikov und B. Rachev. Bd. 4690. Lecture Notes

- in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 66–82. ISBN: 978-3-540-75184-7. DOI: 10.1007/978-3-540-75185-4_7.
- [142] O. I. Abiodun, A. Jantan, E. Omolara A, K. V. Dada, N. A. Mohamed und H. Arshad. “State-of-the-art in artificial neural network applications: A survey”. In: *Heliyon* 4.11 (2018), e00938. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2018.e00938.
- [143] A. Mayr, B. Lutz, M. Weigelt, T. Gläkel, D. Kiskalt, M. Masuch, A. Riedel und J. Franke. “Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings”. In: *2018 8th International Electric Drives Production Conference (EDPC)*. IEEE, 2018, S. 1–7.
- [144] Y. C. Liang, W. D. Li, X. Lu und S. Wang. “Fog computing and convolutional neural network enabled prognosis for machining process optimization”. In: *Journal of Manufacturing Systems* 52 (2019), S. 32–42. ISSN: 02786125. DOI: 10.1016/j.jmsy.2019.05.003.
- [145] C. K. M. Lee, Y. Z. Huo, S. Z. Zhang und K. K. H. Ng. “Design of a Smart Manufacturing System With the Application of Multi-Access Edge Computing and Blockchain Technology”. In: *IEEE Access* 8 (2020), S. 28659–28667. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2972284.
- [146] Y. Lu, X. Hu und Y. Su. “Framework of industrial networking sensing system based on edge computing and artificial intelligence”. In: *Journal of Intelligent & Fuzzy Systems* 38.1 (2020), S. 283–291. ISSN: 10641246. DOI: 10.3233/JIFS-179403.
- [147] Y. Wang, K. Hong, J. Zou, T. Peng und H. Yang. “A CNN-Based Visual Sorting System With Cloud-Edge Computing for Flexible Manufacturing Systems”. In: *IEEE Transactions on Industrial Informatics* 16.7 (2020), S. 4726–4735. ISSN: 1551-3203. DOI: 10.1109/TII.2019.2947539.
- [148] D. Hou, T. Liu, Y.-T. Pan und J. Hou. “AI on edge device for laser chip defect detection”. In: IEEE, 2019, S. 0247–0251. ISBN: 978-1-7281-0554-3. DOI: 10.1109/CCWC.2019.8666503.
- [149] S. Trinks und C. Felden. “Image Mining for Real Time Fault Detection within the Smart Factory”. In: *2019 IEEE 21st Conference on Business Informatics (CBI)*. IEEE, 2019, S. 584–593. ISBN: 978-1-7281-0650-2. DOI: 10.1109/CBI.2019.00074.

LITERATUR

- [150] C.-C. Lin, D.-J. Deng, Y.-L. Chih und H.-T. Chiu. “Smart Manufacturing Scheduling With Edge Computing Using Multiclass Deep Q Network”. In: *IEEE Transactions on Industrial Informatics* 15.7 (2019), S. 4276–4284. ISSN: 1551-3203. DOI: 10.1109/TII.2019.2908210.
- [151] A. Luckow u. a. “Artificial Intelligence and Deep Learning Applications for Automotive Manufacturing”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 10.12.2018 - 13.12.2018, S. 3144–3152. ISBN: 978-1-5386-5035-6. DOI: 10.1109/BigData.2018.8622357.
- [152] J. Queiroz, P. Leitão, J. Barbosa, E. Oliveira und G. Garcia. “An Agent-Based Industrial Cyber-Physical System Deployed in an Automobile Multi-stage Production System”. In: *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*. Cham: Springer International Publishing, 2020, S. 379–391. DOI: 10.1007/978-3-030-27477-1_29.
- [153] M. Syafrudin, N. Fitriyani, G. Alfian und J. Rhee. “An Affordable Fast Early Warning System for Edge Computing in Assembly Line”. In: *Applied Sciences* 9.1 (2019), S. 84. DOI: 10.3390/app9010084.
- [154] J. Zietsch, L. Büth, M. Juraschek, N. Weinert, S. Thiede und C. Herrmann. “Identifying the potential of edge computing in factories through mixed reality”. In: *Procedia CIRP* 81 (2019), S. 1095–1100. ISSN: 22128271. DOI: 10.1016/j.procir.2019.03.259.
- [155] A. Mayr, M. Weigelt, J. von Lindenfels, J. Seefried, M. Ziegler, A. Mahr, N. Urban, A. Kuhl, F. Huttel und J. Franke. “Electric Motor Production 4.0 – Application Potentials of Industry 4.0 Technologies in the Manufacturing of Electric Motors”. In: *2018 8th International Electric Drives Production Conference (EDPC)*. IEEE, 2018, S. 1–13. ISBN: 978-1-7281-0148-4. DOI: 10.1109/EDPC.2018.8658294.
- [156] V. D. I. (VDI). *Entwicklungsmethodik für mechatronische Systeme (VDI 2206): Design methodology for mechatronic systems*. VDI, 2004. URL: <https://books.google.de/books?id=V5BjtAEACAAJ>.
- [157] H. Czichos. *Mechatronik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019. ISBN: 978-3-658-26293-8. DOI: 10.1007/978-3-658-26294-5.
- [158] I. Gräßler und X. Yang. “Entwicklung mechatronischer Systeme und Moduleleichtbau: Anforderungen am Beispiel von Unmanned Aerial Vehicles (UAVs): Fachtagung Mechatronik 2019”. In: (2019). DOI: 10.17619/UNIPB/1-774.

- [159] K. Ehrlenspiel. *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*. 4., aktualisierte Aufl. München: Hanser, 2009. ISBN: 978-3-446-42157-8.
- [160] VDI-2870-Blatt-1:2012-07. *Ganzheitliche Produktionssysteme - Grundlagen, Einführung und Bewertung*. 2012.
- [161] G. M. und Automatisierungstechnik und F. S. und Messsysteme. *Sensoren und Messsysteme 2012 Vorträge der 16. GMA-ITG-Fach- tagung*. 2012.
- [162] M. Rother und S. Kinkel. *Die Kata des Weltmarktführers: Toyotas Erfolgsmethoden*. Frankfurt: Campus Verl, 2009. ISBN: 978-3-593-38996-7.
- [163] B. Jung, S. Schweißer und J. Wappis. *Qualitätssicherung im Produktionsprozess*. Pocket Power 66. München: Hanser, 2013. ISBN: 978-3-446-43454-7.
- [164] W. Geiger und W. Kotte. *Handbuch Qualität: Grundlagen und Elemente des Qualitätsmanagements: Systeme - Perspektiven*. 5., vollst. überarb. und erw. Aufl. Praxis und Studium. Wiesbaden: Vieweg, 2008. ISBN: 978-3-8348-0273-6.
- [165] M. Chen, S. Mao und Y. Liu. “Big Data: A Survey”. In: *Mobile Networks and Applications* 19.2 (2014), S. 171–209. ISSN: 1383-469X, 1572-8153. DOI: 10.1007/s11036-013-0489-0.
- [166] T. Ōno, W. Hof, E. C. Stotko und M. Rother. *Das Toyota-Produktionssystem: das Standardwerk zur Lean Production*. 3., erw. und aktualisierte Aufl. Produktion. Frankfurt New York: Campus Verlag, 2013. ISBN: 978-3-593-39929-4.
- [167] J. Gausemeier, A. Trächtler und W. Schäfer. *Semantische Technologien im Entwurf mechatronischer Systeme*. Hanser eLibrary. München: Hanser, 2013. ISBN: 978-3-446-43845-3.
- [168] K. Schiebold. *Zerstörungsfreie Werkstoffprüfung - Sichtprüfung*. Aufl. 2015. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. ISBN: 978-3-66244-667-6.
- [169] D. D. I. für Normung e.V. *Zerstörungsfreie Prüfung – Terminologie – Begriffe der Sichtprüfung*. Berlin, 2002.
- [170] J. Huber, C. Tammer, D. Schneider, C. Seidel und G. Reinhart. “Non-destructive Quality Testing of Battery Separators”. In: *Procedia CIRP* 62 (2017), S. 423–428. ISSN: 22128271. DOI: 10.1016/j.procir.2016.06.002.
- [171] A. Azevedo und M. Santos. “KDD, semma and CRISP-DM: A parallel overview”. In: Jan. 2008, S. 182–185. URL: https://www.researchgate.net/publication/220969845_KDD_semma_and_CRISP-DM_A_parallel_overview.

LITERATUR

- [172] U. Fayyad, G. Piatetsky-Shapiro und P. Smyth. “From data mining to knowledge discovery in databases”. In: *AI magazine* 17.3 (1996), S. 37. DOI: 10.1609/aimag.v17i3.1230.
- [173] U. Fayyad, G. Piatetsky-Shapiro und P. Smyth. “The KDD process for extracting useful knowledge from volumes of data”. In: *Communications of the ACM* 39.11 (1996), S. 27–34. ISSN: 00010782. DOI: 10.1145/240455.240464.
- [174] U. Shafique und H. Qaiser. “A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA)”. In: *International Journal of Innovation and Scientific Research* 12 (Nov. 2014), S. 2351–8014. ISSN: 2351-8014.
- [175] R. Wirth. “CRISP-DM: Towards a standard process model for data mining”. In: *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*. 2000, S. 29–39.
- [176] V. D. Ingenieure. *Industrielle Bildverarbeitung: Abnahme klassifizierender Bildverarbeitungssysteme*. 2017.
- [177] B. AG. *Fehlersicherheit von attributiv klassifizierenden Kamerasystemen*. 2018.
- [178] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li und L. Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, S. 248–255. ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206848.
- [179] K. Yang, K. Qinami, L. Fei-Fei, J. Deng und O. Russakovsky. “Towards Fairer Datasets: Filtering and Balancing the Distribution of the People Subtree in the ImageNet Hierarchy”. In: (2019). DOI: 10.1145/3351095.3375709. URL: <http://arxiv.org/pdf/1912.07726v1>.
- [180] L. Taylor und G. Nitschke. “Improving deep learning using generic data augmentation”. In: *arXiv preprint arXiv:1708.06020* (2017).
- [181] S. C. Wong, A. Gatt, V. Stamatescu und M. D. McDonnell. “Understanding data augmentation for classification: when to warp?” In: *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016, S. 1–6. DOI: 10.1109/DICTA.2016.7797091.
- [182] H. Xiao, K. Rasul und R. Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747* (2017).

-
- [183] Y. LeCun, L. Bottou, Y. Bengio und P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324. DOI: 10.1109/5.726791.
- [184] K. He, X. Zhang, S. Ren und J. Sun. “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), S. 1904–1916. DOI: 10.1109/TPAMI.2015.2389824.
- [185] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar und L. Fei-Fei. “Large-Scale Video Classification with Convolutional Neural Networks”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 23.06.2014 - 28.06.2014, S. 1725–1732. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.223.
- [186] M. Oquab, L. Bottou, I. Laptev und J. Sivic. “Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, S. 1717–1724. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.222.
- [187] H.-C. Shin, H. R. Roth, M. Gao, L. Le, Z. Xu, I. Nogues, J. Yao, D. Mollura und R. M. Summers. “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning”. In: *IEEE transactions on medical imaging* 35.5 (2016), S. 1285–1298. DOI: 10.1109/TMI.2016.2528162.
- [188] P. Kuhlang, L. Mátyás, X. Sunk und M. Németh. *Produktion und Qualität*. Hanser eLibrary. München: Hanser, 2016. ISBN: 3446449914.
- [189] K. Börner. *Arbeitsorganisation in der automatisierten Produktion: Strategien zur Optimierung von Anlaufprozessen*. Diplomica Verlag, 2009. ISBN: 3836676931.
- [190] F. Himpel. *Marktorientiertes Produkt- und Produktionsmanagement: Zur Gestaltung der Interaktion zwischen Marketing und Produktion*. Gabler edition Wissenschaft. Forum produktionswirtschaftliche Forschung. Wiesbaden: Deutscher Universitätsverlag, 2004. ISBN: 3322816931.
- [191] R. Kruse. *Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze ; 43 Tab.* 1. Aufl. Studium. Wiesbaden: Vieweg + Teubner, 2011. ISBN: 978-3-83481-275-9.

LITERATUR

- [192] A. Riedel, M. Masuch, M. Weigelt, T. Gläβel, A. Kühn, S. Reinstein und J. Franke. “Challenges of the Hairpin Technology for Production Techniques”. In: *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*. IEEE. 2018, S. 2471–2476. DOI: 10.23919/ICEMS.2018.8549105.
- [193] T. Glaessel, J. Seefried und J. Franke. “Challenges in the manufacturing of hairpin windings and application opportunities of infrared lasers for the contacting process”. In: *2017 7th International Electric Drives Production Conference (EDPC)*. IEEE. 2017, S. 1–7. DOI: 10.1109/EDPC.2017.8328150.
- [194] E. Beyer. *Schweißen mit Laser: Grundlagen*. Springer-Verlag, 2013. ISBN: 978-3-642-75759-4.
- [195] J. Bliedtner, H. Müller und A. Barz. *Lasermaterialbearbeitung: Grundlagen-Verfahren-Anwendungen-Beispiele*. Carl Hanser Verlag GmbH Co KG, 2013. ISBN: 978-3-446-42168-4.
- [196] A. Blom, P. Dunias, P. van Engen, W. Hoving und J. de Kramer. “Process spread reduction of laser microspot welding of thin copper parts using real-time control”. In: *Photon Processing in Microelectronics and Photonics II*. Hrsg. von A. Pique u. a. SPIE Proceedings. SPIE, 2003, S. 493. DOI: 10.1117/12.478612.
- [197] A. Heider, P. Stritt, A. Hess, R. Weber und T. Graf. “Process Stabilization at welding Copper by Laser Power Modulation”. In: *Physics Procedia* 12 (2011), S. 81–87. ISSN: 18753892. DOI: 10.1016/j.phpro.2011.03.011.
- [198] P. Schmidt. “Laserstrahlschweißen elektrischer Kontakte von Lithium Ionen Batterien in Elektro- und Hybridfahrzeugen”. Dissertation. München: Technische Universität München, 2015.
- [199] U. Reisinger, S. Olschok und C. Turner. “Laser Beam Welding of Copper: Reduced pressure as key for sound welding of high plate thicknesses”. In: *Laser Technik Journal* 13.5 (2016), S. 34–37. DOI: 10.1002/latj.201600029.
- [200] S. Engler, R. Ramsayer und R. Poprawe. “Process studies on laser welding of copper with brilliant green and infrared lasers”. In: *Physics Procedia* 12 (2011), S. 339–346. DOI: 10.1016/j.phpro.2011.03.142.
- [201] *Dramatischer Preisverfall bei Festplattenspeichern*. 2014. URL: <https://de.statista.com/statistik/daten/studie/36846/umfrage/anteil-der-wirtschaftsbereiche-am-bruttoinlandsprodukt/> (besucht am 15.06.2020).

-
- [202] P. P. Tallon und R. Scannell. “Information life cycle management”. In: *Communications of the ACM* 50.11 (2007), S. 65–69. ISSN: 00010782. DOI: 10.1145/1297797.1297799.
- [203] A. Krizhevsky, I. Sutskever und G. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, S. 1097–1105. DOI: 10.1145/3065386.
- [204] K. Simonyan und A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. URL: <http://arxiv.org/pdf/1409.1556v6>.
- [205] K. He, X. Zhang, S. Ren und J. Sun. *Deep Residual Learning for Image Recognition*. URL: <http://arxiv.org/pdf/1512.03385v1>.
- [206] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke und A. Rabinovich. *Going Deeper with Convolutions*. URL: <http://arxiv.org/pdf/1409.4842v1>.
- [207] I. N Da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni und S. F. dos Reis Alves. “Artificial neural networks”. In: *Cham: Springer International Publishing* (2017).
- [208] S. Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).
- [209] J. Tompson, R. Goroshin, A. Jain, Y. LeCun und C. Bregler. “Efficient Object Localization Using Convolutional Networks”. In: *CoRR* abs/1411.4280 (2014). URL: <http://arxiv.org/abs/1411.4280>.
- [210] D. Guinard. “A Web of Things Application Architecture Integrating the Real-World into the Web”. Dissertation. Zürich: ETH Zürich, 2011.
- [211] R. T. Fielding. “Architectural Styles and the Design of Network-based Software Architectures”. Dissertation. University of California Irvine: Irvine, 2000.
- [212] S. Thattil. 2019. URL: <https://www.yuhiro.de/rest-vs-soap-einfuehrung-unterschiede-vorteile/> (besucht am 28.05.2020).
- [213] *What is a Container?* URL: <https://www.docker.com/resources/what-container> (besucht am 24.12.2019).
- [214] Siemens. *Datenblatt SIMATIC IPC427E (Microbox PC)*. 2020. URL: <https://mall.industry.siemens.com/mall/de/WW/Catalog/Product/?mlfb=6AG4141-.....-.....>