

KI-Projektierung und Modellierung mit Betrachtung der möglichen
Einbindung in die Systemlandschaft eines Industriellen
Sondermaschinenbaus

Bachelorarbeit

für die Prüfung zum
Bachelor of Engineering

des Studiengangs Maschinenbau, Konstruktion und Entwicklung
an der Dualen Hochschule Baden-Württemberg Stuttgart Campus Horb

von

Paul Rienäcker

05.09.2023

Bearbeitungszeitraum:	12 Wochen
Matrikelnummer, Kurs:	5787298, TMB2020KE1
Dualer Partner:	Hugo Kern und Liebers GmbH & Co. KG
Betreuer des Dualen Partners:	B. Eng. Andrej Hein
Gutachter der Dualen Hochschule:	Dipl.-Ing. (FH) Andreas Berndsen

I. Erklärung

Gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW-Technik“ vom 29. September 2017.

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema:

„KI-Projektierung und Modellierung mit Betrachtung der möglichen Einbindung in die Systemlandschaft eines Industriellen Sondermaschinenbaus“

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

Horb, 05.09.2023

Unterschrift

Paul Renäcker

II. Abstract Deutsch

Diese Arbeit widmet sich der Untersuchung zur effektiven Integration von künstlicher Intelligenz in die Systemlandschaft des industriellen Sondermaschinenbaus. Besonderes Augenmerk wird dabei auf das Feld der optischen Bildverarbeitung gelegt, das für die Teileprüfung von hoher Relevanz ist. Die neuesten Fortschritte in der Technik und im Bereich KI werden präsentiert und erläutert. Im Anschluss erfolgt eine gründliche Analyse durch einen strukturierten Ansatz, bei dem verschiedene Methoden betrachtet werden, darunter selbst entwickelte Ansätze, Open-Source-Lösungen sowie kommerzielle Produkte. Dabei steigt die Komplexität der untersuchten Methoden sukzessive an. Durch ein umfassendes Auswahlverfahren werden anschließend die vielversprechendste Option ermittelt. Die Arbeit schließt mit der Ausarbeitung und Bewertung einer potenziellen Integration der ausgewählten Methode in die bestehende Systemlandschaft ab. Dabei werden nicht nur technische Gesichtspunkte berücksichtigt, sondern auch die praktische Umsetzbarkeit in Betracht gezogen.

III. Abstract Englisch

This paper is dedicated to the investigation of effective integration of Artificial Intelligence into the system landscape of industrial custom machinery construction. Special attention is given to the field of Optical Image Processing, which holds significant relevance for component inspection. The latest advancements in AI are presented and explained. Following this, a thorough analysis is conducted through a structured approach, encompassing various methods including proprietary approaches, open-source solutions, as well as commercial products. The complexity of the examined methods progressively increases. Furthermore, a comprehensive selection process identifies the most promising options. The paper concludes with the elaboration and evaluation of a potential integration of the selected method into the existing system landscape. Herein, not only technical aspects are taken into account, but also the practical feasibility is considered.

IV. Inhaltsverzeichnis

I. Erklärung	II
II. Abstract Deutsch.....	III
III. Abstract Englisch.....	III
V. Abkürzungsverzeichnis und Begriffserklärungen	VIII
VI. Abbildungsverzeichnis.....	IX
VII. Tabellenverzeichnis.....	X
VIII. Formelverzeichnis	X
1. Problemstellung	1
2. Ziel der Arbeit.....	1
3. Ist-Stand.....	2
3.1 Die NanoBlade.....	2
3.2 Die Parkscheibe	3
3.3 Die bisherige BV	5
4. Stand der Technik.....	7
4.1 Beurteilung eines binären Klassifikators	7
4.1.1 Statistische Bewertungsparameter	9
4.2 Convolutional Neural Networks	11
4.3 Supervised/Unsupervised optische Anomalieerkennung	12
4.3.1 Supervised Learning	13
4.3.2 Unsupervised Learning.....	13
4.4 Aufteilung von Unsupervised Learning Algorithmen	14
4.4.1 Rekonstruktionsbasiertes Lernen	14
4.4.2 Feature Embedding	15
4.5 EfficientNetV2	16
4.6 Keras CV	17
4.7 Anomalib	18
4.7.1 Funktionsweise von Anomalib	19
4.7.2 Anwenden von Anomalib auf eigenen Datensätzen.....	20
4.8 Merlic und das Deep Learning Tool von MVTec.....	21
4.9 Astrocyte von Teledyne Dalsa.....	22
5. Probleme und Lösungen von KI im industriellen Sondermaschinenbau	23
5.1 Erstellen eines geeigneten Datensatzes	23

5.2 Klare Definition von „Normal“ und „Anormal“	24
5.3 Vermeiden von Klassendominanz.....	24
5.4 Lange Akquirierungszeiträume nötig	25
5.5 Manuelles klassifizieren der Bilder	25
5.6 Kurze Trainingszeiten.....	26
5.7 Kontinuierliche Datenakquirierung	27
5.8 Modularer Betrieb des Systems	28
5.9 Maskieren von Bildern.....	28
5.10 Kurze Inference Zeiten	29
6. Vorteile von KI im industriellen Sondermaschinenbau	30
6.1 Sehr hohe Genauigkeit	30
6.2 Potenziell sehr hohe Zeit- und Kostenersparnis	30
7. Datenakquirierung und Erstellen eines Datensatzes	31
7.1 Klassifizierungsschema NanoBlade	31
7.1.1 Fertiger Datensatz NanoBlade.....	32
7.2 Klassifizierungsschema Parkscheibe	32
7.2.1 Parkscheibe mit Schmutz	34
7.2.2 Schäden auf der Parkscheibe	35
7.3 Fertiger Datensatz Parkscheibe	35
8. Verbesserung des Loggings.....	36
9. Fehleranfälligkeit des bisherigen Systems	37
10. Logik CNN Autoencoder	38
10.1 Training	39
10.2 Ausgabebilder.....	40
10.3 Ergebnisse.....	40
11. Patch CNN Autoencoder	41
11.1 Training	42
11.2 Ausgabebilder.....	43
11.3 Ergebnisse.....	43
12. Mögliche Ursache für unzureichende Performance der Autoencoder	44
13. Klassifizierung durch Supervised Learning.....	45
13.1 Training	46
13.2 Ergebnisse.....	47
14.3 Evaluieren der Anomalib Modelle	48
14.3.1 Testmethodik.....	48

14.3.2 Interpretation der Ergebnisse	49
14.3.3.1 NanoBlade.....	50
14.3.3.2 Parkscheibe	50
14.3.3.3 Vergleich zu EfficientNetV2-S	51
14.3.3.4 Zeiten	52
14.4 Zusammenfassung Anomalib.....	53
15. DLT von MVTec	54
15.1 Interpretation der Ergebnisse	54
16. Astrocyte von Teledyne Dalsa.....	55
16.1 Interpretation der Ergebnisse	56
17. Nutzwertanalyse der unterschiedlichen Lösungen	57
17.1 Zusammenfassung der Nutzwertanalyse.....	58
18. Einbindung von Open-Source in die bestehende Systemlandschaft.....	59
18.1.1 Vorteile Alternative 1	60
18.1.2 Nachteile Alternative 1	60
18.2.1 Vorteile Alternative 2	60
18.2.2 Nachteile Alternative 2.....	60
18.3 Umsetzung von Alternative 2	61
19. Monetäre Bewertung der neuen Methode.....	63
19.1 Zeitersparnis	63
20. Fazit	64
21. Ausblick	64
22. Literatur	65
Anhang.....	A
[A] Flowchart für Klassifizierung der Parkscheibe	A
[B] Rangliste der Anomalib Modelle und ihre Genauigkeit	B
[C] Genauigkeit Anomalib NanoBlades	B
[D] Genauigkeit Anomalib Parkscheibe.....	C
[E] Zeiten Anomalib	C
[F] Genauigkeit DLT NanoBlades	D
[G] Genauigkeit DLT Parkscheibe	D
[H] Zeiten DLT	E
[I] Genauigkeit Astrocyte NanoBlade	E
[J] Genauigkeit Astrocyte Parkscheibe	F
[K] Zeiten Astrocyte.....	F

[L] Nutzwertanalyse der Bildverarbeitungs-Software	G
[M] Zeitersparnis von KI im industriellen Sondermaschinenbau	G
[N] Bildquellen	H
[O] Code Verzeichnis und Dateien	I

V. Abkürzungsverzeichnis und Begriffserklärungen

KI	Künstliche Intelligenz
BV	Bildverarbeitung
IO	In-Ordnung
NIO	Nicht-In-Ordnung
AUROC	Area-Under-Receiver-Operating-Characteristics de. Grenzwertoptimierungskurve
CNN	Convolutional Neural Network, de. Faltendes Neuronales Netzwerk
Supervised Learning	de. Überwachtes Lernen
Unsupervised Learning	de. Unüberwachtes Lernen
Feature Embedding	de. Eingearbeitete Merkmale (in einen Vektor)
Feature Extractor	de. Merkmals Extraktor, Netzwerk das wichtige Informationen aus Bildern gewinnt
Keras CV	Keras Computer Vision
Deep Learning	de. Tiefes Lernen, Beschreibt Modelle, die aus vielen, hintereinander angeordneten Schichten bestehen
Batch Size	de. Chargen-Menge, Anzahl an Bildern die gleichzeitig für das Training verwendet werden
DLT	Deep Learning Tool, Software von MVTec
Inference	de. Schlussfolgerung
Logging	de. Speichern von Bildern
One-Class-Classification	de. Ein-Klassen-Klassifikation, eine unüberwachte Lernmethode
Patches	de. Ausschnitte
Overfitting	de. Überanpassung
FPS	de. Bilder pro Sekunde

VI. Abbildungsverzeichnis

Abbildung 1: Bosch NanoBlade	2
Abbildung 2: Diffuse Durchlichtbeleuchtung	3
Abbildung 3: Die Parkscheibe	4
Abbildung 4: Koaxialbeleuchtung	4
Abbildung 5: Sherlock 7 Benutzeroberfläche	5
Abbildung 6: Konfusions Matrix	7
Abbildung 7: Bildung des AUROC-Wertes	10
Abbildung 8: Funktionsweise eines CNN	12
Abbildung 9: Autoencoder-Netzwerk für rekonstruktionsbasierten Ansatz	14
Abbildung 10: Feature Embedding Methode	15
Abbildung 11: Anomalib Logo	18
Abbildung 12: MVTec Logo	21
Abbildung 13: Teledyne Dalsa Logo	22
Abbildung 14: Angedeutete Maske der Parkscheibe	29
Abbildung 15: Fehlerhafte NanoBlade (links), fehlerfreie NanoBlade (rechts)	31
Abbildung 16: Fehlerklassen Parkscheibe: Fehlerfrei (oben links), Grauflecken (oben rechts), Schmutz (unten links), Schaden (unten rechts)	33
Abbildung 17: Trainingsverlauf Logik CNN Autoencoder	39
Abbildung 18: Ausgabe des Logik CNN Autoencoder	40
Abbildung 19: Funktionsweise von Patches	41
Abbildung 20: Trainingsverlauf Patch CNN Autoencoder	42
Abbildung 21: Ausgabe des Patch CNN Autoencoder	43
Abbildung 22: Trainingsverlauf EfficientNetV2-S	46
Abbildung 23: Konzeption des neuen Bildverarbeitungssystems	61

VII. Tabellenverzeichnis

Tabelle 1: Genauigkeit bisherige BV	37
Tabelle 2: Genauigkeit Logik CNN Autoencoder Parkscheibe	40
Tabelle 3: Genauigkeit Patch CNN Autoencoder Parkscheibe	43
Tabelle 4: Genauigkeit EfficientNetV2-S Parkscheibe	47

VIII. Formelverzeichnis

Formel 1: Präzision.....	9
Formel 2: Trefferquote	9
Formel 3: F1-Metrik.....	10

1. Problemstellung

Die rasante Entwicklung von künstlicher Intelligenz (KI) in den letzten Jahren hat große Teile der Industrie maßgeblich verändert. Viele Unternehmen stehen nun vor der Frage, wie sie die neusten KI-Lösungen gewinnbringend in ihre Fertigung integrieren sollen. Der Bereich des industriellen Sondermaschinenbaus in Kern-Liebers steht genau vor dieser Frage. Einen möglichen Ansatz für KI bildet der Bereich der Bildverarbeitung (BV). Hier werden Teile in Prüfautomaten mit Kameras geprüft und anhand von Rahmenbedingungen aussortiert. Aufgrund des umfangreichen Angebots von kommerziellen sowie auch Open-Source-Lösungen besteht Unklarheit, welches System in Zukunft die BV bei Kern-Liebers auf den neusten Stand der Technik bringen soll.

2. Ziel der Arbeit

Das Hauptziel dieser Arbeit besteht darin, aktuelle Technologien im Bereich der BV zu erforschen, zu evaluieren und anschließend, basierend auf den Ergebnissen, die optimale Lösung für Kern-Liebers zu ermitteln. Hierfür werden beispielhaft zwei Bauteile aus dem Produktportfolio von Kern-Liebers untersucht. Auf Grundlage der Erkenntnisse, die in dieser Arbeit erarbeitet werden, soll die Entscheidungsfindung für eine mögliche Implementierung neuer Algorithmen erleichtert werden. Des Weiteren werden potenzielle Herausforderungen bei der Anwendung dieser Algorithmen beleuchtet, und es werden Handlungsempfehlungen entwickelt, die einen reibungslosen Betrieb dieser Algorithmen sicherstellen sollen.

3. Ist-Stand

Um die bestehende BV zu verbessern, ist es nötig, eine umfassende Untersuchung des bestehenden Systems durchzuführen. Hierfür werden die zwei Produkte, anhand welcher die neuen Methoden evaluiert werden sollen, erklärt und weitere wichtige Sachverhalte festgehalten.

3.1 Die NanoBlade

Die Bosch NanoBlade ist ein äußerst wichtiges Produkt für Kern-Liebers. Mit ihnen können „ungeübte Heimwerker besonders einfach, vibrationslos und damit weniger ermüdend sägen“ [3]. Wie Bosch selbst beziffert, stecken „Mehr als 40 Patentanmeldungen [...] in dem Gerät“ [3]. Sie spiegelt die schon vorhandenen Errungenschaften einer Kettensäge und transferiert sie in einen kleineren Maßstab. So kann eine Bosch NanoBlade, anders wie ihr Konkurrent, die Stichsäge, vibrationslos und sehr genau arbeiten [3]. Bemerkenswert an diesem Produkt ist, dass es vollständig im Stammwerk von Kern-Liebers gefertigt wird. Das schließt das Stanzen, Härten, Schleifen, Nieten der Einzelteile und die Endmontage mit ein.

Die Bosch NanoBlade ist eine komplexe Baugruppe, die genaustens gefertigt werden muss, um die gewünschte Funktion zu gewährleisten.

Eine Bosch NanoBlade besteht aus insgesamt 12 einzelnen Komponenten, dabei zeigt Abb. 1 den prinzipiellen Aufbau der NanoBlade. Von den 12 Komponenten werden 10 von Kern-Liebers im Standort Schramberg Sulgen gefertigt. Es werden verschiedene Ausführungen von NanoBlades gefertigt, die sich in der Länge und der Schnittgeschwindigkeit unterscheiden.



Abbildung 1: Bosch NanoBlade

Die BV setzt bei der Feder an. Hier wird der Sitz der Feder kontrolliert, um später einen einwandfreien Betrieb zu gewährleisten. Die Feder spannt die Kette und gleicht Vibrationen während des Sägens aus. Sie ist ein essenzieller Bestandteil der NanoBlade. Befindet sich die Feder nicht auf ihrer vorgesehenen Position, könnte die Kette herausfallen, was zu einem Defekt der NanoBlade führen würde.

Die Herausforderung bei der Feder liegt an ihrem Drehwinkel. Die Feder kann sich auf ihrer Position um ihre eigene Achse drehen, und dabei die Funktion der NanoBlade nicht beeinflussen. Dieses Problem ist durch konventionelle BV schwer zu identifizieren, wobei neue KI basierte Algorithmen besser auf dieses Problem reagieren sollten.

Mit Hilfe der diffusen Durchlichtbeleuchtung, siehe Abb. 2, lassen sich Umrisse und Bohrungen äußerst präzise überprüfen. Diese spezielle Beleuchtungsmethode wird bei der Prüfung der NanoBlade angewendet. Hierbei wird eine diffuse Beleuchtung hinter dem Objekt platziert, wodurch das entstehende Schattenbild von der Kamera erfasst wird [1]. Das Werkstück erscheint in diesem Fall dunkel, während die Bohrungen weiß hervortreten. Durch den starken Kontrast lassen

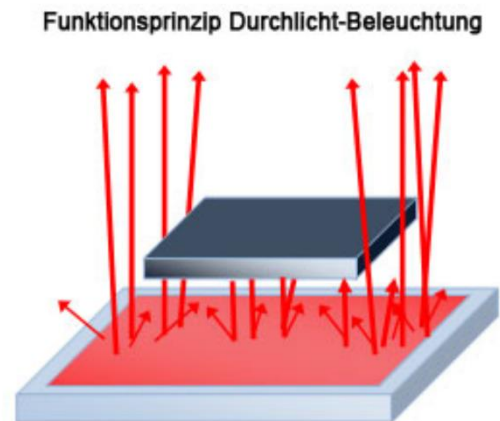


Abbildung 2: Diffuse Durchlichtbeleuchtung

sich Abmessungen des Werkstücks gut abmessen. Es ist jedoch wichtig, einen geringen Abstand zwischen der Beleuchtung und dem Werkstück zu vermeiden, um die Bildqualität nicht durch Streulicht zu beeinträchtigen. Durch die starken Kontraste und fehlenden Grauwerte, welche von dieser Beleuchtungsmethode erzeugt wird, wird hier die Hypothese aufgestellt, dass KI in diesem Fall eine bessere Klassifikationsgenauigkeit erreichen wird.

3.2 Die Parkscheibe

Die Parkscheibe ist ein komplexes Stanzteil, das aufgrund seines Einsatzes in einem Getriebe höchster Qualität entsprechen muss.

Ein Mehrfachwerkzeug stanzt die äußere Form, wobei hier auch direkt die Biegungen entstehen. Anschließend entfernt ein Gleitschleifprozess kleinste Schmutzpartikel und glättet zusätzlich die Oberfläche. In der Mitte der Parkscheibe befindet sich ein runder, eingepresster Einsatz aus Metall. Der obere Teil der Parkscheibe wird später bei einem Kunden mit Magnetpolymer umspritzt, bevor es final in einem Fahrzeug-Getriebe eingebaut wird. Die Parkscheibe kann in Abb. 3 betrachtet werden.



Abbildung 3: Die Parkscheibe

Die Herausforderung bei der Parkscheibe liegt in der Oberfläche. Kameras überprüfen Sie auf Schmutz und Schäden. Schmutz lässt sich auf unzureichendes Gleitschleifen zurückführen, wobei Schäden oft vom Stanzprozess hervorgerufen werden. Die Belichtung verfügt über eine automatische Regelung, welche die Helligkeit anhand der Oberflächenspiegelung reguliert. Diese Helligkeit wird über mehrere Bauteile hinweg gemittelt und langsam eingestellt. Jedoch erweist sich dieses System als fehleranfällig, wenn stark verdunkelte oder reflektierende Teile im Wechsel geprüft werden. Die BV hat mit diesen Helligkeitsschwankungen Probleme, weshalb sich hier der Einsatz eines neuen Algorithmus lohnen könnte.

Die Parkscheibe wird mit einer Standardkoaxialbeleuchtung geprüft. Sie wird für flache, reflektierende Oberflächen eingesetzt. Das Licht wird mithilfe eines Strahlenteilers oder halbdurchlässigen Spiegels auf das Objekt gerichtet, während die Kamera durch diesen Strahlenteiler oder Spiegel hindurchblickt [2]. Ein klarer Vorteil dieser Methode liegt in der gleichmäßigen Ausleuchtung und ihrer Unempfindlichkeit gegenüber reflektierenden Oberflächen (Abb. 4).

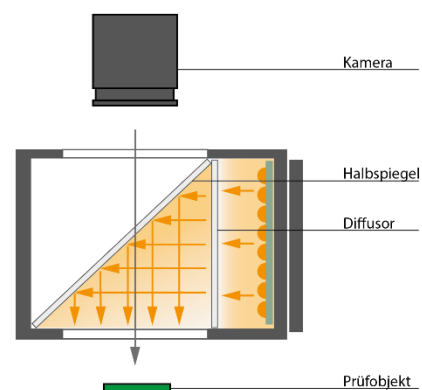


Abbildung 4: Koaxialbeleuchtung

In dieser Beleuchtungssituation erscheint das Objekt in der Regel hell. Analog zur Durchlichtbeleuchtung werden bei dieser Methode, ähnlich wie bei der Untersuchung von Schattenbildern, die äußeren Maße des Werkstücks ermittelt und Bohrungen überprüft. Eine zusätzliche Anwendungsmöglichkeit ist die Bewertung der Helligkeit. Sollte das Werkstück besonders dunkel erscheinen, könnte dies auf mögliche Oberflächenfehler hinweisen. Durch die Grauwerte und reflektierenden Oberflächen wird hier die Hypothese aufgestellt, dass die Parkscheibe für KI-Algorithmen eine größere Herausforderung darstellt als die NanoBlade.

3.3 Die bisherige BV

Bei Kern-Liebers werden eine Vielzahl verschiedenster Bildverarbeitungssysteme eingesetzt. Primäre Anwendung finden diese Systeme in Prüfautomaten, welche Kontur, Abmessungen und Oberflächen von Bauteilen prüfen und sicherstellen, dass jedes Bauteil innerhalb seiner Toleranzen liegt.

Diese Systeme werden mithilfe der Software Sherlock 7 von Teledyne Dalsa erstellt und implementiert. Sherlock 7 ist ein Bildverarbeitungsprogramm, das zur automatischen Inspektion und Analyse von visuellen Informationen in industriellen Anwendungen verwendet wird [4]. Es basiert auf fortschrittlichen Bildverarbeitungstechniken und Algorithmen, die speziell entwickelt wurden, um Mustererkennung, Defekterkennung, Qualitätskontrolle und andere Aufgaben durchzuführen.

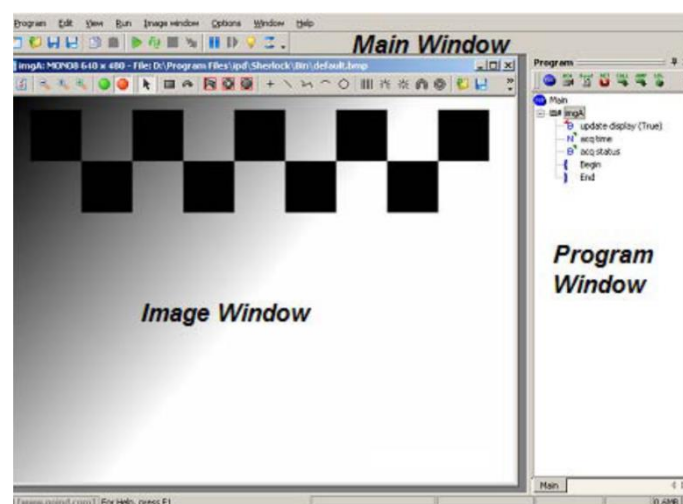


Abbildung 5: Sherlock 7 Benutzeroberfläche

Sherlock zeichnet sich durch seine Benutzerfreundlichkeit aus, was es zu einer hervorragenden Lösung für die Integration in die Fertigungsumgebung macht.

Die Software bietet eine intuitive Benutzeroberfläche (Abb. 5), die es Benutzern ermöglicht, Inspektionsaufgaben ohne umfangreiche Programmierkenntnisse zu erstellen und zu konfigurieren.

Durch die visuelle Darstellung und die vorgefertigten Bildverarbeitungswerkzeuge in der Bibliothek können Benutzer ihre Inspektionsprozesse einfach durch Drag-and-Drop zusammenstellen.

Jedoch treten, durch die statische Natur von Sherlock, oft Herausforderungen auf, insbesondere in Anwendungen, in denen Bauteile variieren können oder die Inspektionsaufgaben komplexer werden. Durch die Abstinenz von KI-Algorithmen kann es schwierig sein, bestimmte Bauteile oder Muster in den Bildern eindeutig zu erkennen, wenn sie sich in Form, Größe oder Farbe unterscheiden. Dies kann zu falschen Ergebnissen führen, was die Zuverlässigkeit der Inspektion beeinträchtigen kann.

Trotz der Herausforderungen mit der NanoBlade und Parkscheibe, bleibt Sherlock durch seine einfache Handhabung und gute Genauigkeit ein wichtiger Bestandteil der Fertigung von Kern-Liebers.

Aus diesem Grund ist es nicht das Ziel dieser Arbeit, Sherlock zu ersetzen, sondern eine wertvolle Alternative zu präsentieren, auf die sich Kern-Liebers betriebsintern stützen kann. Sherlock bleibt eine etablierte und leistungsfähige Bildverarbeitungssoftware, während die vorgeschlagene Alternative eine zusätzliche Option und Flexibilität für die Fertigung bei Kern-Liebers bereitstellt.

4. Stand der Technik

4.1 Beurteilung eines binären Klassifikators

Die Systeme in der Parkscheibe und der NanoBlade sind im Grunde alle binäre Klassifikatoren [24, 25]. Sie teilen die Bauteile in zwei Klassen ein, entweder IO (In-Ordnung) oder NIO (Nicht-In-Ordnung).

Dabei sind diese Systeme niemals perfekt und machen Fehler bei der Klassifizierung. Aus diesem Grund wird eine Matrix erstellt, auf der sich leicht erkennen lässt, wie viele Fehler der Klassifikator macht. Diese Matrix wird als Konfusions Matrix oder Wahrheits Matrix bezeichnet [26]. Eine Konfusions Matrix lässt sich in Abb. 6 erkennen. Teilweise kann diese Matrix auch in transponierter Form vorliegen.

		Wahre Klasse	
		Richtig	Falsch
Vorhergesagte Klasse	Richtig	Richtig Positiv (RP)	Falsch Positiv (FP)
	Falsch	Falsch Negativ (FN)	Richtig Negativ (RN)

Abbildung 6: Konfusions Matrix

Die vier Felder beschreiben die vier möglichen Zustände, die auftreten können. Diese vier Zustände werden im Einzelnen kurz beleuchtet:

1. Richtig Positiv (RP) (links oben)

Wenn das System ein Gutteil (IO-Teil) auch in diese Klasse einordnet, hat es die richtige Entscheidung getroffen. Das Bauteil wird somit richtigerweise als positiv eingestuft.

2. Richtig Negativ (RN) (rechts unten)

Wenn das System ein Schlechtteil (NIO-Teil) auch in diese Klasse einordnet, hat es die richtige Entscheidung getroffen. Das Bauteil wurde also richtigerweise als negativ eingestuft.

Die Fehler bei einem binären Klassifikator sehen folgendermaßen aus:

3. Falsch Positive Fehler (FP) (rechts oben)

Bei diesem Fehler wird ein Bauteil, welches eigentlich NIO ist, als IO eingestuft. Das Bauteil wird somit fälschlicherweise als positiv klassifiziert.

In der Fertigung, kann ein FP-Fehler eine Kundenreklamation verursachen, da ein Teil, welches eigentlich schlecht ist, beim Kunden ankommt.

4. Falsch Negative Fehler (FN) (links unten)

Bei dieser Art an Fehler wird ein Bauteil, welches eigentlich IO ist, als NIO eingestuft. Das Bauteil wurde somit fälschlicherweise als negativ klassifiziert.

Auf die Fertigung bezogen bedeuten FN-Fehler Pseudoausschuss. Das Unternehmen, in diesem Fall Kern-Liebers, sortiert Teile aus, die sich eigentlich noch für den Verkauf geeignet hätten. FN-Fehler schmälern somit dem Gewinn.

4.1.1 Statistische Bewertungsparameter

Ein binärer Klassifikator wird häufig durch eine Konfusionsmatrix veranschaulicht, die es ermöglicht, die Leistung des Modells zu bewerten. Für den Vergleich zwischen verschiedenen Modellen sind klare numerische Kennzahlen erforderlich. Solche Klassifikatoren können zwei Arten von Fehlern machen: Entweder werden NIO-Teile fälschlicherweise als IO klassifiziert (FP), oder IO-Teile werden fälschlicherweise als NIO klassifiziert (FN).

Zur Bewertung der Leistung eines Klassifikators können verschiedene Metriken herangezogen werden, darunter die Präzision und die Trefferquote.

Die Präzision (P) beschreibt das Verhältnis der richtig positiven Klassifikationen zu allen positiven Klassifikationen [26]. Sie entspringt folgender Formel:

Formel 1: Präzision

$$P = \frac{RP}{RP + FP}$$

Eine hohe Präzision deutet darauf hin, dass nur wenige NIO-Teile als IO klassifiziert wurden. Die irrtümliche Klassifikation von NIO-Teilen als IO wird intern als „Schlupf“ bezeichnet.

Die Trefferquote (T) hingegen beschreibt das Verhältnis der richtig positiven Klassifikationen zu allen falsch negativen Klassifikationen [26]. Sie wird mit folgender Formel berechnet:

Formel 2: Trefferquote

$$T = \frac{RP}{RP + FN}$$

Eine hohe Trefferquote deutet darauf hin, dass nur wenige IO-Teile als NIO klassifiziert wurden. Diese Art von Fehler wird intern auch als „Pseudoausschuss“ bezeichnet.

Klassifikatoren sollten eine möglichst hohe Präzision sowie auch Trefferquote besitzen. Die Kombination aus diesen beiden Werten wird F1-Metrik genannt [26]. Die F1-Metrik kann folgendermaßen berechnet werden:

Formel 3: F1-Metrik

$$F1 = 2 * \frac{P * T}{P + T}$$

Die F1-Metrik wird sehr gerne verwendet, um Klassifikatoren zu bewerten. Aus diesem Grund werden sämtliche Modelle mit dieser Metrik bewertet.

Zusätzlich zur F1-Metrik gibt es eine weitere wichtige Bewertungsmaßnahme namens Area-Under-Receiver-Operating-Characteristics (AUROC) (de. Grenzwertoptimierungskurve). Die AUROC-Kurve ist ein grafisches Darstellungsinstrument (Abb. 7), das die Leistungsfähigkeit eines Klassifikators über verschiedene Schwellenwerte veranschaulicht [26]. Sie stellt das Verhältnis zwischen den richtig positiven und den falsch positiven Klassifikationen in Abhängigkeit von der Schwellenwertvariation dar. Im Kern vermittelt sie, wie gut die Wahrscheinlichkeitsverteilungen voneinander getrennt sind.

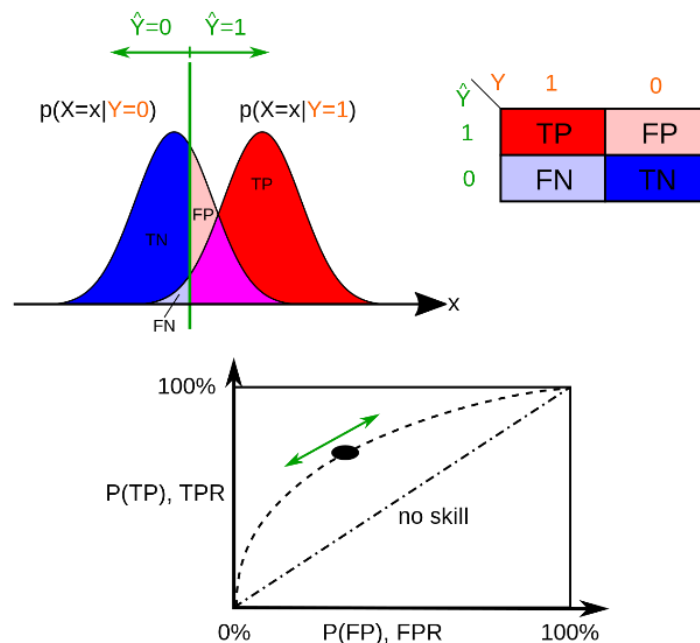


Abbildung 7: Bildung des AUROC-Wertes

Die AUROC-Kurve ist besonders nützlich, da sie unabhängig von der spezifischen Schwellenwertwahl ist und somit die Fähigkeit eines Klassifikators, sowohl positive als auch negative Beispiele zu klassifizieren, umfassend bewertet. Ein AUROC-Wert von 0,5 entspricht einer zufälligen Klassifikation, während ein Wert von 1 eine fehlerfreie Klassifikation anzeigt.

Klassifikatoren mit einer AUROC von 0,5 bis 0,7 gelten als schlecht, von 0,7 bis 0,9 als akzeptabel und über 0,9 als ausgezeichnet [26].

Die Kombination der F1-Metrik und der AUROC-Kurve ermöglicht eine umfassende Beurteilung der Leistung von Klassifikatoren, was von entscheidender Bedeutung ist, um angemessene Entscheidungen in praktischen Anwendungen zu treffen.

Da die manuelle Implementierung der AUROC-Kurve sehr aufwendig ist, wird sie gerne von kommerziellen Produkten oder Komplettlösungen verwendet. Für eigens implementierte Modelle wird die F1-Metrik verwendet.

4.2 Convolutional Neural Networks

Die Entwicklung der Convolutional Neural Networks (CNNs) stellte zweifellos einen bedeutenden Meilenstein in der BV dar. CNNs sind eine spezielle Art von neuronalen Netzwerken, die speziell für die Verarbeitung von visuellen Informationen optimiert sind. Ihr Durchbruch kam mit der Einführung des „LeNet“-Modells von Yann LeCun [8] in 1998 und wurde später durch bahnbrechende Arbeiten wie das „AlexNet“ von Krizhevsky [9] und das „VGGNet“ von Simonyan und Zisserman im Jahr 2014 [10] verstärkt.

Was CNNs besonders leistungsfähig macht, ist ihre Fähigkeit zur hierarchischen Merkmalsextraktion. Das Konzept der Faltungsschichten ermöglicht es, lokale Merkmale wie Kanten oder Texturen zu identifizieren, während die Pooling-Schichten dazu dienen, räumliche Abstraktionen zu erzeugen und die Rechenbelastung zu reduzieren [11]. Diese Schichten können mehrfach hintereinander gestapelt werden, um immer komplexere und abstraktere Merkmale zu erlernen. Ein anschauliches Beispiel hierfür ist in Abb. 8 zu erkennen. Durch die Kombination von vielen Schichten kann das Netzwerk schließlich komplexe visuelle Repräsentationen lernen, die es befähigen, Objekte, Gesichter, Muster und andere komplexe Strukturen in Bildern zu erkennen.

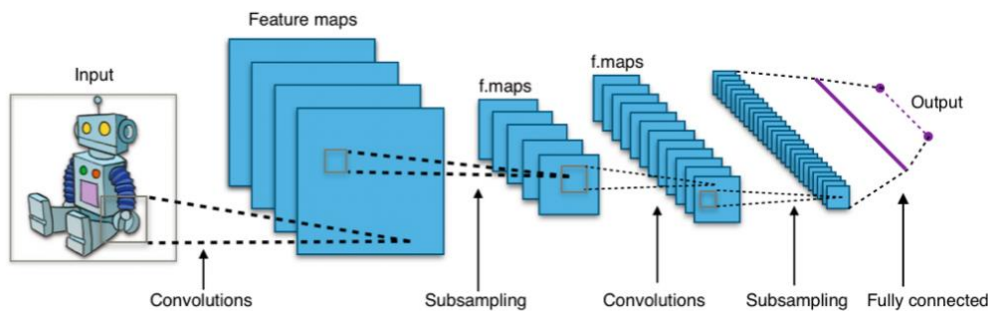


Abbildung 8: Funktionsweise eines CNN

Heutzutage werden CNNs in einer Vielzahl von Anwendungen in der BV eingesetzt. In der industriellen Fertigung werden sie zur Qualitätskontrolle und Defekterkennung eingesetzt, um automatisch fehlerhafte Produkte zu identifizieren und die Produktionsqualität zu verbessern [11].

4.3 Supervised/Unsupervised optische Anomalieerkennung

Supervised Learning (de. Überwachtes Lernen) und Unsupervised Learning (de. Unüberwachtes Lernen) sind zwei grundlegende Ansätze des maschinellen Lernens, die auch in der BV weit verbreitet sind [5]. Diese Methoden sind leistungsstarke Werkzeuge, um komplexe visuelle Informationen zu analysieren und zu verstehen. Es ist daher wichtig, den genauen Unterschied zwischen diesen beiden Ansätzen zu erläutern, damit sie effektiv eingesetzt werden können.

4.3.1 Supervised Learning

Supervised Learning ist eine Lernmethode, bei der das Modell anhand eines annotierten Datensatzes trainiert wird, der Eingabedaten (Bilder) und die zugehörigen Ausgabeetiketten (Klassen oder Labels) enthält. Das Modell lernt, die Beziehung zwischen den Eingabedaten und den Ausgabeetiketten zu erfassen und kann dann neue, nicht annotierte Bilder entsprechend klassifizieren oder vorhersagen. In der BV kann Supervised Learning für Aufgaben wie Objekterkennung, Defekterkennung oder Klassifizierung von Bildern verwendet werden.

Durch die Verwendung von Markierungen oder Etiketten im Training kann das Modell spezifische Merkmale oder Muster auf den Bildern erkennen und zuverlässige Vorhersagen treffen [6].

4.3.2 Unsupervised Learning

Im Gegensatz dazu basiert Unsupervised Learning auf einem unannotierten Datensatz, bei dem es keine klaren Ausgabeetiketten gibt. Das Modell wird angewiesen, Muster und Strukturen in den Eingabedaten zu entdecken, ohne zu wissen, welche Klasse sie darstellen. Unsupervised Learning kann verwendet werden, um ähnliche Objekte oder Regionen in Bildern zu gruppieren oder relevante Merkmale für die Bildrepräsentation zu extrahieren, ohne dass vorherige Wissensgrundlagen vorhanden sind [6].

Unsupervised Learning bietet Vorteile gegenüber Supervised Learning bei der Anomalieerkennung, da es nicht auf annotierte Daten angewiesen ist. Da Anomalien selten und unvorhersehbar sein können, ist es oft schwierig oder kostspielig, ausreichend Daten für das Supervised Learning zu erstellen. Unsupervised Learning Modelle können auf normalen Bildern trainiert werden und anschließend unbekannte Anomalien erkennen.

Aus diesem Grund ist Unsupervised Learning die am meisten verwendete Methode zum Erkennen von Anomalien in der BV [7].

4.4 Aufteilung von Unsupervised Learning Algorithmen

Innerhalb des Unsupervised Learnings lassen sich zwei bedeutende Untergruppen unterscheiden. Rekonstruktionsbasiertes Lernen und Feature Embedding (de. Eingearbeitete Merkmale) [12]. Beide Untergruppen werden kurz beleuchtet.

4.4.1 Rekonstruktionsbasiertes Lernen

Beim rekonstruktionsbasierten Lernen wird das Modell darauf trainiert, die Eingabedaten so zu rekonstruieren, dass sie den Originaldaten möglichst nahekommen.

Das Modell lernt dabei, interne Darstellungen der Eingabedaten zu erstellen und diese dann zur Rekonstruktion der Eingaben zu verwenden [13].

Im Kern dieser Methode steht die Verwendung von CNNs, die als mächtige Feature Extractors (de. Merkmals Extraktor) dienen. Während des Trainingsprozesses werden die CNNs auf einem Datensatz normaler Bilder vortrainiert. Anschließend werden die extrahierten Merkmale als Referenz verwendet, um normale Strukturen und Muster der Bilder zu erfassen.

In der Testphase werden diese vortrainierten CNNs verwendet, um die Merkmale der Testbilder zu extrahieren. Durch den Vergleich der rekonstruierten Testbilder mit den originalen Bildern können Abweichungen oder Anomalien erkannt werden.

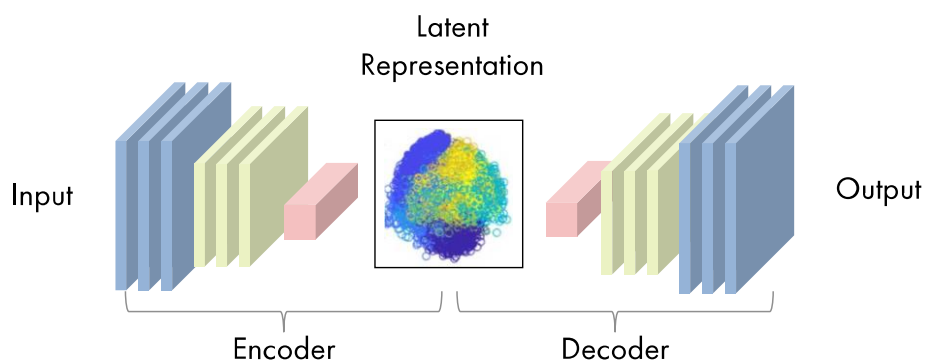


Abbildung 9: Autoencoder-Netzwerk für rekonstruktionsbasierten Ansatz

Die Annahme dabei ist, dass normale Bilder gut von den CNNs rekonstruiert werden können, während Anomalien zu einer schlechteren Rekonstruktion führen. Eine mögliche Netzwerkarchitektur hierfür ist der Autoencoder, welcher auch in Abb. 9 zu sehen ist.

Die erzielte Anomalie-Karte, die aus der Differenz zwischen originalem und rekonstruiertem Bild besteht, zeigt, wie stark sich die rekonstruierten Testbilder von den gelernten normalen Mustern unterscheiden. Anomalien werden anhand dieser Abweichungen lokalisiert und können durch Schwankungen oder Abweichungen von den bekannten Mustern erkannt werden. Prominente Beispiele für solche Algorithmen sind Autoencoder Netzwerke [12] und DREAM [13].

4.4.2 Feature Embedding

Beim Feature Embedding erfolgt die Extraktion der Merkmale durch eine Transformation der hochdimensionalen Daten in einen niedrigdimensionalen Vektorraum [5], ähnlich wie die Encoder-Seite beim rekonstruktiven Ansatz. Im Kern versucht diese Methode, schon im komprimierten Vektorraum potenzielle Anomalien zu entdecken. Ein visuelles Beispiel hierfür ist in Abb. 10 zu erkennen. Hier werden Merkmale aus Bildern in einen komprimierten Raum gemäß ihren Abständen zueinander klassifiziert.

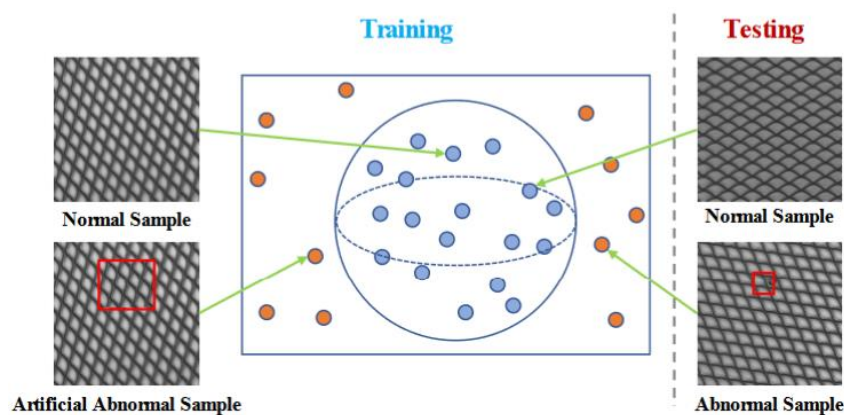


Abbildung 10: Feature Embedding Methode

Die Extraktion der Merkmale zielt darauf ab, semantisch ähnliche oder relevante Datenpunkte im Vektorraum näher zusammenzuführen, während weniger relevante oder anormale Datenpunkte voneinander entfernt werden. Dies ermöglicht es, die inhärenten Muster und Strukturen der Daten auf effiziente Weise zu erfassen und die Dimensionalität der Daten zu reduzieren, ohne dabei wesentliche Informationen zu verlieren.

Die extrahierten Merkmale im niedrigdimensionalen Vektorraum werden als „Embeddings“ bezeichnet und können als neue, aussagekräftigere Darstellung der ursprünglichen Daten betrachtet werden. Dadurch, dass das Netzwerk nur mit normalen Daten trainiert, liegen die normalen extrahierten Merkmale näher beieinander, wobei Anomalien einen größeren Abstand in der komprimierten Repräsentation haben [12].

In der modernen BV haben sich ResNet (Residual Neural Network) und EfficientNetV2 als herausragende Kandidaten für die Feature Extraction etabliert [12]. Ihre hohe Leistung hat dazu beigetragen, die Effizienz und Genauigkeit von Bildverarbeitungsalgorithmen in einer Vielzahl von Anwendungsbereichen zu verbessern. ResNet [14] hat die Art und Weise revolutioniert, wie tiefe CNNs entwickelt werden. Durch die Einführung von Residualblöcken ermöglicht ResNet den effizienten Einsatz extrem tiefer Netzwerkarchitekturen. Tiefere Netze haben den Vorteil, komplexere Merkmale zu lernen, was für die Feature Extraction in hochdimensionalen Datensätzen entscheidend ist.

EfficientNetV2 wiederum nutzt das innovative Konzept des Compound Scaling [15], um die Balance zwischen Leistung und Ressourcenverbrauch zu optimieren. Durch gleichzeitige Skalierung in den Dimensionen Tiefe, Breite und Auflösung kann EfficientNetV2 leistungsstärkere Modelle mit minimalem Ressourcenverbrauch erzeugen. Dies ist besonders wertvoll für Anwendungen mit begrenzten Ressourcen, wie mobile Anwendungen oder eingebettete Systeme.

Im Rahmen des Feature Embeddings haben sich besonders EfficientAD [16], CFlow [17], PatchCore [18] und PaDim [19] als besonders leistungsstarke Algorithmen zur Anomalieerkennung erwiesen.

4.5 EfficientNetV2

Zum Zeitpunkt der Untersuchung hat dieses Netzwerk die höchste Leistung im Vergleich zu anderen Extraktoren [15]. In der BV werden Algorithmen zur Merkmalsextraktion als Rückgrat (eng. Backbones) bezeichnet, da sie das Grundgerüst vieler Bildverarbeitungsalgorithmen bilden.

EfficientNetV2 ist eine verbesserte Architektur, die auf Compound Scaling [31] basiert, einem neuen Ansatz zur gleichzeitigen Skalierung von Modelltiefe, -breite und -Auflösung. Durch die Verwendung von wiederholten Mustern wird die Architektur einfacher und effizienter zu trainieren und zu optimieren [15].

Die besondere Leistungsfähigkeit von EfficientNetV2 gegenüber anderen Netzwerken resultiert aus der besseren Anpassung an verschiedene Datensätze und Aufgaben sowie der verbesserten Ausdrucksfähigkeit, geringerem Berechnungsaufwand und weniger Parameter.

EfficientNetV2-S bietet hier die beste Balance zwischen Größe und Geschwindigkeit, weshalb es für die Untersuchung verwendet wird.

4.6 Keras CV

Keras Computer Vision (CV) ist eine leistungsstarke Bibliothek, die auf dem weit verbreiteten Keras Grundgerüst aufbaut und speziell für die Anwendung von Deep Learning (de. Tiefes Lernen) in der BV entwickelt wurde [32]. Diese fortschrittliche Bibliothek ermöglicht Entwicklern eine effiziente und einfache Bewältigung komplexer Aufgaben in der BV.

Keras CV stellt eine Vielzahl vortrainierter Modelle und vordefinierter Schichten bereit, die es den Nutzern ermöglichen, eigenständig Bildklassifizierungs-, Objekterkennungs- und Segmentierungsaufgaben schnell und unkompliziert zu lösen [33]. Die Integration mit Keras erleichtert auch Anwendern mit begrenzter Erfahrung im Bereich des Deep Learnings den Zugang zu dieser Bibliothek.

Die herausragenden Eigenschaften von Keras CV liegen in seiner Benutzerfreundlichkeit und hohen Anpassungsfähigkeit. Die Bibliothek bietet eine breite Palette von Optimierungsalgorithmen, Verlustfunktionen und Metriken, die speziell für die BV optimiert wurden. Zusätzlich ermöglicht die GPU-Unterstützung eine beschleunigte Verarbeitung großer Mengen an Bilddaten.

EfficientNetV2-S ist ebenfalls in der KerasCV-Bibliothek verfügbar, was diese Bibliothek zu einer bevorzugten Wahl macht.

4.7 Anomalib



Abbildung 11: Anomalib Logo

Anomalib (Abb. 11) ist eine Deep Learning Bibliothek, die eine Sammlung von modernsten Anomalieerkennungsalgorithmen bereitstellt. Die Bibliothek bietet verschiedene sofort einsatzbereite Implementierungen von Algorithmen aus der aktuellen Literatur, sowie Tools zur Entwicklung und Implementierung individueller Modelle. Der Schwerpunkt der Bibliothek liegt auf der Anomalieerkennung in Bildern, wobei das Ziel des Algorithmus darin besteht, anomale Bilder oder anomale Pixelregionen in Bildern eines Datensatzes zu identifizieren. Anomalib wird kontinuierlich mit neuen Algorithmen und Erweiterungen für Training und Inferenz aktualisiert [34].

Die Hauptmerkmale der Bibliothek sind:

- Aufgrund von Open Source auch für industrielle Anwendungen kostenlos.
- Die größte öffentliche Sammlung von sofort einsatzbereiten Deep Learning Anomalieerkennungsalgorithmen und Benchmark-Datensätzen.
- Implementierungen von Modellen, basierend auf PyTorch Lightning, um Code zu reduzieren und die Implementierung auf das Wesentliche zu beschränken.
- Alle Modelle können in das OpenVINO Format exportiert werden, um die Algorithmen auf Intel Produkten zu beschleunigen.
- Eine Reihe von Tools zur schnellen und einfachen Bereitstellung von Standard- oder benutzerdefinierten Anomalieerkennungsmodellen.

Anomalib ist somit sehr gut für die Untersuchung der NanoBlade und der Parkscheibe geeignet. Die Rangliste der Algorithmen, die auf dem MVTec AD Datensatz [20] trainiert wurde, ist auf der offiziellen Website einsehbar. Ebenfalls wurde eine Tabelle dieser Rangliste im Anhang unter [B] beigelegt.

Die Rangliste zeigt, dass EfficientAD derzeit das leistungsstärkste unsupervised Modell zur Erkennung von Anomalien ist, was ebenfalls von anderen Quellen bestätigt wird [16]. Im Folgenden wird nun darauf eingegangen, wie Anomalib funktioniert und wie Modelle mit eigenen Datensätzen trainieren können.

4.7.1 Funktionsweise von Anomalib

Im Kern von Anomalib stehen Konfigurationsdateien, kurz Configs. Mit ihnen können alle wichtigen Einstellungen getätigt werden und anschließend ausgeführt werden. Die wichtigsten Einstellungen beeinflussen:

- Datensätze
 - Auswahl, ob ein eigener Datensatz verwendet werden soll.
 - Speicherort der normalen Trainings und Testdaten, sowie anormalen Testdaten.
 - Definieren der Aufgabe, Segmentierung oder Klassifikation.
 - Festlegen der Bildgröße, sowie Trainings und Test Batch Size (de. Chargen Menge).
 - Wahl des Validierungssatzes (hier immer die Testdaten selbst).
- Modell
 - Auswahl des KI-Modells.
 - Auswahl des Feature Extractors.
 - Andere Modellspezifische Kriterien (meist vorgegeben).
- Metriken
 - Statistische Bewertungsmetriken wie Präzision, Trefferquote, F1 und AUROC.
- Visualisierung
 - In welcher Ausführung und wo Ausgabebilder des Modells gespeichert werden.
- Projekt
 - Speicherpfad des Projekts und Projektname, darunter werden alle Bilder und das Modell gespeichert.
- Speichern des Trainings
 - Speichern des Trainings mit Tensorboard, Comet, WandB oder in einer CSV-Datei.
- Optimierung
 - Auswählen einer Optimierungsbibliothek, keine Auswahl bedeutet Pytorch-Lightning.

- Trainer
 - Alle wichtigen Einstellungen bezüglich des Trainings.
 - Maximale Anzahl an Epochen oder Schritten.
 - Bevorzugtes Rechengerät (CPU oder GPU).
 - Automatisches Finden der Lernrate.

Wird Anomalib standardmäßig durch Klonen des Anomalib GitHub Repos installiert, stehen die modellspezifischen Configs zur Verfügung. Es wird an dieser Stelle von einer übermäßigen Anpassung der Configs gewarnt, da oft die Standardeinstellungen eine gute Grundlage bilden.

4.7.2 Anwenden von Anomalib auf eigenen Datensätzen

Für die Konfiguration von Anomalib zur Verwendung eines benutzerdefinierten Datensatzes werden die oben genannten Konfigurationseinstellungen genutzt. Dabei erfolgt die Anpassung im Abschnitt „Datensätze“. Im Gegensatz zu den Standardeinstellungen wird hier das Format auf „Folder“ geändert, um anzuzeigen, dass ein alternativer Datensatz verwendet wird. Die erforderliche Verzeichnisstruktur ist wie folgt:

- Parkscheiben Datensatz
 - IO
 - 1.bmp
 - 2.bmp
 - ...
 - NIO_Test
 - 1.bmp
 - 2.bmp
 - ...
 - IO_Test
 - 1.bmp
 - 2.bmp
 - ...

Anschließend wird der Pfad im Feld „Path“ angegeben, der zum neuen Datensatz führt. In den Feldern „normal_dir“, „abnormal_dir“ und „normal_test_dir“ werden nun die Unterordner entsprechend den Kategorien normaler Daten, abnormaler Testdaten und normaler Testdaten festgelegt.

In diesem konkreten Beispiel entspricht „normal_dir“ dem Ordner „IO“, „abnormal_dir“ dem Ordner „NIO_Test“ und „normal_test_dir“ dem Ordner „IO_Test“. Von besonderer Bedeutung ist, dass die Ordner „IO_Test“ und „NIO_Test“ die gleiche Anzahl von Bildern enthalten müssen. Während des Trainings lernt das Modell ausschließlich mit den Bildern aus dem Ordner „IO“. In der Testphase wird die Leistung des Modells dann anhand der Bilder aus den Ordnern „IO_Test“ und „NIO_Test“ bewertet.

Um eine unverfälschte Ergebnisbewertung zu gewährleisten und eine mögliche Dominanz einer Klasse zu vermeiden, ist es von hoher Bedeutung, dass diese beiden Klassen die gleiche Anzahl von Bildern umfassen. Die verwendeten Configs befinden sich in einem GitHub-Repo das im Rahmen dieser Arbeit erstellt wurde. Es kann im Anhang unter [O] begutachtet werden.

4.8 Merlic und das Deep Learning Tool von MVTec



Abbildung 12: MVTec Logo

MVTec (Abb. 12), mit Sitz in München, hat sich als ein renommiertes Unternehmen im Feld der industriellen BV etabliert. Seit einigen Jahren bietet MVTec eine breite Palette von Lösungen an, die von Texterkennung bis hin zu Deep Learning Methoden reichen [36].

Die prominentesten Angebote von MVTec sind die Produkte Halcon [37] und Merlic [38]. Halcon zeichnet sich durch eine Benutzeroberfläche aus, die Parallelen zur Struktur von Matlab aufweist, wobei hier ein programmatischer Ansatz verfolgt wird. Im Gegensatz dazu bietet Merlic eine intuitivere Benutzeroberfläche, die mit dem Produkt Sherlock von Teledyne Dalsa vergleichbar ist. Merlic ist durch seine einfache Bedienung und Handhabung ein Favorit gegenüber Halcon.

MVTec stellt Interessenten zusätzlich das Deep Learning Tool (DLT) kostenlos zur Verfügung [39]. Obwohl dieses Tool nicht direkt in der BV angewendet wird, wurde es von MVTec gezielt entwickelt, um ihre neuesten Algorithmen auf Basis von KI zu testen und zu evaluieren. Modelle, die mithilfe des DLT trainiert wurden, können nahtlos in Produkte wie Merlic integriert werden. Daher spiegeln die erzielten Ergebnisse hier eine repräsentative Grundlage für andere MVTec-Produkte wider, da sie im Wesentlichen auf denselben Algorithmen basieren.

Im späteren Verlauf wird ein gezielter Vergleich zwischen Merlic und anderen Produkten erstellt. Hier wird das DLT lediglich zur Ermittlung der KI-Performance von MVTec Produkten verwendet und fließt später in die Bewertung von Merlic mit ein.

4.9 Astrocyte von Teledyne Dalsa



Abbildung 13: Teledyne Dalsa Logo

Teledyne Dalsa (Abb. 13) ist ein bekannter Name in der Branche für BV. Das Unternehmen aus Waterloo, Kanada, bietet neben verschiedensten Kameras und Equipment auch Softwarepakete an [40]. Kern-Liebers nutzt selbst die Software Sherlock 7, welche von Teledyne Dalsa entwickelt wurde. Aufgrund der rasanten Entwicklung im Bereich künstlicher Intelligenz hat Teledyne Dalsa Astrocyte auf den Markt gebracht. Mit diesem Tool ist es möglich, verschiedenste Aufgaben im Bereich der BV mit KI zu lösen. Dieses Tool bietet zudem den Vorteil, dass es aufgrund desselben Herstellers leichter in den Sondermaschinenbau integriert werden kann.

5. Probleme und Lösungen von KI im industriellen Sondermaschinenbau

KI stellt den industriellen Sondermaschinenbau aktuell vor große Herausforderungen. Für einen effektiven Betrieb einer KI basierten Lösung ist es nötig, diese Herausforderungen zu definieren und mögliche Lösungsansätze zu präsentieren. Daher werden nun im Folgenden alle aktuellen Probleme mit KI definiert und versucht diese auf bestmöglichem Weg zu lösen.

5.1 Erstellen eines geeigneten Datensatzes

Die Leistungsfähigkeit von modernen Deep Learning Algorithmen ist maßgeblich von der Zusammenstellung des Datensatzes anhängig [20]. Daher wird in dieser Arbeit eine besondere Sorgfalt auf die korrekte Erstellung eines Datensatzes gelegt.

Zu Beginn ist es wichtig zu verstehen, dass Deep Learning Modelle auf großen, vielfältigen und repräsentativen Datensätzen trainiert werden müssen, um eine hohe Generalisierungsfähigkeit zu erreichen [20]. Diese Datensätze von normalen und anomalen Zuständen müssen aus Bildern bestehen, die in einer realen Produktionsumgebung erfasst wurden. Im Idealfall sollte der Datensatz eine ausreichende Anzahl von Beispielen für verschiedene Arten von Anomalien enthalten, um sicherzustellen, dass das trainierte Modell in der Lage ist, eine Vielzahl von Fehlern zu erkennen.

Zwangsläufig müssen die Datensätze für Deep Learning Algorithmen manuell von Mitarbeitenden klassifiziert werden. Dazu sollten mindestens zwei Personen unabhängig voneinander alle Daten den entsprechenden Klassen zuordnen. Anschließend vergleichen beide Personen ihre Ergebnisse und beraten bei Unstimmigkeiten. Durch diese Redundanz steigt die Qualität des Datensatzes maßgeblich an.

5.2 Klare Definition von „Normal“ und „Anormal“

Die genaue Festlegung der normalen und anomalen Teile im Vorfeld ist von entscheidender Bedeutung, um die Wirksamkeit von Modellen im industriellen Sondermaschinenbau zu gewährleisten. Eine präzise Definition normaler Teile ermöglicht es, ein Modell zu trainieren, das in der Lage ist, tatsächliche Anomalien zuverlässig zu identifizieren, während es normale Variationen und Schwankungen toleriert. Dieser Prozess erfordert ein tiefes Verständnis der Produktionsprozesse und Kundenwünsche. Eine klare Definition der Anomalien ermöglicht eine gezielte Datensammlung, um ausreichend Beispiele für verschiedene Anomaliearten zu erhalten. Dadurch wird das Modell präziser und robuster, was letztendlich zu einer effektiveren Implementierung der Anomalieerkennung führt.

Ein entscheidender Schritt hierbei ist die sorgfältige Beratung mit allen Beteiligten. Oftmals haben verschiedene Mitarbeitende unterschiedliche Auffassungen darüber, was als „normal“ und „anormal“ betrachtet werden sollte. Um zu verhindern, dass wichtige Aspekte übersehen werden und potenziell eine fehlerhafte Klassifizierung erfolgt, ist es von großer Bedeutung, alle möglichen Fehlerklassen zu identifizieren und in Diskussionen eindeutig zu definieren. Durch diese kooperative Herangehensweise wird sichergestellt, dass der Datensatz alle relevanten Informationen enthält und die Grundlage für ein effektives Modell bildet, das den Anforderungen und Gegebenheiten des industriellen Sondermaschinenbaus gerecht wird.

5.3 Vermeiden von Klassendominanz

Eine weitere Herausforderung bei der Erstellung eines Datensatzes liegt darin, dass es oft schwierig ist, ausreichend viele Anomalien zu sammeln, da sie in der industriellen Fertigung selten auftreten können. Dies kann dazu führen, dass der Datensatz unausgewogen ist, wobei die Anzahl der normalen Beispiele die Anzahl der anomalen Beispiele bei weitem übersteigt. Eine unausgewogene Verteilung kann dazu führen, dass das Modell die Anomalien nicht angemessen erkennt, da es aufgrund der Dominanz der normalen Beispiele „voreingenommen“ wird. Hierbei ist wichtig zu erwähnen, dass der gewählte Ansatz, Supervised oder Unsupervised, maßgeblichen Einfluss auf die Verteilung der Daten hat. Bei einem Supervised Ansatz müssen die Klassen im Verhältnis 1:1 vorliegen, wobei bei einer Unsupervised Methode nur mit normalen Daten trainiert wird und Anomalien nur in der Testphase benötigt werden.

5.4 Lange Akquirierungszeiträume nötig

Ein langer Akquirierungszeitraum ist in vielerlei Hinsicht für die Entwicklung von geeigneten Modellen von großer Bedeutung. Ein langer Zeitraum ermöglicht eine umfassende Erfassung saisonaler, zyklischer und langfristiger Trends, die in den Daten auftreten können. Dadurch können Muster und Verhaltensweisen erkannt werden, die möglicherweise nur über einen ausgedehnten Zeitraum sichtbar werden. Es bietet auch die Möglichkeit, selten auftretende Anomalien zu erfassen, die in kürzeren Zeiträumen möglicherweise nicht ersichtlich sind. Darüber hinaus hilft ein langfristiger Datensatz, die Variabilität und Stabilität des normalen Verhaltens zu verstehen, was bei der Festlegung präziser Grenzwerte und Schwellenwerte für die Anomalieerkennung von entscheidender Bedeutung ist. Zudem ermöglicht die Verwendung von langfristigen Daten auch eine realistischere Evaluierung der Modelle, um sicherzustellen, dass sie für den tatsächlichen Einsatz in der industriellen Fertigung gut geeignet sind.

Im Verlauf dieser Arbeit werden die Daten für die Parkscheibe über einen Zeitraum von einem Monat aufgezeichnet. Aufgrund der geringeren Produktionsmenge der NanoBlades wird der Erfassungszeitraum hier auf sieben Wochen ausgeweitet.

5.5 Manuelles klassifizieren der Bilder

Die Anwendung von maschinellem Lernen und Deep Learning erfordert die Verfügbarkeit eines geeigneten Datensatzes, der zuvor von Mitarbeitenden entsprechend klaren Strukturen klassifiziert wurde. Die Erstellung eines solchen Datensatzes kann je nach Komplexität des zu lösenden Problems erheblich Zeit in Anspruch nehmen, bis zu mehreren Tagen. Ein exemplarisches Szenario hierfür ist die Entwicklung einer Parkscheiben-Erkennung, bei der sich das Problem als anspruchsvoller herausstellte als ursprünglich angenommen.

Häufig erfordert die effektive Datensatzgestaltung mehrere Iterationen, in denen verschiedene Ansätze vorbereitet und evaluiert werden. Das Ziel dabei ist, einen Datensatz zu erhalten, der sowohl den Anforderungen der Belegschaft und Kunden entspricht. Die Vereinigung dieser beiden Aspekte kann äußerst herausfordernd und zeitaufwändig sein, insbesondere bei der Lösung eines Klassifizierungsproblems, bei dem alle Beteiligten zufriedengestellt werden sollen.

Veränderungen in der Fertigung, sei es durch Materialwechsel, Umformverfahren oder Oberflächenbehandlungen, erfordern möglicherweise die Erstellung eines neuen Datensatzes. Im Gegensatz zu konventionellen Methoden, bei denen Klassifizierungsgrenzen, unkompliziert, verschoben werden können, ist die Anhebung der Grenzwerte bei maschinellem Lernen begrenzt, da dies die Genauigkeit der Methode beeinträchtigen kann. Falls die Teile in der Fertigung erheblich voneinander abweichen, ist die Neuerstellung des Datensatzes erforderlich.

Es ist wichtig zu betonen, dass der Prozess der Datensatzerstellung nicht als einmalige Aktion angesehen werden darf, sondern vielmehr als fortlaufender Prozess integriert werden muss. Im Falle unerwarteter Fehler auf den Bauteilen ist ein schnelles Handeln erforderlich, um die Produktion wieder auf das normale Niveau zurückzuführen. In dieser Situation ist speziell geschultes Personal unabdingbar, um die Bauteile gemäß den vorgegebenen Kriterien zu klassifizieren. Ein solcher kontinuierlicher Ansatz gewährleistet eine effiziente und präzise Produktionsprozesskontrolle, die den Qualitätsanforderungen gerecht wird.

5.6 Kurze Trainingszeiten

In der industriellen Fertigung, die von Schnelligkeit und Dynamik geprägt ist, ist es äußerst wichtig, dass der Algorithmus für den Fall unerwarteter Fertigungsanomalien optimal vorbereitet ist. Es kann vorkommen, dass der Algorithmus in solchen Situationen alle Teile als anomal klassifiziert, was unerwünscht ist. Um diesem Szenario entgegenzuwirken, ist es entscheidend, dass die Trainingszeit, also die Zeit, die der Algorithmus benötigt, um sich nach einem Fehler wieder anzupassen, minimal ist. Dadurch wird sichergestellt, dass die Algorithmen in der Re-Learning-Phase schnell und effizient arbeiten und nach einem Ausfall rasch wieder einsatzfähig sind. Lange Trainingszeiten von vielen Stunden wären in dieser Hinsicht inakzeptabel, da sie zu erheblichen finanziellen Verlusten in der Fertigung führen könnten. Daher ist ein besonders leistungsfähiger Algorithmus erstrebenswert, um diese Herausforderung zu bewältigen.

5.7 Kontinuierliche Datenakquirierung

Um sicherzustellen, dass im Falle eines Algorithmus-Ausfalls ein effektives Training durchgeführt werden kann, erweist es sich als entscheidend, im Vorfeld kontinuierlich Bilddaten zu speichern. So kann der Algorithmus direkt mit aktuellen Daten trainiert und die Ausfallzeit reduziert werden.

Die fortlaufende Aufzeichnung von Bildern bietet idealerweise die Möglichkeit, auch seltene Muster zu erfassen, die anfangs eventuell fälschlicherweise aussortiert wurden. Diese Integration selten auftretender Muster in den Trainingsdatensatz verbessert die Robustheit und Genauigkeit des Systems erheblich.

Jedoch birgt die Speicherung der Bilddaten gewisse Herausforderungen, insbesondere wenn die verfügbare Rechenleistung begrenzt ist oder die Fertigung sehr hohe Taktzeiten aufweist. In solchen Szenarien besteht die Gefahr, dass die Speicherzeit länger dauert als die verfügbare Taktzeit, was unerwünschte Auswirkungen auf den Produktionsprozess haben kann.

Eine mögliche Lösung besteht darin, die Bilder auf ein kleineres Format zu skalieren, das trotz geringerer Größe ausreichende Informationen für den Algorithmus bereitstellt. Durch diese Maßnahme wird eine effiziente Verarbeitung der Bilddaten gewährleistet, ohne die Leistungsfähigkeit des Algorithmus zu beeinträchtigen. Beispielsweise kann eine Skalierung auf 256x256 Pixel ausreichend sein, da viele Algorithmen zur Anomalieerkennung keine höhere Auflösung erfordern [16, 19, 21].

Um den übermäßigen Speicherbedarf zu begrenzen, kann eine Festlegung auf eine maximale Anzahl von Bildern erfolgen, die ab einem bestimmten Zeitpunkt mit neuen Bildern überschrieben werden. Dieser Ansatz zielt darauf ab, eine kontrollierte und begrenzte Anzahl von Reservebildern zu gewährleisten, um unnötige Speicherbelastungen zu vermeiden. Durch die Festlegung einer maximalen Bildanzahl wird sichergestellt, dass der verfügbare Speicherplatz effizient genutzt wird und keine Ressourcen verschwendet werden.

5.8 Modularer Betrieb des Systems

Deep Learning Algorithmen im Bereich der Anomalieerkennung sind äußerst sensibel gegenüber Abweichungen. Diese Sensitivität ist zweifelslos vorteilhaft, wenn es darum geht, feinste Anomalien zu identifizieren. Jedoch kann sie auch zu einem Nachteil werden, da gelegentlich Muster erkannt werden können, die keine tatsächlichen Anomalien darstellen. Beispielsweise können Anomalien in Bildern erkannt werden, die sich nicht auf den Bauteilen selbst befinden, sondern im Hintergrund auftreten, wie etwa ein Schmutzpartikel auf einem Greifarm.

Angesichts dieser Empfindlichkeit ist es unerlässlich, die Deep Learning Algorithmen parallel oder modular zur normalen Fertigung zu betreiben. Dadurch besteht die Möglichkeit, das System bei Bedarf abzuschalten, während die reguläre Fertigung ungestört fortgesetzt werden kann. Diese Konfiguration gewährleistet, dass potenzielle Fehlalarme oder algorithmische Ausfälle die Produktion nicht beeinträchtigen. Somit wird ein effizienter und zuverlässiger Produktionsprozess sichergestellt, während gleichzeitig die Vorteile der Anomalieerkennung genutzt werden, um die Produktqualität zu erhöhen.

5.9 Maskieren von Bildern

Eine vielversprechende Methode, um das zuvor beschriebene Problem zu minimieren, besteht darin, bei Bedarf Bauteile gezielt zu maskieren. Dieser Prozess des Maskierens beinhaltet das Beibehalten der eigentlichen Bauteile im Bild und das Abdunkeln aller anderen Informationen im Hintergrund. Als Ergebnis wird der Algorithmus nicht mehr in der Lage sein, irrelevante Informationen aus dem Hintergrund für seine Klassifizierung zu verwenden. Die Maske für die Parkscheibe lässt sich in Abb. 14 erkennen. Hier werden alle umliegenden Merkmale geschwärzt, während das Bauteil selbst unverändert bleibt. Die Maske in Abb. 14 ist nur teilweise ausgeführt, um den Effekt besser zur Geltung zu bringen.

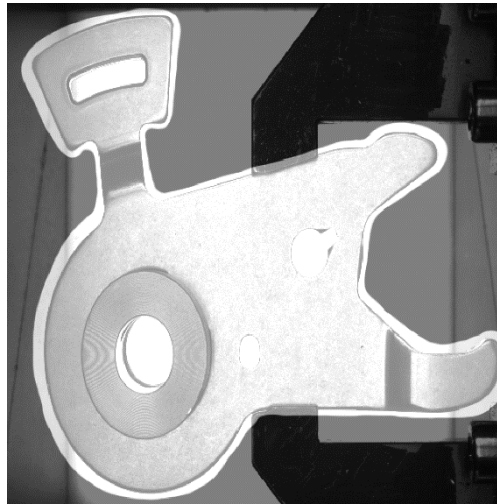


Abbildung 14: Angedeutete Maske der Parkscheibe

Durch das Maskieren der Bauteile wird die Aufmerksamkeit des Algorithmus ausschließlich auf die relevanten Merkmale gelenkt, die für die Anomalieerkennung entscheidend sind. Dies führt zu einer präziseren und zuverlässigeren Klassifizierung, da störende Elemente im Hintergrund eliminiert werden. Darüber hinaus ermöglicht das Maskieren eine bessere Anpassung des Algorithmus an verschiedene Fertigungsbedingungen und reduziert die Wahrscheinlichkeit von Fehlalarmen durch nicht relevante Hintergrundinformationen.

Durch die gezielte Anwendung von Masken wird somit die Leistungsfähigkeit und Effizienz der Anomalieerkennung gesteigert, was zu einer insgesamt verbesserten Produktionskontrolle und Qualitätssteigerung führt.

5.10 Kurze Inference Zeiten

Als Inference (de. Schlussfolgerung) Zeit wird die Zeit bezeichnet, die das Modell für die Verarbeitung eines Bildes benötigt. Bei besonders hohen Taktzeiten oder niedriger Rechenleistung kann die Taktzeit die Inference Zeit überschreiten, was zu Verzögerungen in der Produktion führen kann. Daher ist es äußerst wichtig Algorithmen auszuwählen, welche wenig Rechenleistung und niedrige Inference Zeit benötigen. So kann sichergestellt werden, dass es durch den Algorithmus nicht zu einer Verzögerung der Fertigung kommt.

6. Vorteile von KI im industriellen Sondermaschinenbau

6.1 Sehr hohe Genauigkeit

In den vergangenen Jahren hat sich Deep Learning zunehmend in der BV etabliert, und dies lässt sich durch einen fundamentalen Grund erklären: Die Genauigkeit von Deep Learning Modellen übertreffen deutlich die der herkömmlichen Methoden [5, 7, 12]. Durch den Einsatz von Deep Learning kann ein maßgeschneiderter Datensatz erstellt werden, der das zu lösende Problem für den Algorithmus beschreibt. Im Anschluss daran ist der Algorithmus in der Lage, eigenständig zu lernen und die effektivste Lösungsstrategie für das gegebene Problem zu entwickeln. Dieser Ansatz ermöglicht es, das bestmögliche Ergebnis zu erzielen.

6.2 Potenziell sehr hohe Zeit- und Kostenersparnis

Oft führen vermeintlich einfache Probleme in der BV zu komplexen Lösungsmustern. Die mögliche Anzahl an Fehlerzuständen kann vor allem im industriellen Sondermaschinenbau ein sehr hohes Maß annehmen. Jedes spezifische Fehlermerkmal manuell in einen Algorithmus einzubinden, kann sehr zeitaufwendig und mühselig sein. Wobei die Probleme oft für Menschen schnell und eindeutig zu klassifizieren sind. Hier setzt Deep Learning an.

Wenn der Mensch einen Datensatz erstellt, der eine große Anzahl an Fehlermerkmalen bzw. Zuständen aufweist, kann der Deep Learning Algorithmus dieses Wissen nutzen, um eigenständig zu klassifizieren. Da der Algorithmus die nötigen Merkmale und Eigenschaften aus dem Datensatz selbstständig extrahiert, muss der Mensch nicht eingreifen. Der Algorithmus lernt selbst, wie das Problem am besten zu lösen ist. Somit fällt wertvolle menschliche Arbeitszeit heraus, die anderweitig eingesetzt werden kann.

Hinzu kommt, dass durch die höhere Genauigkeit, Kundenreklamationen und Pseudoausschuss sinken. Kundenreklamationen können im Zweifelsfall viel Zeit und Geld kosten und die Produktion stilllegen. Pseudoausschuss reduziert die Teileausbringung und erhöht die Kosten der Nacharbeit. Daher ist eine höhere Genauigkeit für Kern-Liebers äußerst erstrebenswert.

7. Datenakquirierung und Erstellen eines Datensatzes

Die Datenakquirierung für das Beispiel der NanoBlade und der Parkscheibe erfolgt über mehrere Wochen mithilfe des Logging (de. Speichern von Bildern) -Tools von Sherlock. In diesem Prozess werden die Bilddaten in Originalauflösung auf dem lokalen System gespeichert. Um ein Klassenungleichgewicht zu vermeiden, wird für jedes NIO-Bild im Verhältnis 1:1 ein IO-Bild abgespeichert. Später können die Daten über Fernzugriff aus dem System extrahiert werden.

7.1 Klassifizierungsschema NanoBlade

Das Kamerasystem mit Durchlichtprüfung wird bei der NanoBlade eingesetzt, um den korrekten Sitz der Feder zwischen Scheibe und Halter zu überprüfen. Wenn die Feder nicht ordnungsgemäß positioniert ist, wird das Teil aussortiert. Eine besondere Herausforderung besteht darin, dass die Feder trotz möglicher Verdrehungen auf ihrer Position sitzen kann. Diese Drehungen erschweren die Erkennung erheblich, und in der Vergangenheit war ein beträchtlicher Aufwand erforderlich, um mit konventionellen Methoden eine ausreichende Genauigkeit zu erzielen. Aufgrund der Durchlichtprüfung gibt es in den Bildern nur starke Kontraste, und daher sind die Grauwerte und Abstufungen begrenzt. Dies erleichtert grundsätzlich die Fehlererkennung. Die Untersuchung hat gezeigt, dass ein umfassendes Klassifizierungsschema für die NanoBlade nicht erforderlich ist. Ein menschlicher Beobachter kann den korrekten Sitz der Feder schnell und mit hoher Genauigkeit identifizieren. Die Fehlerarten sind ebenfalls überschaubar, da die Feder entweder korrekt auf ihrer Position sitzt oder teilweise bzw. ganz aus dem Sitz austritt. Die Unterscheidung zwischen diesen Zuständen kann effizient und präzise durchgeführt werden. Ein Beispiel lässt sich in Abb. 15 erkennen.

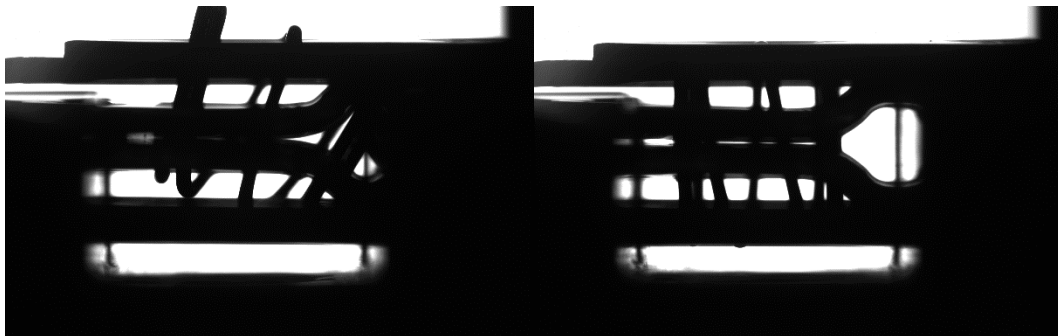


Abbildung 15: Fehlerhafte NanoBlade (links), fehlerfreie NanoBlade (rechts)

7.1.1 Fertiger Datensatz NanoBlade

Der vollständige Datensatz für die NanoBlade besteht aus insgesamt 1000 Trainingsbildern, 500 Testbildern und 680 Validierungsbildern. Während des Trainings wird die Leistungsfähigkeit anhand der Testbilder bewertet. Nach Abschluss des Trainings erfolgt die Bewertung der Performance auf dem Validierungsdatensatz, um die Fähigkeit des Algorithmus zur Verarbeitung unbekannter Daten zu beurteilen.

7.2 Klassifizierungsschema Parkscheibe

Im Zuge der Untersuchung werden sorgfältig verschiedene Klassen der Parkscheibe ermittelt, die insgesamt vier Kategorien umfassen:

1. **Fehlerfrei:** Diese Teile weisen nur wenig oder keinerlei Mängel auf und sind für den Verkauf geeignet.
2. **Grauflecken:** Artefakte aus dem Gleitschleifprozess verursachen eine graue Oberfläche. Die Funktion des Bauteils ist jedoch weiterhin gegeben, weshalb sich diese Teile ebenfalls für den Verkauf eignen.
3. **Schmutz:** Hierbei handelt es sich um Teile, die mittlere bis hin zu große Schmutzpartikeln aufweisen, die sich auf der Oberfläche anhaften und durch unzureichendes Gleitschleifen entstehen können. Diese Teile gelten als Ausschuss und sind nicht für den Verkauf geeignet.
4. **Schaden:** Dieser Typ umfasst Teile, die einen Schaden erlitten haben, unabhängig davon, ob er vom Stanzen oder Gleitschleifen herrührt. Ähnlich wie bei Schmutz sind diese Teile ebenfalls als Ausschuss zu betrachten und für den Verkauf ungeeignet.

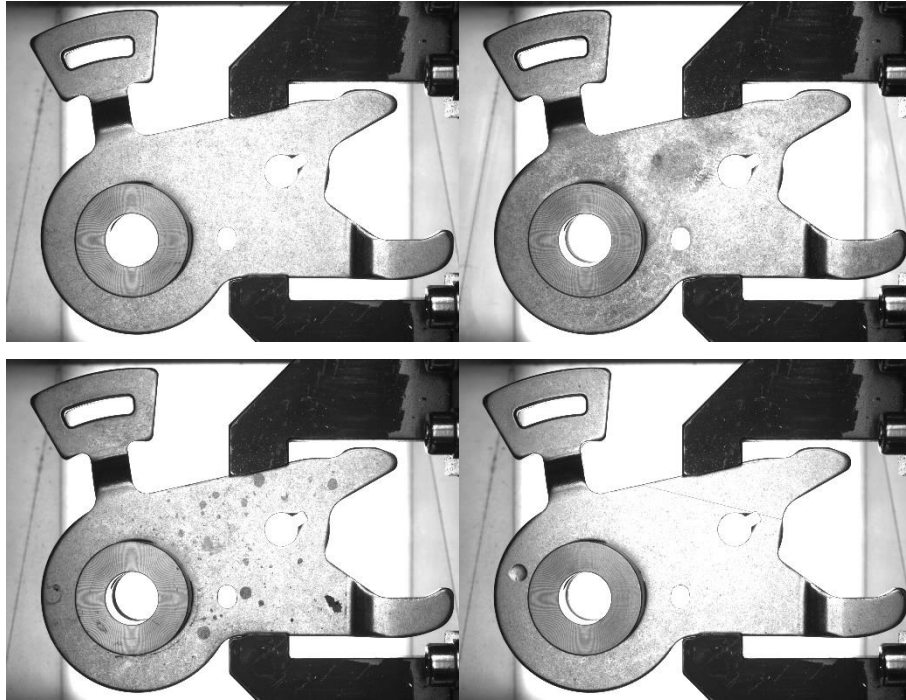


Abbildung 16: Fehlerklassen Parkscheibe: Fehlerfrei (oben links), Grauflecken (oben rechts), Schmutz (unten links), Schaden (unten rechts)

Alle Fehlerklassen lassen sich in Abb. 16 genauer betrachten.

Der rohe Datensatz umfasst insgesamt 4000 Bilder und erfordert nun eine manuelle Klassifizierung durch Mitarbeitende. Diese Vorgehensweise ist von entscheidender Bedeutung, da sie zwei wesentliche Ziele verfolgt.

Erstens ermöglicht sie die Bestimmung der Genauigkeit des bisherigen Systems, indem die tatsächliche Abweichung zwischen den automatisch zugewiesenen Klassen und den manuell festgelegten Klassen ermittelt wird.

Zweitens ist die manuelle Klassifizierung essenziell, um einen korrekten Datensatz zu erstellen, der für jede Klasse, IO und NIO, Teile enthält. Durch die präzise Zuordnung der einzelnen Bauteile zu den entsprechenden Klassen wird ein hochwertiger Datensatz erstellt.

Die Bewertung von Fehlern bei der Parkscheibe ist ein komplexes Problem. Die Vielfalt der möglichen Abstufungen innerhalb der genannten Kategorien erschwert die quantitative

Bewertung erheblich. Ein Beispiel hierfür ist das Vorhandensein von Schmutz auf einem Teil, welcher jedoch für die Funktion der Parkscheibe irrelevant ist und das Teil somit weiterhin verkaufsfähig macht. Vor der Klassifizierung der Daten ist es daher essenziell, ein klares Schema zu entwickeln, das als Entscheidungshilfe dient. Dieses Schema ermöglicht eine eindeutige Zuordnung der Fehlerkategorien und erleichtert damit die Bewertung der Daten.

Das einheitliche Klassifizierungsschema kann im Anhang unter **[A]** eingesehen werden und wird nun im Detail erläutert.

7.2.1 Parkscheibe mit Schmutz

Im Rahmen der Spezifizierung und Quantifizierung der einzelnen Klassen wird insbesondere der Schmutz näher betrachtet.

Die Definition von Schmutzpartikeln lautet folgendermaßen:

Dunkelgraue bis schwarze, punktuelle oder unförmige Verfärbungen, die sich auf der Oberfläche des Bauteils befinden. Es muss ein deutlicher Kontrastunterschied zwischen dem Bauteil und dem Schmutz erkennbar sein.

Die Größenbestimmung erfolgt anhand der Pixel in der Originalauflösung. Durch diese Vorgehensweise lässt sich die Größe von Schmutzpartikeln schnell und effektiv ermitteln. Hierbei werden sowohl die horizontale als auch die vertikale Ausdehnung des Schmutzes gemessen, wobei der größere der beiden Werte herangezogen wird.

Im Zuge der Klassifizierung werden Schmutzpartikel in eine von vier Kategorien eingestuft:

1. **Winzig:** Schmutzpartikel mit einer maximalen Größe von 12 Pixeln. Diese geringe Größe ist für die Untersuchung irrelevant und wird daher nicht weiter berücksichtigt.
2. **Klein:** Schmutzpartikel mit einer Größe zwischen 13 und 17 Pixeln. In dieser Kategorie sind auf dem Bauteil maximal zwei Schmutzpartikel zulässig.
3. **Mittel:** Schmutzpartikel mit einer Größe zwischen 18 und 25 Pixeln. In dieser Klasse darf auf dem Bauteil nur ein Schmutzpartikel vorkommen.
4. **Groß:** Schmutzpartikel, die eine Größe von über 26 Pixeln aufweisen. Bauteile, die Schmutz in dieser Kategorie aufweisen, werden umgehend aussortiert.

Die detaillierte Klassifizierung nach Größe ermöglicht eine präzise und zuverlässige Bewertung.

7.2.2 Schäden auf der Parkscheibe

In der zweiten Evaluationsphase wird die Beschaffenheit der Parkscheibe anhand von Schäden beurteilt. Diese Schäden werden durch längliche oder rundliche plastische Deformationen auf der Oberfläche oder am Rand des Bauteils definiert, wobei ein heller und dunkler Bereich erkennbar sein muss. Dabei werden zwei grundlegende Schadensarten unterschieden.

Kratzer: Die erste Art ist der „Kratzer“, der durch eine längliche plastische Verformung gekennzeichnet ist und meist eine helle und eine dunkle Seite aufweist. Ein Kratzer ist in der Breite mit 6 bis 12 Pixel definiert. In der Länge darf er höchstens 50 Pixel betragen. Es ist wichtig zu beachten, dass ein Teil aussortiert wird, wenn zwei Kratzer direkt nebeneinander liegen. Des Weiteren ist zu berücksichtigen, dass Kratzer äquivalent zu mittlerem Schmutz sind und daher nur ein Kratzer auf dem Bauteil erlaubt ist.

Kerbe: Die zweite Schadensart ist die „Kerbe“. Wenn ein Schaden die Spezifikationen für einen Kratzer überschreitet, wird er als Kerbe klassifiziert. Bei einer Kerbe liegt eine runde Verformung vor, die ähnlich wie Schmutz erscheint. Aus diesem Grund wird eine Kerbe als äquivalent zu einem Schmutzpartikel betrachtet. Die Evaluationsmethode für Kerben entspricht dabei der für Schmutz.

7.3 Fertiger Datensatz Parkscheibe

Der Datensatz für die Parkscheibe, nach manueller Klassifizierung, beinhaltet insgesamt 2300 Trainingsbilder, 1200 Testbilder und 1050 Validierungsbilder. Während des Trainings erfolgt die Leistungsbewertung anhand der Testbilder. Nach Abschluss des Trainings wird die Performance auf dem Validierungsdatensatz getestet, um die Effizienz des Algorithmus bei der Verarbeitung von unbekannten Daten zu ermitteln.

8. Verbesserung des Loggings

Im Zuge der Untersuchung wird festgestellt, dass das Logging in der Fertigung durch gezielte Verbesserungen optimiert werden kann. Gängige Algorithmen zur Anomalieerkennung setzen meist auf die sogenannte One-Class-Classification (de. Ein-Klassen-Klassifikation) -Methode [23], bei der die Klassen nicht im Verhältnis 1:1 gespeichert werden müssen. Hierbei finden NIO-Bilder ausschließlich Verwendung während des Testens solcher Algorithmen und nicht im eigentlichen Training. Dies führt zu einer erheblichen Beschleunigung des Logging-Prozesses, da keine NIO-Bilder für das Training benötigt werden. Die Herausforderung bei der Akquirierung von Schlechtteilen besteht allgemein darin, dass diese nicht in der gleichen Häufigkeit auftreten wie Gutteile.

Um diese Problematik anzugehen, wird ein neuer Ansatz vorgestellt, der die Datenakquirierung maßgeblich beschleunigt. Zum Training solcher Algorithmen werden grundsätzlich drei Klassen benötigt:

1. Klasse „IO“: Diese Klasse umfasst ausschließlich Gutteile und wird für das Training des One-Class-Classifiers verwendet.
2. Klasse „IO_Test“: Hier werden ebenfalls nur Gutteile abgelegt, jedoch dient diese Klasse ausschließlich zum Testen.
3. Klasse „NIO_Test“: Diese Klasse enthält ausschließlich Schlechtteile und wird ebenfalls ausschließlich zum Testen verwendet.

Das Größenverhältnis zwischen der Klasse „IO_Test“ und der Klasse „NIO_Test“ muss konstant gehalten werden, um ein Ungleichgewicht zwischen den Klassen zu vermeiden. Die anderen Verhältnisse können nach einer Faustformel festgelegt werden:

- IO_Train: 5/7 (72%)
- IO_Test: 1/7 (14%)
- NIO_Test: 1/7 (14%)

Durch diese festgelegten Verhältnisse kann ein Datensatz wesentlich schneller generiert werden, da für jedes Schlechtteil sechs Gutteile gespeichert werden. Diese Methode beschleunigt die Datenakquirierung erheblich und trägt somit zu einer höheren Dynamik und Effizienz in der Fertigung bei.

9. Fehleranfälligkeit des bisherigen Systems

Eine Evaluierung des bestehenden Systems wird durchgeführt, indem alle gespeicherten Bilder von der Parkscheibe und der NanoBlade manuell von Mitarbeitenden klassifiziert werden. Anhand der vorhandenen Information über die Entscheidungen des bisherigen Systems kann die Fehleranfälligkeit leicht ermittelt werden. Die manuelle Klassifikation wird nun mit der Klassifikation des bestehenden Systems verglichen.

Tabelle 1: Genauigkeit bisherige BV

NanoBlade	
Präzision:	99,5%
Trefferquote:	87,9%
F1-Metrik:	93,3%
Parkscheibe	
Präzision:	89,0%
Trefferquote:	60,5%
F1-Metrik:	72,0%

Aus der Untersuchung, siehe. Tabelle 1, geht hervor, dass die Präzision bei der NanoBlade sehr hoch ist. Die F1-Metrik der NanoBlade liegt bei 93,3 %, was auf eine äußerst hohe Genauigkeit schließen lässt. Jedoch schwächelt das System bei der Trefferquote und produziert viel Pseudoausschuss.

Die Implementierung eines neuen Algorithmus muss daher darauf abzielen, den Pseudoausschuss zu reduzieren.

Die Parkscheibe wiederum hat Schwierigkeiten bei Präzision und Trefferquote. Dieses Problem ist bereits aus der Fertigung bekannt. Die bisherige BV kann bei der Parkscheibe nur eine F1-Metrik von 72% erreichen.

Daraus folgt, dass die Klassifizierung der Parkscheibe anspruchsvoller ist als die der NanoBlade. Aus diesem Grund wird in den nächsten Kapiteln die Parkscheibe primär zur Evaluierung von Algorithmen verwendet. So wird sichergestellt, dass nur äußerst leistungsstarke Algorithmen implementiert werden. Die NanoBlade dient hierbei als zweite Evaluierung und zur Sicherstellung der Performance.

In den folgenden Kapiteln werden nun Lösungsansätze diskutiert. Dabei wird stark darauf geachtet, die Komplexität schrittweise zu erhöhen.

Es ist sinnvoll, zunächst einfache Lösungsmethoden zu suchen, bevor komplexe Methoden in Betracht gezogen werden. Durch die Anwendung simpler Methoden kann erheblicher Aufwand eingespart werden, falls diese bereits ausreichende Ergebnisse liefern. Die Evaluation einfacher Ansätze ermöglicht es, die Notwendigkeit und den Nutzen komplexerer Implementierungen besser zu beurteilen. Somit wird eine systematische Vorgehensweise gewährleistet, die eine effiziente und effektive Lösungsfindung begünstigt.

10. Logik CNN Autoencoder

Nun wird damit begonnen, das bisherige System mit Deep Learning Methoden zu verbessern. Ein vielversprechender erster Lösungsansatz besteht in der Verwendung eines Logik CNN Autoencoders, der durch die Kombination von CNN-Schichten wichtige Merkmale extrahiert und durch die Autoencoder-Architektur ausschließlich anhand der normalen Testdaten lernt. Das Kürzel „Logik“ bezieht sich hier darauf, dass der Autoencoder den gesamten Bildbereich betrachtet und nicht nur einen kleinen Ausschnitt, auch als Patches (de. Ausschnitte) bezeichnet, wie es in einigen anderen Architekturen der Fall ist [16].

Die Architektur des Modells wird aus Quelle [16] entnommen und basiert auf der Komprimierung und Dekomprimierung der Merkmale ausschließlich über CNN-Schichten.

Die Implementierung des Modells erfolgt unter Verwendung von TensorFlow und Keras, da diese Bibliotheken eine effiziente Handhabung und Entwicklung solcher Modelle ermöglichen. Zudem werden alle Bilder auf eine Größe von 256x256 skaliert, um eine angemessene Netzwerkgröße zu gewährleisten.

Diese Bildgröße bringt eine gute Balance zwischen Speicherplatz und Detailreichtum. Aus Komplexitätsgründen wird das Netzwerk nicht über Transfer Lernen [27] vortrainiert.

Die präsentierte Architektur stellt einen vielversprechenden Ansatz dar, um die Anomalieerkennung in Bildern zu verbessern und das Problem des Pseudoausschlusses bei der Parkscheibe zu adressieren. Durch die Betrachtung des gesamten Bildes und die Verwendung von CNN-Schichten zur Merkmalsextraktion ist das Modell in der Lage, wichtige Informationen zu erfassen. In den folgenden Experimenten wird die Leistung des Logik CNN Autoencoders ausgiebig getestet und mit anderen Ansätzen verglichen, um seine Wirksamkeit und Einsatzmöglichkeiten zu evaluieren.

Alle folgenden Code Skripte werden in ein GitHub Repo eingebunden, welches sich im Anhang unter [O] finden lässt.

10.1 Training

Das Training und Validieren wird für alle folgenden Netzwerke auf einer GTX 1080 Grafikkarte mit 8GB Speicher durchgeführt. In Abb. 17 ist der Trainingsverlauf des Netzwerks zu erkennen. Während des Trainings konnte der Fehler zwischen Eingabebild und Rekonstruktion kontinuierlich verringern werden. Dabei ist wichtig, dass auch der Testfehler sinkt, da sonst Overfitting (de. Überanpassung) [28] vorliegen könnte.

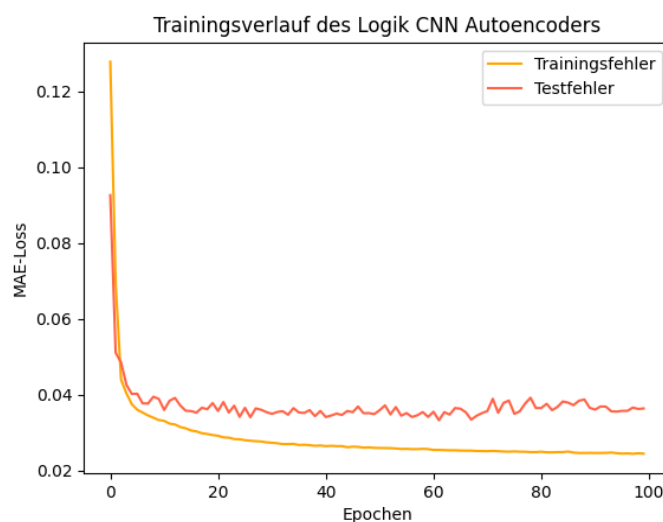


Abbildung 17: Trainingsverlauf Logik CNN Autoencoder

10.2 Ausgabebilder

Die Ausgabebilder des Logik CNN Autoencoders sind in Abb. 18 zu erkennen. Die Anomalie-Karte zeigt die Differenz zwischen der Eingabe und der Ausgabe. Der Autoencoder kann gut die Eingabedaten nachbilden und sollte somit in der Lage sein, Anomalien effektiv zu erkennen.



Abbildung 18: Ausgabe des Logik CNN Autoencoder

10.3 Ergebnisse

Tabelle 2: Genauigkeit Logik CNN Autoencoder Parkscheibe

Logik CNN Autoencoder	
Präzision:	70,9%
Trefferquote:	50,7%
F1-Metrik:	59,1%

Für eine genaue Evaluierung werden die statistischen Werte herangezogen, welche sich auch in Tabelle 2 genauer betrachten lassen.

Es wird deutlich, dass bei Anwendung dieser Methode der Pseudoausschuss in beträchtlichem Maße ansteigt, da die Trefferquote lediglich 50,7% erreicht. Zusätzlich zeigt sich, dass im Vergleich zur bisherigen Methode keine Verbesserung der Präzision erzielt werden kann. Dementsprechend ist es dem Modell auch unter Berücksichtigung der F1-Metrik nicht möglich, die erforderliche Genauigkeit zu erreichen.

Angesichts dieser Ergebnisse wird nun die Komplexität erhöht und ein weiterentwickelter Ansatz in Betracht gezogen, um die Anomalieerkennung zu verbessern.

Die Entscheidung für eine anspruchsvollere Methode basiert auf der Erkenntnis, dass der bisherige Ansatz nicht den gewünschten Erfolg erzielt hat. Durch die Implementierung eines fortschrittlicheren Modells wird angestrebt, die Leistungsfähigkeit zu steigern und das Problem des Pseudoausschusses bei der Parkscheibe zu bewältigen.

11. Patch CNN Autoencoder

Ein weiterentwickelter Ansatz zur Anomalieerkennung ist der Patch CNN Autoencoder. Im Gegensatz zum vorherigen Logik CNN Autoencoder betrachtet diese Variante nur kleine Ausschnitte des Bildes, sogenannte Patches. Patches sind in der Regel kleine Bildausschnitte, die eine reduzierte Größe von 32x32 Pixeln aufweisen. Das Gesamtbild, welches ursprünglich 256x256 Pixel umfasst, wird somit in 64 solcher Patches unterteilt. Die Grundannahme dieser Methode ist, dass durch den Einsatz kleinerer Bildausschnitte das Netzwerk Anomalien leichter erkennen kann. Durch die Dimensionsreduktion werden potenziell unnötige Daten verworfen, wodurch die Erkennung von Anomalien erleichtert wird. In Abb. 19 ist diese Funktionsweise ersichtlich. Feine Konturen lassen sich durch diesen Ansatz leichter erkennen.

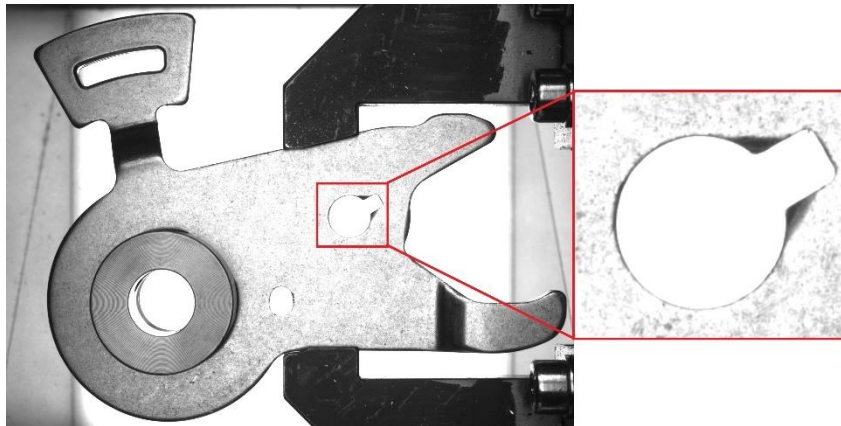


Abbildung 19: Funktionsweise von Patches

Die Idee, kleinere Patches in der BV zu verwenden, basiert auf wissenschaftlichen Erkenntnissen, die zeigen, dass Patch-basierte Ansätze die Fähigkeit des Netzwerks zur Merkmalsextraktion und damit zur Anomalieerkennung verbessern können [5, 16].

Untersuchungen haben gezeigt, dass Patch-Netzwerke effektiv sind, um lokale Strukturen und Muster in Bildern zu erkennen, was zu einer verbesserten Leistung bei der Erkennung von Anomalien führt [16].

Ein weiterer Vorteil dieser Methode ist, dass die Modelle durch die Verwendung von kleineren Eingabensensoren deutlich weniger Speicherplatz benötigen. Dies ist von besonderer Bedeutung in der Praxis, da der Speicherbedarf bei großen Bildern und komplexen Modellen schnell ansteigen kann.

Die Architektur des Patch CNN Autoencoders wird, ähnlich wie bei der vorherigen Variante, aus wissenschaftlich fundierten Quellen entnommen [16]. Durch die Verwendung von CNN-Schichten kann eine effiziente Merkmalsextraktion erreicht werden, und der Aufbau des Patch CNN Autoencoders unterscheidet sich nur geringfügig von dem des Logik CNN Autoencoders, obwohl letzterer eine größere Größe aufweist.

11.1 Training

In den folgenden Experimenten wird die Leistung des Patch CNN Autoencoders anhand der Parkscheibe getestet und mit dem vorherigen Ansatz verglichen, um seine Wirksamkeit zu bewerten.

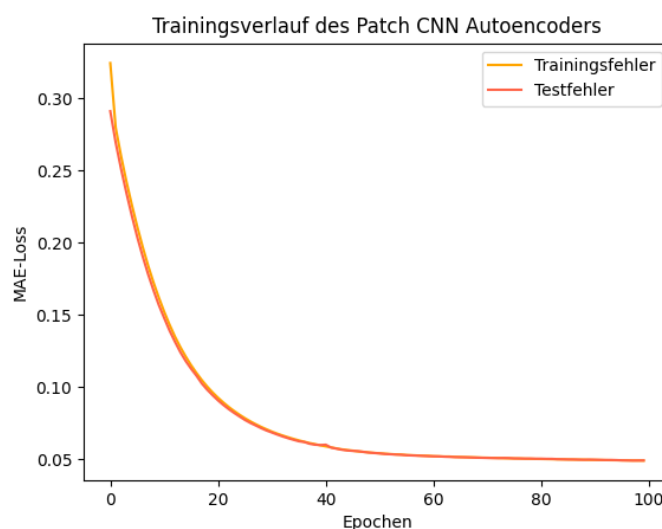


Abbildung 20: Trainingsverlauf Patch CNN Autoencoder

Analog zum Logik CNN Autoencoder wird der Trainingsverlauf (Abb. 20) begutachtet. Hier zeigt sich wieder ein sehr ausgeglichenes Bild. Das Netzwerk kann hier einen sehr konstanten und gleichmäßigen Verlauf vorzeigen. Der Testfehler ist im Verhältnis zum Trainingsfehler geringer als beim Logik CNN Autoencoder. Auch hier gibt es keine Anzeichen auf Overfitting, das Netzwerk konnte das Training somit erfolgreich abschließen.

11.2 Ausgabebilder

Die Ausgabebilder des Patch CNN Autoencoders sind in Abb. 21 zu erkennen. Hier ist die Anordnung analog zum Logik CNN Autoencoder. Es lässt sich gut erkennen, wie das Netzwerk über die einzelnen Patches versucht, eine umfängliche Rekonstruktion zu generieren. Daher kann hier ebenfalls von einem erfolgreichen Training ausgegangen werden.

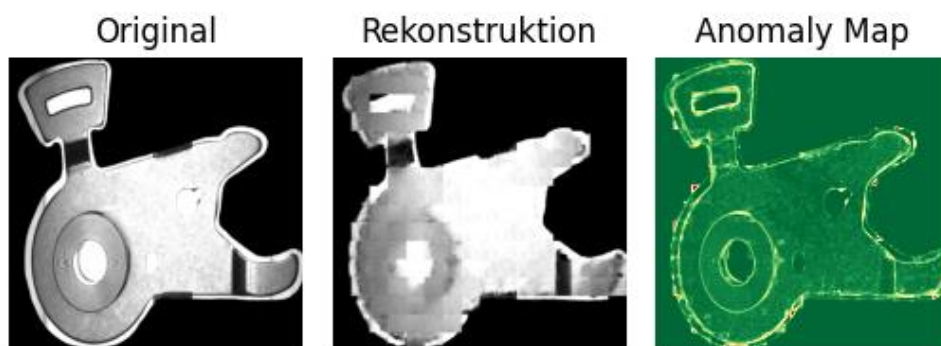


Abbildung 21: Ausgabe des Patch CNN Autoencoder

11.3 Ergebnisse

Analog zum Logik CNN Autoencoder, wird hier ebenfalls wieder die Genauigkeit anhand der statistischen Werte überprüft.

Tabelle 3: Genauigkeit Patch CNN Autoencoder Parkscheibe

Patch CNN Autoencoder	
Präzision:	68,0%
Trefferquote:	56,0%
F1-Metrik:	61,4%

Die Ergebnisse, siehe Tabelle 3, zeigen, dass der Patch CNN Autoencoder einen geringen Leistungsgewinn im Vergleich zum Logik CNN Autoencoder erzielt. Der Patch CNN Autoencoder kann bei der Trefferquote den Logik CNN Autoencoder um 5,3% übertreffen. Jedoch muss sich der Patch CNN Autoencoder bei der Trefferquote dem Logik CNN Autoencoder geschlagen geben, der ein 2,9% besseres Ergebnis erzielen konnte. Insgesamt ist die F1-Metrik mit 2,3% geringfügig höher. Dennoch ist die Genauigkeit des Patch CNN Autoencoders nicht besser als die bisherige Methode, weshalb ein neuer Ansatz nötig ist.

12. Mögliche Ursache für unzureichende Performance der Autoencoder

Eine mögliche Erklärung für die begrenzte Leistungsfähigkeit beider Methoden könnte in ihrer mangelnden Anpassbarkeit liegen. Beim Patch CNN Autoencoder wird das Bild in kleine Patches unterteilt, was zu einer lokalen Betrachtung führt. Ähnlich verhält es sich mit dem Logik CNN Autoencoder, der das gesamte Bild betrachtet. In beiden Fällen könnte die begrenzte Fähigkeit zur Merkmalsextraktion und Rekonstruktion dazu führen, dass wichtige Informationen verloren gehen oder Verzerrungen in der Rekonstruktion auftreten.

Bei genauer Betrachtung der Beispielbilder wird deutlich, dass die Rekonstruktion bei den IO-Teilen teilweise von den Originalen abweicht. Dies könnte auf Schwierigkeiten bei der Erfassung und Wiedergabe der komplexen Merkmale in den Daten hinweisen. Es scheint, dass die Autoencoder-Modelle nicht in der Lage sind, die Feinheiten und spezifischen Charakteristiken der IO-Teile mit hoher Genauigkeit zu extrahieren und korrekt zu rekonstruieren.

Ein weiterer möglicher Grund für die begrenzte Leistung könnte in der begrenzten Kapazität der gewählten Architekturen liegen. Sowohl der Patch CNN Autoencoder als auch der Logik CNN Autoencoder basieren auf bestimmten vorgegebenen Strukturen und Schichten, die möglicherweise nicht ausreichend flexibel sind, um die Komplexität der vorliegenden Daten angemessen zu erfassen.

Um die Leistung der Autoencoder-Modelle zu verbessern, könnten alternative Architekturen, wie zum Beispiel tiefere und breitere Netzwerke oder andere Formen von Autoencodern, in Erwägung gezogen werden. Zudem könnten regelmäßige Anpassungen der Hyperparameter und eine sorgfältige Auswahl der Verlustfunktion dazu beitragen, die Rekonstruktionsgenauigkeit und damit die Gesamtleistung der Modelle zu verbessern.

Insgesamt verdeutlichen die Ergebnisse, dass die Anwendung von Patch- und Logik CNN Autoencodern zur Anomalieerkennung in diesem spezifischen Kontext mit Herausforderungen verbunden ist. Weitere Untersuchungen und möglicherweise alternative Ansätze sind daher erforderlich.

13. Klassifizierung durch Supervised Learning

Nun wird EfficientNetV2-S aus Keras CV für eine Klassifikation der Parkscheibe verwendet. Die hohe Leistungsfähigkeit des vortrainierten Netzwerks erzeugt einen vielversprechenden Ansatz, die Parkscheibe effektiv zu klassifizieren.

Die Keras CV Bibliothek hat ihren Schwerpunkt vor allem auf die Verwendung vortrainierter Backbones gelegt, die hauptsächlich für die Klassifikationsaufgabe trainiert wurden und daher in diesem Bereich eine bessere Leistung erzielen [14, 15]. Aus diesem Grund wird hier bewusst eine Entscheidung für ein Klassifikationsszenario getroffen. Es ist bekannt, dass diese Entscheidung möglicherweise später zu einem Problem führen kann, da unbekannte Daten das Netzwerk verwirren könnten. Jedoch liegt der Fokus hier eher darauf herauszufinden, ob eine unzureichende Merkmalsextraktion das Problem im Autoencoder-Ansatz war.

EfficientNetV2-S wird daher in einem Rahmen einer Klassifikation mit zwei Klassen eingebunden.

13.1 Training

Für eine Klassifikationsaufgabe wird ein neuer Datensatz verwendet, der eine andere Klassenkonstellation erfordert. In dieser Konstellation müssen die Klassen im gleichen Verhältnis zueinander stehen. Im Gegensatz zu früheren Datensätzen, die tendenziell weniger NIO- im Vergleich zu IO-Teilen aufweisen, muss hier die Gesamtanzahl an Bildern verringert werden.

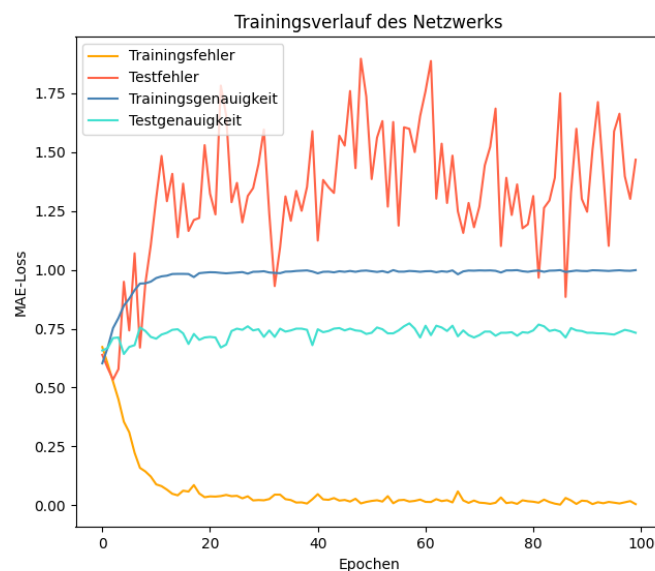


Abbildung 22: Trainingsverlauf EfficientNetV2-S

EfficientNetV2-S wird mit 1700 Trainingsbildern trainiert und während des Trainings auf 400 Bildern getestet. Später findet die Evaluierung auf 680 Validierungsbildern statt. Der Trainingsverlauf, abgebildet in Abb. 22, illustriert den allmählichen Rückgang des Trainingsfehlers über die Epochen hinweg und den entsprechenden Anstieg der Genauigkeit.

Dabei ist besonders wichtig, dass die Testgenauigkeit nicht signifikant abfällt, was auf ein mögliches Overfitting hindeuten würde. Diese Problematik ist hier nicht vorhanden.

Zusammenfassend lässt sich aus dem Verlauf des Trainings ableiten, dass das Modell erfolgreich trainiert wurde.

13.2 Ergebnisse

EfficientNetV2-S wird, analog zu den beiden vorherigen Ansätzen, getestet.

Die statistischen Werte geben wieder die nötige Auskunft über die Genauigkeit der Methode.

Tabelle 4: Genauigkeit EfficientNetV2-S Parkscheibe

EfficientNetV2-S	
Präzision:	85,7%
Trefferquote:	72,3%
F1-Metrik:	78,5%

Die Auswertung der Tabelle 4 zeigt, dass EfficientNetV2-S den Patch CNN Autoencoder in der F1-Metrik um 17,1% und die bisherige BV um 6,5% überbieten kann. Den Pseudoausschuss kann EfficientNetV2-S um 11,8% reduzieren. Jedoch konnte die Präzision nicht erhöht werden. Hier liegt die bisherige BV 3,3% vor EfficientNetV2-S.

Diese Ergebnisse unterstreichen eindeutig die Eignung dieses vortrainierten Netzwerks für die Erkennung von Parkscheiben. Interessanterweise liegt die Genauigkeit nicht deutlich über der bisherigen Methode. Dies bestätigt die Hypothese, dass die Extraktion von relevanten Merkmalen bei Parkscheiben äußerst anspruchsvoll ist und daher auch die mangelnde Genauigkeit der Autoencoder basierten Methode erklärt.

Es ist jedoch wichtig anzumerken, dass Klassifikationsmodelle aus bereits erläuterten Gründen nur bedingt für den Einsatz im Sondermaschinenbau geeignet sind. Die Präzision ist unter anderem geringer als bei der bisherigen BV, was zu Kundenreklamationen führen könnte. Aus diesem Grund werden nun Unsupervised Learning Modelle aus Anomalib untersucht.

14.3 Evaluieren der Anomalib Modelle

14.3.1 Testmethodik

Um einen umfassenden Vergleich zwischen den Anomalib-Modellen sowie später auch kommerziellen Varianten zu ermöglichen, bedarf es einer präzisen Testmethodik. Diese Methodik zielt darauf ab, verschiedene Kennzahlen zu ermitteln, die als Grundlage für eine gründliche Bewertung dienen sollen.

Zusätzlich zu den bereits bekannten Kennzahlen (Präzision, Trefferquote, F1-Metrik und AUROC) werden nun weitere Kennzahlen hinzugefügt, die besonders für eine spätere Implementierung im industriellen Sondermaschinenbau wichtig sind.

- **Trainingsdauer:** Die Zeit, die benötigt wird, um das Modell während des Trainings auf einem gegebenen Datensatz zu optimieren. Die Trainingsdauer wird direkt von Anomalib angegeben.
- **Inferenzzeit in Millisekunden:** Die Zeit, die das Modell benötigt, um eine einzelne Inferenz auf einem Testbild durchzuführen, gemessen in Millisekunden.
- **Frames pro Sekunde (FPS):** Beschreibt die Anzahl der Bilder, die in einer Sekunde verarbeitet werden können. Ergibt sich aus dem Kehrwert der Inferencezeit.

Besonders im Kontext des industriellen Sondermaschinenbaus gewinnt eine kurze Trainingsdauer sowie eine geringe Inferenzzeit an Bedeutung. Die Trainingsdauer wird direkt für den NanoBlade-Datensatz ermittelt, der 1000 Trainingsbilder umfasst. Diese Dauer kann zur Extrapolation der Trainingszeiten für andere Datensätze genutzt werden.

Um die Inferenzzeit der trainierten Modelle zu quantifizieren, werden die Modelle nach dem Trainingsvorgang in den Evaluierungsmodus versetzt. Dieser Schritt wird durch die Integration der Modelle in ein dediziertes Skript realisiert, das die Inferenzzeit misst [35].

An dieser Stelle wird bewusst beschlossen, die Ergebnisse des EfficientNetV2-S-Modells in die anschließende Auswertung einzubeziehen. Diese Entscheidung ist wichtig, um einen präzisen quantitativen Vergleich zwischen der Klassifizierungsmethode von KerasCV und der Anomalieerkennung von Anomalib zu ermöglichen.

Die ermittelten Kennzahlen ermöglichen einen objektiven quantitativen Vergleich der verschiedenen Modelle. Dieser Vergleich bildet eine solide Grundlage für die fundierte Auswahl und Implementierung der optimalen Lösung, die den Anforderungen am besten gerecht wird.

14.3.2 Interpretation der Ergebnisse

Aufgrund der umfangreichen Untersuchung befinden sich alle Ergebnisse im Anhang unter [C, D, E].

Alle Modelle in den Tabellen spiegeln die beste Konfiguration wieder, welche in der Bearbeitungszeit gefunden werden konnte. Dafür wurde Bildgröße, Epochen, Lernraten und Feature Extractor angepasst und evaluiert. Alle Modelle wurden mit einer Auflösung von 256x256 Pixeln trainiert. Diese Größe hat sich aufgrund vorheriger Tests bewährt und wird ebenfalls von den meisten Modellen standardmäßig verwendet. Die Batch Size wird, wenn nötig, entsprechend dem Speicherbedarf angepasst oder aus den Standardeinstellungen übernommen.

Zudem ist es wichtig, anzumerken, dass die Integration von PatchCore in die bestehende Systemarchitektur leider nicht möglich war. Die genaue Ursache für dieses Problem konnte bislang nicht eindeutig ermittelt werden. Trotz einer kontinuierlichen Trainingsdauer von mehr als zwölf Stunden musste das Training vorzeitig abgebrochen werden, da keinerlei signifikante Fortschritte erzielt werden konnten.

Um dieses Hindernis zu überwinden, könnte eine vielversprechende Option darin bestehen, auf ein System umzusteigen, das über einen Grafikspeicher von mindestens 16 Gigabyte verfügt. Dies könnte die erforderliche Ressourcenzuweisung ermöglichen, um den Trainingsprozess von PatchCore erfolgreich durchzuführen.

14.3.3.1 NanoBlade

Durchweg erzielen die Modelle bei der NanoBlade die höchste Genauigkeit. Diese überlegene Leistung ist höchstwahrscheinlich auf die spezifische Struktur der Bilder zurückzuführen. Dank der Anwendung einer diffusen Durchlichtbeleuchtung weisen die Bilder einen hohen Kontrast auf und zeigen nur begrenzte Grauwertunterschiede. Dieses klare Bild ermöglicht es den Algorithmen, Fehler präzise zu identifizieren, was sich wiederum in den herausragenden Ergebnissen widerspiegelt.

Das CFA-Modell [21] zeichnet sich als Spitzenreiter aus. Es erzielt in sämtlichen Messwerten die beste Leistung und stellt somit die optimale Wahl für diesen Anwendungsbereich dar. Im direkten Vergleich zur vorherigen BV weist das CFA-Modell eine beachtliche Steigerung von 3% in der F1-Metrik auf. Die Präzision erhöhte sich moderat um 0,5%, was insbesondere der bereits hohen Präzision der vorherigen BV geschuldet ist. Besonders bemerkenswert ist die Erhöhung der Trefferquote um 5%, was den bedeutendsten Teil der Testergebnisse ausmacht. Diese gesteigerte Trefferquote führt dazu, dass weniger Bauteile fälschlicherweise aussortiert werden, die tatsächlich noch den Anforderungen für den Verkauf entsprechen würden.

Zusammenfassend lässt sich feststellen, dass die Algorithmen von Anomalib sich außerordentlich gut für die Federprüfung in der NanoBlade eignen.

14.3.3.2 Parkscheibe

Ein differenziertes Bild zeichnet sich hinsichtlich der Parkscheibenprüfung ab, bei der die Algorithmen auf erheblich größere Herausforderungen stießen. Im Kontrast zur einheitlichen Struktur der NanoBlades präsentieren sich bei der Parkscheibe eine Vielzahl unterschiedlicher Grauwerte sowie subtile Details. In diesem Szenario konnte das PaDim-Modell als das Leistungsstärkste hervortreten. Durch eine geringfügige Verbesserung der F1-Metrik um 1,9% gegenüber der vorherigen Methode setzte sich PaDim als beste Wahl durch.

Des Weiteren verzeichnete das PaDim-Modell eine minimale Steigerung der Trefferquote um 0,4%. Hinsichtlich der Präzision erzielte das CFlow-Modell mit einem Wert von 97,7% die höchste Wertung.

Trotz dieser Bemühungen ist anzumerken, dass Anomalib bei der Parkscheibenprüfung keine wesentliche Steigerung der Genauigkeit realisieren konnte. Das Ziel, den Pseudoausschuss maßgeblich zu minimieren, wird nur bedingt erreicht. Es deutet sich an, dass die Fähigkeit der Algorithmen zur effektiven Umsetzung der zuvor definierten Entscheidungsrichtlinie begrenzt ist. Diese Limitierung könnte auf die Funktionsweise der verwendeten Algorithmen zurückzuführen sein. Eine Anomalieerkennung, basierend auf Unsupervised Learning, scheint möglicherweise nicht in der Lage zu sein, Anomalien zu kategorisieren und angemessen zu bewerten. Dennoch konnten die Algorithmen eine beachtliche Leistungsfähigkeit bei der Identifikation von winzigen Schmutzpartikeln aufweisen, welche sich von der Umgebung abheben.

Zusammenfassend lässt sich sagen, dass sich die Algorithmen von Anomalib gut für die Parkscheibe eignen würden, da sie eine höhere Genauigkeit gegenüber der bisherigen BV aufweisen.

14.3.3.3 Vergleich zu EfficientNetV2-S

Im Kontrast dazu steht der Klassifikationsansatz mit EfficientNetV2-S, bei dem bessere Ergebnisse erzielt wurden, als im Vergleich zu den Anomalib-Modellen. Diese Erkenntnis steht im Widerspruch zur bisherigen Annahme, dass im Bereich der Anomalieerkennung die Klassifikation schlechtere Leistungen erzielt als das Unsupervised Learning [5, 12].

Daraus lässt sich Schlussfolgern, dass die Aufgabenstellung der Parkscheibenprüfung, basierend auf den gegebenen Entscheidungsparametern, nicht länger im Kontext der Anomalieerkennung betrachtet werden sollte, sondern vielmehr dem Bereich der Klassifikation zugeordnet werden muss.

Aus diesem Grund ist es notwendig, für jede Problemstellung im Bereich des Sondermaschinenbaus, die mithilfe von künstlicher Intelligenz gelöst werden soll, sowohl die Klassifikation mittels KerasCV EfficientNetV2-S als auch die Anomalib-Modelle zu testen. Nur durch diesen Ansatz kann die optimale Leistung erzielt werden.

Dennoch bleibt ein unbestreitbarer Vorteil der Anomalib-Modelle ihre herausragende Präzision. Modelle aus dem Bereich des Unsupervised Learnings sind oft deutlich strenger, als es bei einer Klassifikation der Fall wäre. Daher ist es von essenzieller Bedeutung, im Vorfeld sorgfältig zu überlegen, welcher Aspekt priorisiert werden soll. Falls höchste Präzision im Vordergrund steht, wie es beispielsweise im Automotive-Bereich häufig der Fall ist, erweisen sich die Anomalib-Unsupervised-Modelle als die geeignetere Wahl. Andererseits, wenn der Kunde zufrieden ist mit Bauteilen, die nicht zwingend höchsten Qualitätsstandards genügen, bieten sich Klassifikationsalgorithmen wie das EfficientNetV2-Modell an.

14.3.3.4 Zeiten

Besonders aufschlussreich sind die Trainings- und Inferencezeiten im Kontext der Anomalib-Modelle. Hinsichtlich der Trainingsdauer sticht hervor, dass EfficientAD signifikant mehr Zeit in Anspruch nimmt im Vergleich zu anderen Netzwerken. Dies könnte vor allem auf die niedrige Batch Size und den komplexen Trainingsprozess zurückzuführen sein, der die Anpassung mehrerer Netzwerke beinhaltet. Bemerkenswert ist jedoch, dass PaDim eine äußerst bemerkenswerte Trainingszeit von lediglich 4 Minuten aufweist. Diese Leistung wurde in mehreren Überprüfungen bestätigt.

Die herausragende Effizienz von PaDim ist auf seine Architektur zurückzuführen, in der kein herkömmliches neuronales Netzwerk trainiert wird, sondern vielmehr eine Gaußsche Verteilung der Daten generiert wird [19]. Diese Besonderheit spiegelt sich ebenfalls in den Inferencezeiten wieder, wobei PaDim mit lediglich 3,4 Millisekunden deutlich die kürzeste Zeit aufweist. Diese Geschwindigkeit ermöglicht 294 Frames pro Sekunde, was selbst den anspruchsvollsten Anwendungen im industriellen Sondermaschinenbau gerecht wird.

Es sei jedoch angemerkt, dass die Inferencezeiten von EfficientAD und CFA ausreichend sind. In den meisten Szenarien ist eine solch hohe Geschwindigkeit kaum erforderlich. Die Inference bei CFlow kann unter Umständen zu langsam sein, was dazu führt, dass von der Verwendung dieser Architektur abgeraten wird.

14.4 Zusammenfassung Anomalib

Basierend auf den vorliegenden Ergebnissen kann zweifelslos festgestellt werden, dass Anomalib auch im Kontext des industriellen Sondermaschinenbaus gut geeignet ist. Die Erkenntnisse verdeutlichen eindrucksvoll, dass gegenwärtige Methoden der BV nicht die Genauigkeit der Anomalib-Modelle erreichen können. Zudem wurde das Bedenken entkräftet, dass der Einsatz von Unsupervised Learning aufgrund unvorhersehbarer Anomalien dazu führen könnte, dass die Algorithmen sämtliche Bauteile aussortieren, sofern Bauteile im Zweifel maskiert werden.

Dennoch bleibt die Notwendigkeit bestehen, sowohl Anomalib als auch KerasCV in Kombination zu nutzen, um sicherzustellen, dass die vorliegende Herausforderung nicht möglicherweise durch eine Klassifikation effizienter gelöst werden kann. Darüber hinaus ist es relevant anzumerken, dass der Einsatz von Open-Source-Lösungen im industriellen Sondermaschinenbau oftmals problematisch ist, da häufig hochqualifiziertes und spezialisiertes Fachpersonal erforderlich ist.

Infolgedessen steht nun die Evaluierung von Anomalib und KerasCV im direkten Vergleich zu kommerziell erhältlichen Produkten an, darunter Merlic von MVTec und Astrocyte von Teledyne Dalsa. Dieser Vergleich wird im nächsten Schritt durchgeführt.

15. DLT von MVTec

Nun wird damit begonnen das DLT von MVTec anhand der Datensätze zu testen.

Die Testmethodik des DLTs entspricht derjenigen der Anomalib-Modelle, um einen aussagekräftigen Vergleich zu ermöglichen. Hierbei wurden gleichermaßen die identischen Datensätze und Validierungssätze verwendet, wobei auch die Auflösung beibehalten wurde. Das DLT bietet sowohl Klassifikation als auch Anomalieerkennung, weshalb beide Varianten in dieser Untersuchung berücksichtigt werden. Die Trainingszeiten sowie die Inferencezeiten werden direkt während des Trainings- bzw. der Inference-Phase vom DLT ausgegeben. Für diese Auswertung wurden die Standardkonfigurationen des DLT verwendet.

15.1 Interpretation der Ergebnisse

Ebenfalls, analog zu Anomalib, finden sich die Ergebnisse im Anhang unter **[F, G, H]**.

Die Ergebnisse des DLTs unterscheiden sich teilweise von den Ergebnissen von EfficientNetV2-S und Anomalib.

Bei der Klassifikation der NanoBlade liegt das DLT in der F1-Metrik 0,8% unter EfficientNetV2-S, wobei auch die Präzision und die Trefferquote geringer sind. Grund hierfür ist der nicht so leistungsstarke Feature Extractor des DLTs. Wie aus den Einstellungen ersichtlich ist, wird auf ResNet50 gesetzt, der nicht eine so hohe Leistungsfähigkeit besitzt wie EfficientNetV2-S [15].

Bei der Anomalieerkennung der NanoBlade liegt das DLT fast gleichauf mit dem CFA-Modell aus Anomalib. Hier unterscheiden sich die beiden Modelle nur minimal im Bereich unter 0,2%.

Bei der Parkscheibe ist ein ähnliches Bild zu erkennen. Bei der Klassifikation konnte EfficientNetV2-S das DLT um 2,5% schlagen. Bei der Anomalieerkennung muss sich das DLT PaDim geschlagen geben, welches eine 5,2% höhere F1-Metrik erzielen konnte.

Bezüglich der Trainings- und Inference Zeiten kann sich das DLT im Bereich der Klassifikation 13 Minuten vor EfficientNetV2-S setzen. Bei der Anomalieerkennung muss sich das DLT PaDim geschlagen geben, welches einen sehr großen Vorsprung von 1h 28 min aufweisen kann.

Die Inference Zeit bei der Klassifikation ist vom DLT 17ms schneller als EfficientNetV2-S und bei der Anomalieerkennung 14,5ms langsamer als Anomalib.

Zusammenfassend ergibt sich, dass das DLT in Bezug auf seine Präzision sowohl bei der Klassifikation als auch bei der Anomalieerkennung hinter den Ergebnissen von EfficientNetV2-S und Anomalib zurückbleibt. Im Rahmen der Untersuchung der F1-Metrik zeigt sich, dass das DLT in der Klassifikation um 0,8% bis 2,5% unter den Ergebnissen von EfficientNetV2-S liegt, während es in der Anomalieerkennung um 0,1% bis 5,2% hinter den Ergebnissen von Anomalib zurückfällt.

Somit lässt sich auch festhalten, dass Merlic von MVTec nicht die Genauigkeit erreichen wird, wie die Ansätze mit EfficientNetV2-S und Anomalib. Die Zeiten vom DLT sind auf einem akzeptablen Niveau und sollten für den Einsatz im industriellen Sondermaschinenbau geeignet sein.

16. Astrocyte von Teledyne Dalsa

Für die Vergleichbarkeit werden alle Tests analog zum DLT von MVTec durchgeführt. Dazu bleiben die Datensätze sowie die Auflösung mit 256x256 unverändert. Es werden ebenfalls die Standardeinstellungen von Astrocyte verwendet. Leider verwendet Astrocyte für die Bewertung ihrer Modelle nur die AUROC Metrik. Da das DLT und EfficientNetV2-S diese Metrik jedoch nicht ausgeben, kann hier leider nur bedingt ein Vergleich geschaffen werden. Dennoch bietet dieser Test interessante Resultate, da Anomalib die AUROC-Metrik ausgeben kann.

16.1 Interpretation der Ergebnisse

Ebenfalls, analog zu den beiden Methoden zuvor, befinden sich die Ergebnisse im Anhang unter [I, J, K].

Analog zum MVTec DLT konnte auch Astrocyte von Teledyne Dalsa die Genauigkeit der beiden vorherigen Modelle nicht übertreffen. Da Astrocyte ausschließlich AUROC-Werte ausgibt, ist das Vergleichspotenzial begrenzt. Es lassen sich dennoch qualitative Vergleiche aufstellen, wenn die Methoden von Astrocyte genauer untersucht werden. In diesem Kontext nutzt Astrocyte die ResNet50-Architektur, die mit dem MVTec DLT vergleichbar ist. Daher lässt sich festhalten, dass die Leistung im Bereich der Klassifikation der des DLT nahekommt. Aus diesem Grund wird Astrocyte nicht das gleiche Maß an Genauigkeit erreichen wie EfficientNetV2-S.

Im Bereich der Anomalieerkennung zeigt sich ein ähnliches Bild.

Astrocyte muss sich bei den NanoBlades dem CFA-Modell aus Anomalib geschlagen geben. CFA konnte hier eine 1,4% höhere Genauigkeit erreichen als Astrocyte.

Bei der Parkscheibe muss Astrocyte sich PaDim aus Anomalib geschlagen geben. PaDim ist in der Lage, Astrocyte um bemerkenswerte 10,1% zu übertreffen, was einen signifikanten Vorsprung darstellt.

Bezüglich der Trainingszeit ist Astrocyte um 11 Minuten schneller als EfficientNetV2-S. Wobei sich Astrocyte bei der Anomalieerkennung PaDim unterordnen muss, welches 5 Minuten schneller war, was für Astrocyte dennoch ein guter Wert ist.

Die Inference von Astrocyte ist bei der Klassifizierung im 13ms schneller und bei der Anomalieerkennung um 4,5ms langsamer.

Zusammenfassend lässt sich festhalten, dass Astrocyte ebenso wie das MVTec DLT nicht die Genauigkeit der Open-Source-Lösungen EfficientNetV2-S und Anomalib erreichen kann. Die Trainings- und Inference Zeiten sind auf einem guten Niveau.

17. Nutzwertanalyse der unterschiedlichen Lösungen

Die Ergebnisse aus Kapitel 15 und 16 verdeutlichen einen spürbaren Vorteil der Open-Source-Lösungen gegenüber den kommerziellen Varianten. Angesichts dieser Erkenntnisse widmet sich das aktuelle Kapitel einem quantitativen Vergleich der Implementierungsfreundlichkeit von Open-Source- und kommerziellen Lösungen in die bestehende Systemlandschaft des Sondermaschinenbaus. Hierbei wird auf eine Nutzwertanalyse nach Zangenmeister [41] als Grundlage zurückgegriffen.

Bei der Ausführung einer solchen Nutzwertanalyse zur Bewertung von Bildverarbeitungsprogrammen, speziell in Bezug auf den Vergleich zwischen Open-Source-Lösungen und kommerziellen Produkten, kommen diverse entscheidende Kriterien zur Geltung. Im Folgenden befinden sich die Kriterien, anhand deren die Methoden abgewogen werden:

1. **Niedrige Kosten:** Die Kosten für die Anschaffung der Software.
2. **Genauigkeit:** Beschreibt die Fähigkeit des Systems, Fehler zuverlässig zu erkennen.
3. **Leistungsfähigkeit:** Verarbeitungsgeschwindigkeit des Systems.
4. **Anpassungsmöglichkeiten:** Die Flexibilität und Anpassbarkeit der Software an spezifische Anforderungen und Szenarien.
5. **Support:** Die Größe und Aktivität der Nutzergemeinschaft sowie der verfügbare Kundensupport und technische Hilfe.
6. **Dokumentation:** Die Qualität und Vollständigkeit der bereitgestellten Dokumentation, Anleitungen und Tutorials.
7. **Integration:** Die Fähigkeit der Software, in bestehende Systeme und Arbeitsabläufe integriert zu werden.
8. **Skalierbarkeit:** Die Möglichkeit, die Software je nach Bedarf auf größere Projekte oder höhere Verarbeitungsmengen auszuweiten.
9. **Sicherheit und Datenschutz:** Die Sicherheitsmaßnahmen der Software, insbesondere wenn sie mit sensiblen oder vertraulichen Daten arbeitet.
10. **Zukunftssicherheit:** Die Wartbarkeit und Aktualisierbarkeit der Software sowie die Wahrscheinlichkeit von zukünftigen Weiterentwicklungen und Updates.
11. **Benutzerfreundlichkeit:** Die Benutzerfreundlichkeit der Software, einschließlich der Benutzeroberfläche und der Lernkurve für neue Benutzer.

Alle diese Kriterien werden für einen Vergleich ausführlich recherchiert und abgewogen. Es wird hier auf eine detaillierte Beschreibung der Punkte für jede einzelne Methode verzichtet, da sonst der Umfang übermäßig ansteigen würde.

17.1 Zusammenfassung der Nutzwertanalyse

Die quantitative Nutzwertanalyse lässt sich im Anhang unter [L] einsehen.

Die durchgeführte Nutzwertanalyse hebt die Vorzüge der Open-Source-Lösung im Vergleich zu den kommerziellen Alternativen deutlich hervor. Insbesondere in den Bereichen niedrige Kosten, Sicherheit und Datenschutz, Anpassungsmöglichkeiten, Genauigkeit und Zukunftssicherheit erweist sich die Open-Source-Option als überlegen.

Unterdessen konnte die Lösung von Merlic von MVTec, insbesondere im Bereich Benutzerfreundlichkeit, überzeugen. Astrocyte konnte in den Bereichen Dokumentation und Integration viele Punkte sammeln. Es ist erwähnenswert, dass beide kommerzielle Produkte in den Bereichen Anpassungsmöglichkeit, Support, Skalierbarkeit sowie Sicherheit und Datenschutz dieselbe Punktzahl aufweisen.

Jedoch sollte stets im Hinterkopf behalten werden, dass im Falle einer tatsächlichen Implementierung eine eingehende Untersuchung der einzelnen Optionen unabdingbar ist. Die Entscheidung sollte auf individueller Basis getroffen werden und dabei die spezifischen Anforderungen, verfügbaren Ressourcen und übergeordneten Ziele angemessen berücksichtigen.

18. Einbindung von Open-Source in die bestehende Systemlandschaft

Die Open-Source-Lösungen Anomalib und EfficientNetV2-S haben sich auch im direkten Vergleich mittels Nutzwertanalyse von den kommerziellen Varianten Merlic und Astrocyte abheben können. Allerdings sind für die Open-Source-Lösungen bisher im industriellen Sondermaschinenbau keine Ansätze zur Implementierung vorhanden. Daher widmet sich dieses Kapitel potenzieller Ansätze zur Integration in die Systemlandschaft des industriellen Sondermaschinenbaus.

Es ergeben sich zwei mögliche Alternativen für die Implementierung solcher Algorithmen:

Alternative 1:

Betrieb der Algorithmen auf demselben System wie die bisherige BV.

Diese Option verfolgt die Idee, Anomalib und EfficientNetV2-S auf demselben Computer auszuführen, der gleichzeitig von den bisherigen Algorithmen der BV genutzt wird.

Alternative 2:

Betrieb der Algorithmen auf einem separaten System, das parallel zur Fertigung betrieben wird.

Diese Alternative setzt darauf, Anomalib und EfficientNetV2-S in einem eigenständigen System laufen zu lassen, das parallel zum Produktionsprozess arbeitet.

18.1.1 Vorteile Alternative 1

- Keine zusätzlichen Computer notwendig. Bestehendes System kann genutzt werden.
- Potenzielle Nutzung der bestehenden Kameras.
- Bereits trainierte Modelle für die NanoBlade und Parkscheibe können verwendet werden.

18.1.2 Nachteile Alternative 1

- Ausfall des Systems kann Ausfall der Fertigung bedeuten.
- Schwierige Einrichtung der Software.
- Komplikationen des gleichzeitigen Bildeinzugs der neuen Algorithmen und der bisherigen BV.
- Geringe Leistungsfähigkeit der Systeme in der Fertigung.
- Arbeit nur während Stillstand der Maschine möglich.
- Komplexere Startsequenz der Maschine.

18.2.1 Vorteile Alternative 2

- Nach Bedarf abschaltbar.
- Absturz der Algorithmen bedeutet keinen Ausfall der Fertigung.
- Testbetrieb möglich.
- Einrichtung nicht an Stillstand der Maschine gebunden.
- Computer mit höherer Leistungsfähigkeit möglich.
- Erleichterte Fehlersuche.

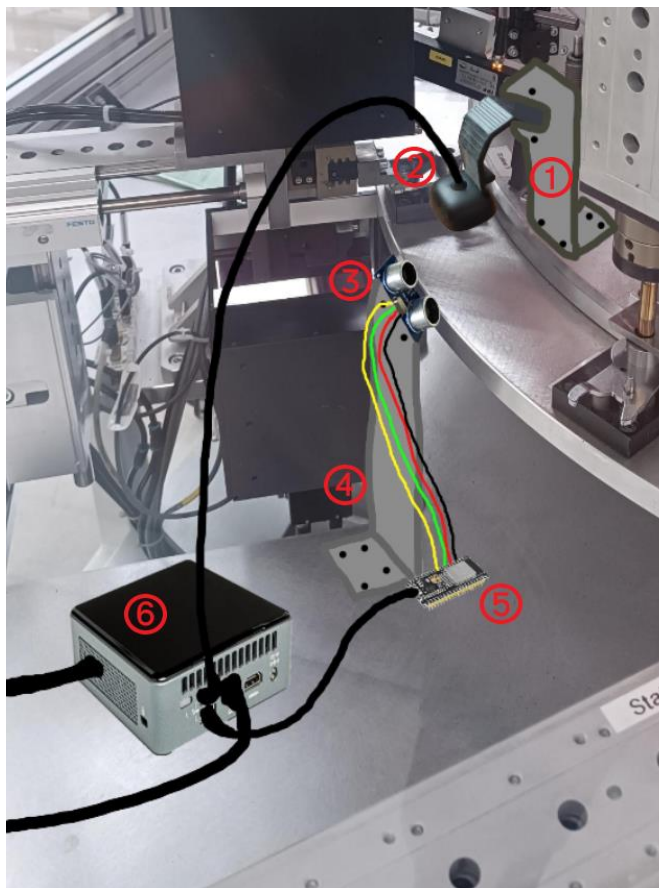
18.2.2 Nachteile Alternative 2

- Nutzung einer separaten Kamera notwendig.
- Erstellen eines neuen Datensatzes.
- Neuer Computer nötig.
- Verdrahtung mit SPS notwendig.
- Bildeinzug muss an Taktzeit der Maschine angepasst sein.

Nach sorgfältiger Abwägung der beiden Alternativen zeigt sich, dass Alternative 2 die bevorzugte Wahl darstellt. Die Entscheidung basiert auf mehreren Faktoren, darunter das minimierte Risiko eines Produktionsausfalls sowie die physische Trennung des neuen experimentellen Systems.

18.3 Umsetzung von Alternative 2

Angesichts der Tatsache, dass die Algorithmen eine Auflösung von 256x256 Pixeln verwenden, bietet sich die Verwendung einer einfachen Webcam als Kamera an. Diese kann über USB an einen Computer eigener Wahl angeschlossen werden. Bei der Auswahl eines geeigneten Rechners ist es ratsam, entweder auf einen mit einer Nvidia-Grafikkarte oder einer neueren Generation von Intel-CPU's zu setzen. Anomalib kann Inference auf beiden Systemen durchführen, während EfficientNetV2-S eine Nvidia-Grafikkarte erfordert. Das System kann anschließend Bilder über die Webcam erfassen und intern auf der Hardware verarbeiten.



- ① Webcamhalterung
- ② Webcam
- ③ Ultraschall Näherungssensor
- ④ Sensorhalterung
- ⑤ ESP32
- ⑥ Computer

Abbildung 23: Konzeption des neuen Bildverarbeitungssystems

Ein mögliches Konzept für solch ein System kann in Abb. 23 betrachtet werden.

Ein entscheidendes Anliegen betrifft das Timing des Bildeinzugs. Dies sollte mit der Taktzeit synchronisiert sein, um im richtigen Moment ein Bild aufzunehmen. Hier bietet sich die Verwendung eines einfachen Schaltkreises an, der einen ESP32 und einen Ultraschall-Näherungssensor umfasst.

Diese unkomplizierte Lösung ist in der Lage, Teile effektiv zu lokalisieren und bei Vorhandensein eines Teils an der richtigen Stelle, ein Auslösesignal an die Kamera zu senden.

Der Computer muss an die SPS (Speicherprogrammierbare Steuerung) angebunden werden, damit das interne Steuersystem Teile entsprechend den Ausgaben der Algorithmen effektiv sortieren kann. Da die Computer bereits im Besitz von Kern-Liebers sind, würden sich die Kosten nur auf den ESP32, die Webcam und den Ultraschallsensor beziehen. Die Artikel sollten sich zusammen auf unter 50€ belaufen.

Dieses System sollte in der Lage sein, die Leistungsfähigkeit der Algorithmen effektiv zu nutzen und eine hohe Genauigkeit bei der Sortierung zu gewährleisten. Darüber hinaus eignet es sich gut für Testläufe und kann parallel zur Maschine entwickelt werden.

19. Monetäre Bewertung der neuen Methode

Für einen finalen Abschluss wird nun eine monetäre Bewertung einer potenziellen Implementierung der neuen Prüfmethode durchgeführt. Dies ist äußerst wichtig, da am Ende ein finanzieller Aspekt im industriellen Sondermaschinenbau am stärksten im Vordergrund steht. Ein finanzieller Vorteil kann sich auf zwei Arten ergeben.

1. Zeitersparnis

Wird weniger Zeit für die Implementierung aufgewendet, resultiert durch kürzere Arbeitszeit des Personals ein direkter finanzieller Vorteil.

2. Erhöhte Genauigkeit

Durch eine höhere Genauigkeit sinkt der Pseudoausschuss, wodurch aufwändige Nacharbeit sinkt und mehr Teile verkauft werden können.

Im Verlauf der Untersuchung hat sich bedauerlicherweise gezeigt, dass eine präzise monetäre Bewertung der verbesserten Genauigkeit äußerst anspruchsvoll ist. Als Ergebnis dieser Erkenntnis wurde im Einvernehmen beschlossen, auf eine detaillierte Kalkulation zu verzichten.

19.1 Zeitersparnis

Die Arbeit mit der NanoBlade hat hervorgebracht, wie mächtig moderne KI-Lösungen geworden sind. Im Rahmen der Untersuchung wurde ein Datensatz präpariert, ein Modell trainiert und auf neuen Daten evaluiert. Hierfür wird beispielhaft das EfficientNetV2-S Modell gewählt. Es konnte die beste F1-Metrik erreichen und eignet sich daher am besten für eine Implementierung.

Der gesamte Zeitaufwand für das Erstellen eines Modells im Vergleich zur bisherigen BV kann im Anhang unter [M] eingesehen werden.

Die Tabelle verdeutlicht klar, dass die modernere Methode sowohl in puncto Zeit als auch in Bezug auf die Kosten einen signifikanten Vorteil gegenüber der bisher verwendeten BV aufweist. Die Zeitaufwendung beläuft sich hierbei auf fast ein Drittel im Vergleich zu denen der herkömmlichen BV. Dieser Vergleich veranschaulicht eindrucksvoll, wie KI-Lösungen im Bereich der BV erhebliche Zeit- und Kosteneinsparungen ermöglichen können.

20. Fazit

Diese Arbeit veranschaulicht eindrucksvoll, wie moderne Bildverarbeitungsalgorithmen, die auf KI basieren, erfolgreich in die Systemlandschaft des industriellen Sondermaschinenbaus integriert werden können. In den Tests konnte sich EfficientNetV2-S bei Klassifizierungsaufgaben klar vor die Kontrahenten Merlic von MVTec und Astrocyte von Teledyne Dalsa setzen. Zudem wurde bewiesen, dass im Bereich der Anomalieerkennung sich Anomalib ebenfalls vor die oben genannten Produkte setzen kann. Auch in Anbetracht anderer quantitativer Faktoren können sich Open-Source-Lösungen klar vor kommerziellen Produkten platzieren. Diese Ergebnisse verdeutlichen eindrucksvoll, dass sich Open-Source-Lösungen für die Bildverarbeitung in einem industriellen Sondermaschinenbau am besten eignen.

21. Ausblick

Diese Arbeit legt den Grundstein für zukünftige Forschung im Bereich der KI im industriellen Sondermaschinenbau. Die präsentierten Ergebnisse bieten eine solide Basis für die Entwicklung und Erprobung eines Prototyps, der auf Open-Source-Algorithmen wie Anomalib oder EfficientNetV2-S basiert. Dieser Schritt ist von großer Bedeutung, um die geforderte Dynamik und Agilität in der Fertigung zu gewährleisten, die der Markt auch zukünftig verlangt.

Darüber hinaus spielt die Gewinnung der Belegschaft für solche Methoden eine entscheidende Rolle. KI besitzt das Potenzial, bestehende Verfahren deutlich zu beschleunigen und zu verbessern. Daher ist es unerlässlich, mit diesen Technologien zu arbeiten, sie zugänglich zu machen und die Vorteile solcher Methoden anschaulich zu demonstrieren und umzusetzen.

22. Literatur

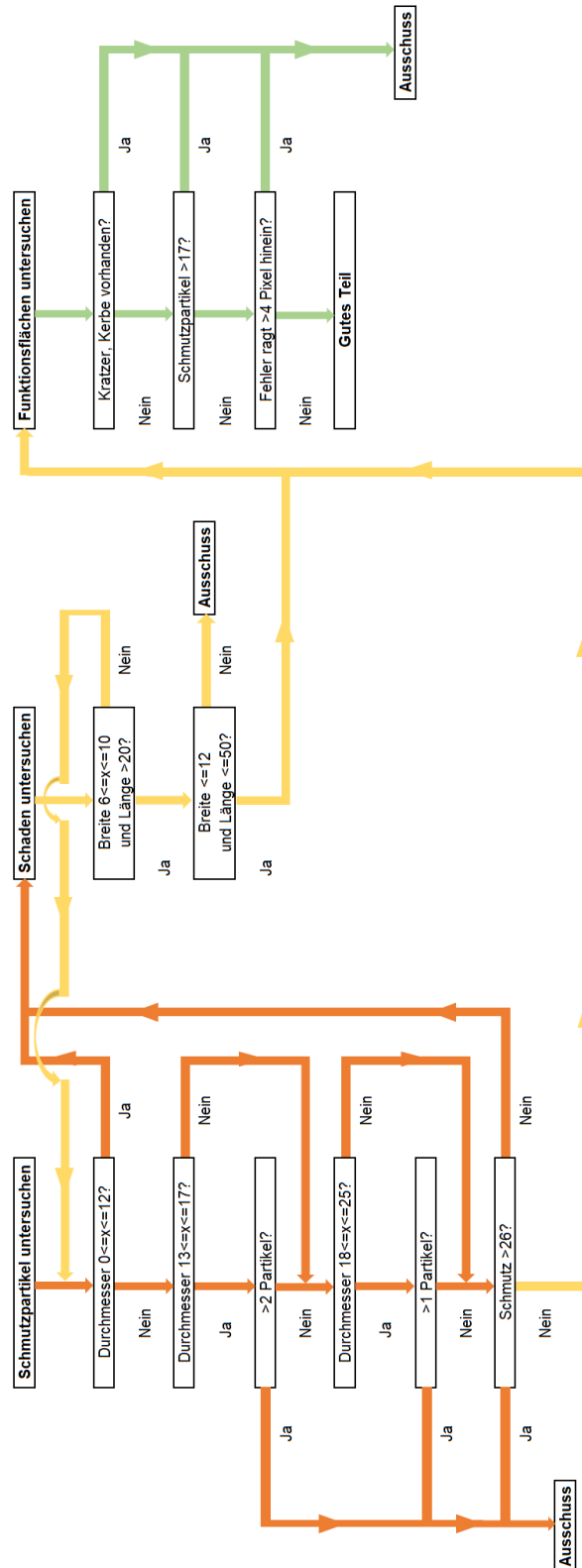
- [1] Büchner Lichtsysteme GmbH. „Diffuse Hintergrundbeleuchtung.” <https://buechnerlichtsysteme.de/wiki-led/diffuse-hintergrundbeleuchtung/> (Zugriff am: 28. August 2023).
- [2] „Koaxialbeleuchtung | Optische Beleuchtungstechniken | Glossar zur Mikroskopie | KEYENCE Deutschland.” https://www.keyence.de/ss/products/microscope/microscope_glossary/lighting/coaxial_lighting.jsp (Zugriff am: 28. August 2023).
- [3] Bosch Global. „NanoBlade - Kleine Kette, große Wirkung.” <https://www.bosch.com/de/stories/nanoblade-kleine-kette-grosse-wirkung/> (Zugriff am: 28. August 2023).
- [4] „Sherlock | Teledyne DALSA.” <https://www.teledynedalsa.com/en/products/imaging/vision-software/sherlock/> (Zugriff am: 28. August 2023).
- [5] J. Yang, R. Xu, Z. Qi und Y. Shi, „Visual Anomaly Detection for Images: A Systematic Survey,” *Procedia Computer Science*, Jg. 199, S. 471–478, 2022, doi: 10.1016/j.procs.2022.01.057.
- [6] C. Janiesch, P. Zschech und K. Heinrich, „Machine learning and deep learning,” *Electron Markets*, Jg. 31, Nr. 3, S. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.
- [7] J. Zipfel, F. Verworner, M. Fischer, U. Wieland, M. Kraus und P. Zschech, „Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning models,” *Computers & Industrial Engineering*, Jg. 177, S. 109045, 2023, doi: 10.1016/j.cie.2023.109045.
- [8] Y. Lecun, L. Bottou, Y. Bengio und P. Haffner, „Gradient-based learning applied to document recognition,” *Proc. IEEE*, Jg. 86, Nr. 11, S. 2278–2324, 1998, doi: 10.1109/5.726791.
- [9] A. Krizhevsky, I. Sutskever und G. E. Hinton, „ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, Jg. 60, Nr. 6, S. 84–90, 2017, doi: 10.1145/3065386.
- [10] K. Simonyan und A. Zisserman, „Very Deep Convolutional Networks for Large-Scale Image Recognition,” Sep. 2014. [Online]. Verfügbar unter: <https://arxiv.org/pdf/1409.1556>
- [11] K. O'Shea und R. Nash, „An Introduction to Convolutional Neural Networks,” Nov. 2015. [Online]. Verfügbar unter: <http://arxiv.org/pdf/1511.08458v2>
- [12] J. Liu *et al.*, „Deep Industrial Image Anomaly Detection: A Survey,” Jan. 2023. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2301.11514v4>
- [13] V. Zavrtanik, M. Kristan und D. Skočaj, „DRAEM -- A discriminatively trained reconstruction embedding for surface anomaly detection,” Aug. 2021. [Online]. Verfügbar unter: <https://arxiv.org/pdf/2108.07610v2>
- [14] K. He, X. Zhang, S. Ren und J. Sun, „Deep Residual Learning for Image Recognition,” Dez. 2015. [Online]. Verfügbar unter: <http://arxiv.org/pdf/1512.03385v1>
- [15] M. Tan und Q. Le V, „EfficientNetV2: Smaller Models and Faster Training,” *International Conference on Machine Learning*. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2104.00298v3>
- [16] K. Bätzner, L. Heckler und R. König, „EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies,” Mrz. 2023. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2303.14535v2>

- [17] D. Gudovskiy, S. Ishizaka und K. Kozuka, „CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows,“ Jul. 2021. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2107.12571v1>
- [18] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox und P. Gehler, „Towards Total Recall in Industrial Anomaly Detection,“ Jun. 2021. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2106.08265v2>
- [19] T. Defard, A. Setkov, A. Loesch und R. Audigier, „PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization,“ Nov. 2020. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2011.08785v1>
- [20] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger und C. Steger, „The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,“ *Int J Comput Vis*, Jg. 129, Nr. 4, S. 1038–1059, 2021, doi: 10.1007/s11263-020-01400-4.
- [21] S. Lee, S. Lee und B. C. Song, „CFA: Coupled-hypersphere-based Feature Adaptation for Target-Oriented Anomaly Localization,“ Jun. 2022. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2206.04325v1>
- [22] OpenAI, „GPT-4 Technical Report,“ Mrz. 2023. [Online]. Verfügbar unter: <https://arxiv.org/pdf/2303.08774>
- [23] M. M. Moya und D. R. Hush, „Network constraints and multi-objective optimization for one-class classification,“ *Neural Networks*, Jg. 9, Nr. 3, S. 463–474, 1996, doi: 10.1016/0893-6080(95)00120-4.
- [24] J. Hedderich, *Angewandte Statistik: Methodensammlung mit R*, 16. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. [Online]. Verfügbar unter: <http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1566808>
- [25] T. G. Dietterich, „Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms,“ *Neural computation*, Jg. 10, Nr. 7, S. 1895–1923, 1998, doi: 10.1162/089976698300017197.
- [26] T. Fawcett, „An introduction to ROC analysis,“ *Pattern Recognition Letters*, Jg. 27, Nr. 8, S. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.
- [27] S. Bozinovski, „Reminder of the First Paper on Transfer Learning in Neural Networks, 1976,“ *IJCAI*, Jg. 44, Nr. 3, 2020, doi: 10.31449/inf.v44i3.2828.
- [28] B. Christian und T. Griffiths, *Algorithms to live by: The computer science of human decisions*. New York: Henry Holt and Company, 2016.
- [29] S. Basodi, C. Ji, H. Zhang und Y. Pan, „Gradient amplification: An efficient way to train deep neural networks,“ *Big Data Min. Anal.*, Jg. 3, Nr. 3, S. 196–207, 2020, doi: 10.26599/BDMA.2020.9020004.
- [30] J. Terven und D. Cordova-Esparza, „A Comprehensive Review of YOLO: From YOLOv1 and Beyond,“ Apr. 2023. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2304.00501v4>
- [31] „Papers with Code - EfficientNet Explained.“ <https://paperswithcode.com/method/efficientnet> (Zugriff am: 29. August 2023).
- [32] K. Team. „Keras documentation: KerasCV API.“ https://keras.io/api/keras_cv/ (Zugriff am: 29. August 2023).
- [33] K. Team. „Keras documentation: KerasCV Models.“ https://keras.io/api/keras_cv/models/ (Zugriff am: 29. August 2023).
- [34] GitHub. „openvinotoolkit/anomalib: An anomaly detection library comprising state-of-the-art algorithms and features such as experiment management, hyper-parameter optimization, and edge inference.“ <https://github.com/openvinotoolkit/anomalib> (Zugriff am: 29. August 2023).

- [35] „Slower inference/testing with EfficientAD than promised. · openvinotoolkit/anomalib · Discussion #1183.” <https://github.com/openvinotoolkit/anomalib/discussions/1183> (Zugriff am: 29. August 2023).
- [36] „MVTec Software GmbH - Experten für Machine Vision.” <https://www.mvtec.com/de/> (Zugriff am: 29. August 2023).
- [37] „HALCON – Machine Vision Software: MVTec Software.” <https://www.mvtec.com/de/produkte/halcon> (Zugriff am: 29. August 2023).
- [38] „MERLIC - die easy-to-use Machine Vision Software.” <https://www.mvtec.com/de/produkte/merlic/> (Zugriff am: 29. August 2023).
- [39] MVTec Software GmbH. „Deep Learning Tool: MVTec Software.” <https://www.mvtec.com/de/produkte/deep-learning-tool> (Zugriff am: 29. August 2023).
- [40] „Home | Teledyne DALSA.” <https://www.teledynedalsa.com/en/home/> (Zugriff am: 29. August 2023).
- [41] C. Zangemeister, *Nutzwertanalyse in der Systemtechnik: E. Methodik zur multidimensionalen Bewertung u. Ausw. von Projektalternativen*, 4. Aufl. Hamburg: Zangemeister, 1976.

Anhang

[A] Flowchart für Klassifizierung der Parkscheibe



[B] Rangliste der Anomalib Modelle und ihre Genauigkeit

Model	Extractor	Avg
EfficientAD	PDN-S	0.982
EfficientAD	PDN-M	0.975
PatchCore	Wide ResNet-50	0.980
PatchCore	ResNet-18	0.973
CFlow	Wide ResNet-50	0.962
CFA	Wide ResNet-50	0.956
CFA	ResNet-18	0.930
PaDiM	Wide ResNet-50	0.950
PaDiM	ResNet-18	0.891
DFM	Wide ResNet-50	0.943
DFM	ResNet-18	0.936
STFPM	Wide ResNet-50	0.876
STFPM	ResNet-18	0.893
DFKDE	Wide ResNet-50	0.774
DFKDE	ResNet-18	0.762
GANomaly		0.421

[C] Genauigkeit Anomalib NanoBlades

NanoBlades	AUROC	F1-Metrik	Präzision	Trefferquote
EfficientAD	99,7%	96,1%	100,0%	92,4%
PatchCore	-	-	-	-
CFA	99,9%	96,3%	100,0%	92,9%
CFlow	97,4%	93,2%	98,5%	88,5%
PaDim	99,9%	95,2%	100,0%	90,9%
EfficientNetV2-S (K)	-	98,1%	97,1%	99,1%
Bisherige BV	-	93,3%	99,5%	87,9%

[D] Genauigkeit Anomalib Parkscheibe

Parkscheibe	AUROC	F1-Metrik	Präzision	Trefferquote
EfficientAD	79,9%	68,8%	97,3%	53,2%
PatchCore	-	-	-	-
CFA	62,1%	67,1%	97,0%	51,3%
CFlow	64,9%	69,3%	97,7%	53,7%
PaDim	83,3%	73,9%	93,9%	60,9%
EfficientNetV2-S (K)	-	78,5%	85,7%	72,3%
Bisherige BV	-	72,0%	89,0%	60,5%

[E] Zeiten Anomalib

Zeiten	Trainingszeit*	Inferencezeit	FPS
EfficientAD	2h 26 min	16,4 ms	61
PatchCore	-	-	-
CFA	25 min	23,8 ms	42
CFlow	53 min	155 ms	6,5
PaDim	4 min	3,4 ms	294
EfficientNetV2-S (K)	1h 10 min	31 ms	32

(*Trainingszeit bezieht sich auf NanoBlade-Datensatz mit 1000 Trainingsbildern)

[F] Genauigkeit DLT NanoBlades

NanoBlade

Klassifikation	AUROC	F1-Metrik	Präzision	Trefferquote
DLT	-	97,3%	96,5%	98,2%
EfficientNetV2-S	-	98,1%	97,1%	99,1%
Δ Delta	-	0,8%	0,6%	0,9%
Anomalieerkennung	AUROC	F1-Metrik	Präzision	Trefferquote
DLT	-	96,2%	100,0%	92,7%
CFA	99,9%	96,3%	100,0%	92,9%
Δ Delta	-	0,1%	0,0%	0,2%

[G] Genauigkeit DLT Parkscheibe

Parkscheibe

Klassifikation	AUROC	F1-Metrik	Präzision	Trefferquote
DLT	-	76,0%	84,5%	70,7%
EfficientNetV2-S	-	78,5%	85,7%	72,3%
Δ Delta	-	2,5%	1,2%	2,3%
Anomalieerkennung	AUROC	F1-Metrik	Präzision	Trefferquote
DLT	-	68,7%	96,1%	53,5%
PaDim	83,3%	73,9%	93,9%	60,9%
Δ Delta	-	5,2%	2,2%	7,4%

[H] Zeiten DLT

Zeiten

Klassifikation	Trainingszeit*	Inferencezeit	FPS
DLT	57 min	14 ms	71
EfficientNetV2-S	1h 10 min	31 ms	32
Δ Delta	13 min	17 ms	39
Anomalieerkennung	Trainingszeit	Inferencezeit	FPS
DLT	1h 32 min	17,9 ms	56
PaDim	4 min	3,4 ms	294
Δ Delta	1h 28 min	14,5 ms	238

(*Trainingszeit bezieht sich auf NanoBlade-Datensatz mit 1000 Trainingsbildern)

[I] Genauigkeit Astrocyte NanoBlade

NanoBlade

Klassifikation	AUROC	F1-Metrik	Präzision	Trefferquote
Astrocyte	99,3%	-	-	-
EfficientNetV2-S	-	98,1%	97,1%	99,1%
Δ Delta	-	-	-	-
Anomalieerkennung	AUROC	F1-Metrik	Präzision	Trefferquote
Astrocyte	98,5%	-	-	-
CFA	99,9%	96,30%	100,00%	92,90%
Δ Delta	1,4%	-	-	-

[J] Genauigkeit Astrocyte Parkscheibe

Parkscheibe

Klassifikation	AUROC	F1-Metrik	Präzision	Trefferquote
Astrocyte	82,7%	-	-	-
EfficientNetV2-S	-	78,5%	85,7%	72,3%
Δ Delta	-	-	-	-
Anomalieerkennung	AUROC	F1-Metrik	Präzision	Trefferquote
Astrocyte	73,2%	-	-	-
PaDim	83,3%	73,9%	93,9%	60,9%
Δ Delta	10,1%	-	-	-

[K] Zeiten Astrocyte

Zeiten

Klassifikation	Trainingszeit*	Inferencezeit	FPS
Astrocyte	59 min	18 ms	56
EfficientNetV2-S	1h 10 min	31 ms	32
Delta	11 min	13 ms	29
Anomalieerkennung	Trainingszeit	Inferencezeit	FPS
Astrocyte	9 min	7,9 ms	126
Padim	4 min	3,4 ms	294
Delta	5 min	4,5 ms	168

(*Trainingszeit bezieht sich auf NanoBlade-Datensatz mit 1000 Trainingsbildern)

[L] Nutzwertanalyse der Bildverarbeitungs-Software

Nutzwertanalyse	Gewichtung	Merlic	Astrocyte	Open-Source
Niedrige Kosten	10	4	2	10
Genauigkeit	9	8	7	9
Leistungsfähigkeit	7	7	8	8
Anpassungsmöglichkeiten	7	8	8	9
Support	8	8	8	6
Dokumentation	6	8	9	5
Integration	9	7	9	5
Skalierbarkeit	7	8	8	8
Sicherheit und Datenschutz	7	9	9	10
Zukunftssicherheit	8	7	8	9
Benutzerfreundlichkeit	7	9	7	5
Σ Gewichtete Summe		630	626	656

[M] Zeitersparnis von KI im industriellen Sondermaschinenbau

Vergleich Zeitaufwand (NanoBlades)	EfficientNetV2-S	Bisherige BV
Bilder Loggen aktivieren	1 h	-
Datensatz manuell klassifizieren	4 h	-
Modell trainieren	1,5 h	-
Modell evaluieren	0,5 h	-
Summe Zeit	7 h	20 h
Vergleich Kosten	EfficientNetV2-S	Bisherige BV
Stundenlohn (EG10)	30 €	30 €
Gesamtkosten	210 €	600 €

[N] Bildquellen

Abb. 1, Bosch NanoBlade:

<https://www.gardenmachinesltd.co.uk/ekmps/shops/garmac ltd/images/bosch-nanoblade-wood-speed-65-saw-blade-for-easycut-12-ec12-135725-p.png>

Eingesehen am 28.08.23

Abb. 2, Diffuse Durchlichtbeleuchtung:

https://www.vision-doctor.com/images/stories/beleuchtung/durchlicht/durchlicht_prinzip.jpg

Eingesehen am 28.08.23

Abb. 4, Koaxialbeleuchtung:

https://wahtari.io/wp-content/uploads/2022/01/koaxialbeleuchtung_Zeichenfläche-1.png

Eingesehen am 28.08.23

Abb. 5, Sherlock 7 Benutzeroberfläche:

<https://www.teledynedalsa.com/download/eab28ede-d154-4cca-9c53-e1c103584198/>

Eingesehen am 28.08.23

Abb. 7, Bildung des AUROC-Wertes

https://upload.wikimedia.org/wikipedia/commons/thumb/4/4f/ROC_curves.svg/1280px-ROC_curves.svg.png

Eingesehen am 28.09.23

Abb. 8, Funktionsweise eines CNN:

https://upload.wikimedia.org/wikipedia/commons/thumb/6/63/Typical_cnn.png/790px-Typical_cnn.png

Eingesehen am 28.09.23

Abb. 9, Autoencoder-Netzwerk für rekonstruktionsbasierten Ansatz

https://de.mathworks.com/discovery/autoencoder/_jcr_content/mainParsys/image.adapt.480.medium.svg/1665035676699.svg

Eingesehen am 28.09.23

Abb. 10, Feature Embedding Methode:

Entnommen aus [12]

Abb. 11, Anomalib Logo:

<https://raw.githubusercontent.com/openvinotoolkit/anomalib/main/docs/source/images/logos/anomalib-wide-blue.png>

Eingesehen am 28.09.23

Abb. 12, MVTec Logo:

https://tse4.mm.bing.net/th?id=OIP.YVOdprwg2e4ojN_nZB8CiAHaDB&pid=Api

Eingesehen am 28.09.23

Abb. 13, Teledyne Dalsa Logo:

https://tse4.mm.bing.net/th?id=OIP.ojBK8TbTvmBu_x8-MlaxFwHaBa&pid=Api

Eingesehen am 28.09.23

[O] Code Verzeichnis und Dateien

GitHub Repo Link:

https://github.com/Palettenbrett/Bachelor_Arbeit_05_09_2023