Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Shahrukh Khan

Einsatz von Deep Learning für die Qualitätskontrolle in der Fertigung

Fakultät Technik und Informatik
Department Maschinenbau und Produktion

Faculty of Engineering and Computer Science
Department of Mechanical Engineering and Production Management

Shahrukh Khan

Einsatz von Deep Learning für die Qualitätskontrolle in der Fertigung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Produktionstechnik und -management
am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer/in: Prof. Dr. -Ing Mauricio Porath
Zweitprüfer/in: Prof. Dr. Alexander Koch

Abgabedatum: 31.05.2023

Zusammenfassung

Shahrukh Khan

Thema der Bachelorthesis

Einsatz von Deep Learning für die Qualitätskontrolle in der Fertigung

Stichworte

Künstliche Intelligenz, Deep Learning, Qualitätskontrolle, Modellentwicklung, Neuronale Netze, Computer Vision

Kurzzusammenfassung

Diese Bachelorarbeit untersucht den Einsatz von Deep Learning für die Qualitätskontrolle in der Fertigung. Durch die Einbindung von Künstlicher Intelligenz (KI) und Deep Learning in Qualitätssicherungs- und Kontrollprozesse wird das Potenzial für eine signifikante Verbesserung der industriellen Fertigung aufgezeigt. Die Arbeit präsentiert Qualität als dynamisches Konzept, beleuchtet die Rolle und Herausforderungen der Qualitätssicherung in der Fertigung und führt in die Konzepte von KI und Deep Learning ein, wobei ein Schwerpunkt auf der Mustererkennung in komplexen Daten liegt. Konkrete Anwendungen von Deep Learning in der Qualitätskontrolle sowie die Implementierung eines Deep Learning-Modells werden detailliert dargestellt und diskutiert. Die Ergebnisse zeigen, dass Deep Learning eine innovative und effektive Methode zur Steigerung der Effizienz und Genauigkeit in der Fertigung darstellt, wobei sowohl die Vorteile als auch die Herausforderungen dieser Technologie beleuchtet werden.

Title of the paper

Application of deep learning for quality control in manufacturing.

Keywords

Artificial intelligence, deep learning, quality control, model development, neural networks, computer vision

Abstract

This Bachelor's thesis examines the application of Deep Learning for quality control in manufacturing. By incorporating Artificial Intelligence (AI) and Deep Learning into quality assurance and control processes, the potential for a significant improvement in industrial manufacturing is demonstrated. The work presents quality as a dynamic concept, highlights the role and challenges of quality assurance in manufacturing, and introduces the concepts of AI and Deep Learning, with a focus on pattern recognition in complex data. Specific applications of Deep Learning in quality control as well as the implementation of a Deep Learning model are detailed and discussed. The results show that Deep Learning is an innovative and effective method to enhance efficiency and accuracy in manufacturing, shedding light on both the benefits and challenges of this technology.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

CNN	Convolutional Neural Network
CPU	Central Processing Units
CV	Computer Vision
DIN	Deutsches Institut für Normung
DL	Deep Learning
GPU	Graphics Processing Units
ISO	Internationale Organisation für Normung
KI	Künstliche Intelligenz
KNN	Künstliche Neuronale Netze
ML	Machine Learning
MSE	Mean Squared Error
NLP	Natural Language Processing
QMS	Qualitätsmanagementsystem
QS	Qualitätssicherung
SHAP	SHapley Additive exPlanations
TQM	Total Quality Management

1 Einleitung

1.1 Hintergrund des Themas

In der modernen Fertigungsindustrie gewinnt die Qualitätssicherung zunehmend an Bedeutung, da die Anforderungen der Kunden und regulatorischen Vorschriften immer höher werden. Die globale Fertigungsindustrie wird auf einen Wert von etwa 12 Billionen US-Dollar im Jahr 2017 geschätzt und ist ein entscheidender Wirtschaftsfaktor.[1] Fehlerhafte Produkte oder minderwertige Qualitätskontrollen können erhebliche Kosten verursachen und das Vertrauen der Kunden in die Produkte und Marken beeinträchtigen.[2] Laut einer Studie belaufen sich die Kosten für mangelhafte Qualität in der Fertigung auf jährlich etwa 2,4 Billionen US-Dollar weltweit, was etwa 20% des gesamten Umsatzes der Fertigungsindustrie entspricht.[3]

Angesichts der hohen Kosten und der wachsenden Bedeutung der Qualitätssicherung suchen Unternehmen nach effizienteren und präziseren Lösungen, um die Qualitätskontrolle in der Fertigung zu verbessern. Traditionelle Methoden wie manuelle Inspektionen und regelbasierte Systeme stoßen dabei an ihre Grenzen, da sie zeitaufwendig, fehleranfällig und oft nur schwer skalierbar sind. Die fortschreitende Digitalisierung und Automatisierung der Fertigungsprozesse erfordern innovative Ansätze, um den steigenden Anforderungen gerecht zu werden.[4]

In diesem Zusammenhang bieten künstliche Intelligenz (KI) und insbesondere Deep Learning (DL) vielversprechende Möglichkeiten zur Verbesserung der Qualitätskontrolle in der Fertigung.[5] Laut einer Studie von Global Market Insights wird der globale Markt für KI in der Fertigung bis 2025 auf etwa 16 Milliarden US-Dollar geschätzt, was die wachsende Bedeutung dieser Technologien in diesem Sektor unterstreicht.[6] Deep Learning, ein Teilbereich des Machine Learnings, ermöglicht es, komplexe Muster und Zusammenhänge in Daten zu erkennen und daraus Entscheidungen abzuleiten. Insbesondere im Bereich der optischen Inspektion und Computer Vision zeigt sich das Potenzial von Deep Learning-Techniken zur Erkennung von Fehlern und Abweichungen in Produktionsprozessen.[5]

1.2 Problemstellung und Zielsetzung

Die Qualitätssicherung in der Fertigung ist ein entscheidendes Element zur Gewährleistung der Produktqualität und der Erfüllung von Kundenanforderungen. Die Relevanz der

Qualitätskontrolle in der Fertigung ergibt sich unter anderem aus Kostengründen. Die Rule of ten besagt, dass sich die Kosten von unentdeckten Fehlern bei jedem weiteren Schritt im Produktionsprozess verzehnfachen, bis sie letztendlich zu einer kostspieligen Rückrufaktion des Kunden führen können.[7]

Traditionelle Qualitätskontrollmethoden, wie manuelle Inspektionen und regelbasierte Systeme, stoßen jedoch zunehmend an ihre Grenzen. Einerseits werden Qualitätskontrollen meist nur stichprobenartig durchgeführt, da sie teuer und zeitaufwendig sind. Andererseits sind automatisierte Qualitätskontrollen mit den herkömmlichen Methoden häufig sehr aufwendig, da sie hart programmiert werden müssen und nicht leicht an neue Bedingungen anpassungsfähig sind. Diese Limitationen führen dazu, dass Fehler in der Produktion unentdeckt bleiben oder nur mit erheblichem Aufwand identifiziert werden können.[8]

Die vorliegende Arbeit beschäftigt sich daher mit dem Einsatz von Deep Learning-Methoden für die Qualitätskontrolle in der Fertigung und untersucht, inwieweit diese Technologien zu einer effizienteren und präziseren Qualitätskontrolle beitragen können. Dabei werden sowohl theoretische Grundlagen als auch ein praxisnahes Beispiel betrachtet, um ein fundiertes Verständnis für die Möglichkeiten und Herausforderungen des Einsatzes von KI und Deep Learning in der Fertigungsqualitätskontrolle zu erlangen.

Das Ziel der vorliegenden Arbeit besteht darin, die Möglichkeiten und Potenziale des Einsatzes von künstlicher Intelligenz (KI), insbesondere neuronalen Netzen und Deep Learning, für die Qualitätskontrolle in der Fertigung zu untersuchen. Dabei soll erforscht werden, inwieweit der Einsatz dieser Technologien eine 100%ige Qualitätskontrolle im laufenden Betrieb ermöglichen oder zumindest vereinfachen kann. Hierzu werden folgende Forschungsfragen beantwortet:

1. **Wie können Deep Learning und neuronale Netze in der Qualitätskontrolle eingesetzt werden, um die Effizienz und Genauigkeit der Inspektionen zu verbessern?**

2. **Inwiefern können Deep Learning-basierte Ansätze bestehende Herausforderungen und Limitationen traditioneller Qualitätskontrollmethoden überwinden?**

3. **Welche Vorteile und Herausforderungen ergeben sich aus dem Einsatz von KI und Deep Learning für die Qualitätskontrolle im Vergleich zu herkömmlichen Methoden?**

Um diese Forschungsfragen zu beantworten, wird in dieser Arbeit ein praxisnahes Fallbeispiel untersucht, die den Einsatz von KI und Deep Learning für die Qualitätskontrolle in der Fertigung exemplarisch aufzeigt. Durch die Analyse des Fallbeispiels sollen sowohl das Potenzial als auch die Herausforderungen und Grenzen von Deep Learning-basierten Ansätzen für die Qualitätskontrolle aufgedeckt und bewertet werden. Dadurch soll ein fundiertes Verständnis für die Möglichkeiten und Implikationen des Einsatzes von KI und Deep Learning in der Fertigungsqualitätskontrolle geschaffen werden.

1.3 Aufbau der Arbeit

Die vorliegende Bachelorarbeit ist strukturiert in fünf Hauptteile, welche jeweils eine Reihe von Unterabschnitten enthalten. Sie beginnt mit einer Einleitung, die die Problematik beleuchtet und das Ziel der Arbeit definiert. Hier wird das Forschungsfeld umrissen und die Relevanz der gewählten Thematik verdeutlicht.

Darauf folgen die theoretischen Grundlagen, die das Fundament für das Verständnis der Arbeit bilden. Hier werden zentrale Begriffe wie Qualität, Qualitätssicherung und Qualitätskontrolle definiert. Im Anschluss daran wird eine Einführung in die Themen künstliche Intelligenz und Deep Learning gegeben, wobei die Unterscheidung zwischen künstlicher Intelligenz und Machine Learning sowie die spezifischen Merkmale und die Funktionsweise von Deep Learning erörtert werden. Der zweite Abschnitt schließt mit einer umfassenden Betrachtung der Rolle von Computer Vision und optischer Inspektion in der Fertigung ab, wobei Vorteile und Herausforderungen sowie potenzielle Anwendungsgebiete thematisiert werden.

Im dritten Teil der Arbeit liegt der Fokus auf der Methodik der vorgestellten Studie. Zunächst werden Zielsetzung und Auswahl der Datensätze sowie die Architektur des Modells dargestellt. Darauf folgt die Erklärung des technischen Aspekts der Arbeit, einschließlich der Importierung der benötigten Bibliotheken, des Zugriffs auf und Lesens von Daten, der Modellbewertung und -interpretation sowie der Datenanalyse und -visualisierung. Der Abschnitt endet mit einer Darstellung des Prozesses der

Datenvorverarbeitung und Datenaugmentation sowie der Auswahl und Konstruktion des Deep Learning-Modells.

Der vierte Abschnitt widmet sich den erzielten Ergebnissen. Hier wird die Evaluation des Modells mit Testdaten präsentiert und diskutiert. Anschließend werden die Ergebnisse mit traditionellen Methoden der Qualitätskontrolle verglichen.

Der abschließende fünfte Teil fasst die wichtigsten Erkenntnisse der Arbeit zusammen und gibt einen Ausblick auf potenzielle weitere Forschungsrichtungen in diesem Gebiet. Hier werden die Schlussfolgerungen der Arbeit gezogen und mögliche nächste Schritte skizziert.

2 Theoretische Grundlagen

Um ein fundiertes Verständnis für den Einsatz von Deep Learning-Methoden für die Qualitätskontrolle in der Fertigung zu erlangen, sind einige theoretische Grundlagen notwendig. In diesem Kapitel werden daher verschiedene Aspekte der Qualitätssicherung sowie der künstlichen Intelligenz und Deep Learning erläutert, die für die weitere Betrachtung relevant sind.

Zunächst wird die Qualitätssicherung als grundlegender Bestandteil der Fertigungsprozesse betrachtet. Im Anschluss daran werden die grundlegenden Konzepte der künstlichen Intelligenz vorgestellt, um ein besseres Verständnis für die Funktionsweise von Deep Learning zu schaffen. Hierbei wird auf die verschiedenen Arten von Machine Learning, neuronale Netze und schließlich auf die speziellen Eigenschaften von Deep Learning eingegangen. Abschließend wird Computer Vision in der Fertigung betrachtet und die Anwendungsgebiete sowie Vorteile und Herausforderungen dieser Technologien erläutert.

2.1 Definition Qualität

Qualität ist ein Konzept, das tief in der Geschichte der Industrieproduktion verankert ist. Über die Zeit hat sich die Bedeutung von Qualität stetig weiterentwickelt, um den dynamischen Bedürfnissen von Konsumenten, Industrie und Gesellschaft gerecht zu werden. In diesem Abschnitt wird eine eingehende Betrachtung der historischen Evolution des Qualitätsbegriffs und seiner aktuellen Implikationen sowie der vielfältigen Aspekte, die damit einhergehen, vorgenommen.[9]

In den frühen Phasen der Qualitätsentwicklung lag der Schwerpunkt auf der Sicherstellung der Übereinstimmung von Produkten und Prozessen und der Minimierung der Streuung in der Produktion. Wegbereiter wie Shewhart (1931) und Deming (1986) führten das Konzept der Produktkonformität und die Notwendigkeit der statistischen Prozesskontrolle ein. In diesem Kontext wurde Qualität in erster Linie als Erfüllung vordefinierter Spezifikationen und Normen interpretiert.[10]

Mit der Zeit verlagerte sich der Fokus immer mehr hin zu einer stärkeren Kundenzentrierung. Juran und Gryna (1970) entwickelten den Ausdruck "Fitness for Use" für Qualität, was impliziert, dass ein Produkt oder eine Dienstleistung den Anforderungen und Bedürfnissen des Kunden gerecht werden muss.[11] Deming (1986) unterstrich auch die

Kundenorientierung und betonte, dass Qualität den Bedürfnissen des Kunden sowohl jetzt als auch in der Zukunft gerecht werden sollte.[10]

In den darauffolgenden Jahren wurden zahlreiche Qualitätsmanagement-Methoden und -Philosophien etabliert, die die Interpretation von Qualität weiter präzisierten und erweiterten. Dazu gehören die Implementierung des Total Quality Management (TQM), des Lean Management und Six Sigma. Diese Methoden führten zu einem größeren Schwerpunkt auf kontinuierlicher Verbesserung, Kundenorientierung und der Integration von Mitarbeitern in den Qualitätsverbesserungsprozess.[12]

Heute hat die Qualität eine noch weitreichendere Definition angenommen. Sie umfasst nicht nur die Befriedigung der Kundenbedürfnisse, sondern berücksichtigt auch Aspekte der Nachhaltigkeit, gesellschaftliche Verantwortung und die Einbeziehung der Interessen verschiedener Stakeholder. Darüber hinaus wird die Rolle der Digitalisierung in der Qualität immer wichtiger, da sie neue Wege zur Datenerfassung, -analyse und -anwendung zur Verbesserung von Produkten und Prozessen ermöglicht.[13]

Die Definition von Qualität ist ein dynamisches Konzept, das sich im Laufe der Zeit und im Zuge der technologischen und wirtschaftlichen Entwicklung ständig weiterentwickelt. Laut der Internationalen Organisation für Normung (ISO) wird Qualität als "der Grad, zu dem ein Satz von inhärenten Merkmalen eines Objekts Anforderungen erfüllt" definiert. Diese Definition stellt klar, dass Qualität nicht nur von den Eigenschaften eines Produkts oder einer Dienstleistung abhängt, sondern auch von den Erwartungen und Bedürfnissen des Kunden. Es bedeutet, dass ein Qualitätsprodukt nicht nur die funktionalen und ästhetischen Anforderungen des Kunden erfüllen muss, sondern auch den festgelegten Standards und Spezifikationen entsprechen muss.[14]

In der Praxis jedoch wird eine Vielzahl von Begriffen, wie Qualität, Leistungsklasse, Qualitätsniveau, Gebrauchstauglichkeit oder umfassendes Qualitätsmanagement, häufig synonym verwendet. Diese Verwendung kann zu einer gedanklichen Vermischung von Anforderungen, Aktivitäten und Ergebnissen führen.[2]

Um eine klarere Strukturierung zu gewährleisten, hat GARVIN acht wesentliche Dimensionen der Qualität identifiziert. Diese Dimensionen ermöglichen es, die unterschiedlichen Aspekte von Qualität systematisch zu analysieren und zu bewerten. Im Folgenden

werden die acht Dimensionen von GARVIN zusammengefasst und deren Anwendbarkeit auf verschiedene Produkt- und Dienstleistungsbereiche erläutert.

1. Leistung: Hier werden die grundlegenden Leistungseigenschaften eines Produkts oder einer Dienstleistung festgelegt.
2. Merkmale (Ausstattung): Diese Dimension bezieht sich auf spezielle Eigenschaften des Produkts oder der Dienstleistung, die den Leistungseigenschaften untergeordnet sind.
3. Verlässlichkeit: Diese Dimension beschreibt die Wahrscheinlichkeit eines Produktversagens.
4. Übereinstimmung: Hier wird angegeben, inwieweit die Produkteigenschaften den festgelegten Normen und Spezifikationen entsprechen.
5. Haltbarkeit: Diese Dimension beinhaltet sowohl eine ökonomische als auch eine technische Komponente.
6. Benutzbarkeit: Diese Dimension betrifft Aspekte wie Geschwindigkeit und Reparaturfreundlichkeit.
7. Ästhetik: Die subjektive Dimension der Qualität bezieht sich auf Erscheinungsbild, Klang, Geschmack oder Geruch eines Produkts.
8. Wahrgenommene Qualität: Diese Dimension betrifft die Darstellung und Präsentation des Produkts.[15]

2.1.1 Qualitätssicherung

Qualitätssicherung (QS) ist ein zentraler Begriff in der Wissenschaft und Industrie, der sich auf die systematische Vorgehensweise bezieht, mit der eine Organisation sicherstellt, dass ihre Produkte oder Dienstleistungen die festgelegten Anforderungen und Standards erfüllen. Qualitätssicherung umfasst eine Vielzahl von Aktivitäten, die darauf abzielen, mögliche Mängel und Unzulänglichkeiten in Prozessen oder Produkten frühzeitig zu erkennen und zu beheben. Die Qualitätssicherung ist von entscheidender Bedeutung, um die Zufriedenheit von Kunden und Stakeholdern sicherzustellen und das Vertrauen in die Organisation und ihre Angebote aufrechtzuerhalten.[16]

Die Implementierung von Qualitätssicherung in Unternehmen erfolgt in der Regel durch die Einführung von Qualitätsmanagementsystemen (QMS), wie beispielsweise ISO 9001. Ein QMS stellt einen strukturierten Rahmen dar, der die Planung, Durchführung, Überwachung und Verbesserung von Prozessen und Aktivitäten unterstützt, die die Qualität von Produkten und Dienstleistungen beeinflussen. Die Implementierung eines QMS beginnt häufig mit einer Analyse der bestehenden Prozesse und der Identifizierung von

Bereichen, in denen Verbesserungen erforderlich sind. Anschließend werden Qualitätsziele und -richtlinien festgelegt, die auf die spezifischen Anforderungen und Erwartungen der Kunden und Stakeholder abgestimmt sind.[17]

Eine erfolgreiche Implementierung von Qualitätssicherung erfordert zudem die Einbindung und Schulung aller Mitarbeiter, um sicherzustellen, dass sie die Qualitätsziele und -verfahren verstehen und umsetzen können. Dies kann durch regelmäßige Kommunikation, Schulungen und die Schaffung einer Umgebung erreicht werden, die Mitarbeiter dazu ermutigt, Qualitätsprobleme zu identifizieren und Lösungen vorzuschlagen.
Die Qualitätssicherung umfasst eine Vielzahl von Elementen, darunter:

- Prozesskontrolle und -überwachung: Die systematische Überprüfung von Fertigungs- und Dienstleistungsprozessen, um sicherzustellen, dass sie die festgelegten Qualitätsstandards erfüllen.

- Prüfungen und Inspektionen: Die regelmäßige Bewertung von Produkten und Prozessen, um potenzielle Mängel oder Abweichungen von den Qualitätsstandards zu identifizieren.

- Dokumentation und Aufzeichnungen: Die Erfassung und Archivierung von Informationen über Prozesse, Verfahren und Ergebnisse, um die Rückverfolgbarkeit und Nachvollziehbarkeit der Qualitätssicherungsmaßnahmen zu gewährleisten.

- Korrekturmaßnahmen und Vorbeugungsmaßnahmen: Die Identifizierung und Behebung von Qualitätsproblemen sowie die Einführung von Maßnahmen, um zukünftige Probleme zu verhindern oder zu minimieren.

- Kontinuierliche Verbesserung: Die ständige Überprüfung und Anpassung von Prozessen und Verfahren, um die Qualität aufrechtzuerhalten und zu verbessern.[2]

Die Qualitätssicherung stellt Unternehmen vor zahlreiche Herausforderungen. Eine der größten Herausforderungen besteht darin, ein ausgewogenes Verhältnis zwischen den Anforderungen an die Qualität und den damit verbundenen Kosten zu finden. Qualitätssicherungsmaßnahmen können sowohl zeit- als auch kostenintensiv sein, und Unternehmen müssen sorgfältig abwägen, wie sie ihre Ressourcen am besten einsetzen, um die gewünschte Qualität zu erreichen, ohne ihre Rentabilität zu beeinträchtigen.[18]

Eine weitere Herausforderung besteht darin, eine Kultur der Qualität innerhalb des gesamten Unternehmens zu fördern. Dies erfordert eine starke Führung und das Engagement aller Mitarbeiter, um sicherzustellen, dass Qualitätsziele verstanden und verfolgt werden. Führungskräfte müssen aktiv an der Schaffung einer solchen Kultur arbeiten, indem sie eine klare Vision und Strategie für Qualitätssicherung vermitteln, Mitarbeiter für ihre Bemühungen zur Verbesserung der Qualität anerkennen und belohnen und bei Bedarf angemessene Ressourcen für Qualitätssicherungsinitiativen bereitstellen.[16]

Die Einbindung von Lieferanten und externen Partnern in Qualitätssicherungsprozesse stellt eine weitere Herausforderung dar. Unternehmen müssen sicherstellen, dass auch ihre Lieferanten und Partner die erforderlichen Qualitätsstandards einhalten, da Probleme in der Lieferkette die Qualität der Endprodukte oder Dienstleistungen beeinträchtigen können. Dies erfordert eine enge Zusammenarbeit, regelmäßige Kommunikation und möglicherweise die Durchführung von Audits, um die Einhaltung von Qualitätsstandards zu überprüfen.[2]

Technologische Entwicklungen, wie beispielsweise die zunehmende Digitalisierung und Automatisierung von Prozessen, bringen sowohl Chancen als auch Herausforderungen für die Qualitätssicherung mit sich. Einerseits ermöglichen Technologien wie künstliche Intelligenz, maschinelles Lernen und Big Data eine effizientere und präzisere Überwachung und Analyse von Prozessen und Ergebnissen. Andererseits können diese Technologien auch neue Qualitätsprobleme aufwerfen, wie beispielsweise die Sicherheit und Zuverlässigkeit von Software und digitalen Systemen.[13]

2.1.2 Qualitätskontrolle

Qualitätskontrolle bezieht sich auf die systematische Überwachung und Inspektion von Produkten, Dienstleistungen oder Prozessen, um sicherzustellen, dass sie den festgelegten Anforderungen und Standards entsprechen. Die Hauptziele der Qualitätskontrolle sind die Identifizierung von Abweichungen oder Fehlern, die Korrektur von Mängeln und die Verbesserung der Produkt- oder Prozessqualität. In diesem Zusammenhang sind einige Besonderheiten und Herausforderungen der Qualitätskontrolle hervorzuheben.[19]

Die Besonderheiten der Qualitätskontrolle umfassen die verschiedenen Methoden und Techniken, die zur Überwachung und Messung der Qualität eingesetzt werden. Dazu gehören statistische Prozesskontrolle, Stichprobenprüfung, Prüfung von Einheiten auf

Fehler und die Verwendung von Prüf- und Messinstrumenten. Eine erfolgreiche Qualitätskontrolle erfordert eine sorgfältige Planung, die Festlegung klarer Qualitätsziele und die Implementierung von geeigneten Kontrollverfahren.[20]

Eine der Herausforderungen der Qualitätskontrolle besteht darin, die richtige Balance zwischen den Kosten der Qualitätskontrolle und den Vorteilen einer verbesserten Produkt- oder Prozessqualität zu finden. Eine zu strenge Qualitätskontrolle kann zu hohen Kosten und Verzögerungen in der Produktion führen, während eine zu nachlässig ausgeführte Kontrolle das Risiko erhöht, dass fehlerhafte Produkte auf den Markt gelangen und die Kundenzufriedenheit beeinträchtigen.[18]

Die Implementierung der Qualitätskontrolle in Unternehmen erfordert die Einbindung aller Mitarbeiter in den Qualitätsprozess. Dies kann durch die Schaffung einer Unternehmenskultur erreicht werden, die die Bedeutung von Qualität betont und Mitarbeiter dazu ermutigt, an der kontinuierlichen Verbesserung von Produkten und Prozessen teilzunehmen. Die Etablierung von Qualitätskontrollstandards und die regelmäßige Überprüfung dieser Standards sind ebenfalls wichtige Bestandteile einer erfolgreichen Qualitätskontrolle. Darüber hinaus sollte die Implementierung von Qualitätskontrollmaßnahmen durch Schulungen und Weiterbildungen für Mitarbeiter unterstützt werden, um sicherzustellen, dass sie über das notwendige Wissen und die Fähigkeiten verfügen, um die festgelegten Qualitätsstandards einzuhalten.[16]

Qualitätskontrolle ist aus mehreren Gründen wichtig. Erstens trägt sie dazu bei, die Kundenzufriedenheit zu erhöhen, indem sichergestellt wird, dass die angebotenen Produkte und Dienstleistungen den Erwartungen der Kunden entsprechen oder diese übertreffen. Zweitens ermöglicht sie es Unternehmen, Ressourcen effizienter einzusetzen, indem Fehler frühzeitig erkannt und behoben werden, bevor sie zu kostspieligen Rückrufen, Reklamationen oder Umarbeitungen führen. Drittens fördert die Qualitätskontrolle kontinuierliche Verbesserungen, die letztendlich zu einer Steigerung der Wettbewerbsfähigkeit und Rentabilität des Unternehmens führen können.[13]

Insgesamt bietet die Qualitätskontrolle zahlreiche Vorteile für Unternehmen und Kunden gleichermaßen. Durch die konsequente Umsetzung von Qualitätskontrollmaßnahmen kann ein Unternehmen nicht nur seinen Ruf und seine Marktposition verbessern, sondern auch eine nachhaltige Wertschöpfung für seine Stakeholder sicherstellen.

Ein weiterer wichtiger Aspekt der Qualitätskontrolle ist die kontinuierliche Überwachung und Anpassung der Kontrollverfahren und -techniken, um auf Veränderungen in der Produktionsumgebung, den Marktanforderungen oder den technologischen Fortschritten zu reagieren. Dies kann durch regelmäßige Überprüfungen, Audits und Benchmarks erreicht werden, die dazu beitragen, die Effektivität und Effizienz der Qualitätskontrollmaßnahmen zu bewerten und gegebenenfalls Verbesserungen vorzunehmen.[21]

Zusammenfassend ist die Qualitätskontrolle ein zentraler Bestandteil des Qualitätsmanagements und spielt eine entscheidende Rolle bei der Sicherstellung einer hohen Produkt- und Prozessqualität. Durch die Implementierung von Qualitätskontrollmaßnahmen, die Berücksichtigung von branchenspezifischen Standards und die kontinuierliche Verbesserung der Kontrollverfahren können Unternehmen ihre Wettbewerbsfähigkeit stärken, die Kundenzufriedenheit erhöhen und eine nachhaltige Wertschöpfung erzielen.[2]

2.2 Künstliche Intelligenz und Einordung von Deep Learning

In diesem Kapitel wird das Konzept des Deep Learning eingeführt und in den Kontext der Künstlichen Intelligenz (KI) und des Machine Learning eingeordnet. Ziel ist es, ein grundlegendes Verständnis dieser Technologien zu vermitteln und deren Relevanz für die automatisierte Qualitätskontrolle in der Fertigung zu beleuchten.

Zunächst wird ein Überblick über das Feld der Künstlichen Intelligenz gegeben. Die Künstliche Intelligenz ist ein weites Feld, das die Entwicklung von Algorithmen und Technologien umfasst, die es Maschinen ermöglichen, menschenähnliche kognitive Funktionen zu übernehmen. Dabei geht es nicht nur um den Einsatz von Maschinen zur Automatisierung von Aufgaben, sondern auch um die Fähigkeit, aus Erfahrungen zu lernen und Probleme zu lösen, die bisher als spezifisch menschliche Domänen betrachtet wurden.[22]

Anschließend wird die Betrachtung auf das Gebiet des Machine Learning verengt, einem Teilbereich der Künstlichen Intelligenz, der sich auf die Entwicklung von Algorithmen konzentriert, die es Maschinen ermöglichen, aus Daten zu lernen. Im speziellen Fokus stehen dabei Neuronale Netze, eine spezielle Art von Machine Learning Modellen, die auf der Struktur des menschlichen Gehirns basieren und es Maschinen ermöglichen, komplexe Muster in Daten zu erkennen und zu lernen.[23]

Schließlich wird der Fokus auf Deep Learning gelegt, einen Unterbereich der Neuronalen Netze, der sich durch den Einsatz von mehrschichtigen, tiefen neuronalen Netzwerkstrukturen auszeichnet. Deep Learning hat sich als besonders effektiv bei der Erkennung von Mustern in großen und komplexen Datenmengen erwiesen, wie sie beispielsweise in Bildern oder Audiodaten vorkommen. In der Fertigung kann Deep Learning zur automatisierten Qualitätskontrolle von Produkten genutzt werden, indem es Muster in Bildern erkennt, die auf Produktfehler oder Qualitätsmängel hinweisen könnten.[24]

2.2.1 Künstliche Intelligenz

Künstliche Intelligenz (KI) ist ein Begriff, der im akademischen und geschäftlichen Kontext weit verbreitet ist. Trotz seiner tiefen Verankerung fehlt ihm eine universell akzeptierte und präzise Definition, was hauptsächlich auf die interdisziplinäre und komplexe Beschaffenheit des entsprechenden Fachgebiets zurückzuführen ist.[22]

John McCarthy, ein US-amerikanischer Informatiker, prägte ursprünglich den Begriff "Künstliche Intelligenz" bei der Durchführung der ersten akademischen Konferenz zu diesem Thema im Jahr 1956 am Dartmouth College in New Hampshire, die als Wiege der KI angesehen wird. McCarthy charakterisierte KI als "Maschinen, die menschliche Intelligenz nachahmen". In diesem Kontext wird KI für diese Arbeit als ein Feld innerhalb der Informatik definiert, das sich auf die Erstellung von Systemen konzentriert, die in der Lage sind, kognitive Fähigkeiten auszuführen, die normalerweise mit dem menschlichen Verstand in Verbindung gebracht werden. Dies umfasst grundlegende Fähigkeiten wie Lernen, Verständnis natürlicher Sprache, Wahrnehmung oder logisches Denken. Je nach Umfang, in dem KI-Systeme diese Fähigkeiten umsetzen können, wird zwischen "schwacher" und "starker" KI unterschieden. Schwache KI-Systeme nutzen spezifische Aspekte menschlicher Kognition und konzentrieren sich auf ein spezielles Problem, das sie zu lösen trainiert wurden. Starke KI-Systeme hingegen können das volle Spektrum kognitiver Funktionen anwenden, ähnlich wie Menschen, um jede Aufgabe zu bewältigen, die ihnen gestellt wird. Da alle gegenwärtigen KI-Anwendungen auf bestimmte Probleme abzielen und starke KI noch nicht erreicht ist, bezieht sich der Begriff "KI" in diesem Kontext ausschließlich auf die schwache Form dieser Technologie.[25]

Seit der Entstehung der KI im Jahr 1956 hat die Technologie verschiedene Zyklen von Begeisterung und Hype durchlaufen, die bisher immer von sogenannten "KI-Wintern" gefolgt wurden - Zeiträume, die durch schwindendes Interesse, zurückgehende Forschung

und sinkende Finanzierung gekennzeichnet waren. Die großen KI-Winter der 1970er und 1990er Jahre resultierten vorwiegend aus der Unfähigkeit der KI, den damit verbundenen Hype zu erfüllen, was hauptsächlich auf technologische Beschränkungen in diesen Zeiträumen zurückzuführen ist. Aktuelle KI-Aktivitäten deuten jedoch darauf hin, dass sich die KI seit 2013 in einem "KI-Frühling" befindet, in dem die Technologie fest im Alltag verankert ist.[26]

Laut Professor Christian Bauckhage ist die aktuelle Beschleunigung der KI-Entwicklung auf die Konvergenz von drei technologischen Entwicklungen zurückzuführen: gesteigerte Rechenkraft, Zugriff auf umfangreiche Datensätze und Fortschritte in den Algorithmen. KI, eine Technologie, die intensiv auf Computerleistung angewiesen ist, hat vom exponentiellen Wachstum der Chipeffizienz profitiert. Dieses Wachstum wurde durch Moores Gesetz sowie den Einsatz von Grafikprozessoren (GPUs) anstelle von Central Processing Units (CPUs) gefördert. Im Vergleich zu CPUs ermöglichen GPUs eine höhere Auslastung, wodurch die Trainingszeit für KI-Algorithmen erheblich reduziert wird. GPUs wurden ursprünglich entwickelt, um die anspruchsvolleren Rechenanforderungen für das Echtzeit-Rendering von Computerspielgrafiken zu bewältigen. Heute werden sie genutzt, um Hunderte von Aufgaben gleichzeitig zu bearbeiten, was die effektive Ausführung von KI-Anwendungen ermöglicht.[27]

Ein weiterer entscheidender Faktor für den Erfolg der KI ist die Fähigkeit zur Verarbeitung großer Datenmengen in Bezug auf Volumen, Geschwindigkeit und Vielfalt. Traditionelle Computertechniken waren nicht in der Lage, solche großen und oft unstrukturierten Dateneingaben zu verarbeiten. Fortschritte in Deep Learning (DL-Algorithmen) und künstlichen neuronalen Netzwerken (KNN) haben es jedoch ermöglicht, dass KI-Systeme Muster und Zusammenhänge auch in sehr komplexen Datensätzen erkennen können.
Künstliche Intelligenz (KI) repräsentiert nicht eine einzige Technologie, sondern dient, wie in Abbildung 1 abgebildet, als Sammelbegriff für eine Reihe von Technologien, die oft miteinander verknüpft und aufeinander aufbauend sind.[24]

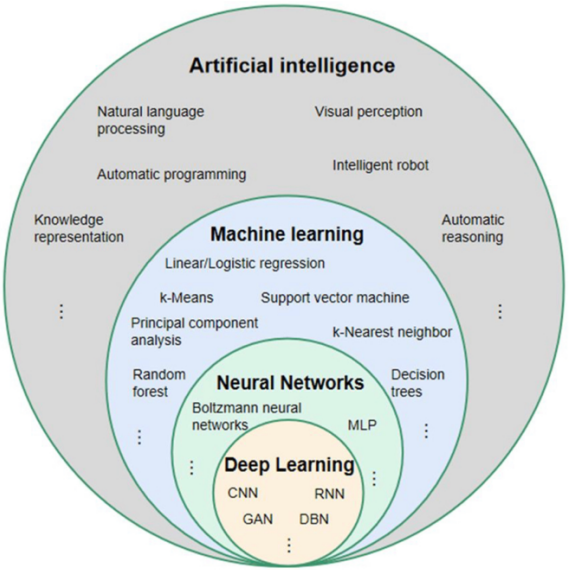


Abbildung 1: Verhältnis zwischen künstlicher Intelligenz, maschinellem Lernen, neuronalen Netzen und Deep Learning[28]

Einige der bedeutsamen Technologien innerhalb des KI-Bereichs umfassen spezifische kognitive Fähigkeiten wie Lernen, Verstehen natürlicher Sprache oder Wahrnehmung umfassen. Einige Zweige fokussieren sich auf die Verarbeitung externer Informationen, wie Natural Language Processing (NLP) und Computer Vision; einige nutzen Informationen, um zu agieren, wie die Robotik, und andere verwenden Informationen, um daraus zu lernen, wie das maschinelle Lernen (ML).[29] In vielen KI-Anwendungen verstärken und ergänzen sich diese Technologiezweige jedoch oft gegenseitig. Beispielsweise bilden ML-Modelle den technologischen Kern vieler fortgeschrittener NLP-, Computer-Vision- und Robotik-Anwendungen.[23]

2.2.2 Machine Learning

Machine Learning ist ein Teilgebiet der KI, das sich mit der Fähigkeit von Computersystemen beschäftigt, aus Daten zu lernen, ohne explizit programmiert zu werden. Es gibt verschiedene Methoden im Machine Learning, wie beispielsweise Supervised Learning, Unsupervised Learning und Reinforcement Learning, von denen Deep Learning eine ist. Ein Teilbereich der KI, das maschinelle Lernen (ML), gewährt IT-Systemen die Fähigkeit, Muster und Regelmäßigkeiten zu erkennen und Lösungen auf der Grundlage vorhandener Datensätze und Algorithmen zu entwickeln. Durch das Training mit Daten generiert ML künstliches Wissen, das verallgemeinert und für neue Problemlösungen oder für die Analyse unbekannter Daten eingesetzt werden kann.

Zwei Hauptmethoden des maschinellen Lernens, Supervised Learning und Unsupervised Learning, dominieren das Feld, obwohl es eine Reihe weiterer Methoden gibt.
Supervised Learning trainiert Algorithmen anhand von gegebenen Paaren von Eingaben und Ausgaben, sogenannten Labels. Dies ermöglicht es dem Lernalgorithmus, seine tatsächliche Ausgabe mit den korrekten Ausgaben zu vergleichen, Fehler zu identifizieren und daraus zu lernen, um das Modell entsprechend anzupassen. Supervised Learning ist besonders wertvoll, wenn aus historischen Daten wahrscheinliche zukünftige Ereignisse abgeleitet werden können.

Im Gegensatz dazu wird Unsupervised Learning für Daten eingesetzt, für die keine historischen Labels vorhanden sind. Der Algorithmus muss hierbei unbekannte Strukturen und Beziehungen zwischen den Daten eigenständig identifizieren. Unüberwachte Lernmethoden sind besonders nützlich in Bereichen wie Textanalyse, Sprachanalyse und automatischer Bilderkennung und -verarbeitung.[30]

Semisupervised Learning kombiniert Aspekte von Supervised und Unsupervised Learning, indem sowohl gelabelte als auch ungelabelte Daten für das Training verwendet werden. Diese Methode ist besonders vorteilhaft, wenn die Kosten für die Kennzeichnung aller Daten unverhältnismäßig hoch wären.[31]

Reinforcement Learning, eine weitere Methode des maschinellen Lernens, wird hauptsächlich in Verbindung mit Robotik, Computerspielen und Navigation eingesetzt. In diesem Lernprozess erkennt der Algorithmus durch Versuch und Irrtum, welche Handlungen die höchste Belohnung erzielen.[32]

Die vorliegende Arbeit fokussiert sich auf die Anwendung der unüberwachten Lernmethode des maschinellen Lernens im Kontext des Einsatzes von Deep Learning für die Qualitätskontrolle in der Fertigung. In diesem spezifischen Anwendungsbereich wird die Unsupervised Learning Methode zur Identifikation von Mustern und Anomalien in Fertigungsdaten eingesetzt. Diese können dann dazu verwendet werden, die Qualität der produzierten Güter zu überwachen und zu verbessern.

Ein Aspekt der Unsupervised Learning Methode, der in diesem Kontext besonders relevant ist, ist die Clusteranalyse. Hierbei werden Datenpunkte, die ähnliche Merkmale aufweisen, in Gruppen oder "Cluster" zusammengefasst. Dies ermöglicht es, Muster und Zusammenhänge in den Daten zu erkennen, die sonst möglicherweise übersehen würden. In

der Fertigung kann diese Methode dazu verwendet werden, um beispielsweise Produktionsfehler zu identifizieren oder die Effizienz des Produktionsprozesses zu optimieren.
Ein weiterer Vorteil des Einsatzes von Unsupervised Learning in der Fertigung liegt in seiner Fähigkeit, mit großen Datenmengen umzugehen. Moderne Fertigungstechnologien generieren eine enorme Menge an Daten, die analysiert und interpretiert werden müssen, um wertvolle Einblicke zu gewinnen. Unsupervised Learning Algorithmen sind in der Lage, diese Daten zu verarbeiten und Muster zu identifizieren, die für die Qualitätskontrolle von Nutzen sein können.

Schließlich sind die Ergebnisse von Unsupervised Learning in der Lage, die Entscheidungsfindung in der Fertigung zu unterstützen. Durch die Identifikation von Mustern und Anomalien in den Daten können diese Algorithmen den Fertigungsverantwortlichen wertvolle Informationen liefern, die dazu beitragen können, die Qualität der produzierten Güter zu verbessern und die Effizienz des Produktionsprozesses zu steigern.[33]

2.2.3 Deep Learning

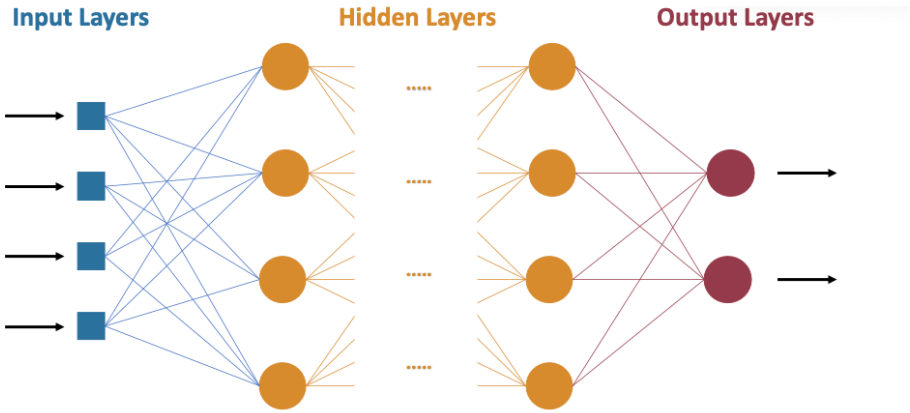
Deep Learning ist eine Unterform von Machine Learning, die auf neuronalen Netzen basiert und Teil der Künstlichen Intelligenz (KI) ist. Es handelt sich dabei um eine Methode, bei der komplexe Muster und Beziehungen in großen Datenmengen automatisch erlernt werden, ohne dass ein Programmierer die Regeln manuell definieren muss.

Insbesondere Deep Learning (DL) hat in den letzten Jahren maßgeblich zur Entstehung vieler KI-Anwendungen beigetragen. DL nutzt tiefe künstliche neuronale Netzwerke (KNN), die in ihrer Funktionsweise dem menschlichen Gehirn ähneln. Wie sich die Verbindungen zwischen den Neuronen im Gehirn durch Erfahrungen anpassen und verbessern, so werden auch die Verbindungen innerhalb der KNN gestärkt oder geschwächt, wenn neue Daten vom Netzwerk empfangen werden. Durch die Verstärkung von Verbindungen, die positive Ergebnisse erzielen, und die Schwächung von Verbindungen, die schlechte Ergebnisse liefern, verbessert sich die Qualität der Ausgabe schrittweise mit jedem Lernzyklus.[24]

KNN bestehen aus vernetzten Neuronen, die in einer Reihe von Schichten organisiert sind. Diese sind in der Lage, alle Arten von Eingabedaten wie Pixel-, Audio- und Textdaten zu verarbeiten. In einem bekannten Anwendungsfall der Computer Vision - der Gesichtserkennung - liefern die Eingabeschichten die Pixeldaten an das Netzwerk.

Verborgene Schichten zerlegen dann die visuellen Komponenten, um die charakteristischen Merkmale eines bestimmten Gesichts zu identifizieren. In der Ausgabeschicht prognostiziert das KNN, ob das Gesicht Person X, Y oder Z zugeordnet ist. Wie bereits erwähnt, steigt die Genauigkeit der Vorhersage mit jeder Lerniteration. Ein tiefes KNN mit vielen versteckten Schichten kann komplexere Probleme lösen, da es immer detailliertere Merkmale der Eingabedaten identifizieren kann.

Künstliche Neuronale Netze (KNN) sind eine Gruppe von maschinellen Lernmodellen, die auf der Struktur des Gehirns basieren. Wissen wird durch die Verbindungen von Neuronen repräsentiert. Die Wurzeln dieser Konzepte reichen bis in das Jahr 1943 zurück, als Warren McCulloch und Walter Pitts ein rechnerisches Modell beschrieben, das prinzipiell jede logische oder mathematische Funktion berechnen kann. Sie stellten später fest, dass solch ein Netzwerk auch für die Mustererkennung eingesetzt werden kann. KNN bestehen aus Schichten von Neuronen. Eine schematische Darstellung eines künstlichen neuronalen Netzwerks ist in Abbildung 2 zu sehen.[34]



Problemstellung	Input	Hidden Layers	Output
Gesichtserkennung	Bilder	Person? Gesicht? Geschlecht? Alter? Augenfarbe?	Ist es Person X ? (%)
Online-Anzeigenschaltung	Browsing-Daten in sozialen Medien	Demographie? Interessen? Browsing Daten Meta Historie	Wird Person X klicken? (%)

Abbildung 2: Aufbau und Funktionalität Künstlicher Neuronaler Netze [Eigene Darstellung in Anlehnung an [35]]

Das von McCulloch und Pitts beschriebene Rechenmodell war jedoch in der Praxis nicht trainierbar. Dies änderte sich, als der Psychologe Donald Hebb 1949 seine Lernregel formulierte. Hebb's Lernregel beschreibt die Veränderung zweier gemeinsam aktiver Neuronen als Gewichtsveränderung. Mathematisch wird dies als $\Delta w_{ij} = \eta a_i o_j \cdot \Delta w_{ij}$ definiert, wobei Δw_{ij} die Gewichtsveränderung von Neuron i zu Neuron j darstellt, η die Lernrate (ein konstanter Faktor), a_i die Aktivierung von Neuron i und o_j die Ausgabe

von Neuron *j* ist. In anderen Worten, je öfter zwei Neuronen gemeinsam aktiv sind, desto stärker reagieren sie aufeinander. Hebb's Lernregel in ihrer allgemeinen Form beschreibt fast alle Lernmethoden für neuronale Netze. Einfach gesagt bedeutet das Training eines künstlichen neuronalen Netzwerks, dass am Ausgang eines Neurons/einer Schicht ein Fehler berechnet wird, auf dessen Basis die Gewichte verändert werden, was in der Regel mit einer Gradientenmethode erfolgt.[36]

Paul Werbos entwickelte 1974 die Rückpropagation, ein Verfahren, das noch heute die Grundlage für Methoden zum Training künstlicher neuronaler Netze ist. Der Rückpropagationsalgorithmus arbeitet in drei Phasen. In der ersten Phase werden die Daten durch das Netzwerk geleitet und ihre Ausgabe berechnet. In der zweiten Phase wird die Ausgabe mit dem gewünschten Wert verglichen und die Differenz, der Fehler, berechnet. In der dritten Phase wird der Fehler durch das Netzwerk zurückgeleitet, von der Ausgabeschicht zur Eingabeschicht, wobei die Gewichte entsprechend ihrem Einfluss auf den Fehler verändert werden.[37]

Deep Learning bezieht sich hauptsächlich darauf, dass neuronale Netze immer tiefer werden, d.h., sie haben viele Schichten. Aufgrund dieser erhöhten Anzahl an Schichten lernen die Netze komplexere Merkmale. Die neuronalen Netzwerke von heute verwenden Millionen von Trainingsdaten, die größtenteils auf GPUs (Grafikkarten) oder ganzen Clustern (auch GPU-Cluster) berechnet werden.[24]

Im folgenden Abschnitt wird die genaue Funktionsweise eines künstlichen neuronalen Netzwerks beschrieben und verschiedene Aspekte werden detaillierter erläutert.

2.2.4 Funktionsweise von Künstlichen Neuronalen Netzen

Neuronale Netze sind die Grundlage von Deep Learning und stellen ein Modell dar, das anhand einer großen Menge an Beispieldaten eine bestimmte Aufgabe lernen kann. Sie sind inspiriert von den Funktionsweisen des menschlichen Gehirns und bestehen aus Schichten von künstlichen Neuronen, die Informationen durch Forward-Propagation und Fehlerkorrektur durch Backpropagation verarbeiten.

In diesem Abschnitt wird erläutert, wie künstliche neuronale Netzwerke funktionieren. Die elementarste Komponente in einem KNN ist ein Neuron. Für ein besseres Verständnis wird daher das kleinste mögliche Netzwerk, bestehend aus einem einzelnen Neuron, betrachtet.

Das Neuron hat die Eingänge $x_i \, \epsilon \, X$, wobei $i \, \epsilon \, 1 \ldots, N$ und N die Anzahl der Eingangswerte ist. Jeder Eingang wird mit w_i gewichtet. Entsprechend berechnet ein Neuron die folgende Funktion:

$$f_n(x) = \sum_{i=1}^{N} w_i x_i \tag{1}$$

Die resultierende Ausgabe eines Neurons hängt von der Aktivierungsfunktion f_A ab. Die Aktivierungsfunktion bestimmt, wie ein Neuron auf einen Eingang reagiert. Für eine weitere Untersuchung sei die Aktivierungsfunktion $f_A(x) = x$. Daher ist die Ausgabe des Neurons $y = f_A\big(f_n(x)\big) = f_A(\sum_i w_i x_i) = \sum_i w_i x_i)$. Als Ergebnis berechnet das neuronale Netzwerk die Funktion $f_{ann}(x) = f_A(f_n(x)) = y$.[38]

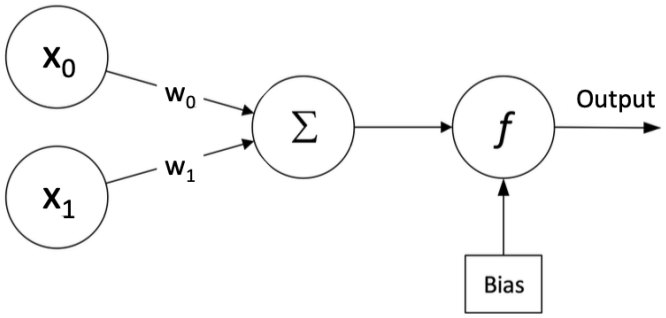


Abbildung 3: Beispiel eines einzelnen Neurons [Eigene Darstellung in Anlehnung an [39]]

Betrachtet man das Beispiel eines konkreten Netzwerks mit einem einzelnen Neuron, wie in Abbildung 3 dargestellt, das zwei Eingänge und einen Ausgabewert hat. Die Gewichte zu Beginn sind $w_0 = 0{,}5$ und $w_1 = 0{,}7$. Das Lernen in einem Netzwerk mit einem Neuron läuft wie folgt ab: Angenommen, die Eingangswerte sind $x_0 = 2$; $x_1 = 3$, die Gewichte $w_0 = 0{,}5$; $w_1 = 0{,}7$ und der erwartete Ausgabewert ist $t = 0$. Dies führt zu der Gleichung $y(x, w) = 0{,}5 \cdot 2 + 0{,}7 \cdot 3 = 3{,}1 \neq 0$.

Der Verlust (Fehler) wird nun auf der Grundlage einer Fehlfunktion berechnet. Im Folgenden wird der Mean Squared Error (MSE) verwendet, was zu einem Verlust von $MSE = (t - y)^2 = (3{,}1 - 0)^2 = 9{,}61$ führt. Das Ziel des Trainingsprozesses ist es, die Fehlfunktion zu minimieren, weshalb die Fehlfunktion oft als Zielfunktion bezeichnet wird. Dies wird durch Anpassung der Gewichte erreicht.

Mit Hilfe des Backpropagation-Algorithmus ergibt sich für die Gewichtsänderung die folgende Formel $\Delta w_{ij} = \eta \delta_j x_{ij}$, wobei η eine konstante Lernrate und $\delta_j = f'_A(x_j)(y_j -$

t_j) für den Fall ist, dass j ein Ausgangsneuron ist, wie im Beispiel gegeben. Dies führt zu $\delta_0 = 1 \cdot (0 - 3{,}1) = -3{,}1$ und mit einer Lernrate von $\eta = 0{,}1$ ist die Gewichtsänderung $\Delta w_{00} = 0{,}1 \cdot -3{,}1 \cdot 2 = -0{,}62$ und $\Delta w_{10} = 0{,}1 \cdot -3{,}1 \cdot 3 = -0{,}93$. Die neuen Gewichte werden mit $w_{ij}^{neu} = w_{ij}^{alt} + \Delta w_{ij}$ berechnet und ergeben folglich $w_{00}^{neu} = w_{00}^{alt} + \Delta w_{00} = 0{,}5 - 0{,}62 = -0{,}12$ und $w_{10}^{neu} = w_{10}^{alt} + \Delta w_{10} = 0{,}7 - 0{,}93 = -0{,}23$.

Das neuronale Netzwerk liefert nun das folgende Ergebnis $y(x, w) = -0{,}12 \cdot 2 - 0{,}23 \cdot 3 = -0{,}93 \neq 0$. Das Ergebnis ist immer noch nicht korrekt, aber der Verlust wurde erheblich auf $MSE = (t - y)^2 = (0{,}93 - 0)^2 = 0{,}8649$ reduziert. Ein solcher Trainingsschritt, der die Verbreitung der Daten, die Rückverbreitung des Verlusts und die Änderung der Gewichte umfasst, wird als Epoche bezeichnet. Der gesamte Trainingsprozess besteht aus mehreren Epochen. In der Praxis wird entweder eine festgelegte Anzahl von Epochen trainiert, oder das Training wird fortgesetzt, bis sich der Verlust nicht mehr signifikant verbessert. Alternativ kann die Genauigkeit als Benchmark für die Bewertung des Netzwerks herangezogen werden. Eine weitere Möglichkeit ist die Verwendung eines zweiten Datensatzes (Validierung) und dessen Auswertung hinsichtlich Verlust oder Genauigkeit.[39]

Wenn es nicht möglich ist, die gesamte Datenmenge im Speicher zu halten, werden die Daten in Chargen aufgeteilt und dann verarbeitet. Während des Trainings werden die Daten dann für jede Epoche in Chargen verarbeitet und die Gewichtsänderungen entsprechend vorgenommen.[38]

2.3 Computer Vision in der Fertigung

Das Konzept der Computer Vision wurde erstmals in den 1970er Jahren vorgestellt. Es besteht breiter Konsens, dass Larry Roberts als der Vater der Computer Vision gilt. Seitdem haben viele Forscher seine Arbeit fortgesetzt. Inzwischen jedoch hat die Welt einen gewaltigen Technologiesprung erlebt, der Computer Vision in einem viel größeren Maße nutzt und es auf die Prioritätenliste zahlreicher Branchen gesetzt hat.[40]

Computer Vision ist ein Bereich der Künstlichen Intelligenz (KI), der die Herausforderung angeht, Computern das Sehen und Interpretieren der visuellen Welt so beizubringen, wie Menschen es tun. Dies erreichen Computer durch Training mit Fotos von Kameras und Videos, indem sie Deep-Learning-Modelle nutzen. Sie sind dann in der Lage, Objekte genau zu identifizieren und auf sie in einer Weise zu reagieren, die der Reaktion

von Menschen auf diese Objekte ähnlich ist. Computer Vision findet heute Anwendung in einer Vielzahl von Branchen, wie zum Beispiel beim Testen von autonomen Fahrzeugen, in der Telekommunikation, in der Landwirtschaft zur Überwachung von Nutztieren und der Gesundheit von Pflanzen, im Gesundheitswesen für tägliche Diagnosen und viele weitere.[41]

Computer Vision ist ein neues Gebiet in der Informatik, das sich mit der Frage beschäftigt, wie ein Computer eine umfangreiche Kenntnis und Wahrnehmung von Informationen erlangen kann, die in Bildern und Videos vorhanden sind. Das Ziel besteht nicht nur darin, dem Computer das "Sehen" zu ermöglichen, sondern auch Aufgaben zu automatisieren, wie es das menschliche Sehvermögen tut. Von der Datenerfassung aus nutzt die Computer Vision bestimmte Verfahren, um maschinelles Lernen, Bildverarbeitung, Computergrafik und Mustererkennung zu kombinieren, um eine benötigte Interpretation zu extrahieren. Anwendungen wie Sicherheitssysteme an öffentlichen Orten wie Flughäfen, bei denen der Computer ein Passagierbild erfasst und anhand dessen Reiseinformationen öffnet, sind eine reale Anwendung der Computer Vision unter vielen anderen. Aufgrund des schnellen Wachstums und der Verbreitung der Computer Vision besteht eine zunehmende Nachfrage nach automatisierten Methoden und Technologien zur Lösung realer Probleme in vielen Bereichen.[42]

2.3.1 Vorteile und Herausforderungen

Die zunehmende Automatisierung in der modernen Fertigungsindustrie erfordert eine parallele Automatisierung der Qualitätskontrolle. Es ist beachtenswert, dass in zahlreichen Fertigungsunternehmen die Qualitätskontrolle nach wie vor manuell durchgeführt wird. Trotz der außergewöhnlichen Fähigkeiten des Menschen, unterschiedliche Objekte visuell zu untersuchen, kann die Monotonie und Ermüdung, die durch wiederholte Aufgaben entsteht, zur Fehleranfälligkeit führen. Daher tendiert der gegenwärtige Trend dazu, eine menschenähnliche oder sogar höhere Genauigkeit in der Qualitätskontrolle durch den Einsatz von Computer Vision (CV) Techniken, genauer gesagt, durch maschinelles Sehen basierend auf neuronalen Netzen, zu erreichen.

Es existieren Produktionsunregelmäßigkeiten, die tolerierbar sind, und andere, die unbedingt identifiziert werden müssen. Da die Wahrnehmung der Kunden das äußere Erscheinungsbild eines Produkts unbewusst mit dessen Qualität verknüpft, legen Hersteller verstärkten Wert auf das ästhetische Erscheinungsbild ihrer Produkte.[41]

Automatisierte, auf Computer Vision (CV) basierende Systeme leisten einen signifikanten Beitrag zur Produktinspektion, indem sie Defekte, Fehlfunktionen oder Verunreinigungen effizient und ohne menschliches Zutun identifizieren. Im Vergleich zur herkömmlichen Qualitätskontrolle bieten automatisierte CV-basierte Systeme eine Reihe von Vorteilen:

Vorteile von Computer Vision in der Qualitätskontrolle:

1. **Erhöhte Präzision**: Eine der Hauptstärken der Anwendung von Computer Vision in der Qualitätskontrolle ist die hohe Genauigkeit, die es bietet. Mit der Fähigkeit, feine Details zu erfassen, übertrifft Computer Vision oft menschliche Inspektionen in Bezug auf Präzision und Konsistenz. Durch die Identifizierung von Fehlern oder Anomalien, die für das menschliche Auge möglicherweise unsichtbar sind, trägt Computer Vision dazu bei, das Endprodukt zu verbessern.

2. **Höhere Geschwindigkeit**: Computer Vision-Systeme können eine große Anzahl von Produkten in einem Bruchteil der Zeit bewerten, die ein menschlicher Inspektor benötigen würde. Dies führt zu einer erheblichen Steigerung der Produktionseffizienz und -geschwindigkeit.

3. **Konsistenz**: Im Gegensatz zu menschlichen Inspektoren, die möglicherweise Ermüdungserscheinungen aufweisen oder deren Leistung schwanken kann, liefern Computer Vision-Systeme konstante Ergebnisse. Sie sind in der Lage, kontinuierlich zu arbeiten und dabei eine gleichbleibende Leistung zu erbringen.

4. **Reduzierte Kosten**: Durch die Automatisierung des Qualitätskontrollprozesses können Unternehmen erhebliche Einsparungen erzielen. Es werden weniger menschliche Inspektoren benötigt und die Kosten für Ausfallzeiten durch fehlerhafte Produkte können reduziert werden.

5. **Echtzeitüberwachung und -feedback**: Computer Vision ermöglicht die Echtzeitüberwachung von Fertigungsprozessen, was eine sofortige Reaktion auf Probleme ermöglicht. Dies reduziert die Ausfallzeiten und verbessert die allgemeine Produktionsleistung.

6. **Verarbeitung von Big Data**: Moderne Fertigungsprozesse erzeugen eine enorme Menge an Daten. Computer Vision-Systeme können diese Daten effizient

verarbeiten und verwertbare Erkenntnisse liefern, die zur Verbesserung der Produktqualität und zur Optimierung von Prozessen genutzt werden können.

7. **Sicherheit**: Computer Vision kann dabei helfen, gefährliche oder gesundheitsschädliche Aufgaben zu automatisieren, was zu einer sichereren Arbeitsumgebung führt.

8. **Datengetriebene Entscheidungen**: Computer Vision Systeme generieren eine Fülle von Daten, die analysiert und genutzt werden können, um fundierte Entscheidungen zu treffen. Diese datengesteuerten Einblicke können zur Optimierung von Prozessen, zur Verbesserung der Produktqualität und zur Steigerung der betrieblichen Effizienz beitragen.

9. **Flexibilität**: Moderne Computer Vision Systeme sind flexibel und anpassungsfähig, was es ermöglicht, sie für eine Vielzahl von Anwendungen und in verschiedenen Produktionsumgebungen einzusetzen.[43]

Angesichts dieser Vorteile hat die CV-Technologie in den letzten Jahren eine Kommerzialisierung erfahren und findet in einer breiten Palette von Anwendungen Einsatz. Nachweisbare Konzepte, die vor einigen Jahren entwickelt wurden, gehen nun in die Produktion über. Dies führt zu einer neuen Reihe von Herausforderungen, die Möglichkeiten für Start-ups bieten. Die schnelle Entwicklung der Technologie, präzise Ergebnisse, sinkende Hardwarekosten und einfache Konnektivität sind nur einige der Faktoren, die zur weltweiten Akzeptanz von CV beitragen.

Herausforderungen bei der Implementierung von Computer Vision in der Qualitätskontrolle:

1. **Hohe Anfangsinvestitionen**: Die Implementierung von Computer Vision-Systemen erfordert eine erhebliche Anfangsinvestition. Es sind Kosten für Hardware, Software und möglicherweise auch für die Anpassung der Produktionslinie zu berücksichtigen.

2. **Komplexität der Einrichtung und Wartung**: Die Einrichtung und Kalibrierung eines Computer Vision-Systems kann komplex sein und erfordert spezielles Fachwissen. Darüber hinaus kann auch die laufende Wartung und Aktualisierung des Systems eine Herausforderung darstellen.

3. **Beschränkungen der Technologie**: Obwohl Computer Vision-Systeme äußerst präzise sind, haben sie ihre Grenzen. Sie können Schwierigkeiten haben, bestimmte Arten von Fehlern zu erkennen, insbesondere solche, die eine menschliche Interpretation oder ein Urteil erfordern. Darüber hinaus können sie möglicherweise nicht mit unerwarteten Änderungen in der Produktionsumgebung umgehen.

4. **Datenschutz und Ethik**: Die Verwendung von Computer Vision in der Fertigung kann datenschutzrechtliche und ethische Bedenken aufwerfen. Es ist wichtig, dass Unternehmen bei der Implementierung von Computer Vision-Systemen die Privatsphäre ihrer Mitarbeiter respektieren und ethische Richtlinien einhalten.

5. **Bedarf an Schulung und Kompetenzentwicklung**: Die Einführung von Computer Vision kann die Notwendigkeit einer Schulung der Mitarbeiter mit sich bringen, sowohl in Bezug auf die Bedienung der Systeme als auch hinsichtlich der Interpretation und Anwendung der von den Systemen generierten Daten und Erkenntnisse.

6. **Integration in bestehende Systeme**: Die Einführung von Computer Vision kann Herausforderungen in Bezug auf die Integration in bestehende Fertigungssysteme und -prozesse mit sich bringen. Dies kann zusätzliche Zeit und Ressourcen erfordern und erfordert eine sorgfältige Planung und Durchführung.

7. **Anpassung an Veränderungen**: Computer Vision Systeme können Schwierigkeiten haben, sich an Änderungen in den Produktionsprozessen oder an neue Produktlinien anzupassen. Dies kann dazu führen, dass regelmäßige Aktualisierungen und Anpassungen erforderlich sind.

8. **Fehlende Standards**: Es gibt noch keine allgemein anerkannten Standards für die Implementierung und Nutzung von Computer Vision in der Fertigung, was die Integration und Kompatibilität mit bestehenden Systemen erschweren kann.

9. **Abhängigkeit von der Qualität der Eingabedaten**: Die Leistung von Computer Vision Systemen hängt stark von der Qualität der Eingabedaten ab. Ungenauigkeiten, Verzerrungen oder Mängel in den Bild- oder Videodaten können zu Fehlinterpretationen und Fehlern führen. Daher ist es wichtig, hochwertige Kameras

und Sensoren zu verwenden und die richtigen Bedingungen für die Datenerfassung zu gewährleisten.

Trotz der Herausforderungen, die mit der Einführung von Computer Vision in der Qualitätskontrolle verbunden sind, sind die potenziellen Vorteile erheblich. Mit einer sorgfältigen Planung und Implementierung können Unternehmen diese Technologie nutzen, um ihre Fertigungsprozesse zu verbessern, die Produktqualität zu steigern und ihre Wettbewerbsfähigkeit zu erhöhen. Mit einer gut durchdachten Implementierungs-strategie kann Computer Vision ein leistungsfähiges Werkzeug für die Qualitätskontrolle in der Fertigung sein. Es ist jedoch wichtig, dass Unternehmen sich bewusst sind und proaktiv auf die Herausforderungen reagieren, die sich aus der Einführung dieser Technologie ergeben können.[44]

2.3.2 Anwendungsgebiete

Nachfolgend werden einige der bekanntesten Anwendungsgebiete von Computer Vision ausführlich erläutert. Dabei liegt der Fokus auf den breit gefächerten Einsatzbereichen der Technologie, die eine transformative Wirkung auf eine Vielzahl von Sektoren hat. Dies reicht von medizinischen Anwendungen bis hin zu robotischen Systemen, vom Einzelhandel bis zur Umweltüberwachung. Der Nutzen und die Auswirkungen dieser fortgeschrittenen Technologie haben sich als signifikant in unterschiedlichsten Domänen erwiesen. Im Folgenden wird eine eingehende Untersuchung dieser vielseitigen Anwendungsbereiche von Computer Vision vorgestellt.

Aus der Forschung geht hervor, dass Computer zunehmend besser darin werden, Bilder zu erkennen und Labels und Objekte in diesen Bildern zu identifizieren. Aus diesem Grund investieren einige der heutigen Technologieunternehmen, wie Google, Amazon, Microsoft und Facebook, Milliarden von Dollar in die Forschung und Entwicklung von Produkten, die Computer Vision nutzen. Hier sind einige Beispiele, wie es heute in der Industrie genutzt wird:

1. **Automatisierte Qualitätskontrolle in der Fertigung**: CV-Systeme können dazu verwendet werden, um Produkte auf Fehler zu überprüfen, indem sie digitale Bilder oder Videos analysieren. Diese Systeme können in der Lage sein, Anomalien zu identifizieren, die möglicherweise für das menschliche Auge zu subtil sind, und sie können eine konsistente Qualitätskontrolle gewährleisten, unabhängig von

menschlicher Ermüdung oder Ablenkung. Darüber hinaus können sie das Potenzial für menschliche Fehler reduzieren und die Effizienz und Präzision der Qualitätskontrollprozesse erhöhen.

2. **Autonome Fahrzeuge**: CV spielt eine entscheidende Rolle in der Technologie autonomer Fahrzeuge. Diese Fahrzeuge verwenden CV-Systeme, um ihre Umgebung zu erkennen und zu interpretieren, einschließlich anderer Fahrzeuge, Fußgänger, Straßenschilder und Ampeln. Diese Information wird dann genutzt, um Entscheidungen über Fahrmanöver zu treffen, um die Sicherheit und Effizienz der Fahrt zu gewährleisten.

3. **Gesundheitswesen**: CV wird zunehmend im Gesundheitswesen eingesetzt, beispielsweise in der medizinischen Bildgebung, wo es zur Erkennung und Diagnose von Krankheiten verwendet wird. CV kann dazu beitragen, Muster in medizinischen Bildern zu erkennen, die für das menschliche Auge schwer zu erkennen sind, und kann auch dazu beitragen, die Geschwindigkeit und Genauigkeit der Diagnose zu erhöhen. Darüber hinaus wird CV in chirurgischen Robotern eingesetzt, um Chirurgen bei komplexen Operationen zu unterstützen.

4. **Sicherheit und Überwachung**: CV wird häufig in Überwachungssystemen verwendet, um verdächtige Aktivitäten zu erkennen und Alarme auszulösen. Es kann auch zur Gesichtserkennung verwendet werden, um Personen zu identifizieren oder zu verifizieren. Darüber hinaus kann CV in Kombination mit maschinellem Lernen verwendet werden, um Muster im Verhalten zu erkennen, die auf potenzielle Sicherheitsbedrohungen hinweisen könnten.

5. **Landwirtschaft**: CV kann in der Landwirtschaft dazu beitragen, die Erträge zu optimieren und die Effizienz zu verbessern. Zum Beispiel kann es dazu verwendet werden, den Zustand von Pflanzen zu überwachen und Anzeichen von Krankheiten oder Schädlingen frühzeitig zu erkennen. Darüber hinaus können CV-basierte Drohnen genutzt werden, um großflächige Landwirtschaftsbetriebe zu überwachen und detaillierte Analysen über den Zustand der Ernte, den Feuchtigkeitsgehalt des Bodens und andere wichtige Parameter zu liefern.

6. **Einzelhandel**: Im Einzelhandel werden CV-Systeme eingesetzt, um das Einkaufserlebnis zu personalisieren und zu optimieren. Sie können beispielsweise zur

Erkennung von Kundenbewegungen innerhalb des Geschäfts verwendet werden, um Einblicke in das Kundenverhalten zu gewinnen und Ladenlayouts zu optimieren. CV kann auch zur Inventarverwaltung und zur Erkennung von Ladendiebstahl eingesetzt werden.

7. **Augmented Reality (AR) und Virtual Reality (VR)**: CV ist ein grundlegender Baustein von AR- und VR-Technologien. Es ermöglicht die Erkennung und das Tracking von Objekten in Echtzeit, was für die Schaffung immersiver und interaktiver virtueller Umgebungen entscheidend ist.

8. **Umweltüberwachung**: CV-Technologien können dazu beitragen, Umweltveränderungen und -bedrohungen zu überwachen und zu quantifizieren. Beispielsweise können Satellitenbilder analysiert werden, um die Ausbreitung von Waldbränden, das Schmelzen von Gletschern oder die Veränderung von Landnutzungsmustern zu verfolgen.

9. **Soziale Robotik**: CV ist ein wichtiger Bestandteil sozialer Roboter, die dazu entwickelt wurden, mit Menschen zu interagieren. Diese Roboter nutzen CV, um Gesichtsausdrücke, Gesten und Körperhaltung zu erkennen, was ihnen hilft, menschliches Verhalten zu interpretieren und angemessen darauf zu reagieren.

Insgesamt zeigt die Vielfalt dieser Anwendungen die transformative Wirkung der Computer Vision in verschiedenen Branchen und Bereichen. Sie bietet bedeutende Vorteile in Bezug auf Effizienz, Genauigkeit und Automatisierung. Gleichzeitig stellt sie jedoch auch Herausforderungen in Bezug auf Datenschutz und ethische Bedenken dar, insbesondere in Bezug auf Überwachung und Bias in Algorithmen, die sorgfältige Überlegungen und Regulierungen erfordern.[45][46]

3 Methodik

In diesem Kapitel wird die Methodik vorgestellt, die zur Beantwortung der Forschungsfrage und zum Erreichen der Zielsetzung der Bachelorarbeit angewendet wird. Im Mittelpunkt steht die Entwicklung eines Deep Learning-Modells zur Erkennung von Mustern in Bilddaten, um eine automatisierte Qualitätskontrolle in der Fertigung zu ermöglichen. Dazu wird zunächst ein passendes Fallbeispiel identifiziert und öffentlich zugängliche Bilddaten aus dem Internet herangezogen.

Der Ablauf dieses Kapitels gliedert sich wie folgt: Zunächst wird die Zielsetzung des Beispiels (Abschnitt 3.1) erläutert. Anschließend erfolgt die Auswahl der Datensätze (Abschnitt 3.2).

Im Abschnitt 3.3 geht es um den Import der benötigten Bibliotheken, welche für die verschiedenen Phasen der Datenanalyse und Modellentwicklung unerlässlich sind. Hier werden unter anderem die Methoden zum Zugriff auf und zum Lesen der Daten (Abschnitt 3.3.1), die spezifischen TensorFlow-Bibliotheken (Abschnitt 3.3.2), Ansätze zur Datenanalyse und -visualisierung (Abschnitt 3.3.3), und Techniken zur Modellbewertung und -interpretation (Abschnitt 3.3.4) erläutert. Abschnitt 3.3.5 stellt zusätzliche Werkzeuge vor, die im Rahmen der Studie eingesetzt wurden.

Im weiteren Verlauf des Kapitels, in den Abschnitten 3.4 und 3.5, steht die konkrete Arbeit mit den Daten im Vordergrund. Nach dem Laden des Datensatzes (Abschnitt 3.4) folgen die Vorverarbeitung der Daten und die Datenaugmentation (Abschnitt 3.5.1), um den Datensatz für die Nutzung im Deep Learning-Modell vorzubereiten.

Abschnitt 3.6 beschreibt den Auswahlprozess des Deep Learning-Modells (Modellarchitektur). Hier wird der gewählte Modelltyp vorgestellt und der Modellaufbau skizziert (Abschnitt 3.6.1). Der abschließende Abschnitt 3.6.2 widmet sich der Überwachung der Modellleistung, welche während des Modelltrainings erfolgt und ein entscheidender Schritt im gesamten Prozess ist.

In der Fertigungsindustrie ist die Reduzierung von Verarbeitungsfehlern im Herstellungsprozess wichtig, um die Gewinne zu maximieren. Die manuelle Überprüfung von Produkten ist ein zeitaufwendiger Prozess, der oft zu ungleicher Genauigkeit und erhöhten Arbeitskosten führt. In diesem Kapitel wird eine Case Study vorgestellt, die den Einsatz

von Machine Learning zur Automatisierung der Qualitätskontrolle in der Gussfertigung demonstriert.[47]

Die Herstellung von Gussprodukten ist ein wichtiger Teil der Fertigungsindustrie. Bei diesem Prozess können jedoch unerwünschte Unregelmäßigkeiten, auch bekannt als Gussfehler, auftreten. Diese Fehler können sich negativ auf die Qualität des Endprodukts auswirken und führen oft zu Ausschuss und Verlusten für das Unternehmen. Die manuelle Überprüfung von Gussprodukten ist zeitaufwendig und ungenau, was die Notwendigkeit einer automatisierten Qualitätskontrolle aufzeigt.[48]

Um die Qualitätskontrolle in der Gussfertigung zu automatisieren, wurde ein Datensatz von Gussfehlern gesammelt, einschließlich Blasen, Lunker, Graten, Schrumpfungsfehlern, Defekten des Formmaterials, Defekten des Gießmaterials und metallurgischen Defekten. Basierend auf diesen Daten wird ein Machine-Learning-Modell trainiert, das auf der Convolutional Neural Network (CNN) Architektur basiert. Der Datensatz steht im Internet frei zur Verfügung.[49]

Das Modell wird auf Testdaten angewendet, um die verschiedenen Arten von Gussfehlern automatisch zu erkennen und zu klassifizieren. Das Modell soll dabei die verschiedenen Gussfehler erkennen und eine Bewertung zur Produktqualität abgeben.

3.1 Zielsetzung des Beispiels

Das Ziel dieses Beispiels ist es, die Wirksamkeit von Deep Learning-Algorithmen bei der automatischen Qualitätskontrolle in der Fertigungsindustrie zu untersuchen. Insbesondere soll dieses Kapitel zeigen, wie Deep Learning-Modelle mithilfe von Bilderkennung eingesetzt werden können, um Defekte in Gussprodukten zu erkennen und zu klassifizieren. Ziel ist es, die Effizienz und Genauigkeit der Qualitätskontrolle zu verbessern, indem manuelle Inspektionsprozesse durch maschinelle Lernmethoden ersetzt werden.

Um diese Zielsetzung zu erreichen, wird ein Deep Learning-Modell trainiert und evaluiert, das in der Lage ist, verschiedene Arten von Defekten in Gussprodukten zu erkennen und zu klassifizieren. Die Performance des Modells wird anhand von Validierungsmetriken gemessen und mit manuellen Inspektionsprozessen verglichen, um die Vorteile der automatischen Qualitätskontrolle zu demonstrieren.

Die Ergebnisse dieser Arbeit können Fertigungsunternehmen helfen, ihre Qualitätskontrollprozesse zu optimieren und ihre Produktionsprozesse effizienter und kostengünstiger zu gestalten.

3.2 Auswahl der Datensätze

Die Auswahl geeigneter Datensätze ist ein wichtiger Aspekt bei der Erstellung eines Deep Learning-Modells. Für dieses Beispiel wurden Bildaufnahmen von einem Unterwasserpumpenlaufrad verwendet. Hierfür wird das Casting Defects Dataset verwendet, der öffentlich zugänglich ist und eine ausreichende Anzahl von Bildern mit verschiedenen Defekten enthält. Der Datensatz umfasst insgesamt 7.348 Bilder, die alle in Graustufen sind und eine Größe von 300x300 Pixeln haben. Um die Leistung des Modells zu verbessern, werden Data Augmentation-Techniken verwendet. Diese Techniken ermöglichen es, mehr Trainingsdaten zu generieren, indem die vorhandenen Bilder manipuliert werden und Verzerrungen hinzugefügt werden, um das Modell robuster zu machen und Overfitting zu vermeiden. Overfitting tritt auf, wenn das Modell die Trainingsdaten zu gut lernt und somit nicht gut auf neue, ungesehene Daten generalisiert. In den Bildern sind bereits Augmentationen angewendet worden. Darüber hinaus wurden für die Datensammlung zusätzlich Bilder mit einer Größe von 512x512 Pixeln und ohne Augmentationen bereitgestellt. Diese Datensätze bestehen aus Bildern von 519 fehlerfreien und 781 defekten Teilen.

Für die Aufnahme der Bilder wurde eine spezielle Beleuchtung verwendet, um stabile Lichtverhältnisse zu gewährleisten.
Die Bilder werden in zwei Kategorien unterteilt:

1. Defekt
2. Ok

Um ein Klassifikationsmodell zu erstellen, wurden die Daten für das Training und Testing in zwei Ordner aufgeteilt. Beide Ordner enthalten Unterordner für defekt_front und ok_front.

Die Verteilung der Bilder in den Trainings- und Testdaten sieht folgendermaßen aus:

1. Trainingsdaten: defekt_front hat 3758 Bilder und ok_front hat 2875 Bilder
2. Testdaten: defekt_front hat 453 Bilder und ok_front hat 262 Bilder

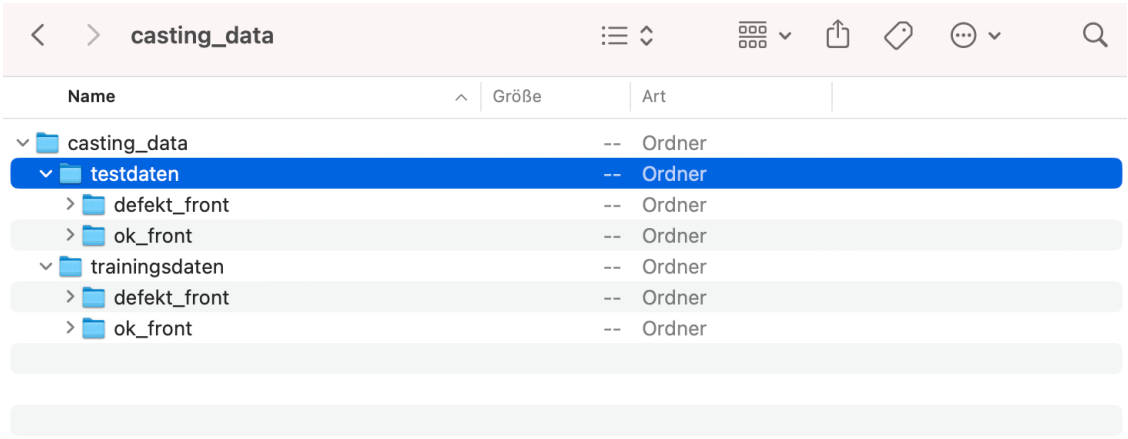


Abbildung 4: Ordnerstruktur des Datensatzes [Eigene Darstellung]

In Abbildung 5 ist das zentrale Objekt der Untersuchung - eine Unterwassermotorpumpe abgebildet. Ein wesentliches Element für die korrekte Funktionsweise der Pumpe ist der Rotor (siehe Abbildung 6), das Herzstück der Pumpe. Dieser ist für die Umwandlung von Energie verantwortlich, die den Betrieb der Pumpe ermöglicht.

Um die Qualität dieses zentralen Bauteils zu überprüfen, wird in der Case Study Computer Vision eingesetzt. Mit ihrer Hilfe können Qualitätsmängel am Rotor erkannt und klassifiziert werden. Diese Technik ermöglicht es, fehlerhafte Bauteile frühzeitig zu identifizieren und auszutauschen, bevor sie zu einer Beeinträchtigung der Pumpenleistung oder gar zu einem Ausfall führen können.

Abbildung 7 zeigt einen intakten Rotor. Aus der Abbildung ist zu erkennen, dass die Oberfläche gleichmäßig und ohne erkennbare Beschädigungen ist.

Im Gegensatz dazu zeigt Abbildung 8 einen defekten Rotor. Hierbei sind deutliche Abweichungen und Mängel zu erkennen, die darauf hinweisen, dass dieser Rotor nicht mehr funktionsfähig ist. Solche Mängel können die Leistung der Pumpe erheblich beeinträchtigen und im schlimmsten Fall einen vollständigen Ausfall verursachen.[50]

Abbildung 5: Bild der Unterwassermotorpumpe [50]

Abbildung 6: Bild vom Rotor [50]

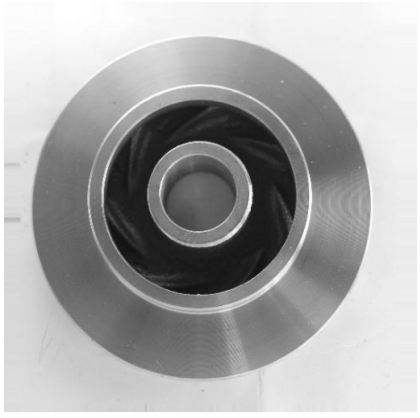


Abbildung 7: Bild von einem fehlerfreien Rotor [50]

Abbildung 8: Bild von einem defekten Rotor [50]

3.3 Importieren der benötigten Bibliotheken

Die Programmiersprache Python wird als Grundlage für die Implementierung des Modells verwendet. Python ist eine höchst modulare und erweiterbare Programmiersprache, die auf dem Konzept von Paketen und Modulen aufgebaut ist. Pakete sind Sammlungen von Modulen, welche wiederum Sammlungen von Funktionen und Klassen sind, die spezifische Aufgaben erfüllen. Die Verwendung von Bibliotheken bietet zahlreiche Vorteile wie die Wiederverwendbarkeit von Code, die Verbesserung der Lesbarkeit und die Optimierung von Entwicklungszeiten. Durch das Einbinden von Bibliotheken können Entwickler auf bereits existierende Lösungen zurückgreifen und sich auf die spezifischen Anforderungen ihres Projekts konzentrieren.[46]

Im folgenden Fallbeispiel, bei dem es darum geht, mithilfe von Computer Vision zu analysieren, welche Teile fehlerhaft sind und welche nicht, werden verschiedene Bibliotheken importiert, die unterschiedliche Funktionen bereitstellen. Im Folgenden werden die relevanten Bibliotheken und ihre Bedeutung für das Fallbeispiel erläutert.

3.3.1 Zugriff auf und Lesen von Daten

```
In [1]: # Importieren der Bibliotheken zum Zugriff auf und Lesen von Daten
        import os
        import pathlib
        import PIL
        import cv2
        import skimage
        from IPython.display import Image, display
        from matplotlib.image import imread
        import matplotlib.cm as plt
```

- **os** und **pathlib**: Diese Bibliotheken bieten grundlegende Funktionen zum Interagieren mit dem Dateisystem. Sie ermöglichen beispielsweise das Navigieren durch Verzeichnisse und das Lesen von Dateien.

- **PIL** (Python Imaging Library): PIL ist eine populäre Bibliothek zur Verarbeitung von Bildern. Sie wird in diesem Fallbeispiel verwendet, um Bilder zu öffnen, zu manipulieren und zu speichern.

- **cv2**: OpenCV (Open Source Computer Vision Library) ist eine Open-Source-Bibliothek, die speziell für Computer Vision entwickelt wurde. Sie bietet zahlreiche Funktionen zur Bildverarbeitung und -analyse.

- **skimage**: Die scikit-image Bibliothek ist eine Sammlung von Algorithmen zur Bildverarbeitung. Sie bietet eine einfach zu bedienende Schnittstelle und ist leicht erweiterbar.

- **IPython.display** und **matplotlib.image**: Diese beiden Bibliotheken bieten Funktionen zum Anzeigen von Bildern innerhalb des Jupyter-Notebooks.[52]

3.3.2 TensorFlow-Bibliotheken

```
In [2]: # Importieren der TensorFlow-Bibliotheken
        from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
        from tensorflow.keras.models import Sequential, load_model
        from tensorflow.keras.layers import Activation, Dropout, Flatten, Dense, Conv2D, MaxPooling2D
        from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
        from tensorflow.keras.utils import plot_model
        from tensorflow.keras import backend
```

- **tensorflow**, **keras** und zugehörige Module: TensorFlow ist eine Open-Source-Softwarebibliothek für maschinelles Lernen und künstliche Intelligenz. Keras ist eine Schnittstelle für TensorFlow und bietet eine benutzerfreundliche API zum Erstellen von neuronalen Netzwerken. Im Fallbeispiel werden TensorFlow und Keras verwendet, um Modelle zur Bilderkennung und -analyse zu erstellen und zu trainieren.

- **ImageDataGenerator**, **load_img**, **img_to_array**, **Sequential**, **load_model**, **Activation**, **Dropout**, **Flatten**, **Dense**, **Conv2D**, **MaxPooling2D**, **EarlyStopping**, **ModelCheckpoint**, **plot_model**, **backend**: Diese Funktionen und Klassen stammen aus der Keras-Bibliothek und werden verwendet, um das neuronale Netzwerk aufzubauen, zu trainieren und zu evaluieren. Dazu gehören die Erstellung von Daten-Generatoren, das Laden und Verarbeiten von Bildern, das Erstellen und Optimieren von Modellen sowie das Speichern und Laden von trainierten Modellen.[53]

3.3.3 Datenanalyse und Visualisierung

```
In [3]: # Importieren der Bibliotheken zur Datenanalyse und Visualisierung
        import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        from matplotlib.image import imread
        import holoviews as hv
        from holoviews import opts hv.extension('bokeh')
```

- **pandas**, **numpy**, **seaborn**, **matplotlib**, **holoviews** und **bokeh**: Diese Bibliotheken bieten umfangreiche Werkzeuge zur Datenanalyse und -visualisierung. Sie werden verwendet, um statistische Zusammenhänge in den Daten zu erkennen, Ergebnisse zu visualisieren und den Trainingsprozess der Modelle zu überwachen.[54]

3.3.4 Modellbewertung und -interpretation

```
In [4]: # Importieren der Bibliotheken zur Modellbewertung und -interpretation
        from sklearn.metrics import confusion_matrix, classification_report
        import shap
```

- **sklearn.metrics**: Diese Bibliothek bietet Funktionen zur Bewertung von Machine-Learning-Modellen, wie zum Beispiel die Erstellung von Konfusionsmatrizen oder Klassifikationsberichten.

- **shap**: Die SHAP (SHapley Additive exPlanations) Bibliothek hilft dabei, die Auswirkungen der einzelnen Merkmale auf die Vorhersagen von Machine-Learning-Modellen zu erklären. Im Fallbeispiel wird SHAP verwendet, um die Entscheidungen des Modells besser zu verstehen und die Identifikation fehlerhafter Teile zu interpretieren.[55]

3.3.5 Zusätzliche Werkzeuge

```
In [5]: # Importieren der Bibliotheken Zusätzliche Werkzeuge
        import warnings
        warnings.filterwarnings('ignore')
        import json
```

- **warnings**: Die **warnings** Bibliothek wird verwendet, um Warnungen zu steuern und im Bedarfsfall zu unterdrücken, um die Lesbarkeit der Notebook-Ausgaben zu erhöhen.

- **json**: Die **json** Bibliothek dient zum Lesen und Schreiben von JSON-Dateien, die zur Speicherung von Konfigurationen und Ergebnissen verwendet werden können.

Insgesamt unterstützen die oben genannten Bibliotheken die Implementierung und Analyse von Computer Vision-Lösungen in unserem Fallbeispiel. Durch die Verwendung dieser Bibliotheken können komplexe Aufgaben wie das Erstellen, Trainieren und Evaluieren von Modellen zur Identifizierung fehlerhafter Teile effizienter und leichter umgesetzt werden.[56]

3.4 Laden des Datensatzes

Um den Datensatz für das entwickelte Modell nutzbar zu machen, müssen die Daten zunächst eingelesen, in einer geeigneten Struktur abgelegt und Python den Zugriff darauf ermöglicht werden. In diesem Unterkapitel wird auf diese Schritte eingegangen.

```
In [6]: # Einlesen und Speicherung der Daten
        datensatz_pfad = "/Users/ShahrukhKhan/Desktop/Datensatz"
        trainingsdatensatz_pfad ="/Users/ShahrukhKhan/Desktop/Datensatz/casting_data/casting_data/trainingsdaten"
        testdatensatz_pfad = "/Users/ShahrukhKhan/Desktop/Datensatz/casting_data/casting_data/testdaten"
```

Die Bilddaten werden in einem Verzeichnis abgelegt, das in der Variable **datensatz_pfad** definiert wird. Hierbei handelt es sich um den Pfad zu dem Ordner, der die Bilder für das Training und das Testen des Modells enthält. Die Pfade zu den Unterordnern für das Training (**trainingsdatensatz_pfad**) und das Testen (**testdatensatz_pfad**) werden ebenfalls definiert. Die Verwendung dieser Pfade ermöglicht es, den Datensatz in einer strukturierten Weise abzulegen und darauf zuzugreifen. Die Organisation der Daten in Trainings- und Testsets ist entscheidend, um das Modell effektiv zu trainieren und seine Leistung auf unbekannte Daten zu bewerten.[55]

3.5 Vorverarbeitung der Daten

Bevor Daten für maschinelles Lernen und insbesondere für Computer Vision Modelle verwendet werden können, ist es häufig notwendig, sie einer Vorverarbeitung zu unterziehen. Die Vorverarbeitung umfasst eine Reihe von Schritten, um die Daten für das Modell besser nutzbar zu machen und die Leistung des Modells zu verbessern. In diesem Unterkapitel wird auf die Gründe für die Vorverarbeitung eingegangen und erläutert, wie dies im Fallbeispiel umgesetzt wird.

Bedeutung der Vorverarbeitung

Die Vorverarbeitung von Daten ist aus mehreren Gründen wichtig:

1. Datenbereinigung: Rohdaten können unvollständig, inkonsistent oder fehlerhaft sein, wodurch die Leistung des Modells beeinträchtigt wird. Durch die Vorverarbeitung werden solche Probleme behoben, um eine höhere Modellgenauigkeit zu gewährleisten.

2. Datenreduktion: Unnötige oder redundante Daten können die Effizienz und die Leistung des Modells beeinträchtigen. Die Vorverarbeitung ermöglicht es, die Größe des Datensatzes zu reduzieren, ohne dabei wichtige Informationen zu verlieren.

3. Datentransformation: Die Rohdaten müssen häufig in ein geeignetes Format gebracht werden, damit sie von einem Modell verarbeitet werden können. Dies kann zum Beispiel die Normalisierung von Pixelwerten oder die Umwandlung von Farbbildern in Graustufenbilder umfassen.

Anwendung der Vorverarbeitung

Im gegebenen Fallbeispiel wird die Vorverarbeitung der Bilddaten mithilfe der Keras-Bibliothek und der **ImageDataGenerator** Klasse durchgeführt. Im Folgenden werden die einzelnen Codeabschnitte und deren Bedeutung für das Fallbeispiel erläutert.

```
In [4]: # Vorverarbeitung der Daten
        image_shape = (300,300,1)


        image_gen = ImageDataGenerator(rescale=1/255,
                                       validation_split = 0.2,
                                       zoom_range=0.1,
                                       brightness_range=[0.9,1.0])

        train_set = image_gen.flow_from_directory(trainingsdatensatz_pfad,
                                                  target_size=image_shape[:2],
                                                  color_mode="grayscale",
                                                  classes={'defekt_front': 0, 'ok_front': 1},
                                                  batch_size=32,
                                                  class_mode='binary',
                                                  subset = 'training',
                                                  shuffle=True,
                                                  seed=0)

        test_set = image_gen.flow_from_directory(testdatensatz_pfad,
                                                 target_size=image_shape[:2],
                                                 color_mode="grayscale",
                                                 classes={'defekt_front': 0, 'ok_front': 1},
                                                 batch_size=32,
                                                 class_mode='binary',
                                                 subset = 'validation',
                                                 shuffle=False,
                                                 seed=0)

        Found 6633 images belonging to 2 classes.
        Found 715 images belonging to 2 classes.
```

Im vorliegenden Code wird eine Instanz von **ImageDataGenerator** für den Trainingsdatensatz erstellt, die die folgenden Parameter verwendet:

- **rescale**: Normalisiert die Pixelwerte des Bildes durch Multiplikation mit einem konstanten Faktor (z.B. 1/255 -> Normalisierung der RGB-Werte jedes Pixels im Bereich von 0,0 bis 1,0)

- **zoom_range**: 0,1, um das Bild leicht zu vergrößern oder zu verkleinern
- **brightness_range**: [0,9, 1,0], um die Helligkeit des Bildes geringfügig zu ändern

Die Instanz von **ImageDataGenerator** wird dann verwendet, um Trainings- und Testdatensätze zu erstellen, indem die Methode **flow_from_directory()** aufgerufen wird. Die Methode nimmt die folgenden Argumente entgegen:

- **train_path** und **test_path**: Pfade zu den Ordnern mit Trainings- und Testbildern
- **target_size**: Die gewünschte Größe der Bilder (in diesem Fall 300x300 Pixel)
- **color_mode**: "grayscale", um die Bilder in Graustufen umzuwandeln
- **classes**: Ein Dictionary, das die Klassenbezeichnungen den entsprechenden Ordnern zuordnet
- **batch_size**: Die Größe der Mini-Batches, die während des Trainings verwendet werden sollen (in diesem Fall 32)
- **class_mode**: "binary", da es zwei Klassen gibt (defekt und einwandfrei)
- **shuffle**: Gibt an, ob die Reihenfolge der Bilder vor jedem Durchlauf gemischt werden soll (True für Trainingsdaten, False für Testdaten)
- **seed**: Ein Seed-Wert, um die Reproduzierbarkeit der Ergebnisse zu gewährleisten[57]

Nachdem die Vorverarbeitung abgeschlossen ist, zeigt die Ausgabe die Anzahl der Bilder in den verschiedenen Datensätzen an. Dies bedeutet, dass der Trainingsdatensatz 6633 Bilder und der Testdatensatz 715 Bilder enthält. Durch die Vorverarbeitung der Bilddaten werden sie in einem geeigneten Format für das Computer-Vision-Modell bereitgestellt, das darauf abzielt, fehlerhafte und einwandfreie Gussprodukte zu analysieren. Die Verwendung von Graustufenbildern, die Normalisierung der Pixelwerte und die Aufteilung der Daten in Trainings-, Validierungs- und Testdatensätze sind wichtige Schritte, um eine höhere Modellleistung und eine bessere Generalisierung auf unbekannte Daten zu erreichen.

3.5.1 Datenaugmentation

In der Computer Vision und insbesondere bei der Bildklassifikation ist die Datenaugmentation eine wichtige Technik, um robustere Modelle zu entwickeln und die Generalisierungsleistung der Modelle zu verbessern. Bei begrenzten Datensätzen kann die Datenaugmentation dazu beitragen, das Problem der Überanpassung zu reduzieren, indem sie die Variation innerhalb des Datensatzes erhöht und die Muster, die das Modell lernt, diversifiziert.

Datenaugmentation bezieht sich auf die Erzeugung neuer Trainingsbilder durch Anwendung von zufälligen Transformationen auf die vorhandenen Bilder. Diese Transformationen können Drehungen, Skalierungen, Scherungen, Helligkeitsänderungen, Spiegelungen und Verschiebungen in der Höhe oder Breite umfassen.

```
In [5]: # Datenaugmentation
        img = cv2.imread(trainingsdatensatz_pfad + 'ok_front/cast_i0_0_1.jpeg')
        img_4d = img[np.newaxis]
        plt.figure(figsize=(25,10))
        generators = {"rotation":ImageDataGenerator(rotation_range=180),
                      "zoom":ImageDataGenerator(zoom_range=0.7),
                      "brightness":ImageDataGenerator(brightness_range=[0.2,1.0]),
                      "height_shift":ImageDataGenerator(height_shift_range=0.7),
                      "width_shift":ImageDataGenerator(width_shift_range=0.7)}
```

Im gegebenen Codebeispiel wird die Datenaugmentation mit Hilfe der Klasse **ImageDataGenerator** von Keras durchgeführt. Die verschiedenen Parameter, die in diesem Beispiel verwendet werden, sind:

- **rotation_range**: Rotiert das Bild innerhalb des angegebenen Bereichs (z.B. 180 -> zufällige Drehung im Bereich von -180° bis 180°)
- **zoom_range**: Skaliert das Bild innerhalb des angegebenen Bereichs (z.B. 0,7 -> zufällige Skalierung im Bereich von 1-0,7 bis 1+0,7)
- **brightness_range**: Ändert die Helligkeit des Bildes innerhalb des angegebenen Bereichs (z.B. [0,2, 1,0] -> zufällige Helligkeitsänderung im Bereich von 0,2 bis 1,0)
- **height_shift_range**: Verschiebt das Bild vertikal innerhalb des angegebenen Bereichs (z.B. 0,7 -> zufällige Verschiebung in einem Bereich von -0,7 * Höhe bis 0,7 * Höhe)
- **width_shift_range**: Verschiebt das Bild horizontal innerhalb des angegebenen Bereichs (z.B. 0,7 -> zufällige Verschiebung in einem Bereich von -0,7 * Breite bis 0,7 * Breite)

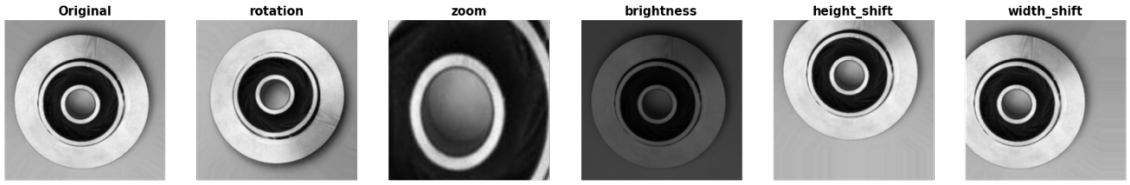


Abbildung 9: Angewendete Datenaugmentationstechniken [Eigene Darstellung]

Der Code zeigt Beispiele für verschiedene Datenaugmentationstechniken, die auf ein einzelnes Bild angewendet werden. In Abbildung 9 ist der Output dieser Beispiele dargestellt. Anschließend wird die Datenaugmentation für den Trainings- und Testdatensatz durchgeführt, indem ein **ImageDataGenerator** mit ausgewählten Parametern erstellt wird. Die Einstellungen für die Datenaugmentation sollten sorgfältig gewählt werden, um eine zu starke Verzerrung der Bilder zu vermeiden, die das Training erschweren könnte. Für jedes Problem und jeden Datensatz müssen die Parameter fein abgestimmt werden, um ein erfolgreiches Training zu gewährleisten.

Im Fallbeispiel ist die Datenaugmentation besonders nützlich, um die Robustheit des Modells zu erhöhen und eine bessere Leistung bei der Identifizierung von Fehlern in neuen, ungesehenen Bildern zu erzielen. Da Industrieprodukte oft unterschiedliche Beleuchtungsbedingungen, Orientierungen und Größen haben können, hilft die Datenaugmentation dabei, das Modell darauf zu trainieren, diese Variationen zu berücksichtigen und trotzdem korrekte Vorhersagen zu treffen.

Nachdem der **ImageDataGenerator** konfiguriert und die Trainings- und Testdatensätze erstellt wurden, kann das Modell mit diesen Daten trainiert und getestet werden.[57]

3.6 Auswahl des Deep Learning-Modells (Modellarchitektur)

Als Deep-Learning-Modellarchitektur wird ein künstliches neuronales Netzwerk (KNN) mit einer Convolutional Neural Network (CNN)-Architektur für die Analyse von Fehlern in Gussteilen eingesetzt. Das CNN wurde gewählt, da es sich um eine einfache, aber effektive Architektur handelt, die sich besonders gut für die Verarbeitung von Bilddaten eignet. CNNs sind besonders gut geeignet für Bilderkennungsaufgaben, da sie in der Lage sind, lokale und hierarchische Muster in Bilddaten zu erkennen. Die Architektur des Modells besteht aus einer Sequenz von Schichten, die jeweils spezifische Funktionen zur Extraktion von Merkmalen aus den Bilddaten erfüllen. Die Hauptbestandteile des Modells sind Faltungsschichten (Conv2D), Pooling-Schichten (MaxPooling2D), Dropout-

Schichten und vollständig verbundene Schichten (Dense). Die Faltungsschichten dienen zur Extraktion von Merkmalen aus den Eingabebilddaten, während die Pooling-Schichten zur Reduzierung der räumlichen Dimensionen beitragen. Die Dropout-Schichten verhindern Overfitting, indem sie während des Trainings zufällig ausgewählte Neuronen deaktivieren. Schließlich dienen die vollständig verbundenen Schichten der Klassifikation der extrahierten Merkmale in fehlerhafte und einwandfreie Gussteile.

Somit besteht das vorgestellte CNN-Modell aus folgenden Schichten:

1. Conv2D-Schichten: Diese Schichten verwenden Filter, um Merkmale aus den Eingabebildern zu extrahieren. Die Anzahl der Filter, die Größe der Filterkerne und die Schrittlänge können angepasst werden, um die Merkmalsextraktion zu optimieren. In diesem Modell werden drei Conv2D-Schichten verwendet, die jeweils 16, 32 und 64 Filter haben.

2. MaxPooling2D-Schichten: Diese Schichten dienen dazu, die Größe der Feature-Maps zu reduzieren, indem sie die maximalen Werte in bestimmten Bereichen auswählen. Dadurch wird die Rechenleistung verringert, und das Modell kann relevante Merkmale besser erfassen. In diesem Modell folgt jeder Conv2D-Schicht eine MaxPooling2D-Schicht.

3. Flatten-Schicht: Diese Schicht konvertiert die mehrdimensionalen Feature-Maps in einen eindimensionalen Vektor, der in den nachfolgenden vollständig verbundenen Schichten verarbeitet werden kann.

4. Dense-Schichten: Diese Schichten stellen vollständig verbundene neuronale Netzwerke dar, die die extrahierten Merkmale verarbeiten und schließlich eine binäre Klassifikation (fehlerhaft oder einwandfrei) vornehmen. In diesem Modell gibt es zwei Dense-Schichten, die erste mit 224 Neuronen und die zweite mit nur einem Neuron, da es sich um eine binäre Klassifikation handelt.

5. Dropout-Schicht: Diese Schicht wird verwendet, um Overfitting zu verhindern, indem zufällig ausgewählte Neuronen während des Trainings deaktiviert werden.

Das CNN-Modell wird mit der Verlustfunktion **binary_crossentropy**, dem Optimierer **adam** und der Metrik **accuracy** kompiliert. Die Wahl dieser Parameter trägt dazu bei, die

Leistung des Modells bei der Klassifikation von fehlerhaften und einwandfreien Gussteilen zu optimieren.

Die Stärken dieses Modells liegen in seiner Einfachheit und Effektivität bei der Verarbeitung von Bilddaten. Es ist jedoch wichtig zu beachten, dass das Modell möglicherweise nicht die höchstmögliche Genauigkeit erreicht, da es sich um ein einfaches CNN handelt und der Schwerpunkt auf dem Erlernen der Grundlagen der Bildanalysetechnologie liegt. Für komplexere Anwendungen oder höhere Genauigkeitsanforderungen könnten fortgeschrittenere Modellarchitekturen wie ResNet oder Inception in Betracht gezogen werden Nachdem die Modellarchitektur und ihre Bestandteile ausführlich erläutert wurden, wird nun der Programmcode erläutert und dessen Bedeutung für das Fallbeispiel beschrieben.

1. Zunächst wird die aktuelle TensorFlow-Sitzung gelöscht, um sicherzustellen, dass keine alten Modelle oder Einstellungen beibehalten werden:

```
In [ ]: backend.clear_session()
```

2. Anschließend wird das Modell als **Sequential**-Objekt erstellt, das als Container für die verschiedenen Schichten dient:

```
In [ ]: model = Sequential()
```

3. Die Conv2D-, MaxPooling2D-, Flatten-, Dense- und Dropout-Schichten werden dem Modell hinzugefügt. Die spezifischen Parameter für jede Schicht wurden zuvor erläutert:

```
In [ ]: model.add(Conv2D(filters=16, kernel_size=(7,7), strides=2, input_shape=image_shape, activation='relu',
                       padding='same'))
        model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
        model.add(Conv2D(filters=32, kernel_size=(3,3), strides=1, input_shape=image_shape, activation='relu',
                       padding='same'))
        model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
        model.add(Conv2D(filters=64, kernel_size=(3,3), strides=1, input_shape=image_shape, activation='relu',
                       padding='same'))
        model.add(MaxPooling2D(pool_size=(2, 2), strides=2))
        model.add(Flatten())
        model.add(Dense(units=224, activation='relu'))
        model.add(Dropout(rate=0.2))
        model.add(Dense(units=1, activation='sigmoid'))
```

4. Das Modell wird mit der zuvor beschriebenen Verlustfunktion, dem Optimierer und der Metrik kompiliert:

```
In [ ]: model.compile(loss='binary_crossentropy',
                      optimizer='adam',
                      metrics=['accuracy'])
```

5. Der Befehl **model.summary()** zeigt eine Zusammenfassung des erstellten Modells, einschließlich der Schichten, ihrer Ausgabegrößen und der Anzahl der Parameter:

```
In [ ]: model.summary()
```

Im Fallbeispiel zur Analyse fehlerhafter und einwandfreier Gussteile mit Computer Vision wird das erstellte CNN-Modell eingesetzt. Nachdem das Modell erstellt und kompiliert wurde, kann es mit den vorbereiteten und augmentierten Bilddaten trainiert werden. Die **train_set**- und **test_set**-Variablen, die zuvor erstellt wurden, enthalten die augmentierten Trainings- und Testdaten und können verwendet werden, um das Modell zu trainieren und seine Leistung zu bewerten.

Das hier vorgestellte Modell ist eine gute Ausgangsbasis. Es kann jedoch erforderlich sein, die Modellarchitektur oder die Hyperparameter weiter anzupassen, um eine höhere Genauigkeit oder Leistung für spezifische Anwendungsfälle zu erzielen.[58]

3.6.1 Modellaufbau

In diesem Unterabschnitt der Bachelorarbeit wird auf den Aufbau des Modells eingegangen und anschließend erklärt, wie das Modell arbeitet.
Der Programmcode besteht aus mehreren Teilen, die jeweils unterschiedliche Funktionen erfüllen. Im Folgenden werden die einzelnen Codeabschnitte erläutert und ihre Bedeutung für das Fallbeispiel dargestellt.

```
In [ ]: early_stop = EarlyStopping(monitor='val_loss',patience=2)
        checkpoint = ModelCheckpoint(filepath=model_save_path, verbose=1, save_best_only=True, monitor='val_loss')
```

1. EarlyStopping und ModelCheckpoint: Diese beiden Funktionen sind wichtige Hilfsmittel zur Verbesserung der Modellleistung und zur Vermeidung von Overfitting. EarlyStopping ermöglicht es, das Training frühzeitig zu beenden, wenn die Validierungsverluste in einer bestimmten Anzahl von aufeinanderfolgenden Epochen nicht weiter verbessert werden. ModelCheckpoint speichert das beste Modell basierend auf der Validierungsverlustfunktion nach jeder Epoche, sodass das am besten geeignete Modell für die spätere Verwendung ausgewählt werden kann.

```
In [ ]: n_epochs = 20
        results = model.fit_generator(train_set,
                                      epochs=n_epochs,
                                      validation_data=test_set,
                                      callbacks=[early_stop,checkpoint])
```

2. Modelltraining: Mit der **fit_generator**-Funktion wird das Modell für eine bestimmte Anzahl von Epochen trainiert. Dabei werden die Trainings- und Testdaten sowie die zuvor definierten EarlyStopping- und ModelCheckpoint-Callbacks

übergeben. Die Trainingsdaten werden verwendet, um das Modell zu trainieren, während die Testdaten zur Validierung der Modellleistung dienen. Nach jeder Epoche wird der Trainingsverlust, die Trainingsgenauigkeit, der Validierungsverlust und die Validierungsgenauigkeit ausgegeben und gegebenenfalls das Modell gespeichert.

```
In [ ]: model_history = { i:list(map(lambda x: float(x), j)) for i,j in results.history.items() }
        with open('model_history.json', 'w') as f:
            json.dump(model_history, f, indent=4)
```

3. Modellhistorie: Die Trainingshistorie des Modells, einschließlich der Verluste und Genauigkeiten für jede Epoche, wird in einem JSON-Format gespeichert. Diese Informationen können später verwendet werden, um die Leistung des Modells während des Trainings zu analysieren und gegebenenfalls Verbesserungen vorzunehmen.

Zusammenfassend lässt sich sagen, dass der bereitgestellte Programmcode ein Convolutional Neural Network erstellt, trainiert und optimiert, um fehlerhafte Gussteile im Fallbeispiel zu erkennen. Die Verwendung von EarlyStopping und ModelCheckpoint stellt sicher, dass das Modell nicht übertrainiert wird und das bestmögliche Modell für die gegebene Aufgabe ausgewählt wird. Die Speicherung der Modellhistorie ermöglicht eine detaillierte Analyse der Modellleistung während des Trainings und die Wiederverwendung eines trainierten Modells erlaubt die Anwendung auf neue Daten oder die weitere Optimierung des Modells.[59]

3.6.2 Überwachung der Modellleistung

Die Überwachung der Modellleistung ist ein entscheidender Schritt in jedem maschinellen Lernverfahren, insbesondere in Computer Vision-Anwendungen. Der primäre Zweck der Leistungsüberwachung besteht darin, die Qualität des Modells zu bewerten und seine Effektivität bei der Durchführung der beabsichtigten Aufgaben zu bestimmen. Bei der Überwachung der Modellleistung geht es nicht nur darum, die aktuellen Fähigkeiten des Modells zu bewerten, sondern auch um die Identifizierung von Bereichen, in denen Verbesserungen möglich und notwendig sind.

Die Leistung eines Modells wird typischerweise über Metriken wie Genauigkeit (accuracy), Verlust (loss) und Validierungsverlust (val_loss) bewertet. Diese Metriken geben einen Einblick in die Leistungsfähigkeit des Modells, sowohl in Bezug auf seine

Fähigkeit, korrekte Vorhersagen zu treffen, als auch in Bezug auf seine Fähigkeit, Fehler zu minimieren.

Ein Modell gilt als gut trainiert, wenn es eine hohe Genauigkeit aufweist und der Verlust minimal ist. Im Gegensatz dazu muss ein Modell, das eine geringe Genauigkeit und einen hohen Verlust aufweist, weiter optimiert und trainiert werden. Darüber hinaus ist es wichtig, nicht nur die Leistung des Modells auf den Trainingsdaten zu bewerten, sondern auch auf den Validierungsdaten, um Overfitting zu vermeiden.

```
In [ ]: losses = pd.DataFrame(model_history)
        losses.index = map(lambda x : x+1, losses.index)
        losses.head(3)
```

Der oben gezeigte Programmcode und dessen Output zeigen eine Methode zur Überwachung der Modellleistung. Der Code erstellt ein DataFrame, das die Modellhistorie speichert. Dies umfasst die Verlust- und Genauigkeitswerte für jede Epoche sowohl für die Trainings- als auch für die Validierungsdaten.

	loss	accuracy	val_loss	val_accuracy
1	0.589113	0.679331	0.393526	0.818182
2	0.405581	0.809287	0.362537	0.844755
3	0.284947	0.881652	0.191958	0.927273

Tabelle 1: Verlust- und Genauigkeitswerte der ersten drei Epochen [Eigene Darstellung]

Die Ausgabe des Codes zeigt die ersten drei Epochen des Modelltrainings. In der ersten Epoche beträgt der Verlust 0,589113 und die Genauigkeit 0,679331. Die entsprechenden Werte für den Validierungsverlust und die Validierungsgenauigkeit betragen 0,393526 bzw. 0,818182. Mit fortschreitendem Training (in der zweiten und dritten Epoche) kann beobachtet werden, dass sowohl der Verlust als auch der Validierungsverlust abnehmen, während die Genauigkeit und die Validierungsgenauigkeit zunehmen. Dies deutet darauf hin, dass das Modell im Laufe der Zeit besser wird.

Ein wesentliches Instrument zur Überwachung der Modellleistung ist das Modelleistungsdiagramm, welches die Entwicklung der Modellmetriken über die Epochen hinweg visualisiert. In einem solchen Diagramm werden typischerweise die Werte für Trainingsverlust (Training Loss), Validierungsverlust (Validation Loss), Trainingsgenauigkeit (Training Accuracy) und Validierungsgenauigkeit (Validation Accuracy) über die Anzahl der Epochen aufgetragen.

Die Metriken Loss und Accuracy sind zentrale Indikatoren für die Qualität eines trainierten Modells. Die Loss-Funktion misst den Fehler eines Modells, das heißt, sie

quantifiziert, wie gut die Vorhersagen des Modells den tatsächlichen Werten entsprechen. Ein niedriger Verlust deutet auf gute Vorhersagen hin, während ein hoher Verlust auf schlechte Vorhersagen hindeutet. Die Accuracy hingegen gibt an, wie oft das Modell die richtige Vorhersage getroffen hat. Eine Accuracy von 1 bedeutet, dass das Modell alle Vorhersagen korrekt getroffen hat, während eine Accuracy von 0 bedeutet, dass alle Vorhersagen falsch waren.[60]

```
In [ ]: g = hv.Curve(losses.loss, label='Training Loss')
            * hv.Curve(losses.val_loss, label='Validation Loss') \
            * hv.Curve(losses.accuracy, label='Training Accuracy')
            * hv.Curve(losses.val_accuracy, label='Validation Accuracy')

        g.opts(opts.Curve(xlabel="Epochs",
                          ylabel="Loss / Accuracy",
                          width=700,
                          height=400,
                          tools=['hover'],
                          show_grid=True,
                          title='Model Evaluation')).opts(legend_position='bottom')
```

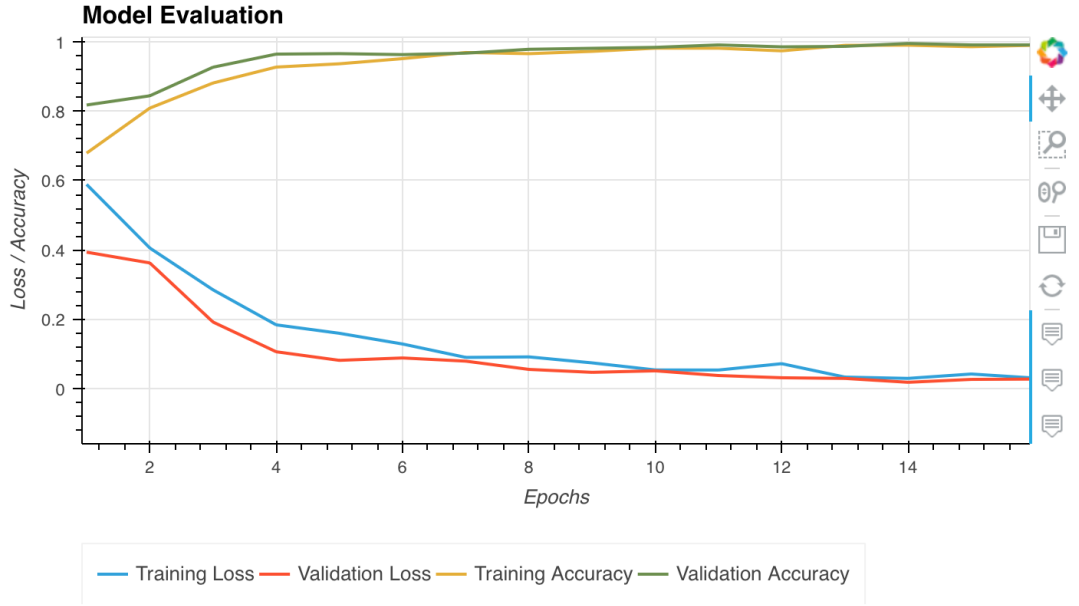


Abbildung 10: Trainingsverlust-, Validierungsverlust-, Trainingsgenauigkeits- und Validierungsgenauigkeitskurve pro Epoche [Eigene Darstellung]

Betrachtet man das spezifische Diagramm, das durch den gegebenen Code erzeugt wird. Zeigt sich, dass mit zunehmender Anzahl an Epochen der Trainings- und Validierungsverlust kontinuierlich abnehmen, während die Trainings- und Validierungsgenauigkeit steigen. Dies deutet darauf hin, dass das Modell im Laufe der Epochen immer besser darin wird, korrekte Vorhersagen zu treffen und den Fehler zu minimieren.

Insbesondere zeigt das Diagramm, dass ab etwa der achten Epoche sowohl die Trainings- als auch die Validierungsgenauigkeit nahezu 1 erreichen und die Trainings- und Validierungsverluste nahezu 0 erreichen. Dies ist ein starkes Indiz dafür, dass das Modell zu

diesem Zeitpunkt gut trainiert ist und effektiv zwischen fehlerhaften und fehlerfreien Teilen unterscheiden kann.

Der vorgegebene Code erstellt ein solches Diagramm mithilfe der Holoviews-Bibliothek. Er zeichnet vier Kurven für die Trainingsverluste, Validierungsverluste, Trainingsgenauigkeiten und Validierungsgenauigkeiten gegen die Anzahl der Epochen. Die Diagrammoptionen werden dann so eingestellt, dass die X-Achse die Epochen und die Y-Achse die Loss/Accuracy repräsentiert. Zudem wird eine Legende am unteren Rand des Diagramms und ein Hover-Tool hinzugefügt, mit dem der Benutzer die genauen Werte für jede Epoche sehen kann.

Dieses Diagramm ermöglicht eine effektive Überwachung des Trainingsprozesses und hilft, den optimalen Zeitpunkt für den Einsatz des Modells zu bestimmen.
Die Tatsache, dass sowohl die Training- als auch die Validierungsgenauigkeit gegen 1 konvergieren und die Verluste gegen 0, deutet auf ein gut trainiertes Modell hin. Dieses Verhalten weist darauf hin, dass das Modell sowohl auf den Trainingsdaten als auch auf den Validierungsdaten, die es während des Trainings nie gesehen hat, gut abschneidet. Dies ist ein deutliches Zeichen dafür, dass das Modell eine hohe Generalisierungsfähigkeit aufweist - eine essenzielle Eigenschaft, um auch auf neuen, unbekannten Daten zuverlässige Vorhersagen treffen zu können.[61]

Das Endziel ist es, ein Modell zu haben, das eine hohe Genauigkeit aufweist und einen minimalen Verlust auf den Validierungsdaten hat. Dadurch kann das Modell effektiv zwischen fehlerhaften und nicht fehlerhaften Teilen unterscheiden, was zur Qualitätssicherung in Produktionsprozessen beitragen kann.

4 Ergebnisse

Das folgende Kapitel behandelt die Validierung und Auswertung des zuvor entwickelten Deep Learning-Modells. Nach einer sorgfältigen und umfassenden Methodenentwicklung, Datenauswahl und Modellbildung wird hier nun geprüft, wie das Modell in der Praxis abschneidet.

Die Evaluierung des Modells mittels Testdaten bildet den ersten Schwerpunkt dieses Kapitels. Dabei wird das trainierte Modell auf einen separaten Datensatz angewendet, um seine Leistung zu beurteilen. Dieser Schritt ist von zentraler Bedeutung, da er Aufschluss darüber gibt, wie gut das Modell voraussichtlich auf neue, unbekannte Daten reagieren wird. Im Speziellen wird hierbei die Fähigkeit des Modells zur Identifizierung und Klassifizierung von Qualitätsmerkmalen untersucht.

Nach der Evaluierung folgt ein direkter Vergleich des Deep Learning-Modells mit traditionellen Methoden der Bildverarbeitung und Qualitätskontrolle. Ziel dieses Vergleichs ist es, ein umfassendes Verständnis der relativen Stärken und Schwächen der jeweiligen Ansätze zu erlangen. Durch diese vergleichende Analyse werden die Vor- und Nachteile des Einsatzes von Deep Learning in der Qualitätskontrolle konkret aufgezeigt.

Abschließend werden die Ergebnisse und Erkenntnisse aus der Modellvalidierung und dem Vergleich diskutiert. Diese Diskussion dient dazu, die gefundenen Resultate in einen größeren Kontext einzuordnen und ihre Implikationen für das Feld der Qualitätskontrolle und die möglichen zukünftigen Anwendungen von Deep Learning-Technologien zu erörtern. Dieser Abschnitt stellt einen kritischen Beitrag zur Interpretation und Verständnis der Ergebnisse dar und bereitet den Weg für das abschließende Fazit und die Zusammenfassung der Arbeit.

4.1 Evaluation des Modells mit Testdaten

Die Evaluation des Modells mit Testdaten ist ein entscheidender Schritt, um die Leistungsfähigkeit eines Modells zu bewerten. Im Allgemeinen wird das Modell nach dem Training auf einem separaten Datensatz getestet, der während des Trainingsprozesses nicht verwendet wurde. Dieser Prozess soll sicherstellen, dass das Modell auf unbekannten Daten gut abschneidet und über die Fähigkeit verfügt, auf neue Eingabebeispiele zu generalisieren.

Im Fallbeispiel wird das Modell zur Erkennung von Fehlern in industriellen Produkten verwendet. Daher wird es mit Bildern von fehlerhaften und fehlerfreien Teilen getestet. Um die Leistung des Modells zu bewerten, werden mehrere Kennzahlen verwendet, darunter Präzision, Recall und F1-Score.

Die Präzision ist das Verhältnis der korrekt positiv vorhergesagten Instanzen (True Positives) zur Gesamtzahl der als positiv vorhergesagten Instanzen (True Positives und False Positives). Im Fallbeispiel wäre die Präzision das Verhältnis der korrekt als fehlerhaft identifizierten Teile zur Gesamtzahl der als fehlerhaft klassifizierten Teile.

Der Recall ist das Verhältnis der korrekt positiv vorhergesagten Instanzen (True Positives) zur Gesamtzahl der tatsächlich positiven Instanzen (True Positives und False Negatives). Im Kontext des Fallbeispiels wäre der Recall das Verhältnis der korrekt als fehlerhaft identifizierten Teile zur Gesamtzahl der tatsächlich fehlerhaften Teile.

Der F1-Score ist das harmonische Mittel von Präzision und Recall und bietet eine einzige Metrik zur Bewertung des Modells, wenn sowohl Präzision als auch Recall wichtig sind. Dies ist oft der Fall, wenn die Kosten für False Positives und False Negatives unterschiedlich sind.[62]

Die folgenden Codeausschnitte erzeugen die Vorhersagen des Modells für den Testdatensatz und visualisieren die Ergebnisse in einer Konfusionsmatrix. Die Konfusionsmatrix ist ein hilfreiches Werkzeug, um die Leistung eines Modells zu visualisieren. Sie zeigt die Anzahl der True Positives, False Positives, True Negatives und False Negatives. Darüber hinaus erstellt der Code einen Klassifikationsbericht, der die oben beschriebenen Metriken für jede Klasse sowie die Gesamtgenauigkeit des Modells enthält.

```
In [ ]: #Konfusionsmatrix
        pred_probability = model.predict_generator(test_set)
        predictions = pred_probability > 0.5

        plt.figure(figsize=(10,6))
        plt.title("Konfusionsmatrix", size=20, weight='bold')
        sns.heatmap(
            confusion_matrix(test_set.classes, predictions),
            annot=True,
            annot_kws={'size':14, 'weight':'bold'},
            fmt='d',
            xticklabels=['Defect', 'OK'],
            yticklabels=['Defect', 'OK'])
        plt.tick_params(axis='both', labelsize=14)
        plt.ylabel('Actual', size=14, weight='bold')
        plt.xlabel('Predicted', size=14, weight='bold')
        plt.show()
```

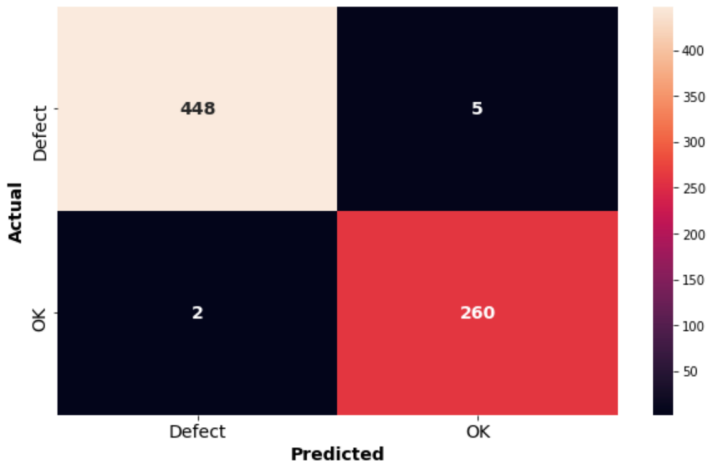


Abbildung 11: Konfusionsmatrix [Eigene Darstellung]

Im Fallbeispiel wurde nur 1 Teil fälschlicherweise als fehlerhaft klassifiziert (False Positive) und 2 Teile wurden fälschlicherweise als fehlerfrei klassifiziert (False Negative). Diese Werte sind entscheidend, um die Auswirkungen von Fehlklassifikationen zu verstehen.

Die Genauigkeitswerte von 0.990 für die Gesamtgenauigkeit und die gewichtete Durchschnittsgenauigkeit sowie die hohen Werte für Präzision, Recall und F1-Score sowohl für die Klasse 0 als auch für die Klasse 1 zeigen, dass das Modell in der Lage ist, fehlerhafte und fehlerfreie Teile effizient zu unterscheiden. Die hohe Präzision von 0.996 für die Klasse 0 deutet darauf hin, dass das Modell nur wenige False Positives erzeugt, also nur wenige fehlerfreie Teile fälschlicherweise als fehlerhaft klassifiziert. Gleichzeitig zeigt der hohe Recall-Wert von 0.992 für die Klasse 1, dass das Modell die meisten fehlerhaften Teile korrekt erkennt und nur wenige False Negatives produziert.

Es ist wichtig zu beachten, dass in bestimmten Anwendungsfällen die Kosten für False Positives und False Negatives unterschiedlich sein können. Im industriellen Kontext unseres Fallbeispiels kann ein False Negative, d.h. ein fehlerhaftes Teil, das als fehlerfrei klassifiziert wird, zu nachfolgenden Prozessproblemen oder sogar zu Fehlern im Endprodukt führen. Auf der anderen Seite kann ein False Positive, d.h. ein fehlerfreies Teil, das als fehlerhaft klassifiziert wird, zu unnötigen Stillstandzeiten in der Produktion oder zu unnötigen Kosten für die Neuproduktion führen. Daher kann es abhängig von den spezifischen Geschäftsanforderungen notwendig sein, das Modell so zu optimieren, dass entweder die Anzahl der False Positives oder die Anzahl der False Negatives minimiert wird. Zusammenfassend lässt sich sagen, dass die Überwachung der Modellleistung und die Evaluation mit Testdaten entscheidende Schritte im Machine Learning Prozess sind. Sie

ermöglichen es, die Qualität der Vorhersagen zu bewerten und gegebenenfalls Anpassungen am Modell oder am Trainingsprozess vorzunehmen, um die Leistung zu verbessern. Im Fallbeispiel hat das Modell gezeigt, dass es in der Lage ist, fehlerhafte und fehlerfreie Teile mit hoher Genauigkeit zu unterscheiden, was es zu einem wertvollen Werkzeug in der industriellen Qualitätskontrolle macht.[46]

4.1.1 Anwendung des trainierten Modells auf Testdatensätze

Das trainierte Modell wird auf Testdatensätzen angewendet, um seine Leistung unter Bedingungen zu bewerten, die denen nahekommen, unter denen das Modell in der Praxis eingesetzt wird. Die Testdatensätze repräsentieren neue, bisher ungesehene Daten, die das Modell zur Evaluation und Validierung verwendet. Die Hauptmotivation für diesen Schritt ist die Überprüfung, ob das Modell in der Lage ist, korrekte Vorhersagen für Daten zu treffen, die es während des Trainingsprozesses noch nicht gesehen hat.

Im Kontext des Fallbeispiels mit industriellen Produkten wird das Modell verwendet, um zu bestimmen, ob bestimmte Teile fehlerhaft sind oder nicht. Die Prognosen basieren auf den visuellen Merkmalen der Teile, die aus den Bildern extrahiert werden. Der Vorgang der Vorhersage beinhaltet mehrere Schritte. Zunächst wird das Bild des zu überprüfenden Teils geladen und auf die gleiche Weise vorverarbeitet wie die Bilder, die für das Training des Modells verwendet wurden. Dann wird das vorverarbeitete Bild dem Modell zur Vorhersage vorgelegt.

```
In [ ]: #Anwendung des trainierten Modells auf Testdatensätze
test_cases = ['ok_front/cast_ok_0_10.jpeg', 'ok_front/cast_ok_0_1026.jpeg',
              'ok_front/cast_ok_0_1031.jpeg', 'ok_front/cast_ok_0_1121.jpeg',
              'ok_front/cast_ok_0_1144.jpeg','def_front/cast_def_0_1059.jpeg',
              'def_front/cast_def_0_108.jpeg', 'def_front/cast_def_0_1153.jpeg',
              'def_front/cast_def_0_1238.jpeg', 'def_front/cast_def_0_1269.jpeg']

plt.figure(figsize=(20,8))
for i in range(len(test_cases)):
    img_pred = cv2.imread(test_path + test_cases[i], cv2.IMREAD_GRAYSCALE)
    img_pred = img_pred / 255 # rescale
    prediction = model.predict(img_pred.reshape(1, *image_shape))

    img = cv2.imread(test_path + test_cases[i])
    label = test_cases[i].split("_")[0]

    plt.subplot(2, 5, i+1)
    plt.title(f"{test_cases[i].split('/')[1]}\n Actual Label : {label}", weight='bold', size=12)
    # Predicted Class : defect
    if (prediction < 0.5):
        predicted_label = "def"
        prob = (1-prediction.sum()) * 100
    # Predicted Class : OK
    else:
        predicted_label = "ok"
        prob = prediction.sum() * 100

    cv2.putText(img=img, text=f"Predicted Label : {predicted_label}",
                org=(10, 30), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
                fontScale=0.8, color=(255, 0, 255), thickness=2)

    cv2.putText(img=img, text=f"Probability : {'{:.3f}'.format(prob)}%",
                org=(10, 280), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.7, color=(0, 255, 0), thickness=2)
    plt.imshow(img,cmap='gray')
    plt.axis('off')

plt.show()
```

Der folgende Programmcode verdeutlicht diesen Prozess. Zuerst wird eine Liste von Testbildern erstellt. Dann wird jedes Bild einzeln geladen, skaliert und an das Modell zur Vorhersage geschickt. Die Vorhersage des Modells wird dann genutzt, um eine Textmeldung auf das ursprüngliche Bild zu schreiben, die die vorhergesagte Klassifikation des Teils und die Wahrscheinlichkeit dieser Klassifikation enthält. Schließlich wird das annotierte Bild zusammen mit dem tatsächlichen Label des Teils angezeigt.

Der Prozess zur Klassifizierung von Bildern mit dem Modell beinhaltet mehrere Schritte. Zunächst wird das Bild mit der OpenCV-Bibliothek in Graustufen konvertiert, da das Modell auf Graustufenbildern trainiert wurde. Das Bild wird dann skaliert, um den Pixelwerten des Bildes eine Reichweite von 0 bis 1 zu geben. Dies entspricht dem Bereich, den das Modell während des Trainings gesehen hat. Anschließend wird das Bild in die richtige Form gebracht, um vom Modell verarbeitet werden zu können. Schließlich wird das vorbereitete Bild an das Modell übergeben, das dann eine Vorhersage trifft.

Dieser Code ermöglicht es, die Vorhersagen des Modells auf eine visuelle und intuitiv verständliche Weise zu prüfen. Die visuelle Präsentation der Ergebnisse kann dabei helfen, das Vertrauen in die Vorhersagen des Modells zu erhöhen und eventuell auftretende Fehler oder Probleme leichter zu identifizieren. Es ist auch ein nützliches Werkzeug für die Kommunikation der Ergebnisse an Personen ohne technischen Hintergrund, da es die abstrakte Vorstellung einer Vorhersage in eine konkrete visuelle Form überführt.

In dem angegebenen Code-Ausschnitt werden verschiedene Schritte unternommen, um Bilder mit Hilfe eines trainierten Modells zu klassifizieren. Der Code repräsentiert die Anwendung des Modells auf einem bestimmten Testdatensatz und demonstriert dabei die Leistung des Modells in einem realen Anwendungsfall.
Nachfolgend eine ausführliche Untersuchung der Code-Segmente:

1. **Definieren der Testdaten**: Hier wird eine Liste **test_cases** mit den Namen der Bilder definiert, die als Testdatensatz verwendet werden. Diese Bilder repräsentieren sowohl normale als auch defekte Produkte und sind somit repräsentativ für den realen Anwendungsfall.

2. **Erstellen einer Plot-Figur**: **plt.figure(figsize=(20,8))** definiert eine neue Plot-Figur mit einer spezifischen Größe, auf der die Ausgabebilder und die entsprechenden Informationen präsentiert werden.

3. **Schleife über die Testdaten**: In der Schleife **for i in range(len(test_cases)):** wird jedes Bild aus der **test_cases**-Liste nacheinander behandelt. Für jedes Bild wird ein spezifisches Verfahren durchlaufen:

   3.1 **Laden und Skalieren des Bildes**: Mit der OpenCV-Funktion **cv2.imread(test_path + test_cases[i], cv2.IMREAD_GRAYSCALE)** wird das Bild geladen. Anschließend wird das Bild skaliert, indem jeder Pixelwert durch 255 geteilt wird, was dazu führt, dass alle Pixelwerte im Bereich von 0 bis 1 liegen.

   3.2 **Vorhersage mit dem Modell**: Die Funktion **model.predict(img_pred.reshape(1, *image_shape))** wird verwendet, um die Vorhersage des Modells für das geladene und skalierte Bild zu erhalten.

   3.3 **Laden des Originalbildes und Extraktion des Labels**: Das Originalbild wird erneut geladen, um es später in der Ausgabe anzeigen zu können. Außerdem wird das tatsächliche Label aus dem Dateinamen extrahiert.

   3.4 **Erstellen eines Subplots und Setzen des Titels**: Mit **plt.subplot(2, 5, i+1)** wird ein neuer Subplot für das aktuelle Bild erstellt (siehe Abbildung 12). Der Titel des Subplots wird festgelegt und enthält den Dateinamen des aktuellen Bildes und das tatsächliche Label.

   3.5 **Interpretation der Modellvorhersage und Berechnung der Wahrscheinlichkeit**: Die Modellvorhersage wird interpretiert und in ein Label übersetzt. Wenn der Vorhersagewert kleiner als 0,5 ist, wird das Bild als defekt klassifiziert, andernfalls als in Ordnung. Die Wahrscheinlichkeit für das vorhergesagte Label wird berechnet und in Prozent ausgedrückt.

   3.6 **Anzeigen des Labels und der Wahrscheinlichkeit auf dem Bild**: Mit **cv2.putText()** werden das vorhergesagte Label und die Wahrscheinlichkeit auf das Bild geschrieben (siehe Abbildung 12).

   3.7 **Anzeigen des Bildes**: Schließlich wird das Bild mit **plt.imshow(img,cmap='gray')** angezeigt und die Achsen werden mit **plt.axis('off')** ausgeblendet.

4. **Anzeigen der vollständigen Figur**: Mit **plt.show()** wird die vollständige Figur mit allen Subplots angezeigt.

Dieser Code liefert ein visuelles Feedback über die Fähigkeit des Modells, zwischen defekten und nicht defekten Teilen zu unterscheiden. Dies ist eine wichtige Phase, um die Leistung des Modells zu bewerten und seine Anwendungsfähigkeit in realen Szenarien zu beurteilen.[57]

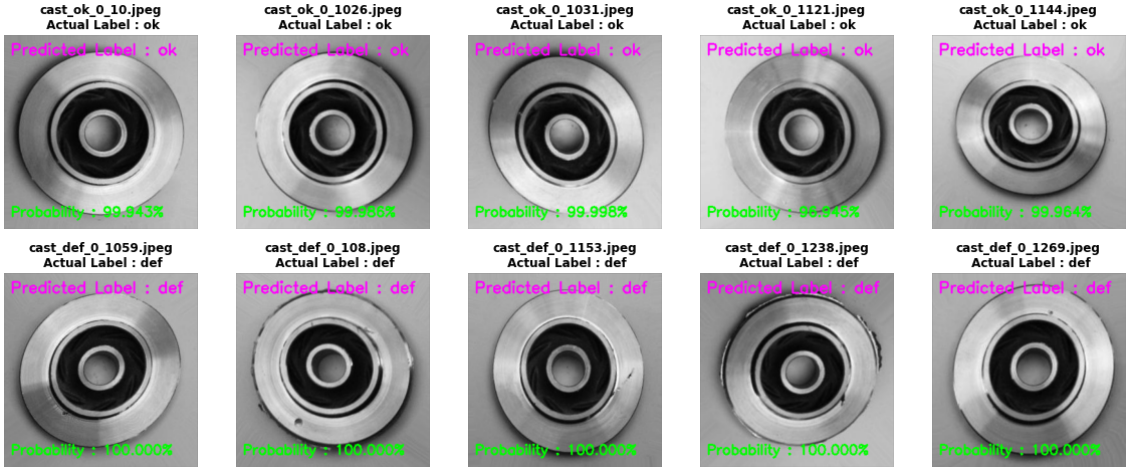


Abbildung 12: Vorhersage des Modells auf Testdatensätzen [Eigene Darstellung]

4.2 Vergleich mit traditionellen Qualitätskontrollmethoden

Traditionelle Qualitätskontrollmethoden in der Fertigung basieren stark auf menschlicher Inspektion und manuellen Tests. Solche Methoden umfassen beispielsweise visuelle Prüfungen, in denen Inspektoren das Endprodukt begutachten, oder komplexere manuelle Tests, bei denen das Produkt auf seine Funktionalität geprüft wird. Ein Hauptmerkmal dieser Methoden ist, dass sie stark von der menschlichen Wahrnehmung und Beurteilung abhängen.

Eine der größten Herausforderungen dieser traditionellen Methoden besteht darin, dass sie in der Regel zeitaufwendig und kostspielig sind. Sie können auch inkonsistent sein, da sie von den Fähigkeiten und der Aufmerksamkeit des jeweiligen Inspektors abhängen. Ein weiterer Nachteil besteht darin, dass sie oft erst am Ende des Produktionsprozesses angewendet werden, wodurch es schwieriger wird, Probleme frühzeitig zu erkennen und zu beheben.[63]

In den letzten Jahren hat die Technologie der Computer Vision einen Paradigmenwechsel in der Qualitätskontrolle eingeleitet. Computer Vision nutzt maschinelles Lernen und künstliche Intelligenz, um Bilder oder Videos zu analysieren und zu interpretieren. In Bezug auf die Qualitätskontrolle können Computer Vision Systeme kontinuierlich und in Echtzeit arbeiten, um Mängel zu erkennen und zu klassifizieren.[62]

Die Evaluierung dieses Modells mit Testdatensätzen hat gezeigt, dass es sehr effektiv ist und nur selten Fehler auftritt. Im Vergleich zu menschlichen Inspektoren können Computer Vision Systeme eine höhere Genauigkeit, Konsistenz und Geschwindigkeit bei der Qualitätskontrolle erreichen. Sie können auch dazu beitragen, die Produktionskosten zu senken und die Produktivität zu steigern, da sie die Notwendigkeit manueller Inspektionen und Tests reduzieren.

Trotz dieser Vorteile haben viele Unternehmen noch nicht auf Computer Vision umgestellt. Eine mögliche Erklärung dafür ist, dass die Implementierung von Computer Vision Systemen technische Expertise und erhebliche Investitionen erfordert. Es kann auch Herausforderungen bei der Integration dieser Systeme in bestehende Produktionslinien geben. Darüber hinaus können Datenschutz- und Sicherheitsbedenken bestehen, insbesondere wenn diese Systeme mit dem Internet verbunden sind. [44]

Unternehmen, die Computer Vision implementieren möchten, müssen mehrere Faktoren berücksichtigen. Dazu gehören die Auswahl des richtigen Systems, das auf ihre spezifischen Anforderungen zugeschnitten ist, die Bereitstellung der notwendigen Hardware und Software, die Schulung der Mitarbeiter und die Sicherstellung der Datensicherheit und -privatsphäre. Darüber hinaus sollten sie sich über die rechtlichen und ethischen Aspekte der Verwendung solcher Systeme im Klaren sein.

Trotz ihrer beeindruckenden Fähigkeiten haben Computer Vision Systeme ihre Grenzen. Sie können beispielsweise Schwierigkeiten haben, subtile oder komplexe Defekte zu erkennen, oder sie können durch Umweltfaktoren wie schlechte Beleuchtung oder Hintergrundrauschen beeinträchtigt werden. Zudem können sie sich in Situationen, die außerhalb ihrer Trainingsdaten liegen, als unzuverlässig erweisen.[43]

Die visuelle Qualitätskontrolle hingegen stellt einen integralen Bestandteil des Produktionsprozesses dar, und seit jeher wurde der menschliche Sehsinn als zuverlässiges Instrument in diesem Bereich betrachtet. Dabei ist die visuelle Inspektion im Vergleich zu anderen Kontrollverfahren relativ einfach durchzuführen, da sie keine spezialisierte technische Ausrüstung erfordert. Die Herausforderung hierbei ist jedoch die beschränkte Zuverlässigkeit des menschlichen Beurteilungsvermögens, das durch diverse Faktoren, einschließlich ergonomischer, beeinflusst wird. Untersuchungen zeigen, dass die Effizienz der visuellen Inspektion sowohl durch unabhängige Faktoren als auch durch menschenbezogene Faktoren beeinflusst wird. Diese Faktoren lassen sich in fünf Kategorien

unterteilen: Technische, psychophysische, organisatorische, arbeitsplatzbezogene und soziale Faktoren.[2]

Technische Faktoren beziehen sich auf die praktische Durchführung der visuellen Inspektion im Produktionsprozess und können beispielsweise die tatsächliche Qualitätsstufe, die inspektierten Produkteigenschaften, Standards, die während der Inspektion genutzten Werkzeuge und weitere Aspekte umfassen. Psychophysische Faktoren beziehen sich auf die mentalen und physischen Zustände der Inspektoren, wie Alter, Geschlecht, Intelligenz, Temperament und Gesundheitszustand. Organisatorische Faktoren betreffen die Unterstützung bei Entscheidungsfindungen während der Inspektion, die Kompetenzentwicklung der Inspektoren, die Anzahl und Art der Inspektionen, Informationen zur Effizienz und Genauigkeit der durchgeführten Inspektionen sowie Stressfaktoren. Arbeitsplatzbezogene Faktoren beziehen sich auf die physischen Bedingungen des Arbeitsplatzes, einschließlich Licht, Lärm, Temperatur und Arbeitsplatzorganisation. Soziale Faktoren umfassen den Druck und die Erwartungen von Kollegen und Vorgesetzten, die möglicherweise im Widerspruch zur Arbeit des Inspektors stehen.[44]

Die Unzulänglichkeiten der menschlichen visuellen Kontrolle wurden schon in den 1950er bis 1970er Jahren erkannt, was zu Bestrebungen führte, den Menschen in der Qualitätskontrolle durch Maschinen zu ersetzen. Hierdurch wurden automatisierte Visionssysteme entwickelt, welche allerdings in der praktischen Umsetzung auf Schwierigkeiten stießen. Diese Systeme zeigten insbesondere bei neuen oder nicht standardisierten Situationen, die bei der Systementwicklung nicht berücksichtigt wurden, eine eingeschränkte Flexibilität. Folglich bleibt menschliches Urteilsvermögen bei der visuellen Inspektion in vielen Produktionsprozessen unverzichtbar.[64]

Trotz ihrer Schwächen, wie etwa einer gewissen Fehleranfälligkeit, bietet die visuelle Inspektion durch Menschen den Vorteil, dass sie wirtschaftlich attraktiv und nicht-destruktiv ist. Das bedeutet, dass sie nicht zu Verschleiß des geprüften Produkts führt. Allerdings haben Untersuchungen gezeigt, dass der menschliche Fehleranteil in der visuellen Inspektion zwischen 3-10% liegen kann. Dies unterstreicht, dass menschliche Inspektion nicht eine hundertprozentige korrekte Beurteilung garantieren kann.[65]

Die Computer Vision bietet in dieser Hinsicht einige Vorteile. Sie bietet eine gesteigerte Präzision, ermöglicht eine schnellere Inspektion, gewährleistet Konsistenz und senkt die Kosten durch Automatisierung. Computer Vision ermöglicht auch Echtzeitüberwachung

und Feedback, effiziente Verarbeitung großer Datenmengen, erhöhte Arbeitssicherheit und datenbasierte Entscheidungsfindung. Zudem können moderne Systeme flexibel angepasst und in vielfältigen Produktionsumgebungen eingesetzt werden.[45]

Trotz dieser Vorteile bringt die Implementierung von Computer Vision auch Herausforderungen mit sich. Dazu zählen hohe Anfangsinvestitionen, die Komplexität der Einrichtung und Wartung, technologische Limitierungen und datenschutzrechtliche sowie ethische Bedenken. Zudem kann die Einführung von Computer Vision eine Mitarbeiterweiterbildung erforderlich machen und Herausforderungen in Bezug auf die Integration in bestehende Systeme mit sich bringen. Eine Anpassung an Änderungen kann schwierig sein, ebenso wie das Fehlen von Standards. Schließlich hängt die Leistung von Computer Vision-Systemen stark von der Qualität der Eingabedaten ab.[42]

Eine Alternative bietet der hybride Ansatz, der Mensch und Maschine kombiniert. Hierbei wurde festgestellt, dass die Kombination aus Mensch und Maschine in Bezug auf die Kosten-Nutzen-Relation die optimale Lösung darstellt. Insbesondere in der Entscheidungsfindungsphase, in der das Produkt einer bestimmten Kategorie (beispielsweise "gut" oder "fehlerhaft") zugeordnet wird, ist die menschliche Beteiligung von entscheidender Bedeutung.[66]

In einer Diskussion über die Vor- und Nachteile von Computer Vision gegenüber menschlicher Kontrolle zeigt sich, dass beide Methoden ihre jeweiligen Stärken und Schwächen aufweisen. Obwohl Computer Vision in vielen Bereichen der Qualitätssicherung Vorteile bietet, gibt es auch Bereiche, in denen menschliche Inspektoren aufgrund ihrer Flexibilität, Anpassungsfähigkeit und Fähigkeit zur Interpretation von Kontextinformationen einen Vorteil haben könnten. Daher scheint ein ausgewogener Ansatz, der die Vorteile beider Methoden nutzt, der effektivste Weg zu sein. Mit der kontinuierlichen Entwicklung und Verbesserung von Technologien im Bereich der Computer Vision ist jedoch zu erwarten, dass ihre Anwendungen und Vorteile in der Zukunft weiter zunehmen werden.

5 Fazit und Ausblick

5.1 Zusammenfassung der Ergebnisse

Die vorliegende Bachelorarbeit befasst sich mit dem Thema "Einsatz von Deep Learning für die Qualitätskontrolle in der Fertigung". Die zentrale Aussage, die sich aus den analysierten Daten und den vorhergehenden Kapiteln ableiten lässt, ist, dass die Integration von Künstlicher Intelligenz (KI) und Deep Learning in die Qualitätssicherung und -kontrolle das Potenzial hat, die industrielle Fertigung signifikant zu verbessern.

Im zweiten Kapitel wurde Qualität als dynamisches Konzept dargestellt, das sich im Laufe der Zeit von der reinen Produkt- und Prozesskonformität zu einer stärkeren Kundenorientierung und einer umfassenderen Betrachtung der Bedürfnisse der Stakeholder weiterentwickelt hat. Dabei wurden sowohl die entscheidende Rolle der Qualitätssicherung und -kontrolle in der Fertigungsindustrie als auch die damit verbundenen Herausforderungen und Anforderungen herausgestellt.[16]

Des Weiteren erfolgte in Kapitel 2 eine Einführung in die Konzepte der Künstlichen Intelligenz und des Machine Learnings, wobei ein besonderer Fokus auf dem Deep Learning lag. Es wurde erläutert, dass Deep Learning auf neuronalen Netzen basiert und besonders effektiv für die Erkennung von Mustern in großen und komplexen Datenmengen ist. Zudem wurde auf die Bedeutung verschiedener Lernmethoden wie dem Unsupervised Learning für die Identifikation von Mustern und Anomalien in Fertigungsdaten eingegangen.[30]

Die spezifischen Anwendungen von Deep Learning in der Qualitätskontrolle in der Fertigung wurden im weiteren Verlauf des zweiten Kapitels behandelt. Hier wurden die Möglichkeiten und Vorteile von Deep Learning für die automatisierte Qualitätskontrolle diskutiert, einschließlich erhöhter Genauigkeit, Geschwindigkeit, Konsistenz und Effizienz, sowie der Fähigkeit, große Datenmengen zu verarbeiten und wertvolle Informationen zur Verbesserung der Produktqualität und Effizienz bereitzustellen.[59]

Das Gebiet der Computer Vision wurde als spezifische Anwendung von KI und Deep Learning in der Fertigungsqualitätskontrolle im zweiten Kapitel dargestellt. Es wurde aufgezeigt, dass Computer Vision eine Vielzahl von Vorteilen für die Qualitätskontrolle in der Fertigung bietet, darunter erhöhte Präzision, Geschwindigkeit, Konsistenz, Kostenreduktion, Echtzeit-Feedback und die Möglichkeit, datengesteuerte Entscheidungen zu treffen. Trotz potenzieller Herausforderungen, wie beispielsweise erheblichen

Anfangsinvestitionen und technologischen Einschränkungen, kann Computer Vision bei sorgfältiger Implementierung und Wartung ein effektives Instrument für die Qualitätskontrolle sein.[43]

Im dritten Kapitel wurde die praktische Implementierung eines Deep Learning-Modells auf der Basis der Programmiersprache Python und der Nutzung diverser Bibliotheken für die Datenverarbeitung, die Modellbildung und -evaluation sowie die Datenanalyse und -visualisierung diskutiert. Ein besonderes Augenmerk lag dabei auf der Verwendung der Keras-Bibliothek zur Datenvorbereitung und zur Implementierung des Modells. Die Anwendung von Datenaugmentation zur Erweiterung der Trainingsdaten und die Verwendung eines künstlichen neuronalen Netzwerks mit Convolutional Neural Network (CNN) Architektur wurden hervorgehoben.[49]

Im vierten Kapitel wurden die Details und Ergebnisse eines spezifischen Deep Learning-Modells zur Fehlererkennung in industriellen Produkten vorgestellt. Es wurde nachgewiesen, dass dieses Modell auf den Testdaten effektiv arbeitet und eine hohe Genauigkeit in der Unterscheidung zwischen fehlerhaften und fehlerfreien Teilen aufweist.

Zusammenfassend zeigt die vorliegende Arbeit, dass der Einsatz von Deep Learning in der Qualitätskontrolle eine innovative und wirksame Methode darstellt, um die Effizienz und Genauigkeit in der Fertigung zu steigern. Sie beleuchtet sowohl die Möglichkeiten als auch die Herausforderungen, die mit der Einführung dieser Technologien verbunden sind.

Es wird deutlich, dass Deep Learning ein enormes Potenzial für die Verbesserung der Qualitätskontrolle in der Fertigungsindustrie bietet. Gleichzeitig wird jedoch betont, dass eine erfolgreiche Integration von Deep Learning in die Qualitätskontrolle nicht nur technisches Know-how und eine geeignete Infrastruktur erfordert, sondern auch eine Kultur des Lernens und der Anpassungsfähigkeit. Es wird argumentiert, dass ein hybrider Ansatz, der traditionelle Methoden der Qualitätskontrolle mit den innovativen Möglichkeiten von Deep Learning verbindet, einen effektiven Weg darstellen kann, um die Vorteile beider Ansätze optimal zu nutzen.[19]

Zukünftige Forschungen und praktische Anwendungen sind notwendig, um die tatsächlichen Auswirkungen von Deep Learning auf die Qualitätskontrolle in der Fertigung umfassender zu verstehen und seine Anwendungsmöglichkeiten weiter auszubauen. Es wird

darauf hingewiesen, dass eine kontinuierliche Evaluation und Anpassung des Modells erforderlich ist, um den technologischen Fortschritt und die sich verändernden Anforderungen der Fertigungsindustrie zu berücksichtigen.

Mit Blick in die Zukunft wird prognostiziert, dass die Rolle von Künstlicher Intelligenz und insbesondere von Deep Learning in der Fertigungsqualitätskontrolle weiter zunehmen wird. Es bleibt spannend zu beobachten, wie sich diese Technologien weiterentwickeln und wie sie die Qualitätskontrollprozesse in der Fertigung transformieren werden.

5.2 Ausblick für zukünftige Forschung

Im Folgenden wird der Blick auf mögliche zukünftige Entwicklungen und Untersuchungen im Bereich der Anwendung von künstlicher Intelligenz und Deep Learning in der Qualitätskontrolle gerichtet. Diese Bachelorarbeit hat die Potenziale und Herausforderungen dieser neuartigen Technologien in der Qualitätskontrolle aufgezeigt, doch das Feld ist dynamisch und wird weiterhin rasant voranschreiten.

Zukünftige Forschungen könnten sich auf eine weitergehende Verbesserung der Modellleistung konzentrieren. Es besteht Potenzial für die Untersuchung verschiedener Architekturen neuronaler Netze und die Integration neuer Techniken aus dem sich schnell entwickelnden Bereich des Deep Learning. Insbesondere die Anwendung fortgeschrittener Techniken wie Transfer Learning oder die Nutzung von Generative Adversarial Networks (GANs) könnte weiter erforscht werden.

Darüber hinaus sollte eine intensivere Erforschung der Implementierung solcher Systeme in der Praxis in Betracht gezogen werden. Es besteht Bedarf an einer tiefergehenden Untersuchung der organisatorischen und technischen Herausforderungen, die mit der Integration von KI-basierten Systemen in bestehende Fertigungsumgebungen einhergehen. Eine weitere zentrale Forschungsrichtung könnte die Untersuchung von Methoden zur Überwindung von Datenbeschränkungen sein. Da Deep Learning-Modelle stark von der Verfügbarkeit großer Mengen hochwertiger Trainingsdaten abhängen, könnten Techniken zur Datenaugmentation oder semi-supervisiertes Lernen von besonderem Interesse sein.

Insgesamt bieten die Ergebnisse dieser Arbeit vielfältige Ansatzpunkte für weitere Untersuchungen. Die Rolle von KI und Deep Learning in der Qualitätskontrolle ist ein spannendes und vielversprechendes Forschungsfeld, dessen volles Potenzial noch längst nicht

ausgeschöpft ist. Es wird spannend zu beobachten sein, wie sich die Technologien weiterentwickeln und welche Auswirkungen dies auf die Qualitätskontrolle in der Fertigung haben wird.

Literaturverzeichnis

[1] K-Zeitung, „Bekannte Fertigung grundlegend verändern“, 31. August 2017. https://www.k-zeitung.de/bekannte-fertigung-grundlegend-veraendern (zugegriffen 15. April 2023).

[2] B. Jung, S. Schweißer, und J. Wappis, *Qualitätssicherung im Produktionsprozess*, 2. Auflage. in Pocket Power, no. 066. München: Hanser, 2021.

[3] Katana, „Cost of poor quality in manufacturing: how to cut out the losses“, *Cost of poor quality (COPQ) is an essential accounting formula for calculating losses from poor quality products and services.*, 4. April 2023. https://katanamrp.com/blog/cost-of-poor-quality-in-manufacturing/#:~:text=Every%20average%20manufacturing%20company%20has,20%25%20of%20the%20total%20sales. (zugegriffen 20. April 2023).

[4] W. Sihn, A. Sunk, T. Nemeth, P. Kuhlang, und K. Matyas, *Produktion und Qualität: Organisation, Management, Prozesse*. in Praxisreihe Qualitätswissen. München: Hanser, 2016.

[5] M. Hassaballah und A. I. Awad, Hrsg., *Deep learning in computer vision: principles and applications*. Boca Raton: CRC Press, Taylor and Francis, 2020.

[6] A. Lynn, „AI in manufacturing market to reach $16bn by 2025“, *Global Market Insights*, 21. Mai 2019. https://www.electronicspecifier.com/news/analysis/ai-in-manufacturing-market-to-reach-16bn-by-2025 (zugegriffen 23. April 2023).

[7] J. Schwarze, *Kundenorientiertes Qualitätsmanagement in der Automobilindustrie*, Gabler edition Wissenschaft. Wiesbaden: Deutscher Universitätsverlag, 2003.

[8] D. Berndt, N. Bauer, und Fraunhofer-Allianz Vision, Hrsg., *Leitfaden zur praktischen Anwendung der Bildverarbeitung*. in Vision, no. 5. Erlangen: Fraunhofer-Allianz Vision, 2002.

[9] H.-D. Zollondz, *Grundlagen Qualitätsmanagement: Einführung in Geschichte, Begriffe, Systeme und Konzepte*, 2., Vollst. überarb. und erw. Aufl. Berlin: Oldenbourg Wissenschaftsverlag, 2009.

[10] W. A. Shewhart und W. E. Deming, *Statistical method from the viewpoint of quality control*. New York: Dover, 1986.

[11] K. Trimmer, T. Newman, und F. F. Padró, Hrsg., *Ensuring Quality in Professional Education Volume II: Engineering Pedagogy and International Knowledge Structures*, 1st ed. 2019. Cham: Springer International Publishing : Imprint: Palgrave Macmillan, 2019. doi: 10.1007/978-3-030-01084-3.

[12] J. Rothlauf, *Total quality management in theorie und praxis: Zum ganzheitlichen Unternehmensverständnis*. Berlin: De Gruyter, 2014.

[13] G. Freisinger, O. Jöbstl, B. Kögler, J. Lipp, und M. Strohrmann, *Die digitale Transformation des Qualitätsmanagements: Potenziale nutzen, Strategien entwickeln, Qualität optimieren*. München: Hanser, 2022.

[14] A. A. W. Scheibeler und F. Scheibeler, *Easy ISO 9001:2015 für kleine Unternehmen*, 2. Auflage. München: Carl Hanser Verlag GmbH & Co. KG, 2019.

[15] D. A. Garvin, „Competing on the Eight Dimensions of Quality“. Harvard Business School, 1987.

[16] G. F. Kamiske, *Qualitätssicherung - Praxiswissen*. München: Hanser, 2015.

[17] T. Spocker, *Roadmap zur Implementierung eines Qualitätsmanagementsystems nach DIN EN ISO 9001:2015*, 1. Auflage, Digitale Originalausgabe. München: GRIN Verlag, 2019.

[18] K. Pichhardt, *Qualitätssicherung Lebensmittel: präventives und operatives Qualitätsmanagement vom Rohstoff bis zum Fertigprodukt: mit 20 Abbildungen*. Berlin Heidelberg: Springer, 2012.

[19] G. Conrads, *Integration der Qualitätsbeurteilung in die Fertigungstätigkeit*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[20] D. Hofmann, *Handbuch Meßtechnik und Qualitätssicherung*. Wiesbaden: Vieweg+Teubner Verlag : Imprint : Vieweg+Teubner Verlag, 2013.

[21] I. Zeilhofer-Ficker, *Null-Fehler-Fertigung Qualität nicht kontrollieren sondern steuern*. München: GBI-Genios Verlag, 2008.

[22] P. Wennker, *Künstliche Intelligenz in der Praxis: Anwendungen in Unternehmen und Branchen: KI wettbewerbs- und zukunftsorientiert einsetzen*. Wiesbaden, Germany [Heidelberg]: Springer Gabler, 2020. doi: 10.1007/978-3-658-30480-5.

[23] N. Stanev, *Machine-Learning-Anwendungen in mittelständischen Industrieunternehmen. Einsatzbereiche und Erfolgsfaktoren*. S.l.: GRIN VERLAG, 2023.

[24] A. W. Trask, *Neuronale Netze und Deep Learning kapieren: der einfache Praxiseinstieg mit Beispielen in Python*, 1. Auflage 2020. Frechen: mitp, 2020.

[25] N. J. Nilsson, R. Hinrichs, und N. J. Nilsson, *Die Suche nach künstlicher Intelligenz: eine Geschichte von Ideen und Erfolgen*. Berlin: AKA, Akademische Verlagsgesellschaft, 2014.

[26] T. Cole, *Erfolgsfaktor künstliche Intelligenz: KI in der Unternehmenspraxis: Potenziale erkennen - Entscheidungen treffen*. München: Hanser, 2020.

[27] A. Bresges und A. Habicher, Hrsg., *Digitalisierung des Bildungssystems: Aufgaben und Perspektiven für die LehrerInnenbildung*. in LehrerInnenbildung gestalten, no. Band 12. Münster New York: Waxmann, 2019.

[28] S. Li, Y.-Q. Deng, Z.-L. Zhu, H.-L. Hua, und Z.-Z. Tao, „A Comprehensive Review on Radiomics and Deep Learning for Nasopharyngeal Carcinoma Imaging“, *Diagnostics*, Bd. 11, Nr. 9, S. 1523, Aug. 2021, doi: 10.3390/diagnostics11091523.

[29] B. K. Mishra und R. Kumar, Hrsg., *Natural language processing in artificial intelligence*. Burlington, ON, Canada: Apple Academic Press, 2021.

[30] T. Jo, *Machine learning foundations: supervised, unsupervised, and advanced learning*. Cham, Switzerland: Springer, 2021.

[31] O. Chapelle, B. Schölkopf, und A. Zien, Hrsg., *Semi-supervised learning*, 1st MIT Press pbk. ed. in Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2010.

[32] A. Zai und B. Brown, *Einstieg in Deep Reinforcement Learning: KI-Agenten mit Python und PyTorch programmieren*. München: Hanser, 2020.

[33] C. N. Nguyen und O. Zeigermann, *Machine Learning - kurz & gut*, 2. Auflage. in kurz & gut. Heidelberg: O'Reilly, 2021.

[34] J. Krohn, G. Beyleveld, und A. Bassens, *Deep Learning illustriert: eine anschauliche Einführung in Machine Vision, Natural Language Processing und Bilderzeugung für Programmierer und Datenanalysten*. Heidelberg: dpunkt.verlag, 2020.

[35] L. Mertens, „Künstliche neuronale Netze am Beispiel der Klassifizierung von Scandaten". 10. Juni 2018.

[36] S. Raschka und V. Mirjalili, *Machine learning mit Python und Keras, TensorFlow 2 und Scikit-learn: das umfassende Praxis-Handbuch für data science, deep learning und predictive analytics*, 3., Aktualisierte und Erweiterte Auflage. Frechen: mitp, 2021.

[37] W.-M. Lippe, *Soft-Computing: mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.

[38] R. Rojas, *Theorie der neuronalen Netze: Eine systematische Einführung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[39] *Neuronale Netze Eine Einführung in die Neuroinformatik*, 2., Überarbeitete und Erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag, 2013.

[40] M. Pomplun und J. Suzuki, Hrsg., *Developing and applying biologically-inspired vision systems: interdisciplinary concepts*. Hershey, PA: Information, Science, Reference, 2013.

[41] L. Priese, *Computer Vision: Einführung in die Verarbeitung und Analyse digitaler Bilder*. in eXamen.press. Berlin [u.a.]: Springer Vieweg, 2015.

[42] H. Süsse und E. Rodner, *Bildverarbeitung und Objekterkennung: Computer Vision in Industrie und Medizin*. in Bildverarbeitung und Objekterkennung. Wiesbaden: Springer Vieweg, 2014.

[43] C. Brown und C. Brown, *Advances in Computer Vision: Volume 1*. Hoboken: Taylor and Francis, 2014.

[44] R. Kashyap und A. V. S. Kumar, Hrsg., *Challenges and applications for implementing machine learning in computer vision*. in IGI global disseminator of knowledge. Hershey, PA: IGI Global/Engineering Science Reference, an imprint of IGI Global, 2020.

[45] E. R. Davies, *Computer vision: principles, algorithms, applications, learning*, Fifth edition. London: Academic Press, 2018.

[46] C. H. Chen, *Emerging topics in computer vision and its applications*. Singapore: World Scientific, 2012.

[47] J. Schöttner, *Umsatz gut, Rendite mangelhaft: das Kostenproblem der Fertigungsindustrie: warum IT, Digitalisierung, PLM & Co. allein nichts ändern - Ursachen und Lösungen*. München: Hanser, 2017.

[48] I. Bode, *Guss-Produkte Jahreshandbuch der Gießereine, Zulieferer, Ausstatter*. Darmstadt [u.a.]: Hoppenstedt, 1998.

[49] M. R. Karim, M. Sewak, und P. Pujari, *Practical Convolutional Neural Networks: Implement advanced deep learning models using Python*. Birmingham: Packt Publishing, 2018.

[50] R. Dabhi, „casting product image data for quality inspection“. 2020. Zugegriffen: 22. April 2023. [Online]. Verfügbar unter: https://www.kaggle.com/account/login?titleType=dataset-downloads&showDatasetDownloadSkip=False&messageId=datasetsWelcome&returnUrl=%2Fdatasets%2Fravirajsinh45%2Freal-life-industrial-dataset-of-casting-product%2Fversions%2F2%3Fresource%3Ddownload

[51] F. Bittmann, *Praxishandbuch Python 3 Konzepte der Programmierung verstehen und anwenden*. Norderstedt: Books on Demand, 2020.

[52] M. Liu, *Make Python talk: build apps with voice control and speech recognition*. San Francisco, CA: No Starch Press, 2021.

[53] A. Géron, *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme*, 2. Auflage. Heidelberg: O'Reilly, 2020.

[54] B. Klein, *Numerisches Python: arbeiten mit NumPy, Matplotlib und Pandas*. München: Hanser, 2019.

[55] M. Harrison, *Machine learning: die Referenz: mit strukturierten Daten in Python arbeiten*, 1. Auflage, Deutsche Ausgabe. Heidelberg: O'Reilly, 2021.

[56] L. Ramalho, *Fluent Python*, First edition. Sebastopol, CA: O'Reilly, 2015.

[57] J. Brownlee, *Deep Learning for Computer Vision. Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019.

[58] F. Chollet, *Deep Learning mit Python und Keras: das Praxis-Handbuch: vom Entwickler der Keras-Bibliothek*, 1. Auflage. Frechen: mitp, 2018.

[59] M. Moocarme und R. Bhagwat, *The deep learning with Keras workshop: learn how to define and train neural network models with just a few lines of code*, Third edition. Birmingham: Packt Publishing, 2020.

[60] Y. Hasija und R. Chakraborty, *Hands on Data Science for Biologists Using Python*. Milton: Taylor & Francis Group, 2021.

[61] L. Vaughan, *Python tools for scientists: an introduction to coding, anaconda, jupyterlab, and the scientific libraries*. San Francisco: No Starch Press, Inc, 2023.

[62] S. K. Singh, P. P. Roy, B. Raman, und P. Nagabhushan, Hrsg., *Computer vision and image processing. Part 1*. in Communications in computer and information science, no. 1376. Singapore: Springer, 2021.

[63] D. Adam, *Produktions-Management*, 9., Überarbeitete Auflage. Wiesbaden: Gabler Verlag : Imprint : Gabler Verlag, 2013.

[64] M. Schulze, *Anpassbare Prozessmodelle in Verfahren zur Qualitätssicherung technischer Produktionsprozesse*. Berlin: Logos-Verl, 2003.

[65] W. Ellouze, *Entwicklung eines Modells für ein ganzheitliches Fehlermanagement: ein prozessorientiertes Referenzmodell zum effizienten Fehlermanagement*. in Berichte aus der Fertigungstechnik. Aachen: Shaker-Verl, 2010.

[66] R. Müller, J. Franke, D. Henrich, B. Kuhlenkötter, A. Raatz, und A. Verl, *Handbuch Mensch-Roboter-Kollaboration*. München: Carl Hanser Verlag GmbH & Co. KG, 2019.

Eidesstattliche Erklärung

Hiermit versichere ich eidesstattlich, dass ich die vorliegende Bachelorarbeit eigenständig und ohne unzulässige Hilfe Dritter verfasst habe. Ich habe ausschließlich die von mir angegebenen Quellen verwendet. Jede direkte oder indirekte Bezugnahme auf fremde Werke ist als solche kenntlich gemacht. Des Weiteren ist diese Arbeit bisher weder in Teilen noch vollständig veröffentlicht worden. Außerdem habe ich die vorgelegte Arbeit bisher weder teilweise noch vollständig im Rahmen einer anderen Prüfungsleistung eingereicht.

Hamburg, den 31.05.2023 Shahrukh Khan