

**BACHELORARBEIT**

# **Contrastive Learning mit Stable Diffusion-basierter Datenaugmentation**

Verbesserung der Bildklassifikation  
durch synthetische Daten

---

vorgelegt am 16. September 2024  
Paul Hofmann

Erstprüferin: Prof. Dr. Larissa Putzar  
Zweitprüfer: Prof. Dr. Jan Neuhöfer

---

**HOCHSCHULE FÜR ANGEWANDTE  
WISSENSCHAFTEN HAMBURG**  
Department Medientechnik  
Finkenau 35  
22081 Hamburg

## **Zusammenfassung**

Der Arbeit beginnt mit einer kurzen Beschreibung ihrer zentralen Inhalte, in der die Thematik und die wesentlichen Resultate skizziert werden. Diese Beschreibung muss sowohl in deutscher als auch in englischer Sprache vorliegen und sollte eine Länge von etwa 150 bis 250 Wörtern haben. Beide Versionen zusammen sollten nicht mehr als eine Seite umfassen. Die Zusammenfassung dient u. a. der inhaltlichen Verortung im Bibliothekskatalog.

## **Abstract**

The thesis begins with a brief summary of its main contents, outlining the subject matter and the essential findings. This summary must be provided in German and in English and should range from 150 to 250 words in length. Both versions combined should not comprise more than one page. Among other things, the abstract is used for library classification.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	1
1.3 Vorarbeit im Praktikum . . . . .	1
1.4 Aufbau der Arbeit . . . . .	1
<b>2 Theoretische Grundlagen</b>	<b>2</b>
2.1 Maschinelles Lernen . . . . .	2
2.1.1 Überwachtes und unüberwachtes Lernen . . . . .	3
2.1.2 Deep Learning . . . . .	3
2.1.3 Neuronale Netze . . . . .	4
2.1.4 Convolutional Neural Networks . . . . .	5
2.1.5 Datenaugmentation . . . . .	5
2.1.6 Out-of-Distribution Daten . . . . .	6
2.2 Synthetische Daten . . . . .	6
2.2.1 Variational Autoencoder . . . . .	6
2.2.2 Generative Adversarial Networks . . . . .	6
2.2.3 Stable Diffusion . . . . .	7
2.2.4 DA-Fusion . . . . .	7
2.3 Contrastive Learning . . . . .	7
2.3.1 Unsupervised Contrastive Learning . . . . .	7
2.3.2 Supervised Contrastive Learning . . . . .	7
2.4 Integration von DA-Fusion und Supervised Contrastive Learning . . . . .	7
2.4.1 Forschungslücke . . . . .	8
2.4.2 Near Out-of-Distribution Augmentationen als Hard Negatives . . . . .	8
2.4.3 Potenzielle Vorteile und Herausforderungen . . . . .	8
<b>3 Methodisches Vorgehen</b>	<b>9</b>
3.1 Forschungsfragen und Hypothesen . . . . .	9

3.2	Datensatz . . . . .	9
3.2.1	EIBA . . . . .	10
3.2.2	Teildatensatz . . . . .	10
3.2.3	Vorverarbeitung . . . . .	10
3.3	Implementierung . . . . .	10
3.3.1	DA-Fusion . . . . .	11
3.3.2	Supervised Contrastive Learning . . . . .	11
3.4	Synthetische Datengenerierung mit DA-Fusion . . . . .	12
3.5	Trainings- und Testdurchläufe mit Supervised Contrastive Learning . . . . .	12
3.6	Evaluationsmethoden und Metriken . . . . .	12
<b>4</b>	<b>Ergebnisse</b>	<b>13</b>
4.1	Die generierten synthetischen Daten . . . . .	13
4.1.1	In-Distribution . . . . .	13
4.1.2	Near Out-of-Distribution . . . . .	13
4.2	Trainings- und Testergebnisse mit Supervised Contrastive Learning . . . . .	13
4.2.1	Contrastive Pre-Training . . . . .	13
4.2.2	Lineare Klassifikation . . . . .	14
4.3	Vergleich der Ergebnisse mit und ohne In-Distribution-Augmentationen . . . . .	14
4.4	Vergleich der Ergebnisse mit und ohne Near Out-of-Distribution-Augmentationen als Hard Negatives . . . . .	14
<b>5</b>	<b>Diskussion</b>	<b>15</b>
5.1	Eignung von DA-Fusion für die synthetische Datengenerierung . . . . .	15
5.2	Wirksamkeit von Near Out-of-Distribution-Daten als Hard Negatives im Supervised Contrastive Learning . . . . .	15
<b>6</b>	<b>Fazit</b>	<b>16</b>
6.1	Zusammenfassung der wichtigsten Erkenntnisse . . . . .	16
6.2	Beantwortung der Forschungsfragen . . . . .	16
6.3	Ausblick und potenzielle Weiterentwicklungen . . . . .	16
	<b>Literatur</b>	<b>17</b>
	<b>Anhang</b>	<b>18</b>

# **Abbildungsverzeichnis**

# **Tabellenverzeichnis**

# **1 Einleitung**

## **1.1 Motivation**

...

## **1.2 Zielsetzung**

...

## **1.3 Vorarbeit im Praktikum**

...

## **1.4 Aufbau der Arbeit**

...

## 2 Theoretische Grundlagen

Im folgenden Kapitel werden die theoretischen Grundlagen des maschinellen Lernens und der verwendeten Modelle erläutert. Es wird auf die Konzepte des maschinellen Lernens, insbesondere des überwachten und unüberwachten Lernens, des Deep Learnings und der neuronalen Netze eingegangen. Anschließend wird die Funktionsweise von Diffusion-Modellen, insbesondere Stable Diffusion und DA-Fusion, sowie von Contrastive Learning und Supervised Contrastive Learning beschrieben. Zuletzt wird die bestehende Forschungslücke und die in dieser Arbeit thematisierte Integration von DA-Fusion und Supervised Contrastive Learning diskutiert.

### 2.1 Maschinelles Lernen

Die ersten großen Durchbrüche in der künstlichen Intelligenz (KI) kamen im Bezug auf Aufgaben, die für Menschen intellektuell eine große Herausforderung darstellten, die aber von Computern relativ einfach zu lösen waren, da sie als Liste formaler, mathematischer Regeln beschrieben werden konnten. Die große Schwierigkeit lag hingegen in den Aufgaben, die für Menschen relativ einfach und intuitiv sind, welche sich aber nur schwer formal beschreiben lassen. Hierunter fallen z.B. die Spracherkennung, oder Objekterkennung. (Goodfellow et al., 2016)

Maschinelles Lernen (ML) beschreibt den Ansatz, Computer mit der Fähigkeit auszustatten, selbstständig Wissen aus Erfahrung zu generieren, indem Muster und Konzepte aus rohen Daten erlernt werden. So kann ein Computerprogramm auf Basis von Beispielen lernen, wie es eine bestimmte Aufgabe lösen soll, ohne dass ihm explizit Regeln oder Algorithmen vorgegeben werden.

Eine allgemeine Definition für maschinelles Lernen bietet (Mitchell, 1997):

Ein Computerprogramm soll aus Erfahrung  $E$  in Bezug auf eine Klasse von Aufgaben  $T$  und Leistungsmaß  $P$  lernen, wenn sich seine Leistung bei Aufgaben  $T$ , gemessen durch  $P$ , mit Erfahrung  $E$  verbessert.



Die Erfahrung  $E$  besteht dabei aus einer Menge von Trainingsdaten, die etwa aus Eingabe-Ausgabe-Paaren bestehen. Die Aufgaben  $T$  können sehr vielfältig sein, von einfachen Klassifikations- und Regressionsaufgaben bis hin zu komplexen Problemen wie Spracherkennung oder autonomen Fahren. Das Leistungsmaß  $P$  gibt an, wie gut das Modell die Aufgaben  $T$  löst, und kann z.B. die Genauigkeit (engl. *accuracy*) einer Klassifikation oder die mittlere quadratische Abweichung bei einer Regression sein.

### 2.1.1 Überwachtes und unüberwachtes Lernen

Wie genau Wissen aus Erfahrung bzw. aus Rohdaten generiert wird hängt vom gewählten Verfahren ab. Im Maschinellen Lernen gibt es dabei verschiedene Paradigmen, wobei die wichtigsten das überwachte (engl. *supervised*) und das unüberwachte (engl. *unsupervised*) Lernen sind.

Beim überwachten Lernen wird das Modell mit einem vollständig annotierten Datensatz trainiert. Das heißt, jeder Datenpunkt ist mit einem Klassenlabel versehen, sodass Eingabe-Ausgabe-Paare entstehen. Das Ziel ist es, eine Funktion zu lernen, die Eingaben (Features) auf die entsprechenden Ausgaben (Labels) abbildet. Beispiele für überwachtes Lernen sind Klassifikations- und Regressionsaufgaben. Ein typisches Beispiel ist die Bilderkennung, bei der ein Modell darauf trainiert wird, Bilder von Katzen und Hunden zu unterscheiden.

Im Gegensatz dazu arbeitet unüberwachtes Lernen mit unbeschrifteten Daten; es gibt also keine vorgegebenen Ausgaben. Stattdessen wird versucht, ein Modell zu befähigen, eigenständig Muster und Strukturen in den Daten zu erkennen und z.B. Cluster zu bilden, oder nützliche Repräsentationen der Eingangsdaten zu erstellen. Zu den häufigsten Methoden des unüberwachten Lernens gehören Clustering- und Assoziationsalgorithmen. Ein Beispiel ist die Segmentierung von Kunden in verschiedene Gruppen basierend auf ihrem Kaufverhalten. (**<empty citation>**)

In der Praxis werden oft auch hybride Ansätze genutzt, wie das semi-überwachte Lernen, bei dem eine Kombination aus beschrifteten und unbeschrifteten Daten verwendet wird, oder das selbstüberwachte Lernen, bei dem das Modell sich selbst überwacht, indem es Teile der Daten als pseudo-beschriftet behandelt.

### 2.1.2 Deep Learning

Das Wissen, das ein Modell aus den Trainingsdaten lernt, wird in Form von Merkmalen (engl. *features*) repräsentiert. Diese Merkmale können einfache Konzepte wie Kanten oder Farben sein, oder komplexere Konzepte wie Gesichter oder Objekte. Unter Deep Learning versteht man eine tiefe, hierarchische Vernetzung dieser Konzepte, sodass komplexere Konzepte auf

simpleren Konzepten aufbauen können. Visuell veranschaulicht entsteht ein Graph mit vielen Ebenen (engl. *deep layers*). Damit ist Deep Learning eine spezialisierte Unterkategorie des maschinellen Lernens, die auf künstlichen neuronalen Netzen basiert. Diese Netzwerke bestehen aus mehreren Schichten, die eine hierarchische Repräsentation von Daten ermöglichen. Jede Schicht transformiert die Eingabedaten in eine etwas abstraktere Darstellung.

Deep Learning hat in den letzten Jahren erhebliche Fortschritte gemacht und findet Anwendung in Bereichen wie Bild- und Spracherkennung, autonomen Fahrzeugen und vielen anderen. Die Popularität von Deep Learning ist auf mehrere Faktoren zurückzuführen, darunter die Verfügbarkeit großer Datensätze, die Leistungsfähigkeit moderner Hardware und die Entwicklung effizienter Algorithmen.

### 2.1.3 Neuronale Netze

Das künstliche neuronale Netz (KNN) bildet die Grundlage der allermeisten Deep-Learning-Algorithmen. Es ist inspiriert von der Struktur und Funktionsweise des menschlichen Gehirns und besteht aus einer Vielzahl von miteinander verbundenen Knoten (Neuronen), die in Schichten organisiert sind. Die Struktur eines neuronalen Netzes besteht aus einer Eingabeschicht, einer oder mehreren versteckten Schichten (engl. *hidden layers*) und einer Ausgabeschicht.

Die einzelnen Neuronen, auf dem diese Netze aufbauen, sind eine mathematische Modellierung des biologischen Neurons, das erstmals 1943 von Warren McCulloch und Walter Pitts vorgestellt wurde (Zhou, 2021). Jedes Neuron empfängt eine Reihe von Eingaben, entweder von externen Quellen oder von den Ausgaben anderer Neuronen. Für jede dieser Eingaben gibt es ein zugehöriges Gewicht (engl. *weight*), das die Stärke und Richtung (positiv oder negativ) des Einflusses der jeweiligen Eingabe auf das Neuron bestimmt. Das Neuron berechnet dann die gewichtete Summe aller Eingabe und falls ein bestimmter Schwellenwert (engl. *bias*) überschritten wurde, wird das Neuron aktiviert. Diese Aktivierung kann durch verschiedene Aktivierungsfunktionen angepasst werden. Häufig wird etwa die sogenannte Sigmoid-Funktion verwendet, welche im Gegensatz zur einfachen Step-Funktion differenzierbar ist und somit die Optimierung des Netzwerk vereinfacht.

Die Optimierung des Netzwerks geschieht durch eine Rückwärtsausbreitung (engl. *back-propagation*), welche den berechneten Fehler rückwärts durch das Netz propagiert, um die Gewichte und Schwellenwerte um einen geringen Wert in die Richtung anzupassen, die den Fehler minimieren würde. ...

## 2.1.4 Convolutional Neural Networks

Ein Convolutional Neural Network (CNN) ist ein spezielles künstliches neuronales Netz, das hauptsächlich für die Bildklassifikation entwickelt wurde. Es verwendet Faltungsebenen, um ein Eingangsbild Schritt für Schritt in immer abstraktere “Feature Maps” zu verarbeiten.

Die Architektur eines CNN besteht typischerweise aus mehreren Schichten, die in der folgenden Reihenfolge angeordnet sind:

1. Eingabeschicht (Input Layer): Diese Schicht nimmt die Rohdaten auf, z.B. ein Bild in Form eines 2D-Arrays von Pixelwerten.
2. Faltungsschicht (Convolutional Layer): Diese Schicht führt die eigentliche Faltung (Convolution) durch, indem sie einen Filter (Kernel) über das Eingabebild verschiebt und Punktoperationen durchführt. Das Ergebnis ist eine Feature-Map, die lokale Merkmale des Bildes extrahiert. Jeder Filter kann unterschiedliche Merkmale wie Kanten, Ecken oder Texturen erkennen.
3. Aktivierungsschicht (Activation Layer): Nach jeder Faltungsschicht wird normalerweise eine Aktivierungsfunktion angewendet, um nichtlineare Eigenschaften des Netzwerks zu modellieren. Die häufig verwendete Aktivierungsfunktion ist die ReLU (Rectified Linear Unit), die alle negativen Werte auf Null setzt und positive Werte unverändert lässt.
4. Pooling-Schicht (Pooling Layer): Diese Schicht reduziert die räumliche Dimension der Feature-Maps, was die Berechnungen effizienter macht und die Gefahr von Überanpassung (Overfitting) verringert. Die gängigsten Pooling-Methoden sind Max-Pooling (wählt den maximalen Wert in einem bestimmten Bereich) und Average-Pooling (berechnet den Durchschnittswert in einem bestimmten Bereich).
5. Vollständig verbundene Schicht (Fully Connected Layer): Dies ist eine herkömmliche neuronale Netzwerkschicht, bei der jeder Neuron mit jedem Neuron der vorherigen Schicht verbunden ist. Sie kombiniert die extrahierten Merkmale, um das endgültige Ergebnis zu liefern, z.B. die Klassifikation des Bildes.
6. Ausgabeschicht (Output Layer): In der letzten Schicht wird eine Aktivierungsfunktion wie Softmax verwendet, um die Wahrscheinlichkeitsverteilung der möglichen Klassen zu berechnen.

## 2.1.5 Datenaugmentation

Datenaugmentation ist ein wichtiger Schritt im Training von neuronalen Netzen, insbesondere bei begrenzten Datensätzen. Sie bezieht sich auf die künstliche Erweiterung des

Trainingsdatensatzes durch Anwenden von Transformationen auf die vorhandenen Daten. Diese Transformationen können z.B. Rotation, Skalierung, Verschiebung, Spiegelung, Helligkeitsanpassung oder Rauschen sein. Das Ziel der Datenaugmentation ist es, das Modell robuster gegenüber Variationen in den Eingabedaten zu machen und die Generalisierungsfähigkeit zu verbessern.

...

### **2.1.6 Out-of-Distribution Daten**

...

## **2.2 Synthetische Daten**

Während die Verfügbarkeit großer Datensätze für das Training von neuronalen Netzen ein entscheidender Faktor für den Erfolg von Deep Learning-Modellen ist, ist es oft schwierig, solche Datensätze zu sammeln, insbesondere in Domänen wie der Medizin oder der Robotik, wo die Daten rar und teuer sind **<empty citation>** In solchen Fällen können synthetische Daten eine nützliche Alternative oder Ergänzung zu echten Daten sein.

Synthetische Daten sind künstlich erzeugte Daten, welche die zugrundeliegenden Muster der realen Daten nachahmen. Sie können durch Simulation, Generierung oder Transformation von echten Daten erstellt werden.

...

### **2.2.1 Variational Autoencoder**

...

### **2.2.2 Generative Adversarial Networks**

...

### 2.2.3 Stable Diffusion

Unter Diffusion versteht man den Prozess der langsamen Vermischung von Partikeln oder Informationen über die Zeit. In der Physik beschreibt die Diffusionsgleichung die zeitliche Entwicklung der Dichte von Teilchen, die sich zufällig bewegen. Dieses Konzept fand erstmals im maschinellen Lernen Anwendung, als Jascha Sohl-Dickstein et al. 2015 das Konzept der Diffusion-Modelle einföhrten (<empty citation>).

Bei Diffusion-Modellen handelt es sich um eine Klasse von generativen Deep Learning-Modellen, die in den letzten Jahren erhebliche Fortschritte erzielt haben. Im Trainingsprozess wird schrittweise die Struktur der Eingabedaten durch Hinzufügen von Rauschen aufgelöst. Das Modell wird dann darauf trainiert, das ursprüngliche Bild aus dem verrauschten Bild zu rekonstruieren.

...

### 2.2.4 DA-Fusion

...

## 2.3 Contrastive Learning

...

### 2.3.1 Unsupervised Contrastive Learning

...

### 2.3.2 Supervised Contrastive Learning

...

## 2.4 Integration von DA-Fusion und Supervised Contrastive Learning

...

### **2.4.1 Forschungslücke**

Der dargestellte Forschungsstand rund um die synthetische Generierung von Bilddaten und die Verwendung von Contrastive Learning zur Bildrepräsentation und Klassifikation ist vielversprechend. Während sich DA-Fusion als flexible Methode zur Datenaugmentation und zur Generierung von synthetischen Daten herausstellt, hat sich Contrastive Learning als effektive Methode zur Verbesserung der Generalisierungsfähigkeit und zur Verbesserung der Robustheit gegenüber Out-of-Distribution-Daten erwiesen. Vor dem Hintergrund des in dieser Arbeit behandelten Anwendungsfalls (s. Einleitung) stellt sich allerdings noch eine Forschungslücke dar, die nun erläutert werden soll.

...

Eine „herkömmliche“ Verwendung von DA-Fusion zur Generierung synthetischer Augmentationen und dessen naive Integration in ein Cross-Entropy Training könnte womöglich die Generalisierungsfähigkeit des Modells im Vergleich mit ausschließlich echten Daten verbessern. ...

...

### **2.4.2 Near Out-of-Distribution Augmentationen als Hard Negatives**

...

### **2.4.3 Potenzielle Vorteile und Herausforderungen**

...

## 3 Methodisches Vorgehen

In diesem Kapitel wird das methodische Vorgehen der Arbeit beschrieben. Es wird auf die Forschungsfragen und Hypothesen eingegangen, der verwendete Datensatz vorgestellt und die Implementierung der Modelle DA-Fusion und Supervised Contrastive Learning erläutert. Anschließend wird die synthetische Datengenerierung mit DA-Fusion und die Trainings- und Testdurchläufe mit Supervised Contrastive Learning beschrieben. Abschließend werden die Evaluationsmethoden und Metriken vorgestellt, die zur Analyse der Ergebnisse verwendet werden.

### 3.1 Forschungsfragen und Hypothesen

Im vorherigen Kapitel wurden Forschungslücken identifiziert, die sich aus der Verwendung von DA-Fusion im Supervised Contrastive Learning und der Verwendung von Near OOD-Augmentationen für das Negative Sampling ergeben. Um diese Lücken zu schließen, werden im Folgenden Forschungsfragen formuliert und Hypothesen aufgestellt, die im Rahmen dieser Arbeit untersucht werden sollen.

**1. Kann DA-Fusion für den EIBA-Datensatz synthetische Augmentationen erzeugen, die die Generalisierungsfähigkeit im Supervised Contrastive Learning verbessern?**

...

**2. Trägt die Verwendung von Out-of-Distribution (OOD) Augmentationen im Supervised Contrastive Learning dazu bei, die Robustheit des Modells gegenüber OOD-Daten zu erhöhen und die Repräsentationen von In-Distribution-Daten zu verbessern?**

...

### 3.2 Datensatz

...

### 3.2.1 EIBA

Grundlage der Forschungsarbeit ist ein am Fraunhofer-IPK entstandener Datensatz von Gebrauchsgegenständen, darunter hauptsächlich Autoteile und Komponenten. Er wurde im Rahmen des Projekts “Sensorische Erfassung, automatisierte Identifikation und Bewertung von Altteilen anhand von Produktdaten sowie Informationen über bisherige Lieferungen” (EIBA) erstellt, das von 2019 bis 2023 lief und von der Circular Economy Solutions GmbH koordiniert und in Kooperation mit der Technischen Universität Berlin und der deutschen Akademie der Technikwissenschaften durchgeführt wurde. **<empty citation>**

Der Datensatz ist multimodal, d.h. er besteht aus verschiedenen Datenquellen, die unterschiedliche Informationen über die Gegenstände enthalten. Neben herkömmlichen RGB-Bildern aus verschiedenen Perspektiven und weiteren Bilddaten, wie z.B. Objektmasken, gibt es auch Metadaten, etwa das Gewicht, oder Beschreibungen der Objekte in natürlicher Sprache durch verschiedene Stichwörter („CarComponent“, „cylinder“, „rusty“, usw.).

...

### 3.2.2 Teildatensatz

Um die Rechenzeit zu reduzieren und die Experimente auf eine bestimmte Objektkategorie zu beschränken, wurde ein Teildatensatz des EIBA-Datensatzes verwendet. Dabei wurden zufällig 20 Klassen aus der super class „CarComponent“ ausgewählt. Es wurden außerdem nur die RGB-Bilder verwendet, allerdings kommen auch die Objektmasken im Pre-Processing der Daten zum Einsatz. ...

...

### 3.2.3 Vorverarbeitung

...

## 3.3 Implementierung

Für die Vorbereitung der in dieser Arbeit durchgeführten Experimente konnte sich größtenteils auf die Implementierung von DA-Fusion und Supervised Contrastive Learning aus den Quellen ... und ... gestützt werden. Beide Implementierungen sind in Python geschrieben und verwenden die Bibliothek PyTorch. ...



Dennoch mussten einige Anpassungen vorgenommen werden, um die Modelle auf den EIBA-Teildatensatz anzuwenden, und um die synthetischen Augmentationen aus DA-Fusion im Supervised Contrastive Learning zu verwenden. ...

### **3.3.1 DA-Fusion**

In ...'s Implementierung von DA-Fusion wird zunächst mit Textual Inversion ein vortrainiertes Stable Diffusion-Modell fine-tuned, indem ein neuer Token für jede Klasse im Datensatz erlernt wird. Um anschließend die Augmentationen zu generieren, werden die Bilder des Datensatzes genommen, Rauschen hinzugefügt und unter Konditionierung auf den entsprechenden Token wiederhergestellt. Je nachdem, wie viel Rauschen hinzugefügt wurde, entstehen so mehr oder weniger stark veränderte Bilder, die als synthetische Daten verwendet werden können.

...

Die Implementierung von DA-Fusion kann weitgehend unverändert angewendet werden, um synthetische Daten für den EIBA-Teildatensatz zu generieren. Es muss lediglich eine eigene Klasse für den Datensatz erstellt werden, die die Bilder und Masken aus dem EIBA-Datensatz lädt und die Token für die Klassen bereitstellt. ...

### **3.3.2 Supervised Contrastive Learning**

...s Implementierung von Supervised Contrastive Learning beinhaltet drei Trainings-Skripte; eines für das Pre-Training der latenten Repräsentationen unter Verwendung der Supervised Contrastive Loss-Funktion, eines für die lineare Klassifikation der Repräsentationen und eines zum Training eines klassischen Klassifikator-Modells mit Cross Entropy Loss (zum Vergleich).

...

Auch hier muss eine eigene Klasse für den EIBA-Teildatensatz erstellt werden, die die Bilder und Masken lädt und die synthetischen Daten von DA-Fusion bereitstellt. Die Klasse muss nun auch Parameter bereitstellen, die die Verwendung der synthetischen Daten steuern, z.B. ob keine Augmentationen, ausschließlich positiv-Beispiele oder auch negativ-Beispiele verwendet werden sollen. ...

...

### **3.4 Synthetische Datengenerierung mit DA-Fusion**

...

### **3.5 Trainings- und Testdurchläufe mit Supervised Contrastive Learning**

...

### **3.6 Evaluationsmethoden und Metriken**

...

## **4 Ergebnisse**

Dieses Kapitel präsentiert die Ergebnisse der Arbeit. Es wird auf die generierten synthetischen Daten eingegangen und die Trainings- und Testergebnisse der Modelle beschrieben. Anschließend wird die Klassifikations-Performance der Modelle verglichen und die Out-of-Distribution-Detektion analysiert.

### **4.1 Die generierten synthetischen Daten**

...

#### **4.1.1 In-Distribution**

...

#### **4.1.2 Near Out-of-Distribution**

...

### **4.2 Trainings- und Testergebnisse mit Supervised Contrastive Learning**

...

#### **4.2.1 Contrastive Pre-Training**

...

#### **4.2.2 Lineare Klassifikation**

...

#### **4.3 Vergleich der Ergebnisse mit und ohne In-Distribution-Augmentationen**

...

#### **4.4 Vergleich der Ergebnisse mit und ohne Near Out-of-Distribution-Augmentationen als Hard Negatives**

...

## **5 Diskussion**

...

### **5.1 Eignung von DA-Fusion für die synthetische Datengenerierung**

...

### **5.2 Wirksamkeit von Near Out-of-Distribution-Daten als Hard Negatives im Supervised Contrastive Learning**

...

## **6 Fazit**

...

### **6.1 Zusammenfassung der wichtigsten Erkenntnisse**

...

### **6.2 Beantwortung der Forschungsfragen**

...

### **6.3 Ausblick und potenzielle Weiterentwicklungen**

...

# Literatur

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning* [<http://www.deeplearningbook.org>]. MIT Press.

Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill.

Zhou, Z.-H. (2021). *Machine Learning*. Springer.

# Anhang

Hier beginnt der Anhang. Siehe die Anmerkungen zur Sinnhaftigkeit eines Anhangs in Abschnitt ?? auf Seite ??.

Der Anhang kann wie das eigentliche Dokument in Kapitel und Abschnitte unterteilt werden. Der Befehl `\appendix` sorgt im Wesentlichen nur für eine andere Nummerierung.



# **Eigenständigkeitserklärung**

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel

## **Viele zufällige Zahlen**

selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Hamburg, 21. Dezember 1940