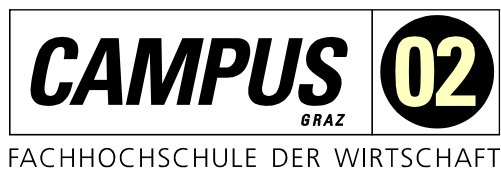


# MASTERARBEIT

## MACHINE LEARNING VERFAHREN FÜR BILDVERGLEICH IM EINSATZGEBIET DER SUCHE NACH ÄHNLICHEN PRODUKTEN EINER E-COMMERCE LÖSUNG

ausgeführt am



Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: André Gines

Personenkennzeichen: 1810320005

Graz, am 09. Dezember 2019

.....  
Unterschrift

## **EHRENWÖRTLICHE ERKLÄRUNG**

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

## **DANKSAGUNG**

Ich bedanke mich bei meiner Familie, meiner Freundin und meinem Freundeskreis, die mich immer wieder ermutigt und unterstützt haben und immer für mich da sind. Auch meinen JahrgangskollegInnen gilt ein besonderer Dank, die mir bei wichtigen Fragen weitergeholfen haben. Weiters möchte ich mich ausdrücklich bei Herrn DI Martin Bachler dem CTO der Firma Netconomy Software & Consulting GmbH für die Bereitstellung des Themenbereichs und das Vertrauen in meine Person bedanken.

Insbesondere möchte ich mich auch bei meinem Betreuer innerhalb der Firma Herrn DI Aid Ahmetovic bedanken. Ein besonderer Dank gilt auch Herrn Dr. Vid Jelen. Beide sind mir bei der Firma Netconomy Software & Consulting GmbH mit Rat und Tat zur Seite gestanden.

Ein besonderer Dank gilt Herrn DI Hans-Peter Grahl für die Betreuung des gesamten Prozesses der Fertigung dieser Arbeit.

Weiters möchte ich mich bei allen unterstützend mitwirkenden Personen bedanken, welche nicht namentlich oben erwähnt wurden.

## KURZFASSUNG

Diese Arbeit befasst sich mit dem Vergleich von unterschiedlichen Machine Learning Verfahren zur Ermittlung ähnlicher Bilder im Kontext einer Suche nach ähnlichen Produkten in E-Commerce Lösungen. Die Motivation liegt dabei in der Evaluierung möglicher Verfahren für den Einsatz in einem Produktivsystem. Das Ziel ist es, das geeignetste Verfahren bezogen auf einen spezifischen Anwendungsfall zu ermitteln. Konkret werden drei Verfahren in einer Basisvariante umgesetzt und mit geeigneten Metriken evaluiert. Ein bestehendes Datenset an realen Produktbildern wird zu Evaluierungszwecken verwendet. Kein Ziel dieser Arbeit ist es, die jeweiligen Verfahren soweit zu optimieren, sodass diese die jeweils bestmöglichen Ergebnisse zurückliefern. Auf Basis eines Experiments werden die zuvor recherchierten Verfahren anhand des Anwendungsfalles einer Suchfunktionalität nach ähnlichen Produkten evaluiert. Es kommen dabei unterschiedliche Evaluierungsmetriken zum Einsatz. Als Hauptevaluierungsmetrik der Verfahren wird die Mean Average Precision@K verwendet. Transfer Learning mit einem vortrainierten VGG16 Modell und darauf angewandtes Fine-Tuning liefert in Bezug auf diese Metrik die besten Ergebnisse. Dahinter landet das Autoencoder Verfahren. Das Siamese Network Verfahren liefert im direkten Vergleich mit den beiden anderen die schlechtesten Ergebnisse. Dennoch eignen sich alle drei Verfahren für den untersuchten Anwendungsfall, sodass im Schnitt mindestens 50 Prozent der Suchergebnisse innerhalb der top fünf Ergebnisse relevant sind. Für einen potenziellen Einsatz in einer Produktivumgebung wäre es trotzdem erforderlich und empfehlenswert, weitere Optimierungen der dahinterliegenden Modelle vorzunehmen. Passend zur Hypothese haben die Experimente bestätigt, dass Transfer Learning trainiert mit den Bildern aus dem gegebenen Datenset bessere Ergebnisse ermöglichen kann. Weitere Forschungstätigkeiten umfassen die Optimierung der Verfahren und des Datensets sowie die Evaluierung weiterer, alternativer Verfahren beziehungsweise Kombinationen davon, die nicht Teil der vorliegenden Arbeit sind.

## ABSTRACT

This study compares various machine-learning approaches to find similar images within the context of related products search functionality in e-commerce. The goal is to identify viable approaches and find the optimal strategy for a specific use case. Three different methods are implemented with some basic optimisations. These approaches are then evaluated with standard metrics. A provided dataset of real product images serves for evaluation. Custom in-depth optimisations for each machine-learning algorithm to find the best results is beyond the scope of this study. Based on a pre-defined use case, an experiment is conducted for three different approaches which were identified during the literature research. While different evaluation criteria are used, the primary metric is the Mean Average Precision@K. Experiments show that based on this metric, Transfer Learning with a pre-trained and fine-tuned VGG16 model scores best, followed by the Autoencoder approach. The Siamese Network shows the weakest results. However, all three approaches are suitable for the defined use case, since most of the top five search results are relevant. Should any of the three approaches be run in production, further optimisations of the underlying models would still be necessary. Experiments show that Transfer Learning trained with pictures in the provided dataset can lead to better results. Future research activities may include optimisations of the investigated approaches and dataset, and could also address alternative approaches.

# INHALTSVERZEICHNIS

<b>1</b>	<b>EINLEITUNG</b>	<b>1</b>
1.1	Problemstellung	1
1.2	Zielsetzung	1
1.3	Forschungsfrage	2
1.4	Methodik	2
<b>2</b>	<b>MACHINE LEARNING</b>	<b>4</b>
2.1	Definition	4
2.2	Herausforderungen	6
2.3	Kriterien	8
2.4	Arten und Anwendungsgebiete	9
2.4.1	Supervised learning	9
2.4.2	Supervised learning	10
2.4.3	Semi-supervised learning	10
2.4.4	Reinforcement learning	11
2.5	Algorithmen	11
2.6	Prozess	13
2.7	Messkriterien	16
2.8	Neuronale Netze	19
2.9	Deep Learning	21
2.9.1	Netzarten	22
2.9.2	Modell	25
2.9.3	Training	26
2.9.4	Hyperparametertuning	27
<b>3</b>	<b>E-COMMERCE</b>	<b>29</b>
<b>4</b>	<b>PRODUKTEMPFEHLUNGEN</b>	<b>33</b>
<b>5</b>	<b>BILDANALYSE</b>	<b>35</b>
5.1	Unterschied Bildähnlichkeit und Bildklassifizierung	35
5.2	Embedding	35
5.3	Visualisierung	36
5.4	Content-based Image Retrieval	37

5.5	Suche nach Bildern.....	38
<b>6</b>	<b>VERFAHREN FÜR BILDVERGLEICHE .....</b>	<b>40</b>
6.1	Autoencoder .....	41
6.2	Deep Ranking .....	42
6.3	Siamese Network.....	45
6.4	Transfer Learning .....	47
6.5	Evaluierung.....	49
<b>7</b>	<b>EXPERIMENTSETUP .....</b>	<b>52</b>
7.1	Anwendungsfall .....	52
7.2	Eingrenzung.....	52
7.3	Hypothesenbildung .....	53
7.4	Operationalisierung.....	54
7.5	Kriterienkatalog.....	55
<b>8</b>	<b>ARTEFAKTERSTELLUNG .....</b>	<b>57</b>
8.1	Prozess.....	57
8.1.1	Geschäftsproblem identifizieren .....	57
8.1.2	Machine Learning Zuordnung.....	58
8.1.3	Datensammlung.....	58
8.1.4	Qualitätsüberprüfung der Daten .....	58
8.1.5	Datenauswahl und Transformierung .....	58
8.1.6	Feature Analyse und Auswahl & Machine Learning Modellierung.....	58
8.1.7	Modellevaluierung.....	58
8.1.8	Modellauswahl und Deployment.....	59
8.1.9	Performanceüberwachung des Modells .....	59
8.2	Datenherkunft .....	59
8.3	Datensammlung.....	60
8.4	Vorverarbeitung der Daten .....	61
8.4.1	Image Pre-Processing .....	61
8.4.2	Aufteilung der Daten .....	62
8.4.3	Datenset .....	64
8.5	Tools .....	68

8.5.1	Python.....	69
8.5.2	NumPy.....	69
8.5.3	TensorFlow.....	69
8.5.4	Keras.....	69
8.5.5	TensorBoard.....	70
8.5.6	PyCharm.....	70
8.5.7	MLflow.....	70
8.5.8	Git.....	70
8.5.9	Matplotlib.....	70
8.6	Implementation der Verfahren.....	71
8.6.1	Allgemein.....	71
8.6.2	Vorverarbeitung.....	72
8.6.3	Laden und Transformieren.....	72
8.6.4	Modell Setup: Autoencoder.....	73
8.6.5	Modell Setup: Siamese Network.....	73
8.6.6	Modell Setup: Transfer Learning.....	73
8.6.7	Vorhersage.....	74
8.6.8	Suche.....	74
8.6.9	Evaluierung.....	75
8.6.10	Visualisierung.....	75
<b>9</b>	<b>MESSUNG.....</b>	<b>78</b>
9.1	Sicherstellung der Validität, Zuverlässigkeit und Objektivität.....	78
9.2	Durchführung.....	79
9.3	Ergebnisse.....	79
9.3.1	Autoencoder.....	80
9.3.2	Siamese Network.....	84
9.3.3	Transfer Learning.....	88
<b>10</b>	<b>BEWERTUNG.....</b>	<b>92</b>
<b>11</b>	<b>CONCLUSIO.....</b>	<b>96</b>
11.1	Ergebnisvergleich.....	96
11.2	Interpretation.....	97
11.3	Erkenntnisgewinn.....	97
11.4	Hypothesen.....	98
11.5	Forschungsfrage.....	98



11.6 Weitere Forschungsmöglichkeiten .....	99
<b>ANHANG A - AUTOENCODER ERGEBNISSE .....</b>	<b>100</b>
<b>ANHANG B - SIAMESE NETWORK ERGEBNISSE .....</b>	<b>106</b>
<b>ANHANG C - TRANSFER LEARNING OHNE FINE-TUNING ERGEBNISSE .....</b>	<b>112</b>
<b>ANHANG D - TRANSFER LEARNING MIT FINE-TUNING ERGEBNISSE.....</b>	<b>118</b>
<b>ABKÜRZUNGSVERZEICHNIS.....</b>	<b>124</b>
<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>125</b>
<b>TABELLENVERZEICHNIS .....</b>	<b>126</b>
<b>LITERATURVERZEICHNIS .....</b>	<b>127</b>

# 1 EINLEITUNG

Eine der Domänen in denen Empfehlungssysteme hauptsächlich zum Einsatz kommen ist E-Commerce. Ein Beispiel dafür sind Empfehlungen für Produkte auf Basis der ebenfalls von anderen Kunden gekauften Produkte anhand des gerade angezeigten Artikels. (Thanaki, 2018)

Den BenutzerInnen werden personalisierte Inhalte zur Verfügung gestellt. Diese Inhalte werden auf Basis von bereits vorhandenen Daten ermittelt und den BenutzerInnen auf der jeweiligen Seite angezeigt. Ein Beispiel für ein solches Empfehlungssystem stellt die Funktionalität der Suche nach ähnlichen Produkten dar. Auf Basis von bereits vorhandenen Daten werden geeignete Vorschläge identifiziert. Im Rahmen dieser Arbeit wird der Fokus auf diese Suche nach ähnlichen Produkten im Kontext einer E-Commerce Plattform gelegt.

## 1.1 Problemstellung

Es gibt unterschiedliche Verfahren zur Ermittlung von ähnlichen Produkten. Einfache Verfahren stellen die Suche ausschließlich anhand von textuell verfügbaren Inhalten dar. Diese Inhalte können Produktdaten sein, welche das Produkt näher charakterisieren, wie zum Beispiel die Maße oder die Farbe des Produktes. Anhand dieser Daten wird nach ähnlichen Produkten gesucht.

Mithilfe von Machine Learning (ML) gibt es nun weitere Verfahren, um Empfehlungssysteme umzusetzen. Die derartige Suche kann mithilfe von Bildanalyseverfahren umgesetzt werden. Anhand von Produktbildern werden Bildvergleiche eingesetzt, um ähnliche Produkte zu finden, welche den BenutzerInnen im Anschluss empfohlen werden.

Auch eine Kombination von textuellen Inhalten und Bildanalyseverfahren kann durchaus sinnvoll sein.

## 1.2 Zielsetzung

Im Rahmen dieser Arbeit sollen mögliche Verfahren für Bildvergleiche mit Machine Learning im Kontext einer Suche nach ähnlichen Produkte Funktionalität ermittelt werden. Zuerst werden bestehende Verfahren für den Bildvergleich erhoben. Nach dem Abgleich mithilfe von definierten Kriterien werden ausgewählte Verfahren zur genaueren Untersuchung übernommen.

Die im Rahmen dieser Arbeit zu definierenden Messwerte aus den jeweiligen Verfahren werden gesammelt und für einen Vergleich anhand der definierten Kriterien herangezogen. Dieser Vergleich ermöglicht eine Bewertung der Verfahren. Anschließend können Aussagen darüber getroffen werden, welche der Verfahren gut für Bildvergleiche im Kontext einer E-Commerce

Plattform geeignet sind. Auch die Vorteile und Nachteile der jeweiligen Verfahren können ermittelt werden.

Die im Rahmen dieser Arbeit zu verfolgenden Ziele lauten somit wie folgt:

- Erhebung von Machine Learning Verfahren für Bildvergleiche
- Test auf Eignung dieser Verfahren im Kontext einer Suchfunktionalität nach ähnlichen Produkten einer E-Commerce Plattform

### **1.3 Forschungsfrage**

Das übergeordnete Ziel dieser Arbeit ist die Beantwortung der Forschungsfrage. Diese lautet wie folgt: Welche ausgewählten Machine Learning Verfahren eignen sich besonders gut für den Bildvergleich im konkreten Einsatzgebiet der Suche nach ähnlichen Produkten für eine E-Commerce Lösung?

### **1.4 Methodik**

Für die Beantwortung der Forschungsfrage muss eine Vorgehensweise definiert werden, um ein strukturiertes Beantworten zu ermöglichen. Im ersten Schritt werden die Konstrukte der Forschungsfrage näher erläutert.

Das Thema Machine Learning stellt einen zentralen Bereich dieser Arbeit dar und muss mit entsprechender Priorität behandelt werden. Dieser Bereich soll die Grundlage für das Verständnis der nachfolgenden Verfahren schaffen. Es soll geklärt werden, was Machine Learning überhaupt ist, welche Arten es gibt, wie ein typischer Machine Learning Prozess aussieht, wie benötigte Datenquellen beschaffen werden und welche Tools dabei zum Einsatz kommen können.

Auch das Thema E-Commerce spielt eine zentrale Rolle. Eine Begriffsdefinition soll dabei erfolgen, welche Relevanz das Thema generell hat, wie ein typischer Aufbau einer solchen Plattform erfolgt und wo diesbezüglich Produktempfehlungen typischerweise angezeigt werden. Anhand der Erläuterung dieses Themas wird sich im Rahmen dieser Arbeit zeigen, inwiefern dieses Thema Einfluss auf die Definition der Kriterien zur Bewertung der Verfahren hat.

Empfehlungssysteme und spezifisch Produktempfehlungen werden im Anschluss näher erläutert. Es gilt dabei zu klären was Empfehlungssysteme überhaupt sind, wo diese auf der Webseite platziert werden und welche Anforderungen an diese einher gehen. Auch diese Anforderungen können in späteren Schritten für die Definition der Kriterien zur Bewertung der Verfahren herangezogen werden.

Als Nächstes wird die in den Verfahren zum Einsatz kommende Bildanalyse, welche den Bildvergleich anschließend ermöglicht, genauer untersucht. Es wird genauer dargelegt was genau mit diesem Begriff gemeint ist. Außerdem wird erhoben, welche Möglichkeiten es gibt,

Daten aus Bildern zu erheben, welche für einen Vergleich von Bildern anschließend herangezogen werden können. Die Verknüpfung mit Machine Learning wird ebenso dargelegt.

Nachdem die Grundlage für das weitere Verständnis dieser Arbeit geschaffen wurde, werden bestehende Verfahren für Bildvergleiche erhoben. Unterschiedliche Verfahren und deren Umsetzung sowie Einsatzgebiete sollen für die weitere Bearbeitung innerhalb dieser Arbeit beschrieben werden.

Anhand der geschaffenen Basis kann eine weitere Eingrenzung des Themengebiets erfolgen. Es werden nur ausgewählte Verfahren einer genaueren Untersuchung vollzogen. Diese werden anhand definierter Kriterien ausgewählt.

Die Hypothesenbildung erfolgt anhand der ausgewählten Verfahren. Diese ist daher erst nach der Auswahl der Verfahren möglich. Die Literaturrecherche zeigt dabei auf, welche Hypothese im Rahmen der Arbeit sinnvoll ist und wird anhand dessen definiert.

Damit die Hypothese überprüft werden kann benötigt es eine entsprechende Operationalisierung. Unter Operationalisierung wird die Definition der Messwerte zur Beurteilung der Verfahren verstanden. Die unterschiedlichen Messwerte münden in einem Kriterienkatalog anhand dessen die Verfahren in späteren Schritten analysiert und bewertet werden. Das Ergebnis dieser Phase stellt der Kriterienkatalog selbst dar.

Die Artefakterstellung stellt die nächste Phase dar. Zuerst wird die Herkunft der Daten ermittelt, welche für die weitere Verarbeitung und zum Vergleich der Verfahren verwendet werden. Da Bilder viel Speicherplatz auf der Festplatte einnehmen können, muss ein möglicher hoher Speicherverbrauch der Daten berücksichtigt werden. Diese Daten werden daher mit entsprechenden Tools verwaltet. Die Verfahren zum Bildvergleich werden anhand entsprechender Tools implementiert. Bevor dies geschehen kann muss eine Toolauswahl erfolgen. Das Artefakt selbst ist der Programmcode. Dieser variiert innerhalb der unterschiedlichen Verfahren. Außerdem muss sichergestellt sein, dass alle im Kriterienkatalog definierten Kriterien aus dem jeweiligen Verfahren abgeleitet werden können und somit die Messmöglichkeit innerhalb eines jeden Verfahrens gegeben ist.

Nach der Erstellung des Artefakts werden Messungen vorgenommen. Damit Validität und Reliabilität gegeben sind, werden mehrere Testdurchläufe angedacht. Für jedes Verfahren werden im Anschluss die Messwerte abgelesen. Anhand der abgelesenen Messwerte und des Kriterienkatalogs kann der Vergleich der untersuchten Verfahren erfolgen. Nach dem Vergleich der unterschiedlichen Verfahren erfolgt eine Bewertung anhand des Kriterienkatalogs.

Nachdem dies geschehen ist, werden die Ergebnisse dargestellt und interpretiert. Aus den Erkenntnissen im Hinblick auf die Eignung der unterschiedlichen Verfahren für den Bildvergleich können anschließend die Hypothesen und die Forschungsfrage beantwortet werden. Abschließend wird ein Ausblick für weiterführende Fragestellungen und Forschung in diesem Gebiet gegeben.

## 2 MACHINE LEARNING

Der Begriff Machine Learning (ML) wird heutzutage sehr häufig verwendet. Oftmals findet ML Anwendung hinter Funktionalitäten, in denen man es nicht erwarten würde. Ein Beispiel hierfür stellt die Spam Erkennung und Filterung von Emails dar. Typische Einsatzgebiete stellen die Entscheidungsfindung in Firmen, Produktivitätssteigerungen, Diagnosen, Wettervorhersagen dar, wobei es noch viele weitere gibt. Mithilfe von ML ist es möglich, Daten einen Sinn zu geben. Es transformiert somit Daten zu Informationen, um ein klareres Bild zur Lösung der Aufgabenstellung zu erhalten. (Harrington, 2012)

Wie genau ist aber ML eigentlich definiert? Wann soll ML angewendet werden? Welche Herausforderungen bringt es mit sich? Welche Arten von ML gibt es? Welche Bereiche innerhalb von ML gibt es und wie sieht ein typischer Prozess aus? Welche Messkriterien können dabei zum Einsatz kommen? Was sind neuronale Netze und was bedeutet der Begriff Deep Learning? Die nachfolgenden Kapitel befassen sich mit diesen Fragestellungen.

### 2.1 Definition

Eine Definition von Machine Learning steht in Harrington (2012) beschrieben:

*“Machine learning lies at the intersection of computer science, engineering, and statistics and often appears in other disciplines.” (Harrington, 2012, S. 5)*

ML ist also eine Überschneidung der Disziplinen Computerwissenschaften, Ingenieurwesen und Statistik und kommt in den unterschiedlichsten Bereichen vor. Für Probleme, wofür zu wenig Rechenleistung vorhanden ist, kommt Statistik zum Einsatz. Ein Beispiel hierfür ist die Motivation von Menschen. Dieses Problem ist zu schwer zu modellieren, da jeder Mensch unterschiedlich ist. Generell wird die Wichtigkeit von ML in der Zukunft steigen. (Harrington, 2012)

Eine weitere Definition steht in Simeone (2017) beschrieben und geht genauer auf den Zweck von ML ein:

*“The machine learning alternative is to collect large data sets, e.g., of labelled speech, images or videos, and to use this information to train general-purpose learning machines to carry out the desired task.” (Simeone, 2017, S. 7)*

Aus dieser Definition lässt sich ableiten, dass große Datenmengen unabhängig von der Art von ML verwendet werden können, um Informationen zu erhalten. Mit diesen Informationen lassen sich wiederum Modelle trainieren, um die gegebene Problemstellung zu lösen. Was ist aber nun ein Modell?

In Bonaccorso (2018b) werden Modelle als mathematische Systeme beschrieben. Sie ermöglichen es komplexe Systeme einfacher zu verstehen.

In Simeone (2017) wird außerdem der Unterschied zwischen einem Standard Entwicklungsprozess und der des Machine Learnings näher beschrieben. Der Standard Entwicklungsprozess basiert auf Domänenwissen und benötigt für gewisse Probleme mehr Aufwand, um diese zu lösen. Außerdem muss dieser konkret auf die Aufgabenstellung zugeschnitten werden. ML hingegen verwendet Daten, um generelle Modell zu trainieren. Diese Modelle sagen wiederum mögliche Ausgänge auf Basis des trainierten Modells hervor. (Simeone, 2017)

Die Beschreibung des Lernvorganges soll für ein besseres Verständnis für Machine Learning im Allgemeinen sorgen.

Die in Mitchell (1997) dargestellte Definition ist generell gehalten und stützt sich auf das Lernproblem selbst und die damit verbundenen Komponenten:

*“A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .” (Mitchell, 1997, S. 2)*

Der Lernvorgang besteht laut Mitchell (1997) somit aus den drei Komponenten Aufgabe  $T$ , der Performancemessung  $P$  und der Erfahrung  $E$ . Ein dargestelltes Beispiel stellt die Handschrifterkennung dar. Die Erkennung der Schrift in Bildern ist die Aufgabe  $T$ . Der Prozentwert der korrekt erkannten Wörter ist die Performancemessung  $P$ . Die Erfahrung  $E$  ist eine Datenbank von handgeschriebenen Wörtern und der dazugehörigen Klassifizierung. Auch erwähnt wird, dass ML seine Ideen von unterschiedlichen Disziplinen erhält unter anderem von Artificial Intelligence (AI).

Was genau ist jetzt aber AI und wodurch unterscheidet es sich von ML?

Brink, Richards & Fetherolf (2017) beschreiben ML als Teilgebiet von AI. Der Begriff AI ist somit breiter gefasst. Dennoch lässt sich gut abgrenzen was ML ist und was nicht. Sie beschreiben den Begriff als die Kombination von mehreren Strategien, um ein Problem zu lösen. Eine Strategie besteht wiederum aus mehreren Algorithmen. Diese Algorithmen wiederum lernen anhand von Beispielen, erhalten somit Erfahrung und haben die Fähigkeit dieses Wissen für die Zukunft zu verwenden und zu generalisieren. Weiters wird beschrieben, dass ML aus mehreren Disziplinen besteht. Diese Disziplinen umfassen Statistik, Computerwissenschaften, Robotik und angewandte Mathematik. Ebenso werden die Online Suche, Entertainment, Internetwerbung und Sprachübersetzung als Beispiele genannt. ML ist besonders nützlich für Probleme, welche aufgrund hoher Datenmengen manuell nicht lösbar sind. Außerdem wird ein Beispiel veranschaulicht in denen Bilder von Katzen und Hunden richtig zugeordnet werden sollen. Die menschliche Fehlerrate liegt bei ungefähr sieben Prozent während das Machine Learning System 98,914 Prozent richtig erkannt hat. Als wichtiger Punkt wird ebenso angeführt, dass Entscheidungen anhand von Daten gemacht werden und diese Entscheidungen anhand der Daten wiederum optimiert werden.

*“The process of learning from data, and subsequently using the acquired knowledge to inform future decisions, is extremely powerful. Indeed, machine learning is rapidly becoming the engine that powers the modern data-driven economy.” (Brink et al., 2017, S. 7)*

Wie oben beschrieben wird auch hier der ML Prozess als sehr wichtig angesehen und hilft ebenso die Datenverarbeitung zu unterstützen.

ML arbeitet mit Algorithmen. Was aber genau ist ein ML Algorithmus? In Bonaccorso (2018b) wird dieser Begriff genauer definiert:

*“Machine learning algorithms work with data. They create associations, find out relationships, discover patterns, generate new samples, and more, working with welldefined datasets.” (Bonaccorso, 2018b, S. 9)*

Aus dieser Definition ist ersichtlich wofür ML Algorithmen verwendet werden. Sie arbeiten mit Daten und versuchen daraus Beziehungen und Muster zu erkennen. Die dabei verwendeten Daten sollen dabei eine hohe Qualität aufweisen. (Bonaccorso, 2018b)

Die oben erwähnten Definitionen und Aussagen zur Beschreibung von ML weichen von den einzelnen Quellen durchaus voneinander ab und fokussieren sich jeweils auf andere Schwerpunkte. Mithilfe dieser Aussagen kann man sich jedoch einen Überblick über ML verschaffen. Die oben genannten zentralen Aussagen über ML gliedern sich wie folgt:

- ML ist ein Lernprozess für die Entscheidungsfindung
- Training verbessert diese Entscheidungsfindung
- Modelle werden anhand von Algorithmen und Trainingsdaten erstellt und trainiert
- Es fließen mehrere Disziplinen in ML ein
- Umgekehrt hat ML auch in mehreren Gebieten Verwendung
- ML kann mit großen Datenmengen umgehen
- ML ist für Problemstellungen geeignet, welche manuell nicht oder kaum durchführbar wären
- ML bekommt in der Zukunft noch größeren Einfluss

## **2.2 Herausforderungen**

Der Einsatz von ML bringt auch Herausforderungen mit sich. In diesem Kapitel werden einige Herausforderungen beschrieben, um in der nachfolgenden Arbeit genauer darauf eingehen zu können und diese zu meistern.

In Brink et al. (2017) werden unterschiedliche Herausforderungen beschrieben. Die erste Herausforderung stellt dabei die Datenaufbereitung dar. Diese Aufbereitung soll laut Datenspezialisten schätzungsweise 80 Prozent ihrer Zeit in Anspruch nehmen. Der Grund

hierfür ist, dass immer mehr Daten generiert werden und diese wiederum brauchbar für Machine Learning gemacht werden müssen. Auch das Filtern von brauchbaren Daten kann sehr viel Zeit in Anspruch nehmen.

Eine weitere Herausforderung stellt die Formulierung des Problems dar, sodass ML darauf angewendet werden kann. Messbare Ergebnisse stellen dabei das Ziel dar. Am Beispiel eines optimalen Media Mix in Kombination mit Werbeeinheiten, um Markenbewusstsein für eine neue Produktlinie zu schaffen soll dies dargestellt werden. Es muss zuerst definiert werden, wie man Markenbewusstsein überhaupt messen kann. (Brink et al., 2017)

Die nachfolgende Herausforderung beschäftigt sich mit ML Algorithmen, welche bereits in den vorigen Kapiteln näher erläutert wurden.

Die Wahl des Algorithmus und wie dieser angewendet wird, kann einen hohen Aufwand darstellen. Die Daten müssen erst extrahiert werden und in das richtige Format gebracht werden. Dieser Prozess wird auch Feature Engineering genannt. Am Beispiel eines Modells in der Medizin, um die Wahrscheinlichkeit von Komplikationen bei Operationen zu messen ist dies ersichtlich. Ein Elektrokardiogramm muss erst in Daten umgewandelt werden, womit der Algorithmus arbeiten kann. (Brink et al., 2017)

Ein Feature ist das Ergebnis der Umwandlung von Rohdaten in für das Modell brauchbare Daten. Es sollten sich deshalb auch Gedanken darüber gemacht werden, wie man die einzelnen Features im Feature Engineering Prozess produziert. (Smith, 2018)

Overfitting ist ein Problem, welches auftreten kann, wenn aus dem Trainingsdatensatz zu viele Informationen geholt werden. Die Folge ist, dass das Modell nur gut für diese Trainingsdaten funktioniert aber für weitere Datensätze nicht. Ebenso kann Overfitting auftreten, wenn das Modell so komplex gestaltet wird, sodass jeder Trainingsdatensatz passt. Ein Beispiel für Overfitting stellt eine Musterprüfung dar. Es werden die Fragen und Antworten gelernt und bereitet sich so für die echte Prüfung vor. Möglicherweise werden diese sogar auswendig gelernt und wendet diese dann in der echten Prüfung ebenso an. In der Realität wird man deshalb bei der echten Prüfung eher schlechter abschneiden, da es unwahrscheinlich ist, die exakt selben Fragen und Antworten zu erhalten. (Liu, 2019)

Um Overfitting zu vermeiden gibt es diverse Ansätze. Einige werden in Rothman et al. (2018) beschrieben. Einer der möglichen Ansätze ist mehr Daten für das Training zu generieren und zu verwenden. Ein weiterer Ansatz ist die Reduktion der Komplexität des Netzes indem die Anzahl der Layer angepasst werden.

Underfitting stellt den Gegensatz zu Overfitting dar. Das Trainingsdatensatz als auch das Testdatensatz schneiden angewendet auf das vorliegende Modell schlecht ab. Das Problem ist, dass das Modell den Trend innerhalb der Daten nicht erkennt. Auftreten kann dies aufgrund zu wenig vorhandener Daten, um das Modell zu trainieren. Ein Beispiel dazu ist das zu wenig Zeit in die Vorbereitung für die Prüfung investiert wurde. Underfitting kann auch auftreten, wenn das falsche Modell auf die Daten angewandt wird. Ein Beispiel hierfür ist, dass man bei allen Prüfungen schlecht abschneidet, da einfach falsch gelernt wird. (Liu, 2019)



Generell gilt es einen Bias-Variance trade-off zu finden. Bei hohem Bias kommt es zu einem Underfitting Problem, bei zu hoher Varianz zu einem Overfitting Problem. Es gilt beide in Einklang zu halten, da ein hoher Bias die Varianz reduziert und umgekehrt. (Liu, 2019)

Im Rahmen der weiteren Arbeit wird versucht auf diese Herausforderungen zu reagieren und entsprechend zu lösen.

## 2.3 Kriterien

Wann kann ML überhaupt angewendet werden? Wann macht es Sinn? Diese Fragen versucht dieses Kapitel zu beantworten. Die Kriterien werden dann mit der konkreten Aufgabestellung innerhalb dieser Arbeit gegenübergestellt.

In Simeone (2017) werden acht Kriterien vorgeschlagen. Diese lauten wie folgt:

- Aufgabenstellung wandelt klar definierte Eingabewerte in klar definierte Ausgabewerte um
- Große Mengen an Daten sind vorhanden (zum Beispiel eine hohe Anzahl an Bildern oder Videos)
- Aufgabenstellung benötigt klares Feedback mit klaren definierten Zielen und Werten
- Aufgabenstellung hat keine lange Verkettung von Logiken, welche abhängig von Hintergrundwissen ist
- Aufgabenstellung benötigt keine detaillierte Erklärung wie die Entscheidung zustande kam
- Fehlertoleranz ist gegeben und es wird keine optimale Lösung vorausgesetzt
- Das zu lernende Objekt sollte sich nicht rapide ändern
- Keine spezielle Geschicklichkeit, keine physischen Fähigkeiten oder Mobilität muss vorhanden sein

Die Aufgabenstellung dieser Arbeit ist es, ML Verfahren für Bildvergleiche zu analysieren und zu bewerten. Stellt man diese Aufgabenstellung diesen Kriterien gegenüber, so kann man erkennen, dass alle von diesen erfüllt sind. Eingabewerte sind die Eigenschaften eines Produktes, welche anhand der Bildanalyse extrahiert werden können und die Ausgabe sind ähnliche Produkte sowie Verweise auf deren Bilder. Aufgrund der Produktanzahl sind hohe Mengen an Daten zu erwarten, da Bilder sehr viel Speicherplatz einnehmen können. Das in dieser Arbeit verwendete nicht modifizierte Datenset besteht aus insgesamt 43759 Produkten. Das Ziel ist es möglichst ähnliche Produkte zu finden, wobei hier der Schwellwert, ob ein Produkt sich ähnlich ist oder nicht, im Laufe der Arbeit noch konkretisiert wird. Es ist kein Hintergrundwissen und keine Erklärung, warum dies nun ein ähnliches Produkt ist, notwendig. Die Fehlertoleranz ist auch gegeben in dem Wissen, dass möglicherweise ein nicht optimales ähnliches Produkt gefunden wird. Die Art und Weise wie die Eigenschaften der Produkte

ermittelt werden verändert sich auch nicht rapide und spezielle Fähigkeiten müssen auch nicht vorhanden sein.

## 2.4 Arten und Anwendungsgebiete

Die Einsatzgebiete unterscheiden sich zwischen den Arten. In diesem Kapitel werden Arten und Anwendungsgebiete näher erläutert und mit Beispielen erklärt. Es wird nur auf jene Arten genauer eingegangen, welche für diese Arbeit von Relevanz sind.

In Kapoor (2019) werden drei Arten von ML näher erläutert:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

### 2.4.1 Supervised learning

Supervised learning (Überwachtes Lernen) basiert auf Trainingsdaten und den dazugehörigen passenden Antworten. Der Algorithmus generalisiert diese und es wird versucht anhand dieser Daten die richtige Antwort für ein weiteres gegebenes Beispiel vorherzusagen. Supervised learning lernt also anhand von bereits gegebenen Daten mit bekanntem Ergebnis. (Marsland, 2015)

Regression und Klassifikation sind zwei Arten von Problemstellungen innerhalb von Supervised learning. (Kumar, 2019)

Die Regression selbst versucht eine Beziehung zwischen den gegebenen Inputvariablen herzustellen. Das Ergebnis davon ist eine stetige Variable. (Ciaburro, 2019)

Ein Beispiel für die Regression ist in Kumar (2019) beschrieben. Die Vorhersage von Hauspreisen basiert auf mehreren Inputvariablen, wie zum Beispiel die Anzahl der Räume, Stadt, Region und viele weitere. Der vorhergesagte Preis ist eine stetige Variable.

Die Klassifikation nimmt ebenso Inputvariablen entgegen und versucht den Datensatz innerhalb vordefinierter Klassen zuzuweisen. Bei der Klassifikation selbst ist das Ergebnis eine diskrete Variable. Jedes Beispiel gehört somit genau zu einer Klasse. Die Anzahl der Klassen ist der mögliche Ergebnisbereich. (Marsland, 2015)

Ein Beispiel für eine binäre Klassifikation ist in Kumar (2019) beschrieben. Das Beispiel beschäftigt sich mit der Frage, ob ein Mitarbeiter die Firma verlässt oder nicht, basierend auf vorhandenen Daten, wie zum Beispiel dem Gehalt, Alter, Anzahl der Familienmitglieder und mögliche weitere. Das Ergebnis ist eine diskrete Variable mit den möglichen Werten „Ja“ oder „Nein“.

Klassische Anwendungsgebiete für Supervised learning sind in Bonaccorso (2018) beschrieben. Diese umfassen folgende Bereiche:

- Vorhersagende Analysen
- Spamerkennung
- Mustererkennung
- Natural Language Processing
- Sentimentanalyse
- Bildklassifizierung
- Sequence Processing wie Musik oder Sprache

### **2.4.2 Supervised learning**

Supervised learning (Unüberwachtes Lernen) verfolgt einen anderen Ansatz. Es werden keine Trainingsdaten und auch keine passenden Antworten benötigt. Es wird versucht Ähnlichkeiten zwischen den Eingabedaten zu ermitteln und ähnliche Daten zu gruppieren. (Marsland, 2015)

Klassische Anwendungsgebiete für Supervised learning sind in Bonaccorso (2018) beschrieben. Diese umfassen folgende Bereiche:

- Objektsegmentierung, wie zum Beispiel Benutzer, Produkte, Filme, Lieder und viele Weitere
- Erkennung von Ähnlichkeiten
- Automatische Kennzeichnung
- Empfehlungssystem

### **2.4.3 Semi-supervised learning**

Ein weiterer Ansatz namens Semi-supervised learning (Halbüberwachtes Lernen) wird in Witten, Pal, Frank & Hall (2017) vorgestellt. Das Ziel des Ansatzes ist eine Klassifizierung, jedoch werden hier sowohl neben den erforderlichen gekennzeichneten zusätzlich auch ungekennzeichnete Eingabedaten verwendet. Es werden allerdings gekennzeichnete Daten benötigt, um diese den Klassen richtig zuzuordnen zu können. Der Grund für die Verwendung von beiden Datensätzen liegt in der aufwendigen Kennzeichnung der Daten. Ein gegebenes Beispiel ist die Erkennung von Berühmtheiten im Fernsehen. Stundenlanges Material könnte dabei aufgenommen werden, aber die Kennzeichnung der Personen müsste dennoch manuell erfolgen.

Das Ziel von Semi-supervised learning ist eine bessere Vorhersage der Klasse zu ermöglichen im Gegensatz zum Modell, wo nur ausschließlich gekennzeichnete Daten verwendet werden. Ein weiteres Beispiel hierfür ist die Art wie Menschen lernen. Ein Kind weiß anfangs viele

Sachen nicht (ungekennzeichnete Daten). Ein Vater erklärt dem Kind diese Sachen und das Kind lernt somit (gekennzeichnete Daten). (Mohammed, Bashier & Khan, 2017)

#### **2.4.4 Reinforcement learning**

Reinforcement learning benötigt die Information, wann eine errechnete Antwort falsch ist. Es wird aber nicht gesagt, welcher Fehler gemacht wurde und wie man diesen Fehler ausmerzen kann, um eine richtige Antwort zu erhalten. Der Algorithmus findet den Weg, um eine richtige Antwort zu erhalten mithilfe von mehreren Versuchen selbst heraus. (Marsland, 2015)

Klassische Anwendungsgebiete für Reinforcement learning sind in Bonaccorso (2018) beschrieben. Diese umfassen folgende Bereiche:

- Automatische Robotersteuerung
- Lösen von Spielen
- Analyse von Aktienhandel basierend auf Basis von Feedback

Anhand der Beschreibung und Anwendungsgebiete lässt sich daraus schließen, dass Reinforcement learning auf den ersten Blick eher ungeeignet ist zur Lösung der Aufgabenstellung dieser Arbeit. Anders sieht es bei den anderen Arten aus. Supervised learning kann für Bildklassifizierung verwendet werden. Auch Supervised learning wäre eine mögliche Option, da Ähnlichkeiten erkannt werden können, aber auch eine automatische Kennzeichnung erfolgen kann. Eine weitere Möglichkeit wäre es Semi-supervised learning zu verwenden, insofern die vorhandenen Produktdaten nicht bereits ausreichend gekennzeichnet sind. Dies ist daher abhängig von den Daten. Möglicherweise ist auch eine Kombination der unterschiedlichen Arten sinnvoll. Im Rahmen der weiteren Arbeit, wenn klar ist, welche Daten vorhanden sind, wird sich zeigen, welche Arten sinnvoll sind.

## **2.5 Algorithmen**

In diesem Kapitel soll ein kurzer Überblick über verschiedene Algorithmen gegeben werden. Der Fokus liegt auf Supervised und Supervised learning Algorithmen. Die Algorithmen werden nur namentlich erwähnt, da dies sonst den Rahmen dieser Arbeit sprengen würde. Sind diese jedoch für die weitere Arbeit von Relevanz, werden diese bei der jeweiligen konkreten Verwendung genauer beschrieben. Eine erste Übersicht erleichtert jedoch das Finden von Algorithmen für die Anwendung auf die Aufgabenstellung dieser Arbeit.

In Ajay (2017) ist eine Übersicht über ML Algorithmen gegeben. Diese sieht wie folgt aus:

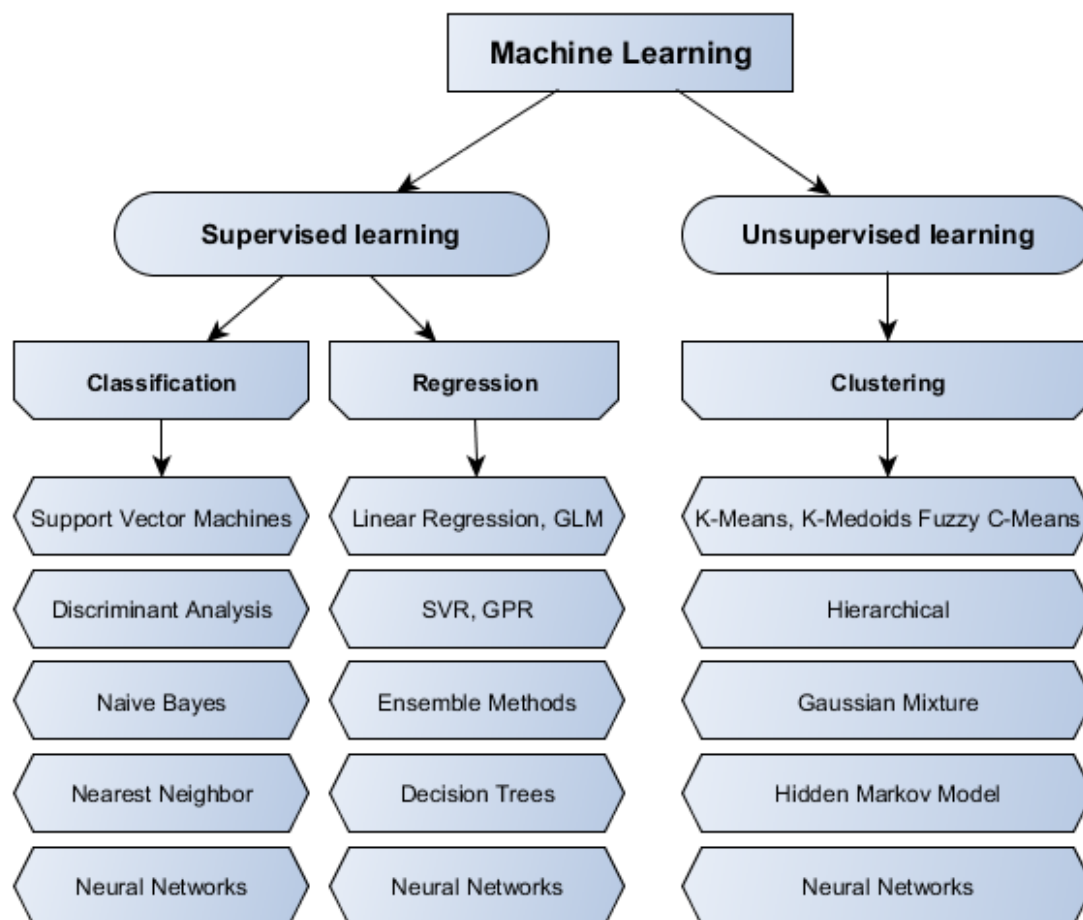


Abbildung 2-1: Übersicht von Machine Learning Algorithmen (vgl. Ajay, 2017)

Die in Abbildung 2-1 dargestellte Übersicht von ML Algorithmen beinhaltet die Trennung von Supervised learning Algorithmen und Unsupervised learning Algorithmen. Supervised learning Algorithmen unterteilen sich nochmals in Klassifikations- und Regressionsalgorithmen. Vertreter von Klassifikationsalgorithmen sind Support Vector Machines, Discriminant Analysis, Naive Bayes, Nearest Neighbor und Neural Networks. Genannte Regressionsalgorithmen sind Linear Regression, Generalized Linear Models (GLM), Support Vector Regression (SVR), Gaussian Process Regression (GPR), Ensemble Methods, Decision Trees und Neural Networks. Unsupervised learning beinhaltet Clustering Algorithmen wie K-Means, K-Medoids, Fuzzy C-Means, Hierarchical, Gaussian Mixture, Hidden Markov Model und Neural Networks. (Ajay, 2017)

Welche von diesen Algorithmen in dieser Arbeit von Relevanz sind, wird sich im weiteren Verlauf zeigen.

## 2.6 Prozess

Dieses Kapitel befasst sich mit dem Prozess zur Umsetzung einer ML Aufgabenstellung. Es wird versucht, eine strukturierte Vorgehensweise zu finden, um die in dieser Arbeit gegebene Aufgabenstellung zu lösen.

Die Analyse von großen Datenmengen ist unter dem Begriff Data Mining zusammengefasst. Anhand eines Einkaufsprozesses wird dies veranschaulicht. Die Verarbeitung eines Einkaufes wird mithilfe von Barcodes vereinfacht. Preise für Produkte können anhand dessen sehr schnell ermittelt werden. Auch andere Informationen über das Produkt können somit sehr schnell eingesehen werden. (Olson, 2018)

In Wirth & Hipp (2000) wird ein Prozess für Data Mining beschrieben. Data Mining wird als kreativer Prozess angesehen, in welchem der Erfolg von den ausführenden Personen abhängig ist. Um solche Data Mining Projekte umzusetzen wurde ein Standard Prozessmodell eingeführt. Dieses trägt den Namen Cross Industry Standard Process for Data Mining (CRISP-DM). Die Verwendung dieses Modells verfolgt die folgenden Ziele:

- Ausführung von großen Data Mining Projekten
- Kosteneinsparungen
- Erhöhung der Zuverlässigkeit
- Erhöhung der Wiederholbarkeit
- Bessere Übersicht
- Schnellere Umsetzung

CRISP-DM besteht aus mehreren Schritten. Die nachfolgende Abbildung zeigt diese Schritte und deren Wechselwirkungen auf:

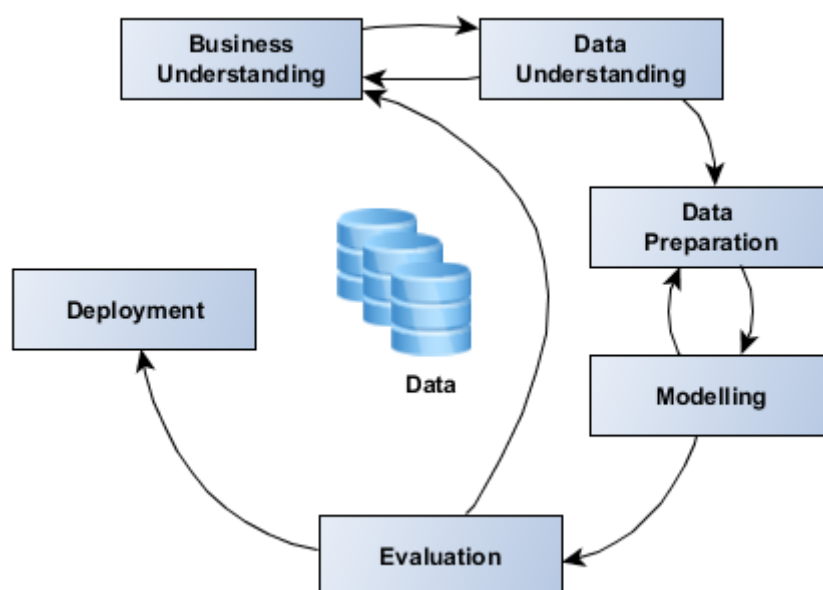


Abbildung 2-2: CRISP-DM (vgl. Wirth & Hipp, 2000)

In Abbildung 2-2 ist der gesamte CRISP-DM Prozess ersichtlich. Zwischen den einzelnen Phasen existieren Abhängigkeiten. Das Ergebnis einer Phase stellt die Eingabe für die darauffolgende Phase dar. (Wirth & Hipp, 2000)

In Cirillo (2017) sind die einzelnen Phasen beschrieben. Die Phase Business Understanding beschäftigt sich mit der Suche nach den Gründen des Business für das Data Mining Problem. Weiters werden in dieser Phase die Ziele für das Projekt festgelegt. Investiert man hier nicht genügend Zeit und Sorgfalt, kann dies Ergebnisse nach sich ziehen, welche nicht von Relevanz sind. Außerdem wäre es möglich das eigentliche Kernproblem nicht zu lösen. Deshalb wird empfohlen einen Projektplan zu erstellen.

Data Understanding befasst sich mit der Beschaffung der initialen Daten. Es wird versucht sich ein Verständnis über die Daten aufzubauen. Ist dies erfolgt, kann mit der Identifikation von Qualitätsproblemen innerhalb dieser Daten begonnen werden. Weiters können hier bereits interessante Muster ersichtlich sein, welche wiederum für eine Hypothesenbildung verwendet werden können. (Wirth & Hipp, 2000)

Nachfolgend kann damit begonnen werden, die Daten für die weitere Verarbeitung zu bearbeiten. Dies erfolgt in der Phase Data Preparation. Die Daten werden dabei gereinigt, gefiltert und transformiert. Das Ziel ist die Daten so aufzubereiten, dass diese für das Modell verwendet werden können. (Cirillo, 2017)

In der Phase Modelling wird versucht Wissen aus den vorbereiteten Daten zu generieren. Das Ziel ist die Beantwortung der Problemstellungen, welche innerhalb der ersten Phasen erstellt wurden. Diese Phase wird durchaus öfter durchlaufen, um Ergebnisse zu erzielen. Wichtige Schritte, welche innerhalb dieser Phase durchlaufen werden, sind die Definition einer Modelling Strategie und die Bewertung des Modells selbst. (Cirillo, 2017)

Die Phase Evaluation beschäftigt sich mit der Validierung der Ergebnisse aus dem Modell. Wie gut performt das Modell für Vorhersagen? Wurde die eigentliche Frage aus der ersten Phase beantwortet? Hierfür sind Metriken notwendig, um die erste Frage zu beantworten. (Cirillo, 2017)

Die letzte Phase trägt den Namen Deployment. Hier wird das Wissen aus dem Modell verwendet und so aufbereitet, dass es nutzbar wird. Die Inbetriebnahme kann unterschiedliche Komplexitätsstufen einnehmen. Basierend auf der jeweiligen Anforderung muss man sich Gedanken machen wie man Nutzen aus dem Modell ziehen kann. (Wirth & Hipp, 2000)

Ein weiterer Prozess wird in Kamath & Choppella (2017) beschrieben. Dieser ist an den CRISP-DM angelehnt. Er beinhaltet zusätzlich noch Machine Learning spezifische Erweiterungen für Machine Learning selbst. Dieser Prozess sieht folgendermaßen aus:

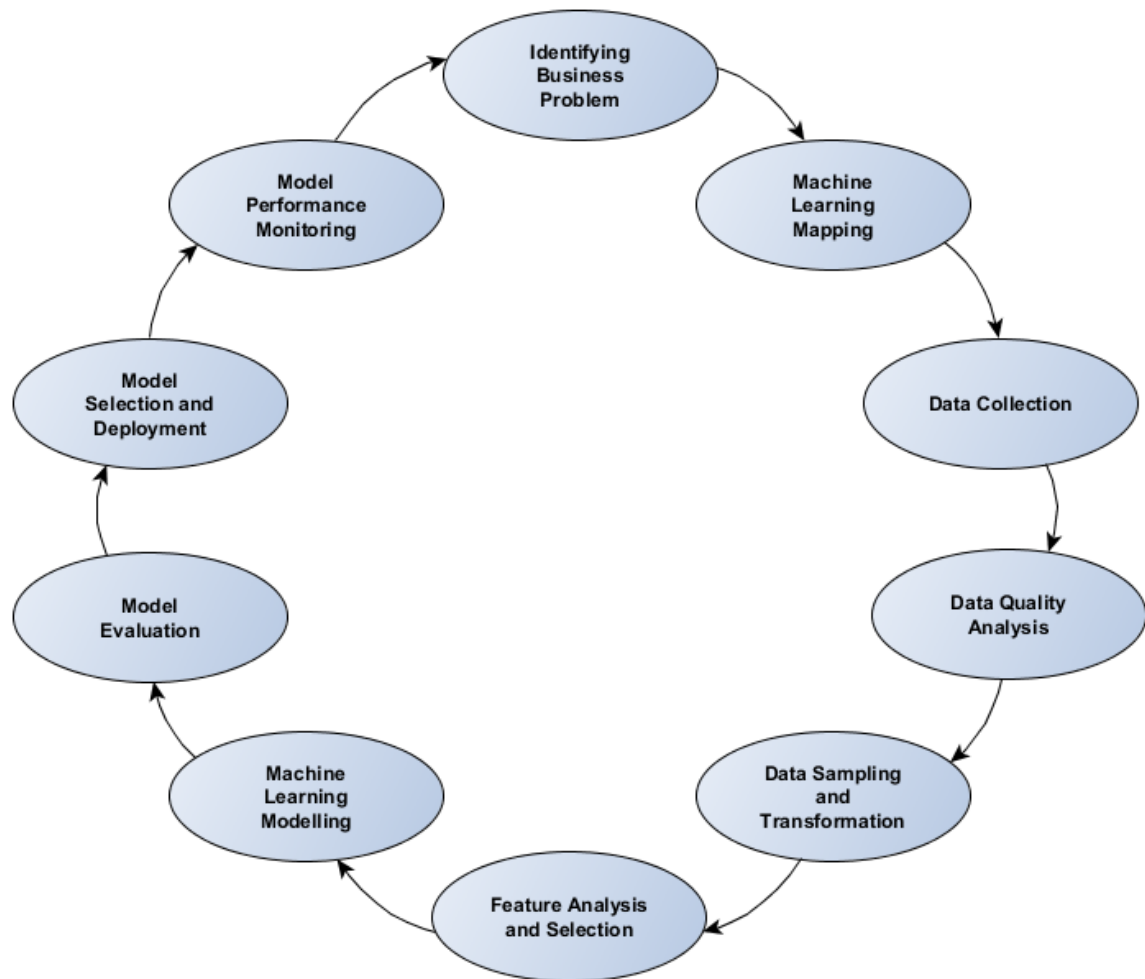


Abbildung 2-3: Machine Learning Prozess (vgl. Kamath & Choppella, 2017)

Abbildung 2-3 stellt eine weitere Definition eines ML Prozesses dar. Dieser ist angelehnt an das CRISP-DM und besteht aus zehn Phasen. Diese sind jeweils in den Ellipsen dargestellt. Die Pfeile sagen aus, dass dies ein iterativer Prozess ist. Die Phasen in der richtigen Reihenfolge dargestellt lauten wie folgt:

- Geschäftsproblem identifizieren (Identifying Business Problem)
- Machine Learning Zuordnung (Machine Learning Mapping)
- Datensammlung (Data Collection)
- Qualitätsüberprüfung der Daten (Data Quality Analysis)
- Datenauswahl und Transformierung (Data Sampling and Transformation)
- Feature Analyse und Auswahl (Feature Analysis and Selection)
- Machine Learning Modellierung (Machine Learning Modelling)



- Modellevaluierung (Model Evaluation)
- Modellauswahl und Deployment (Model Selection and Deployment)
- Performanceüberwachung des Modells (Model Performance Monitoring)

Wie bereits ersichtlich gibt es Überschneidungen mit dem CRISP-DM Prozess. Eine Ergänzung stellt die Phase Machine Learning Zuordnung dar.

In der Phase Machine Learning Zuordnung wird versucht das Problem aus der Phase Geschäftsproblem identifizieren den ML Typen zuzuordnen. Dies wird meist von ML Experten durchgeführt und anhand dieser Zuweisung wird entschieden, welche Arten von ML zum Einsatz kommen. (Kamath & Choppella, 2017)

Die Phasen Datensammlung, Qualitätsüberprüfung der Daten und Datenauswahl und Transformierung kommen wiederum im CRISP-Modell in den Phasen Data Understanding und Data Preparation vor. Eine weitere Ergänzung stellt die Phase Feature Analyse und Auswahl dar.

Innerhalb dieser Phase wird versucht Features zu finden, diese zu transformieren und zu filtern, sodass diese im Modell verwendet werden können. (Kamath & Choppella, 2017)

Die Phase Machine Learning Modellierung befasst sich mit der Auswahl und Verwendung von verschiedenen Algorithmen. Es werden Hypothesen generiert, Parameter verändert und Ergebnisse abgelesen, um ein geeignetes Modell zu erstellen. (Kamath & Choppella, 2017)

Die Phasen Modellevaluierung, Modellauswahl und Deployment und Performanceüberwachung des Modells kommen ebenso im CRISP-DM Modell vor. Diese sind allerdings in den Phasen der Evaluierung und Deployment versteckt.

Für die Aufgabenstellung dieser Arbeit sind beide Prozessmodelle geeignet. Das CRISP-DM Modell ist ein Standard Modell für Data Mining. Der für ML maßgeschneiderte Prozess gibt die Phasen spezifischer an, als es im Gegensatz zum CRISP-DM Modell der Fall ist. Aufgrund dieser Tatsache wird innerhalb dieser Arbeit der für ML maßgeschneiderte Prozess zum Lösen der Aufgabenstellung verwendet.

## 2.7 Messkriterien

Performancemessung und Evaluierung sind wichtige Phasen, um das Modell zu bewerten und um eine Ergebnisverbesserung zu erzielen. Hierfür gibt es mehrere Messkriterien anhand deren ein Modell bewertet werden kann. Für die Bewertung der Varianten im Rahmen dieser Arbeit werden entsprechende Messkriterien benötigt. In diesem Kapitel werden einige davon aufgelistet.

In Smith (2018) wird anhand des Beispiels einer Betrugserkennung eine sogenannte **Confusion Matrix** vorgestellt. Eine Confusion Matrix sieht wie folgt aus:

		Predicted	
		True	False
Actual	True	True Positives	False Negatives
	False	False Positives	True Negatives

Abbildung 2-4: Confusion Matrix (vgl. Smith, 2018)

Abbildung 2-4 zeigt den Aufbau einer Confusion Matrix. Am Beispiel einer Betrugserkennung zeigt der Wert in True Positives alle korrekt vorhergesagten Betrugsfälle des Modells. Alle nicht korrekt vorhergesagten Betrugsfälle zeigt der Wert False Positive an. Der Wert in True Negatives zeigt alle korrekten Vorhergesagten normalen Transaktionen an, welche keine Betrugsfälle sind. Hingegen liefert der Wert in False Negatives alle vom Modell falsch vorhergesagten normalen Transaktionen, welche aber eigentlich Betrugsfälle wären. (Smith, 2018)

Auf Basis des Aufbaus einer Confusion Matrix kann man ableiten, dass die Werte in der Hauptdiagonale (von links oben nach rechts unten) möglichst hoch sein sollten, während die restlichen Werte möglichst gering sein sollten.

Die Werte aus der Confusion Matrix ermöglichen weitere Berechnungen zur Bewertung von Modellen. Ein Wert stellt die sogenannte **Precision** dar. Die Precision wird berechnet als die Anzahl an True Positives dividiert durch die Summe aller positiven Vorhersagen (True Positives + False Positives). Anhand des Beispiels zur Betrugserkennung würde eine niedrige Precision bedeuten, dass wenig Betrugsfälle erkannt werden. (Smith, 2018)

Die Formel zur Berechnung der Precision ist laut Smith (2018) wie folgt aufgebaut:

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Ein weiterer Wert ist der **Recall**. Er wird auch True Positive Rate genannt. Er ist definiert als die Anzahl der True Positives dividiert durch die Summe aller positiven Erkennungen (True

Positives + False Negatives). Ein niedriger Recall Wert würde bedeuten, dass es leicht wäre zu betrügen. (Smith, 2018)

Die Formel zur Berechnung des Recalls ist laut Smith (2018) wie folgt aufgebaut:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Die **False Positive Rate** gibt an wie hoch die Chance ist, dass das Modell True zurückliefert, obwohl die eigentliche Antwort False ist. Sie ist definiert durch die Anzahl an False Positives dividiert durch die Summe der negativen Antworten (True Negatives + False Positives). (Smith, 2018)

Die Formel zur Berechnung der False Positive Rate ist laut Smith (2018) wie folgt aufgebaut:

$$\text{False Positive Rate} = \frac{\text{False Positives}}{\text{True Negatives} + \text{False Positives}}$$

In Smith (2018) wird außerdem die **Receiver-Operating characteristic (ROC) Kurve** beschrieben. Diese gibt das Verhältnis zwischen True Positive Rate und False Positive Rate an. Die **Area under the curve (AUC)** gibt an wie gut das Modell ist. Je höher der Wert, desto besser das Modell. Ein Modell könnte verworfen werden, insofern der AUC Wert unter 0,5 liegen würde, da dies sogar schlechter als ein zufälliges Modell wäre. Folgende Abbildung zeigt wie diese aussehen kann:

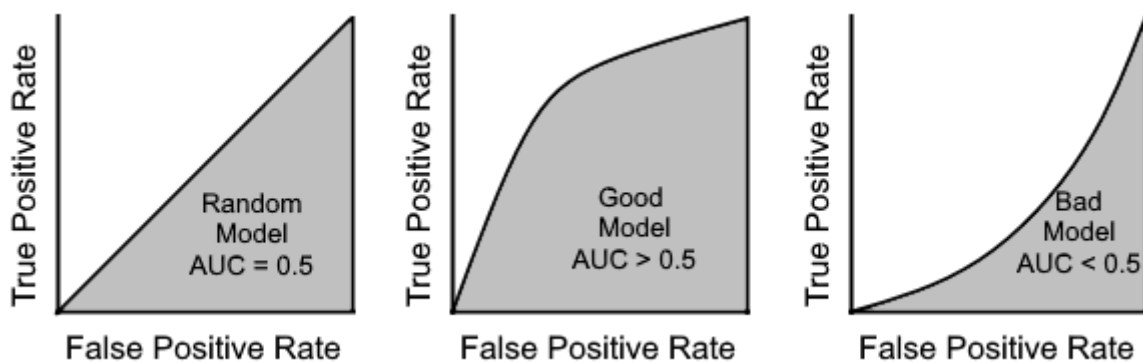


Abbildung 2-5: ROC Kurve mit AUC (in Anlehnung an Smith, 2018)

Abbildung 2-5 zeigt drei Graphen, um die Funktionsweise der ROC Kurve und dem AUC zu verdeutlichen. Der erste Graph zeigt ein zufälliges Modell. Der AUC Wert liegt hier bei 0,5. Der mittlere Graph zeigt eine ROC Kurve, welche angibt, dass das Modell ein Gutes ist. Hier liegt der AUC Wert über 0,5. Der letzte Graph in der Abbildung zeigt ein schlechtes Modell. Hier liegt der AUC Wert unter 0,5. (Smith, 2018)

Weitere Werte zur Bewertung von Modellen werden in Kumar (2019) veranschaulicht. Einer davon ist die **Accuracy**. Diese gibt in Prozent an wie oft korrekt vorhergesagt wird. Sie ist definiert durch die True Positives addiert mit True Negatives und deren Summe dividiert durch die Anzahl der vorhandenen Fälle. Die **Misclassification Rate** ergibt sich ebenso durch die

berechnete Accuracy indem einfach 1 minus Accuracy gerechnet wird. Die Formeln für beide Werte beschrieben in Kumar (2019) sehen wie folgt aus:

$$Accuracy = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total cases}}$$

$$Missclassification Rate = 1 - Accuracy$$

Die **Prevalence** gibt an wie viele True Fälle gegeben sind und wird durch die Anzahl der True Fälle dividiert durch die Anzahl aller Fälle. (Kumar, 2019)

Die folgende Formel in Kumar (2019) zeigt die Berechnung der Prevalence:

$$Prevalence = \frac{\text{Actual: Yes}}{\text{Total Sample}}$$

Die **Null error rate** gibt an, wie falsch das Modell liegen würde, insofern nur die Hauptklasse vorhergesagt werden würde. Sie wird berechnet durch die Anzahl aller False Fälle dividiert durch die Anzahl aller Fälle. (Kumar, 2019)

Die folgende Formel in Kumar (2019) zeigt die Berechnung der Null error rate:

$$Null\ error\ rate = \frac{\text{Actual: No}}{\text{Total Sample}}$$

**Cohen's Kappa value** ist ein Wert zur Berechnung wie gut ein Modell klassifiziert hat. Dieser setzt in Relation wie das Modell zufälligerweise die Klassifizierung vorgenommen hätte. (Kumar, 2019)

Der **F-Score** arbeitet mit Recall und Precision. Ein Wert gegen 1 bedeutet, dass das Modell gut ist, ein Wert gegen 0 jedoch bedeutet, dass das Modell nicht gut ist. Der Wert ergibt sich aus der Multiplikation von 2 mit Recall und Precision dividiert durch die Multiplikation von Recall addiert mit der Precision. (Kumar, 2019)

Die folgende Formel in Kumar (2019) zeigt die Berechnung des F-Score:

$$F - Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

## 2.8 Neuronale Netze

Ein gutes Beispiel für den Einsatz von neuronalen Netzen ist die Bildklassifizierung. Hier geht es darum Bildern eine richtige Kategorie zuzuweisen. Über die Jahre ist die Accuracy in diesem Bereich angestiegen und befindet sich mittlerweile bei 97,7 Prozent. Dass neuronale Netze immer mehr an Bedeutung gewinnen liegt an mehreren Faktoren. Immer schneller werdende Rechner, die Verwendung der Grafikkarte für schnellere Berechnungen und bessere Algorithmen sind beste Voraussetzungen dafür. Auch die ständig wachsende Anzahl an Daten kann mit neuronalen Netzen verarbeitet werden. (Vasilev, 2019)

Was kann unter einem neuronalen Netz verstanden werden?

Eine Beschreibung von neuronalen Netzen ist in Patterson & Gibson (2017) gegeben. Diese lautet wie folgt:

*“Neural networks are a computational model that shares some properties with the animal brain in which many simple units are working in parallel with no centralized control unit. The weights between the units are the primary means of long-term information storage in neural networks. Updating the weights is the primary way the neural network learns new information.” (Patterson & Gibson, 2017, S. 41)*

Anhand dieser Beschreibung kann man bereits sehen, dass ein neuronales Netz aus mehreren Komponenten besteht, welche miteinander in Beziehung stehen.

In Vasilev (2019) wird ein neuronales Netz als System welches Informationen und Eingaben verarbeitet beschrieben. Die Informationsverarbeitung erfolgt über sogenannte **Neurons**. Diese sind miteinander verbunden und tauschen über sogenannte **Connection Links** Informationen aus. Je nach Stärke des Connection Links erfolgt die Informationsverarbeitung. Jedes Neuron hat einen **eigenen Speicher**, welcher von all den eingehenden Verbindungen anderer Neurons beeinflusst wird. Das Ausgabesignal eines Neurons ist über eine sogenannte **Activation Function** bestimmt. Diese wird über den eigenen Speicher gerechnet. Ein neuronales Netz kann außerdem aus mehreren **Layern** bestehen. Der Eingabelayer stellt das Datenset und die Anfangsbedingungen dar. Der Ausgabebereich kann eine unterschiedliche Anzahl an Neurons beinhalten. Bei der Klassifizierung stellt somit jedes Neuron im Ausgabebereich eine Klasse dar.

Für ein einfacheres Verständnis wird der Aufbau eines Neurons nochmal grafisch dargestellt. Folgende Abbildung zeigt laut Ravichandiran (2019) ein solches Neuron:

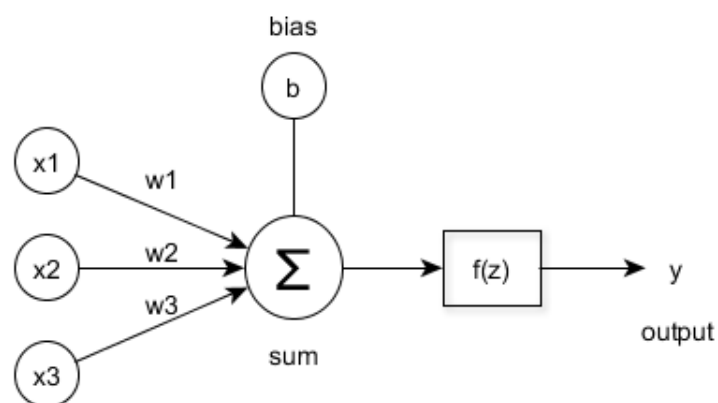


Abbildung 2-6: Aufbau eines Neuron (in Anlehnung an Ravichandiran, 2019)

In Abbildung 2-6 wird die Berechnung des Outputs anhand des Aufbaus dargestellt. Das Neuron bekommt als Eingabe  $x_1$ ,  $x_2$ , und  $x_3$  und multipliziert diese mit den Gewichten  $w_1$ ,  $w_2$ ,  $w_3$ . Diese Multiplikation wird deswegen vollzogen, da ein Eingabewert eventuell mehr Gewicht als ein anderer besitzt. Diese Werte werden anschließend summiert. Ebenfalls wird ein Bias  $b$  hinzugefügt und somit  $z$  erstellt. Anschließend wird eine Activation Function auf  $z$  angewandt  $f(z)$  und daraus erfolgt letztlich der Output  $y$ . (Ravichandiran, 2019)

Bias wird dabei als fiktiver Input verstanden, welcher nicht von den Eingabewerten beeinflusst wird. Er dient dazu die Activation Function zu beeinflussen und in weiterer Folge das Lernen des Modells selbst zu beeinflussen. Der Bias wird zu Beginn des Trainings zufällig initialisiert und wird währenddessen laufend angepasst. (Purkait, 2019)

In Kostadinov (2018) werden die Layer eines neuronalen Netzes aufgezeigt. Diese sind vereinfacht in der folgenden Abbildung ersichtlich:

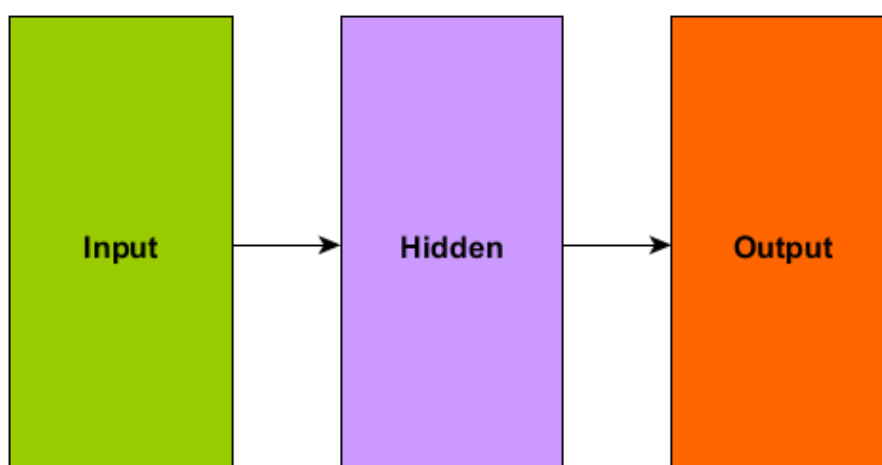


Abbildung 2-7: Neuronales Netz Layer (vgl. Kostadinov, 2018)

Abbildung 2-7 veranschaulicht ein einfaches neuronales Netz, welches aus drei Layern besteht. Einem Eingabelayer, einem oder mehreren Hidden Layern und einem Ausgabelayer. Neuronale Netze mit mehr als einem Hidden Layer werden auch als Deep bezeichnet. (Kostadinov, 2018)

Das Feed-Forward Multilayer Neural Network ist eine der bekanntesten Vertreter von neuronalen Netzen und ist am einfachsten zu verstehen. Es hat einen Eingabelayer, ein oder mehrere Hidden Layers und einen Ausgabelayer. Jedes Neuron eines Layers ist mit jedem Neuron des nachfolgenden Layers verbunden. (Patterson & Gibson, 2017)

Training eines neuronalen Netzes erfolgt über **Backpropagation**. Es wird versucht die Fehler am Ergebnis des Netzwerks zu reduzieren, indem die Gewichtungen der Verbindungen innerhalb des neuronalen Netzes optimiert werden. (Patterson & Gibson, 2017)

## 2.9 Deep Learning

In Patterson & Gibson (2017) ist Deep Learning als ein Teilgebiet von Machine Learning beschrieben. Im Gegensatz zu einfachen neuronalen Netzen bestehen Deep Learning Netze aus mehr als zwei Layern.

Typische Anwendungsgebiete für Deep Learning Algorithmen sind Machine Vision, Natural Language Processing und Reinforcement learning. Unter Machine Vision versteht man die Erkennung von Objekten. Beispiele hierfür sind die Unterscheidung von handgeschriebenen Zahlen oder die Rasse von Hunden, Gesichtserkennung und viele Weitere. (Krohn, Grant & Aglaé, 2018)

Aufgrund der Anwendung von Deep Learning im Machine Vision Bereich ist diese Erkenntnis für diese Arbeit von Relevanz, da damit auch Bildvergleiche möglich sind. Natural Language Processing und Reinforcement learning sind für die Aufgabenstellung nicht relevant und werden daher nicht weiter behandelt.

### 2.9.1 Netzarten

Es gibt unterschiedliche Arten von Deep Learning Netzen. Vier Vertreter werden in Osinga (2018) vorgestellt:

- Fully Connected Networks
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Adversarial Networks und Autoencoders

Bei **Fully Connected Networks** ist jedes Ergebnis zu jeder Eingabe verbunden. Das Ergebnis wird berechnet als gewichtete Summe aller Eingaben. Diese Art von Netzen waren die ersten, welche untersucht wurden. Die in neuronalen Netzen verwendeten Hidden Layer sind nach außen nicht direkt sichtbar. Die optimale Zusammenstellung von Anzahl und Größe der Hidden Layers erfolgt in der Praxis häufig experimentell. (Osinga, 2018)

In Sarkar, Bali & Ghosh (2018) wird der Aufbau eines Deep Neural Network mit fully connected Layer dargestellt:

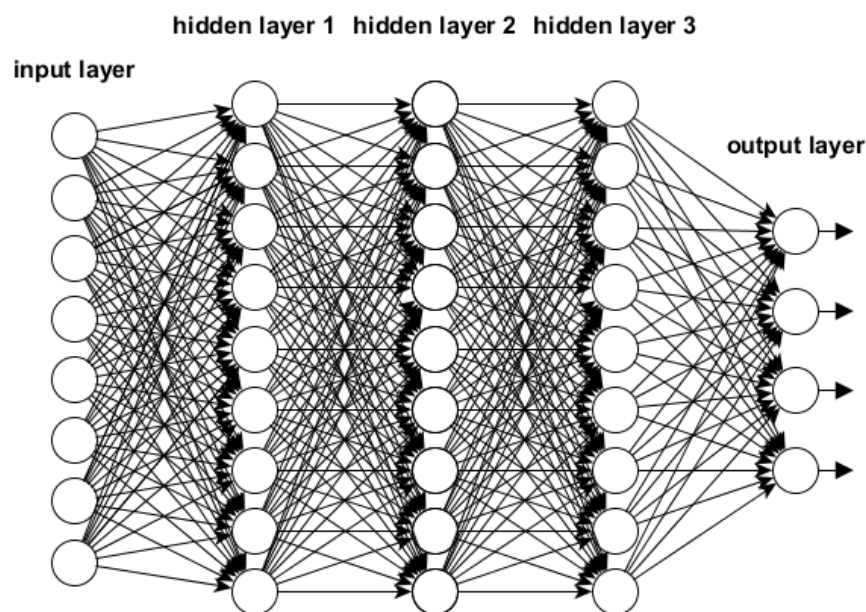


Abbildung 2-8: Deep Neural Network mit Fully Connected Layer (vgl. Sarkar et al., 2018)

Abbildung 2-8 zeigt dabei den Aufbau eines Deep Neural Networks, welche die einfachste Form eines neuronalen Netzes darstellen. Die Neurons sind dabei nach Layern angeordnet und jeder benachbarte Layer ist dabei fully connected. (Sarkar et al., 2018)

**CNNs** spielen ihre Stärke bei der Bildklassifizierung aus. Bei einem großen Bild als Eingabewert würde die Anzahl der Parameter in einem Fully Connected Network sehr hoch sein und ist daher zumeist nicht die optimale Wahl. Außerdem würde dies zu einem Overfitting Problem führen. CNNs ermöglichen die Klassifizierung anhand weniger Parameter. Es wird die sogenannte Convolution angewandt. Es wird dabei nur ein Teilbereich des gesamte Eingabebildes betrachtet und auf diesen ein sogenannter Kernel (auch Filter genannt) angewandt. Das Ergebnis davon ist möglicherweise ein kleineres Bild und verringert somit die Anzahl der benötigten Parameter. Subsampling löst das Problem, dass ein Objekt über das gesamte Bild dargestellt wird. Hierfür gibt es die zwei Ansätze Strided Convolutions und Pooling. Bei Strided Convolutions werden einfach ein oder mehrere Pixel ausgelassen. (Osinga, 2018)

Pooling benutzt sogenannte Pooling Layers. Es wird dabei die Dimension von CNNs reduziert und es wird versucht Overfitting zu reduzieren. Die Featureerkennung wird dadurch stabiler. Bei Pooling wird eine Matrix in nicht überlappende Bereiche aufgeteilt und im Falle von Max Pooling das Maximum aus jedem dieser Bereiche genommen. Das Ergebnis ist eine kleinere Matrix. (Bernico, 2018)

Um eine Vorhersage treffen zu können werden somit mehrere Convolutional Layers und Pooling Layers aneinandergereiht und mit ein oder zwei Fully Connected Layers am Kopf des Netzes verbunden. (Osinga, 2018)

In Kim (2017) wird eine Beispielarchitektur eines CNNs beschrieben. Folgende Abbildung soll die Architektur eines CNNs veranschaulichen:

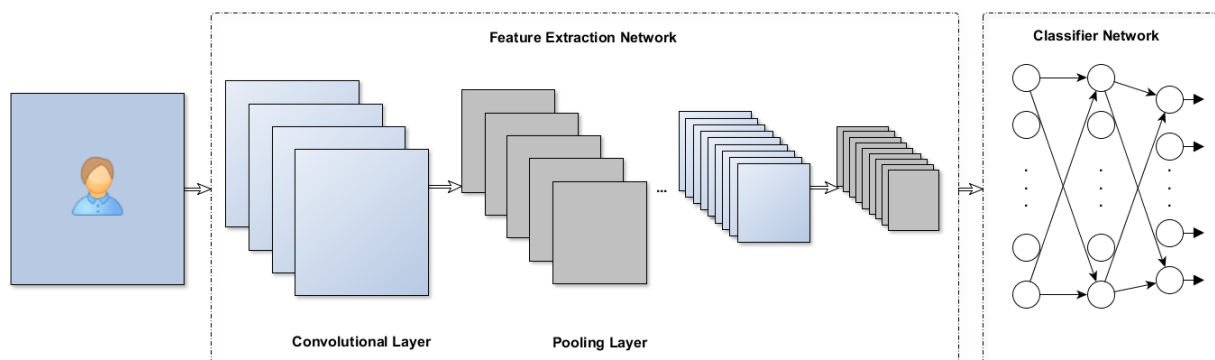


Abbildung 2-9: CNN Architektur (in Anlehnung an Kim, 2017)

In Abbildung 2-9 wird eine Beispielarchitektur veranschaulicht. Das Eingabebild wird an das Featureextrahierungsnetzwerk weitergegeben. Dort werden die Features mithilfe von mehreren zusammenhängenden Convolutional Layer und Pooling extrahiert. Diese Features können dann an ein Klassifizierungsnetzwerk weitergegeben werden. (Kim, 2017)



**RNNs** werden häufig dort eingesetzt wo sequenzielle Daten erwartet werden, wie zum Beispiel Sprachverarbeitung und ermöglichen es mehrere Teilprobleme eines Problems gleichzeitig zu verarbeiten. (Osinga, 2018)

Sogenannte Recurrent Layers ermöglichen es, einen Status zu speichern. Neuronen in einem Recurrent Layer haben auch Verbindungen untereinander. (Buduma & Locascio, 2017)

In Kostadinov (2018) wird der Aufbau eines Recurrent Neural Networks erläutert. Folgende Abbildung zeigt diese vereinfacht dargestellt:

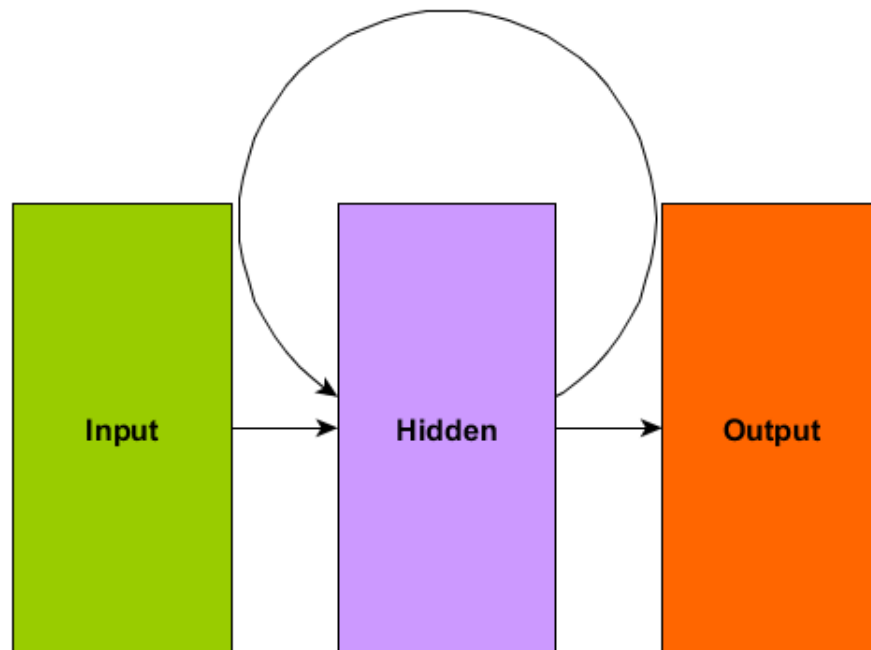


Abbildung 2-10: Recurrent Neural Network Architektur (vgl. Kostadinov, 2018)

Abbildung 2-10 zeigt dabei auf, dass im Gegensatz zu einem einfachen neuronalen Netz eine zyklische Verbindung innerhalb der Hidden Layer gegeben ist. RNNs verwenden sogenannte zyklische Hidden States. Mithilfe derer ist es möglich, alle Informationen eines vorhergehenden Inputs zu halten. (Kostadinov, 2018)

**Adversarial Networks und Autoencoders** verwenden für die jeweilige Problemstellung die passende Struktur. Im Falle von Bildern sind das Convolutions. Sie differenzieren sich in der Art wie diese trainiert werden. Sie sind darauf spezialisiert das Originalbild innerhalb des Netzes wiederherzustellen. Die Hidden Layers im Netz nehmen also eine Kompression des Bildes vor. Dies skaliert aber nur bei kleinen Bildern. Mithilfe von Adversarial Networks wird dieser Umstand gelöst und es werden zwei Netze erstellt. Ein Generator Netzwerk und ein Discriminator Netzwerk. Das Generator Netzwerk produziert ein Bild und das Discriminator Netzwerk überprüft, ob es original ist oder vom Generator Netzwerk stammt. Ein Generator Netzwerk kann somit qualitativ hochwertige Bilder erzeugen. (Osinga, 2018)

Folgende Abbildung zeigt den Aufbau eines Adversarial Networks:

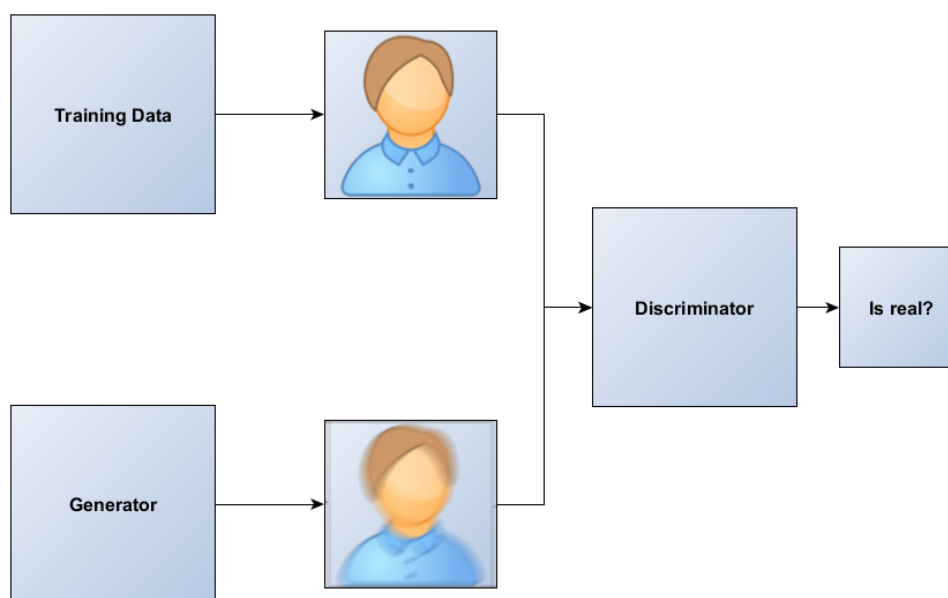


Abbildung 2-11: Adversarial Network Architektur (in Anlehnung an Osinga, 2018)

In Abbildung 2-11 sind die in Osinga (2018) beschriebenen Bestandteile eines Adversarial Networks ersichtlich. Mithilfe eines Generators und eines Diskriminators wird das Trainingsbild überprüft.

In Osinga (2018) wird die Erstellung eines eigenen Typs von Netzen als schwer angesehen. Daher wird empfohlen ein bestehendes Beispiel für den jeweiligen Anwendungsfall zu finden und dieses Schritt für Schritt auf die eigenen Bedürfnisse anzupassen.

Für die Aufgabenstellung in dieser Arbeit sind somit besonders CNNs relevant, da diese eine Bildklassifizierung ermöglichen. Wie die Ähnlichkeit von Produktbildern konkret festgestellt und quantifiziert wird, muss im weiteren Verlauf der Arbeit noch erarbeitet werden.

## 2.9.2 Modell

Eine Definition eines Deep Learning Modells ist in Kumar (2019) gegeben. Diese lautet wie folgt:

*“A deep learning model is a highly non-linear model that has got multiple layers with multiple nodes acting in sequence to solve a business problem. Every layer has been assigned a different task.” (Kumar, 2019, S. 212)*

Die Nicht-Linearität ist mithilfe der Activation Function gegeben. Die Activation Function wird deswegen ausgeführt, damit komplexe Muster innerhalb der Daten erkannt werden können. Würde das nicht gemacht werden, wäre ein Neuron vergleichbar mit einer linearen Regression. (Ravichandiran, 2019)

Die in Kumar (2019) beschriebene Definition weist dabei jedem Layer eine eigene Aufgabe zu. Am Beispiel einer Gesichtserkennung würde der erste Hidden Layer die Kanten innerhalb des

Bildes ermitteln. Der zweite Hidden Layer wäre dafür zuständig, die Kombinationen von Kanten zu ermitteln. In diesem Layer wird bereits damit begonnen die Formen von Augen, Nasen und weiteren Bereichen zu erkennen. Der dritte Hidden Layer würde die Form des Gesichts erkennen.

In Whitenack (2017) wird ein Deep Learning Modell als große und komplizierte Architektur verstanden. Diese benötigt zu Trainingszwecken eine große Anzahl an Daten. Das Ergebnis dieser Struktur ist dabei nur schwer interpretierbar. Generell wird ein Deep Learning Modell aus der Kombination von Fully Connected Networks, Convolutional Neural Networks und/oder Recurrent Neural Networks verstanden.

### 2.9.3 Training

Das Netz startet mit zufälligen Gewichten. An diesem Punkt kann zwar schon eine Vorhersage gemacht werden, hat aber in diesem Stadium noch wenig Bedeutung. Mithilfe von Trainingsdaten, welche als Eingabe für Neurons verwendet werden sowie mittels der Activation Functions wird der Output definiert. Dieser Output wird für den nächsten Layer verwendet. Der Output des letzten Layer stellt schlussendlich die Vorhersage dar. Das Training eines Beispiels vom Eingabelayer bis zum Ausgabelayer wird als *pass* bezeichnet und erfolgt üblicherweise über Batches, welche aus mehreren Trainingsbeispielen bestehen. Das Training eines Batches erfolgt im Rahmen einer sogenannten Iteration. Nachdem ein Batch trainiert wurde, werden die Gewichtungen im Modell angepasst. Das Training mit allen verfügbaren Trainingsdaten wird auch *epoch* genannt. Nachdem mehrere Iterationen und Epochen vollzogen sind, wurden die ursprünglich randomisierten Gewichte innerhalb des Netzes geeignet angepasst und das Modell lernt dadurch korrekte Vorhersagen zu liefern. (Moolayil, 2019)

Optimizer werden laut Moolayil (2019) als wichtigster Part des Trainings eines Modells beschrieben. Diese sind dafür zuständig dem Modell Feedback zu geben. Backpropagation ist einer von diesen Optimierungsalgorithmen. Einer der bekanntesten und am meisten verwendeten Optimizer ist Adam. Ein Optimizer hilft dabei herauszufinden, welche Schritte und Aktualisierungen an den Gewichtungen notwendig sind, um die Verlustfunktion (engl. loss function) zu reduzieren. Diese definiert wie gut oder schlecht die Gewichtungen am gegebenen Trainingsset performen. Der Optimizer hilft dabei zu verstehen, wie viel Änderungen das Netz in der Loss Function sieht, während die Gewichtungen der Neuronen angepasst werden. Die Änderung in der Loss Function hilft dabei die Richtung zu bestimmen, in welche die Gewichtungen angepasst werden müssen. Dies kann eine Erhöhung oder eine Verringerung sein.

In Lane, Howard & Hapke (2019) wird erklärt, dass ein Modell nur so gut ist, wie die Daten anhand derer es trainiert wurde. Ein Begriff, welcher dabei erwähnt wird, ist ein sogenanntes unbalanciertes Trainingsset. In diesem sind in manchen Klassen mehr Objekte verfügbar als in anderen Klassen. Ein Beispiel ist mit der Vorhersage von Hunden und Katzen anhand Bilder gegeben. Bei 20000 Bildern von Katzen und nur 200 Bildern von Hunden würde das Modell fast jedes Eingabebild als Katze vorhersagen. Die Gewichte des Modells werden dabei öfter in Richtung der dominanten Klasse aktualisiert. Generell wird aber empfohlen mehr Daten zu

beschaffen. Da dies nicht immer möglich ist, werden drei Lösungsmöglichkeiten aufgezeigt, um mit diesem Problem in der Praxis umzugehen. Diese lauten wie folgt:

- Oversampling
- Undersampling
- Augmenting

Oversampling reproduziert Bilder aus der unterrepräsentierten Klasse. Am Beispiel der Hunde und Katzen würden die 200 Bilder von Hunden auf 20000 anwachsen. Das Trainingsset würde somit aus 20000 Hundebilder und aus 20000 Katzenbilder bestehen. Das Kopieren der Hundebilder würde jedoch auch bedeuten, dass nur diese 200 Hundebilder gut vorhergesagt werden könnten. (Lane et al., 2019)

Undersampling entfernt Bilder aus der überrepräsentierten Klasse. Es würden zufällig 19800 Katzenbilder entfernt. Es wären somit je 200 Hundebilder und 200 Katzenbilder im Datenset. (Lane et al., 2019)

Augmenting oder auch Image Augmentation genannt wird in Loy (2019) näher erläutert. Dabei werden zusätzliche Trainingsdaten erzeugt, indem Bilder editiert werden. Diese Editierungen können eine Rotation des Bildes, ein Zoom in das Bild oder eine horizontale Spiegelung darstellen.

Im Rahmen dieser Arbeit wird aufgrund der zusätzlich benötigten Rechenleistung keine Augmentierung des Datensets vorgenommen. Dies könnte innerhalb einer weiteren Arbeit erfolgen, um möglicherweise bessere Ergebnisse zu erhalten.

K-Fold Cross Validation beschreibt eine Technik zur Validierung des Modells anhand von Daten, welches es vorher noch nicht gesehen hat. Das Trainingsdatenset wird dabei in sogenannte Folds aufgeteilt. Der Buchstabe  $k$  beschreibt dabei deren Anzahl. Bei jedem Durchlauf wird mit  $k - 1$  Folds trainiert während der eine, nicht verwendete Fold zur Validierung der Performanz verwendet wird. Dieser Vorgang wird  $k$ -Mal wiederholt, sodass jeder Fold einmal zur Validierung verwendet wurde. (Raschka & Mirjalili, 2018)

## 2.9.4 Hyperparametertuning

Parameter, welche nicht direkt von einem Modell gelernt werden, werden auch Hyperparameter genannt. Sogenannte parametric Modelle hingegen lernen die Parameter selbst im Zuge des Trainings der Daten. Non-parametric Modelle lernen diese Parameter hingegen nicht. Hyperparameter dienen dem Modell selbst und können die Performanz eines Modells verbessern. Der Vorgang zur Optimierung dieser Parameter und folglich zur Steigerung der Performanz eines Modells wird dabei Hyperparametertuning genannt. (Fuentes, 2018)

Ein sogenanntes Validierungsdatenset wird dazu verwendet, die Performanz eines Modells zu prüfen. Es werden dabei unterschiedliche Werte innerhalb der Hyperparameter ausprobiert. Das Validierungsdatenset muss dabei unterschiedlich zum Testdatenset sein, da ansonsten anhand des Testdatensets das Modell angepasst werden würde. Anhand der Überprüfung der

verwendeten Evaluierungsmetrik im Modell können die Hyperparameter eingestellt werden. Je besser dabei die Evaluierungsmetrik ausfällt unter der Einstellung eines gewissen Wertes für den Hyperparameter, desto besser ist der Wert für den spezifischen Parameter geeignet. (Fuentes, 2018)

Weiters ist es möglich, dass es mehr als einen Hyperparameter im Modell gibt. Exhaustive Grid Search ist eine Methode, um mit mehreren Parametern umzugehen. Es werden dabei alle möglichen Kombinationen der unterschiedlichen Hyperparameter überprüft. Dieser Vorgang kann dabei sehr rechenintensiv werden. Nach der Berechnung mit unterschiedlichen Kombinationen werden die Ergebnisse verglichen, sodass eine Auswahl für die beste gefundene Parameterkombination getroffen werden kann. (Fuentes, 2018)

### 3 E-COMMERCE

Die Auswahl der Verfahren erfolgt nach der Definition der Begriffe E-Commerce und Produktempfehlungen. Diese Begriffe schaffen die Rahmenbedingungen, welche die Auswahl der zu untersuchenden Verfahren beeinflussen können. Diese werden daher in diesen und dem nächsten Kapitel im Detail erläutert.

E-Commerce kann mehrere Begriffsdefinitionen aufweisen. Im weiteren Sinne kann unter E-Commerce die Durchführung wirtschaftlicher Handlungen auf Basis elektronischer Verbindungen verstanden werden. Im engeren Sinne wird noch zusätzlich die Einschränkung auf die Verwendung des Internets getroffen. Es muss außerdem die Möglichkeit gegeben sein einen Kaufvertrag über das Internet abzuwickeln. (Böing, 2001)

Im Rahmen dieser Arbeit werden auch relevante Anforderungen an E-Commerce Lösungen erläutert, welche Relevanz für die Umsetzung der Verfahren aufweisen können.

In Heinemann (2019) werden neben den typischen Differenzierungsfaktoren drei Basisvoraussetzungen für einen Online-Händler aufgezählt:

- Sortimentskompetenz
- Aktualität
- Servicekompetenz

Sortimentskompetenz beschreibt hier, dass das Online-Produktsortiment dem Sortiment gegenüber Offline-Händlern überragen soll. Als Beispiel kann hierfür Amazon angeführt werden. Aktualität bezieht sich auf Neuheiten und Angebote, um den Kunden zum Wiederbesuch des Shops zu animieren. Die Aktualität steht aber auch für Funktionen, um Inhalt zeitpunktgenau aufschalten zu können. Unter Servicekompetenz wird die Fachberatung auf der Website und die Verwendung von Komponenten wie Kundenbewertungen, technische Produktvergleiche, Wunschlisten, Zubehörlisten und dynamische Produktempfehlungen verstanden. (Heinemann, 2019)

Die Sortimentskompetenz kann somit Einfluss auf die Anzahl der verfügbaren Produkte haben, welche zum Training der Verfahren herangezogen werden. Die Aktualität der Inhalte kann Einfluss auf die Neuberechnung der Produktempfehlungen haben.

Eine weitere Anforderung ist die Ladezeit einer Website. Eine erfolgreiche Website sollte schnelle Ladezeiten aufweisen und einfach verständlich sein. Gründe für lange Ladezeiten sollten vermieden werden. (Reynolds, 2004)

Aufgrund dieser Anforderung ist es möglicherweise vorteilhaft, die Empfehlungen bereits vorberechnen und zwischenspeichern können, um lange Ladezeiten zu vermeiden. Es könnte sogenanntes Caching zum Einsatz kommen.

Caching wird als das Zwischenspeichern von Daten bezeichnet. Diese Daten können dann wiederverwendet werden, ohne die Berechnung nochmal durchführen zu müssen. (Wessels, 2001)

Eine weitere Möglichkeit, um lange Ladezeiten zu vermeiden wäre es die Empfehlungen nach Laden der Produktdetailseite asynchron nachzuladen. Die Berechnung der Empfehlung kann auch wiederum vorberechnet werden insofern das verwendete Verfahren dies unterstützt.

Ein Online-Shop kann unterschiedlich aufgebaut sein. Ein typischer Aufbau einer E-Commerce Lösung wird in Nielsen (2001) beschrieben und gliedert sich in vier Bereiche:

- Kategorieweiten
- Checkout und Registrierung
- Suche
- Produktseiten

Unter dem Begriff Kategorieweiten wird auch die Startseite verstanden und besteht aus mehreren Bereichen. Sie soll wichtige Informationen liefern, wie zum Beispiel welche Produkte bezogen werden können oder nicht. (Nielsen, 2001)

Aufgrund der Einschränkung, dass ein Produkt für die Berechnung der Produktempfehlungen selektiert werden muss, wird nicht näher auf den Bereich der Kategorieweiten eingegangen.

Der Bereich Checkout und Registrierung beschreibt wie Produkte gekauft werden können und KundInnen sich im Shop registrieren können. (Nielsen, 2001)

Auch dieser Bereich hat keinen Einfluss auf die Berechnung der Produktempfehlungen, da die Einschränkung getroffen wurde, dass nur ein Produkt selektiert wird und der Checkout selbst aber möglicherweise mehrere Produkte beinhalten kann. Der Bereich wird somit nicht näher erläutert.

Mithilfe der Suche wird es KundInnen ermöglicht nach spezifischen Produkten innerhalb des Online-Shops zu suchen. Die Suchergebnisse sollen dabei alle relevanten Informationen über das jeweilige Produkt beinhalten. (Nielsen, 2001)

Da auch hier mehrere Produkte vorhanden sein können wird auch hier nicht näher auf diesen Bereich eingegangen.

Eine Definition einer Produktdetailseite ist in Nielsen (2001) beschrieben:

*“A product page is one that is primarily devoted to presenting the details about a single product, including descriptive text, images, and purchasing information, such as availability and price. A product page also includes some mechanism for acquiring the item, such as a Buy button.” (Nielsen, 2001, S. 17)*

In Nielsen (2001) werden Produktseiten anhand von Beispielen näher erläutert und es werden die Bestandteile einer Produktseite wie Produktbilder, Preise, Kosten, Verfügbarkeit und Varianten von Produkten beschrieben.

Der Aufbau einer Produktseite kann aber je nach Online-Shop unterschiedliche Informationen beinhalten.

In Kollwe & Keukert (2014) wird die Produktseite unter dem Namen Produktdetailseite vorgestellt. Auch hier werden verschiedene Bestandteile näher erläutert. Die Bestandteile bestehen aus Produktfotos, Produktvideos, Produkttitel, Produktbeschreibung, Detailinformationen, Varianten, Buybox, Preisangabe, Lieferbarkeit, Lieferfrist, Lagerbestand, Bewertungen, Links, Versandkosten, Zahlungsarten, Produkte in den Warenkorb legen, andere Produkte auf der Detailseite (Cross-Selling) und Weitere.

Der im Rahmen dieser Arbeit wichtigste Bereich ist der Bereich des Cross-Sellings. Die anderen Bereiche werden nicht im Detail erläutert.

Cross-Selling bedeutet, dass neben dem zu verkaufenden Produkt noch weitere Produkte auf der Produktdetailseite abgebildet werden können. Diese Produkte können unterschiedliche Zwecke verfolgen. Ergänzende Produkte sind Produkte die zusätzlich zum eigentlichen Produkt gekauft werden sollen. Ein Beispiel hierfür ist Zubehör zum eigentlichen Produkt. Weiters gibt es beim Cross-Selling sogenannte alternative Produkte. Es könnte sich um das gleiche Produkt nur von unterschiedlichen Herstellern handeln. Es könnte sich aber auch um andere Produkte vom gleichen Hersteller handeln. Das Ziel für eine solche Funktionalität ist es den Absprung des Interessenten aus dem Online-Shop zu verhindern. (Kollwe & Keukert, 2014)

Weitere in Kollwe & Keukert (2014) vorgestellten Cross-Selling Arten lauten wie folgt:

- Kunden, die dieses Produkt gekauft haben, haben auch diese Produkte gekauft
- Zuletzt angesehene Artikel
- Artikel dieser Serie
- Welchen anderen Artikel kauften Kunden, nachdem sie diesen Artikel angesehen haben

Aufgrund der Tatsache, dass alle Cross-Selling Arten außer die der alternativen Produkte weitere Daten, wie zum Beispiel Daten von anderen BenutzerInnen benötigen, werden diese nicht weiter behandelt. Im Rahmen dieser Arbeit wird der Fokus auf die Suche nach alternativen Produkten gelegt, welche sich anhand der Bildrepräsentation ähnlich sind. Somit wäre es möglich, dass dasselbe Produkt nur von unterschiedlichen Herstellern (und möglicherweise mit anderen Preisen) gefunden wird.

Zusammengefasst sind folgende Anforderungen und Rahmenbedingungen für die weitere Arbeit im Rahmen einer E-Commerce Lösung relevant:

- Der Fokus wird auf ähnliche Produktempfehlungen dargestellt auf einer Produktdetailseite gelegt
- Es wird keine Personalisierung berücksichtigt
- Lange Ladezeiten sollen vermieden werden
- Wenn möglich Vorberechnung der Empfehlungen durchführen



- Es werden unterschiedliche Arten von Produkten in unterschiedlichen Kategorien herangezogen
- Die Aktualität der Inhalte und Empfehlungen soll gewährleistet sein

Nachdem die generellen Anforderungen an eine E-Commerce Lösung genauer definiert wurden, kann damit begonnen werden, näher auf den Bereich der Produktempfehlungen einzugehen.

## 4 PRODUKTEMPFEHLUNGEN

Die in dieser Arbeit zu untersuchenden Verfahren sollen in der Lage sein, Produkttempfehlungen im Bereich ähnlicher Produkte anhand eines selektierten Produktes ohne Personalisierung zu ermöglichen. Empfehlungssysteme können aber unterschiedliche Ziele verfolgen. Innerhalb dieses Kapitels wird versucht eine genauere Abgrenzung des Begriffs Produkttempfehlungen vorzunehmen. Es soll im Zuge dessen auch aufgezeigt werden, welche Informationen bezogen auf eine spezifische Empfehlung angezeigt werden sollten. Weiters soll dargelegt werden, welche Vorteile Empfehlungssysteme aufweisen und welches Verfahren zur Ermittlung von Empfehlungen im Rahmen dieser Arbeit verwendet wird.

In Meier & Stormer (2012) ist eine Definition von Produkttempfehlungen gegeben:

*Ein Empfehlungssystem oder Recommender System vergleicht Profile verschiedener Personen und berechnet aus diesen zu empfehlende Objekte. Ein Empfehlungssystem wird in den verschiedensten Kontexten eingesetzt und funktioniert normalerweise vollständig automatisch. Ein bekannter Kontext ist der elektronische Handel. In diesem handelt es sich bei den Personen um Kunden und bei den zu empfehlenden Objekten um angebotene Produkte. (Meier & Stormer, 2012, S. 110)*

Der Automatismus innerhalb dieser Arbeit liegt in der Vorhersage der Produkte innerhalb der einzelnen Verfahren. Produkttempfehlungen werden im Rahmen dieser Arbeit anhand des selektierten Produktes berechnet.

Unterschiedliche Gründe für den Einsatz von Empfehlungssystemen sind auch in Meier & Stormer (2012) dargelegt und können je nach Kontext variieren:

- Zusätzliche Verkäufe
- Kundenzufriedenheit verbessern
- Produkte schneller finden
- Verweildauer im Shop erhöhen
- Personalisierung

Es gibt unterschiedliche Verfahren zur Berechnung von Empfehlungen. Bei **Content-Based Filtering** werden Empfehlungen anhand der Eigenschaften eines Produkts berechnet. **Collaborative Filtering** berechnet Empfehlungen anhand anderer ähnlicher Kunden und berücksichtigt dabei die Bewertung der Empfehlungen von diesen Kunden. Zusätzlich gibt es **hybride Verfahren**, welche Ansätze von Content-Based Filtering und Collaborative Filtering kombinieren. (Meier & Stormer, 2012)

Im Rahmen dieser Arbeit wird der Ansatz des Content-Based Filtering verwendet. Produktbilder werden für die Berechnung der Produktempfehlungen herangezogen. Metadaten für das Produkt werden nicht berücksichtigt, da im Rahmen dieser Arbeit Empfehlungen ausschließlich anhand Bildvergleiche berechnet werden. Personalisierung der Produktempfehlungen ist im Rahmen dieser Arbeit nicht vorgesehen, da Bildvergleichsverfahren analysiert werden sollen.

Die Funktionalität des Hochladens von Bildern zur Suche von ähnlichen Produkten wird innerhalb dieser Arbeit nicht behandelt. Es werden ausschließlich bereits vorhandene Produktbilder zur Berechnung der Empfehlung herangezogen. Dies hat auch Einfluss auf die Auswahl der Verfahren, da Werte zur Berechnung der Empfehlung bereits vorberechnet werden können und nicht erst nach dem Hochladen des Bildes berechnet werden müssen.

Laut Nielsen (2001) sollte ein Suchergebnis eine kurze Beschreibung des Produktes inklusive dem Markennamen, dem Preis des Produkts, ein Produktbild und die Information, ob noch Bestand für ein Produkt verfügbar ist oder nicht beinhalten.

Eine Produktempfehlung könnte mit den oben genannten Produktdaten dargestellt werden.

## 5 BILDANALYSE

Dieses Kapitel befasst sich mit der grundsätzlichen Begriffserklärung der in dieser Arbeit verwendeten Verfahren für Bildvergleiche. Es wird beschrieben, wie eine solches Verfahren funktioniert und welche Ziele damit verfolgt werden sollen. Weiters wird erörtert, welche Metriken es gibt, um ähnliche Bilder beziehungsweise ähnliche Produkte zu finden.

In Shanmugamani (2018) wird das Konzept des Image Retrievals vorgestellt. Deep Learning wird auch als Representation Learning bezeichnet, weil Features während des Modelltrainings gelernt werden. Das Modell lernt dabei Kanten, Muster und viele weitere Merkmale zu erkennen. Dafür kommen unterschiedliche Layer zum Einsatz. Innerhalb der Hidden Layers werden diese visuellen Features gelernt und können verwendet werden, um eine Metrik berechnen zu können, sodass der Abstand zwischen den einzelnen Objekten berechnet werden kann.

### 5.1 Unterschied Bildähnlichkeit und Bildklassifizierung

Zingade (2017) unterscheidet die Bildähnlichkeit von der Bildklassifizierung anhand des nachfolgenden Beispiels. Es werden drei Autos mit unterschiedlichen Farben herangezogen. Diese Farben sind weiß, schwarz und grau. Innerhalb einer Klassifizierung wären diese drei Autos lediglich der Klasse Auto zuzuordnen. Es erfolgt keine Unterscheidung der jeweiligen Farbe. Ein Modell für Bildähnlichkeit berücksichtigt allerdings Farben und weitere Aspekte eines Autos. So könnte bei der Bildähnlichkeit eine Rangordnung der drei Autos anhand ihrer Farben erstellt werden. Würde das Abfragebild das schwarze Auto sein, so würde innerhalb der Rangordnung zuerst das graue und erst danach das weiße Auto zurückgeliefert werden.

### 5.2 Embedding

Ein sogenanntes Embedding (oder auch Bottleneck Features genannt) ist eine Repräsentation von floating-point Werten. Innerhalb eines CNNs werden dabei die wichtigsten Informationen eines Bildes extrahiert. Diese Werte werden beim Training eines CNN geformt, wobei ähnliche Elemente eine niedrige Euklidische Distanz aufweisen, wohingegen unterschiedliche Elemente eine hohe Euklidische Distanz aufweisen. (Kasam, Ganju & Koul, 2019)

## 5.3 Visualisierung

Ein Deep Learning Modell innerhalb eines neuronalen Netzes ist oft schwer für einen Menschen zu interpretieren. Um diese Tatsache zu umgehen, wurden mehrere Methoden entwickelt, um diese zu visualisieren und um zu verstehen, wie so ein Modell funktioniert. (Shanmugamani, 2018)

In Shanmugamani (2018) werden vier Techniken zur Visualisierung vorgestellt:

- Nearest neighbor
- Dimensionality reduction
- Maximal patches
- Occlusion

Nearest Neighbor ist ein Algorithmus, welcher die Distanz zwischen Objekten berechnet. Es gibt verschiedene Metriken, um die Distanz zu berechnen. (Gollapudi, 2016)

Bei Maximal patches wird ein Neuron aktiviert und die entsprechende Stelle mit der maximalen Aktivierung wird verwendet. (Shanmugamani, 2018)

Bei Occlusion kommen Heatmaps zum Einsatz, um zu zeigen welche Bereiche eines Bildes von Bedeutung sind. (Shanmugamani, 2018)

Dimensionality reduction ist eine Technik, um die Anzahl an Features zu reduzieren. Somit wird aus einem hartnäckigen Problem ein leichter zu lösendes Problem. Der Grund für die Einführung liegt darin, dass tausende oder sogar Millionen an Features für jedes Trainingsobjekt involviert sind und somit das Training verlangsamen. Auch die Problemlösung kann damit erschwert werden. Wichtig zu wissen ist, dass bei der Reduzierung von Dimensionalität Informationen verloren gehen. So kann es zwar sein, dass das Training schneller von statten geht, aber durch den Informationsverlust keine guten Ergebnisse zurückgeliefert werden. Für die Datenvisualisierung kommt auch Dimensionality reduction zum Einsatz. Mithilfe dieser Visualisierung können die Ergebnisse somit auch Personen (oder möglicherweise EntscheidungsträgerInnen) visuell vorgelegt werden. (Géron, 2019)

In Shanmugamani (2018) werden zwei Techniken zu Dimensionality reduction näher vorgestellt:

- Principal component analysis (PCA)
- T-Distributed Stochastic Neighbor Embedding (t-SNE)

PCA ist laut Géron (2019) einer der populärsten Dimensionality reduction Algorithmen. Hier wird eine sogenannte Hyperebene anhand der am nächsten gelegenen Daten ermittelt und projiziert die Daten darauf. (Géron, 2019)

T-SNE versucht ähnliche Objekte nahe beieinander zu halten und unterschiedliche Objekte auseinander zu ziehen. (Géron, 2019)

Eine weitere Möglichkeit zur Visualisierung und Dimensionality reduction ist Uniform Manifold Approximation and Projection (UMAP). Es ist schneller und skaliert mit großen Datenmengen besser als t-SNE. Weiters unterstützt es mehrere Distanzmetriken. (Jansen, 2018)

## 5.4 Content-based Image Retrieval

Im Rahmen dieser Arbeit wird wie bereits erwähnt der Fokus auf Content-based Filtering gelegt, weshalb Empfehlungen anhand der Eigenschaften eines Produkts und im Detail anhand des Produktbildes berechnet werden. Aufgrund dieser Tatsache kommt Content-based Image Retrieval zum Einsatz.

Content-based Image Retrieval ist ein Ansatz, bei dem ein sogenanntes Abfragebild als Eingabe verwendet wird und Zielbilder innerhalb einer Datenbank anhand dem Abfragebild geordnet werden, sodass ein Ergebnis zurückgeliefert werden kann. Es wird also mithilfe von einem Bild nach anderen Bildern gesucht. Die Zielbilder mit dem geringsten Abstand vom Abfragebild werden als Ergebnis zurückgeliefert. Die Suche selbst kann dabei sehr zeitaufwendig sein. (Bonaccorso, Fandango & Shanmugamani, 2018)

In Shanmugamani (2018) werden zwei Methoden für eine schnellere Suche vorgestellt:

- Locality sensitive hashing (LSH)
- Multi-index hashing

LSH ermöglicht eine schnellere Suche, indem die Anzahl der paarweisen Vergleiche reduziert wird. Es werden dabei sogenannte Buckets erstellt. Diese Buckets sollen ähnliche Objekte beinhalten. Somit müssen nur bestimmte Buckets durchsucht werden. (Garner, 2015)

Multi-index hashing erzeugt einen Hash aus Features und ermöglicht so eine schnellere Suche. Es wird dabei die sogenannte hamming distance verwendet. Diese untersucht die numerischen Unterschiede der einzelnen Locations der binären Repräsentation. (Shanmugamani, 2018)

Eine Methode zur Modifikation von Suchergebnissen wird auch Re-ranking genannt. Dadurch können bessere Suchergebnisse erzielt werden. Die bereits durch das Abfragebild erhaltenen Zielbilder werden anhand von bestimmten Kriterien neu geordnet. Somit wird eine Optimierung der Ergebnisse vorgenommen. (Shanmugamani, 2018)

Ein weiteres Verfahren für eine schnellere Suche wird in Bonaccorso et al. (2018) vorgestellt und trägt den Namen *approximate nearest neighbor oh yeah* (ANNOY).

ANNOY basiert auf der Verwendung von Bäumen. Es werden Bäume anhand von zufälligen Prognosen erstellt. Die Baumstruktur selbst vereinfacht die Suche nach dem ähnlichsten Objekt. Für die Suche muss ein Index erstellt werden. Dieser benötigt die Dimension der Features. Nach Erstellung des Index werden die Objekte diesem hinzugefügt und der Baum wird gebaut. Je mehr Bäume verwendet werden, desto genauer werden die Ergebnisse. Jedoch geht dies zu Lasten auf die Geschwindigkeit. Ein Vorteil von ANNOY ist, dass der Index im Hauptspeicher gehalten werden kann und darauf optimiert wurde. Ein weiterer Vorteil ist, dass

unterschiedliche Distanzmetriken wie Manhattan, Kosinus oder Euklidische Distanz zur Berechnung der Ähnlichkeit verwendet werden können. (Bonaccorso et al., 2018)

Im Rahmen dieser Arbeit wird ANNOY für die Suche innerhalb jedes Verfahrens verwendet.

## 5.5 Suche nach Bildern

In Bonaccorso et al. (2018) wird die sogenannte Retrieval Pipeline vorgestellt. Diese beschreibt die nötigen Schritte, um von einem Eingabebild die besten Ergebnisse der Zielbilder zu erhalten. Die folgende Grafik soll einen Überblick über die verschiedenen Schritte aufzeigen:

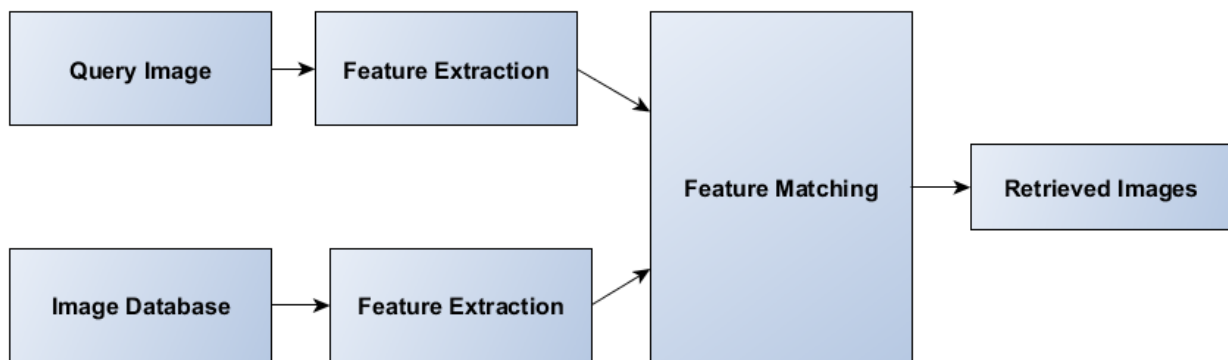


Abbildung 5-1: Retrieval Pipeline (vgl. Bonaccorso et al., 2018)

Abbildung 5-1 stellt den Ablauf grafisch dar. Innerhalb der Schritte werden die Features für die Bilddatenbank extrahiert und innerhalb der Datenbank gespeichert. Aus dem Eingabebild werden ebenso Features extrahiert. Nachdem dies erfolgt ist kann die Ähnlichkeit über alle Zielbilder mithilfe der Features berechnet werden und die Bilder anhand dieser Berechnung für das Ergebnis sortiert werden. (Bonaccorso et al., 2018)

Eine der verwendeten Distanzmetriken, welche innerhalb von Nearest Neighbor zum Einsatz kommt, ist die Euklidische Distanz. Mithilfe dieser kann die Ähnlichkeit zwischen zwei Instanzen berechnet werden. Sind sich die Objekte ähnlich so ist der Wert gering. Weitere Distanzmetriken sind Minkowski, Manhattan und die Kosinusdistanz. (Ciaburro, 2019)

Innerhalb der Retrieval Pipeline werden die Features für die Berechnung der Ähnlichkeit herangezogen. Diese Berechnung müsste allerdings für jedes Bild vollzogen werden. Bei mehreren Millionen Bildern wäre dies somit nicht effizient, da geringe Antwortzeiten im Millisekundenbereich erwartet werden. Mithilfe von ANNOY wird die Distanzsuche beschleunigt. (Bonaccorso et al., 2018)

Zusammengefasst kann mit folgenden Schritten die Bildanalyse erfolgen und ein Grundgerüst für den Aufbau der Struktur der Verfahren erstellt werden:

- Training des Modells
- Extraktion der Features der einzelnen Bilder mithilfe des trainierten Modells
- Speicherung der Features

- Suche nach ähnlichen Bildern indem Features aus dem Abfragebild extrahiert werden
- Vergleich der Features
- Auslieferung der ähnlichen Bilder anhand des Vergleichs



## 6 VERFAHREN FÜR BILDVERGLEICHE

Nachdem grundlegende Begriffe über die Bildanalyse näher erläutert und generelle Kriterien und Anforderungen an das Empfehlungssystem getroffen wurden, kann mit der Erhebung von Verfahren für Bildvergleich begonnen werden. Die einzelnen Kriterien und Anforderungen setzen sich dabei aus generellen Anforderungen für ein Empfehlungssystem und jenen, welche für einen möglichen Einsatzzweck beim Kunden von Relevanz sind zusammen. Diese werden hier nochmal zusammengefasst:

- Der Fokus wird auf die Empfehlung visuell ähnlicher Produkte gelegt, die auf einer Produktdetailseite anzuzeigen sind
- Es wird dabei keine Personalisierung auf einzelne BenutzerInnen einfließen
- Empfehlungen können vorberechnet werden und in einem Speicher zur schnelleren Wiederverwendung gehalten werden
- Der Speicher kann dazu beitragen die Ladezeiten für Empfehlungen zu verkürzen
- Wird ein Produktbild geändert oder kommen weitere Produkte hinzu, so werden die Empfehlungen neu berechnet, um die Aktualität der Empfehlungen zu gewährleisten
- Es werden unterschiedliche Arten von Produkten in unterschiedlichen Kategorien herangezogen
- Die Modelle der Verfahren können mit der Mindestanzahl von Produkten im Datenset trainiert werden

Da es viele verschiedene Verfahren zur Ermittlung von ähnlichen Bildern gibt werden hier nur Ausgewählte einer näheren Betrachtung unterzogen. Es werden insgesamt drei Verfahren im Rahmen dieser Arbeit näher erläutert:

- Autoencoder
- Transfer Learning
- Siamese Network

Ein weiteres Verfahren zur Ermittlung von ähnlichen Bildern wird Deep Ranking genannt und wird in Zingade (2017) vorgestellt. Dieses Verfahren wird im Rahmen dieser Arbeit zwar theoretisch erläutert aber praktisch nicht umgesetzt und fließt somit nicht in die Bewertung mit ein. Der Aufwand zur Umsetzung des Verfahrens würde den Rahmen dieser Arbeit übersteigen. Die dafür benötigten Daten sind nicht in der Form vorhanden, um gute Ergebnisse erzielen zu können. Diese müssten manuell gewartet werden was bei einem Datenset mit vielen Objekten viel Aufwand darstellen würde.

## 6.1 Autoencoder

In Hubens (2018a) wird ein Verfahren zur Ermittlung von ähnlichen Bildern mithilfe von Autoencoder vorgestellt. Das Verfahren beruht auf Content-based Image Retrieval und verwendet daher ausschließlich die Bilddaten zum Training des Netzwerks als auch zur Ermittlung von ähnlichen Bildern. Es kommen folglich keine zusätzlichen Features wie textueller Inhalt zur Anwendung. Der Ansatz wird dem Supervised learning zugeordnet.

Ein Autoencoder ist neuronales Netzwerk, welches versucht die Eingabedaten (in diesem Falle ein Bild) zu Features (auch latente Repräsentation genannt) zu komprimieren und aus dieser Repräsentation das Bild wieder zu rekonstruieren.

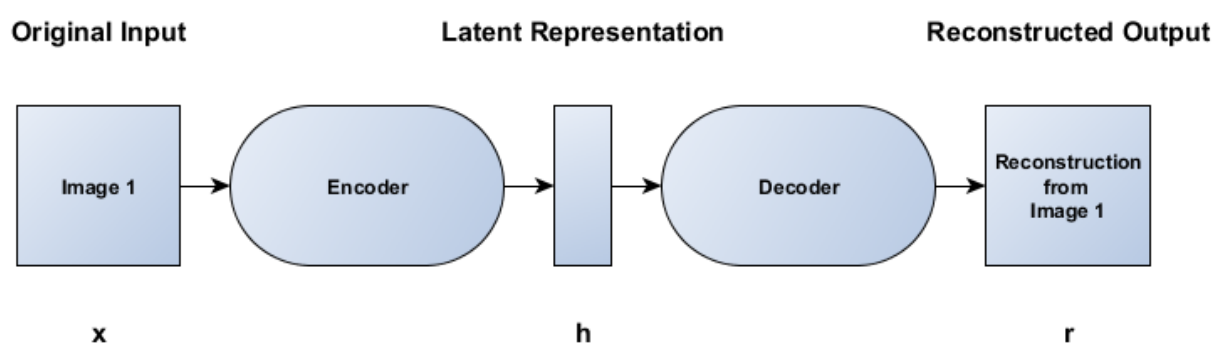


Abbildung 6-1: Autoencoder (in Anlehnung an Hubens, 2018a)

In Abbildung 6-1 werden die Bestandteile eines Autoencoders aufgeführt. Er besteht aus einem Encoder und aus einem Decoder. Beim Encoder wird die Eingabe  $x$  (auch Eingabebild) in eine latente Repräsentation  $h$  gebracht. Der Decoder ist dafür zuständig diese latente Repräsentation wieder in das Eingabebild zu rekonstruieren und das Ausgabebild  $r$  zu generieren. (Hubens, 2018a)

Der Grund für diese Vorgehensweise wird in einem weiteren Artikel in Hubens (2018b) beantwortet und liegt in der Extraktion sinnvoller Features für die weitere Bearbeitung durch Training des Autoencoders. Die latente Repräsentation wird als der Weg beschrieben, den das neuronale Netzwerk gefunden hat, ein Bild zu kodieren. Ähnliche Bilder sollen somit ähnlich kodiert werden. Außerdem sollte darauf geachtet werden, dass die latente Repräsentation eine geringere Dimension als das Eingabebild aufweist. Mithilfe der in der Umsetzung verwendeten Layer ist dies gegeben. Erst so können sinnvolle Features extrahiert werden. Dieser Vorgang, bei dem die latente Repräsentation eine geringere Dimension aufweist, wird auch *undercomplete* genannt. *Overcomplete* ist das Gegenteil und führt dazu, dass die Dimension der latenten Repräsentation größer als das Eingabebild ist. In diesem Falle können keine sinnvollen Features extrahiert.

In dem von Hubens (2018a) beschriebenen Autoencoder Verfahren kommt ein sogenannter Convolutional Autoencoder zum Einsatz.

Ein Convolutional Autoencoder unterscheidet sich von anderen Typen von Autoencodern in seiner Architektur. Es kommen mehrere Hidden Layers innerhalb des Netzes vor. Diese Layer

bestehen aus Convolution, Pooling und Upsampling eines downgesampelten Bildes. (Ayyadevara, 2019)

In Kapitel 2.9.1 wurden Convolution und Pooling Layer bereits behandelt.

Ein Upsampling Layer stellt dabei das Gegenstück zu Pooling dar. Die Dimension wird dabei erhöht anstatt reduziert. (Bonaccorso et al., 2018)

Das in Hubens (2018a) beschriebene Verfahren verwendet ein Trainingsset für das Training des Autoencoders. Der Autoencoder lernt somit die Bilder in eine latente Repräsentation zu kodieren.

Nach dem Training des Modells wird nur mehr der Encoder-Teil des Netzwerkes verwendet. Dieser wird dafür benötigt, um jenes Bild, zu welchem ähnliche Bilder gesucht werden sollen zu kodieren. Auf Basis dieser resultierenden Kodierung kann mittels eines geeigneten Verfahrens nach ähnlichen Bildern gesucht werden. Für die Suche wird innerhalb dieses Verfahrens der Nearest Neighbor Algorithmus verwendet. Die Anzahl der zurückzuliefernden Ergebnisse ist dabei vorgegeben. Für die Berechnung der Distanz wird die Euklidische Distanz verwendet. Diese kann jedoch mit einer beliebigen Distanzmetrik ersetzt werden. (Hubens, 2018a)

Innerhalb dieses Verfahrens wird der Nearest Neighbor Algorithmus für die Suche nach ähnlichen Bildern verwendet. Dieser soll im Zuge der Implementierung des Verfahrens mit einer effizienteren Suche namens ANNOY ersetzt werden.

## 6.2 Deep Ranking

In vielen Verfahren zur Ermittlung von ähnlichen Bildern werden Bilder als ähnlich definiert insofern diese derselben Kategorie angehören. Deep Ranking ermöglicht die Unterscheidung zwischen ähnlichen Bildern innerhalb derselben Kategorie. Diese Unterscheidung wird für den Anwendungsfall einer Suche mithilfe eines Beispielbilds benötigt. Eine entsprechende Kategorisierung sollte daher bereits im Vorhinein gegeben sein. Mithilfe von Deep Learning Modells ist es möglich, anhand eines Abfragebilds nach ähnlichen Bildern zu suchen. Innerhalb dieses Verfahrens kommen sogenannte Triplets zum Einsatz. (Zingade, 2017)

Wie in Zingade (2017) beschrieben besteht ein Triplet aus den folgenden drei Komponenten:

- Abfragebild
- Positives Bild
- Negatives Bild

Die nachfolgende Grafik zeigt ein Beispiel eines Triplets:

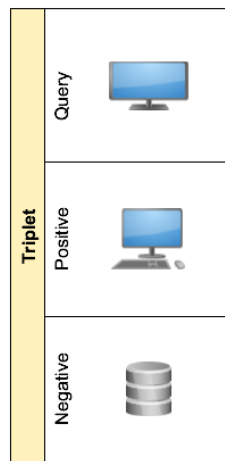


Abbildung 6-2: Triplet (in Anlehnung an Zingade, 2017)

Abbildung 6-2 zeigt dabei ein sogenanntes Triplet mit den unterschiedlichen Komponenten. Es zeigt auch auf, dass sich das Abfragebild (Query) und das positive Bild (Positive) ähnlicher sind als das Abfragebild im Vergleich zum negativen Bild (Negative). (Wang et al., 2014)

Ein Deep Ranking Modell kann aus diesen Triplets lernen, welche Bilder sich aufgrund der Eigenschaften innerhalb des Triplets ähnlich sind und welche nicht. Die Anzahl an Triplets hängt mit der Anzahl an Bildern zusammen. Eine Verwendung aller Triplets würde somit viel Rechenaufwand benötigen. Bei 12 Millionen Bildern müssten somit  $12.000.000^3 = 1.728 * 10^{21}$  Triplets prozessiert werden. Ein passender Algorithmus soll damit zur Auswahl der Triplets verwendet werden, um die Anzahl dieser zu reduzieren. (Wang et al., 2014)

In Wang et al.(2014) wird zur Erstellung der Triplets ein sogenannter Total Relevance Score von einem Bild verwendet. Dieser gibt Aufschluss, wie relevant das Bild im Vergleich zu anderen Bildern innerhalb einer Kategorie ist. Für ein positives Bild wird ein Bild mit hohem Relevance Score innerhalb derselben Kategorie herangezogen. Für ein negatives Bild werden Bilder innerhalb der gleichen Kategorie verwendet, welche aber weniger Relevanz zum Abfragebild aufweisen. Weiters werden Bilder außerhalb der Kategorie als negatives Bild verwendet.

Die verwendete Architektur sieht dabei wie folgt aus:

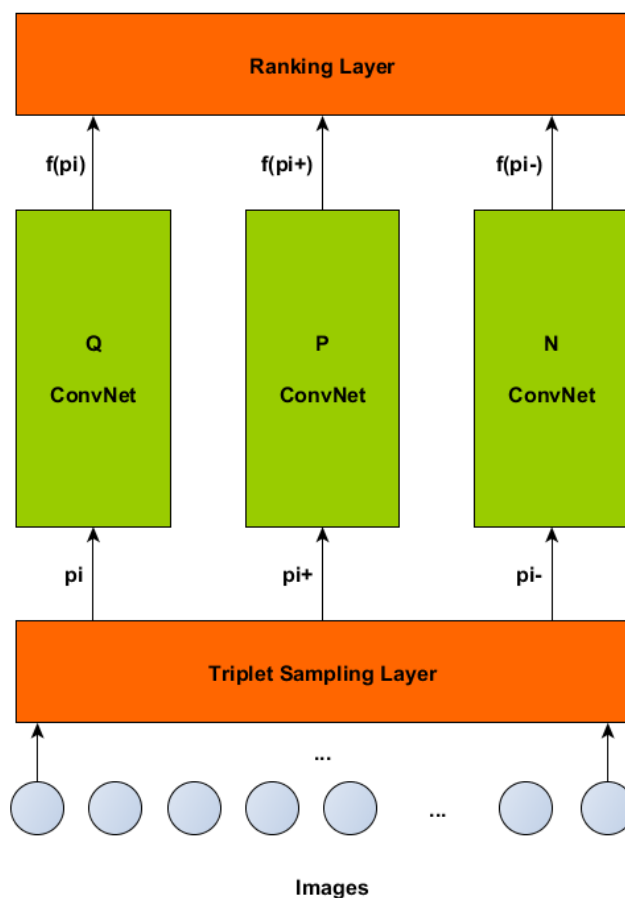


Abbildung 6-3: Deep Ranking Architektur (vgl. Zingade, 2017)

Abbildung 6-3 zeigt die Architektur des Netzwerks auf, welche aus drei Teilen besteht. Dem Triplet Sampling Layer, dem ConvNet und dem Image Similarity Ranking Layer. Der Triplet Sampling Layer bekommt Bilder als Eingabe und erzeugt daraus Triplets. Von einem Triplet sind das Abfragebild als  $p_i$ , das positive Bild als  $p_{i+}$  und das negative Bild als  $p_{i-}$  gekennzeichnet. Die drei ConvNets nehmen jeweils ein Bild. Q nimmt das Abfragebild  $p_i$ , P nimmt das positive Bild  $p_{i+}$  und N das negative Bild  $p_{i-}$ . Aus diesen Bildern wird das Embedding berechnet und an den Ranking Layer weitergegeben. Der Ranking Layer berechnet die sogenannte Hinge Loss eines Triplets. Die Hinge Loss misst dabei die Verletzung der Rangfolge, welche im Triplet definiert ist. Während des Trainings wird evaluiert, inwiefern die Ranking Order verletzt wird. Die berechneten Gradienten werden an die unteren Layer mithilfe von Backpropagation weitergegeben, sodass die Parameter angepasst werden und der Ranking Loss minimiert wird. (Zingade, 2017)

Ein Gradient hilft dabei den Loss innerhalb einer Loss Function zu minimieren. Es werden dabei die Modellparameter berücksichtigt. Diese Modellparameter werden anhand der Berechnung der Gradienten und anhand einer Update-Regel wieder aktualisiert. (Ravichandiran, 2019)

Eine Loss Function ist eine Metrik, welche einem Netzwerk dabei hilft, sich beim Training in die gewünschte Richtung zu bewegen. Als Beispiel dienen Testergebnisse von fünf eigenen Tests

im gleichen Unterrichtsgegenstand. Jene Metrik, welche dabei verwendet wird, ist die Punktzahl. Bei Testergebnissen von 56, 60, 78, 90 und 96 von 100 Punkten ist erkennbar, dass eine Steigerung beim Testergebnis vorhanden ist. Würde die Punktzahl fallen, so müsste etwas geändert werden, wie zum Beispiel die Art der Methode des Lernens oder das Lernmaterial. Als Beispiel am Modell selbst wird nun eine Vorhersage getroffen, inwiefern ein Test von einer Person positiv abgeschlossen wird oder nicht. Der Wert eins würde bedeuten, dass die Person den Test zu 100 Prozent positiv abschließen wird wobei null bedeuten würde, dass der Test mit Sicherheit negativ ausfallen wird. Angenommen das Modell sagt den Wert 0.87 vorher. Der Loss wäre 0.13 ( $1 - 0.87$ ). Bei geänderten Parametern im Modell würde der Loss auf zum Beispiel auf 0.40 steigen. Das Modell würde dadurch erkennen, dass es nicht in die richtige Richtung lernt. Bei einer Änderung vom Loss auf 0.05 wäre dies hingegen schon der Fall. (Moolayil, 2019)

In Wang et al.(2014) wird das Ranking von ähnlichen Bildern auch als Nearest Neighbor Suchproblem dargestellt. Es wird dabei der Euklidische Raum verwendet. Für die Suche kann ein Approximate Nearest Neighbor Algorithmus wie beispielsweise ANNOY zum Einsatz kommen.

### 6.3 Siamese Network

Eine Definition eines Siamese Networks ist in Bonaccorso et al. (2018) gegeben:

*"A Siamese network, as the name suggests, is a neural network model where the network is trained to distinguish between two inputs. A Siamese network can train a CNN to produce an embedding by two encoders." (Bonaccorso et al., 2018, S. 158)*

Mithilfe dieser Technik können CNNs trainiert werden mit denen es möglich ist, Embeddings zu erzeugen. Als Eingabedaten werden jeweils Bilder verwendet wobei jedes CNN ein Bild als Eingabe erhält. Dieses Bildpaar kann positiv oder negativ sein. (Bonaccorso et al., 2018)

Was positive oder negative Bilder sind, wurde bereits in Kapitel 6.2 im Rahmen von Triplets näher erläutert.

Die oben genannte Definition würde darauf schließen lassen, dass nur zwei Bilder einem Vergleich unterzogen werden können und der Anwendungsfall sich ausschließlich auf den Vergleich zweier Bilder konzentriert.

In Appalaraju & Chaoji (2017) wird jedoch auch ein Anwendungsfall aufgezeigt, in welchem ähnliche Bilder anhand eines neuen Bildes ermittelt werden. Dieses Verfahren wird auch SimNet genannt. Für ein Embedding werden hierfür Feature Vektoren gelernt. Es wird dabei eine Distanzmetrik gelernt, um die Ähnlichkeit von Bildern zu berechnen.

Die Auswahl der positiven und negativen Bildpaare ist wichtig, damit eine gute Modell Performanz erreicht werden kann. Für die Auswahl der Paare wird ein Verfahren namens Online Pair Mining Strategy vorgestellt. Für das Training selbst kommt sogenanntes Curriculum Learning zum Einsatz. Bei Curriculum Learning wird ermittelt, welche Beispiele zuerst für das

Training verwendet werden, sodass das Modell schneller bessere Ergebnisse liefert. So werden schwerer unterscheidbare Bilder erst später im Training berücksichtigt. Bei Online Pair Mining Strategy werden diese sogenannten harten Bilderpaare ermittelt. Diese können zwischen der Grenze der positiven und negativen Bilder eingeordnet werden. (Appalaraju & Chaoji, 2017)

Es kommt ein sogenanntes Multi-Scale CNN zum Einsatz, um 4096-dimensional Embeddings von einem Bild zu lernen. Dieses performt besser als gewöhnliche CNNs um ähnliche Bilder zu ermitteln. Innerhalb der beschriebenen Multi-scale Architektur werden Features auf unterschiedlichen Wegen ermittelt und in hinteren Layern wieder zusammengefügt. Für jedes CNN innerhalb des Siamese Network kommt diese Architektur zum Einsatz. (Appalaraju & Chaoji, 2017)

One-Shot Learning ist ein Anwendungsfall von Siamese Networks und wird in Lamba (2019) beschrieben. Es wird als eine Technik beschrieben, wobei nur ein Trainingsbeispiel für eine Klassifizierung benötigt wird. Anstatt einer Klassifizierung werden die beiden Eingabebilder verglichen. Dies hat den Vorteil, dass nur sehr wenig Trainingsdaten pro Klasse verwendet werden müssen.

Weitere Vorteile von Siamese Networks werden in Panchal (2019) beschrieben:

- es werden Ähnlichkeitsfunktionen erlernt
- einfaches Training
- einfache Architektur

Folgende Abbildung zeigt eine Beispielarchitektur eines Siamese Network:

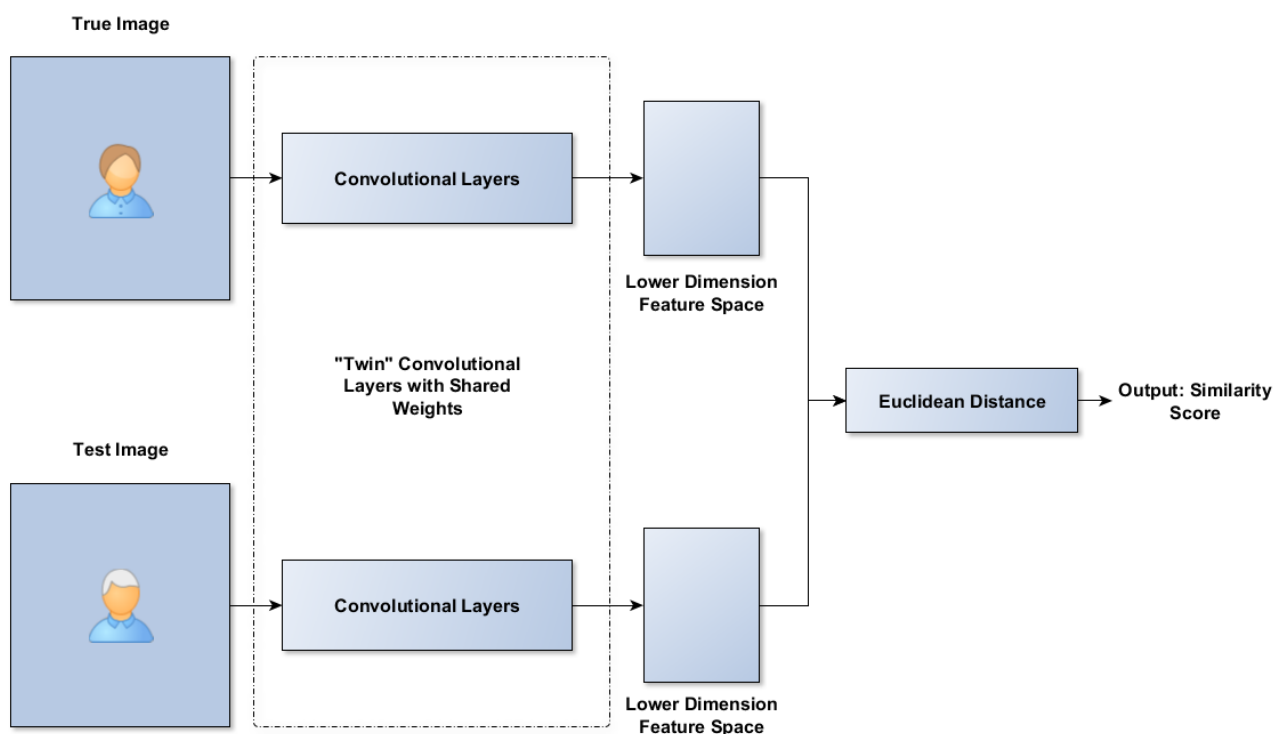


Abbildung 6-4: Siamese Network Architektur (in Anlehnung an Loy, 2019)

Abbildung 6-4 zeigt hier den Aufbau eines Siamese Network. Es werden zwei Bilder (abgebildet als True Image und Test Image) an jeweils zwei unterschiedliche CNNs zur Berechnung der Features gegeben. Anhand der Features wird mithilfe der Euklidischen Distanz der Ähnlichkeitswert berechnet. Wichtig ist, dass die Gewichtungen der CNNs untereinander geteilt sind. Dies wird am Beispiel einer Gesichtserkennung verdeutlicht. Würden diese unterschiedliche Gewichtungen aufweisen, so wäre es möglich, dass gleiche Gesichter auf unterschiedliche Punkte innerhalb des Feature Space berechnet werden würden und die Euklidische Distanz somit nicht sinnvoll wäre. (Loy, 2019)

Aufgrund des Aufwandes wäre es durchaus möglich eine leichtgewichtigere Version dieses Verfahrens umzusetzen. Dies wird im Rahmen der weiteren Arbeit ermittelt.

Auch der Bereich der Abfrage der Bilder selbst kann noch optimiert werden. Ein möglicher Ansatz ist der Einsatz von ANNOY für die Suche selbst anhand der ermittelten Features.

## 6.4 Transfer Learning

Menschen haben die Fähigkeit sich Wissen anzueignen. Wissen kann jedoch auch verwendet werden, um andere Aufgaben zu lösen, welche ähnlich zu den Ursprünglichen sind, aus denen Wissen abgeleitet wurde. Die Ähnlichkeit der Aufgaben spielt eine wichtige Rolle. Je ähnlicher sich Aufgaben sind, desto einfacher ist es, das Wissen auf die jeweils andere Aufgabe anzuwenden. Transfer Learning beschreibt genau diese Idee, Wissen über ähnliche Aufgaben hinweg auf eine andere Aufgabe anzuwenden, um das jeweilige Problem zu lösen. (Sarkar et al., 2018)

Im konkreten Bereich des Machine Learnings wird in Sarkar et al. (2018) folgende Definition des Begriffs verwendet:

*“Transfer learning takes the process of learning one step further and more inline with how humans utilize knowledge across tasks. Thus, transfer learning is a method of reusing a model or knowledge for another related task. Transfer learning is sometimes also considered as an extension of existing ML algorithms.” (Sarkar et al., 2018, S.*

156)

Es werden ML Modelle für bestimmte Aufgaben trainiert und diese dann auf ähnliche Aufgaben transferiert. Transfer Learning bietet wie in Sarkar et al. (2018) beschrieben aufgrund des bereits vorhandenen Wissens eine Zeitverbesserung bei Entwicklung und Training von Modellen.

In Wong (2017) wird ein Verfahren zur Suche nach ähnlichen Bildern mithilfe von Transfer Learning beschrieben. Es wird keinerlei Training benötigt, da ein bereits bestehendes und trainiertes Netzwerk für die Objektklassifizierung in Bildern zum Einsatz kommt. Die bereits trainierten Gewichte werden für das neue Modell verwendet. Somit werden anschließend Features aus den Bildern generiert mit deren Hilfe man wieder feststellen kann, welche Bilder sich ähnlich sind und welche nicht. Das verwendete Netzwerk trägt den Namen VGG.



VGG wurde von der Oxford Visual Geometry Group (VGG) entwickelt. Das Netzwerk selbst hat einen einfachen Aufbau und besteht aus mehreren Layern. Weitere Ausführungen sind VGG-16 und VGG-19. Die Zahlen 16 und 19 stehen jeweils für die Anzahl der verwendeten Layer. Im Bereich von Bildrepräsentationen ist VGG besonders gut geeignet. (Sarkar et al., 2018)

Das bestehende VGG Netzwerk kann allerdings nicht ohne Modifizierung übernommen werden. Ein Beispiel hierfür ist ein Modell, welches auf die Klassifizierung von Hunden und Katzen ausgelegt und dafür trainiert wurde. Eine Unterscheidung zwischen unterschiedlichen Marken von Schuhen wäre somit nur bedingt möglich und würde möglicherweise keine richtigen Ergebnisse liefern. Eine Lösung hierfür ist es, nur die Low-Level Features wie zum Beispiel Kanten, Texturen und Farben mithilfe von Transfer Learning aus dem jeweiligen Layer zu übernehmen. Die Erkennung von fremden High-Level Features wie Gesichtern, Körpern oder Wörtern ist erschwert, weshalb kein problemloser Transfer erfolgen kann, um diese anderen Aufgaben zu lösen. (Wong, 2017)

Das Konzept für die Modifizierung des Netzes nennt sich Fine-Tuning.

Mithilfe von Fine-Tuning kann das vortrainierte Modell auf das gegebene Datenset optimiert werden. Es wird dabei das bestehende Wissen aus dem vortrainierten Modell auf den konkreten Bereich, also dem Datenset, angewandt. Dabei werden nur einzelne Layer des bestehenden Modells anhand des Datensets trainiert. (Di, Bhardwaj & Wei, 2018)

Mithilfe von Fine-Tuning werden im Rahmen dieser Arbeit bessere Ergebnisse bei der Generierung der Features am gegebenen Datenset erwartet.

Anhand eines in einzelne Layer aufgeteilte VGG Netzes wird der schematische Ablauf von Transfer Learning näher beschrieben:

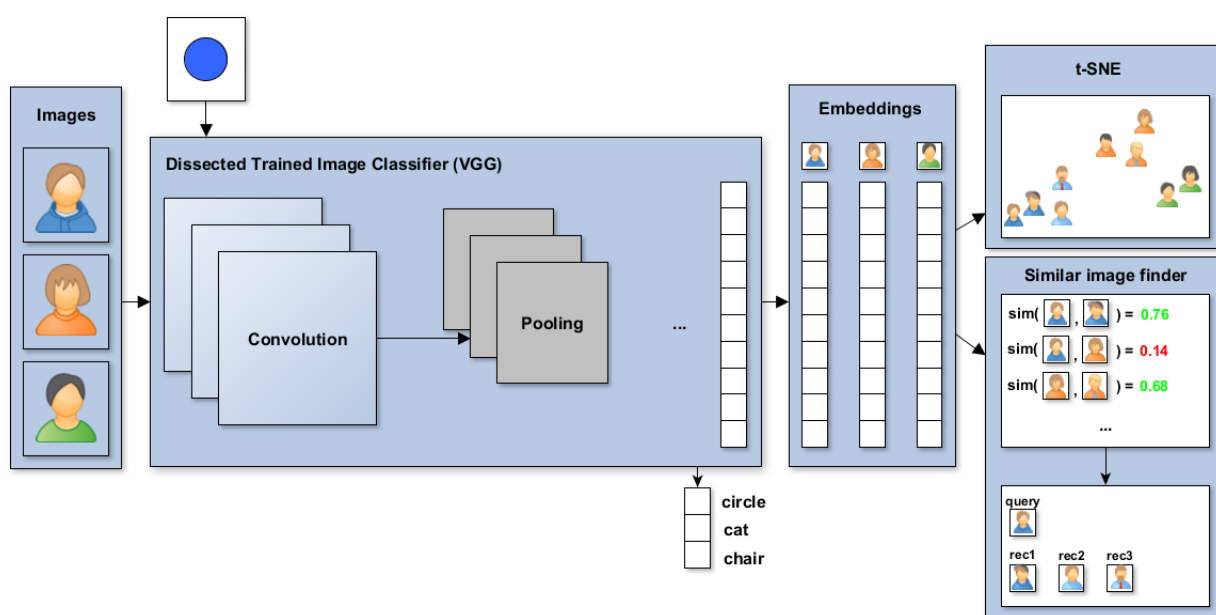


Abbildung 6-5: Schematischer Ablauf von Transfer Learning anhand Dissected VGG (in Anlehnung an Wong, 2017)

Abbildung 6-5 zeigt dabei die Komponenten und den Ablauf. Die Bilder werden innerhalb des VGG Netzes prozessiert und Embeddings daraus generiert. Diese können dann für die Suche

nach ähnlichen Bildern verwendet werden. Für die Extraktion der Low-Level Features wird der letzte High-Level Layer aus dem VGG Netzwerk entfernt. Dieses abgespeckte Modell kann dann verwendet werden, um Bilder in Feature Vektoren zu überführen, welche wiederum die Low-Level Features aus dem VGG Netzwerk darstellen. Mit diesen Feature Vektoren kann mithilfe einer Distanzmetrik die Ähnlichkeit der Bilder berechnet werden. Eine Visualisierung kann mithilfe von t-SNE ebenfalls vorgenommen werden. (Wong, 2017)

In diesem beschriebenen Verfahren wird eine eigene Bilddatenbank verwendet. Im Laufe der Umsetzung des Verfahrens muss die Bilddatenbank mit den in dieser Arbeit verwendeten Bildern ersetzt werden.

Zusammengefasst sind laut Wong (2017) folgende Schritte für die Umsetzung notwendig:

- Bilddatenbank vorbereiten
- VGG Modell downloaden und letzte Layer entfernen, damit aus dem VGG Objektklassifizierungsnetzwerk nun Bilder zu Features konvertiert werden können
- Feature Vektoren über alle Bilder aus der Datenbank mithilfe des abgespeckten VGG Modells berechnen
- Ähnlichkeiten mithilfe der Feature Vektoren und mithilfe einer Distanzmetrik berechnen
- Für jedes Bild sollten die Top-N ähnlichsten Bilder für die Empfehlung herangezogen werden
- Als optionaler Schritt kann noch zusätzlich anhand der Feature Vektoren eine Visualisierung mithilfe von t-SNE vorgenommen werden

Zusätzlich muss wie bereits beschrieben eine Modifikation des bestehenden Netzes auf das eigene Datenset erfolgen.

## 6.5 Evaluierung

Für die Evaluierung von Content-based Image Retrieval Verfahren müssen geeignete Metriken gefunden werden. Mithilfe derer können die Suchergebnisse der Verfahren verglichen werden und anschließend eine Aussage getroffen werden, welches der Verfahren anhand dieser Messkriterien die besseren Suchergebnisse zurückliefert. In Kapitel 2.7 wurden bereits Precision und Recall näher erläutert.

Im Bereich der Suchalgorithmen werden in Rosenfeld, Morville & Arango (2015) ebenfalls Precision und Recall vorgestellt, allerdings ist die Definition dieser abgeändert. Die Definition von Precision und Recall lautet dabei wie folgt:

$$\text{Precision Ratio} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$
$$\text{Recall Ratio} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents in system}}$$

Die Interpretation dieser Metriken hängt vom jeweiligen Anwendungsfall ab. Werden nur wenige aber dafür hoch qualitative Suchergebnisse benötigt so soll der Precision Wert hoch sein. Es ist dabei unwichtig, wie viele relevante Dokumente zurückgeliefert werden, da bereits die ersten Suchergebnisse zur Lösung führen. Werden jedoch viele Dokumente benötigt mit unterschiedlicher Relevanz so soll der Recall Wert hoch sein. Beide Werte können nicht gleichzeitig hoch sein. Von daher muss eine Balance der Werte gefunden werden, welche zum Anwendungsfall passt. (Rosenfeld et al., 2015)

In Jing et al. (2015) wird ein Bild als relevant bezeichnet insofern das Abfragebild das gleiche Label aufweist. Für die Evaluierung der Ergebnisse wird eine Metrik namens **Precision@K (P@K)** verwendet. Diese Metrik wird für die Evaluierung von unterschiedlichen Modellen anhand von fünf und zehn Suchergebnissen verwendet.

P@K gibt dabei das Verhältnis an, wie viele empfohlene Ergebnisse im Top-Set relevant sind. Das k steht dabei für die Anzahl an zu empfehlenden Produkten in denen die Precision berechnet wird. Ein Precision@10 Wert von 80 Prozent würde bedeuten, dass 8 von 10 Ergebnisse relevant sind.

Die Formel für P@K ist laut Malaeb (2017) wie folgt definiert:

$$\text{Precision@K} = \frac{\text{Number of recommended items @k that are relevant}}{\text{Number of recommended items @k}}$$

Im Rahmen dieser Arbeit kann somit ein Dokument als Produktbild angenommen werden. Ein Produktbild wird für eine Abfrage eines Bildes als relevant bezeichnet, insofern das Produktbild dieselbe Kategorie aufweist wie das Abfragebild.

In Dua, Ghotra & Pentreath (2017) wird eine weitere Metrik zur Evaluierung vorgestellt. Diese trägt den Namen **Mean Average Precision@K (MAP@K)**. Diese bildet sich aus dem Durchschnitt aller **Average Precision@K** Werte des jeweilig zu betrachtenden Datensets. Average Precision@K (AP@K) stellt dabei den Durchschnitt aller Relevanzscores von einem Set von Top-K Ergebnissen einer Abfrage dar. Die Platzierung der relevanten Ergebnisse in den Suchergebnissen spielt innerhalb dieser Metrik eine Rolle. AP@K liefert einen höheren Wert, je weiter vorne relevante Ergebnisse gelistet sind. Diese Metrik ist daher für Empfehlungssysteme gut geeignet. In Dua et al. (2017) ist auch die Umsetzung der Berechnung für die AP@K gegeben. Diese Berechnung wird innerhalb dieser Arbeit verwendet.

In Sawtelle (2016) ist eine Formel für die Berechnung von AP@K gegeben. Diese lautet dabei wie folgt:

$$AP@K = \frac{1}{m} \sum_{k=1}^K P(k) * rel(k)$$

In der Formel stellt rel(k) dar, inwiefern das k-te Element von Relevanz ist oder nicht. K steht dabei für den Wert der zu empfehlenden Dokumente und m steht für die Anzahl der relevanten Dokumente. P(k) stellt dabei den P@K Wert an der k-ten Position dar. (Sawtelle, 2016)

Precision@K zeigt somit auf, wie viele Ergebnisse innerhalb aller Suchergebnisse relevant sind und MAP@K berücksichtigt dabei zusätzlich die Position der relevanten Ergebnisse innerhalb von K Suchergebnissen. Aufgrund des Anwendungsfalles und der Eigenschaften dieser Metriken werden diese für die Evaluierung der Verfahren herangezogen. Diese Werte können sowohl für einzelne Kategorien und auch über alle Kategorien im Trainingsdatenset gerechnet werden. Die Testdaten stellen dabei das Abfragebild mit dem jeweiligen Label (Kategorie) dar.

## 7 EXPERIMENTSETUP

Dieses Kapitel definiert wie das Experiment innerhalb dieser Arbeit vollzogen wird. Das Experiment wird anhand eines definierten Anwendungsfalles vollzogen. Die Erkenntnisse der bisherigen Ausarbeitung tragen zu einer besseren Abgrenzung des Themengebiets bei und stellen gleichzeitig die Basis für die praktischen Untersuchungen im Rahmen dieser Arbeit dar. Daran anknüpfend, werden Hypothesen aus dem konkreten Themengebiet generiert. Auf Basis der Hypothesenbildung erfolgt eine entsprechende Operationalisierung.

Unter Operationalisierung wird im Rahmen dieser Arbeit die Definition der Messwerte verstanden. Die unterschiedlichen Messwerte münden in einen Kriterienkatalog anhand dessen die Verfahren in späteren Schritten analysiert und bewertet werden.

### 7.1 Anwendungsfall

Zusätzlich zu den generellen Anforderungen und Kriterien an das Empfehlungssystem, welche in Kapitel 6 beschrieben wurden, wird ein Anwendungsfall definiert. Dieser wird anhand von Annahmen für eine mögliche Inbetriebnahme des Empfehlungssystems im Produktivsystem erstellt. Der Anwendungsfall innerhalb dieser Arbeit lautet wie folgt:

- Fünf Produkte werden initial auf der Produktdetailseite empfohlen
- Weitere 20 empfohlene Produkte sind über entsprechende Schaltflächen einsehbar
- Mithilfe einer Konfiguration kann die Anzahl der Suchergebnisse gesteuert werden
- Die Maximalanzahl liegt bei 50 Produkten
- Es ist wichtig, dass zumindest die ersten 25 Produkte eine hohe Relevanz aufweisen
- Die höchste Priorität liegt in den ersten fünf Produkten. Diese sollen die höchstmögliche Relevanz aufweisen. Es sollen dabei im Schnitt mindestens 50 Prozent der Produkte relevant sein, damit die Empfehlung als ausreichend angesehen werden kann.

Aufgrund des Anwendungsfalles ist es daher wichtig die Relevanz innerhalb von fünf, 25, und 50 Suchergebnissen zu messen. Die Evaluierung der Verfahren selbst erfolgt ebenso auf Basis des Anwendungsfalles.

### 7.2 Eingrenzung

Aufgrund der Vielzahl an Möglichkeiten muss eine genaue Eingrenzung erfolgen, was innerhalb dieser Arbeit behandelt wird.

Das Datenset selbst ist vorgegeben und besteht aus 43759 Produkten aufgeteilt in 714 Kategorien. Dieses Datenset wird nach vorheriger Prozessierung in Trainingsdaten, Validierungsdaten und Testdaten aufgeteilt. In jedem der Verfahren werden dieselben Produkte verwendet, damit es möglich ist, die Ergebnisse aus der Berechnung der Verfahren untereinander zu vergleichen.

Eine Anforderung für die verwendeten Produkte innerhalb dieser Arbeit ist es, dass sie jeweils nur eine Kategorie aufweisen. Im vorgegebenen Datenset ist dies bereits der Fall, sodass keine zusätzliche Bearbeitung notwendig ist. Mithilfe dieser Kategorisierung ist es später möglich die Genauigkeit des jeweiligen Verfahrens bei der Berechnung der Empfehlungen festzustellen.

Ein Produkt kann anhand weiterer Merkmale, wie zum Beispiel Farbe oder Größe klassifiziert werden. Im Rahmen dieser Arbeit werden diese Möglichkeiten nicht berücksichtigt und das Verfahren selbst entscheidet ohne Einbeziehung weiterer Merkmale über die Berechnung der Empfehlung. Es erfolgt auch kein sogenanntes Re-ranking, wodurch die Suchergebnisse im Nachhinein aufgrund von Zusatzinformationen beeinflusst würden. Diese Einschränkung wird aufgrund der Tatsache getroffen, dass ausschließlich Content-based Image Retrieval zum Einsatz kommen soll. Außerdem wäre aufgrund der hohen Produktanzahl die Datenbeschaffung, die Datenkontrolle und Aufbereitung dieser Daten mit hohem Aufwand verbunden.

Die Umsetzung der Verfahren erfolgt anhand bereits bestehender Beispielimplementierungen, welche die zu Grunde liegenden Verfahren lediglich in ihrer Basisvariante abbilden. Das heißt konkret, dass der Fokus nicht darauf liegt, die einzelnen Verfahren soweit zu optimieren, dass diese die bestmöglichen Ergebnisse zurückliefern können. Geringfügige Optimierungen wie die Umsetzung von Fine-Tuning werden jedoch durchgeführt. Überdies wird aufgrund der zum Teil sehr hohen Anforderungen an benötigter Rechenleistung nur manuelles Hyperparameter tuning vorgenommen. Diese Einschränkungen führen dazu, dass die Ergebnisse der Experimente sowie davon abgeleitete Aussagen klarerweise nicht generalisierbar sind, sondern lediglich im Kontext dieser Arbeit valide sind.

### 7.3 Hypothesenbildung

Im Rahmen dieser Arbeit gilt es herauszufinden, welche Verfahren sich besonders gut für den Bildvergleich im konkreten Einsatzgebiet der Suche nach ähnlichen Produkten für eine E-Commerce Lösung eignen.

Aufgrund der bereits erläuterten Messkriterien im Kapitel 6.5 werden die Evaluierungsmetriken Precision@K und Mean Average Precision@K verwendet, um die Ergebnisse der Verfahren vergleichbar zu machen.

Transfer Learning nimmt ein bereits bestehendes und trainiertes Modell, passt dieses mithilfe von Fine-Tuning auf den eigenen Anwendungsfall an und erzeugt aus denen Features. Das Modell wurde dabei mit vielen Daten vortrainiert und wurde daher mit mehr Daten trainiert als mit Autoencoder und Siamese Network Verfahren.

In Chollet (2018) wird darauf hingewiesen, dass ein Modell, welches anhand mehr Daten trainiert wird, eine bessere Generalisierung aufweist und somit besser mit neu hinzugefügten Daten im Modell umgehen kann. Mehr vorhandene Daten verhindern, dass irrelevante oder falsche Muster gelernt werden und tragen auch dazu bei Overfitting zu vermeiden.

Auch die Komplexität des VGG16 Modells könnte dabei eine Rolle spielen. Die Anzahl der Layer ist dabei höher als in den anderen Verfahren. Es kann somit die Annahme getroffen werden, dass Transfer Learning mit VGG16 besser auf die Komplexität des vorhandenen Datensets passt und bessere Ergebnisse zurückliefern kann.

Aufgrund dessen wird die Annahme getroffen, dass diese Fakten Einfluss auf die Suchergebnisse haben. Es wird daher folgende gerichtete Hypothese generiert, welche es im Rahmen der weiteren Arbeit zu prüfen gilt:

Transfer Learning mit VGG16 liefert auf Basis des vorhandenen Datensets bessere Ergebnisse als die anderen Verfahren mit deren Umsetzung und Setup in dieser Arbeit.

Das Testen der Hypothese erfolgt im Laufe der weiteren Arbeit mithilfe des gegebenen Datensets, welches für alle Verfahren gleich ist. Was als „besser“ definiert ist wird im nachfolgenden Bereich näher erläutert.

## 7.4 Operationalisierung

Um die Hypothese zu prüfen muss diese messbar gemacht werden. Dies erfolgt im Rahmen einer Operationalisierung. Zuerst müssen die Konstrukte aus der Hypothese abgeleitet werden. Anschließend können diese Konstrukte messbar gemacht werden.

Das erste Konstrukt sind die verschiedenen untersuchten Verfahren selbst, was gleichbedeutend mit der unabhängigen Variable ist. Es wird Transfer Learning mit VGG16 verwendet. Fine-Tuning wird ebenso angewandt und das vortrainierte Modell explizit an das Datenset angepasst. Autoencoder und Siamese Network sind die beiden anderen Verfahren. Gemessen wird dieses Konstrukt jeweils anhand des zum Einsatz kommenden Verfahren.

Das zweite Konstrukt stellen die Suchergebnisse dar, welche wiederum die abhängige Variable darstellen. Die Suche nach ähnlichen Produkten erfolgt anhand von Produktbildern. Anhand von vordefinierten Testbildern aus dem eigenen Datenset werden die Ergebnisse der jeweiligen Verfahren verglichen.

Nachdem die Konstrukte definiert wurden, müssen diese messbar gemacht werden.

Das erste Konstrukt ist die Durchführung des Trainings des Modells mit dem jeweiligen Verfahren. Gemessen wird dieses Konstrukt anhand dessen, mit welchem Verfahren das Training durchgeführt wurde.

Das zweite Konstrukt sind die Suchergebnisse. Mithilfe der Metriken P@K und MAP@K können die Suchergebnisse der Verfahren gemessen und evaluiert werden. Aufgrund der Berücksichtigung der Reihenfolge der relevanten Suchergebnisse wird MAP@K als

Evalierungsmetrik herangezogen. Aufgrund der Wichtigkeit für den Anwendungsfall wird dabei die MAP@K für die ersten fünf Produkte herangezogen.

Nachdem die Ergebnisse vorliegen, kann somit versucht werden die Hypothese und in weiterer Folge die Forschungsfrage zu beantworten.

## 7.5 Kriterienkatalog

Nachdem mit der P@K und der MAP@K bereits zwei Messwerte für den Vergleich der Verfahren definiert wurden, werden weitere zum Teil weiche und auch mit Subjektivität behaftete Kriterien in diesem Kapitel vorgestellt. Diese Messwerte werden innerhalb der Umsetzung der Verfahren erhoben und auch anschließend diskutiert. Aufgrund der hohen Subjektivität, welche diese Messwerte aufweisen, werden diese jedoch nicht in die endgültige Bewertung einfließen, sondern ausschließlich erhoben und separat diskutiert.

Ein Beispiel für Messwerte dieser Art wäre die Anzahl benötigter Codezeilen zur Umsetzung des Verfahrens. Codezeilen, welche für alle Verfahren gleich sind, werden dabei nicht mitberücksichtigt. Bereiche, welche innerhalb unterschiedlicher Verfahren variieren können, sind folgende:

- Datenvorbereitung
- Modellumsetzung

Die Summe der benötigten Codezeilen ergibt somit den Messwert. Nach der Umsetzung der Verfahren kann die Anzahl der Zeilen aus den jeweiligen Quellcode-Dateien abgelesen werden.

Der Aufwand an Arbeitsstunden stellt ebenso einen Messwert dar. Dieser Aufwand wird für jedes Verfahren mithilfe einer Zeitmessung bei der Umsetzung erhoben. Auch hier ist eine gewisse Subjektivität gegeben. Dennoch ist es anhand dieser Messung möglich eine Diskussion über Aufwand und Komplexität zu führen.

Die Trainingsdauer des Modells wird ebenso gemessen und einem Vergleich unterzogen. Die Dauer wird in Stunden gemessen. Ein Durchlauf ist abgeschlossen, wenn alle Trainingsdaten verarbeitet wurden. Die Trainingsdauer wird dabei ebenso den Ergebnissen gegenübergestellt. Es könnte somit sein, dass das Training nur sehr kurzweilig war, dafür die Ergebnisse auch dementsprechend schlecht abschneiden. Dieser Umstand wird mitberücksichtigt und diskutiert.

Auch interessant ist der Vorbereitungsaufwand der Trainingsdaten in den unterschiedlichen Verfahren. Dieser Aufwand wird wiederum einem Vergleich unterzogen. Es wird gemessen, ob überhaupt eine Vorbereitung für die Daten für das Verfahren notwendig ist oder nicht. Ein konkretes Beispiel stellt dabei die Generierung der Bildpaare für das Training des Siamese Network dar. Allgemeine Vorbereitungsarbeiten wie das Image Pre-Processing fließen hier nicht mit ein.

Die Abfragedauer für ein Bild zur Suche ähnlicher Produkte ist auch von Relevanz. Die Verfahren weisen zur besseren Vergleichbarkeit jedoch alle dieselbe Suchstrategie auf. Es wird daher direkt die Abfragedauer eines Verfahrens anhand des trainierten Modells berechnet und



verglichen. Die Abfragedauer wird in Millisekunden gemessen. Es wird der Durchschnitt der Abfragedauern der Testdaten gebildet. Weiters werden mehrere Messreihen gebildet und aus diesen wiederum der Durchschnitt gebildet.

Die visuellen Ergebnisse der Verfahren werden ebenso dargestellt. Es wird versucht aus den Ergebnissen der besten und schlechtesten Kategorien gewisse Muster zu finden und diese zu diskutieren.

Zusammenfassend werden folglich nur zwei Werte gemessen, welche direkt in die Bewertung einfließen:

- Mean Average Precision@K
- Precision@K

Folgende Werte werden gemessen und separat verglichen und diskutiert:

- Anzahl an Codezeilen
- Aufwand in Arbeitsstunden für Umsetzung
- Abfragezeit für Suchergebnisse in Millisekunden
- Vorbereitungsaufwand der Daten für das Verfahren notwendig
- Trainingsdauer des Modells in Sekunden
- Visuelle Ergebnisse anhand festdefinierter Testbilder

## 8 ARTEFAKTERSTELLUNG

Die Artefakte im Rahmen dieser Arbeit stellen die jeweilige Umsetzung der bereits genannten Verfahren in konkrete und für die Praxis verwendbare Programme dar. Es wird zuerst die Umsetzung der bereits genannten Verfahren in Anlehnung an den im Kapitel 2.6 näher ausgeführten Machine Learning Prozess beschrieben. Ebenso wird auf die Herkunft der Daten, welche für die weitere Verarbeitung und zum Vergleich der Verfahren verwendet werden, eingegangen. Diese Daten und Artefakte werden mit entsprechenden Tools verwaltet. Die Verfahren zum Bildvergleich werden anhand entsprechender Bibliotheken implementiert. Bevor dies umgesetzt wird, erfolgt eine Auswahl aus dafür geeigneten Möglichkeiten. Die Artefakte selbst sind der selbst geschriebene beziehungsweise teilweise übernommene sowie adaptierte Programmcode. Dieser variiert innerhalb der unterschiedlichen Verfahren. Außerdem muss sichergestellt sein, dass alle im Kriterienkatalog definierten Kriterien aus dem jeweiligen Verfahren abgeleitet werden können und somit deren Messbarkeit gewährleistet ist.

### 8.1 Prozess

Anhand des Machine Learning Prozesses werden die Verfahren umgesetzt. Dies ermöglicht eine strukturierte Vorgehensweise bei der Erstellung der Artefakte und stellt sicher, dass keine wichtigen Schritte ausgelassen werden.

#### 8.1.1 Geschäftsproblem identifizieren

Die geschäftlichen Ziele, welche Mithilfe von qualitativ hochwertigen Produktempfehlungen erreicht werden sollen sind laut Meier & Stormer (2012) wie folgt definiert:

- Zusätzliche Verkäufe
- Steigerung der Kundenzufriedenheit
- Produkte schneller finden
- Verweildauer im Shop erhöhen

Diese Produktempfehlungen sollen mithilfe von ML für ein spezifisches Produkt errechnet und auf der Produktdetailseite platziert werden. Der Fokus liegt auf der Ähnlichkeit der Empfehlungen zum spezifischen Produkt basierend auf dem Produktbild. Eine Personalisierung soll im Rahmen dieser Arbeit bewusst nicht vorgenommen werden.

### **8.1.2 Machine Learning Zuordnung**

Innerhalb dieser Phase wird versucht das Problem den ML Typen zuzuordnen. Um das Problem zu lösen, werden unterschiedliche Verfahren zu Ermittlung von ähnlichen Bildern herangezogen. Es werden drei Verfahren einem Vergleich unterzogen. Grundsätzlich lassen sich die jeweiligen Verfahren alle unter dem Teilbereich des Deep Learning innerhalb von Machine Learning einordnen.

### **8.1.3 Datensammlung**

In dieser Phase werden notwendigen Daten für die Verfahren beschaffen. Diese Daten umfassen das Produktbild selbst, als auch die entsprechende Artikelnummer und die Kategoriebezeichnung des Produktes. Die genaue Vorgehensweise wird in Kapitel 8.3 erläutert.

### **8.1.4 Qualitätsüberprüfung der Daten**

Nachdem das Datenset vorliegt, kann die Qualität der Daten überprüft werden. Es wird das Augenmerk auf die erstellte Ordnerstruktur gelegt. Es sollte für jede Kategorie eine entsprechende Bezeichnung gegeben sein und für jedes Bild die entsprechende Artikelnummer als Namen der Datei verwendet werden. Weiters wird darauf geachtet, dass die Bilder alle dieselbe Größe aufweisen und in Farbe vorhanden sind. Mithilfe eines Skripts zur automatisierten Berechnung der Bilddimensionen wird dies sichergestellt.

### **8.1.5 Datenauswahl und Transformierung**

In dieser Phase werden mögliche Abweichungen der Bilddimensionen korrigiert. Die Bilder werden auf dieselbe Dimension transformiert. Weiters erfolgt die Aufteilung der Daten in ein Trainings- und Validierungsdatenset und einem Testdatenset.

### **8.1.6 Feature Analyse und Auswahl & Machine Learning Modellierung**

In diesen Phasen kommen die jeweiligen ML Verfahren zum Einsatz, wodurch die entsprechenden Features aus den Bilddaten extrahiert werden können. Anschließend können mit einem ausgewählten Suchverfahren die Ähnlichkeiten zwischen einzelnen Produkten ermittelt werden.

### **8.1.7 Modellevaluierung**

In der Phase Modellevaluierung wird das im jeweiligen Verfahren verwendete Modell überprüft und möglicherweise eine Verbesserung mithilfe anderer Parameter für das jeweilige Verfahren erzielt.

### **8.1.8 Modellauswahl und Deployment**

Innerhalb der Phase Modellauswahl und Deployment wird zuerst das Modell mithilfe von optimierten Parametern ausgewählt. Die Umsetzung des jeweiligen Verfahrens ist damit abgeschlossen, weshalb im nächsten Schritt die Bereitstellung der Funktionalität zur Suche nach ähnlichen Produkten stattfinden kann. Hierfür wird eine Schnittstelle benötigt mit deren Hilfe es möglich ist, anhand einer Artikelnummer jeweilige ähnliche Produkte zu finden. Diese Umsetzung kann mithilfe einer bereitgestellten Representational State Transfer (REST) Schnittstelle erfolgen.

Die Umsetzung dieser Schnittstelle wird im Rahmen dieser Arbeit aufgrund des Aufwandes jedoch nicht weiter behandelt.

### **8.1.9 Performanceüberwachung des Modells**

Die letzte Phase stellt die Performanceüberwachung des Modells dar. In dieser werden Daten extrahiert, um zu erkennen, wie gut das im jeweiligen Verfahren verwendete Modell funktioniert. Diese Daten können dann anschließend in die Beurteilung der jeweiligen Verfahren miteinfließen.

## **8.2 Datenherkunft**

Die für den untersuchten Anwendungsfall zu Grunde liegenden Daten kommen von einem namhaften Möbelverkäufer in Österreich. Dieser bietet unterschiedliche Produkte aus mehreren Kategorien an. Im Datenset ist jedes Produkt mit nur einem Produktbild vertreten. Das entsprechende Produktbild ist jenes, welches in der E-Commerce Lösung als Hauptbild auf der Produktdetailseite aufscheint. Ebenso ist sichergestellt, dass jedes Produkt nur eine Kategorie aufweist, wobei der Kategorienname als Label dient.

Die Datenqualität der verschiedenen Produktbilder ist aufgrund der Tatsache, dass diese dem Kunden direkt auf der Website präsentiert werden und diese nicht von BenutzerInnen selbst hochgeladen werden müssen, entsprechend hoch. Dies betrifft sowohl die Professionalität der Bilder als auch deren Auflösung.

Folgende Abbildung zeigt ein Beispiel eines Produktbildes innerhalb des verwendeten Shops:



*Abbildung 8-1: Beispiel eines Produktbildes*

Abbildung 8-1 zeigt ein Beispiel eines Produkts. Zu beachten gilt es, dass das Objekt selbst in den Vordergrund gestellt wird, somit immer das Hauptmotiv darstellt. Der Hintergrund des Bildes selbst ist homogen in weiß gehalten. Weiters ist visuell auch zu erkennen, dass die Produktrepräsentation eine hohe Qualität aufweist.

### **8.3 Datensammlung**

Eine Sammlung der Daten ist nicht notwendig, da das Datenset bereits in benötigter Form gegeben ist. Folgende Informationen zu einem Produkt werden berücksichtigt:

- Artikelnummer
- Produktbild
- Kategoriebezeichnung

Das Datenset selbst wird in einer geeigneten Ordnerstruktur am lokalen Rechner abgelegt. Ein beispielhafter Auszug dieser Ordnerstruktur sieht wie folgt aus:

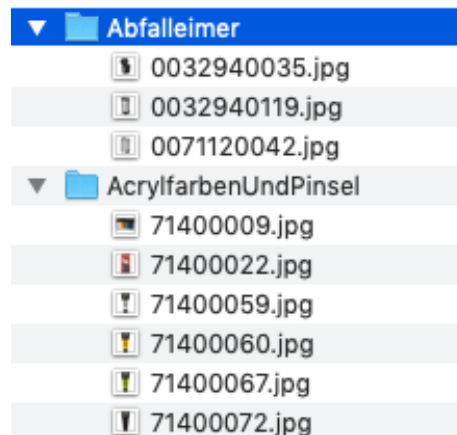


Abbildung 8-2: Kategoriestructur

Abbildung 8-2 zeigt anhand eines Auszugs die Ordnerstruktur auf. Der Ordernamen stellt die Kategoriebezeichnung dar. Die darunterliegenden Dateien stellen die Produktbilder dar, wobei jedes Produkt genau mit nur einem Bild in der Ordnerstruktur vorkommt.

Die Beschaffung der Daten erfolgt vorab einmalig zu einem bestimmten Zeitpunkt. Das Datenset ist dabei fix vorgegeben. Im Rahmen dieser Arbeit ist es nicht Ziel, dass immer aktuelle Daten verfügbar sein müssen.

## 8.4 Vorverarbeitung der Daten

Nachdem alle benötigten Daten beschaffen wurden, müssen diese für die Verwendung innerhalb der Verfahren vorverarbeitet werden. Das Ziel ist es für jedes Verfahren dieselben Daten zu verwenden, um somit eine Vergleichbarkeit dieser zu ermöglichen.

Die Vorverarbeitung der Daten umfasst laut Grigorev, Shanmugamani, Boschetti, Massaron & Thakur (2018) folgende Bereiche:

- Image Pre-Processing
- Aufteilung in Datensets

Die oben genannten Schritte sind allgemein gehalten. Je nach Verfahren werden weitere Vorverarbeitungsschritte benötigt. Ein Beispiel hierfür ist die Ermittlung der Bildpaare innerhalb des Verfahrens Siamese Network. Diese Schritte werden innerhalb der betroffenen Verfahren selbst genauer erläutert.

### 8.4.1 Image Pre-Processing

Image Processing stellt dabei die automatisierte Prozessierung, Manipulation, Analyse und Interpretation von Bildern mithilfe von Algorithmen dar. Ein Bild kann dabei unterschiedlich aufgebaut sein. Ein Single-Channel Bild ist ein Graustufenbild oder Schwarz-Weiß Bild. Die

Koordinaten eines Bilds werden auf einen Wert abgebildet, welcher den Intensitäts- oder Farbwert darstellt. Dieser Punkt wird auch Pixel genannt. Ein Bild kann aber auch mehrere Channels haben. Diese werden auch Multi-Channel Bilder genannt. Ein Beispiel für Bilder mit drei Channels wären sogenannte RGB Bilder, wobei R für rot, G für grün und B für blau steht. Ein Pixel besteht hiermit aus einem Dreier-Tupel  $(r_{xy}, g_{xy}, b_{xy})$ . Diese Bilder werden mithilfe multidimensionaler Arrays abgespeichert. Single-Channel Bilder benötigen zwei Dimensionen wobei ein RGB-Bild ein dreidimensionales Array benötigt. (Dey, 2018)

Im Rahmen dieser Arbeit werden Multi-Channel Bilder mit drei Channels (RGB) verwendet.

In Dey (2018) wird Image Processing vorgestellt und besteht aus den folgenden Schritten:

- Holen und Speicherung der Bilder
- Bild in den Speicher laden und zurückspeichern, nachdem Algorithmen auf das Bild angewandt wurden (Pre-Processing)
- Manipulation, Erweiterung und Restaurierung von Bildern
- Segmentation, um das Objekt, welches von Interesse ist zu ermitteln
- Informationen extrahieren (Features)
- Bild interpretieren und verstehen

Das Holen und Speichern der Bilder wurde im vorherigen Abschnitt bereits behandelt. Das Pre-Processing selbst wird für die Aufbereitung der Bilder benötigt. Die Schritte nach dem Pre-Processing erfolgen in den jeweiligen Verfahren und werden dort genauer behandelt.

Pre-Processing wird in Grigorev et al. (2018) näher erläutert. Der erste Schritt beschreibt das Einlesen der Bilder und diese in ein einheitliches Format zu bringen. Es werden daher die Bilder mithilfe von reshaping in eine einheitliche Größe transformiert. Weitere Transformationen, wie eine Umwandlung des Bildes in ein Graustufenbild können ebenfalls angewandt werden. Der letzte Schritt nach dem Pre-Processing ist das Aufteilen der Daten.

#### **8.4.2 Aufteilung der Daten**

Nachdem alle Bilder dieselben Eigenschaften aufweisen, kann damit begonnen werden, die Auswahl der Trainings- und Testbilder für die Verfahren zu treffen.

In Coelho, Richert & Brucher (2018) werden einfach zu vermeidende Fehler bei der Auswahl dieser vorgestellt:

- dieselben Daten sowohl für das Training als auch zum Testen verwenden
- unbalancierte Trainingsdaten

Im Rahmen dieser Arbeit wird daher darauf Wertgelegt, dass unterschiedliche Daten für das Testen verwendet werden. Weiters werden Produkte aus unterschiedlichen Kategorien für das Training herangezogen.

In Moolayil (2019) wird ein weitverbreiteter Weg vorgeschlagen, um eine Aufteilung der Trainingsdaten vorzunehmen. So wird vorgeschlagen 60 Prozent der vorhandenen Daten für das Training zu verwenden, 20 Prozent für die Validierung und wiederum 20 Prozent für das Testen. Es wird auch erwähnt, dass diese Aufteilung keinesfalls zwanghaft ist und das bei größeren Datenmengen wie zum Beispiel mehr als einer Million Datensätze durchaus andere Aufteilungen wie 95 Prozent für Training, zwei Prozent für Validierung und drei Prozent für das Testen verwendet werden können.

Das Validierungsdatenset wird laut Halder & Ozdemir (2018) für zwei Zwecke verwendet:

- Hyperparameter tuning
- Modelleffizienz abschätzen

Es enthält andere Daten als das Trainingsset und das Testdatenset. Das Validierungsset wird auch nicht dafür verwendet, um das Modell zu trainieren. Das Testdatenset wird erst für die Modellauswahl verwendet, nachdem das Modell mithilfe der Validierungsdaten optimiert wurde. (Halder & Ozdemir, 2018)

Im Rahmen der Arbeit kommt K-Fold Cross Validation zum Einsatz. Es wird dazu eine Aufteilung des Datenset im Verhältnis 80 zu 20 gewählt. Das bedeutet, es werden 80 Prozent der Bilder für das Training und Validierung und 20 Prozent der Bilder für das Testen verwendet. Innerhalb der 80 Prozent werden jeweils abwechselnd 20 Prozent der Bilder für die Validierung verwendet und 80 Prozent für das Training. Dieser Vorgang erfolgt abwechselnd fünf Mal. Es werden somit die 80 Prozent der Bilder in fünf gleiche große Bereiche aufgeteilt. Diese Bereiche werden auch Splits genannt. In jedem Vorgang werden vier Splits für das Training und jeweils ein Split für die Validierung verwendet. Nach einem Vorgang wird ein anderer Split für die Validierung ausgewählt und die restlichen vier werden für das Training verwendet. Nach fünf Vorgängen ist somit jeder Split für die Validierung verwendet worden.

Die Ermittlung dieser Produkte erfolgt durch eine einmalige zufällige Ziehung mithilfe eines Skripts. Nachdem die Produkte für die Ziehung ausgewählt wurden, werden diese fixiert, sodass jedes Verfahren mit demselben Set an Daten arbeitet. Auch die Produkte innerhalb eines Splits bei K-Fold-Cross Validation werden fixiert. Bei der Zufallsziehung wird darauf Wert gelegt, dass die Verteilung der jeweiligen Produkte innerhalb einer Kategorie im jeweiligen Trainings-, Validierungs- und Testdatenset ausgeglichen ist. Als Beispiel hierfür sollen im Trainings- und Validierungsdatenset aus der Kategorie Tische 400 von 500 Produkte und aus der Kategorie Sofa je 800 von 1000 Produkte stammen und so weiter. Beim Testdatenset würden wiederum je 100 von 500 Produkte aus der Kategorie Tische und 200 von 1000 Produkte der Kategorie Sofa zum Einsatz kommen. Weiters wird bei der Ziehung berücksichtigt, dass die Testdaten nicht für das Training verwendet werden.



### 8.4.3 Datenset

Die Vorverarbeitung des Datensets ist notwendig, um eine Evaluierung mithilfe der Metriken vornehmen zu können und das Datenset auch auf den jeweiligen Anwendungsfall innerhalb der Arbeit anwenden zu können.

Das originale und noch nicht modifizierte Datenset dieser Arbeit besteht aus insgesamt 43759 Produkten aufgeteilt in 714 Kategorien. Die Verteilung des originalen Datensets ist in folgender Abbildung ersichtlich:

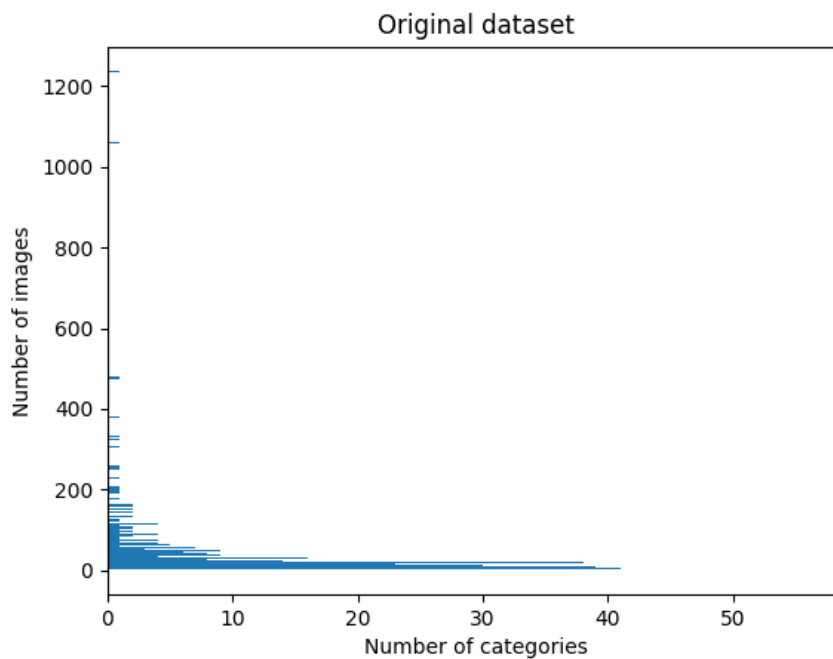


Abbildung 8-3: Histogramm originales und unmodifiziertes Datenset

Abbildung 8-3 zeigt dabei auf, dass sehr viele Kategorien jeweils nur sehr wenige Produkte beinhalten. Das Minimum an Produkten innerhalb einer Kategorie liegt bei eins, während das Maximum an Produkten innerhalb einer Kategorie bei 1237 liegt. Das Datenset kann aufgrund dieses Umstandes als unausgeglichenes angesehen werden.

Der ergänzende Boxplot zeigt ebenfalls die Verteilung des originalen Datensets an:

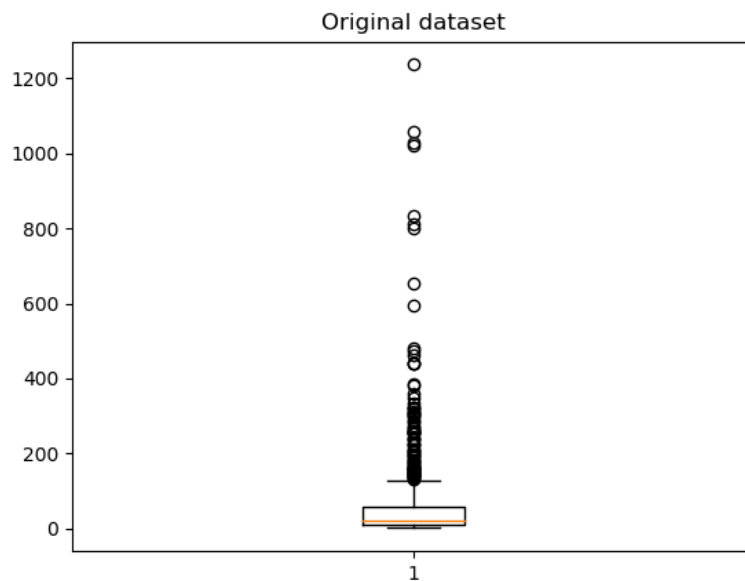


Abbildung 8-4: Boxplot originales und unmodifiziertes Datenset

Ergänzend ist in Abbildung 8-4 zu sehen, dass der Median an Kategorien unter 100 Produkten innerhalb der Kategorie liegt und im Bereich zwischen 0 und 100 mehr als 50 Prozent der Kategorien angesiedelt sind. Es gibt sehr viele Ausreißer nach oben hin.

Für den Anwendungsfall stellen sehr wenigen Produkte innerhalb ihrer jeweiligen Kategorie ein Problem dar. Es müssen also Maßnahmen definiert werden, wie damit umzugehen ist. Folgende Vorgehensweise wurde getroffen, um das Datenset auf den Anwendungsfall innerhalb dieser Arbeit vorzubereiten:

- Manuelle Durchsicht des Datensets
- Ermittlung von gleichbedeutenden und (zu) feingranularen Kategorien
- Kopie des originalen Datensets erstellen und nur an Kopie Modifikationen durchführen
- Zusammenführung von gleichen Kategorien und der enthaltenen Produkte
- Entfernung von Kategorien mit Produktanzahl unterhalb eines definierten Schwellwerts angepasst auf den Anwendungsfall
- Bildbreite und Bildhöhe verkleinern, sodass Berechnungen am lokalen System erfolgen können
- Modifiziertes Datenset in Trainings- und Validierungsdatsenset, sowie Testdatenset aufteilen, wobei die Produkte innerhalb der fünf Splits für K-Fold Cross Validation fixiert werden

Bei der manuellen Durchsicht des Datensets konnte erkannt werden, dass viele Kategorien sehr feingranular aufgeteilt wurden. Ein Beispiel hierfür ist, dass es im Datenset insgesamt 11 unterschiedliche Kategorien an Sofas gegeben hat, unter anderem Schlafsofa, Textilsofa,

Ecksofas und weitere. Diese Kategorien wurden in einer Überkategorie Sofas zusammengefasst. Insgesamt wurden 95 Kategorien innerhalb von neuen Überkategorien zugewiesen. Dies wurde daher vollzogen, um die Produktanzahl innerhalb der Kategorien zu erhöhen und eine höhere Aussagekraft der Evaluierungsmetriken zu erreichen. Im Rahmen dieser Arbeit ist keine feingranulare Unterscheidung, wie zum Beispiel zwischen Stoffen wie Leder oder Textil bei Sofas notwendig. Hierfür könnten Metadaten herangezogen werden. Dies ist aber kein Ziel dieser Arbeit.

Um die Produktanzahl in den Kategorien zu erhöhen könnten noch weitere Produkte hinzugezogen werden insofern diese vorhanden sind. Weiters könnte die sogenannte Image Augmentation Technik zum Einsatz kommen.

Das Hinzuziehen weiterer Produkte für Kategorien mit wenigen Produkten und der Ansatz des Image Augmentation werden innerhalb dieser Arbeit aufgrund der zeitintensiven Berechnungen nicht weiterverfolgt.

Die Modifikation des Datensets erfolgt nicht am originalen Datenset. Es wird eine Kopie des originalen Datensets angelegt. Diese Kopie stellt dann das Datenset dar und wird anschließend modifiziert.

Nachdem das Datenset für Modifikationen existiert, werden Kategorien zu Überkategorien zusammengeführt. Dennoch gibt es nach dieser Zusammenführung noch viele Kategorien mit wenigen Produkten. Für den Anwendungsfall muss also ein geeigneter Schwellwert gefunden werden, um Kategorien zu entfernen, welche weniger Produkte als der Schwellwert aufweist, damit relevante Ergebnisse an KundInnen zurückgeliefert werden können. Dieser Schwellwert wird innerhalb dieser Arbeit auf 100 Produkte gesetzt. Die Gründe dafür lauten wie folgt:

- Es werden maximal 50 Produkte zurückgeliefert
- Manuelle Durchsicht zeigt teilweise eine gewisse Varietät der Produkte innerhalb der Kategorie (visuelle Ergebnisse stimmen also möglicherweise nicht mit Evaluierungsmetriken überein)
- Trainings- und Validierungsdaten, sowie Testdaten können aufgrund geringer Produktanzahl nicht gebildet werden

Nachdem die Kategorien mit einer Produktanzahl unter dem festgelegten Schwellwert aus dem Datenset entfernt wurden, besteht das Datenset nun aus 25992 Produkten aufgeteilt in 48 Kategorien. Die Verteilung des modifizierten Datensets sieht nun wie folgt aus:

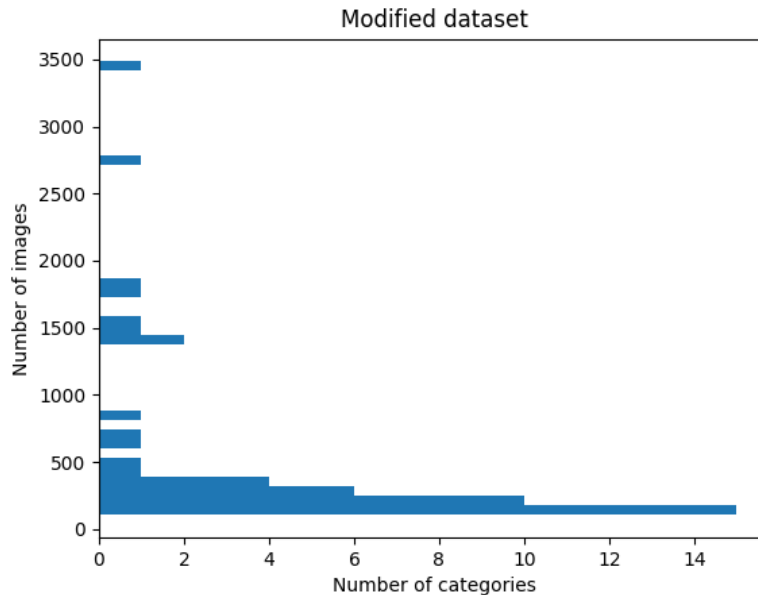


Abbildung 8-5: Histogramm modifiziertes Datenset

Abbildung 8-5 zeigt dabei auf, dass die Kategorieanzahl im Gegensatz zum originalen Datenset verringert ist, die schiefe Verteilung aber noch immer gegeben ist. Sie zeigt aber auch auf, dass alle Kategorien nun mindestens 100 Produkte aufweisen und somit die Anforderungen des Anwendungsfalles erfüllen. Das Minimum an Produkten in einer Kategorie liegt nun bei 104 und das Maximum bei 3485. Wie diese Kategorien anhand der Testdaten mithilfe der Evaluierungsmetriken abschneiden wird sich im Verlauf der weiteren Arbeit zeigen.

Ergänzend wird ebenfalls der Boxplot des modifizierten Datensets zur Schau gestellt:

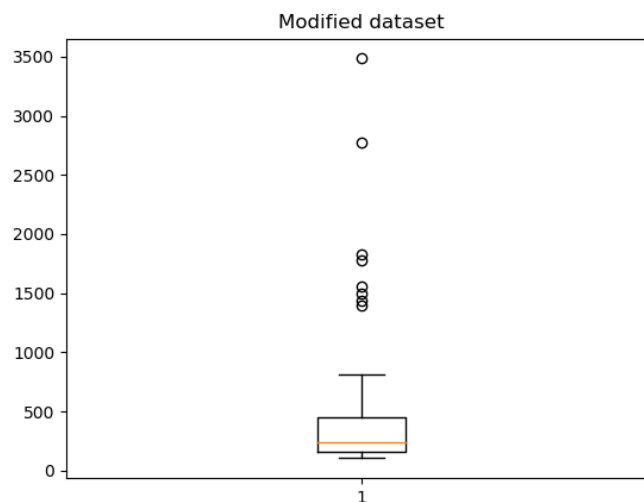


Abbildung 8-6: Boxplot modifiziertes Datenset

Wie in Abbildung 8-6 zu erkennen, sind auch wie im originalen Datenset noch immer einige Ausreißer erkennbar und mindestens 50 Prozent aller Kategorien zwischen 100 und 500 Produkte liegen.

Eine Reduzierung der Produkte innerhalb der Kategorien auf die Anzahl der Kategorie mit den wenigsten Produkten mit der Anzahl 104 erfolgt dabei nicht. Dies würde bedeuten, dass viele Produkte innerhalb des Trainings nicht berücksichtigt und somit auch die möglichen Suchergebnisse noch weiter eingeschränkt werden würden.

Für die Durchführung des Modelltrainings am lokalen Rechner (16 Gigabyte Arbeitsspeicher) werden alle Bilder im Datenset von der Dimension von 1024 x 1024 auf die Dimension 32 x 32 verkleinert. Die Durchführung des Trainings der ML Verfahren mit Bildern größerer Dimension (wie zum Beispiel 300 x 300) brachte keine besseren Testergebnisse.

Abschließend wird das modifizierte Datenset noch in Trainings- und Validierungsdatsenset, sowie in ein Testdatenset aufgeteilt. Die Aufteilung erfolgt wie zuvor beschrieben im Verhältnis 80:20. Aus jeder Kategorie werden Produkte im jeweiligen Verhältnis gezogen. Beinhaltet eine Kategorie 100 Bilder so werden 80 für das Training und die Validierung verwendet und 20 für das Testdatenset. Bei 200 Bildern innerhalb einer Kategorie wären dies 160 Bilder für Training und Validierung und 40 für das Testdatenset. Es wird darauf geachtet, dass ein Produkt in nur einem Datenset vorkommt. Das Datenset wird in absoluten Zahlen gemessen wie folgt aufgeteilt:

- Trainings- und Validierungsdatsenset: 20814 Produkte aus 48 Kategorien
- Testdatenset: 5178 Produkte aus 48 Kategorien

Die Bilder werden dabei einmal gezogen und fixiert. Diese werden nicht bei jedem Trainingsdurchlauf des Modells erneuert. Dies ermöglicht eine Vergleichbarkeit zwischen den Verfahren.

Zusammengefasst ist die Vorbereitung des Datensets ein notwendiger Bestandteil, um relevante Ergebnisse zu bekommen und diese im Anschluss mithilfe von Evaluierungsmetriken vergleichen zu können.

## 8.5 Tools

Für die Umsetzung der Verfahren werden unterschiedliche Werkzeuge benötigt. Diese umfassen die lokale Entwicklung der Verfahren, die verwendete Programmiersprache, wichtige Bibliotheken und Werkzeuge, um unterschiedliche Versionen der Verfahren zu verwalten. Dieses Kapitel stellt die wichtigsten Werkzeuge vor, welche im Rahmen dieser Arbeit zum Einsatz kommen. Aufgrund des Umfangs wird jedoch nicht im Detail auf die Installation und Nutzungsweise des jeweiligen Tools eingegangen. Es sind in diesem Kapitel auch keine Codebeispiele angeführt, da diese innerhalb der Umsetzung der Verfahren gegeben sind.

### **8.5.1 Python**

In Coelho et al. (2018) wird Python aufgrund der Fähigkeit Sachen einfach auszuprobieren und den einfach zu benutzenden Bibliotheken für Machine Learning empfohlen. Generell wird Machine Learning als Prozess beschrieben, in welchem sehr oft ausprobiert und möglicherweise Schritte zurück und wieder nach vorne gemacht werden müssen.

Aufgrund dieser Übereinstimmung wird daher der Programmcode für die Verfahren mithilfe von Python umgesetzt.

### **8.5.2 NumPy**

NumPy ist eine Erweiterung für Python, um große und multidimensionale Arrays verarbeiten zu können. Normale Python Lists sind in deren Implementierung zu rechenintensiv, um diese sinnvoll nutzen können. Weiters beinhaltet diese Bibliothek mehrere mathematische Funktionen, um Arrays modifizieren zu können. (Lee, 2019)

In Singh (2019) wird eine Verwendung von NumPy für die Bildverarbeitung vorgestellt. Ein Bild wird dabei pixelweise eingelesen und in einem Array gespeichert. Dieses Array wird intern als NumPy Array repräsentiert. Anhand definierter Funktionen ist es möglich, die Dimensionen des Bilds auszulesen und Modifikationen durchzuführen.

NumPy wird daher im konkreten Einsatzbereich der Bildverarbeitung und Bildmanipulation innerhalb dieser Arbeit verwendet.

### **8.5.3 TensorFlow**

TensorFlow wird in Ravichandiran (2019) als eine Open Source Bibliothek von Google beschrieben, welche es ermöglicht, numerische Berechnungen durchzuführen und ist eine der populärsten Bibliotheken, um Deep Learning Modelle zu erstellen.

TensorFlow kommt innerhalb der Umsetzung der Verfahren zum Einsatz.

### **8.5.4 Keras**

Keras ist eine high-level API für TensorFlow und wird für das Prototyping, Erzeugen und Trainieren von Deep Learning Modellen verwendet. TensorFlow kümmert sich um die Berechnungen wohingegen Keras die Schnittstelle ist, mit welcher definiert wird, was berechnet werden soll. Der Aufbau von Keras gleicht dabei einem Modell eines neuronalen Netzwerks. Die Schritte Build, Kompilierung, Training, Evaluierung und Vorhersagen treffen können mithilfe von Keras umgesetzt werden. (Holdroyd, 2019)

Aufgrund der Unterstützung von unterschiedlichen Arten von Modellen, der strukturierten Vorgehensweise bei der Verwendung von Modellen, der Einfachheit bei der Verwendung selbst und des generellen Funktionsumfangs wird Keras für die Umsetzung der unterschiedlichen Verfahren verwendet.

### **8.5.5 TensorBoard**

TensorBoard ist ein Visualisierungstool und ist in TensorFlow enthalten. Es hilft dabei Deep Learning Modelle zu verstehen. Mithilfe des Embedding Projector und PCA oder t-SNE wird eine Visualisierung von Embeddings ermöglicht. (Bisong, 2019)

Bei der Umsetzung der Verfahren wird mithilfe dieser Technik versucht, das Modell und die Gruppierung von ähnlichen Bildern visuell darzustellen und grob zu bewerten. Sind die Ergebnisse der Gruppierung nicht ansatzweise stimmig so muss das Modell beziehungsweise das Verfahren selbst hinterfragt und nachgebessert werden.

### **8.5.6 PyCharm**

Damit die Verfahren von der Theorie in die Praxis überführt werden können muss Programmcode geschrieben werden. Damit dieser Vorgang erleichtert wird kommt ein Tool namens PyCharm zum Einsatz.

PyCharm ist eine Entwicklungsumgebung von JetBrains. Es basiert dabei auf der ebenfalls von JetBrains entwickelten IntelliJ IDEA. Funktionen wie Debugger, Autocompletion, Syntax Highlighter und weitere werden unterstützt. (Ortega, 2018)

### **8.5.7 MLflow**

MLflow ist ein weiteres Tool, welches innerhalb dieser Arbeit Anwendung findet.

In Callaert (2018) wird MLflow als Framework beschrieben, welches den Machine Learning Prozess unterstützt. Mithilfe von MLflow ist es möglich, das Modell selbst während des Trainings oder der Laufzeit zu überwachen, ein Modell zu speichern, ein Modell in die Produktivumgebung zu laden oder eine Pipeline zu erstellen. MLflow Tracking ermöglicht es Werte rund um das Modell selbst zu loggen.

Es wird daher verwendet, um die Nachvollziehbarkeit innerhalb der Arbeit zu gewährleisten.

### **8.5.8 Git**

Im Rahmen dieser Arbeit wird Git zur Versionskontrolle verwendet, damit neue Funktionen und Änderungen innerhalb der Verfahren nachvollziehbar bleiben. Es wird ein Repository für alle Verfahren geben, damit die Übersicht nicht verloren geht und zusammenhängende und geteilte Bereiche einfach verwendet werden können. Das Datenset selbst wird dabei nicht im Repository gespeichert.

### **8.5.9 Matplotlib**

Für die Visualisierung der Suchergebnisse wird Matplotlib verwendet.

Matplotlib ist eine Python Bibliothek zur Erstellung von zweidimensionalen Graphen. Es können dabei auch Bilder innerhalb der Graphen angezeigt werden. (Poladi, 2018)

## 8.6 Implementation der Verfahren

Nachdem die Verfahren zur Suche nach ähnlichen Bildern, sowie die Evaluierung dieser näher erläutert wurden, können diese nun in die Praxis umgesetzt werden. Dies erfolgt mit Hilfe der oben genannten Tools. Rund um die Verfahren gibt es allgemeinere Aufgaben, welche unabhängig sind und damit gleich abgehandelt werden. Sie bestehen aus demselben Programmcode. Nachfolgend werden die einzelnen Schritte aufgezeigt, wie die Umsetzung der Verfahren aufgebaut ist:

- Vorbereitung der Daten
- Laden und Transformierung der Datensets
- Implementierung des Verfahrens, sowie Training oder Laden des Modells
- Ermittlung der Features anhand des trainierten Modells
- Suche nach ähnlichen Bildern anhand der Features
- Evaluierung der Suchergebnisse
- Aufbereitung der Ergebnisse zur Visualisierung mithilfe von t-SNE und visuellem Vergleich der Abfragebilder und Suchergebnisse

In den nachfolgenden Kapiteln werden diese Arbeitsschritte näher erläutert.

### 8.6.1 Allgemein

Die Verfahren wurden so umgesetzt, dass es mithilfe eines Parameters möglich ist, zwischen einem Trainingsmodus und einem Lademodus zu unterscheiden. Im Trainingsmodus wird das Modell des jeweiligen Verfahrens anhand des vorbereiteten Datensets trainiert und gespeichert. Im Lademodus wird das vorher abgespeicherte Modell geladen und verwendet. Es gibt aber noch mehr Einstellungen, welche die Ausführung des Verfahrens beeinflussen können. Die Konfigurationsoptionen lauten wie folgt:

- `dataset`: Ermöglicht die Verwendung von unterschiedlichen Datensets
- `search_type`: Suchtyp wie ANNOY oder reine Nearest Neighbor Suche
- `dataset_original`: Pfad zum originalen und unmodifizierten Datenset
- `dataset_original_size_dir`: Pfad zum modifizierten Datenset mit Bilddimensionen aus dem originalen Datenset
- `dataset_dir`: Pfad zum modifizierten Datenset
- `training_dir`: Pfad zum definierten Trainingsdatenset
- `validation_dir`: Pfad zum definierten Validierungsdatenset



- `test_dir`: Pfad zum definierten Testdatenset
- `k_fold_splits`: Anzahl der Splits für K-Fold Cross Validation
- `search_counts`: Eine Liste an Werten, welche jeweils die Anzahl der zu erwartenden Suchergebnisse darstellen

In Bezug auf spezifische Datensets können weitere Einstellungen wie folgt getroffen werden:

- `img_width`: Die Breite der Bilder
- `img_height`: Die Höhe der Bilder
- `channels`: Die Anzahl der Channels der Bilder
- `type`: Der Typ des Datensets, welcher für feine Einstellungen und Transformationen innerhalb der Implementierung notwendig ist, um diese zu unterstützen

Anhand dieser Parameter können wiederum Vergleiche zur Evaluierung der Verfahren durchgeführt werden. Sämtliche dafür benötigte Metriken werden mithilfe des Tracking Tools MLflow aufgezeichnet und zur Evaluierung herangezogen.

Das Training der Modelle erfolgt je nach Verfahren unterschiedlich. Aufgrund der K-Fold Cross Validation wird in jedem Verfahren nur eine Epoche zum Training innerhalb eines Splits herangezogen. Die Aufbereitung der Trainingsdaten wird in den einzelnen Verfahren näher erläutert, da sich diese unterscheiden. Die Trainingsdauer selbst wird auch vermerkt und gespeichert.

### 8.6.2 Vorverarbeitung

Innerhalb des ersten Schrittes wird die in Kapitel 8.4 beschriebene Modifikation des Datensets vorgenommen. Dieser Vorgang wird einmalig ausgeführt. Das daraus generierte Trainingsdatenset, Validierungsdatenset und Testdatenset wird anschließend für jedes Verfahren verwendet. Für die Zusammenführung der Kategorien wurde eine `.csv` Datei erstellt. Eine Zeile innerhalb dieser Datei besteht dabei aus dem Namen der Kategorie für die Zusammenführung und der übergeordneten Kategorie. Es wird ein neues Verzeichnis für die übergeordnete Kategorie angelegt (sofern nicht bereits vorhanden) und die Produkte aus der zu zusammenführenden Kategorie in die übergeordnete Kategorie verschoben. Anschließend wird die zusammenführende Kategorie gelöscht.

Kategorien, welche noch immer zu wenig Produkte für die Evaluierung und den Anwendungsfall aufweisen, werden anschließend ermittelt und gelöscht.

### 8.6.3 Laden und Transformieren

Innerhalb dieses Schrittes werden die Datensets geladen. Dies erfolgt anhand der getroffenen Einstellungen im Verfahren selbst. Es wird dabei der Pfad zum jeweiligen Verzeichnis des Datensets benötigt. Das Laden des Datensets erfolgt dabei mithilfe der Funktion `flow_from_directory`. Mit dieser ist es möglich, dass Labels (Kategorienamen) automatisch

ermittelt und geladen werden und eine spätere Evaluierung der Suchergebnisse möglich machen. Die Produktbilder werden anschließend in die in den Einstellungen angegebenen Dimensionen transformiert. Zusätzlich werden Statistiken und Grafiken über das Datenset generiert, ausgegeben und in mlflow für eine spätere Verwendung gespeichert.

#### **8.6.4 Modell Setup: Autoencoder**

Das Autoencoder Verfahren selbst wird anhand einer bestehenden Umsetzung<sup>1</sup> aufgebaut und entsprechend modifiziert. Der Autoencoder besteht wie bereits in Kapitel 6.1 beschrieben aus zwei Teilen. Einem Encoder und einem Decoder.

Der Encoder besteht dabei aus drei Convolutional Layern und drei Pooling Layern. Als Eingabe wird ein Bild mit den Dimensionen in den Einstellungen erwartet.

Der Decoder bekommt als Eingabe den Encoder übergeben. Er besteht aus vier Convolutional Layern und drei Upsampling Layern.

Der Autoencoder stellt das eigentliche Modell dar und bekommt den Encoder übergeben.

Das Training des Modells erfolgt anhand des Trainingsdatensets und erfordert keine zusätzliche Modifikation. Die Validierung des Modells erfolgt anhand K-Fold Cross Validation.

#### **8.6.5 Modell Setup: Siamese Network**

Für die Umsetzung dieses Verfahrens wird als Basis die beschriebene Implementierung in Mader (2017) herangezogen und modifiziert. Das Siamese Network nimmt als Eingabe die Dimensionen der Bilder. Es wird jedoch ein Bildpaar übergeben. Für das Training werden Bildpaare zufällig anhand deren Kategorie erzeugt. Es wird darauf geachtet, dass gleich viele positive Paare wie negative Bildpaare erzeugt werden. Positive Paare bilden Bilder, welche in derselben Kategorie liegen, während bei negativen Paaren die Bilder in unterschiedlichen Kategorien liegen. Dieser Vorgang geschieht ebenso für die Validierungsdaten. Die Features werden mithilfe des Feature Generation Networks erzeugt. Für jedes Eingabebild wird ein Feature Generation Network verwendet. Das Siamese Network verwendet somit zwei Feature Generation Networks. Für jeden Split der K-Fold Cross Validation werden zufällige Bildpaare gebildet, wobei die Splits jedoch fixiert sind.

#### **8.6.6 Modell Setup: Transfer Learning**

Beim Transfer Learning Verfahren wird VGG16 verwendet. VGG16 liefert nach manueller Evaluierung geringfügig bessere Ergebnisse als VGG19 für das gegebene Datenset. Für die

---

<sup>1</sup> <https://github.com/nathanhubens/Unsupervised-Image-Retrieval/blob/master/Unsupervised%20Image%20Retrieval.ipynb>

Umsetzung wird eine existierende Lösung<sup>2</sup> herangezogen und modifiziert. Es wird dabei Fine-Tuning vorgenommen, um das vortrainierte Modell auf das Datenset anzupassen.

Für die Verwendung von VGG16 müssen die Dimensionen der Bilder angegeben werden. Diese werden wie in den anderen Verfahren aus den Einstellungen des Datensets übernommen. Indem man den Parameter `include_top` auf `False` stellt werden die letzten Fully Connected Layer aus dem Netz entfernt wie in Kapitel 6.4 beschrieben. Mithilfe des Parameters `weights='imagenet'` wird ein bereits vortrainiertes Modell übernommen. Für das Fine-Tuning werden weitere Layer hinzugefügt, um mithilfe einer Klassifizierung die Gewichte des Modells anhand der vorhandenen 48 Kategorien zu modifizieren. Auch hier kommt K-Fold Cross Validation zum Einsatz. Es wird anschließend der trainierte Modell Layer VGG16 zur Ermittlung der Features entnommen.

### 8.6.7 Vorhersage

Die Generierung der Features erfolgt konfigurationsabhängig nach dem Training oder Laden des Modells. Es wird dabei die Funktion `predict` verwendet. Diese wird jeweils für alle Daten des Trainingsdatensets, sowie für alle Daten des Testdatensets durchgeführt. Es werden dabei in jedem Verfahren 512 Features generiert, um die Vergleichbarkeit der Verfahren zu ermöglichen. Im Anschluss sind jeweils alle Features für Bilder des Trainingsdatensets und des Testdatensets vorhanden. Anhand dieser Features kann im Anschluss die Suche erfolgen.

### 8.6.8 Suche

Sind die Features vorhanden, kann die Ermittlung der Suchergebnisse mithilfe von einem Abfragebild erfolgen. Es werden dazu jeweils die Features und die dazugehörigen Labels des Trainingsdatensets und des Testdatensets benötigt. Im Rahmen dieser Arbeit werden unterschiedliche Arten von Suchalgorithmen unterstützt. Diese lauten Nearest Neighbors und ANNOY. Eine Ausführung wird allerdings immer nur mit einem in den Einstellungen getroffenen Suchalgorithmus durchgeführt. Für die Suche wird das gesamte Testdatenset verwendet. Es wird also für jedes Produktbild im Testdatenset eine eigene Suchabfrage gestartet. Die Anzahl der benötigten Suchergebnisse wird auch innerhalb der Einstellungen festgelegt. Für jede Abfrage wird die Abfragedauer in Millisekunden gemessen, um anschließend einen Durchschnittswert aus diesen zu bilden. Für die Hauptevaluierungsmetrik Mean Average Precision@K wird zusätzlich die Standardabweichung berechnet.

Für die Ermittlung der Suche im Rahmen dieser Arbeit wird ANNOY verwendet. Bei der Verwendung von ANNOY wird zuerst ein sogenannter Index aufgebaut. Dieser wird anhand der Features des Trainingsdatensets aufgebaut. Für die Ermittlung der Suche im Rahmen dieser Arbeit wird ANNOY verwendet. Bei der Verwendung von ANNOY wird zuerst ein sogenannter Index aufgebaut. Dieser wird anhand der Features des Trainingsdatensets aufgebaut. Nach

---

<sup>2</sup> [https://github.com/ankonzoid/artificio/tree/master/image\\_retrieval](https://github.com/ankonzoid/artificio/tree/master/image_retrieval)

mehreren manuellen Versuchen wurde die Anzahl der Bäume für den Index auf 400 gesetzt. Bei dieser Zahl konnten die besten Ergebnisse anhand der Evaluierungsmetriken erzielt werden. Es werden somit geringfügig mehr Prozent bei MAP@K erreicht. Es wurde dabei eine manuelle Steigerung des Wertes vorgenommen. Eine weitere Erhöhung des Wertes brachte nur minimalst bessere Ergebnisse, aber dafür zu Lasten von erhöhter Berechnungsdauer. Eine höhere Anzahl an Bäumen bringt auch genauere Ergebnisse mit sich, jedoch wird der Index dadurch größer. Mithilfe der Funktion `get_nns_by_vector` wird die Suche anhand des Abfragebildes und der in den Einstellungen definierten Suchanzahl durchgeführt.

Nachdem die Suche anhand des Produktbildes im Testdatenset durchgeführt wurde, werden diese Ergebnisse zusammengefasst und für die weitere Verwendung in den Phasen Evaluierung und Visualisierung zusammengefasst.

### 8.6.9 Evaluierung

Die Evaluierung der Suchergebnisse erfolgt in dieser Phase. Für jede einzelne Kategorie im modifizierten Datenset werden P@K und MAP@K anhand der Suchergebnisse erhoben. Diese werden aus dem Durchschnitt der einzelnen Abfragen pro Kategorie im jeweiligen Datenset gebildet. Diese Werte werden außerdem auch über alle Kategorien gerechnet, um so einen Vergleich der Verfahren zu ermöglichen. Die Daten werden ausgegeben, sowie für die weitere Verwendung in MLflow gespeichert. Das Programm wird dazu mit unterschiedlichen Einstellungen für die Anzahl der Suchergebnisse ausgeführt.

### 8.6.10 Visualisierung

Bei der Visualisierung werden die Suchergebnisse aufbereitet und mithilfe von matplotlib grafisch dargestellt. Die folgende Abbildung zeigt ein beispielhaftes Suchergebnis:

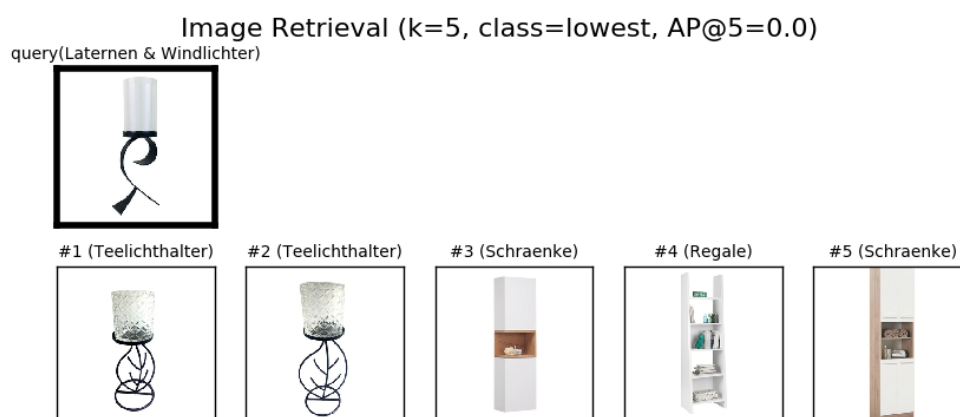


Abbildung 8-7: Suchergebnis visuelle Darstellung

Wie in Abbildung 8-7 ersichtlich besteht ein Suchergebnis aus dem Abfragebild und mehreren ähnlichen Produkten. Der Kategorienname des jeweiligen Produkts wird oberhalb des Bildes angezeigt. Ebenso wird die Platzierung innerhalb der Suchergebnisse dargestellt. Die Variable `k` steht für die Anzahl der zurückgelieferten Suchergebnisse. In diesem Fall beträgt diese fünf.

Die Variable class gibt dabei für die Kategorie des Abfragebilds an, dass dieses den höchsten oder niedrigsten AP@5 Wert annimmt. AP@5 steht dabei für den entsprechenden AP@K Wert. Die Visualisierung wurde auf Basis einer existierenden Umsetzung<sup>3</sup> vorgenommen und für den Anwendungsfall modifiziert.

Weiters werden alle Maßnahmen zur Veranschaulichung der Features des Trainingsdatensets mithilfe von TensorBoard getroffen. Mit einer Dimensionality reduction Technik, wie zum Beispiel t-SNE können die Bilder und deren Distanz zu einander grafisch aufbereitet werden. Für die Aufbereitung wird eine Datei benötigt, in welcher alle Features abgelegt werden. Zusätzlich wird eine Datei benötigt, welche alle Labels beinhaltet. Ein Bild, welches alle Bilder des Trainingsdatensets in Miniaturform beinhaltet, wird ebenso generiert und gespeichert. Die Reihenfolge der Bilder ist dabei wichtig. Abschließend wird eine Konfigurationsdatei für TensorBoard angelegt. Mithilfe des Befehls `tensorboard --logdir .` im Verzeichnis der Konfigurationsdatei startet TensorBoard und zeigt nach Aufruf von <http://localhost:6006/#projector> im Browser die Visualisierung der Produktbilder an. Gute Ergebnisse bei der Visualisierung werden in der nachfolgenden Abbildung gezeigt:



Abbildung 8-8: Visualisierung mit TensorBoard und t-SNE gute Ergebnisse

Abbildung 8-8 zeigt dabei einen Auszug der Visualisierung. Innerhalb dieses Bereichs werden gute Ergebnisse erzielt. Es ist erkennbar, dass Stühle gruppiert dargestellt werden. Auch ein paar Ausreißer also Produkte, welche sich nicht ähnlich sind, sind erkennbar.

---

<sup>3</sup> [https://github.com/ankonzoid/artificio/tree/master/image\\_retrieval](https://github.com/ankonzoid/artificio/tree/master/image_retrieval)

Die nachfolgende Abbildung zeigt eher schlechtere Ergebnisse bei der Visualisierung:



*Abbildung 8-9: Visualisierung mit TensorBoard und t-SNE schlechte Ergebnisse*

Abbildung 8-9 zeigt dabei eher schlechtere Ergebnisse. Es ist darauf zu erkennen, dass unterschiedliche Objekte, wie zum Beispiel Schränke, Bilder, Kommoden und Kissen zusammen gruppiert werden. Die Visualisierung hat daher für diese Produkte nicht gut funktioniert.

Aufgrund der schweren Vergleichbarkeit über unterschiedliche Verfahren wird diese Visualisierung im Verlauf der weiteren Arbeit nur zur internen Ermittlung der Ergebnisse herangezogen.

## 9 MESSUNG

Die für die Messungen benötigten Mechanismen wurden innerhalb der Verfahren integriert. Eine Sicherstellung der Validität, Zuverlässigkeit und Objektivität muss vorher berücksichtigt werden. Dafür müssen entsprechende Probleme identifiziert und entsprechende Maßnahmen dafür dargelegt werden. Anschließend kann die Messung je Verfahren erfolgen. Es werden dabei die Werte des im Kapitel 7.5 definierten Kriterienkatalog abgelesen und die Ergebnisse dargelegt.

### 9.1 Sicherstellung der Validität, Zuverlässigkeit und Objektivität

Bei der Validität wird überprüft, ob die Werte überhaupt das Richtige messen. Es wird darauf geachtet, dass glaubwürdige Ergebnisse zurückgeliefert werden. Anhand der vorher behandelten Hypothese und anhand der Sinnhaftigkeit kann die Validität der Messwerte sichergestellt werden. Es würde somit keinen Sinn machen einen negativen Wert bei der Berechnung der Mean Average Precision@K zu erhalten.

Damit die Validität der Ergebnisse gegeben ist wurden folgende Maßnahmen durchgeführt:

- Jedes Verfahren arbeitet mit den gleichen Daten
- Der verwendete Programmcode ist bis auf die Modellbildung und der Trainingsdatenvorbereitung derselbe für alle Verfahren
- Geeignete Metriken zur Evaluierung der Suchergebnisse werden verwendet
- Die Schlussfolgerungen beziehen sich exakt auf die im Rahmen dieser Arbeit umgesetzten Verfahren und dem verwendeten Datenset

Die Zuverlässigkeit stellt die Reproduzierbarkeit der Ergebnisse sicher. Bei einem wiederholten Durchlauf unter denselben Rahmenbedingungen können dieselben Ergebnisse erzielt werden. Folgende Maßnahmen werden getroffen, um die Zuverlässigkeit sicherzustellen:

- Bei der Messung werden alle nicht benötigten Applikationen am Computer geschlossen, um von außen einwirkende Einflüsse zu minimieren.
- Es wird bei der Messung immer derselbe Computer verwendet
- Die Messungen werden kurz hintereinander durchgeführt
- Zur Durchschnittsbildung werden insgesamt drei Messungen pro Verfahren durchgeführt
- Die Reproduzierbarkeit ist gegeben
- Die Interpretationen und Schlussfolgerungen sind verständlich erklärt

Die Objektivität muss ebenso sichergestellt sein. Folgende Maßnahmen wurden zur Sicherstellung der Objektivität getroffen:

- Literaturrecherche zur Erhebung der Verfahren
- Literaturrecherche zur Ermittlung von Evaluierungskriterien

## 9.2 Durchführung

Die Messung der Ergebnisse erfolgt am lokalen Gerät unter Berücksichtigung der Maßnahmen zur Sicherstellung der Zuverlässigkeit. Für den Anwendungsfall innerhalb dieser Arbeit sind dabei die Werte für fünf, 25 und 50 Suchergebnisse von Relevanz. Die Anzahl der Suchergebnisse kann jedoch auch andere Metriken wie beispielsweise die Abfragezeit für Suchergebnisse in Millisekunden beeinflussen. Die visuellen Ergebnisse anhand festdefinierter Testbilder werden nur anhand der ersten fünf Produkte im Rahmen der ersten Messung dargestellt. Es wird jeweils ein Auszug der aus der Kategorie mit dem höchsten und mit dem niedrigsten Mean Average Precision@K Wert gegeben und diskutiert. Es werden somit folgende Messwerte bei jeder Messung erhoben:

- Mean Average Precision@5, Mean Average Precision@25, Mean Average Precision@50
- Precision@5, Precision@25, Precision@50
- Trainingsdauer des Modells in Sekunden
- Abfragezeit für Suchergebnisse in Millisekunden bei Produktanzahl von fünf, 25 und 50

Folgende Messwerte sind bei jeder Messung gleich und werden daher nur einmalig je Verfahren erhoben:

- Anzahl an Codezeilen
- Aufwand in Arbeitsstunden für Umsetzung
- Vorbereitungsaufwand der Daten für das Verfahren notwendig
- Visuelle Ergebnisse anhand festdefinierter Testbilder

## 9.3 Ergebnisse

Nachdem die Messwerte definiert wurden werden diese nun für jedes Verfahren ermittelt. Dies erfolgt jeweils drei Mal und es werden daraus die jeweiligen Durchschnittswerte gebildet. Zahlen mit Nachkommastellen werden jeweils auf zwei Nachkommastellen gerundet dargestellt. Die Bezeichnung Avg steht dabei für Durchschnitt. Std bedeutet die jeweilige Standardabweichung. P@K steht dabei für Precision@K und MAP@K steht für Mean Average Precision@K. Die Ergebnisse dieser Messungen werden innerhalb dieses Kapitels veranschaulicht.



### 9.3.1 Autoencoder

Die Ergebnisse tabellarisch aufbereitet und dargestellt mit jeweils drei Messungen im Trainingsmodus des Autoencoder Verfahrens sehen dabei wie folgt aus:

	Messung 1	Messung 2	Messung 3	Avg
<b>MAP@5 (Avg (±Std)) [%]</b>	64,21 (±41,24)	63,64 (±41,18)	65,33 (±40,99)	<b>64,39 (±41,14)</b>
<b>MAP@25 (Avg (±Std)) [%]</b>	54,52 (±40,59)	54,06 (±40,35)	55,62 (±40,57)	<b>54,73 (±40,50)</b>
<b>MAP@50 (Avg (±Std)) [%]</b>	49,97 (±40,05)	49,24 (±39,78)	51,11 (±40,34)	<b>50,11 (±40,06)</b>
<b>P@5 (Avg) [%]</b>	67,39	67,10	68,51	<b>67,66</b>
<b>P@25 (Avg) [%]</b>	60,46	60,11	61,37	<b>60,65</b>
<b>P@50 (Avg) [%]</b>	56,66	55,97	57,59	<b>56,74</b>
<b>Abfragezeit 5 Suchergebnisse (Avg) [ms]</b>	0,94	1,06	0,93	<b>0,98</b>
<b>Abfragezeit 25 Suchergebnisse (Avg) [ms]</b>	1,48	1,62	1,55	<b>1,55</b>
<b>Abfragezeit 50 Suchergebnisse (Avg) [ms]</b>	2,03	2,47	2,46	<b>2,32</b>
<b>Trainingsdauer [s]</b>	157,97	159,39	162,80	<b>160,05</b>

Tabelle 1: Autoencoder Evaluierungsmetriken

Tabelle 1 zeigt die Ergebnisse der unterschiedlichen Messreihen. Die Standardabweichung ist bei den MAP@K Werten sehr hoch. Diese wird über die jeweiligen MAP@K Werte der Kategorien gerechnet. Dies besagt, dass eine hohe Streuung der Ergebnisse vorhanden ist und somit viele gute Ergebnisse, aber auch viele schlechte Ergebnisse innerhalb bestimmter Kategorien gegeben sind.

Generelle Metriken des Autoencoder Verfahrens sind nachfolgend aufgelistet:

<b>Anzahl der Codezeilen</b>	<b>54</b>
<b>Aufwand für Umsetzung [Stunden]</b>	<b>4</b>
<b>Vorbereitungsaufwand der Daten für Verfahren [Ja, Nein]</b>	<b>Nein</b>

Tabelle 2: Autoencoder generelle Metriken

Tabelle 2 zeigt subjektiv den Aufwand des Verfahrens. Die Anzahl der Codezeilen beläuft sich auf 54, wobei insgesamt für die Umsetzung rund vier Stunden notiert wurden. Es ist kein zusätzlicher Aufwand zur Vorbereitung der Daten für das Training vorhanden.

Die nachfolgende Tabelle zeigt die jeweiligen Ergebnisse der besten fünf Kategorien sowie der schlechtesten fünf Kategorien sortiert anhand der Ergebnisse von MAP@5:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg (±Std)) [%]	MAP@25 (Avg (±Std)) [%]	MAP@50 (Avg (±Std)) [%]
1	Bettwasche	154	38	93,86 (±22,59)	86,23 (±26,36)	70,42 (±28,2)
2	Sofas	2788	697	92,95 (±19,82)	89,71 (±20,27)	87,28 (±21,21)
3	Leintucher	279	69	88,86 (±18,94)	81,29 (±20,85)	69,50 (±20,61)
4	Sessel	1152	288	87,73 (±28,51)	83,11 (±30,65)	79,31 (±32,34)
5	Matratzen	518	129	87,12 (±31,53)	89,03 (±23,51)	88,45 (±23,67)
44	Vitrinen	204	51	13,44 (±19,54)	6,57 (±11,02)	4,83 (±6,68)
45	Regale	371	92	12,54 (±19,84)	5,79 (±7,61)	4,14 (±5,06)
46	Teelichthalter	91	22	6,71 (±9,35)	1,99 (±2,54)	1,05 (±1,29)
47	Nachtkastchen	137	34	3,72 (±7,66)	1,50 (±1,90)	0,96 (±1,09)
48	Laternen & Windlichter	84	20	0,70 (±2,36)	0,63 (±1,05)	0,57 (±0,85)

Tabelle 3: Autoencoder Ergebnisse Kategorien

Tabelle 3 zeigt, dass die Spannweite der Ergebnisse bei MAP@5 in den jeweiligen Kategorien sehr hoch ist. Die beste Kategorie erreicht dabei einen Wert von 93,66 Prozent wohingegen die schlechteste Kategorie einen Wert von 0,7 Prozent liefert. Dies liegt an der jeweiligen visuellen Ähnlichkeit der betroffenen Produkte in anderen Kategorien. Dies verdeutlichen auch die nachfolgenden visuellen dargestellten Suchergebnisse. Die Anzahl der Bilder scheint dabei nur bedingt eine Rolle zu spielen. Die Kategorie am ersten Platz weist dabei nur 154 Trainingsbilder auf wohingegen die Kategorie auf Rang 45 immerhin 371 Trainingsbilder aufweist.

Die Kategoriennamen wurden für die Umsetzung der Verfahren leicht modifiziert. Bei Bedarf wurden dabei Umlaute und sonstige Sonderzeichen entfernt. Die modifizierten Kategoriennamen werden daher im Rahmen der gesamten Arbeit verwendet.

Die folgende Grafik veranschaulicht die besten Ergebnisse der ersten fünf Kategorien visuell:

## Top 5 categories best results

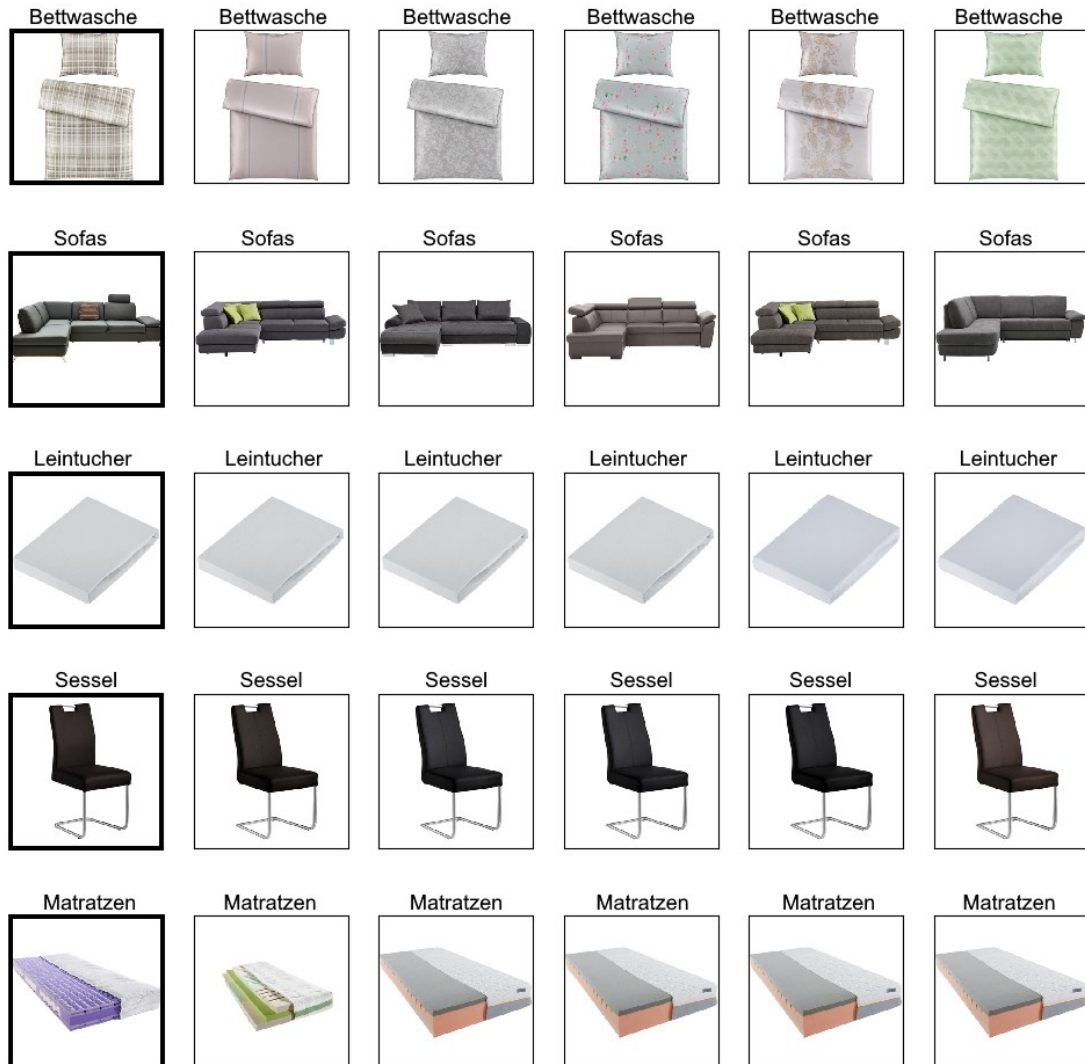


Abbildung 9-1: Autoencoder beste Ergebnisse ersten fünf Kategorien visuell

Abbildung 9-1 veranschaulicht, dass die visuellen Ergebnisse jeweils ein Produkt der Kategorie des jeweiligen Abfragebilds (erstes Bild in jeder Zeile) darstellen. Es werden dabei teilweise auch Farben berücksichtigt, wobei bei manchen Suchergebnissen die Farbe vernachlässigt wird.

Die folgende Grafik veranschaulicht die schlechtesten Ergebnisse der letzten fünf Kategorien visuell:

## Bottom 5 categories worst results

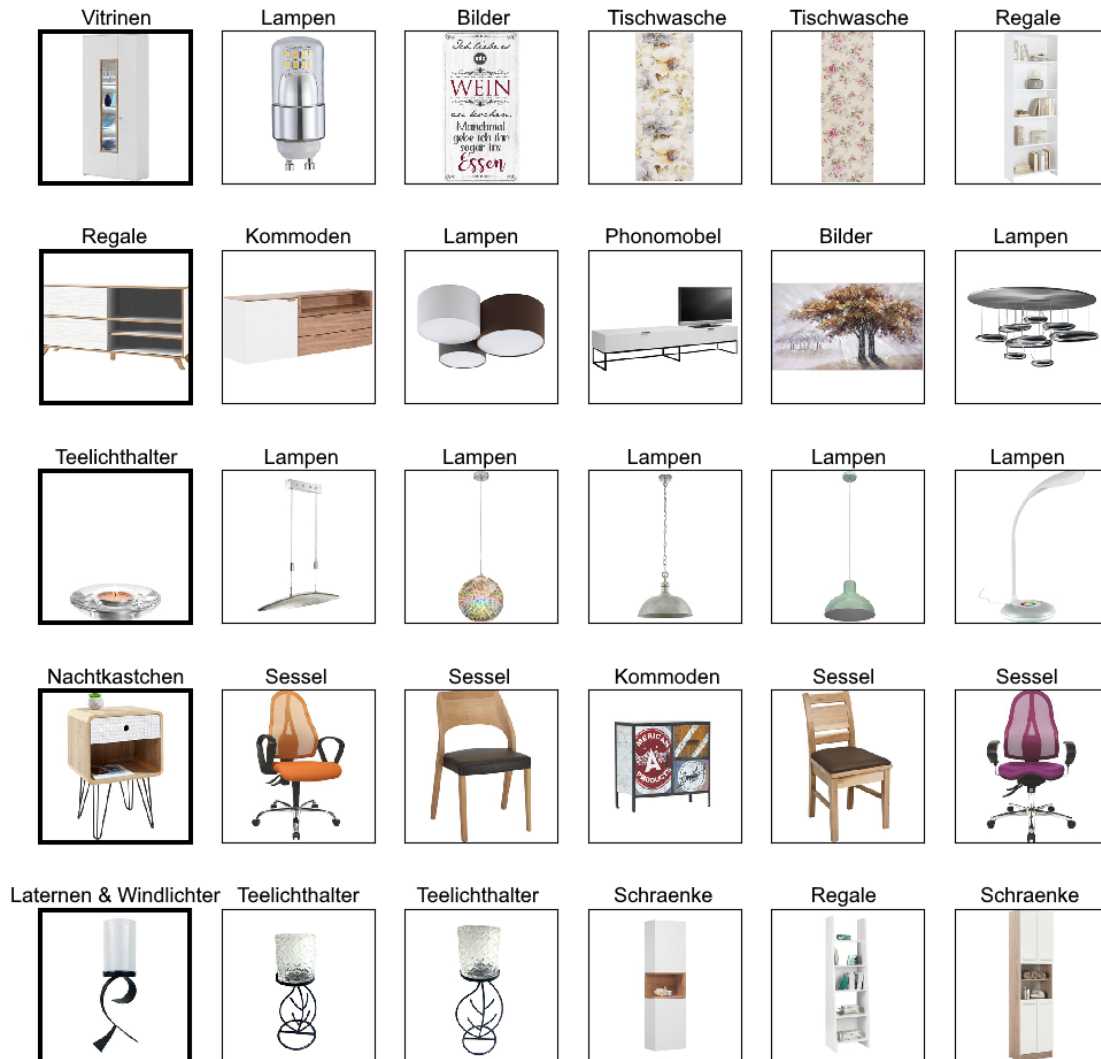


Abbildung 9-2: Autoencoder schlechtesten Ergebnisse letzten fünf Kategorien visuell

Abbildung 9-2 veranschaulicht, dass visuell teilweise zwar eine gewisse Ähnlichkeit mit dem Abfragebild zu erkennen ist, jedoch die Kategorie der Suchergebnisse häufig nicht mit dem des Abfragebilds übereinstimmt. Es drängt sich die Frage auf, inwiefern diese Kategorien überhaupt für den visuellen Vergleich geeignet sind? Gegebenenfalls könnten zusätzliche Optimierungen helfen, um auch in diesen Kategorien praxistaugliche Ergebnisse zu erreichen.

### 9.3.2 Siamese Network

Die Ergebnisse tabellarisch aufbereitet und dargestellt mit jeweils drei Messungen im Trainingsmodus des Siamese Network Verfahrens sehen dabei wie folgt aus:

	Messung 1	Messung 2	Messung 3	Avg
<b>MAP@5 (Avg (±Std)) [%]</b>	63,81 (±41,16)	64,48 (±41,37)	63,43 (±41,73)	<b>63,91 (±41,42)</b>
<b>MAP@25 (Avg (±Std)) [%]</b>	55,39 (±41,10)	56,25 (±40,84)	55,32 (±41,38)	<b>55,65 (±41,11)</b>
<b>MAP@50 (Avg (±Std)) [%]</b>	51,96 (±40,97)	52,63 (±40,74)	51,74 (±41,11)	<b>52,11 (±40,94)</b>
<b>P@5 (Avg) [%]</b>	67,28	67,93	66,71	<b>67,31</b>
<b>P@25 (Avg) [%]</b>	61,32	62,21	61,25	<b>61,59</b>
<b>P@50 (Avg) [%]</b>	58,54	59,41	58,42	<b>58,79</b>
<b>Abfragezeit 5 Suchergebnisse (Avg) [ms]</b>	0,91	0,91	0,99	<b>0,94</b>
<b>Abfragezeit 25 Suchergebnisse (Avg) [ms]</b>	1,50	1,37	1,43	<b>1,43</b>
<b>Abfragezeit 50 Suchergebnisse (Avg) [ms]</b>	2,02	2,06	1,89	<b>1,99</b>
<b>Trainingsdauer [s]</b>	394,05	393,35	402,03	<b>396,48</b>

Tabelle 4: Siamese Network Evaluierungsmetriken

Tabelle 4 zeigt die Ergebnisse des Siamese Network. Es wurden dabei drei Messreihen vollzogen. Die hohe Standardabweichung ist bei den MAP@K Werten auffallend. Diese wird über die jeweiligen MAP@K Werte der Kategorien gerechnet. Es ist somit eine hohe Streuung innerhalb der Ergebnisse vorhanden. Innerhalb der Kategorien gibt es viele gute Ergebnisse, aber auch viele schlechte Ergebnisse.

Generelle Metriken des Siamese Network Verfahrens sind nachfolgend aufgelistet:

<b>Anzahl der Codezeilen</b>	<b>105</b>
<b>Aufwand für Umsetzung [Stunden]</b>	<b>6</b>
<b>Vorbereitungsaufwand der Daten für Verfahren [Ja, Nein]</b>	<b>Ja</b>

Tabelle 5: Siamese Network generelle Metriken

Tabelle 5 zeigt subjektiv den Aufwand des Verfahrens. Die Anzahl der Codezeilen beläuft sich auf 105, wobei insgesamt für die Umsetzung rund sechs Stunden angenommen wurden. Sowohl die Anzahl der Codezeilen, also auch die Dauer für die Umsetzung ist gegenüber dem Autoencoder Verfahren größer. Für die Aufbereitung der Trainingsdaten ist zusätzlicher Aufwand für die Generierung der Bildpaare vorhanden.

Die nachfolgende Tabelle zeigt die jeweiligen Ergebnisse der ersten fünf Kategorien sowie der letzten fünf Kategorien basierend auf der sortierten Reihenfolge anhand MAP@5:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg (±Std)) [%]	MAP@25 (Avg (±Std)) [%]	MAP@50 (Avg (±Std)) [%]
1	Leintucher	279	69	100,00 (±0,00)	95,18 (±13,88)	91,75 (±17,37)
2	Bademantel	128	32	100,00 (±0,00)	99,18 (±1,97)	98,46 (±3,15)
3	Bettwasche	154	38	97,11 (±14,87)	95,54 (±19,08)	95,20 (±20,36)
4	Lattenroste	260	64	93,75 (±24,40)	89,68 (±26,45)	84,31 (±28,64)
5	Matratzen	518	129	92,25 (±20,66)	87,22 (±19,96)	81,00 (±22,03)
44	Zierkissen	101	25	13,96 (±14,29)	8,62 (±6,64)	7,79 (±5,37)
45	Vitrinen	204	51	12,67 (±21,38)	6,73 (±7,83)	5,31 (±5,41)
46	Laternen & Windlichter	84	20	11,77 (±19,40)	3,68 (±5,86)	2,22 (±3,34)
47	Nachtkastchen	137	34	5,96 (±12,59)	2,19 (±4,64)	1,39 (±2,73)
48	Teelichthalter	91	22	3,11 (±6,18)	1,34 (±1,87)	0,79 (±1,03)

Tabelle 6: Siamese Network Ergebnisse Kategorien

Tabelle 6 zeigt, dass die ersten beiden Kategorien einen MAP@5 einen Wert von 100 Prozent aufweisen. Auch die Kategorie auf Rang 48 liefert mit 3,11 Prozent bessere Ergebnisse als die Kategorie auf demselben Platz im Autoencoder Verfahren mit 0,7 Prozent. Die Spannweite der Ergebnisse bei MAP@5 in den jeweiligen Kategorien ist jedoch sehr hoch. Die Anzahl der Bilder scheint keine Rolle zu spielen, da die Kategorie am zweiten Platz 128 Trainingsbilder aufweist, während die Kategorie auf Rang 47 128 Trainingsbilder aufweist.

Die folgende Grafik veranschaulicht die besten Ergebnisse der ersten fünf Kategorien visuell:

## Top 5 categories best results

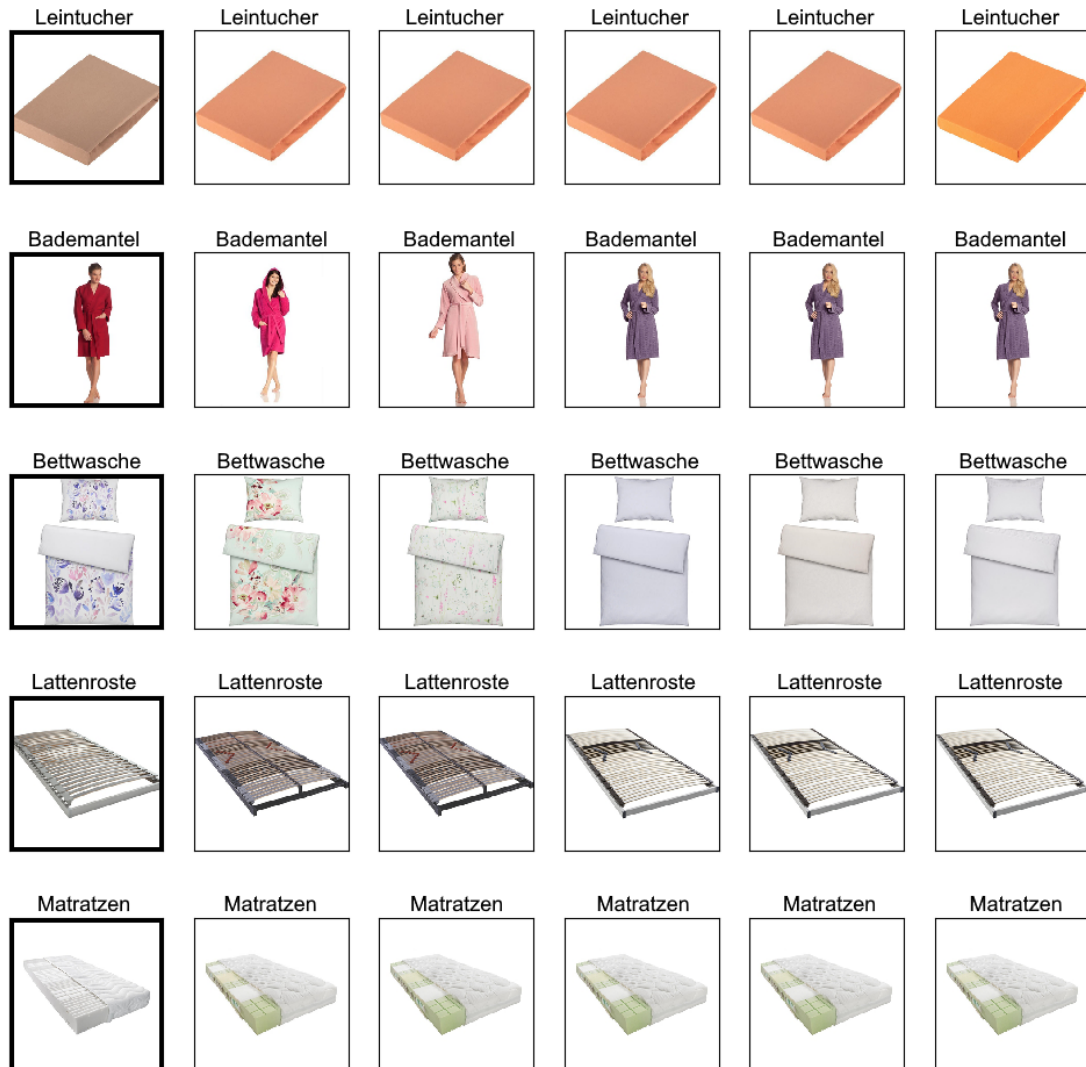


Abbildung 9-3: Siamese Network beste Ergebnisse ersten fünf Kategorien visuell

Abbildung 9-3 veranschaulicht visuelle Ergebnisse der Top fünf Kategorien. Die Kategorie der Suchergebnisse stimmen mit der des Abfragebilds jeweils überein. Auch Farben werden teilweise berücksichtigt.

Die folgende Grafik veranschaulicht die schlechtesten Ergebnisse der letzten fünf Kategorien visuell:

## Bottom 5 categories worst results

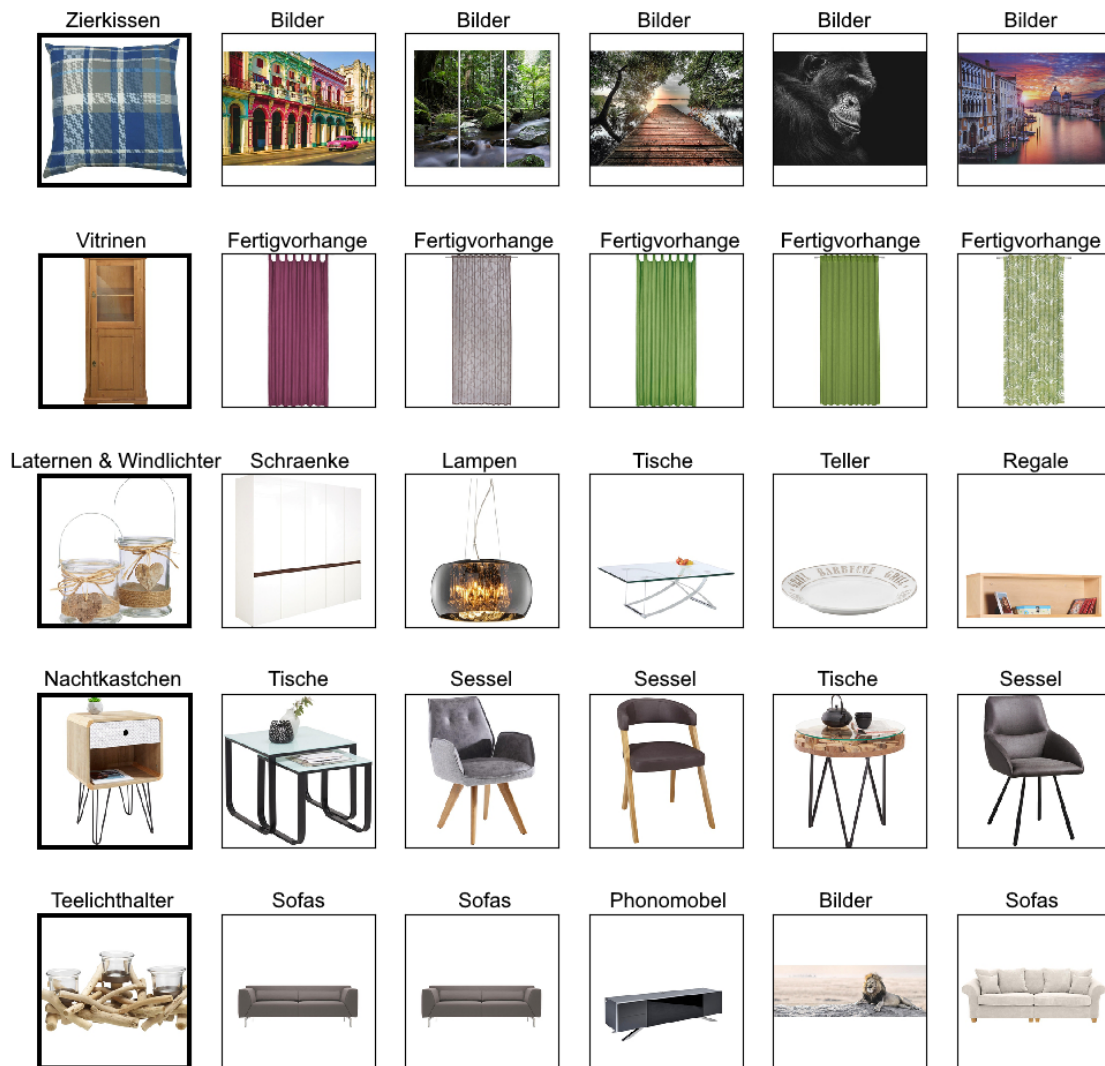


Abbildung 9-4: Siamese Network schlechtesten Ergebnisse letzten fünf Kategorien visuell

Abbildung 9-4 zeigt die schlechtesten Ergebnisse der letzten fünf Kategorien für das Siamese Network Verfahren. Es ist dabei keine Übereinstimmung der Kategorie der Suchergebnisse mit der des Abfragebilds vorhanden. Visuell ist teilweise jedoch eine gewisse Ähnlichkeit mit dem Abfragebild zu erkennen. Weitere Optimierungen wären somit notwendig, um bessere Ergebnisse zu erhalten.



### 9.3.3 Transfer Learning

Die Ergebnisse tabellarisch aufbereitet und dargestellt mit jeweils drei Messungen des Transfer Learning Verfahrens mit VGG16 sehen dabei wie folgt aus:

	Messung 1	Messung 2	Messung 3	Avg
<b>MAP@5 (Avg (±Std)) [%]</b>	71,58 (±40,45)	71,87 (±40,20)	73,37 (±39,44)	<b>72,27 (±40,03)</b>
<b>MAP@25 (Avg (±Std)) [%]</b>	68,52 (±40,80)	69,24 (±40,39)	70,95 (±39,75)	<b>69,57 (±40,31)</b>
<b>MAP@50 (Avg (±Std)) [%]</b>	67,31 (±41,13)	68,37 (±40,59)	70,19 (±39,96)	<b>68,62 (±40,56)</b>
<b>P@5 (Avg) [%]</b>	74,43	74,87	76,25	<b>75,19</b>
<b>P@25 (Avg) [%]</b>	73,25	73,92	75,51	<b>74,23</b>
<b>P@50 (Avg) [%]</b>	72,45	73,42	75,12	<b>73,66</b>
<b>Abfragezeit 5 Suchergebnisse (Avg) [ms]</b>	0,78	0,88	0,90	<b>0,85</b>
<b>Abfragezeit 25 Suchergebnisse (Avg) [ms]</b>	1,13	1,39	1,38	<b>1,30</b>
<b>Abfragezeit 50 Suchergebnisse (Avg) [ms]</b>	1,63	1,83	1,83	<b>1,76</b>
<b>Trainingsdauer [s]</b>	1094,28	1039,75	1022,20	<b>1052,08</b>

Tabelle 7: Transfer Learning Evaluierungsmetriken

Tabelle 7 zeigt die Ergebnisse der unterschiedlichen Messreihen. Anhand der hohen Standardabweichung bei den MAP@K Werten ist eine hohe Streuung der Ergebnisse zu erkennen.

Generelle Metriken des Transfer Learning Verfahrens sind nachfolgend aufgelistet:

<b>Anzahl der Codezeilen</b>	<b>57</b>
<b>Aufwand für Umsetzung [Stunden]</b>	<b>4</b>
<b>Vorbereitungsaufwand der Daten für Verfahren [Ja, Nein]</b>	<b>Nein</b>

Tabelle 8: Transfer Learning generelle Metriken

Tabelle 8 listet die subjektiven Metriken des Verfahrens auf. Es werden dabei 57 Codezeilen für die Umsetzung benötigt. Die Umsetzungsdauer ist mit vier Stunden von allen Verfahren am geringsten. Ein zusätzlicher Vorbereitungsaufwand der Daten ist nicht gegeben. Das Fine-Tuning des Modells wird nicht als Vorbereitungsaufwand angesehen.

Die nachfolgende Tabelle zeigt die jeweiligen Ergebnisse der ersten fünf Kategorien sowie der letzten fünf Kategorien basierend anhand der Reihenfolge sortiert mit MAP@5:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg (±Std)) [%]	MAP@25 (Avg (±Std)) [%]	MAP@50 (Avg (±Std)) [%]
1	Matratzen	518	129	99,07 (±8,97)	99,08 (±8,95)	99,13 (±8,86)
2	Bademantel	128	32	98,50 (±5,90)	98,07 (±3,53)	98,27 (±2,32)
3	Betten	1246	311	95,97 (±18,66)	95,42 (±18,61)	95,30 (±18,77)
4	Leintucher	279	69	95,65 (±20,54)	93,12 (±22,98)	91,43 (±24,06)
5	Stumpenkerzen	191	47	93,53 (±13,54)	90,82 (±17,15)	90,48 (±18,85)
44	Diffuser	89	22	21,52 (±30,90)	8,28 (±9,98)	5,34 (±6,78)
45	Nachtkastchen	137	34	15,23 (±22,80)	10,89 (±15,57)	8,33 (±11,10)
46	Tischwasche	240	60	11,10 (±21,55)	5,13 (±10,29)	3,42 (±6,19)
47	Teelichthalter	91	22	6,58 (±10,58)	2,82 (±3,88)	1,83 (±2,28)
48	Laternen & Windlichter	84	20	1,80 (±3,71)	1,79 (±2,94)	1,67 (±3,27)

Tabelle 9: Transfer Learning Ergebnisse Kategorien

Tabelle 9 veranschaulicht die Ergebnisse der Kategorien des Transfer Learning Verfahrens. Interessant sind dabei die Werte von 99,08 Prozent MAP@25 und 99,13 Prozent MAP@25 der Kategorie auf Rang 1. Diese sind im Vergleich zu MAP@5 mit 99,07 Prozent sogar erhöht. Dies kann aufgrund der Platzierung von relevanten Suchergebnissen innerhalb der zurückgelieferten Ergebnisse der Fall sein. Die Kategorie mit den schlechtesten Ergebnissen hat einen MAP@5 Wert von 1,8 Prozent. Auch hier könnten Optimierungen zu einer Verbesserung der Ergebnisse führen.

Die folgende Grafik veranschaulicht die besten Ergebnisse der ersten fünf Kategorien visuell:

## Top 5 categories best results



Abbildung 9-5: Transfer Learning beste Ergebnisse ersten fünf Kategorien visuell

In Abbildung 9-5 ist zu erkennen, dass auch hier alle Suchergebnisse mit der Kategorie des Abfragebilds übereinstimmen. Auch Farben werden hier teilweise berücksichtigt, wie am Beispiel des roten Leintuches zu erkennen ist.

Die folgende Grafik veranschaulicht die schlechtesten Ergebnisse der letzten fünf Kategorien visuell:

## Bottom 5 categories worst results

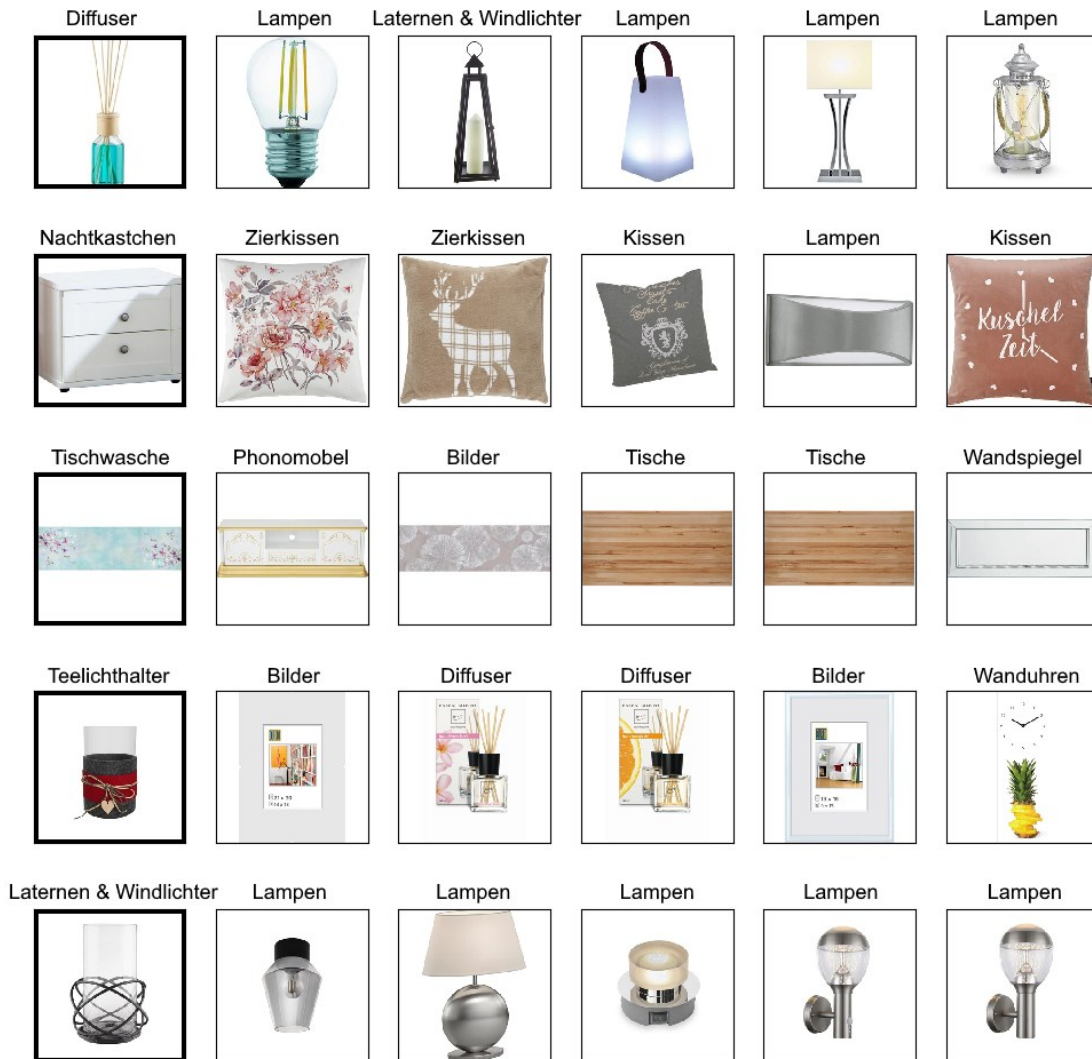


Abbildung 9-6: Transfer Learning schlechtesten Ergebnisse letzten fünf Kategorien visuell

Abbildung 9-6 zeigt die schlechtesten Ergebnisse aus den letzten fünf Kategorien des Transfer Learning Verfahrens. Innerhalb der dargestellten Ergebnisse gibt es keine Übereinstimmung der Kategorie des Abfragebilds mit den Kategorien der Suchergebnisse. Auffallend ist jedoch die optische Ähnlichkeit der Ergebnisse. Das Hinzuziehen von weiteren Produktdaten könnte für eine Verbesserung der Ergebnisse innerhalb dieser Kategorien sorgen.

Auch ein Vergleich mit einem vortrainierten VGG16 Modell ohne Fine-Tuning wurde durchgeführt. Bei drei Messreihen wurde ein MAP@5 Wert von durchschnittlich 61,68 Prozent erreicht. Im Vergleich zum Autoencoder mit durchschnittlich 64,39 Prozent und zum Siamese Network Verfahren mit durchschnittlich 63,91 Prozent, erreicht somit VGG16 ohne Fine-Tuning auch gute Ergebnisse.

## 10 BEWERTUNG

Nachdem die Ergebnisse nun vorliegen, kann damit begonnen werden, die Verfahren zu vergleichen. Dies erfolgt anhand der Durchschnittswerte der Metriken der Messwerte. Die subjektiven Metriken fließen dabei nicht in die Bewertung mit ein.

Folgende Übersicht zeigt einen Vergleich der drei Verfahren und deren anhand der Messreihen generierten Durchschnittswerte der Metriken:

	Autoencoder	Siamese Network	Transfer Learning
MAP@5 (Avg (±Std)) [%]	64,39 (±41,14)	63,91 (±41,42)	<b>72,27 (±40,03)</b>
MAP@25 (Avg (±Std)) [%]	54,73 (±40,50)	55,65 (±41,11)	<b>69,57 (±40,31)</b>
MAP@50 (Avg (±Std)) [%]	50,11 (±40,06)	52,11 (±40,94)	<b>68,62 (±40,56)</b>
P@5 (Avg) [%]	67,66	67,31	<b>75,19</b>
P@25 (Avg) [%]	60,65	61,59	<b>74,23</b>
P@50 (Avg) [%]	56,74	58,79	<b>73,66</b>
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,98	0,94	<b>0,85</b>
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,55	1,43	<b>1,30</b>
Abfragezeit 50 Suchergebnisse (Avg) [ms]	2,32	1,99	<b>1,76</b>
Trainingsdauer [s]	<b>160,05</b>	396,48	1052,08

Tabelle 10: Metrikenübersicht der Verfahren

Tabelle 10 zeigt einen Überblick über alle drei Verfahren. Die jeweils fett markierten Werte zeigen für die entsprechende Metrik den Bestwert an. In der generellen Übersicht ist zu erkennen, dass Transfer Learning im Bereich der Metriken zur Evaluierung der Suchergebnisse die besten Ergebnisse erreichen kann. Auch die durchschnittliche Abfragezeit ist hier am geringsten. Zu beobachten ist allerdings auch die Trainingsdauer. Diese ist von allen drei Verfahren die Höchste. Dies ist auch dadurch gegeben, da das bereits vorhandene vortrainierte Modell mehr Layer als die anderen Modelle in den Verfahren aufweist. Es könnte somit die Annahme getroffen werden, dass die Komplexität des Netzes durchaus Einfluss auf die Trainingsdauer und die Suchergebnisse haben könnte. Der Autoencoder und das Siamese Network erreichen fast ähnliche Ergebnisse. Die Trainingsdauer selbst ist jedoch beim Siamese Network Verfahren im Vergleich zum Autoencoder jedoch mehr als doppelt so hoch. Die Abfragezeit ist allerdings beim Siamese Network geringer als beim Autoencoder.

Für den Anwendungsfall haben die Metriken MAP@5 und MAP@25 die höchste Relevanz, weshalb MAP@5 als Hauptevaluierungsmetrik herangezogen werden. Auf Basis dieser Metrik ergibt sich, dass Transfer Learning im Vergleich mit den anderen Verfahren die besten Ergebnisse zurückliefert. Dahinter liegt das Autoencoder Verfahren. Das Siamese Network liegt knapp dahinter auf dem letzten Platz. Die P@K Werte sind gegenüber den jeweiligen MAP@K

Werten erhöht. Dies liegt vermutlich daran, dass bei P@K die Reihenfolge der relevanten Ergebnisse innerhalb der zurückgelieferten Suchergebnisse nicht von Bedeutung ist.

Interessant zu betrachten sind auch die generellen Metriken. Diese sind nachfolgend aufgelistet:

	Autoencoder	Siamese Network	Transfer Learning
<b>Anzahl der Codezeilen</b>	<b>54</b>	105	57
<b>Aufwand für Umsetzung [Stunden]</b>	<b>4</b>	6	<b>4</b>
<b>Vorbereitungsaufwand der Daten für Verfahren [Ja, Nein]</b>	<b>Nein</b>	Ja	<b>Nein</b>

Tabelle 11: Übersicht genereller Metriken der Verfahren

Tabelle 11 zeigt einen Überblick über die generellen Metriken. Die markierten Werte stellen dabei wiederum die Bestwerte für die jeweilige Metrik dar. Autoencoder kann in allen drei Metriken Bestwerte erzielen. Transfer Learning benötigt nur unwesentlich mehr Zeilen Code als das Autoencoder Verfahren und teilt sich die Bestwerte im Bereich des Aufwandes für Umsetzung und des Vorbereitungsaufwands mit dem Autoencoder Verfahren. Im Siamese Network ist ein Vorbereitungsaufwand gegeben, da hier Bildpaare für das Training generiert werden.

Nachfolgende Grafik zeigt einen Vergleich der unterschiedlichen Suchergebnisse von fixierten Abfragebildern:



Abbildung 10-1: Vergleich visuelle Suchergebnisse

Abbildung 10-1 stellt zwei fixierte Abfragebilder und deren Suchergebnisse dar. Das erste Abfragebild wurde dabei aus einer der ersten fünf Kategorien herangezogen. Das zweite Abfragebild wurde dabei aus einer der letzten fünf Kategorien herangezogen. Auffallend sind dabei die Farben der Produkte der Suchergebnisse. Diese werden von keinem der Verfahren exakt ermittelt wohingegen eine Annäherung zu erkennen ist. Eine mögliche Begründung könnte darin liegen, dass die Struktur der Objekte den Farben vorangeht. Beim zweiten Abfragebild werden in keinem der Verfahren gute Ergebnisse erzielt. Eine visuelle Ähnlichkeit ist jedoch erkennbar. In Bezug auf Erkennung der Farben kann jedoch das Autoencoder Verfahren gute Ergebnisse aufweisen. Das Transfer Learning Verfahren zeigt jedoch deutlichere Unterschiede in den Farben der Suchergebnissen auf. Möglicherweise ist dies aufgrund des trainierten VGG16 Modells der Fall, wobei Bildklassifizierung trainiert wurde und Farben womöglich nicht so ins Gewicht gefallen sind. Das vortrainierte Modell wurde jedoch mit farbigen Bildern trainiert. Auffallend beim Siamese Network Verfahren ist jedoch, dass Suchergebnisse teilweise keine optische Ähnlichkeit mit dem zweiten Abfragebild aufweisen. Dies könnte aufgrund der zufälligen Auswahl der Bildpaare zustande kommen.

Interessant ist auch ein Vergleich der Suchergebnisse, bei denen die Average Precision gleich null ist. Nachfolgend werden diese visuell dargestellt. Ein entsprechendes Suchergebnis des Autoencoder Verfahrens sieht dabei wie folgt aus:

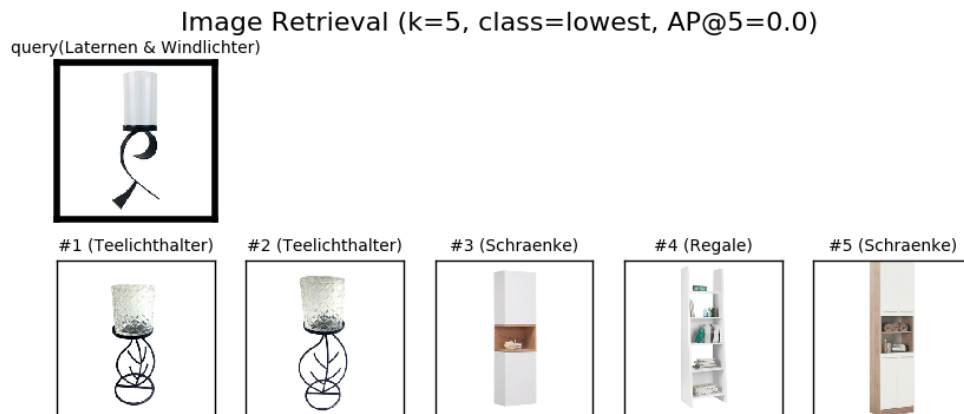


Abbildung 10-2: Autoencoder Suchergebnis mit niedrigster Average Precision

Abbildung 10-2 zeigt, dass die Suchergebnisse zumindest visuell dem Abfragebild ähnlich sind. Auch Farben werden hier gut dargestellt. Interessant sind dabei auch die ersten beiden Suchergebnisse. Diese sehen dem Abfragebild sehr ähnlich, befinden sich aber in unterschiedlichen Kategorien. Produkte aus der Kategorie Teelichthalter sind manchen Produkten aus der Kategorie Laternen & Windlichter optisch ähnlicher als Produkte innerhalb der Kategorie. Eine entsprechende Umstrukturierung der Produkte innerhalb der Kategorien könnte dabei helfen, insofern nur die visuellen Ergebnisse relevant sind.

Die nachfolgende Abbildung zeigt ein Suchergebnis des Siamese Network Verfahrens mit niedrigster Average Precision:

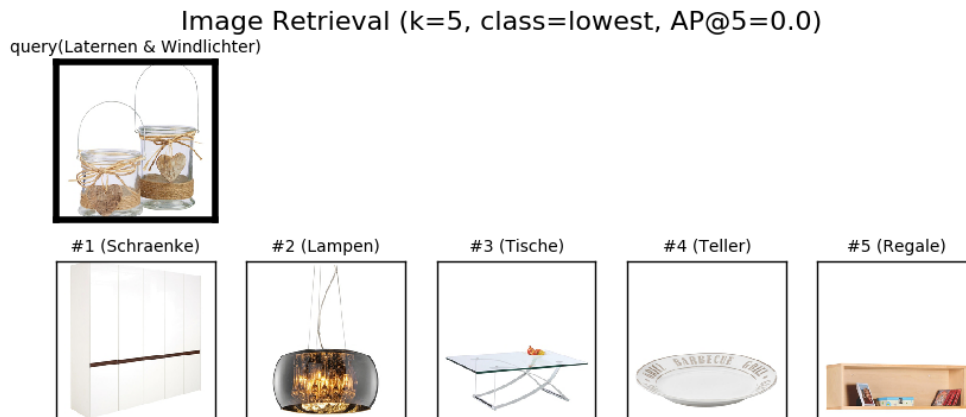


Abbildung 10-3: Siamese Network Suchergebnis mit niedrigster Average Precision

Abbildung 10-3 zeigt, dass die Suchergebnisse teilweise komplett voneinander abweichen und auch teilweise keine optische Ähnlichkeit vorhanden ist. Auch Farben selbst werden dabei nicht gut erkannt. Es werden somit weder innerhalb, als auch außerhalb der Kategorie ähnliche visuelle Produkte gefunden. Eine manuelle Kontrolle der Produkte innerhalb der Kategorie "Laternen & Windlichter" zeigt jedoch, dass zumindest optisch ähnlichere Produkte als die dargestellten Suchergebnisse vorhanden sind.

Ein Suchergebnis mit niedrigster Average Precision aus dem Transfer Learning Verfahren sieht wie folgt aus:



Abbildung 10-4: Transfer Learning Suchergebnis mit niedrigster Average Precision

Abbildung 10-4 veranschaulicht, dass die Suchergebnisse zumindest eine geringe optische Ähnlichkeit zum Abfragebild aufweisen. Auch Farben werden dabei gut erkannt, wobei Unterschiede erkennbar sind. Es stellt sich somit die Frage, inwiefern eine Umstrukturierung von Produkten innerhalb der Kategorien sinnvoll ist, da sich die Ergebnisse zumindest optisch ähnlich sind.



## 11 CONCLUSIO

Nachdem nun die Ergebnisse der Messungen vorliegen und eine Bewertung dieser vorgenommen wurde, können diese nun diskutiert werden. Die Ergebnisse werden dabei nochmal präsentiert und interpretiert. Die Erkenntnisse werden ebenfalls diskutiert. Die Hypothese und die Forschungsfrage werden beantwortet und ein Ausblick für weitere Forschungsmöglichkeiten gegeben.

### 11.1 Ergebnisvergleich

Es wurden drei Verfahren für die Suche nach ähnlichen Bildern in die Bewertung aufgenommen. Die Bewertung erfolgt anhand eines konkreten Anwendungsfalles. Die höchste Priorität liegt dabei in den Top fünf Suchergebnissen. Die Relevanz der Suchergebnisse wird mit Mean Average Precision@5 gemessen. Anhand dieser Metrik ergibt sich folgende Platzierung der Verfahren:

1. Transfer Learning (72,27 Prozent)
2. Autoencoder (64,39 Prozent)
3. Siamese Network (63,91 Prozent)

Transfer Learning liefert somit die besten Ergebnisse im Vergleich mit den anderen Verfahren. Autoencoder und Siamese Network liefern fast ähnliche Ergebnisse. Im Bereich der Farberkennung schneidet das Autoencoder Verfahren im visuellen Vergleich gut ab.

Im Bereich der generellen und mit Subjektivität behafteten Metriken schneidet Autoencoder im Vergleich mit den anderen Verfahren am besten ab. Hier ist jedoch nur ein minimaler Unterschied zum Transfer Learning Verfahren gegeben. Es wird da am wenigsten Code für die Umsetzung benötigt, die Umsetzungsdauer selbst ist im Vergleich gering und ein Datenvorbereitungsaufwand ist nicht notwendig. Aufgrund dessen kann Autoencoder von wirtschaftlicher Bedeutung für ein Unternehmen sein. Die Umsetzung kann somit im Vergleich zu den anderen Verfahren schneller und ressourcenschonender erfolgen, jedoch auf Kosten von schlechteren Suchergebnissen.

Die visuellen Ergebnisse zeigen jedoch auch, dass selbst wenn ein niedriger Average Precision@5 Wert gegeben ist, die Suchergebnisse dem Abfragebild zumindest optisch ähnlich sind.

## 11.2 Interpretation

Die Ergebnisse bezogen auf Mean Average Precision@5 Wert erreichen den Höchstwert mit 72,27 Prozent beim Transfer Learning Verfahren. Je nach Anforderung an das Empfehlungssystem sind die Ergebnisse der Verfahren ausreichend oder nicht. Im Rahmen des Anwendungsfalles dieser Arbeit werden zumindest im Schnitt 50 Prozent der Suchergebnisse innerhalb der ersten fünf Suchergebnisse richtig vorhergesagt und sind somit ausreichend.

Bezogen auf die Ergebnisse der Kategorien bei der ersten Messung im Autoencoder Verfahren haben 23 von 48 Kategorien einen höheren Wert als 50 Prozent bei Mean Average Precision@5. Beim Siamese Network Verfahren erreichen 22 von 48 Kategorien einen höheren Wert als 50 Prozent bei Mean Average Precision@5. Im Transfer Learning Verfahren sind es 25 von 48 Kategorien. Somit gibt es viele Kategorien, welche weniger als 50 Prozent erreichen. Es stellt sich somit die Frage, inwiefern der reine visuelle Vergleich für diese Kategorien überhaupt geeignet ist und ob hier zusätzlich Metadaten hinzugezogen werden sollen oder generell mehr Daten vorhanden sein müssen.

Werden bessere Ergebnisse gefordert, so müssen weitere Optimierungen wie zum Beispiel die Anpassung des Datensets und der Kategorien, eine mögliche Augmentierung, eine Umsetzung der in den Verfahren beschriebenen Techniken, welche in der Basisvariante nicht umgesetzt wurden oder ein Re-Ranking und Einbeziehung von Metadaten umgesetzt werden.

Interessant sind auch die visuellen Ergebnisse. Diese sind bei jedem Verfahren recht ähnlich. Auch die am besten und am schlechtesten abschneidenden Kategorien bezogen auf Mean Average Precision@5 sind teilweise über die Verfahren hinweg dieselben. Dennoch ist auffallend, dass selbst bei den am schlechtesten abschneidenden Kategorien mit den niedrigsten Average Precision@5 Werten, eine gewisse visuelle Ähnlichkeit der Suchergebnisse mit dem Abfragebild besteht.

## 11.3 Erkenntnisgewinn

Aufgrund der Ergebnisse muss man sich die Frage stellen, welchen Stellenwert vorherige Optimierungen am Datenset haben und inwiefern noch weitere Optimierungen notwendig sind. In einigen Kategorien sind die Ergebnisse in Bezug auf Mean Average Precision@K sehr gut, wobei in anderen Kategorien schlechte Ergebnisse vorhanden sind. Ein möglicher Grund für schlechte Ergebnisse in einigen Kategorien liegt in der Varietät in den Kategorien selbst. Visuell ähnliche Produkte können somit in anderen Kategorien liegen, werden dementsprechend von den Verfahren im Rahmen der Suche vorhergesagt und führen zu schlechten Ergebnissen bei der Evaluierung der Verfahren. Möglicherweise sind auch nicht genug Bilder für das Training des Modells vorhanden. Eine Augmentierung könnte Abhilfe schaffen.

Weiters kann man sich die Frage stellen, inwiefern die Verfahren an das Datenset angepasst werden müssen. Das Autoencoder Verfahren könnte in dessen Umsetzung noch weiter optimiert werden, in dem die Anzahl der Layer des Netzes angepasst und weiteres Hyperparameter-tuning vorgenommen wird. Auch beim Siamese Network Verfahren könnten die

in Kapitel 9.3.2 beschriebenen Techniken implementiert werden, um mögliche bessere Ergebnisse zu erzielen.

Weitere Techniken zur Optimierung wie ein Re-Ranking der Suchergebnisse oder die Einbeziehung von Metadaten der Produkte könnten ebenfalls zu einer Verbesserung der Ergebnisse beitragen.

## 11.4 Hypothesen

Die in dieser Arbeit zu beantwortende Hypothese lautet wie folgt:

Transfer Learning mit VGG16 und Fine-Tuning liefert auf Basis des vorhandenen Datensets bessere Ergebnisse als die anderen Verfahren mit deren Umsetzung und Setup in dieser Arbeit.

Aufgrund der Ergebnisse der Verfahren kann diese Hypothese im Rahmen dieser Arbeit bestätigt werden. Das Transfer Learning greift dabei auf ein bereits vortrainiertes Modell zurück und wurde noch zusätzlich mit dem eigenen Trainingsdaten optimiert. Die anderen Verfahren wurden ausschließlich mit einem eigenen Trainingsdatenset trainiert. Transfer Learning selbst liefert bei Mean Average Precision@5 den höchsten Wert zurück. Dies könnte jedoch bei einer Optimierung der Verfahren mithilfe der bereits genannten Techniken anders aussehen. Diese Aussage ist somit nicht generalisierbar und nur im Rahmen dieser Arbeit und diesem Setup gültig.

## 11.5 Forschungsfrage

Die Forschungsfrage innerhalb der Arbeit lautet wie folgt:

Welche ausgewählten Machine Learning Verfahren eignen sich besonders gut für den Bildvergleich im konkreten Einsatzgebiet der Suche nach ähnlichen Produkten für eine E-Commerce Lösung?

Im Rahmen dieser Arbeit wurden insgesamt drei Verfahren einer genaueren Untersuchung unterzogen. Diese lauten wie folgt:

- Autoencoder
- Siamese Network
- Transfer Learning

Anhand der Literaturrecherche wurden diese speziell für den Einsatz von Bildvergleichen beschrieben. Im Rahmen dieser Arbeit wurden diese auf einen konkreten Anwendungsfall untersucht. Diese wurden jeweils in einer Basisvariante umgesetzt. Weitere Optimierungen sind jedoch notwendig, um bessere Ergebnisse zu erzielen. Für die Basisvariante mit geringfügigen Optimierungen werden jedoch in den Suchergebnissen selbst im Schnitt mehr als 50 Prozent der Bilder bei fünf Suchergebnissen richtig erkannt siehe Mean Average Precision@5 der Verfahren. In Bezug auf die Forschungsfrage könnte somit gesagt werden, dass alle drei

Verfahren gut für Bildvergleiche im Bereich der Suche nach ähnlichen Produkten einer E-Commerce Lösung geeignet sind. Diese Aussage ist jedoch nur im Rahmen dieses Setups und dem gegebenen Datenset valide und kann schwer generalisiert werden.

## **11.6 Weitere Forschungsmöglichkeiten**

Im Rahmen dieser Arbeit wurden nur Basisvarianten der Verfahren mit geringfügigen Optimierungen umgesetzt. Eine weitere Arbeit könnte sich mit der Optimierung dieser Verfahren bezogen auf das Datenset stützen und einen weiteren Vergleich vornehmen. Im Zuge dessen könnte ein sogenanntes Re-Ranking der Suchergebnisse durchgeführt oder aber auch das Datenset noch weiter optimiert werden, um noch bessere Ergebnisse zu erhalten.

Im Bereich des Transfer Learnings wurde nur ein bereits vortrainiertes Modell mit dem Namen VGG16 verwendet. Weitere Forschungsmöglichkeiten könnten unterschiedliche Modelle aus dem Bereich evaluieren.

Deep Ranking wäre ein weiteres alternatives Verfahren, welches nicht in den Vergleich dieser Arbeit aufgenommen wurde. Weitere Untersuchungen könnten dieses Verfahren in die Praxis umsetzen und ebenso einen Vergleich mit den bereits bestehenden Verfahren durchführen.

## ANHANG A - Autoencoder Ergebnisse

Ergebnisse der einzelnen Kategorien der ersten Messung mit Autoencoder Verfahren:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Bettwasche	154	38	93,86	22,59	94,21	86,23	26,36	87,26	70,42	28,29	72,95
2	Sofas	2788	697	92,95	19,82	94,46	89,71	20,27	92,62	87,28	21,21	90,84
3	Leintucher	279	69	88,86	18,94	92,75	81,29	20,85	87,01	69,50	20,61	76,38
4	Sessel	1152	288	87,73	28,51	88,96	83,11	30,65	85,68	79,31	32,34	82,40
5	Matratzen	518	129	87,12	31,53	89,46	89,03	23,51	91,97	88,45	23,67	90,95
6	Sockelleisten	128	31	86,81	30,90	87,10	79,47	36,29	81,16	76,16	37,06	78,00
7	Betten	1246	311	86,68	31,01	88,68	83,13	29,79	86,32	80,96	31,31	84,86
8	Bademantel	128	32	83,75	36,79	83,75	63,38	32,36	70,63	48,67	30,00	57,56
9	Bilder	1120	280	80,71	29,66	82,00	67,57	34,23	72,70	62,96	35,28	68,68
10	Lattenroste	260	64	80,61	34,42	83,44	75,90	34,39	79,38	66,09	31,30	71,72
11	Eckbanke	141	35	79,56	27,84	82,29	39,44	25,70	46,06	28,18	16,91	39,20
12	Teppiche	2218	554	76,82	35,46	80,18	72,72	33,43	78,99	71,37	33,46	78,36
13	Fertigvorhänge	308	76	76,79	33,82	81,32	65,75	32,27	73,79	58,75	29,33	69,42
14	Sitzbanke	165	41	72,38	33,99	74,63	50,31	34,68	56,29	38,84	30,07	46,68
15	Stumpenkerzen	191	47	70,38	34,47	71,91	41,17	29,04	48,51	28,58	21,52	38,21
16	Tische	1466	366	66,87	39,33	69,84	54,93	37,59	62,37	48,88	35,36	57,78
17	Bratpfannen	120	29	65,77	40,45	66,90	33,24	32,03	38,90	21,92	23,58	28,48
18	Vorhangstoffe	288	71	56,26	39,47	62,25	43,71	30,71	54,48	38,52	28,24	50,70
19	Sitzsacke	157	39	54,15	42,89	56,92	39,85	41,99	43,90	27,08	30,24	31,85
20	Lampen	1197	299	52,71	41,24	57,53	39,50	35,19	47,91	33,83	31,60	43,70
21	Hocker	220	55	51,50	44,32	56,36	41,88	41,20	48,95	40,04	41,42	48,47
22	Badtextilien	542	135	51,15	45,79	52,89	33,07	36,82	37,84	26,13	31,16	32,61
23	Schraenke	1425	356	50,35	39,55	55,39	34,91	32,17	44,79	26,65	26,19	37,51
24	Fumatten	111	27	48,75	28,99	59,26	27,82	17,31	40,15	21,76	14,54	35,19
25	Teller	180	44	47,83	32,52	54,09	22,45	20,18	32,18	15,58	14,21	25,95
26	Wohnwaende	353	88	46,42	41,74	49,32	31,45	33,42	38,27	24,13	27,17	31,86
27	Kaffeetbecher	92	22	43,39	33,66	49,09	13,91	12,78	20,00	8,26	7,19	13,91
28	Kissen	153	38	39,30	30,98	49,47	25,73	17,83	40,74	19,94	13,41	34,95
29	LED-Deckenleuchten	250	62	35,67	34,16	42,90	21,38	20,29	33,35	14,84	13,89	26,74
30	Phonmobel	87	21	29,70	33,50	33,33	11,77	14,01	16,57	6,51	7,78	10,19
31	Wandboards	148	36	29,67	30,17	36,67	13,69	15,35	22,67	9,03	10,23	18,28
32	Kuscheldecken	88	21	25,71	30,52	31,43	9,66	15,10	14,29	5,34	8,51	9,14

33	Keramik-schalen	114	28	25,19	33,81	30,00	9,39	12,45	15,43	5,23	6,49	10,64
34	Kom-moden	649	162	24,66	28,32	29,51	11,59	15,13	19,98	7,86	10,71	15,95
35	Sitz-hocker	137	34	21,47	25,31	25,88	7,17	8,45	11,88	3,81	4,45	7,24
36	Baby-kleidung	192	47	21,23	38,25	22,55	12,54	20,84	19,32	7,57	12,01	13,36
37	Wand-spiegel	244	60	18,94	29,64	21,33	5,89	8,82	10,27	3,44	5,15	7,77
38	Roll-container	144	35	18,21	26,28	21,71	10,54	21,16	16,34	7,99	17,76	13,66
39	Zier-kissen	101	25	18,12	21,25	26,40	10,95	8,42	25,28	7,20	4,83	19,12
40	Diffuser	89	22	17,94	33,76	19,09	4,62	7,51	7,82	2,42	3,85	4,55
41	Tisch-wasche	240	60	17,34	20,29	22,00	4,93	5,60	9,93	2,83	3,06	7,43
42	Bade-matten	228	57	16,90	27,25	19,30	6,46	12,53	11,65	3,89	6,64	9,72
43	Wand-uhren	116	28	16,15	25,08	20,71	6,71	9,69	12,71	4,04	5,57	9,50
44	Vitrinen	204	51	13,44	19,54	19,22	6,57	11,02	14,82	4,83	6,68	14,31
45	Regale	371	92	12,54	19,84	18,04	5,79	7,61	13,87	4,14	5,06	12,46
46	Teelicht-halter	91	22	6,71	9,35	11,82	1,99	2,54	5,09	1,05	1,29	3,27
47	Nacht-kastchen	137	34	3,72	7,66	6,47	1,50	1,90	5,88	0,96	1,09	4,94
48	Laternen & Wind-lichter	84	20	0,70	2,36	2,00	0,63	1,05	4,20	0,57	0,85	4,20

Zusammengefasste Ergebnisse der ersten Messung mit Autoencoder Verfahren:

MAP@5 (Avg) [%]	64,21
MAP@5 (Std) [%]	41,24
MAP@25 (Avg) [%]	54,52
MAP@25 (Std) [%]	40,59
MAP@50 (Avg) [%]	49,97
MAP@50 (Std) [%]	40,05
P@5 (Avg) [%]	67,39
P@25 (Avg) [%]	60,46
P@50 (Avg) [%]	56,66
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,94
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,48
Abfragezeit 50 Suchergebnisse (Avg) [ms]	2,03
Trainingsdauer [s]	157,97

Ergebnisse der einzelnen Kategorien der zweiten Messung mit Autoencoder Verfahren:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Matratzen	518	129	94,92	14,82	96,28	93,34	15,38	94,98	88,29	19,31	90,22
2	Bademantel	128	32	93,75	14,76	93,75	71,11	26,21	76,63	53,68	28,66	60,19
3	Bettwasche	154	38	93,18	22,29	94,21	85,91	26,90	87,37	70,54	28,00	73,42
4	Sofas	2788	697	91,17	21,71	93,14	87,86	21,54	91,41	85,54	22,27	89,75
5	Leintucher	279	69	90,09	16,85	93,62	81,21	21,46	86,09	66,26	22,08	72,67
6	Sessel	1152	288	88,67	27,34	90,21	83,12	30,61	85,50	79,79	32,25	82,77
7	Betten	1246	311	87,73	29,99	89,00	83,72	31,09	86,35	81,05	32,51	84,35
8	Sockelleisten	128	31	86,81	30,90	87,10	79,15	36,14	81,03	74,14	38,16	76,19
9	Lattenroste	260	64	82,39	34,55	84,38	73,01	35,51	76,81	62,94	33,70	68,91
10	Bilder	1120	280	82,17	29,35	84,21	69,26	33,14	74,66	61,47	34,36	67,22
11	Fertigvorhänge	308	76	75,11	35,32	79,74	65,01	30,89	74,16	58,90	27,29	70,61
12	Teppiche	2218	554	74,43	36,78	78,56	71,32	32,53	78,36	70,23	32,73	77,81
13	Sitzbanke	165	41	73,27	34,37	75,12	50,59	33,81	56,39	39,94	29,90	47,37
14	Stumpenkerzen	191	47	72,67	33,30	75,32	48,51	31,27	56,09	36,72	26,75	45,57
15	Eckbanke	141	35	72,30	31,64	76,00	41,88	28,02	49,60	31,19	20,07	41,71
16	Bratpfannen	120	29	68,55	39,87	68,97	35,06	32,89	40,14	21,98	22,85	27,38
17	Tische	1466	366	64,41	39,80	67,87	52,77	37,62	60,30	47,26	36,06	56,17
18	Vorhangstoffe	288	71	53,02	38,06	59,15	43,70	31,75	54,03	37,51	28,89	48,25
19	Hocker	220	55	52,65	41,24	56,36	42,04	40,99	48,80	39,40	41,38	46,87
20	Lampen	1197	299	52,09	40,19	57,19	40,07	34,44	48,99	34,03	30,93	44,21
21	Schraenke	1425	356	51,07	39,74	56,74	33,54	30,81	43,09	25,13	24,83	35,76
22	Badtextilien	542	135	48,72	44,33	50,67	32,40	36,31	37,81	25,57	30,39	32,50
23	Sitzsacke	157	39	47,97	43,05	53,33	38,53	41,81	43,90	29,41	34,12	34,87
24	Wohnwaende	353	88	44,80	41,67	48,18	27,88	29,66	35,32	21,03	23,61	29,77
25	Teller	180	44	44,12	32,54	50,45	23,51	21,01	33,73	15,86	14,53	26,14
26	Fumatten	111	27	43,44	29,31	51,85	24,91	16,02	39,56	20,06	12,88	35,41
27	Kissen	153	38	38,32	30,22	47,89	23,28	17,49	37,89	16,95	12,08	31,21
28	LED-Deckenleuchten	250	62	33,46	33,48	40,32	18,85	19,39	29,68	12,43	12,54	22,94
29	Phonomobel	87	21	30,51	33,80	33,33	11,35	13,33	14,86	6,86	8,13	10,29
30	Wandboards	148	36	30,44	32,21	37,78	12,81	14,89	21,11	7,85	9,07	15,83
31	Kaffeebecher	92	22	29,18	30,44	37,27	10,03	12,19	15,64	6,35	7,49	12,27
32	Kuscheldecken	88	21	28,11	32,00	29,52	10,03	14,87	14,29	5,31	7,55	9,14
33	Kommoden	649	162	24,88	28,78	31,11	11,74	16,02	19,38	7,72	10,84	15,20
34	Babykleidung	192	47	21,36	39,61	21,70	11,68	18,97	17,87	7,09	10,45	12,72

35	Wand-uhren	116	28	20,68	27,76	26,43	7,42	9,30	13,57	4,44	5,37	9,86
36	Wand-spiegel	244	60	20,42	29,54	22,67	7,20	11,23	11,27	4,24	6,63	8,57
37	Zier-kissen	101	25	19,23	16,05	27,20	9,78	6,91	21,92	7,12	4,70	19,04
38	Keramik-schalen	114	28	18,26	29,30	20,71	7,70	11,35	14,00	4,62	6,49	10,50
39	Diffuser	89	22	17,58	33,84	18,18	4,16	7,27	6,73	2,19	3,61	4,27
40	Bade-matten	228	57	16,22	27,66	18,95	6,36	13,56	11,44	3,91	7,47	9,40
41	Tisch-wasche	240	60	16,08	19,77	21,00	4,63	5,36	9,33	2,68	3,11	7,17
42	Sitz-hocker	137	34	14,72	20,63	21,18	5,90	7,52	12,82	3,41	4,42	8,76
43	Vitrinen	204	51	14,21	22,64	20,78	7,84	10,96	16,94	5,72	7,14	15,80
44	Roll-container	144	35	13,96	21,69	18,86	8,00	18,36	13,83	6,07	14,93	11,66
45	Regale	371	92	12,61	19,87	17,61	6,17	8,97	14,35	4,09	5,50	11,67
46	Teelicht-halter	91	22	5,68	9,42	8,18	1,93	2,70	4,55	1,03	1,34	3,27
47	Nacht-kastchen	137	34	4,37	7,02	8,82	1,29	1,73	5,29	0,86	0,96	4,59
48	Laternen & Wind-lichter	84	20	2,12	4,81	5,00	1,21	2,51	4,60	0,99	1,98	5,00

Zusammengefasste Ergebnisse der zweiten Messung mit Autoencoder Verfahren:

MAP@5 (Avg) [%]	63,64
MAP@5 (Std) [%]	41,18
MAP@25 (Avg) [%]	54,06
MAP@25 (Std) [%]	40,35
MAP@50 (Avg) [%]	49,24
MAP@50 (Std) [%]	39,78
P@5 (Avg) [%]	67,10
P@25 (Avg) [%]	60,11
P@50 (Avg) [%]	55,97
Abfragezeit 5 Suchergebnisse (Avg) [ms]	1,06
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,62
Abfragezeit 50 Suchergebnisse (Avg) [ms]	2,47
Trainingsdauer [s]	159,39



Ergebnisse der einzelnen Kategorien der dritten Messung mit Autoencoder Verfahren:

R a n g	Kategorie	Trainings bilder (20814)	Testbil der (5178)	MAP@5 (Avg) [%]	MAP@ 5 (Std) [%]	P@5 (Avg) [%]	MAP@ 25 (Avg) [%]	MAP@ 25 (Std) [%]	P@25 (Avg) [%]	MAP@ 50 (Avg) [%]	MAP@ 50 (Std) [%]	P@50 (Avg) [%]
1	Bett- wasche	154	38	94,76	19,93	95,26	18,85	92,21	21,95	92,84	82,49	24,68
2	Lein- tucher	279	69	93,58	15,97	95,65	11,31	86,72	19,91	90,38	75,94	22,38
3	Sofas	2788	697	90,63	22,75	92,74	18,65	87,80	21,61	91,55	86,14	22,13
4	Betten	1246	311	89,82	26,71	91,19	24,80	86,50	27,51	88,89	83,81	29,17
5	Matratzen	518	129	89,60	26,20	90,85	22,91	90,11	24,89	92,28	89,77	24,71
6	Sessel	1152	288	89,47	27,02	90,83	24,81	84,45	29,85	86,67	81,08	31,60
7	Bade- mantel	128	32	88,67	19,61	90,63	16,05	67,86	26,09	76,00	53,70	28,22
8	Sockel- leisten	128	31	87,74	30,84	87,74	30,84	83,29	34,14	84,00	79,59	35,98
9	Bilder	1120	280	84,27	28,90	86,36	26,03	74,42	30,06	78,51	70,95	31,89
10	Eckbanke	141	35	81,49	29,87	82,86	28,34	48,27	25,59	55,43	34,15	20,60
11	Fertig- vorhänge	308	76	78,56	34,32	81,84	30,84	66,96	33,17	73,68	60,09	31,48
12	Teppiche	2218	554	78,05	34,88	81,66	32,01	73,10	33,87	79,31	71,06	34,27
13	Latten- roste	260	64	77,06	40,05	77,19	39,98	68,12	41,23	71,75	63,10	37,13
14	Stumpen- kerzen	191	47	75,44	34,40	77,45	32,20	47,64	30,39	54,81	34,36	23,98
15	Sitzbanke	165	41	74,72	34,07	76,10	33,23	55,72	35,36	61,95	43,78	32,16
16	Tische	1466	366	68,26	38,11	71,42	36,11	57,26	38,33	63,83	50,67	36,38
17	Vorhang- stoffe	288	71	61,43	38,96	67,89	35,85	50,74	29,73	61,86	45,54	27,33
18	Sitzsacke	157	39	59,81	44,53	62,05	43,24	42,12	40,99	46,56	32,63	35,42
19	Brat- pfannen	120	29	58,16	38,13	59,31	37,22	33,79	36,40	38,21	22,15	25,87
20	Schraen- ke	1425	356	55,71	39,45	60,90	36,67	37,56	32,57	46,91	29,01	27,51
21	Kaffee- becher	92	22	54,48	35,96	56,36	35,26	21,06	16,17	27,27	12,50	9,70
22	Hocker	220	55	51,56	43,20	53,82	41,79	39,68	40,75	45,16	35,34	40,36
23	Lampen	1197	299	51,47	40,72	55,85	39,24	38,62	34,60	47,09	32,06	30,59
24	Fumatten	111	27	51,27	33,97	60,00	31,38	29,08	18,29	41,78	23,04	15,03
25	Bad- textilien	542	135	49,11	45,47	50,52	44,71	32,73	37,26	37,24	26,23	32,36
26	Teller	180	44	45,00	33,08	51,82	32,37	23,59	19,29	35,09	16,45	13,94
27	Wohn- waende	353	88	44,14	40,87	47,95	40,18	25,64	26,66	32,91	17,94	19,44
28	Kissen	153	38	42,24	29,11	51,58	28,14	25,96	17,24	41,68	20,76	13,51
29	LED- Decken- leuchten	250	62	34,81	34,33	41,61	34,03	18,77	19,45	29,81	12,63	12,52
30	Phono- mobel	87	21	33,05	35,54	34,29	36,96	14,96	17,26	18,67	8,26	9,54
31	Kuschel- decken	88	21	28,79	30,88	31,43	30,71	10,62	15,28	15,24	5,65	7,70
32	Wand- boards	148	36	28,37	34,42	32,78	35,83	12,73	15,34	20,56	8,14	9,48
33	Kom- moden	649	162	27,97	31,42	33,83	32,32	14,00	17,63	22,25	9,40	12,03
34	Baby- kleidung	192	47	27,66	45,22	27,66	45,22	13,68	21,29	18,89	8,65	13,32

35	Keramik-schalen	114	28	26,74	34,47	32,86	34,95	10,72	12,79	19,43	6,32	6,98
36	Sitz-hocker	137	34	21,74	27,79	26,47	30,24	7,97	9,03	14,82	4,55	5,27
37	Diffuser	89	22	20,98	33,02	25,45	32,77	5,27	8,01	7,45	2,70	4,12
38	Tisch-wasche	240	60	18,38	21,00	23,67	24,84	6,00	6,73	12,07	3,46	3,84
39	Wand-uhren	116	28	18,38	28,03	21,43	29,78	6,94	9,79	13,29	4,11	5,62
40	Roll-container	144	35	18,02	26,31	22,86	27,50	10,53	21,37	16,23	8,19	19,41
41	Bade-matten	228	57	17,71	29,77	21,40	30,91	7,06	14,09	12,14	4,40	8,79
42	Zier-kissen	101	25	15,96	13,42	28,00	21,60	10,56	7,84	26,08	8,03	5,64
43	Wand-spiegel	244	60	15,88	26,19	18,33	26,88	4,54	7,31	8,13	2,57	3,99
44	Vitrinen	204	51	13,07	21,45	19,22	24,64	6,43	9,54	15,14	4,58	5,86
45	Regale	371	92	11,17	16,87	16,09	20,38	4,44	5,60	11,83	3,22	3,92
46	Teelicht-halter	91	22	7,95	10,29	10,91	13,42	2,16	2,53	5,45	1,11	1,28
47	Nacht-kastchen	137	34	4,97	9,55	8,82	14,09	1,64	2,13	5,88	1,01	1,28
48	Laternen & Wind-lichter	84	20	1,40	4,55	3,00	7,33	0,76	1,66	3,80	0,70	1,26

Zusammengefasste Ergebnisse der dritten Messung mit Autoencoder Verfahren:

MAP@5 (Avg) [%]	65,33
MAP@5 (Std) [%]	40,99
MAP@25 (Avg) [%]	55,62
MAP@25 (Std) [%]	40,57
MAP@50 (Avg) [%]	51,11
MAP@50 (Std) [%]	40,34
P@5 (Avg) [%]	68,51
P@25 (Avg) [%]	61,37
P@50 (Avg) [%]	57,59
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,93
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,55
Abfragezeit 50 Suchergebnisse (Avg) [ms]	2,46
Trainingsdauer [s]	162,80

## ANHANG B - Siamese Network Ergebnisse

Ergebnisse der einzelnen Kategorien der ersten Messung mit Siamese Network Verfahren:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Leintücher	279	69	100,00	0,00	100,00	95,18	13,88	95,77	91,75	17,37	93,33
2	Bademantel	128	32	100,00	0,00	100,00	99,18	1,97	99,38	98,46	3,15	98,94
3	Bettwasche	154	38	97,11	14,87	97,37	95,54	19,08	95,89	95,20	20,36	95,68
4	Lattenroste	260	64	93,75	24,40	93,75	89,68	26,45	90,63	84,31	28,64	85,97
5	Matratzen	518	129	92,25	20,66	92,25	87,22	19,96	91,66	81,00	22,03	85,40
6	Sessel	1152	288	89,67	25,25	90,90	84,31	29,28	86,96	81,96	31,60	85,04
7	Betten	1246	311	88,75	29,16	89,84	87,75	29,76	89,17	87,03	30,52	88,59
8	Sockelleisten	128	31	88,39	30,89	88,39	84,74	33,69	85,03	83,36	34,96	84,26
9	Sofas	2788	697	86,24	27,46	88,55	80,78	29,51	85,33	79,58	29,52	85,38
10	Kaffeetbecher	92	22	85,30	30,28	86,36	77,64	37,59	79,45	62,65	32,39	66,27
11	Bilder	1120	280	75,91	36,16	77,64	64,68	40,90	68,03	59,34	39,40	63,42
12	Bratpfannen	120	29	75,84	37,71	77,93	59,08	38,35	63,59	46,59	35,90	52,14
13	Stumpfenkerzen	191	47	75,35	31,20	77,87	39,64	25,67	46,72	24,98	15,77	34,43
14	Eckbanke	141	35	72,12	32,62	74,86	62,87	35,17	70,74	58,95	31,53	69,14
15	Sitzbanke	165	41	71,28	37,55	75,12	53,37	36,35	60,88	44,04	34,49	52,34
16	Tische	1466	366	71,27	38,81	74,10	62,88	38,58	68,61	60,59	38,54	67,51
17	Fertigvorhänge	308	76	71,18	34,73	75,26	61,78	35,74	69,95	57,55	34,01	67,47
18	Teppiche	2218	554	70,38	36,64	75,92	67,95	33,07	76,20	65,45	32,94	73,94
19	Hocker	220	55	66,23	39,10	70,55	58,98	41,48	65,82	59,08	42,09	66,47
20	Lampen	1197	299	56,95	41,01	61,87	46,37	37,37	54,56	42,51	36,31	51,85
21	Badtextilien	542	135	55,18	46,67	56,00	44,28	42,86	47,23	37,98	39,99	41,73
22	Teller	180	44	51,07	33,49	56,36	28,60	19,78	40,18	22,67	17,40	34,91
23	Wohnwände	353	88	48,30	41,00	54,32	35,01	33,51	43,00	25,66	24,92	35,09
24	Fumatten	111	27	47,99	35,73	55,56	29,01	21,74	38,96	22,73	17,61	34,81
25	Sitzsacke	157	39	46,03	46,74	47,18	42,55	45,35	45,85	39,64	43,24	43,18
26	Schraenke	1425	356	44,95	38,40	51,18	28,94	29,22	39,24	23,58	26,77	34,40
27	Babykleidung	192	47	42,94	46,99	44,26	28,81	32,03	34,72	18,98	20,58	25,19
28	Phonomobil	87	21	41,52	39,16	43,81	27,72	28,67	33,71	17,61	19,07	23,43
29	Keramikschalen	114	28	39,14	29,90	42,86	14,99	10,51	25,71	9,51	6,51	19,07
30	Kissen	153	38	38,53	26,44	48,95	26,25	18,14	41,05	23,04	17,66	38,74
31	Wandboards	148	36	36,30	37,70	43,89	25,73	26,25	38,22	20,27	21,13	32,72
32	Diffuser	89	22	35,50	36,86	38,18	10,10	9,36	14,18	5,21	4,72	8,36

## Siamese Network Ergebnisse

33	Kommoden	649	162	26,70	29,19	33,21	13,61	16,99	22,20	9,61	12,73	18,38
34	Sitzhocker	137	34	25,88	25,18	28,24	6,96	7,50	10,82	4,51	5,17	9,41
35	Vorhangstoffe	288	71	24,56	28,74	31,27	17,99	26,32	26,87	16,73	25,53	26,39
36	Rollcontainer	144	35	23,79	34,09	26,29	14,31	31,56	16,00	11,96	29,17	13,60
37	Kuscheldecken	88	21	22,06	28,48	26,67	6,64	8,34	12,76	4,09	4,58	10,10
38	Badematten	228	57	21,63	32,68	24,21	12,88	23,41	20,21	8,56	16,79	15,19
39	LED-Deckenleuchten	250	62	21,45	26,18	28,39	8,94	10,55	17,16	6,07	7,04	14,42
40	Tischwasche	240	60	19,13	26,71	23,67	6,32	8,55	11,93	3,69	4,82	9,03
41	Regale	371	92	17,76	25,12	22,61	9,53	13,47	18,61	7,62	11,09	17,02
42	Wanduhren	116	28	17,08	26,05	20,71	5,69	8,85	9,71	3,23	4,98	6,86
43	Wandspiegel	244	60	14,07	27,79	17,00	5,28	10,34	8,93	3,07	5,84	6,70
44	Zierkissen	101	25	13,96	14,29	22,40	8,62	6,64	23,36	7,79	5,37	23,20
45	Vitrinen	204	51	12,67	21,38	20,39	6,73	7,83	17,73	5,31	5,41	16,59
46	Laternen & Windlichter	84	20	11,77	19,40	16,00	3,68	5,86	8,20	2,22	3,34	6,50
47	Nachtkastchen	137	34	5,96	12,59	10,59	2,19	4,64	6,35	1,39	2,73	5,53
48	Teelichthalter	91	22	3,11	6,18	6,36	1,34	1,87	5,45	0,79	1,03	3,82

Zusammengefasste Ergebnisse der ersten Messung mit Siamese Network Verfahren:

MAP@5 (Avg) [%]	63,81
MAP@5 (Std) [%]	41,16
MAP@25 (Avg) [%]	55,39
MAP@25 (Std) [%]	41,10
MAP@50 (Avg) [%]	51,96
MAP@50 (Std) [%]	40,97
P@5 (Avg) [%]	67,28
P@25 (Avg) [%]	61,32
P@50 (Avg) [%]	58,54
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,91
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,50
Abfragezeit 50 Suchergebnisse (Avg) [ms]	2,02
Trainingsdauer [s]	394,05

Ergebnisse der einzelnen Kategorien der zweiten Messung mit Siamese Network Verfahren:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Bettwasche	154	38	96,11	17,80	96,32	95,12	21,06	95,26	94,75	21,76	94,89
2	Leintucher	279	69	95,83	19,72	96,52	94,04	18,99	95,71	91,32	22,46	92,70
3	Bademantel	128	32	93,75	24,59	93,75	83,26	21,86	87,88	82,48	16,27	89,44
4	Matratzen	518	129	89,30	30,52	89,30	88,78	20,40	92,43	86,08	21,66	90,67
5	Sofas	2788	697	89,27	24,01	91,54	85,78	25,02	90,05	84,46	24,37	89,57
6	Sessel	1152	288	89,24	27,38	90,21	84,97	29,55	86,93	82,15	32,39	84,23
7	Kaffeebecher	92	22	88,38	22,16	89,09	74,09	33,36	79,45	67,26	37,02	72,45
8	Lattenroste	260	64	87,50	31,37	87,50	74,09	32,49	77,94	54,34	25,92	60,78
9	Sockelleisten	128	31	86,73	31,85	87,74	84,99	31,78	86,97	83,44	33,81	85,48
10	Bilder	1120	280	84,91	28,30	87,14	82,55	30,75	86,06	79,07	31,55	83,56
11	Betten	1246	311	84,89	33,17	85,98	81,22	35,06	82,80	79,84	35,78	81,98
12	Eckbanke	141	35	74,59	33,36	78,86	60,53	34,40	68,34	50,82	27,61	61,03
13	Stumpenkerzen	191	47	72,86	33,78	75,74	40,71	26,29	49,02	27,55	22,28	36,09
14	Teppiche	2218	554	72,72	38,09	77,51	68,03	33,73	75,14	66,88	33,15	75,55
15	Tische	1466	366	71,54	37,20	74,48	60,84	37,63	67,05	56,94	38,10	64,10
16	Sitzbanke	165	41	69,56	33,69	75,61	61,83	31,99	69,95	47,66	29,13	55,85
17	Fertigvorhänge	308	76	69,18	38,90	73,68	61,42	34,21	70,68	56,19	31,35	67,32
18	Hocker	220	55	61,88	38,97	66,55	55,40	41,35	62,04	52,46	42,81	58,58
19	Bratpfannen	120	29	58,29	44,59	60,69	46,64	42,45	52,41	35,54	34,49	42,48
20	Lampen	1197	299	56,27	41,33	60,80	42,43	35,18	50,68	36,40	32,19	46,27
21	Badtextilien	542	135	55,03	46,54	55,56	46,27	45,61	48,00	40,93	43,20	43,56
22	Sitzsacke	157	39	54,56	47,69	56,41	50,33	47,14	52,82	48,44	47,40	51,13
23	Keramikschalen	114	28	53,43	38,85	57,14	23,64	17,57	32,43	14,37	10,60	22,79
24	Schraenke	1425	356	47,42	38,45	53,82	29,96	28,36	40,38	24,00	25,81	35,70
25	Teller	180	44	47,16	29,91	53,64	33,32	24,86	44,18	27,37	21,70	39,18
26	Wohnwaende	353	88	43,89	39,01	49,55	28,46	29,99	37,73	23,11	25,56	33,75
27	Fumatten	111	27	38,28	34,42	45,19	30,71	24,93	43,11	25,36	21,39	37,85
28	Vorhangstoffe	288	71	37,58	34,01	46,76	26,39	25,82	38,76	23,51	23,51	37,38
29	Kissen	153	38	37,26	28,11	47,37	24,52	19,75	39,16	20,19	16,97	34,89
30	Babykleidung	192	47	37,13	45,20	37,87	30,78	40,38	36,34	25,08	32,56	32,09
31	Wandboards	148	36	36,26	39,06	40,56	21,62	25,45	29,89	14,75	18,98	22,61
32	Phonomobel	87	21	35,79	38,13	40,95	18,25	19,84	25,71	10,83	11,63	17,14
33	Wanduhren	116	28	27,60	37,93	30,71	17,43	26,60	22,57	11,80	17,95	17,29
34	Kuscheldecken	88	21	26,83	24,22	33,33	8,95	7,01	16,76	5,82	5,09	12,76
35	Kommoden	649	162	26,82	30,73	33,33	13,50	16,21	23,26	9,72	11,96	19,65

## Siamese Network Ergebnisse

36	LED-Deckenleuchten	250	62	26,09	31,66	33,23	13,47	15,82	23,42	9,30	10,11	19,35
37	Roll-container	144	35	24,31	33,15	27,43	13,75	26,46	17,14	11,93	26,27	15,20
38	Badematten	228	57	23,75	34,48	27,02	11,57	22,30	16,21	8,14	16,72	13,93
39	Diffuser	89	22	22,48	33,13	24,55	6,05	7,16	11,82	3,11	3,55	6,64
40	Tischwasche	240	60	19,02	27,13	24,33	7,60	10,43	14,60	4,31	5,57	9,80
41	Sitzhocker	137	34	16,65	26,66	20,00	4,81	7,06	9,18	2,69	3,68	6,82
42	Regale	371	92	16,64	25,81	21,52	8,74	14,87	17,13	6,07	10,26	14,26
43	Vitrinen	204	51	13,79	19,13	21,96	6,90	7,56	16,55	5,19	5,46	15,53
44	Wandspiegel	244	60	12,54	22,36	15,00	4,75	7,79	10,33	3,65	5,71	9,67
45	Zierkissen	101	25	11,08	19,69	16,80	7,34	8,58	19,68	6,30	5,65	20,24
46	Laternen & Windlichter	84	20	8,47	17,16	10,00	2,78	4,53	4,80	1,62	2,52	4,60
47	Teelichthalter	91	22	7,45	10,22	10,00	2,04	2,41	5,09	1,09	1,22	3,55
48	Nachtkastchen	137	34	4,59	13,59	6,47	1,50	3,29	4,59	0,86	1,76	3,47

Zusammengefasste Ergebnisse der zweiten Messung mit Siamese Network Verfahren:

MAP@5 (Avg) [%]	64,48
MAP@5 (Std) [%]	41,37
MAP@25 (Avg) [%]	56,25
MAP@25 (Std) [%]	40,84
MAP@50 (Avg) [%]	52,63
MAP@50 (Std) [%]	40,74
P@5 (Avg) [%]	67,93
P@25 (Avg) [%]	62,21
P@50 (Avg) [%]	59,41
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,91
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,37
Abfragezeit 50 Suchergebnisse (Avg) [ms]	2,06
Trainingsdauer [s]	393,35

Ergebnisse der einzelnen Kategorien der dritten Messung mit Siamese Network Verfahren:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Bademantel	128	32	100,00	0,00	100,00	96,11	12,94	96,75	93,58	16,25	95,19
2	Bettwasche	154	38	95,96	17,78	96,32	94,97	21,63	95,05	94,86	22,12	94,89
3	Leintucher	279	69	94,85	20,95	95,07	92,98	20,59	94,67	90,93	21,71	92,90
4	Sessel	1152	288	90,87	25,37	91,81	86,55	26,63	88,88	83,62	28,55	86,51
5	Betten	1246	311	89,86	27,47	90,42	85,20	30,88	86,68	83,40	31,96	85,49
6	Lattenroste	260	64	89,06	28,27	89,06	86,04	30,98	87,94	77,67	29,60	80,50
7	Sockelleisten	128	31	88,39	29,11	88,39	82,00	35,00	83,48	81,85	35,70	83,87
8	Kaffebecher	92	22	87,53	28,85	89,09	79,46	33,86	82,00	70,26	31,92	74,18
9	Sofas	2788	697	86,84	28,87	88,41	82,77	29,62	86,66	81,56	29,20	86,51
10	Sitzbanke	165	41	82,91	33,87	84,39	76,39	32,17	81,95	66,80	31,98	72,34
11	Matratzen	518	129	82,77	35,30	84,19	82,24	30,15	87,10	84,35	25,89	90,64
12	Eckbanke	141	35	77,86	29,40	81,14	67,63	29,39	75,66	60,03	25,73	69,49
13	Bilder	1120	280	75,83	35,03	77,50	65,55	39,75	69,31	56,47	36,86	61,56
14	Stumpenkerzen	191	47	72,78	32,18	75,74	39,50	26,35	47,83	27,36	20,89	36,89
15	Tische	1466	366	72,39	38,01	74,75	59,53	38,48	64,84	54,46	38,51	61,32
16	Teppiche	2218	554	70,97	37,75	76,71	69,32	34,06	77,26	67,80	34,22	75,84
17	Fertigvorhänge	308	76	69,45	38,47	72,63	61,36	37,01	69,26	58,24	35,65	67,66
18	Bratpfannen	120	29	61,74	35,90	62,76	42,85	35,55	52,55	35,32	32,60	45,45
19	Lampen	1197	299	57,14	41,69	61,40	46,70	37,95	54,69	42,75	36,78	51,85
20	Hocker	220	55	56,36	41,45	62,18	50,80	42,03	58,47	48,86	42,39	57,93
21	Badtextilien	542	135	55,65	46,49	56,15	42,67	43,47	44,39	36,51	41,34	39,32
22	Wohnwände	353	88	51,95	42,97	54,55	38,82	37,74	45,68	31,20	31,77	39,25
23	Sitzsacke	157	39	46,87	49,18	47,69	42,57	47,30	45,64	40,80	46,96	44,10
24	Keramikschalen	114	28	43,29	34,15	47,14	17,16	12,89	26,29	10,85	8,09	20,07
25	Schraenke	1425	356	43,23	38,01	50,11	29,35	28,48	40,84	23,14	24,38	35,38
26	Kissen	153	38	41,32	26,59	52,11	28,53	16,45	45,05	24,70	14,82	42,32
27	Teller	180	44	41,09	30,12	48,64	25,61	20,61	36,45	21,83	19,78	33,77
28	Fumatten	111	27	36,11	38,78	39,26	22,12	20,90	33,48	18,51	18,47	31,33
29	Diffuser	89	22	34,80	35,84	42,73	9,68	8,27	14,18	5,09	4,14	8,82
30	Vorhangstoffe	288	71	33,19	33,45	39,44	26,21	30,32	37,13	22,78	28,22	33,61
31	Wandboards	148	36	33,06	32,65	41,11	19,69	18,55	32,33	14,23	13,37	26,56
32	Kuscheldecken	88	21	30,57	29,63	34,29	11,68	10,10	19,24	7,04	5,73	14,48
33	Phonomobel	87	21	27,83	28,80	32,38	14,80	20,07	20,95	8,83	12,11	14,38
34	LED-Deckenleuchten	250	62	27,82	30,74	34,52	12,52	14,27	21,81	8,43	9,05	18,23
35	Wand-	116	28	24,18	32,14	29,29	11,26	17,23	17,71	6,77	10,20	12,36

	uhren											
36	Zierkissen	101	25	24,08	23,84	32,80	11,81	9,07	24,64	8,05	5,61	20,32
37	Badematten	228	57	23,37	35,35	26,32	10,40	20,36	15,23	6,88	13,60	13,12
38	Sitzhocker	137	34	21,88	26,13	22,35	6,22	7,98	9,41	3,89	5,21	8,06
39	Kommoden	649	162	21,57	27,63	27,41	12,09	16,54	21,26	8,65	12,23	17,80
40	Babykleidung	192	47	19,98	37,10	22,55	13,99	22,86	20,51	10,14	14,35	18,77
41	Wandspiegel	244	60	19,32	30,50	21,67	7,98	13,87	12,27	4,40	7,58	8,20
42	Laternen & Windlichter	84	20	19,32	34,21	22,00	6,37	9,92	12,00	3,63	5,50	8,30
43	Regale	371	92	15,90	21,41	22,39	8,23	10,35	17,26	6,36	8,04	15,61
44	Tischwasche	240	60	15,74	25,00	20,67	5,62	8,81	11,20	3,20	4,86	8,30
45	Vitrinen	204	51	14,99	23,94	20,39	7,60	9,71	17,25	5,41	6,04	15,41
46	Rollcontainer	144	35	13,05	25,73	17,71	9,46	24,27	13,83	9,19	23,94	14,51
47	Teelichthalter	91	22	10,45	18,89	12,73	2,36	3,90	4,55	1,31	2,07	3,55
48	Nachtkastchen	137	34	5,42	14,97	9,41	1,85	3,87	5,06	1,09	2,02	4,29

Zusammengefasste Ergebnisse der dritten Messung mit Siamese Network Verfahren:

MAP@5 (Avg) [%]	63,43
MAP@5 (Std) [%]	41,73
MAP@25 (Avg) [%]	55,32
MAP@25 (Std) [%]	41,38
MAP@50 (Avg) [%]	51,74
MAP@50 (Std) [%]	41,11
P@5 (Avg) [%]	66,71
P@25 (Avg) [%]	61,25
P@50 (Avg) [%]	58,42
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,99
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,43
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,89
Trainingsdauer [s]	402,03



## ANHANG C - Transfer Learning ohne Fine-Tuning Ergebnisse

Ergebnisse der einzelnen Kategorien der ersten Messung mit Transfer Learning und vortrainiertem Modell ohne Fine-Tuning:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Sofas	2788	697	92,24	21,56	93,26	87,60	23,40	90,79	85,06	24,24	89,19
2	Stumpfenkerzen	191	47	89,89	22,47	91,91	70,98	27,83	75,91	56,89	27,99	63,66
3	Sockelleisten	128	31	89,29	28,54	89,68	79,28	34,52	80,39	75,11	37,51	76,97
4	Bilder	1120	280	89,16	24,12	89,86	78,89	31,11	82,06	69,11	30,43	73,99
5	Bademantel	128	32	86,97	19,23	91,88	69,83	25,11	76,38	53,22	28,60	60,06
6	Sessel	1152	288	84,16	30,51	85,83	73,51	34,01	78,40	68,65	35,33	74,79
7	Leintucher	279	69	83,61	27,13	85,80	65,95	23,17	74,20	52,28	18,44	62,67
8	Bratpfannen	120	29	83,11	32,97	85,52	74,85	33,01	79,72	65,37	31,68	71,52
9	Betten	1246	311	80,43	34,53	83,09	74,09	33,37	79,56	69,20	33,96	76,11
10	Matratzen	518	129	78,85	32,08	82,64	72,32	29,85	81,05	64,67	26,34	75,12
11	Bettwäsche	154	38	73,44	35,09	75,79	53,45	34,02	59,58	35,69	23,63	43,53
12	Lattenroste	260	64	73,20	36,66	77,19	57,67	33,70	66,88	47,53	31,23	57,41
13	Teppiche	2218	554	63,39	36,69	72,02	60,57	33,37	71,13	56,52	32,48	67,60
14	Tische	1466	366	63,09	37,86	68,09	53,73	37,37	62,15	48,81	35,99	58,51
15	Fertigvorhänge	308	76	62,80	38,22	68,68	55,91	35,12	65,58	50,68	32,35	61,89
16	Sitzsacke	157	39	59,29	43,61	62,56	48,57	41,31	54,36	41,40	38,64	47,90
17	Wohnwände	353	88	56,71	40,82	60,91	45,21	35,46	54,23	38,40	30,87	49,23
18	Schraenke	1425	356	55,88	38,05	61,97	41,32	31,65	53,03	34,35	28,40	47,53
19	Lampen	1197	299	53,56	41,00	58,66	39,63	34,08	48,64	33,42	29,99	44,61
20	Eckbanke	141	35	51,67	36,41	56,00	25,63	24,55	33,60	16,10	15,03	24,97
21	Teller	180	44	47,71	30,52	54,09	31,71	21,64	46,73	27,04	18,73	43,77
22	Sitzbanke	165	41	44,02	42,23	45,85	29,94	34,60	36,49	22,06	26,37	30,20
23	Kaffeebecher	92	22	43,92	35,01	50,91	16,87	15,53	23,45	9,49	8,60	15,55
24	Badtextilien	542	135	42,67	40,58	45,78	26,65	29,18	34,90	22,13	24,43	32,65
25	Hocker	220	55	41,10	39,71	46,18	34,14	38,65	41,38	30,28	36,28	37,96
26	Kissen	153	38	38,84	32,00	47,89	24,85	24,92	34,00	17,38	18,50	25,89
27	Vorhangstoffe	288	71	35,22	31,81	43,94	26,74	27,55	38,08	22,33	25,37	33,52
28	Wandboards	148	36	34,14	34,42	40,56	16,14	19,03	25,22	10,20	11,99	19,11
29	Keramikschalen	114	28	33,69	41,21	35,71	18,65	27,15	25,86	11,38	16,89	18,43
30	LED-Deckenleuchten	250	62	32,61	35,02	38,71	18,58	21,99	28,77	13,09	15,85	23,39
31	Babykleidung	192	47	27,79	43,47	30,64	21,70	32,23	29,87	15,62	22,98	23,28

32	Wand-uhren	116	28	25,31	37,71	30,00	13,96	19,17	20,86	9,41	12,84	16,14
33	Phono-mobel	87	21	25,11	23,73	32,38	9,45	10,35	17,33	5,69	6,03	12,38
34	Kom-moden	649	162	24,70	29,96	31,11	12,52	16,40	21,88	8,78	11,68	18,22
35	Bade-matten	228	57	23,18	32,88	28,77	9,38	12,22	16,84	5,95	7,11	13,65
36	Sitz-hocker	137	34	21,14	24,44	24,71	8,01	10,07	13,53	4,95	6,16	10,24
37	Wand-spiegel	244	60	21,01	26,95	24,67	7,83	10,62	14,27	5,08	6,76	12,07
38	Roll-container	144	35	20,41	33,36	22,29	15,14	28,23	18,97	10,97	22,07	14,63
39	Fumatten	111	27	20,17	28,34	25,93	14,97	20,72	27,11	13,04	18,95	25,04
40	Kuschel-decken	88	21	18,51	22,67	27,62	7,44	8,15	16,57	4,81	4,90	12,95
41	Diffuser	89	22	18,39	29,91	20,91	5,15	8,71	7,64	2,95	5,27	5,82
42	Zier-kissen	101	25	17,37	16,64	25,60	7,82	6,61	19,04	5,35	4,28	15,20
43	Vitrinen	204	51	16,79	22,64	24,71	10,57	13,15	22,20	7,34	8,95	18,27
44	Regale	371	92	15,78	25,10	20,00	9,29	15,25	17,30	7,00	11,81	15,11
45	Tisch-wasche	240	60	13,17	23,64	17,33	4,86	7,21	11,00	3,09	4,24	9,17
46	Teelicht-halter	91	22	11,62	17,45	15,45	3,01	3,61	6,73	1,62	1,88	4,64
47	Nacht-kastchen	137	34	6,61	12,80	10,59	2,50	4,40	7,18	1,46	2,45	5,29
48	Laternen & Wind-lichter	84	20	4,55	10,46	7,00	1,80	3,40	5,80	0,99	1,85	3,60

Zusammengefasste Ergebnisse der ersten Messung mit Transfer Learning und vortrainiertem Modell ohne Fine-Tuning:

MAP@5 (Avg) [%]	61,64
MAP@5 (Std) [%]	40,77
MAP@25 (Avg) [%]	52,09
MAP@25 (Std) [%]	39,17
MAP@50 (Avg) [%]	46,75
MAP@50 (Std) [%]	37,76
P@5 (Avg) [%]	65,86
P@25 (Avg) [%]	59,50
P@50 (Avg) [%]	55,14
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,66
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,30
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,88
Trainingsdauer [s]	0,00

Ergebnisse der einzelnen Kategorien der zweiten Messung mit Transfer Learning und vortrainiertem Modell ohne Fine-Tuning:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Sofas	2788	697	92,34	21,35	93,37	87,57	23,45	90,78	85,04	24,28	89,17
2	Stumpfenkerzen	191	47	90,55	21,15	92,77	70,88	27,94	75,74	56,79	28,01	63,53
3	Sockelleisten	128	31	89,29	28,54	89,68	79,28	34,52	80,39	75,11	37,51	76,97
4	Bilder	1120	280	89,16	24,12	89,86	78,88	31,12	82,04	69,11	30,43	73,99
5	Bademantel	128	32	86,97	19,23	91,88	69,83	25,11	76,38	53,27	28,65	60,06
6	Sessel	1152	288	84,13	30,33	85,90	73,51	34,06	78,31	68,67	35,29	74,82
7	Leintuch	279	69	83,61	27,13	85,80	65,95	23,17	74,20	52,28	18,44	62,67
8	Bratpfannen	120	29	83,11	32,97	85,52	74,84	33,01	79,72	65,37	31,68	71,52
9	Betten	1246	311	80,73	34,48	83,34	74,17	33,33	79,65	69,35	33,95	76,27
10	Matratzen	518	129	78,85	32,08	82,64	72,32	29,85	81,05	64,67	26,34	75,12
11	Lattenroste	260	64	73,20	36,66	77,19	57,67	33,70	66,88	47,53	31,23	57,41
12	Bettwasche	154	38	73,12	35,34	75,26	53,35	34,06	59,37	35,65	23,59	43,47
13	Teppiche	2218	554	63,61	36,68	72,02	60,52	33,45	70,97	56,55	32,46	67,61
14	Tische	1466	366	63,03	37,94	68,03	53,66	37,33	62,11	48,83	36,05	58,54
15	Fertigvorhänge	308	76	62,80	38,22	68,68	55,91	35,12	65,58	50,68	32,35	61,89
16	Sitzsacke	157	39	59,03	43,94	62,05	48,92	41,47	54,56	41,47	38,65	47,90
17	Wohnwände	353	88	56,71	40,82	60,91	45,22	35,45	54,23	38,41	30,87	49,23
18	Schraenke	1425	356	55,82	38,09	61,80	41,28	31,62	52,99	34,34	28,41	47,49
19	Lampen	1197	299	53,70	40,92	58,80	39,63	34,07	48,66	33,42	29,99	44,60
20	Eckbanke	141	35	52,00	35,93	56,57	25,63	24,55	33,60	16,10	15,03	24,97
21	Teller	180	44	47,71	30,52	54,09	31,71	21,64	46,73	27,02	18,74	43,73
22	Kaffeetbecher	92	22	44,64	37,19	50,91	16,93	15,60	23,09	9,59	8,63	15,55
23	Sitzbanke	165	41	44,02	42,23	45,85	29,94	34,60	36,49	22,06	26,37	30,20
24	Badtextilien	542	135	42,67	40,58	45,78	26,66	29,18	34,93	22,13	24,43	32,65
25	Kissen	153	38	40,93	32,55	49,47	24,78	24,94	33,89	17,31	18,51	25,89
26	Hocker	220	55	40,85	39,86	45,82	34,14	38,65	41,38	30,27	36,28	37,93
27	Vorhangstoffe	288	71	35,22	31,81	43,94	26,74	27,55	38,08	22,35	25,37	33,58
28	Wandboards	148	36	34,14	34,42	40,56	16,14	19,03	25,22	10,17	11,93	19,06
29	Keramikschalen	114	28	33,69	41,21	35,71	18,65	27,15	25,86	11,38	16,89	18,43
30	LED-Deckenleuchten	250	62	32,68	34,96	39,03	18,56	22,00	28,71	13,07	15,83	23,39
31	Babykleidung	192	47	28,45	43,36	31,49	21,70	32,23	29,87	15,62	22,98	23,32
32	Wanduhren	116	28	25,31	37,71	30,00	13,96	19,17	20,86	9,41	12,84	16,07
33	Phonomobil	87	21	24,87	23,22	32,38	9,64	10,91	17,52	5,68	6,06	12,38
34	Kom-	649	162	24,61	30,09	30,86	12,50	16,23	21,90	8,78	11,66	18,23

	moden											
35	Badematten	228	57	23,18	32,88	28,77	9,38	12,22	16,84	5,94	7,11	13,58
36	Sitzhocker	137	34	21,14	24,97	24,71	8,41	11,09	14,00	4,98	6,35	10,47
37	Wandspiegel	244	60	20,99	26,96	24,67	7,73	10,38	14,20	5,06	6,68	12,13
38	Fumatten	111	27	20,17	28,34	25,93	14,97	20,72	27,11	13,04	18,95	25,04
39	Rollcontainer	144	35	20,15	34,18	21,71	14,65	28,24	18,17	10,91	22,04	14,57
40	Kuscheldecken	88	21	18,51	22,67	27,62	7,44	8,15	16,57	4,81	4,90	12,95
41	Diffuser	89	22	17,70	29,43	20,00	5,15	8,71	7,64	2,95	5,27	5,82
42	Zierkissen	101	25	17,37	16,64	25,60	7,75	6,54	18,72	5,40	4,32	15,36
43	Vitrinen	204	51	16,79	22,64	24,71	10,57	13,15	22,20	7,34	8,95	18,27
44	Regale	371	92	16,33	25,46	20,43	9,29	15,25	17,30	7,02	11,81	15,13
45	Tischwasche	240	60	13,17	23,64	17,33	4,85	7,22	10,93	3,09	4,24	9,20
46	Teelichthalter	91	22	11,29	16,81	14,55	3,02	3,62	6,73	1,61	1,88	4,64
47	Nachtkastchen	137	34	6,41	12,69	10,00	2,52	4,41	7,18	1,47	2,45	5,29
48	Laternen & Windlichter	84	20	4,55	10,46	7,00	1,79	3,40	5,80	0,99	1,85	3,60

Zusammengefasste Ergebnisse der zweiten Messung mit Transfer Learning und vortrainiertem Modell ohne Fine-Tuning:

MAP@5 (Avg) [%]	61,72
MAP@5 (Std) [%]	40,77
MAP@25 (Avg) [%]	52,07
MAP@25 (Std) [%]	39,18
MAP@50 (Avg) [%]	46,76
MAP@50 (Std) [%]	37,77
P@5 (Avg) [%]	65,90
P@25 (Avg) [%]	59,46
P@50 (Avg) [%]	55,15
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,75
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,34
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,88
Trainingsdauer [s]	0,00

Ergebnisse der einzelnen Kategorien der dritten Messung mit Transfer Learning und vortrainiertem Modell ohne Fine-Tuning:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Sofas	2788	697	92,50	21,13	93,54	87,61	23,37	90,81	85,06	24,22	89,20
2	Stumpfenkerzen	191	47	89,72	21,71	91,91	70,92	27,89	75,83	56,85	28,01	63,62
3	Sockelleisten	128	31	89,29	28,54	89,68	79,28	34,52	80,39	75,11	37,51	76,97
4	Bilder	1120	280	89,16	24,12	89,86	78,88	31,12	82,04	69,12	30,42	74,00
5	Bademantel	128	32	86,97	19,23	91,88	69,83	25,11	76,38	53,22	28,60	60,06
6	Leintuch	279	69	84,13	27,10	86,09	65,95	23,17	74,20	52,28	18,44	62,67
7	Sessel	1152	288	84,10	30,48	85,83	73,50	34,02	78,36	68,69	35,28	74,85
8	Bratpfannen	120	29	83,11	32,97	85,52	74,84	33,01	79,72	65,37	31,67	71,52
9	Betten	1246	311	80,43	34,53	83,09	74,17	33,35	79,65	69,24	33,98	76,17
10	Matratzen	518	129	78,85	32,08	82,64	72,32	29,85	81,05	64,67	26,34	75,12
11	Lattenroste	260	64	73,20	36,66	77,19	57,67	33,70	66,88	47,53	31,23	57,41
12	Bettwasche	154	38	72,81	35,70	75,26	53,43	34,05	59,47	35,69	23,63	43,53
13	Teppiche	2218	554	63,42	36,67	72,02	60,49	33,47	70,90	56,52	32,50	67,56
14	Tische	1466	366	63,06	37,82	68,14	53,55	37,39	62,05	48,84	36,04	58,54
15	Fertigvorhänge	308	76	62,80	38,22	68,68	55,91	35,12	65,58	50,68	32,35	61,89
16	Sitzsacke	157	39	59,03	43,94	62,05	48,70	41,37	54,36	41,40	38,63	47,85
17	Wohnwaende	353	88	56,71	40,82	60,91	45,23	35,44	54,23	38,41	30,86	49,23
18	Schraenke	1425	356	55,90	38,04	61,97	41,27	31,60	53,01	34,35	28,40	47,50
19	Lampen	1197	299	53,73	40,94	58,80	39,63	34,07	48,67	33,43	29,99	44,63
20	Eckbanke	141	35	52,01	36,29	56,57	25,63	24,55	33,60	16,12	15,03	25,03
21	Teller	180	44	47,71	30,52	54,09	31,71	21,64	46,73	27,02	18,74	43,73
22	Kaffeebecher	92	22	45,52	36,95	51,82	16,91	15,57	23,27	9,54	8,61	15,55
23	Sitzbanke	165	41	44,61	42,26	46,83	29,94	34,60	36,49	22,06	26,37	30,20
24	Badtextilien	542	135	42,67	40,58	45,78	26,65	29,18	34,90	22,13	24,43	32,64
25	Hocker	220	55	41,10	39,71	46,18	34,12	38,66	41,31	30,27	36,28	37,93
26	Kissen	153	38	38,89	31,09	48,42	24,50	25,06	33,47	17,30	18,52	25,84
27	Vorhangstoffe	288	71	35,22	31,81	43,94	26,73	27,56	38,03	22,34	25,37	33,58
28	Wandboards	148	36	34,14	34,42	40,56	16,14	19,03	25,22	10,17	11,92	19,06
29	Keramikschalen	114	28	33,69	41,21	35,71	18,65	27,15	25,86	11,38	16,89	18,43
30	LED-Deckenleuchten	250	62	32,61	35,02	38,71	18,56	22,00	28,71	13,07	15,83	23,35
31	Babykleidung	192	47	27,79	43,47	30,64	21,70	32,23	29,87	15,61	22,99	23,28
32	Wanduhren	116	28	25,31	37,71	30,00	13,98	19,16	21,00	9,41	12,84	16,07
33	Phonomobel	87	21	25,11	23,73	32,38	9,24	9,71	17,14	5,73	6,06	12,48
34	Kom-	649	162	24,96	30,04	31,36	12,54	16,27	21,95	8,76	11,62	18,25

	moden											
35	Badematten	228	57	23,18	32,88	28,77	9,38	12,22	16,84	5,95	7,11	13,61
36	Wandspiegel	244	60	21,01	26,95	24,67	7,67	10,13	14,07	5,10	6,80	12,17
37	Rollcontainer	144	35	20,57	32,96	22,86	14,94	27,97	18,63	10,77	21,92	14,34
38	Fumatten	111	27	20,17	28,34	25,93	14,97	20,72	27,11	13,04	18,95	25,04
39	Sitzhocker	137	34	20,07	23,02	22,94	8,45	10,87	14,12	4,87	6,03	10,18
40	Kuscheldecken	88	21	18,51	22,67	27,62	7,44	8,15	16,57	4,81	4,90	12,95
41	Diffuser	89	22	17,70	29,43	20,00	5,15	8,71	7,64	2,95	5,27	5,82
42	Zierkissen	101	25	16,97	16,94	24,80	7,80	6,63	18,88	5,31	4,20	15,12
43	Vitrinen	204	51	16,79	22,64	24,71	10,57	13,15	22,20	7,36	8,97	18,31
44	Regale	371	92	16,04	25,16	20,22	9,28	15,22	17,30	7,00	11,76	15,11
45	Tischwasche	240	60	13,17	23,64	17,33	4,88	7,21	11,07	3,09	4,24	9,20
46	Teelichthalter	91	22	10,83	16,98	13,64	2,94	3,66	6,18	1,60	1,89	4,55
47	Nachtkastchen	137	34	6,59	12,85	10,00	2,52	4,41	7,18	1,45	2,43	5,18
48	Laternen & Windlichter	84	20	4,50	10,46	7,00	1,79	3,40	5,80	0,99	1,85	3,60

Zusammengefasste Ergebnisse der dritten Messung mit Transfer Learning und vortrainiertem Modell ohne Fine-Tuning:

MAP@5 (Avg) [%]	61,69
MAP@5 (Std) [%]	40,75
MAP@25 (Avg) [%]	52,06
MAP@25 (Std) [%]	39,18
MAP@50 (Avg) [%]	46,76
MAP@50 (Std) [%]	37,77
P@5 (Avg) [%]	65,91
P@25 (Avg) [%]	59,46
P@50 (Avg) [%]	55,15
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,75
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,35
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,89
Trainingsdauer [s]	0,00

## ANHANG D - Transfer Learning mit Fine-Tuning Ergebnisse

Ergebnisse der einzelnen Kategorien der ersten Messung mit Transfer Learning und vortrainiertem Modell mit Fine-Tuning:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Matratzen	518	129	99,07	8,97	99,07	99,08	8,95	99,16	99,13	8,86	99,19
2	Bademantel	128	32	98,50	5,90	98,75	98,07	3,53	99,00	98,27	2,32	99,19
3	Betten	1246	311	95,97	18,66	96,33	95,42	18,61	95,96	95,30	18,77	95,95
4	Leintücher	279	69	95,65	20,54	95,65	93,12	22,98	93,57	91,43	24,06	92,12
5	Stumpenkerzen	191	47	93,53	13,54	95,32	90,82	17,15	94,13	90,48	18,85	93,87
6	Sofas	2788	697	91,53	25,16	92,42	90,14	26,58	91,82	89,85	26,90	91,81
7	Bilder	1120	280	91,40	23,47	92,71	90,38	24,93	92,59	89,42	26,47	91,74
8	Sessel	1152	288	89,43	28,91	90,35	89,03	29,85	89,88	88,80	30,27	89,60
9	Sockelleisten	128	31	88,28	31,07	89,03	87,23	33,25	87,74	87,13	33,61	87,61
10	Lattenroste	260	64	87,50	33,33	87,50	87,50	33,33	87,50	87,50	33,33	87,50
11	Eckbanke	141	35	86,63	29,73	86,86	81,79	35,38	83,31	78,16	36,80	80,40
12	Bettwasche	154	38	85,55	30,65	86,84	81,02	35,54	83,05	78,55	38,12	80,42
13	Tische	1466	366	79,84	35,89	81,69	76,65	37,40	80,16	75,48	37,80	79,52
14	Bratpfannen	120	29	77,08	36,34	79,31	71,65	41,98	74,90	69,81	43,55	72,90
15	Teppiche	2218	554	71,84	40,70	74,26	66,87	39,75	72,74	65,22	40,03	71,79
16	Wohnwände	353	88	70,12	41,30	74,09	68,57	41,97	73,32	68,02	42,09	73,02
17	Sitzbanke	165	41	69,63	42,08	71,71	69,37	39,06	75,61	70,00	38,31	76,24
18	Schraenke	1425	356	69,31	40,10	73,43	65,88	40,09	72,35	65,01	40,33	72,05
19	Hocker	220	55	67,73	42,70	70,91	64,94	41,40	70,18	62,34	41,56	68,36
20	Kaffeebecher	92	22	67,70	42,35	70,91	63,78	45,45	67,27	59,17	44,63	63,09
21	Lampen	1197	299	66,89	40,07	71,44	62,77	39,14	69,73	61,52	38,87	69,32
22	Badtextilien	542	135	63,77	45,35	65,93	61,39	44,65	65,63	59,94	44,04	64,74
23	Fertigvorhänge	308	76	59,79	40,96	66,05	59,55	40,63	68,32	59,77	40,47	68,50
24	Sitzsacke	157	39	56,74	49,12	57,95	55,74	48,36	58,67	54,58	48,42	57,54
25	Vorhangstoffe	288	71	54,68	45,48	58,03	48,99	41,73	54,48	47,19	41,09	52,96
26	Teller	180	44	52,57	28,22	63,64	45,53	21,25	62,00	43,25	20,01	60,68
27	Kissen	153	38	50,81	33,25	62,11	40,28	26,69	56,42	36,37	23,38	54,16
28	Babykleidung	192	47	48,84	45,43	51,06	45,31	47,28	48,09	44,74	47,86	48,00
29	Wandboards	148	36	46,51	40,47	53,33	40,48	34,77	51,33	38,60	33,62	49,94
30	Wandspiegel	244	60	44,46	43,76	48,33	41,78	41,48	47,80	38,94	38,87	45,73
31	Kuscheldecken	88	21	44,02	46,36	48,57	41,43	45,48	45,90	30,05	34,23	34,38

32	Fumatten	111	27	43,56	36,86	50,37	36,14	33,54	47,26	31,15	28,57	43,04
33	Keramikschalen	114	28	42,20	42,01	47,14	39,46	42,07	46,00	36,78	41,19	44,07
34	Kommoden	649	162	41,65	38,73	49,38	38,66	34,20	50,86	36,72	33,24	48,93
35	Zierkissen	101	25	41,17	29,81	51,20	23,00	14,62	37,60	18,94	11,46	36,08
36	LED-Deckenleuchten	250	62	40,46	39,20	45,81	29,88	29,96	41,42	28,08	28,39	41,03
37	Phonomobil	87	21	35,75	35,59	43,81	27,71	28,01	38,67	24,22	26,07	34,95
38	Sitzhocker	137	34	33,36	36,63	37,06	30,60	35,28	37,18	27,11	31,63	33,94
39	Rollcontainer	144	35	31,84	37,09	40,00	26,67	39,36	32,91	25,17	40,51	30,17
40	Wanduhren	116	28	27,43	38,95	30,71	21,36	30,15	28,14	17,44	23,59	25,21
41	Badematten	228	57	25,64	32,97	31,23	17,08	21,87	27,44	13,29	16,68	24,04
42	Regale	371	92	22,42	33,32	27,83	18,14	26,46	28,61	15,99	22,25	27,83
43	Vitrinen	204	51	21,60	23,95	30,98	14,86	13,28	29,80	12,62	10,28	28,71
44	Diffuser	89	22	21,52	30,90	27,27	8,28	9,98	16,55	5,34	6,78	13,36
45	Nachtkastchen	137	34	15,23	22,80	22,35	10,89	15,57	19,06	8,33	11,10	17,53
46	Tischwasche	240	60	11,10	21,55	16,33	5,13	10,29	10,93	3,42	6,19	10,10
47	Teelichthalter	91	22	6,58	10,58	10,91	2,82	3,88	9,09	1,83	2,28	7,45
48	Laternen & Windlichter	84	20	1,80	3,71	6,00	1,79	2,94	7,80	1,67	3,27	7,50

Zusammengefasste Ergebnisse der ersten Messung mit Transfer Learning und vortrainiertem Modell mit Fine-Tuning:

MAP@5 (Avg) [%]	71,58
MAP@5 (Std) [%]	40,45
MAP@25 (Avg) [%]	68,52
MAP@25 (Std) [%]	40,80
MAP@50 (Avg) [%]	67,31
MAP@50 (Std) [%]	41,13
P@5 (Avg) [%]	74,43
P@25 (Avg) [%]	73,25
P@50 (Avg) [%]	72,45
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,78
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,13
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,63
Trainingsdauer [s]	1094,28



Ergebnisse der einzelnen Kategorien der zweiten Messung mit Transfer Learning und vortrainiertem Modell mit Fine-Tuning:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Bademantel	128	32	100,00	0,00	100,00	99,75	0,98	99,75	97,05	5,24	97,69
2	Stumpenkerzen	191	47	98,20	7,16	98,72	96,59	8,67	97,70	95,64	10,15	97,23
3	Matratzen	518	129	98,10	13,00	98,29	97,38	13,72	97,83	97,46	13,91	97,91
4	Betten	1246	311	95,54	18,57	96,01	95,47	18,99	96,15	95,48	18,97	96,21
5	Leintucher	279	69	94,14	22,56	94,20	83,97	27,14	86,09	74,74	29,72	78,93
6	Bilder	1120	280	93,65	18,92	95,71	93,13	20,46	94,86	93,05	19,93	95,18
7	Sofas	2788	697	93,55	21,25	94,58	91,81	22,95	93,47	91,45	23,62	93,34
8	Sessel	1152	288	91,27	26,66	91,88	90,77	27,39	91,74	90,72	27,50	91,78
9	Lattenroste	260	64	88,63	30,61	90,31	88,80	29,00	90,75	88,57	26,70	91,19
10	Sockelleisten	128	31	88,06	31,67	88,39	87,30	33,54	87,48	87,20	33,81	87,29
11	Bettwasche	154	38	87,63	32,34	87,89	88,19	31,30	88,74	88,41	31,19	89,05
12	Eckbanke	141	35	84,01	32,95	84,57	81,32	33,81	84,23	80,84	34,34	83,54
13	Tische	1466	366	81,75	34,59	84,10	81,45	34,55	84,43	81,29	34,65	84,37
14	Kaffeetbecher	92	22	81,03	36,63	81,82	75,26	39,60	78,73	74,29	40,51	77,91
15	Bratpfannen	120	29	80,72	37,14	82,07	78,29	39,79	80,14	77,48	40,57	79,45
16	Teppiche	2218	554	72,46	41,01	75,20	69,97	39,38	75,04	69,38	39,54	74,58
17	Sitzbanke	165	41	70,83	38,92	75,61	67,01	40,68	73,17	64,50	42,25	70,83
18	Schraenke	1425	356	69,60	39,78	73,60	67,51	39,14	73,96	66,51	39,16	73,53
19	Wohnwaende	353	88	66,48	44,79	69,09	64,46	44,71	68,36	63,86	45,00	68,23
20	Sitzsacke	157	39	64,94	46,00	66,15	61,54	47,55	63,18	60,57	48,30	61,54
21	Lampen	1197	299	63,72	39,75	68,49	59,20	39,48	66,65	58,06	39,53	66,09
22	Fertigvorhange	308	76	63,66	37,85	68,68	61,77	36,05	70,58	62,17	35,33	71,84
23	Badtextilien	542	135	60,40	44,74	62,67	56,17	43,73	60,68	55,77	43,67	60,56
24	Hocker	220	55	59,58	43,90	64,00	54,52	42,54	61,24	51,24	41,69	58,58
25	Vorhangstoffe	288	71	57,94	40,16	62,82	51,45	36,64	60,45	50,70	36,49	60,17
26	Kissen	153	38	55,83	29,41	65,26	44,07	17,93	61,16	39,07	15,22	57,32
27	Teller	180	44	54,53	35,89	61,82	45,30	31,18	57,55	45,53	31,43	59,14
28	Wandboards	148	36	49,24	39,18	57,22	49,07	39,14	58,22	48,48	38,68	57,50
29	Kuscheldecken	88	21	48,14	45,99	50,48	38,95	43,47	43,62	25,99	28,53	30,86
30	Babykleidung	192	47	42,98	48,99	42,98	40,52	47,86	41,96	39,63	47,61	41,36
31	Keramikschalen	114	28	42,37	42,43	47,86	36,18	43,48	42,14	35,69	43,02	42,64
32	Kommoden	649	162	39,86	35,58	48,64	37,57	32,30	50,67	36,50	31,52	50,19
33	Wandspiegel	244	60	39,83	42,14	44,67	37,58	39,49	45,07	35,60	38,19	44,00
34	LED-Decken-	250	62	38,76	37,92	45,81	34,55	32,26	45,94	32,26	30,48	44,16

	leuchten											
35	Roll-container	144	35	37,81	38,29	40,00	27,04	40,41	30,74	26,00	40,98	31,14
36	Phono-mobel	87	21	30,81	33,87	40,95	31,38	34,41	41,90	28,84	32,69	38,86
37	Fumatten	111	27	27,74	38,26	34,07	22,18	27,55	33,78	22,25	26,38	37,19
38	Diffuser	89	22	25,80	41,53	28,18	13,62	20,73	19,27	11,28	17,38	18,64
39	Zier-kissen	101	25	25,20	19,00	37,60	17,06	10,43	34,24	16,14	7,81	35,60
40	Regale	371	92	24,68	33,55	31,09	19,28	27,88	29,00	18,17	25,85	29,39
41	Bade-matten	228	57	24,37	29,35	33,68	16,20	22,13	26,95	13,13	20,51	23,12
42	Wand-uhren	116	28	24,04	34,79	28,57	21,77	32,00	28,86	17,31	23,99	25,36
43	Sitz-hocker	137	34	21,22	32,50	25,88	9,48	16,29	15,06	7,03	13,06	13,12
44	Vitrinen	204	51	18,17	26,98	26,67	15,63	20,23	30,35	14,23	16,87	29,49
45	Tisch-wasche	240	60	16,11	24,41	20,67	7,22	10,85	14,40	5,16	7,60	12,77
46	Teelicht-halter	91	22	14,73	25,38	20,91	8,12	13,23	15,45	5,65	8,38	13,45
47	Laternen & Wind-lichter	84	20	8,87	22,98	13,00	4,76	11,43	10,20	3,47	8,51	8,90
48	Nacht-kastchen	137	34	7,64	14,98	14,71	8,07	11,77	18,82	7,78	10,62	18,76

Zusammengefasste Ergebnisse der zweiten Messung mit Transfer Learning und vortrainiertem Modell mit Fine-Tuning:

MAP@5 (Avg) [%]	71,87
MAP@5 (Std) [%]	40,20
MAP@25 (Avg) [%]	69,24
MAP@25 (Std) [%]	40,39
MAP@50 (Avg) [%]	68,37
MAP@50 (Std) [%]	40,59
P@5 (Avg) [%]	74,87
P@25 (Avg) [%]	73,92
P@50 (Avg) [%]	73,42
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,88
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,39
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,83
Trainingsdauer [s]	1039,75

Ergebnisse der einzelnen Kategorien der dritten Messung mit Transfer Learning und vortrainiertem Modell mit Fine-Tuning:

Rang	Kategorie	Trainingsbilder (20814)	Testbilder (5178)	MAP@5 (Avg) [%]	MAP@5 (Std) [%]	P@5 (Avg) [%]	MAP@25 (Avg) [%]	MAP@25 (Std) [%]	P@25 (Avg) [%]	MAP@50 (Avg) [%]	MAP@50 (Std) [%]	P@50 (Avg) [%]
1	Bademantel	128	32	100,00	0,00	100,00	99,14	2,03	99,38	99,36	1,52	99,69
2	Matratzen	518	129	99,22	8,80	99,22	99,19	8,81	99,19	99,10	8,81	99,13
3	Leintücher	279	69	95,65	20,54	95,65	93,77	22,49	93,91	93,07	23,75	93,51
4	Stumpfenkerzen	191	47	94,77	15,37	95,74	95,09	13,86	97,02	94,96	14,31	96,94
5	Bilder	1120	280	94,61	18,89	95,21	92,00	24,09	93,11	91,93	24,73	93,04
6	Betten	1246	311	94,07	21,97	94,60	92,61	23,68	93,40	91,95	24,71	92,91
7	Sessel	1152	288	93,56	22,87	93,96	93,32	23,30	94,01	93,42	23,24	94,09
8	Sofas	2788	697	93,22	22,10	94,40	93,07	22,22	94,62	93,02	22,37	94,52
9	Bettwäsche	154	38	91,05	26,18	91,05	87,89	31,40	88,32	86,94	32,57	87,53
10	Sockelleisten	128	31	89,10	29,48	89,68	88,76	30,58	89,68	89,23	30,15	90,06
11	Eckbanke	141	35	88,11	27,24	89,14	88,93	27,62	90,29	87,91	27,67	89,37
12	Lattenroste	260	64	86,44	34,01	86,88	87,47	31,60	89,63	87,42	31,92	89,13
13	Sitzbanke	165	41	84,07	34,49	85,85	80,57	33,29	84,29	78,01	32,64	82,83
14	Bratpfannen	120	29	82,93	34,48	86,90	80,87	35,24	84,41	78,66	38,13	81,03
15	Tische	1466	366	82,88	33,76	84,70	81,83	33,45	85,06	82,18	33,31	85,47
16	Sitzsacke	157	39	78,23	40,36	78,97	75,19	41,25	76,41	73,80	42,07	75,28
17	Kaffeebecher	92	22	77,73	36,73	78,18	70,66	40,69	74,00	67,14	41,37	70,64
18	Teppiche	2218	554	76,70	37,93	79,93	75,57	36,36	80,34	74,95	36,15	80,18
19	Fertigvorhänge	308	76	73,43	32,91	76,58	66,10	33,56	75,00	64,50	32,35	75,16
20	Hocker	220	55	70,36	44,42	71,27	66,23	44,16	68,51	65,26	43,98	68,69
21	Schraenke	1425	356	67,42	39,78	71,97	64,25	40,39	71,39	63,45	40,49	71,21
22	Wohnwaende	353	88	66,58	42,87	70,00	65,70	43,61	70,36	65,35	43,75	70,32
23	Badtextilien	542	135	65,06	45,30	66,81	63,28	44,85	66,25	62,87	44,92	66,21
24	Lampen	1197	299	63,48	40,48	68,36	59,74	38,84	67,68	58,29	38,61	66,95
25	Vorhangstoffe	288	71	63,30	41,41	68,45	60,75	39,46	67,27	58,64	37,65	66,03
26	Phonomobil	87	21	52,95	41,14	57,14	43,92	37,62	51,05	39,47	34,94	47,24
27	Rollcontainer	144	35	50,96	42,87	53,71	37,79	38,68	45,14	33,31	39,20	40,29
28	Teller	180	44	47,56	30,68	58,64	38,60	25,41	54,45	41,21	24,99	59,50
29	Wandboards	148	36	45,93	37,52	53,89	40,78	35,46	50,78	38,78	35,49	49,17
30	Kissen	153	38	44,98	34,62	56,84	40,32	24,84	57,16	38,00	21,96	56,58
31	Kuscheldecken	88	21	44,41	47,38	47,62	43,62	46,84	47,24	32,77	35,70	35,90
32	Kommoden	649	162	43,83	39,54	50,99	38,92	35,00	49,65	37,75	34,09	49,52
33	Babykleidung	192	47	42,55	48,07	43,83	40,60	48,84	41,19	40,02	48,76	40,81
34	Sitzhocker	137	34	42,28	45,79	43,53	23,24	27,45	29,65	19,33	23,32	27,88

35	Wandspiegel	244	60	41,96	40,72	47,67	39,27	39,16	47,67	37,80	38,30	46,73
36	Keramikschalen	114	28	40,26	40,34	48,57	39,99	39,50	50,86	41,40	38,98	52,36
37	LED-Deckenleuchten	250	62	36,96	34,35	44,52	32,18	29,92	44,26	30,38	28,69	43,00
38	Wanduhren	116	28	32,44	41,37	36,43	24,71	30,64	31,57	21,96	26,89	30,00
39	Zierkissen	101	25	31,31	31,87	42,40	24,13	24,69	38,40	19,71	17,27	36,24
40	Diffuser	89	22	27,05	34,44	32,73	16,56	24,10	25,45	14,61	22,12	24,73
41	Regale	371	92	24,03	30,92	30,87	19,04	26,06	28,74	17,65	24,40	28,24
42	Fumatten	111	27	23,41	31,20	28,15	20,11	25,93	34,67	18,91	24,04	34,52
43	Badematten	228	57	23,33	28,84	31,58	17,77	19,86	30,74	14,81	15,58	28,42
44	Nachtkastchen	137	34	21,81	32,43	26,47	16,53	24,42	25,65	14,32	20,08	23,88
45	Vitrinen	204	51	20,58	22,02	29,80	15,75	17,17	30,35	13,79	14,19	29,45
46	Tischwasche	240	60	16,94	22,74	22,33	8,82	12,69	17,20	6,63	9,45	15,43
47	Laternen & Windlichter	84	20	7,33	16,00	11,00	4,95	10,83	11,20	4,01	8,79	10,30
48	Teelichthalter	91	22	6,67	13,89	9,09	2,70	5,36	6,73	1,56	3,01	4,91

Zusammengefasste Ergebnisse der dritten Messung mit Transfer Learning und vortrainiertem Modell mit Fine-Tuning:

MAP@5 (Avg) [%]	73,37
MAP@5 (Std) [%]	39,44
MAP@25 (Avg) [%]	70,95
MAP@25 (Std) [%]	39,75
MAP@50 (Avg) [%]	70,19
MAP@50 (Std) [%]	39,96
P@5 (Avg) [%]	76,25
P@25 (Avg) [%]	75,51
P@50 (Avg) [%]	75,12
Abfragezeit 5 Suchergebnisse (Avg) [ms]	0,90
Abfragezeit 25 Suchergebnisse (Avg) [ms]	1,38
Abfragezeit 50 Suchergebnisse (Avg) [ms]	1,83
Trainingsdauer [s]	1022,20

## ABKÜRZUNGSVERZEICHNIS

AI	Artificial Intelligence
ANNOY	Approximate nearest neighbor oh yeah
API	Application Programming Interface
AP@K	Average Precision@K
AUC	Area under the curve
CNN	Convolutional Neural Networks
CRISP-DM	Cross Industry Standard Process for Data Mining
GLM	Generalized Linear Models
GPR	Gaussian Process Regression
LSH	Locality sensitive hashing
MAP@K	Mean Average Precision@K
ML	Machine Learning
PCA	Principal component analysis
P@K	Precision@K
REST	Representational State Transfer
RNN	Recurrent Neural Networks
ROC	Receiver-Operating characteristic
SVR	Support Vector Regression
T-SNE	T-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection

## ABBILDUNGSVERZEICHNIS

Abbildung 2-1: Übersicht von Machine Learning Algorithmen (vgl. Ajay, 2017) .....	12
Abbildung 2-2: CRISP-DM (vgl. Wirth & Hipp, 2000) .....	13
Abbildung 2-3: Machine Learning Prozess (vgl. Kamath & Choppella, 2017) .....	15
Abbildung 2-4: Confusion Matrix (vgl. Smith, 2018) .....	17
Abbildung 2-5: ROC Kurve mit AUC (in Anlehnung an Smith, 2018) .....	18
Abbildung 2-6: Aufbau eines Neuron (in Anlehnung an Ravichandiran, 2019).....	20
Abbildung 2-7: Neurales Netz Layer (vgl. Kostadinov, 2018) .....	21
Abbildung 2-8: Deep Neural Network mit Fully Connected Layer (vgl. Sarkar et al., 2018) .....	22
Abbildung 2-9: CNN Architektur (in Anlehnung an Kim, 2017) .....	23
Abbildung 2-10: Recurrent Neural Network Architektur (vgl. Kostadinov, 2018) .....	24
Abbildung 2-11: Adversarial Network Architektur (in Anlehnung an Osinga, 2018).....	25
Abbildung 5-1: Retrieval Pipeline (vgl. Bonaccorso et al., 2018) .....	38
Abbildung 6-1: Autoencoder (in Anlehnung an Hubens, 2018a).....	41
Abbildung 6-2: Triplet (in Anlehnung an Zingade, 2017).....	43
Abbildung 6-3: Deep Ranking Architektur (vgl. Zingade, 2017).....	44
Abbildung 6-4: Siamese Network Architektur (in Anlehnung an Loy, 2019) .....	46
Abbildung 6-5: Schematischer Ablauf von Transfer Learning anhand Dissected VGG (in Anlehnung an Wong, 2017) .....	48
Abbildung 8-1: Beispiel eines Produktbildes .....	60
Abbildung 8-2: Kategoriestructur .....	61
Abbildung 8-3: Histogramm originales und unmodifiziertes Datenset .....	64
Abbildung 8-4: Boxplot originales und unmodifiziertes Datenset .....	65
Abbildung 8-5: Histogramm modifiziertes Datenset .....	67
Abbildung 8-6: Boxplot modifiziertes Datenset .....	67
Abbildung 8-7: Suchergebnis visuelle Darstellung .....	75
Abbildung 8-8: Visualisierung mit TensorBoard und t-SNE gute Ergebnisse .....	76
Abbildung 8-9: Visualisierung mit TensorBoard und t-SNE schlechte Ergebnisse .....	77
Abbildung 9-1: Autoencoder beste Ergebnisse ersten fünf Kategorien visuell .....	82
Abbildung 9-2: Autoencoder schlechtesten Ergebnisse letzten fünf Kategorien visuell .....	83
Abbildung 9-3: Siamese Network beste Ergebnisse ersten fünf Kategorien visuell .....	86
Abbildung 9-4: Siamese Network schlechtesten Ergebnisse letzten fünf Kategorien visuell .....	87
Abbildung 9-5: Transfer Learning beste Ergebnisse ersten fünf Kategorien visuell .....	90
Abbildung 9-6: Transfer Learning schlechtesten Ergebnisse letzten fünf Kategorien visuell .....	91
Abbildung 10-1: Vergleich visuelle Suchergebnisse .....	93
Abbildung 10-2: Autoencoder Suchergebnis mit niedrigster Average Precision.....	94
Abbildung 10-3: Siamese Network Suchergebnis mit niedrigster Average Precision.....	95
Abbildung 10-4: Transfer Learning Suchergebnis mit niedrigster Average Precision .....	95

## TABELLENVERZEICHNIS

Tabelle 1: Autoencoder Evaluierungsmetriken.....	80
Tabelle 2: Autoencoder generelle Metriken.....	80
Tabelle 3: Autoencoder Ergebnisse Kategorien.....	81
Tabelle 4: Siamese Network Evaluierungsmetriken.....	84
Tabelle 5: Siamese Network generelle Metriken.....	84
Tabelle 6: Siamese Network Ergebnisse Kategorien.....	85
Tabelle 7: Transfer Learning Evaluierungsmetriken.....	88
Tabelle 8: Transfer Learning generelle Metriken.....	88
Tabelle 9: Transfer Learning Ergebnisse Kategorien.....	89
Tabelle 10: Metrikenübersicht der Verfahren.....	92
Tabelle 11: Übersicht genereller Metriken der Verfahren.....	93

## LITERATURVERZEICHNIS

- Ajay. (2017). A Review on Machine Learning Algorithms, Tasks and Applications.
- Appalaraju, S. & Chaoji, V. (2017). Image similarity using deep CNN and curriculum learning. *arXiv preprint arXiv:1709.08761*.
- Ayyadevara, V. K. (2019). *Neural Networks with Keras Cookbook*. Birmingham, England: Packt Publishing.
- Bernico, M. (2018). *Deep learning quick reference. Useful hacks for training and optimizing deep neural networks with TensorFlow and Keras*.
- Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform. A Comprehensive Guide for Beginners* (1st ed. 2019). Verfügbar unter <https://doi.org/10.1007/978-1-4842-4470-8>
- Böing, C. (2001). *Erfolgsfaktoren im Business-to-Consumer-E-Commerce* (Unternehmensführung und Marketing, Bd. 38). Wiesbaden: Gabler Verlag. <https://doi.org/10.1007/978-3-322-96453-3>
- Bonaccorso, G. (2018a). *Machine Learning Algorithms. Popular Algorithms for Data Science and Machine Learning, 2nd Edition* (2nd ed.). Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5504925>
- Bonaccorso, G. (2018b). *Mastering machine learning algorithms. Expert techniques to implement popular machine learning algorithms and fine-tune your models*. Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781788621113>
- Bonaccorso, G., Fandango, A. & Shanmugamani, R. (2018). *Python. Expert Machine Learning Systems and Intelligent Agents Using Python*. Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5626921>
- Brink, H., Richards, J. W. & Fetherolf, M. (2017). *Real-world machine learning*. Shelter Island, NY: Manning Publications Co. Retrieved from <http://proquest.tech.safaribooksonline.de/9781617291920>
- Buduma, N. & Locascio, N. (2017). *Fundamentals of deep learning. Designing next-generation machine intelligence algorithms* (First edition). Sebastopol, CA: O'Reilly Media. Retrieved from <http://proquest.tech.safaribooksonline.de/9781491925607>
- Callaert, Y. (2018). *Getting started with mlFlow*. Zugriff am 07.10.2019. Verfügbar unter <https://towardsdatascience.com/getting-started-with-mlflow-52eff8c09c61>
- Chollet, F. (2018). *Deep learning with Python* (Safari Tech Books Online). Shelter Island, NY: Manning. Retrieved from <http://proquest.safaribooksonline.com/9781617294433>



- Ciaburro, G. J. P. (2019). *PYTHON MACHINE LEARNING COOKBOOK. Over 100 recipes to progress from smart data analytics to deep... learning using real-world datasets, 2nd edition.* [S.I.]: PACKT PUBLISHING LIMITED.
- Cirillo, A. (2017). *R data mining. Implement data mining techniques through practical use cases and real-world datasets.* Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781787124462>
- Coelho, L. P., Richert, W. & Brucher, M. (2018). *Building machine learning systems with Python. Explore machine learning and deep learning techniques for building intelligent systems using scikit-learn and TensorFlow* (Third edition). Birmingham, UK: Packt Publishing.
- Dey, S. (2018). *Hands-On Image Processing with Python. Expert Techniques for Advanced Image Analysis and Effective Interpretation of Image Data.* Birmingham: Packt Publishing Ltd.
- Di, W., Bhardwaj, A. & Wei, J. (2018). *Deep learning essentials. Your hands-on guide to the fundamentals of deep learning and neural network modeling.* Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781785880360>
- Dua, R., Ghotra, M. S. & Pentreath, N. (2017). *Machine learning with Spark. Develop intelligent machine learning systems with Spark 2.x* (Second edition). Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781785889936>
- Fuentes, A. (2018). *Hands-On Predictive Analytics with Python. Master the Complete Predictive Analytics Process, from Problem Definition to Model Deployment.* Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5627919>
- Garner, H. (2015). *Clojure for data science. Statistics, big data, and machine learning for Clojure programmers* (Community experience distilled). Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781784397180>
- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow. Concepts, tools, and techniques to build intelligent systems* (2nd ed.).
- Gollapudi, S. (2016). *Practical machine learning. Tackle the real-world complexities of modern machine learning with innovative and cutting-edge techniques* (Community experience distilled). Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781784399689>
- Grigorev, A., Shanmugamani, R., Boschetti, A., Massaron, L. & Thakur, A. (2018). *TensorFlow Deep Learning Projects. 10 real-world projects on computer vision, machine translation, chatbots, and reinforcement learning.* Birmingham: Packt Publishing.
- Halder, S. & Ozdemir, S. (2018). *Hands-On Machine Learning for Cybersecurity. Safeguard Your System by Making Your Machines Intelligent Using the Python Ecosystem.*

- Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5628279>
- Harrington, P. (2012). *Machine learning in action* (Safari Tech Books Online). Greenwich, Conn.: Manning. Retrieved from <http://proquest.safaribooksonline.com/9781617290183>
- Heinemann, G. (2019). *Der neue Online-Handel. Geschäftsmodelle, Geschäftssysteme und Benchmarks im E-Commerce* (10. Aufl. 2019). Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-23686-1>
- Holdroyd, T. (2019). *TensorFlow 2.0 quick start guide. Get up to speed with the newly introduced features of TensorFlow 2.0*. Birmingham, UK: Packt Publishing.
- Hubens, N. (2018a). *Build a simple Image Retrieval System with an Autoencoder*. Zugriff am 24.08.2018. Verfügbar unter <https://towardsdatascience.com/build-a-simple-image-retrieval-system-with-an-autoencoder-673a262b7921>
- Hubens, N. (2018b). *Deep inside: Autoencoders*. Zugriff am 25.02.2018. Verfügbar unter <https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f>
- Jansen, S. (2018). *Hands-On Machine Learning for Algorithmic Trading. Design and Implement Investment Strategies Based on Smart Algorithms That Learn from Data Using Python*. Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5628283>
- Jing, Y., Liu, D. C., Kislyuk, D., Zhai, A., Xu, J., Donahue, J. et al. (2015). Visual Search at Pinterest. *CoRR*, *abs/1505.07647*.
- Kamath, U. & Choppella, K. (2017). *Mastering Java machine learning. Mastering and implementing advanced techniques in machine learning*. Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781785880513>
- Kapoor, A. (2019). *Hands-on artificial intelligence for IoT. Expert machine learning and deep learning techniques for developing smarter IoT systems*. Birmingham, UK: PACKT PUBLISHING LIMITED.
- Kasam, M., Ganju, S. & Koul, A. (2019). *Practical Deep Learning for Cloud, Mobile, and Edge*.
- Kim, P. (2017). *MATLAB deep learning. With machine learning, neural networks and artificial intelligence* (For professionals by professionals). Berkeley, California: Apress; Springer. Retrieved from <http://www.springer.com/de/book/9781484228456>
- Kollewe, T. & Keukert, M. (2014). *Praxiswissen E-Commerce. [das Handbuch für den erfolgreichen Online-Shop ; für Shopbetreiber und Existenzgründer ; Strategie, Planung, Umsetzung ; mit vielen Beispielszenarien]* (O'Reillys basics). Beijing: O'Reilly. Verfügbar unter <http://www.vlb.de/GetBlob.aspx?strDisposition=a&strIsbn=9783955615086>
- Kostadinov, S. (2018). *Recurrent neural networks with Python Quick Start Guide. Sequential learning and language modeling with TensorFlow*. Birmingham, UK: Packt Publishing.

- Krohn, J., Grant, B. & Aglaé, B. (2018). *Deep Learning Illustrated. A Visual, Interactive Guide to Artificial Intelligence* (1st edition): Addison-Wesley Professional.
- Kumar, R. (2019). *Machine learning quick reference. Quick and essential machine learning hacks for training smart data models*. Birmingham, UK: Packt Publishing.
- Lamba, H. (2019). *One Shot Learning with Siamese Networks using Keras*. Zugriff am 04.10.2019. Verfügbar unter <https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d>
- Lane, H., Howard, C. & Hapke, H. (2019). *Natural language processing in action. Understanding, analyzing, and generating text with Python*.
- Lee, W.-M. (2019). *Python machine learning*. Indianapolis, IN: Wiley.
- Liu, Y. (2019). *Python machine learning by example. Easy-to-follow examples that get you up and running with machine learning* (Second edition). Birmingham, UK: Packt Publishing.
- Loy, J. (2019). *Neural network projects with Python. The ultimate guide to using Python to explore the true power of neural networks through six projects*. Birmingham: Pakt Publishing.
- Mader, K. (2017). *Image Similarity with Siamese Networks*. Zugriff am 30.10.2019. Verfügbar unter <https://www.kaggle.com/kmader/image-similarity-with-siamese-networks>
- Malaeb, M. (2017). *Recall and Precision at k for Recommender Systems*. Zugriff am 27.10.2019. Verfügbar unter [https://medium.com/@m\\_n\\_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54](https://medium.com/@m_n_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54)
- Marsland, S. (2015). *Machine learning. An algorithmic perspective* (Chapman & Hall / CRC machine learning & pattern recognition series, Second edition). Boca Raton, FL: CRC Press. Retrieved from <http://proquest.tech.safaribooksonline.de/9781466583283>
- Meier, A. & Stormer, H. (2012). *eBusiness & eCommerce. Management der digitalen Wertschöpfungskette* (3. Aufl. 2012). Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-29802-8>
- Mitchell, T. M. (1997). *Machine learning* (McGraw-Hill series in computer science, International ed., [Reprint.]). New York, NY: McGraw-Hill.
- Mohammed, M., Bashier, E. B. M. & Khan, M. B. (2017). *Machine learning. Algorithms and applications*. Boca Raton: CRC Press. <https://doi.org/10.1201/9781315371658>
- Moolayil, J. (2019). *Learn Keras for deep neural networks. A fast-track approach to modern deep learning with Python*. New York, NY: Apress.
- Nielsen, J. (2001). *E-commerce user experience*. Fremont, Calif.: Nielsen Norman Group. Retrieved from <http://www.nngroup.com/reports/ecommerce>
- Olson, D. L. (2018). *Data Mining Models, Second Edition*. New York: Business Expert Press. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5355540>

- Ortega, J. M. (2018). *Mastering Python for Networking and Security. Leverage Python Scripts and Libraries to Overcome Networking and Security Issues*. Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5532253>
- Osinga, D. (2018). *Deep Learning Cookbook. Practical Recipes to Get Started Quickly*. Sebastopol, CA: O'Reilly Media. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&AN=1825034>
- Panchal, S. (2019). *Face Recognition from Scratch using Siamese Networks and TensorFlow*. Zugriff am 04.10.2019. Verfügbar unter <https://medium.com/predict/face-recognition-from-scratch-using-siamese-networks-and-tensorflow-df03e32f8cd0>
- Patterson, J. & Gibson, A. (2017). *Deep learning. A practitioner's approach* (First edition). Sebastopol, CA: O'Reilly Media. Retrieved from <http://proquest.tech.safaribooksonline.de/9781491924570>
- Poladi, S. R. (2018). *Matplotlib 3.0 Cookbook. Over 150 Recipes to Create Highly Detailed Interactive Visualizations Using Python*. Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5568270>
- Purkait, N. (2019). *Hands-on neural networks with Keras. Design and create neural networks using deep learning and artificial intelligence principles*. Birmingham, UK: Packt Publishing.
- Raschka, S. & Mirjalili, V. (2018). *Python machine learning. Machine learning and deep learning with Python, scikit-learn, and TensorFlow* (Expert insight, Second edition, fourth release, [fully revised and updated]). Birmingham: Packt Publishing.
- Ravichandiran, S. (2019). *HANDS-ON DEEP LEARNING ALGORITHMS WITH PYTHON. Master deep learning algorithms with math by... implementing them from scratch*. [S.l.]: PACKT PUBLISHING LIMITED.
- Reynolds, J. (2004). *The complete e-commerce book. Design, build & maintain a successful Web-based business* (Safari Books Online, 2nd ed.). San Francisco: Distributed to the Book trade in the U.S. by Publishers Group West. Retrieved from <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10180869>
- Rosenfeld, L., Morville, P. & Arango, J. (2015). *Information architecture. For the Web and beyond* (Fourth edition). Beijing: O'Reilly Media.
- Rothman, D., Lamons, M., Kumar, R., Nagaraja, A., Ziai, A. & Dixit, A. (2018). *Python. Build Applications to Intelligently Interact with the World Around You Using Python*. Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5626933>
- Sarkar, D., Bali, R. & Ghosh, T. (2018). *Hands-On Transfer Learning with Python. Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras*. Birmingham: Packt Publishing Ltd. Retrieved from <https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5507771>

- Sawtelle, S. (2016). *Mean Average Precision (MAP) For Recommender Systems*. Zugriff am 01.11.2019. Verfügbar unter <http://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html#Average-Precision>
- Shanmugamani, R. (2018). *Deep learning for computer vision. Expert techniques to train advanced neural networks using TensorFlow and Keras*. Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781788295628>
- Simeone, O. (2017). A Brief Introduction to Machine Learning for Engineers. *CoRR*, *abs/1709.02840*.
- Singh, H. (2019). *Practical Machine Learning and Image Processing. For Facial Recognition, Object Detection, and Pattern Recognition Using Python*. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-4149-3>
- Smith, J. (2018). *Machine learning systems. Designs that scale*. Shelter Island, NY: Manning Publications Co. Retrieved from <http://proquest.tech.safaribooksonline.de/9781617293337>
- Thanaki, J. (2018). *Machine Learning Solutions. Expert techniques to tackle complex machine learning problems using Python*. Birmingham: Packt Publishing.
- Vasilev, I. (2019). *Python deep learning. Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow* (Second edition). Birmingham, UK: Packt Publishing.
- Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J. et al. (2014). Learning Fine-Grained Image Similarity with Deep Ranking. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, S. 1386–1393. Washington, DC, USA: IEEE Computer Society. Verfügbar unter <https://doi.org/10.1109/CVPR.2014.180>
- Wessels, D. (2001). *Web caching* (1st ed.). Sebastopol, CA: O'Reilly & Associates. Retrieved from <http://proquest.tech.safaribooksonline.de/156592536X>
- Whitenack, D. (2017). *Machine learning with Go. Implement regression, classification, clustering, time-series models, neural networks, and more using the Go programming language*. Birmingham, UK: Packt Publishing. Retrieved from <http://proquest.tech.safaribooksonline.de/9781785882104>
- Wirth, R. & Hipp, J. (2000). CRISP-DM. Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (S. 29–39).
- Witten, I. H., Pal, C. J., Frank, E. & Hall, M. A. (2017). *Data mining. Practical machine learning tools and techniques* (Fourth edition). Cambridge, MA: Morgan Kaufmann. Retrieved from <http://proquest.tech.safaribooksonline.de/9780128043578>
- Wong, A. (2017). *Building a Similar Images Finder without any training!* Zugriff am 01.10.2019. Verfügbar unter <https://towardsdatascience.com/building-a-similar-images-finder-without-any-training-f69c0db900b5>

Zingade, A. (2017). *Image Similarity using Deep Ranking*. Zugriff am 28.09.2019. Verfügbar unter <https://medium.com/@akarshzingade/image-similarity-using-deep-ranking-c1bd83855978>