HAW HAMBURG

BACHELORARBEIT

Contrastive Learning mit Stable Diffusion–basierter Datenaugmentation

Verbesserung der Bildklassifikation durch synthetische Daten

vorgelegt am 30. August 2024

Paul Hofmann

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

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Medientechnik
Finkenau 35
22081 Hamburg

Zusammenfassung

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

Abstract

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

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

[1] LaTeX Word WYSIWYG-Programm (**voss**) oder (**schlosser**) erwerben.

1.1 Motivation

...

1.2 Zielsetzung

...

1.3 Aufbau der Arbeit

...

[1] ...

2 Theoretische Grundlagen

Dieses Kapitel behandelt die theoretischen Grundlagen, die zum Verständnis der zu entwickelnden Methode am wichtigsten sind. Dazu werden zunächst Grundbegriffe und Konzepte des Maschinellen Lernens und der Bildklassifikation erklärt. Anschließend geht es um Stable Diffusion, die darauf basierende Methode DA-Fusion, sowie um Contrastive Learning.

2.1 Maschinelles Lernen

Im Folgenden werden die Grundlagen des maschinellen Lernens erläutert, um ein Verständnis für die Methoden und Konzepte zu schaffen, die in dieser Arbeit verwendet werden. Dazu gehören Definitionen, die Unterscheidung zwischen überwachtem und unüberwachtem Lernen, sowie eine Einführung in Deep Learning und Convolutional Neural Networks.

2.1.1 Definition und Ursprung

Maschinelles Lernen (ML) ist ein Teilbereich der künstlichen Intelligenz (KI), in dem Computern das Lernen anhand von Erfahrung ermöglicht wird. Die ersten Durchbrüche in der KI kamen im Bezug auf Aufgaben, die für Menschen intellektuell eine große Herausforderung darstellten, die aber von Computern relativ einfach zu lösen waren, da sie als Liste formaler, mathematischer Regeln beschrieben werden konnten (Goodfellow et al., 2016). Die große Schwierigkeit lag aber in den Aufgaben, die für Menschen relativ einfach und intuitiv sind, welche sich aber nicht einfach formal beschreiben lassen. Die grundlegende Idee hinter maschinellem Lernen ist daher, Computer mit der Fähigkeit auszustatten, selbstständig Wissen aus Erfahrung zu generieren, indem Muster und Konzepte aus rohen Daten erlernt werden.

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

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

Dabei kommen verschiedene Aufgaben T in Frage, etwa die in dieser Arbeit thematisierte Bildklassifikation, oder aber auch Bilgsegmentierung, Anomaliedetektion, maschinelle Übersetzung, usw. Je nach Aufgabe können verschiedene Leistungsmaße P herangezogen werden, wie etwa die *Accuracy*, welche die Trefferrate richtiger Vorhersagen beschreibt. Auch die Erfahrung E kann je nach Lernmethode variieren, wie im nachfolgenden Abschnitt genauer erklärt wird.

2.1.2 Überwachtes und unüberwachtes Lernen

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

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

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

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

2.1.3 Deep Learning

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

Darstellung. Deep Learning hat in den letzten Jahren erhebliche Fortschritte gemacht und findet Anwendung in Bereichen wie Bild- und Spracherkennung, autonomen Fahrzeugen und vielen anderen.

2.1.4 Neuronale Netze

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

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

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

2.1.5 Convolutional Neural Networks

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

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

1. Eingabeschicht (Input Layer): Diese Schicht nimmt die Rohdaten auf, z.B. ein Bild in Form eines 2D-Arrays von Pixelwerten.

2. Faltungsschicht (Convolutional Layer): Diese Schicht führt die eigentliche Faltung (Convolution) durch, indem sie einen Filter (Kernel) über das Eingabebild verschiebt und Punktoperationen durchführt. Das Ergebnis ist eine Feature-Map, die lokale Merkmale des Bildes extrahiert. Jeder Filter kann unterschiedliche Merkmale wie Kanten, Ecken oder Texturen erkennen.

3. Aktivierungsschicht (Activation Layer): Nach jeder Faltungsschicht wird normalerweise eine Aktivierungsfunktion angewendet, um nichtlineare Eigenschaften des Netzwerks zu modellieren. Die häufig verwendete Aktivierungsfunktion ist die ReLU (Rectified Linear Unit), die alle negativen Werte auf Null setzt und positive Werte unverändert lässt.

4.Pooling-Schicht (Pooling Layer): Diese Schicht reduziert die räumliche Dimension der Feature-Maps, was die Berechnungen effizienter macht und die Gefahr von Überanpassung (Overfitting) verringert. Die gängigsten Pooling-Methoden sind Max-Pooling (wählt den maximalen Wert in einem bestimmten Bereich) und Average-Pooling (berechnet den Durchschnittswert in einem bestimmten Bereich).

5. Vollständig verbundene Schicht (Fully Connected Layer): Dies ist eine herkömmliche neuronale Netzwerkschicht, bei der jeder Neuron mit jedem Neuron der vorherigen Schicht verbunden ist. Sie kombiniert die extrahierten Merkmale, um das endgültige Ergebnis zu liefern, z.B. die Klassifikation des Bildes.

6. Ausgabeschicht (Output Layer): In der letzten Schicht wird eine Aktivierungsfunktion wie Softmax verwendet, um die Wahrscheinlichkeitsverteilung der möglichen Klassen zu berechnen.

2.1.6 Datenaugmentation

...

2.2 Synthetische Daten in der Bildklassifikation

...

2.2.1 Definition und Notwendigkeit synthetischer Daten

...

2.2.2 Vorteile und Herausforderungen

...

2.2.3 Variational Autoencoder

...

2.2.4 Generative Adversarial Networks

...

2.3 Stable Diffusion und DA-Fusion

...

2.3.1 Einführung in Diffusion-Modelle

...

2.3.2 Stable Diffusion

...

2.3.3 Datenaugmentation mit DA-Fusion

...

2.4 Contrastive Learning

...

2.4.1 Grundprinzipien des Contrastive Learning

...

2.4.2 Supervised Contrastive Learning

...

2.5 Integration von DA-Fusion und Supervised Contrastive Learning

...

2.5.1 Motivation für die Kombination

...

2.5.2 Potenzielle Vorteile und Herausforderungen

...

3 Methodisches Vorgehen

Dieses Kapitel behandelt die Methoden und Herangehensweisen, die in dieser Arbeit zur Beantwortung der Forschungsfragen angewendet wurden.

3.1 Forschungsfragen und Hypothesen

...

•Kann DA-Fusion für den Datensatz des Fraunhofer-IPK überzeugende synthetische Daten generieren? •Eignet sich DA-Fusion, um sowohl positiv- als auch negativ-Beispiele für das Contrastive Learning zu generieren? •Kann der beschriebene Ansatz eine bessere Generalisierung und Robustheit erzielen als ohne synthetische Daten bzw. als eine naive Verwendung der synthetischen Daten ohne Contrastive Learning? Starte mit einer Einleitung und dem ersten Punkt zu den Forschungsfragen

3.2 Datensatz

...

3.2.1 EIBA

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

Der Datensatz ist multimodal, d.h. er besteht aus verschiedenen Datenquellen, die unterschiedliche Informationen über die Gegenstände enthalten. Neben herkömmlichen RGB-Bildern

aus verschiedenen Perpektiven und weiteren Bilddaten, wie z.B. Objektmasken, gibt es auch Metadaten, etwa das Gewicht, oder Beschreibungen der Objekte in natürlicher Sprache durch verschiedene Stichwörter („CarComponent“, „cylinder“, „rusty“, ...).

...

3.2.2 Teildatensatz

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

Jede Klasse enthält mindestens ... Bilder. ...

3.2.3 Vorverarbeitung

...

3.3 Implementierung

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

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

3.3.1 DA-Fusion

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

entstehen so mehr oder weniger stark veränderte Bilder, die als synthetische Daten verwendet werden können.

...

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

3.3.2 Supervised Contrastive Learning

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

...

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

Metriken ...

3.4 Synthetische Datengenerierung mit DA-Fusion

...einmal normale Augmentationen, einmal OOD-Augmentationen ...Trial and Error, um Parameter zu bestimmen

3.5 Trainings- und Testdurchläufe mit Supervised Contrastive Learning

...

3.6 Evaluationsmethoden und Metriken

...

3.7 Analyse der Ergebnisse

...

4 Ergebnisse

...

4.1 Implementierung

...

4.1.1 Synthetische Datengenerierung

...Datensatz-Klasse ...Train-Prompts ...Aug-Prompts ...Aug-Maske

4.1.2 Supervised Contrastive Learning

...

4.2 Die generierten synthetischen Daten

...

4.2.1 Normale Augmentationen

...

4.2.2 OOD-Augmentationen

...Versuch 1: ... ...Versuch 2: ... ...

4.3 Trainings- und Testergebnisse

...

4.4 Vergleich der Performance mit und ohne normale Augmentationen

...

4.5 Vergleich der Performance mit und ohne OOD-Augmentationen

...

5 Diskussion

...

5.1 Interpretation der Ergebnisse

...

5.2 Bewertung der Eignung von DA-Fusion und Contrastive Learning

...

5.3 Stärken und Schwächen des Ansatzes

...

5.4 Auswirkungen auf die Generalisierungsfähigkeit und Robustheit der Modelle

...

6 Fazit

...

6.1 Zusammenfassung der wichtigsten Erkenntnisse

...

6.2 Beantwortung der Forschungsfragen

...

6.3 Ausblick

...

Literatur

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

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

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

Anhang

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

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

Eigenständigkeitserklärung

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

Viele zufällige Zahlen

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

Hamburg, 21. Dezember 1940