

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	1
1.3 Aufbau der Arbeit	1
2 Theoretische Grundlagen	2
2.1 Maschinelles Lernen	2
2.1.1 Überwachtes und unüberwachtes Lernen	3
2.1.2 Deep Learning	3
2.1.3 Neuronale Netze	4
2.1.4 Out-of-Distribution Daten	5
2.1.5 Datenaugmentation und Generalisierung	6
2.2 Synthetische Daten	6
2.2.1 Variational Autoencoder	6
2.2.2 Generative Adversarial Networks	7
2.2.3 Diffusion Models	8
2.3 Semantische Datenaugmentation mit DA-Fusion	9
2.4 Robuste Datenrepräsentation durch Contrastive Learning	11
2.4.1 Unsupervised Contrastive Learning	11
2.4.2 Supervised Contrastive Learning	12
2.5 Forschungslücke	12
2.5.1 Herausforderungen bei der Generierung synthetischer Daten	12
2.5.2 Synthetische Out-of-Distribution Daten als negativ-Beispiele im Contrastive Learning	12
2.5.3 Integration von DA-Fusion und Supervised Contrastive Learning	12
3 Methodisches Vorgehen	11
3.1 Forschungsfragen und Hypothesen	11

3.2	Datensatz	11
3.2.1	EIBA	12
3.2.2	Teildatensatz	12
3.2.3	Vorverarbeitung	12
3.3	Implementierung	12
3.3.1	DA-Fusion	13
3.3.2	Supervised Contrastive Learning	13
3.4	Synthetische Datengenerierung mit DA-Fusion	14
3.5	Trainings- und Testdurchläufe mit Supervised Contrastive Learning	14
3.6	Evaluationsmethoden und Metriken	14
4	Ergebnisse	15
4.1	Die generierten synthetischen Daten	15
4.1.1	In-Distribution	15
4.1.2	Near Out-of-Distribution	15
4.2	Trainings- und Testergebnisse mit Supervised Contrastive Learning	16
4.2.1	Contrastive Pre-Training	16
4.2.2	Lineare Klassifikation	16
4.3	Vergleich der Ergebnisse mit und ohne In-Distribution-Augmentationen	16
4.4	Vergleich der Ergebnisse mit und ohne Near Out-of-Distribution-Augmentationen als Hard Negatives	16
5	Diskussion	17
5.1	Eignung von DA-Fusion für die synthetische Datengenerierung	17
5.2	Wirksamkeit von Near Out-of-Distribution-Daten als Hard Negatives im Supervised Contrastive Learning	17
6	Fazit	18
6.1	Zusammenfassung der wichtigsten Erkenntnisse	18
6.2	Beantwortung der Forschungsfragen	18
6.3	Ausblick und potenzielle Weiterentwicklungen	18
	Literatur	19
	Anhang	20

Abbildungsverzeichnis

2.1	Beispiel eines einfachen künstlichen neuronalen Netzes. Quelle: (Zhou, 2021)	5
2.2	Überblick über die GAN-Struktur. Quelle: Google for Developers	8
2.3	Beispiel-Augmentationen aus DA-Fusion. Quelle: (Trabucco et al., 2023) . . .	10
4.1	Beispieltext	15

Tabellenverzeichnis

2 Theoretische Grundlagen

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

2.1 Maschinelles Lernen

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

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

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

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

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

2.1.1 Überwachtes und unüberwachtes Lernen

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

Beim überwachten Lernen wird das Modell mit einem vollständig annotierten Datensatz trainiert. Das heißt, jeder Datenpunkt ist mit einem Klassenlabel versehen, sodass Eingabe-Ausgabe-Paare entstehen. Das Ziel ist es, eine Funktion zu lernen, die Eingaben auf die entsprechenden Ausgaben 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. **<empty citation>**

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. nützliche Repräsentationen der Eingangsdaten zu erlernen. Zu den häufigsten Methoden des unüberwachten Lernens gehören Clustering- und Assoziationsalgorithmen. Ein Beispiel ist die Segmentierung von Kunden in verschiedene Gruppen basierend auf ihrem Kaufverhalten. **<empty citation>**

In der Praxis werden oft auch hybride Ansätze genutzt, wie das semi-überwachte Lernen, bei dem eine Kombination aus beschrifteten und unbeschrifteten Daten verwendet wird, oder das selbstüberwachte Lernen, bei dem das Modell eigenständig Teile der Daten zur Erzeugung von Überwachungssignalen verwendet, anstatt sich auf externe, von Menschen bereitgestellte Labels zu verlassen. **<empty citation>**

2.1.2 Deep Learning

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

versteht man eine tiefe, hierarchische Vernetzung dieser Konzepte, sodass komplexere Konzepte auf simpleren Konzepten aufbauen können. Visuell veranschaulicht entsteht ein Graph mit vielen Ebenen (engl. *deep layers*) <empty citation> Somit ist Deep Learning eine spezialisierte Unterkategorie des maschinellen Lernens, in der künstlichen neuronalen Netzen mit mehreren Schichten verwendet werden, um eine hierarchische Repräsentation von Daten zu ermöglichen. Jede Schicht transformiert die Eingabedaten in eine etwas abstraktere Darstellung.

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

2.1.3 Neuronale Netze

Während die rasante Entwicklung von Deep Learning vor allem in den vergangenen Jahren spürbar geworden ist, sind die zugrundeliegenden Algorithmen und Modelle schon seit Jahrzehnten bekannt <empty citation> Dabei bildet das künstliche neuronale Netz (KNN) die Grundlage der allermeisten Deep-Learning-Modelle. Es ist inspiriert von der Struktur und Funktionsweise des menschlichen Gehirns und besteht aus einer Vielzahl von miteinander verbundenen Knoten (Neuronen), die in Schichten organisiert sind. Die Struktur eines neuronalen Netzes besteht aus einer Eingabeschicht, einer oder mehreren versteckten Schichten (engl. *hidden layers*) und einer Ausgabeschicht.

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

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

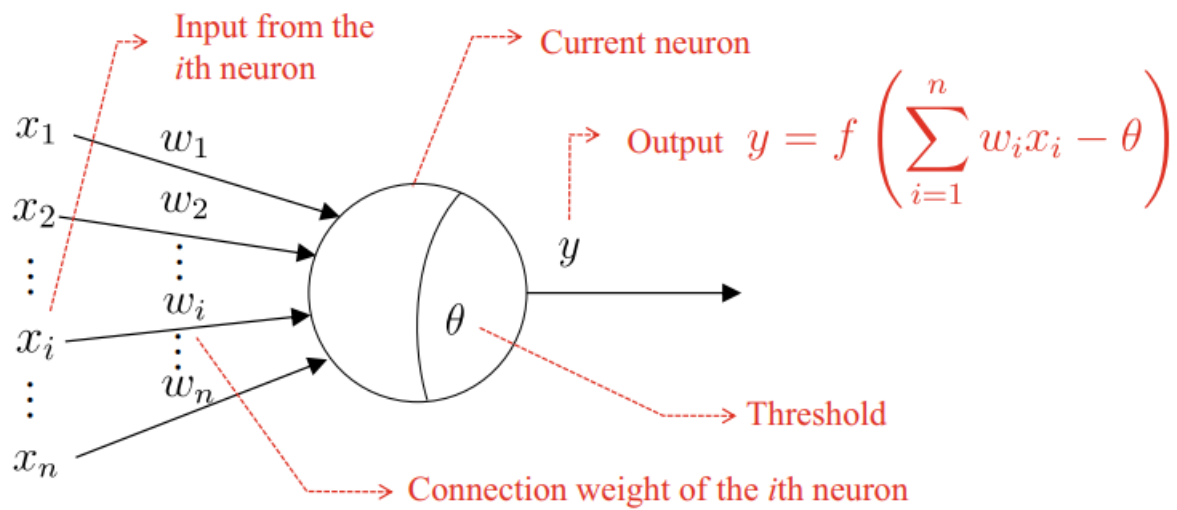


Fig. 5.1 The M-P neuron model

Abbildung 2.1: Beispiel eines einfachen künstlichen neuronalen Netzes. Quelle: (Zhou, 2021)

...

2.1.4 Out-of-Distribution Daten

Wenn ein KI-Modell mit Daten konfrontiert wird, die außerhalb des Bereichs liegen, den es während des Trainings gesehen hat, spricht man von Out-of-Distribution (OOD) Daten. Es handelt sich also um Datenpunkte oder Muster, die sich signifikant von den Trainingsdaten unterscheiden. Dies kann zu Problemen führen, da das Modell möglicherweise nicht in der Lage ist, angemessene Vorhersagen oder Entscheidungen für diese Daten zu treffen. Stattdessen werden falsche Vorhersagen mit übermäßigem Vertrauen getroffen.

Die Erkennung von OOD-Daten ist ein wichtiges Forschungsgebiet im maschinellen Lernen, da sie dazu beitragen kann, die Zuverlässigkeit und Sicherheit von KI-Systemen zu verbessern. Idealerweise sollte ein neuronales Netz höhere Softmax-Wahrscheinlichkeiten für In-Distribution-Daten und niedrigere Wahrscheinlichkeiten für OOD-Daten ausgeben. Durch Festlegen eines Schwellenwerts für diese Wahrscheinlichkeiten können Instanzen unterhalb des Schwellenwerts frühzeitig als OOD-Instanzen erkannt und entsprechend behandelt werden. In der Praxis kommt dieser Ansatz jedoch oft an seine Grenzen, da die Softmax-Wahrscheinlichkeiten nicht immer zuverlässig sind und das Modell auch für OOD-Daten hohe

Wahrscheinlichkeiten ausgeben kann. Daher werden alternative Ansätze verwendet, wie etwa das Training eines binären Klassifikationsmodells zur Unterscheidung von In-Distribution und OOD-Daten.

2.1.5 Datenaugmentation und Generalisierung

Datenaugmentation ist ein wichtiger Schritt im Training von neuronalen Netzen, insbesondere bei begrenzten Datensätzen. Sie bezieht sich auf die künstliche Erweiterung des Trainingsdatensatzes durch Anwenden von Transformationen auf die vorhandenen Daten. Diese Transformationen können z.B. Rotation, Skalierung, Verschiebung, Spiegelung, Helligkeitsanpassung oder Rauschen sein. Das Ziel der Datenaugmentation ist es, das Modell robuster gegenüber Variationen in den Eingabedaten zu machen und die Generalisierungsfähigkeit zu verbessern.

2.2 Synthetische Daten

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

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

...

2.2.1 Variational Autoencoder

Ein Autoencoder ist eine spezielle Art von KI-Modell, das darauf ausgelegt ist, Daten effizient zu komprimieren und dann wieder zu rekonstruieren. Es besteht aus zwei Hauptkomponenten: (Foster, 2020)

- einem **Encoder**-Netzwerk, das hochdimensionale Eingabedaten in einem niederdimensionalen Darstellungsvektor komprimiert, und
- einem **Decoder**-Netzwerk, das einen gegebenen Darstellungsvektor zurück in den ursprünglichen hochdimensionalen Raum umwandelt

Der Darstellungsvektor ist eine Kompression des Originalbilds in einen niedriger dimensionalen latenten Raum, wodurch es sich beim Autoencoder um eine Form des *Representation Learning* handelt.

Das Training eines Autoencoders erfolgt durch Minimierung des Rekonstruktionsfehlers, der die Differenz zwischen den ursprünglichen Eingabedaten und den rekonstruierten Ausgaben beschreibt. Eine gängige Verlustfunktion hierfür ist der *Mean Squared Error* (MSE):

$$Loss = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (2.1)$$

Ein besonders interessantes Versprechen des Autoencoders ist, dass man theoretisch durch die Wahl eines beliebigen Punkts im latenten Raum neue Bilder erzeugen kann, indem man diesen Punkt durch den Decoder schickt, da der Decoder gelernt hat, wie man Punkte im latenten Raum in realistische Bilder umwandelt. (Foster, 2020) In der herkömmlichen Form hat der Autoencoder in Bezug auf diese Aufgabe allerdings einige Schwachstellen.

Der **Variational Autoencoder** (VAE) adressiert diese Schwachstellen und verwendet probabilistische Methoden, um die Datenverteilung im latenten Raum zu modellieren; Anstatt einen einzelnen, festen Punkt im latenten Raum für jede Eingabe zu lernen, wird eine Verteilung gelernt, aus der die latenten Variablen für jede Eingabe stammen. Dadurch entsteht ein strukturierter und kontinuierlicher latenter Raum, der es ermöglicht, neue, realistische Daten zu generieren.

...

2.2.2 Generative Adversarial Networks

Ein Generative Adversarial Network (GAN) ist ein KI-Modell, das in <empty citation> vorgestellt wurde. GANs bestehen aus zwei neuronalen Netzwerken, die gegeneinander antreten, um realistische synthetische Daten zu erzeugen. Diese Technologie hat sich als äußerst mächtig in der Bild- und Datengenerierung erwiesen.

Die Architektur eines GANs besteht aus zwei Hauptkomponenten:

- **Generator:** Das generative Netzwerk nimmt Zufallsrauschen als Eingabe und erzeugt daraus Daten, die möglichst realistisch wirken sollen. Der Generator versucht, die wahre Datenverteilung zu imitieren und realistische Beispiele zu erstellen.

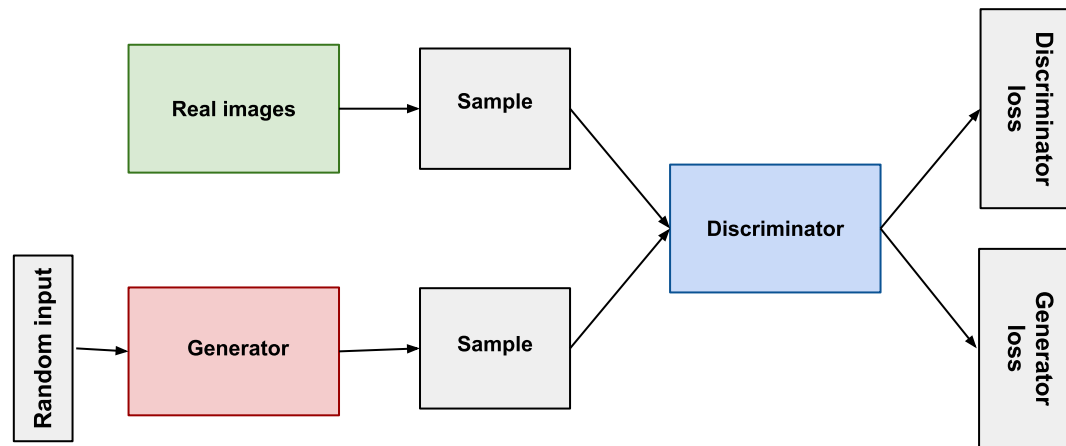


Abbildung 2.2: Überblick über die GAN-Struktur. Quelle: Google for Developers

- **Diskriminator:** Das diskriminative Netzwerk erhält sowohl echte Daten aus dem Trainingsdatensatz als auch die vom Generator erzeugten Daten. Seine Aufgabe ist es, zwischen echten und künstlichen Daten zu unterscheiden. Der Diskriminator gibt eine Wahrscheinlichkeit aus, dass die Eingabedaten echt sind.

Der Trainingsprozess eines GANs ist als minimax-Spiel zwischen dem Generator und dem Diskriminator formuliert:

1. **Diskriminator-Training:** Der Diskriminator wird trainiert, um echte Daten von generierten Daten zu unterscheiden. Dies geschieht durch eine binäre Klassifikation („echt“ oder „synthetisch“). Der Diskriminator passt seine Gewichte an, um die Unterscheidung zu verbessern.
2. **Generator-Training:** Der Generator wird trainiert, um den Diskriminator zu täuschen. Dies geschieht, indem der Generator seine erzeugten Daten durch den Diskriminator laufen lässt und die Rückmeldung (Gradienten) des Diskriminators verwendet, um seine eigenen Parameter zu optimieren. Ziel ist es, den Diskriminator zu überlisten, sodass er die generierten Daten als echt klassifiziert.

...

2.2.3 Diffusion Models

In den letzten Jahren haben sogenannte Diffusion Models für einen enormen Sprung in der Bildgenerierung, insbesondere der Text-to-Image (T2I) Generierung, gesorgt. Das Konzept der Diffusion soll hier deshalb einmal genauer erläutert werden.

Unter Diffusion versteht man den Prozess der langsamen Vermischung von Partikeln oder Informationen über die Zeit. So beschreibt die Diffusionsgleichung in der Physik die zeitliche Entwicklung der Dichte von Teilchen, die sich zufällig bewegen. Im maschinellen Lernen fand das Konzept erstmals in **<empty citation>** Anwendung. Es entstand eine neue Klasse von generativen Deep Learning-Modellen, die Diffusion Models, welche im Trainingsprozess schrittweise die Struktur der Eingabedaten durch das Hinzufügen von Rauschen auflösen und anschließend darauf trainiert werden, das ursprüngliche Bild aus dem verrauschten Bild zu rekonstruieren.

Das Training von Diffusion Models teilt sich somit in zwei Phasen auf, die Vorwärts- und die Rückwärtsdiffusion. Beide Phasen ...

$$q(x^{(0...T)}) = q(x^{(0)}) \prod_{t=1}^T q(x^{(t)}|x^{(t-1)}) \quad (2.2)$$

In der Rückwärtsdiffusion geht das Bild durch ein Modell, welches selbst nur Rauschen generiert. Es ist die Vorhersage darüber, welches Rauschen vom Eingang entfernt werden soll, um das gewünschte entrauschte Bild zu erhalten.

$$p(x^{(0...T)}) = p(x^{(T)}) \prod_{t=1}^T p(x^{(t-1)}|x^{(t)}) \quad (2.3)$$

Anders als bei GANs gibt es keine direkten adversarialen Optimierungsmechanismen, die zu einem Ungleichgewicht führen können. Es wird stattdessen explizit die Wahrscheinlichkeitsdichte zwischen den realen Daten und den erzeugten Daten minimiert, was zu einem robusteren und stabileren Trainingsprozess führt.

Eine entscheidende Weiterentwicklung der Diffusion Models war die Text-Konditionierung des generativen Prozesses. In **<empty citation>** wurde DALL-E vorgestellt, ein Modell, das es ermöglicht, Bilder aus Textbeschreibungen zu generieren. DALL-E ...

...

2.3 Semantische Datenaugmentation mit DA-Fusion

In (Trabucco et al., 2023) wird DA-Fusion, eine flexible, auf Stable Diffusion basierende Methode zur Datenaugmentation vorgestellt, die für diese Arbeit von besonderem Interesse ist. Der traditionelle Ansatz der Datenaugmentation, wie in 2.1.5 beschrieben, hat sich als effektiv erwiesen, um die Generalisierungsfähigkeit von Modellen zu verbessern. Allerdings

erfordert dieser Ansatz auch eine gute Intuition in Bezug auf den verwendeten Datensatz, um zu vermeiden, dass Transformationen gewählt werden, durch die Informationen verloren gehen, die für die Aufgabe des zu trainierenden Modells wichtig sind. Wenn beispielsweise Farbinformationen für die Klassifizierung von Blumen wichtig sind, könnte die Datenaugmentation durch zufällige Farbänderungen die Leistung des Modells verschlechtern. Ein weiteres Beispiel sind Objekte, die klein im Bild sind und durch zufällige Ausschnitte des Bildes aus der Sicht des Modells verschwinden können. DA-Fusion hingegen nutzt das Wissen eines vortrainierten Diffusion Modells, um den Bildinhalt semantisch zu verstehen und automatisch neue, realistische Variationen zu generieren.

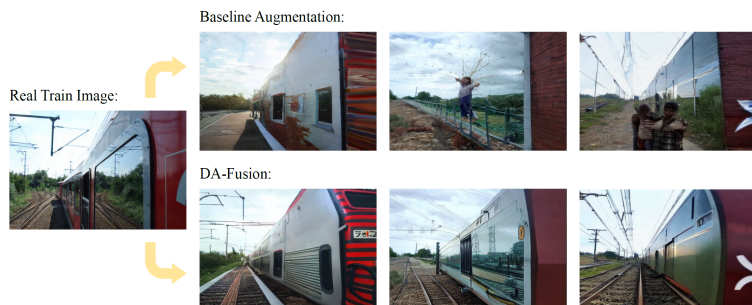


Abbildung 2.3: Beispiel-Augmentationen aus DA-Fusion. Quelle: (Trabucco et al., 2023)

Es wird zunächst die Methode Textual Inversion (**<empty citation>**) angewendet, um ein vortrainiertes Stable Diffusion-Modell auf den gegebenen Datensatz feinabzustimmen. Dazu wird für jedes neue Konzept bzw. für jede neue Klasse ein Pseudo-Token y in das Modell integriert, der unter Verwendung von Trainings-Prompts wie „a photo of a $\langle y \rangle$ “ und den zugehörigen Bilddaten trainiert wird.

Anschließend können aus den echten Bildern neue Augmentationen generiert werden, indem den Bildern eine geringe Menge an Rauschen hinzugefügt wird, welches dann wieder durch das feinabgestimmte Modell entfernt werden soll. Hier kommen wieder die selben Text-Prompts zum Einsatz. Auf diese Weise bleibt die grundlegende Struktur der Bilder erhalten, während gleichzeitig neue, semantisch sinnvolle Variationen und Details erzeugt werden.

Ein Vorteil von DA-Fusion ist die Möglichkeit, den Grad der Augmentation durch die Wahl des Insertion Timesteps zu steuern. Der Insertion Timestep bestimmt, wie weit in den Diffusionsprozess das Bild eingefügt wird und wie stark es dafür vorher verrauscht werden muss. Ein niedriger Timestep führt zu stärkeren Augmentationen, während ein hoher Timestep subtilere Variationen erzeugt.

2.4 Robuste Datenrepräsentation durch Contrastive Learning

Neben den vorgestellten Methoden zur Vervielfältigung der Trainingsdaten soll nun auch das Contrastive Learning als effektive Methode zur Verbesserung der Generalisierungsfähigkeit und Robustheit von Modellen vorgestellt werden. Contrastive Learning kommt ursprünglich aus dem unüberwachten Lernen und zielt darauf ab, ähnliche Beispiele im Datensatz zu gruppieren und unähnliche Beispiele voneinander zu trennen. Durch die Kontrastierung der Daten wird eine Art von Supervision erzeugt, die es dem Modell ermöglicht, nützliche Repräsentationen der Eingabedaten zu lernen.

Mittlerweile gibt es vermehrt Ansätze, Contrastive Learning auch im überwachten Setting anzuwenden, um die Repräsentationen von Daten zu verbessern. Während das Modell im unüberwachten Setting lernt, zwischen einzelnen Instanzen zu unterscheiden, werden im überwachten Setting die Klassenzugehörigkeit der Beispiele berücksichtigt.

In den folgenden Abschnitten werden im Detail sowohl unüberwachte als auch überwachte Varianten des Contrastive Learning vorgestellt, die in den letzten Jahren vielversprechende Ergebnisse erzielt haben.

2.4.1 Unsupervised Contrastive Learning

Da die Annotation von Daten sehr aufwendig sein kann, insbesondere in Domänen, in denen Expertenwissen erforderlich ist, hat sich das unüberwachte Lernen als vielversprechende Alternative erwiesen. Contrastive Learning ist ein Beispiel für ein unüberwachtes Lernverfahren, das es ermöglicht, nützliche Repräsentationen von Daten zu lernen, ohne auf externe Labels angewiesen zu sein.

2.4.2 Supervised Contrastive Learning

2.5 Forschungslücke

2.5.1 Herausforderungen bei der Generierung synthetischer Daten

2.5.2 Synthetische Out-of-Distribution Daten als negativ-Beispiele im Contrastive Learning

...

2.5.3 Integration von DA-Fusion und Supervised Contrastive Learning

...