HAW HAMBURG

Masterarbeit

Ilja Grauer

Interpretierbarkeit neuronaler Klassifikatoren mit Hilfe von Variational Autoencodern

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Ilja Grauer

Interpretierbarkeit neuronaler Klassifikatoren mit Hilfe von Variational Autoencodern

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Andreas Meisel
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 23. Mai 2022

Ilja Grauer

Thema der Arbeit

Interpretierbarkeit neuronaler Klassifikatoren mit Hilfe von Variational Autoencodern

Stichworte

XAI, Variational Autoencoder, Interpretierbarkeit, Bildähnlichkeit, latenter Raum, ncc

Kurzzusammenfassung

Diese Arbeit kombiniert einen neuronalen Klassifikator mit einem Variational Autoencoder, um die Interpretierbarkeit ersteren zu verbessern. Die Repräsentation der Trainingsdaten im latenten Raum wird dafür genutzt die Ähnlichkeit zu einem Eingangsbild zu ermitteln. Normalized Cross Correlation wird als zusätzliche Vergleichsfunktion angewandt. Das Modell wird in verschiedenen Experimenten evaluiert und hinsichtlich seiner Relevanz diskutiert. Abschließend werden die offenen Probleme beschrieben und mögliche Lösungsansätze für zukünftige Arbeiten vorgeschlagen.

Ilja Grauer

Title of Thesis

Interpretability of neural classificators with the help of variational autoencoders

Keywords

xai, variational autoencoder, interpretability, image similarity, latent space, ncc

Abstract

This thesis combines a neural classifier with a variational autoencoder to improve the formers interpretability. The disentangled representation of the training data in the latent space is used to calculate the similarity to a sample image. Normalized cross correlation is used as a second similarity measurement. The model is evaluated in different experiments and discussed regarding its relevance. Finally, the unsolved problems are described and possible solutions for future works are suggested.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Was noch vor wenigen Jahren als *Science-Fiction* galt, ist heutzutage ein allgegenwärtiger Bestandteil des modernen Lebens: Künstliche Intelligenz (KI). Sei es der Sprachassistent in der Küche, der während des Kochens diverse Timer verwaltet und nebenbei über die Nachrichten berichtet, oder die Streaming-Plattform, die den nächsten Film vorschlägt. Die Berührungspunkte zwischen Mensch und KI sind präsent.

Auch sicherheitskritische Bereiche setzten vermehrt auf KI. Autonomes Fahren oder Krebszellenerkennung in Röntgenbildern sind nur einige Beispiele. Studien wie [62] vergleichen die Fähigkeiten entsprechender medizinischer Systeme mit menschlichen Ärzten. Andere [60] schlagen bereits neue interdisziplinäre Arbeitsplätze wie „digitale Ärzte“ vor, die zwischen KI und Mensch interagieren.

Auf der anderen Seite betonen diese Arbeiten aber auch das noch fehlende Vertrauen in solche Systeme. Meldungen über *Microsofts* Chatbot *Tay*[1], der rassistische Äußerungen von sich gegeben hat, bestärken die skeptische Sichtweise gegenüber KI in der Gesellschaft. Auch die Europäische Union hat Bedenken gegenüber Sicherheit, Ethik und Datenschutz geäußert, sieht aber auch gleichzeitig die zukunftsweisenden Möglichkeiten. Aus diesem Grund wird aktuell an einem gesetzlichen Rahmen [42] [27] für KI gearbeitet, der diese Punkte regulieren soll. Solche Richtlinien sollen dabei insbesondere die Entwicklung von interpretierbaren Systemen vorantreiben. Dadurch sollen sogenannte „Blackbox“-Modelle, die kaum in ihrer Funktionsweise verstanden sind, zu „Glassbox“-Modellen werden [61], die für Menschen nachvollziehbar sind.

Die Komplexität dieser Systeme ergibt sich aus den Millionen von Parametern und der Menge der genutzten Daten [46]. Gerade letztere führen oft zu unvorhergesehenem und unbeabsichtigtem Verhalten einer KI.

[1] https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter

1.1 Motivation

Zur Interpretation genannter „Blackbox“-Modelle werden meistens visuelle Methoden genutzt, um die Bedeutung bestimmter Neuronen, Bildpunkte oder -bereiche darzustellen (siehe Kapitel 3.1 für Details). In dieser Arbeit soll ein anderer Ansatz verfolgt werden und in erster Linie der Bezug zu den Trainingsdaten untersucht werden. Diese sind in der Regel maßgebend für die Performance eines Modells und eine nicht zu vernachlässigende Komponente. Unerwartete Ergebnisse eines Klassifikators entstehen oft aus problematischen Datensätzen, die im Nachhinein - aufgrund der großen Datenmenge - oft nicht mehr nachvollzogen werden können.

Um mit diesen Datenmengen arbeiten zu können, müssen diese in einen Raum mit weniger Dimensionen komprimiert werden. Ein Variational Autoencoder (VAE) [40] [41] soll hierfür genutzt werden, da dieser bereits die inhärente Eigenschaft besitzt, Trainingsdaten gleichmäßig im latenten Raum zu verteilen (siehe Kapitel 3.2 für Details). Dadurch ist dieser Netztyp dafür prädestiniert, globale Zusammenhänge der Trainingsdaten zu untersuchen, wie auch in verwandten Arbeiten (siehe Kapitel 2 für Details) gezeigt.

1.2 Ziel

Aus diesen Gedanken heraus soll ein Klassifikator mit einem VAE kombiniert und der gemeinsame latente Raum als Grundlage für eine interpretierbare Klassifizierung genutzt werden. Dabei sollen für ein neues Eingangsbild entsprechende Bildnachbarn dargestellt werden. Zusätzlich sollen sowohl diese Nachbarn als auch die Rekonstruktion hinsichtlich der Bildähnlichkeit mit *Normalized Cross Correlation* (NCC) untersucht werden, um eine weitere Metrik der Bildähnlichkeit hinzuzufügen. Diese ist im Gegensatz zur Berechnung der Bildnachbarn unabhängig vom Modell. Somit lässt sich das Ziel und die Fragestellung der Arbeit folgendermaßen formulieren:

Kann mit Hilfe eines VAE die Entscheidung eines Klassifikators interpretiert werden?

Aus dieser Frage resultierend sollen folgende Teilfragen experimentell beantwortet werden:

F1: Sind die Ergebnisse interpretierbar für richtige als auch falsche Klassifizierungen?

F2: Wie wird der Klassifikator vom VAE beeinträchtigt?

F3: Welchen Einfluss hat ein großer Dimensionsraum?

F4: Welchen Einfluss hat die Regulierung der Verlustfunktionen?

F5: Wie wirkt sich die Varianz auf die Ergebnisse aus?

1.3 Aufbau der Arbeit

Im nachfolgenden Kapitel 2 werden verwandte Arbeiten vorgestellt, die ebenfalls einen VAE zwecks der Interpretierbarkeit nutzen. Daraufhin wird diese darin eingeordnet und abgegrenzt.

In Kapitel 3 werden die Grundlagen vorgestellt, die für die Umsetzung der Arbeit verwendet werden.

Kapitel 4 stellt die Modelle und Methoden vor, die zur Erfüllung der Zielsetzung aus 1.2 verwendet werden. Es wird die Architektur des Modells beschrieben und der Begriff der *Interpretationskanäle* definiert. Außerdem werden der verwendete *Fashion-MNIST*-Datensatz sowie die Evaluationsbilder vorgestellt.

In Kapitel 5 werden die verschiedene Experimente und die dazugehörigen Ergebnisse vorgestellt, mit denen das Modell hinsichtlich der Interpretierbarkeit getestet wird. Diese beziehen sich auf die in Abschnitt 1.2 erwähnten Fragestellungen.

Kapitel 6 bewertet und diskutiert die zuvor ermittelten Ergebnisse. Es wird erörtert, wo die Stärken und Schwächen der Methode liegen und wie man bestimmte Punkte optimieren kann. Abschließend wird beantwortet, inwieweit die Ziele der Arbeit erfüllt wurden.

Im letzten Kapitel 7 werden die wesentlichen Inhalte zusammengefasst und Ausblicke für zukünftige Forschungen gegeben.

2 VAEs als Interpretationswerkzeug

VAEs werden aufgrund ihrer Eigenschaften oft als Werkzeug genutzt, um interpretierbare Aussagen eines neuronalen Netzes zu gewinnen. Einige dieser Arbeiten versuchen einen VAE mit einem Klassifikator [15] [53] [55] [20] oder einer anderen Netzart [73] [77] [23] zu kombinieren. Wieder andere betrachten gezielt den latenten Raum und die genutzten Trainingsdaten [31] [75] [81] [16] [30] oder versuchen eine bessere Verteilung im latenten Raum zu erreichen [82] [17] [49] [44]. Auch alleinstehende VAEs werden zwecks Interpretierbarkeit untersucht und verwendet [67] [72] [43] [64] [54].

Nachfolgend wird eine aktuelle Auswahl der zuvor genannten Arbeiten vorgestellt, die insbesondere mit dem Ziel dieser Arbeit korrelieren. Abschließend wird diese Arbeit abgegrenzt und der Mehrwert formuliert.

In der Arbeit [15] von Cetin et al. wurde am Beispiel von medizinischen Bildern ein inhärent interpretierbares Modell, *Attri-VAE*, entworfen. Dieses wird genutzt, um die Repräsentation im latenten Raum für Interpretationen und Klassifizierung nutzen zu können. Dazu kombinieren die Autoren einen β-VAE mit einem Klassifikator als auch Attribut-basierten Methoden, um eben diese Attribute entzerrt und verständlich in einer zugewiesenen Dimension des latenten Raums darstellen zu können. Das Modell nutzt diesen latenten Raum vor allem für die Interpolation zwischen den genannten Attributen. Zusätzlich wurde *Grad-CAM*[1] [66] verwendet, um visuelle Unterstützung bei der Interpretation zu erhalten.

Nguyen und Martínez kombinieren in ihrer Arbeit [53] ebenfalls einen VAE mit einen Klassifikator. Dabei ist das Ziel die Invarianzen des Klassifikators zu erlernen. Invarianzen sind dabei Faktoren, die sich nicht oder kaum verändern. Das sind beispielsweise Features einer Klasse, die besonders wichtig für die Klassifizierung sind und sich über alle Trainingsdaten hinweg wiederholen. Im Umkehrschluss können dadurch die weniger wichtigen Faktoren ermittelt und das Netz verbessert werden. Im Gegensatz zum β-VAE

[1] Eine Methode, um relevante Bereiche in einem Bild über eine *Heatmap* zu visualisieren.

soll der Parameter β in ihrer Arbeit die Rekonstruktion und die Klassifizierung betreffen. Beide Ausgaben sind dabei für die Interpretierbarkeit wichtig. Zum einen soll die Erkennungsgenauigkeit des Klassifikators gleich gut bleiben, zum anderen wird eine gute Rekonstruktion zur Feststellung der Invarianzen benötigt. Im Versuch werden dann die Rekonstruktionen wieder an das Netz gegeben und eine gleiche Klassifizierung ähnlich zum Originalbild erwartet. Um die Interpretierbarkeit zu testen, werden zufällig Dimensionen des latenten Raums gewählt und auf diesen das Bild interpoliert. Diese generierten Nachbarn zeigen bereits Invarianzen (in dem Beispiel der Autoren der mittlere Teil der Zahl 3), die zwischen den Nachbarn nicht verändert werden. Dadurch lässt sich bereits feststellen, dass der obere und untere Teil der 3 weniger relevant für das Modell ist. Weiter wenden die Autoren Gradienten-basierte Methoden an. Die Besonderheit ist, dass hierbei das Originalbild mit den benachbarten Rekonstruktionen verglichen wird und die wichtigen (invarianten) Stellen visuell hervorgehoben werden. Einen großen Vorteil in der Verwendung eines VAE sehen die Autoren in Hinblick auf die Generierung von Ausreißern, die nicht in den Trainingsdaten vorkommen. Dadurch könnten auch globale Invarianzen erkannt werden, die zum Gesamtverständnis des Modells beitragen.

Tran et al. haben in ihrer Arbeit [72] anhand von vier Experimenten untersucht, inwieweit die Entzerrung im latenten Raum eines VAEs zur Interpretierbarkeit der Features beiträgt. Insbesondere der Versuch, die Qualität der Rekonstruktion unter anderem mit *Mean Square Error* zu messen, um dadurch Anomalien zu erkennen, ist ein ähnlicher Ansatz wie in dieser Arbeit mit NCC.

Ji et al. stellen in ihrer Arbeit [39] einen *Supervised VAE* (SVAE) vor, welcher für einen Agrarroboter zur Anomalieerkennung genutzt wird. Der Roboter soll Hindernisse in Feldern verlässlich erkennen und anschließend umgehen können. Um den hochdimensionalen Zustandsraum zu verarbeiten, setzen die Autoren eine Kombination aus VAE und Klassifikator ein, die simultan trainiert werden. Ihr Modell scheint dabei in ihren präsentierten Ergebnissen die entsprechenden Hindernisse gut klassifizieren zu können.

Einen interessanten Ansatz verfolgen Zhu et al. in [82]. Es wird zwar ein klassischer Autoencoder (AE) verwendet, allerdings vergleichen die Autoren ihr Modell fortwährend mit den Eigenschaften eines VAE und warum sie diesen nicht gewählt haben. CSAE steht für *classification supervised autoencoder* und nutzt ähnlich zu dieser Arbeit eine Kombination aus AE und Klassifikator. Dabei legen sie einen Fokus auf gleich verteilte Zentroiden, also den Zentren von Clustern, um die Klassen bereits vor dem Training im latenten Raum einzugrenzen. Dadurch soll das Netz sich auf das Lernen der Unterschiede

innerhalb der Klasse (*intra-class*) fokussieren anstatt den Klassen an sich (*inter-class*). Bei der Kombinationen zwischen VAE und Klassifikator oder ähnlicher Ansätze kritisieren die Autoren dabei, dass eine zusätzliche Netzarchitektur eingefügt wird, während in ihrem Ansatz direkt aus den Zentroiden des latenten Raumes klassifiziert werden kann. Des Weiteren wurde das *Sampling* mit einer ähnlichen *Noise*-Funktion ersetzt, um den latenten Raum künstlich zu variieren. Die Ergebnisse ihrer Arbeit sind unter anderem eine bessere Rekonstruktion, Generalisierung bei neuen Datensätze sowie eine marginal bessere Genauigkeit bei der Klassifizierung.

Gat et al. nutzen eine ergänzende VAE-Architektur in ihrer Arbeit [31], um das Bias[2] in Datensätzen zu erkennen und in einer interpretierbaren Weise darzustellen. Die Idee der Intervention haben die Autoren von den entsprechenden XAI-Methoden mit Pertubation. Dies wollen sie mit dem latenten Raum eines VAEs erreichen, da dieser deutlich weniger Dimensionen beinhaltet und durch Interpolation in dem Raum die *Intervention* simuliert wird. Konkret wird dabei zunächst ein VAE zu einer Klasse z.B. „attractive“ trainiert, der eine entsprechende Rekonstruktion zur Klasse generiert. Zusätzlich wird im dadurch entstandenen latenten Raum so oft durch die Koordinaten interpoliert bis die zweite Rekonstruktion das beste Ergebnis für die Klasse bekommt. Durch den kausalen Zusammenhang zwischen Rekonstruktion und der Klasse lassen sich dann Rückschlüsse über den verwendeten Datensatz ziehen.

Die vorgestellten Arbeiten zeigen in der Regel sehr genaue Implementierungen für eine bestimmte Problemstellung. Im Gegensatz dazu soll sich diese Arbeit durch folgende Punkte abgrenzen:

(a) Die Kombination eines VAEs mit einem Klassifikator soll im Kontext von **XAI** untersucht und eingeordnet werden.

(b) Das Zusammenspiel mehrerer Ausgaben (**Interpretationskanäle**) soll in einer gesamtheitlichen Darstellung betrachtet und bewertet werden. Das umfasst Klassifikation, Rekonstruktion, Bildnachbarschaft im latenten Raum sowie Bildähnlichkeit.

(c) Es wird eine **zweite Bildähnlichkeit mit Normalized Cross Correlation (NCC)** neben der euklidischen Distanz eingeführt. Diese soll unabhängig vom Modell eine weitere Interpretationshilfe liefern.

[2] Durch falsche Untersuchungsmethoden (z. B. Suggestivfragen) verursachte Verzerrung des Ergebnisses einer Repräsentativerhebung. Quelle: https://www.duden.de/node/22214/revision/612121

3 Grundlagen

3.1 XAI

Das Forschungsgebiet von e**X**plainable **AI** (XAI) befasst sich damit, künstliche neuronale Netze (KNN) für den Menschen verständlicher zu gestalten. Über die letzten Jahre wurden sehr viele Publikationen veröffentlicht, die entweder XAI im Allgemeinen erklären, eine Übersicht über Methoden geben oder Begriffe und Taxonomien definieren [65] [59] [26] [21] [46] [71] [76] [8] [1] [24] [47]. Wieder andere versuchen gesetzliche Normen zu definieren, wie in den USA das Institut für technologische Standards NIST[1] [11] [58] oder in der eingangs genannten EU durch die Europäische Kommission [42].

Trotz der Vielzahl an begriffsdefinierenden Arbeiten, wie der von Doshi-Velez und Kim [24], scheint es immer noch schwierig zu sein einen gemeinsamen Standard zu finden, auch wenn es mittlerweile viele Überschneidungen gibt. *„The abundance of existing literature on structuring the field and methods of XAI has by now reached an overwhelming level for beginners and practitioners.“* benennen es Schwalbe und Finzel in [65]. Nachfolgend sollen die wesentlichen Punkte der Forschung kurz vorgestellt und diese Arbeit darin eingeordnet werden.

3.1.1 Definition der Interpretierbarkeit

Auch wenn die Begriffe *explainability* und *interpretability* von der Bedeutung und Anwendung her ähnlich sind, hat sich in der Fachliteratur eine Differenzierung herausgestellt [65]:

[1] National Institute of Standards and Technology

- **Explainability** ist ein allgemeinerer Begriff und beschreibt, dass (a) der Kontext, (b) das Modell selbst oder (c) die Entscheidungsgrundlage von *Machine Learning*-Modellen (ML) von Menschen verstanden werden kann. Es geht dabei mehr um die Ausgabe im Nachhinein.
- **Interpretability** ist im Gegensatz ein genauerer Begriff, der insbesondere den technischen Grund einer Entscheidung erklären soll. Dabei nutzt zumeist ein Entwickler eines solchen Systems dieses Wissen, um das KNN zu verbessern.

Doshi-Velez und Kim [24] haben den Begriff der Interpretierbarkeit folgendermaßen definiert:

Interpretierbarkeit ist die Fähigkeit [einen Sachverhalt] für den Menschen verständlich erklären oder präsentieren zu können.

3.1.2 Zielgruppen und Gründe für den Einsatz

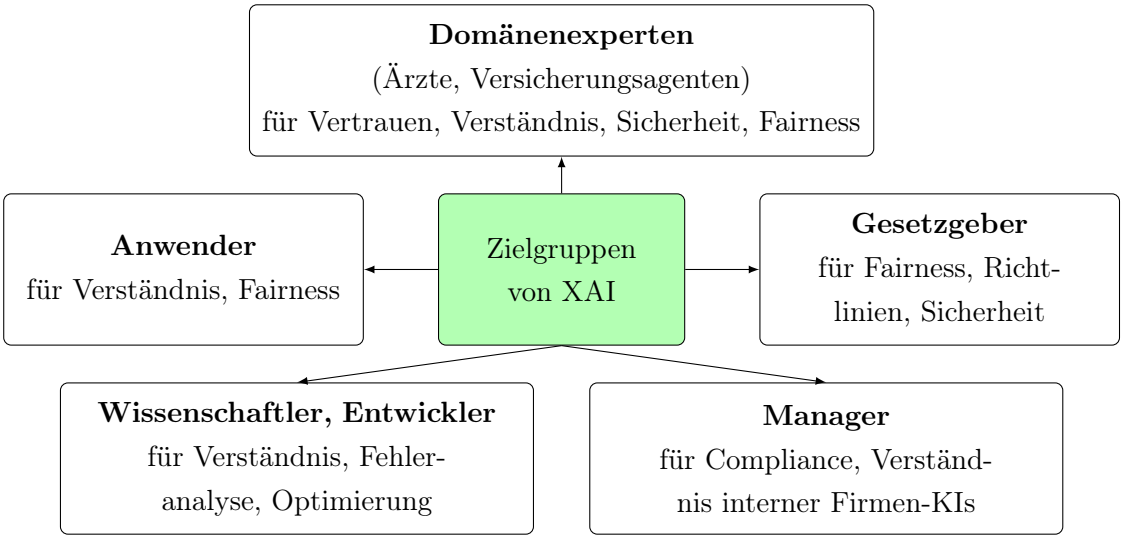


Abbildung 3.1: Zielgruppen interpretierbarer Modelle (Quelle: In Anlehnung an Arrieta et al. [8])

Da Interpretierbarkeit laut Definition nicht ohne Menschen funktionieren kann und die meisten Systeme automatisiert ohne Aufsicht arbeiten, stellt sich die Frage, wo XAI sinnvoll ist. Interpretierbare Modelle sind dort sinnvoll, bei denen die formale Problembeschreibung „unvollständig“ ist [24]. Damit sind Anwendungen oder Systeme gemeint,

wo noch kein ausreichendes Wissen existiert, damit diese alleine operieren können oder sollten. Diese Systeme haben Schnittstellen mit unterschiedlichen Personengruppen, die jeweils eigene Anforderung mit sich bringen. Die Kombination aus den Fragen „**Wer**?“ und „**Warum**?“ [8] beschreibt die Zielgruppen sowie Einsatzgebiete.

Abbildung 3.1 zeigt die Zielgruppen von XAI. Die jeweiligen Gründe sind nachfolgend erklärt [8]:

- **Wissenschaftliches Verständnis:** Aus der Neugierde des Menschen heraus kann XAI dafür genutzt werden komplexe Sachverhalte, insbesondere im ML-Kontext, offen zu legen. Aufgrund der heutzutage immensen Datenmengen ist ML gerade für *Data Scientists* interessant.

- **Sicherheit:** Die in der Einleitung erwähnten Gebiete des autonomen Fahrens und der Medizin sind Beispiele solcher Systeme. Es ist oft nicht ausreichend, wenn ein ML-Modell gut funktioniert, sondern es muss auch ein Verständnis dafür geben. Nur so können bestimmte Fehlerquellen ausgeschlossen werden, die aufgrund der Vielzahl an Möglichkeiten nicht durch Tests abgedeckt werden können.

- **Fehleranalyse/Debugging:** Fehler in Klassifizierungen oder Vorhersagen von ML-Modellen müssen zuerst verstanden werden, bevor sie ausgebessert werden können. Gerade in der Entwicklung solcher Modelle ist es hilfreich die Ausgaben und Funktionsweisen interpretieren zu können. So kann beispielsweise eingeschätzt werden, ob das Problem an den Trainingsdaten oder an anderer Stelle liegt.

- **Sicherstellung der Fairness:** Darunter ist das Bias gemeint, das vor allem bei einer ungewollten Gewichtung der Trainingsdaten zu einem bestimmten Feature vorkommt. Ein Beispiel hierfür sind Risikobewertungen von Versicherungen, die bestimmte Personengruppen diskriminieren. Da dieser Einfluss der Trainingsdaten oft nicht von Menschen wahrgenommen wird, kann XAI diese Probleme offenlegen.

- **Vertrauen:** Ein wichtiges Kriterium ist das Vertrauen bei der Nutzung von ML-Systemen besonders in kritischen Umgebungen wie der Medizin. Dabei führt fehlendes Vertrauen oft dazu, dass das Modell gar nicht erst genutzt wird. Zu viel Vertrauen auf der anderen Seite kann aber auch zu schwerwiegenden falschen Entscheidungen führen, die der Mensch „blind“ übernommen hat. Erzeugt wird Vertrauen entweder durch lange erfolgreiche Nutzung solcher Systeme oder eben durch Verständnis.

Das **Ziel** für den Einsatz in dieser Arbeit ist in erster Linie die **Fehleranalyse** als **Entwickler**. Es soll untersucht werden, ob es im Trainingsset bestimmte Informationen gibt, die eine Verbesserung des Netzes ermöglichen. Zusätzlich trifft aber auch das **(wissenschaftliche) Verständnis** als sowohl **Datenwissenschaftler** als auch **Anwender** zu. Es soll untersucht werden, welche Besonderheiten der Datensatz aufweist und ob es einen Erkenntnisgewinn gibt.

3.1.3 Taxonomie

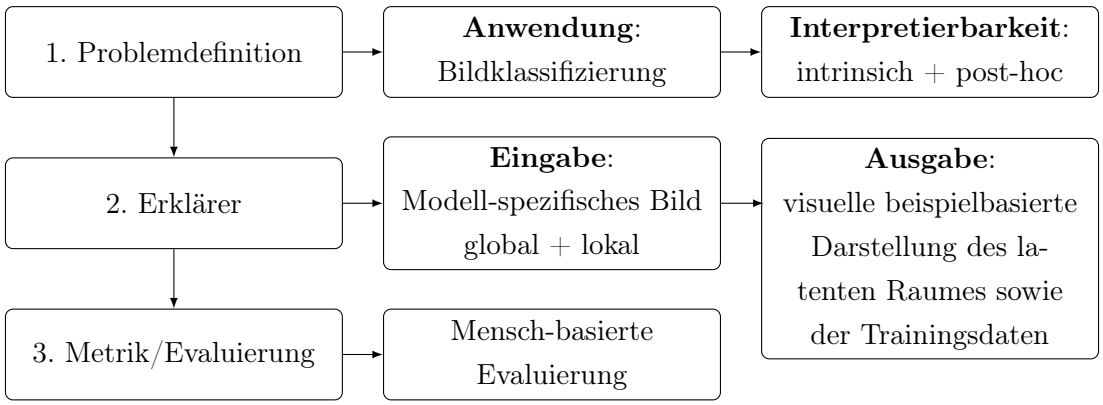


Abbildung 3.2: Vereinfachte Taxonomie von XAI in Bezug auf diese Arbeit (Quelle: In Anlehnung an Schwalbe und Finzel [65])

Abbildung 3.2 zeigt den relevanten Ausschnitt der für diese Arbeit geltenden Taxonomie [65]. Die Einordnung erfolgt dabei in drei Schritten:

1. **Problemdefinition**: Bevor ein Modell interpretiert werden kann, muss es eine Problemstellung geben. Diese ist in dieser Arbeit beispielsweise Bildklassifizierung. Das für die Anwendung gewählte Modell kann dabei bereits von vornherein interpretierbar sein und ist in verschiedene Stufen unterteilt. Die höchste Stufe *intrinsisch* ist dabei scheinbar am besten interpretierbar.

   Die **Anwendung** beinhaltet sowohl den Anwendungstyp sowie dem Datentyp. Im Falle dieser Arbeit ist die Grundlage ein Bildklassifikator.

   Da das definierte Ziel der Arbeit ist einen VAE ergänzend für die **Interpretierbarkeit** zu nutzen, verändert das auch das Modell an sich. Dadurch wird bewusst versucht ein *intrinsisches* Modell zu implementieren. Die Einordnung trifft hier

zu, da ein VAE ein bayessches Netzwerk ist. Zusätzlich ist die Verwendung weiterer Interpretationsmethoden außerhalb des Modells als *post-hoc* zu definieren. Dies wird konkret über NCC als auch die Distanzberechnungen zu den Trainingsdaten erreicht, die nachträglich angewandt werden.

2. **Erklärer**: Hierfür wird zuerst die Eingabe definiert, welche die notwendigen Komponenten enthält, um die Erklärung zu erzeugen. Die Ausgabe ist dann die resultierende Erklärung und wie sie präsentiert wird.

   Die **Eingabe** ist Modell-spezifisch, da nur Bilder erlaubt sind. Die Lokalität beschreibt dabei zum einen *warum* (lokal) ein bestimmtes Bild klassifiziert wird, zum anderen *wie* (global) die Entscheidung getroffen wurde. In dieser Arbeit treffen beide Lokalitäten zu.

   Die Beispiel-basierten **Ausgaben** zeigen mehrere visuelle Darstellungen. Das sind zum einen die intrinsischen Eigenschaften des VAEs, konkret die Verteilung im latenten Raum und die Rekonstruktion, zum anderen die Klassifikation sowie die Modell-unabhängige NCC-Berechnung.

3. **Metrik**: Basierend auf [24] sind hiermit *Funktions-*, *Mensch-* oder *Anwendungs*basierte Metriken gemeint, die zur Evaluierung der Erklärung genutzt werden. In diesem Fall ist das Modell **Mensch-basiert**. Das bedeutet, dass dieser seine subjektive Interpretation der Ausgaben für eine Entscheidung nutzt.

3.1.4 Visuelle Methoden

XAI-Methoden sind zumeist visuelle Verfahren, die bestimmte Bereiche in einem Bild hervorheben, die für die Klassifizierung große Relevanz haben. Obwohl diese Methoden nicht viel mit dieser Arbeit gemeinsam haben, sollen sie dennoch kurz vorgestellt werden, um eine ergänzende Sichtweise für die Diskussion in Kapitel 6 zu ermöglichen.

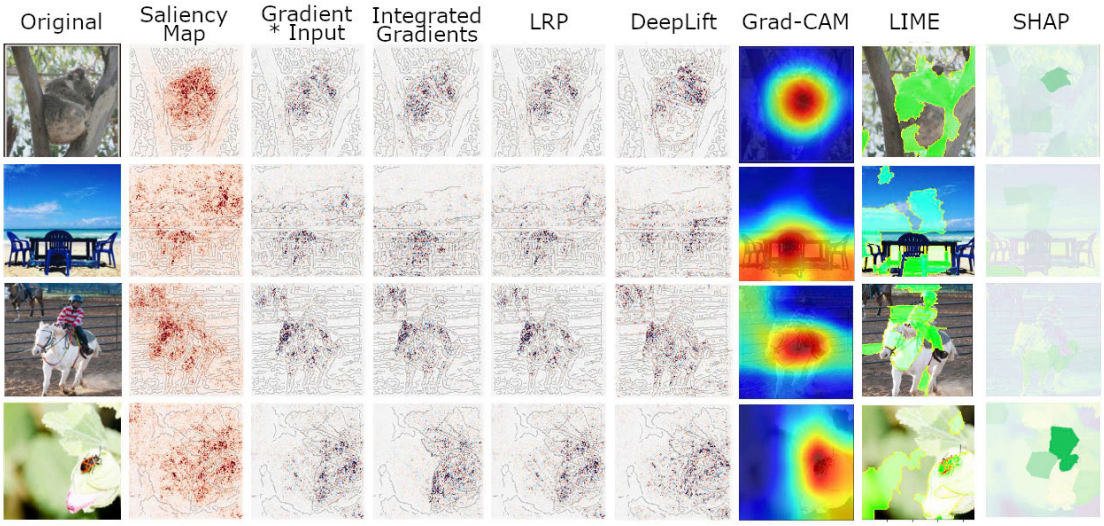


Abbildung 3.3: Übersicht der Gradienten- und Pertubation-basierten XAI-Verfahren (Quelle: Das und Rad [21]).

Abbildung 3.3 aus [21] zeigt eine Übersicht von Gradienten- und Pertubation-basierten XAI-Verfahren. Wie schemenhaft zu sehen, sind *Saliency Map* [70] bis *DeepLift* [69] in ihrer Ausgabe sehr ähnlich und heben vor allem die Pixel hervor. *Grad-CAM* nutzt eine *Heatmap* für wichtige Bereiche. *LIME* [63] und *SHAP* [48] arbeiten mit *Pertubationen* (Störer), die sich durch Überdeckung den Bildbereichen annähern, die den besten *Score* einer Klassifizierung erlangen. Was diese Methoden vereint, ist die gezielte Darstellung relevanter Pixelbereiche. Dadurch sind sie sehr gut dafür geeignet eine Klassifizierung maschinennah zu erklären. Allerdings weniger dafür, aus welchem globalen Zusammenhang (z.B. in Bezug auf Trainingsdaten) diese Visualisierung zu Stande gekommen ist.

3.2 Variational Autoencoder

ML-Modelle lassen sich in zwei Hauptgruppen unterteilen: *Discriminative* und *generative*. Während erstere durch Beobachtungen eine deterministische Funktion zu einem Ergebnis erlernen, ist das Ziel der generativen Netze allgemeinere Strukturen durch Wahrscheinlichkeiten zu formen. Dadurch ist es möglich, entzerrte und interpretierbare Repräsentationen der realen Welt zu modellieren und Erkenntnisse über kausale Zusammenhänge zu gewinnen [41]. Ein VAE ist ein solches generatives Modell, beziehungsweise aufgrund der verbreiteten Verwendung ein *Framework*.

3.2.1 Autoencoder im Allgemeinen

Autoencoder (AE) im Allgemeinen [7] haben vielfältige Anwendungsbereiche und gibt es in mehreren Varianten. Vereinfacht ausgedrückt komprimieren AEs einen Input x auf eine niedrigdimensionale Repräsentation z und können anhand dieser eine sich dem Original annähernde Rekonstruktion x' erzeugen (siehe Abbildung 3.4). Die Repräsentationsschicht wird auch latenter Raum genannt. Diese Schicht wird auch dafür genutzt, die oft komplexen Eingangsdaten hinsichtlich ihrer wesentlichen Features zu reduzieren und dadurch für den Menschen interpretierbar zu machen.

Ein Nachteil von klassischen AEs ist, dass es kaum eine Generalisierung der Daten gibt. Ein unbekanntes Eingangsbild könnte im latenten Raum so gesetzt werden, dass eine unklare Zuordnung zum Ausgangsbild (Rekonstruktion) stattfindet [52]. Dadurch können fehlerhafte Rekonstruktionen bei neuen Eingangsbildern entstehen. Ein weiterer Nachteil ist, dass die Verteilung im latenten Raum inhomogen ist und manche Klassen dadurch viel größere Abstände zueinander haben als andere. Das passiert, weil ein AE eine gute Kompression erlernt, aber nicht zwangsläufig eine gleichmäßige Verteilung.

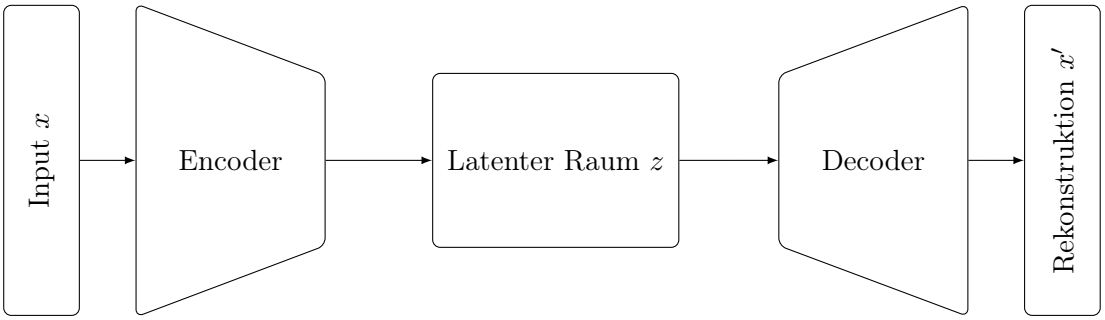


Abbildung 3.4: Schematische Darstellung eines Autoencoders (Quelle: In Anlehnung an Andreas Meisel / Mustererkennung und Machine Learning)

Um die genannten Nachteile eines klassischen AEs zu kompensieren, kann ein VAE verwendet werden. VAEs sind generative ML-Modelle, die neue Ausgaben über eine probabilistische Verteilung im latenten Raum erzeugen können. Vereinfacht ausgedrückt werden so statt Punkten, normal-verteilte Bereiche erzeugt, welche der lückenhaften Repräsentation eines klassischen AEs entgegenwirken.

3.2.2 Aufbau

Für einen VAE wird der klassische AE um zwei Komponenten erweitert: Dem Mittelwert μ und der Abweichung σ. Diese beiden Werte definieren eine Normalverteilung $\mathcal{N}\left(\mu, \sigma^2\right)$. Aus dieser wird zufällig eine Stichprobe (engl. *Sample*) z erzeugt. Durch dieses Verfahren lässt sich der latente Raum eines VAEs regulieren. Abbildung 4.2 zeigt den Aufbau. Im Gegensatz zum AE ist der Encoder hier ein probabilistischer Encoder $q_\phi(z|x)$, wobei ϕ die variablen (engl. *variational*) Parameter symbolisiert. Ziel des Encoders ist es, die meist komplexen Beobachtungen mit der Verteilung $p_\theta(z|x)$ (*posterior*) auf eine einfachere Verteilung $q_\phi(z|x)$ zu approximieren. Das wird durch die Gleichung 3.1 ausgedrückt:

$$q_\phi(z|x) \approx p_\theta(z|x) \tag{3.1}$$

Analog dazu ist der Decoder ein probabilistischer Decoder $p_\theta(x|z)$, der aus einer Stichprobe z aus $p_\theta(z)$ (*prior*) die Rekonstruktion x' generiert.

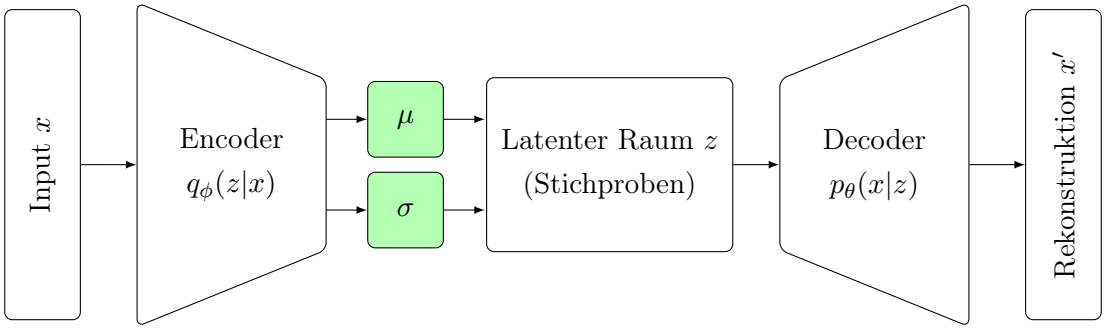


Abbildung 3.5: Schematische Darstellung eines Variational Autoencoders (Quelle: In Anlehnung an Andreas Meisel / Mustererkennung und Machine Learning)

3.2.3 Verlustfunktion

Die Verlustfunktion eines VAE besteht aus dem Rekonstruktionsverlust sowie der Kullback-Leibler-Divergenz (KL-Divergenz). Ersterer berechnet, wie auch im klassischen AE, den Fehler zwischen Eingangs- und Ausgangsbild und zwingt das Netz eine möglichst gute Rekonstruktion zu erzielen. Die zweite Funktion versucht eine Standard-Normalverteilung der Daten zu erreichen, indem Abweichungen bestraft werden.

Der Rekonstruktionsverlust L_R ist abhängig von der Problemstellung und daher anwendungsbezogen. Ziel ist es, die Rekonstruktion p mit dem Originalbild y zu vergleichen und den resultierenden Fehler zu minimieren. Das wird auch damit bezeichnet, den *variational lower bound* bzw. *evidence lower bound (ELBO)* zu maximieren. Die Verlustfunktion L_R ist in dieser Arbeit mit *Binary Cross Entropy* umgesetzt und definiert durch:

$$L_R = -\frac{1}{n} \cdot \sum_{i=1}^{n} \left(y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right) \tag{3.2}$$

Die KL-Divergenz D_{KL} beschreibt die Abweichung zweier Verteilungen, die konkret dafür genutzt wird, die Gleichung 3.1 zu realisieren. Die gewünschte Verteilung $q_\phi(z|x)$ ist dabei in der Regel die Standard-Normalverteilung $\mathcal{N}(0, 1)$. Das Ergebnis der Funktion beschreibt, wie sich diese Verteilungen unterscheiden. Da hierbei eine Distanz berechnet wird und eine perfekte Übereinstimmung 0 ergibt, ist $D_{KL} \geq 0$. Die KL-Divergenz ist definiert durch:

$$D_{KL} = \frac{1}{2} \cdot \sum_{i=1}^{n} \left(\mu_i^2 + \sigma_i^2 - \log(\sigma_i) - 1 \right) \tag{3.3}$$

Da die KL-Divergenz immer positiv ist, wirkt sie regulierend auf den Rekonstruktionsterm. Die gesamte Verlustfunktion ist somit definiert durch:

$$L_{GESAMT} = L_R + D_{KL} \tag{3.4}$$

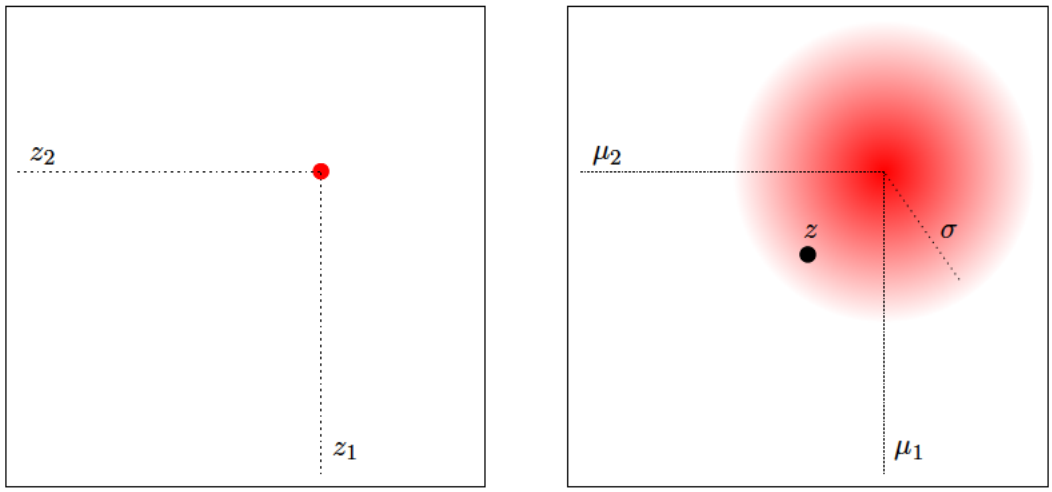
3.2.4 Sampling und Fehlerrückführung

Ein VAE hat das Ziel eine Gaußsche Normalverteilung im latenten Raum zu erreichen. Während klassischer AE einen Punkt z im latenten Raum direkt über die Ausgabe des Encoders setzt, erzeugt ein VAE über den Mittelwert μ und der Abweichung σ einen zufälligen Punkt z innerhalb dieser Verteilung.

Für die Fehlerrückführung ist das allerdings ein Problem. Da diese nicht mit Zufallswerten funktioniert, wurde der *Reparametisierungs-Trick* [40] erarbeitet. Dieser verschiebt die Zufälligkeit aus dem trainierbaren Netz und führt einen neuen Wert ϵ ein. Durch diese Auslagerung kann die Fehlerrückführung die relevanten Parameter z, μ und σ optimieren, während die Zufälligkeit erhalten bleibt.

In Abbildung 3.6 ist dieses unterschiedliche Verhalten zwischen AE und VAE zu sehen. Um in der Normalverteilung einen trainierbaren Punkt z zu setzen, wird dieser mit Hilfe der Gleichung 3.5 ermittelt, sodass für z gilt:

$$z = \mu + \sigma \odot \epsilon, \; wo \;\; \epsilon \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{3.5}$$



(a) AE (b) VAE

Abbildung 3.6: Unterschied zwischen AE und VAE im latenten Raum (Quelle: In Anlehnung an Andreas Meisel / Mustererkennung und Machine Learning)

3.2.5 Anwendungsbereiche

Die Anwendungsbereiche von AEs und VAEs sind vielfältig [7]. Nachfolgend sollen die für die Arbeit relevanten Eigenschaften vorgestellt werden.

- **Pretraining:** AEs sind in der Lage durch ihre besondere Architektur ein Pretraining für einen neuronalen Klassifikator durchzuführen. Zuerst wird der AE ohne die gelabelten Daten trainiert. Danach wird der Decoder durch einen Klassifikator ersetzt und mit gelabelten Daten trainiert. Die Gewichte des Encoders sind dabei „eingefroren“ oder werden nur geringfügig verändert (*Finetuning*). Bei dieser Vorgehensweise wird der Encoder auch *feature extractor* bezeichnet. Die anschließende Genauigkeit des Klassifikators ist nachweislich besser, wie unter anderem in [50] festgestellt wurde.

- **Clustering:** Beim *Clustering* geht es darum, ähnliche Datensätze zu gruppieren und unterschiedliche abzugrenzen. Da klassische AEs dafür aus oben genannten Gründen weniger geeignet sind, betrifft dieser Anwendungsbereich vor allem einen VAE. Dieser ist bereits implizit in der Lage solche Cluster zu erzeugen. Genutzt werden diese Cluster vor allem für die Datenanalyse und im Speziellen für die Interpretation komplexer Problemstellungen aus der realen Welt. Da beim Clustering ein Bezug auf den latenten Raum genommen wird, folgt eine genauere Beschreibung in Abschnitt 3.3.
- **Reduktion der Dimensionen:** Einhergehend mit dem Clustering ist die Reduktion der Dimensionalität wichtig. Um interpretierbare Repräsentationen hoch dimensionaler realer Daten zu erhalten, müssen diese zunächst von den Dimensionen her reduziert werden. Das ist insbesondere wichtig in Bezug auf den *Fluch der Dimensionalität* [9].

3.2.6 β-VAE

Ein β-VAE [34] ist eine Variation eines VAEs, der mehr Kontrolle über den latenten Raum hinsichtlich der „Entzerrung“ (engl. *Disentaglement*) ermöglicht. Dieser wird zumeist aufgrund seines Zielkonfliktes bezüglich Rekonstruktion diskutiert [7] [13] [49]. Durch Setzen eines Regulierungsparameters β, wie in Gleichung 3.6 zu sehen, kann die Gewichtung der KL-Divergenz variiert werden. Bei Werten $\beta > 1$ wird die Abweichung zur Normalverteilung stärker bestraft, ein Wert $\beta < 1$ bevorzugt die Rekonstruktion. Ein Wert von $\beta = 1$ ist ein normaler VAE.

$$L_{GESAMT} = L_R + \beta \cdot L_{KL} \tag{3.6}$$

3.3 Latenter Raum

Der latente Raum (engl. *latent space*) ist im Bereich des maschinellen Lernens weitestgehend als eine Schicht bekannt, die eine vereinfachte und komprimierte interne Repräsentation der Trainingsdaten enthält. Streng genommen ist jede Schicht aus Neuronen in einem KNN ein latenter Raum, doch in dieser Arbeit geht es um die Schicht zwischen

Encoder und Decoder mit der geringsten Anzahl an Dimensionen, auch *Bottleneck* genannt. Ähnlichkeiten von Daten in der realen Welt sind entsprechend nah nebeneinander angeordnet, auch *Nachbarschaft* genannt. Der latente Raum wird oft als eine niedrigdimensionale Grundlage für das Verständnis der sonst komplexen und hochdimensionalen realen Daten genutzt. Aufgrund des steigenden Informationsumfangs der heutigen Zeit wird auch zunehmend an neuen und besseren Lösungen zum Interpretieren des latenten Raums geforscht [30] [81] [32] [75] [31].

3.3.1 Methoden der Visualisierung

Eine verständliche Visualisierung darf nicht mehr als drei Dimensionen beinhalten[2]. Ein so kleiner Raum kann in der Regel nicht für Klassifizierungsprobleme gewählt werden, da durch die Komprimierung essentielle Informationen verloren gehen und die Genauigkeit des Modells beeinträchtigt wird. Um dennoch interpretierbare Darstellungen aus diesen latenten Räumen zu generieren kann auf Methoden der Dimensionsreduktion zurückgegriffen werden. Zwei bekannte Methoden sind *principal component analysis* (PCA) [51] [68] und *t-Distributed Stochastic Neighbor Embedding* (t-SNE) [35] [33]. Während PCA ein deterministisches Verfahren ist und die globalen Distanzen berücksichtigt, ist t-SNE nicht-deterministisch und eher für lokale Relationen wie Nachbarschaften bzw. Cluster geeignet. In dieser Arbeit wird PCA für die Betrachtung des latenten Raums in Kapitel 5 genutzt.

3.3.2 Distanzberechnung

In Bezug auf die Distanzberechnung sind die beiden zuvor genannten Verfahren ungeeignet. Das liegt zum einen daran, dass die Koordinaten eines Eingangsbildes nicht mehr mit dem Koordinatensystem der neuen Repräsentation übereinstimmen und es keinen Bezug mehr zum ursprünglichen latenten Raum gibt. Zum anderen ist der VAE selbst in der Lage, einen solchen Raum abzubilden und diesen wieder umzuwandeln. Ungeachtet der Verfahren sind Distanzberechnungen in hochdimensionalen Räumen allgemein schwierig, wie z.B. in [10] [2] [37] diskutiert.

[2] Bedingt auch vier Dimensionen bei zeitlichen Abfolgen wie z.B. Videos

3.4 Bildähnlichkeit

Es ist schwierig zu beurteilen, was Bildähnlichkeit ist. Es können beispielsweise die Klassen ähnlich sein, obwohl ein Husky und ein Dackel nicht gerade die ähnlichsten Hunderassen sind. Es können aber auch die Positionen von Objekten in einem Bild gemeint sein. Beispielsweise können zwei Bilder von je einem Strand vollkommen unterschiedliche Objekte enthalten, aber dennoch von der Semantik her übereinstimmen.

ML-Algorithmen sind in der Lage diese Ähnlichkeiten und Differenzen selbstständig zu erlernen. Gerade diese erlernten Konzepte tragen viel zur Interpretierbarkeit bei. Da Informationen in Form von Vektoren in einem latenten Raum gespeichert werden, sind die Ähnlichkeiten anhand von Distanzfunktionen zu ermitteln. Diese bringen mathematisch zum Ausdruck, was das KNN in einer bestimmten Schicht als ähnlich ansieht.

Im Gegensatz dazu ermöglicht die Rekonstruktion eines (V)AEs die Ähnlichkeit pixelgenau zu bestimmen. Ein Abweichung kann daher als schlechte Rekonstruktion interpretiert werden und somit einen Hinweis auf ein ungewolltes Verhalten geben.

Beide Arten der Ähnlichkeiten sind daher für einen VAE einsetzbar und nachfolgend werden die verwendeten Methoden genauer erklärt.

3.4.1 Euklidische Distanz

Die Distanzfunktion hat das Ziel die nächsten benachbarten Bilder im latenten Raum zu finden. Es ist bekannt, dass Distanzfunktionen bei zu hohen Dimensionen in Relation zu der trainierten Datenmenge nicht mehr so gute Ergebnisse liefern, wie in [37] evaluiert. Dennoch wird eine solche Funktion benötigt, um die latenten Variablen begreifen zu können.

Die euklidische Distanz ist eine Methode, Entfernungen von zwei Punkten in einem n-dimensionalen kartesischen Koordinatensystem zu berechnen. Sie basiert dabei auf dem Satz des Pythagoras und wird auch in Arbeiten wie [38] [57] zur Distanzermittlung genutzt. Die Formel hierfür ist:

$$d\left(x, y\right) = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2} \tag{3.7}$$

3.4.2 Normalized Cross Correlation

Normalized Cross Correlation ist ein Verfahren des *Template Matchings* und eine robustere Alternative zu *Sum of squared differences* [36]. Ziel dieser Methode ist es Ähnlichkeiten zwischen zwei Bildern zu ermitteln. Das geschieht, indem ein kleinerer Bildausschnitt t der Größe N innerhalb eines größeren Bildes b der Größe N gesucht wird und jede Position einen bestimmten Wert zugewiesen bekommt. Dieser Wert ist durch einen Intervall $[-1, 1]$ definiert, wobei die Werte folgendermaßen zu interpretieren sind:

1 → Perfekte Übersteinstimmung

0 → Keine Übereinstimmung

-1 → Inverse Übereinstimmung

Berechnet wird der Wert wie folgt [52]:

Für den Bildauschnitt b wird ein Mittelwert $\overline{b_M}$ berechnet

$$\overline{b_M} = \frac{1}{N} \sum_{i=1}^{N} b_i \tag{3.8a}$$

Dieser wird dann vom b subtrahiert, sodass ein mittelwert-befreiter Bildauschnitt M entsteht (für den Helligkeitsausgleich)

$$M_i = b_i - \overline{b_M} \tag{3.8b}$$

Für diesen wird dann die Vektorlänge L bestimmt (für den Kontrastausgleich)

$$L_i = \sqrt{\sum_{i=1}^{N} (M_i)^2} \tag{3.8c}$$

Die Division aus M und L ergibt dann den Einsvektor $\hat{b}$

$$\hat{b}_i = \frac{M_i}{L_i} \tag{3.8d}$$

oder auch ausgeschrieben

$$\hat{b}_i = \frac{b_i - \overline{b_M}}{\sqrt{\sum_{i=1}^{N} \left(b_i - \overline{b_M}\right)^2}} \tag{3.8e}$$

Analog zu Bildauschnitt b gilt für Template t

$$\hat{t}_i = \frac{t_i - \overline{t_M}}{\sqrt{\sum_{i=1}^{N} \left(t_i - \overline{t_M}\right)^2}} \tag{3.8f}$$

Somit ergibt sich aus dem Skalarprodukt der Einsvektoren $\hat{b}$ und $\hat{t}$ die Gesamtgleichung

$$NCC\left(b, t\right) = \sum_{i=1}^{N} \hat{b}_i \cdot \hat{t}_i \tag{3.8g}$$

Die Besonderheit von NCC ist, dass Helligkeit und Kontrast im Algorithmus berücksichtigt werden. Dadurch lassen sich auch Vergleiche mit etwas „verschwommenen“ Bildern machen, wie es bei Rekonstruktionen eines VAEs der Fall sein kann. Die Eigenschaft des NCC anhand eines kleineren Bildausschnittes in einem größeren zu suchen ist im Falle dieser Arbeit nicht von Relevanz, da alle Bilder die gleiche Auflösung haben.

4 Methodik

Um die definierten Ziele aus Kapitel 1.2 evaluieren zu können, werden in diesem Kapitel die Architektur, die Implementierung und die Evaluationsmethodik vorgestellt. Die einzelnen Komponenten werden hinsichtlich ihrer Bedeutung für die Interpretierbarkeit und dem jeweiligen Zusammenwirken beschrieben.

4.1 Architektur

Die Architektur in Abbildung 4.1 unterteilt sich in zwei Gruppen: Zum einen dem ML-Modell, welches trainiert und anschließend für die Inferenz genutzt wird. Zum anderen Komponenten, die nach dem Training auf Grundlage der Inferenz berechnet werden.

Das ML-Modell (weiß dargestellt) soll die Eigenschaft eines VAEs nutzen, die Trainingsbilder hinsichtlich ihrer Merkmale im latenten Raum entsprechend ihrer Klassen zu gruppieren. Dafür wird eine Y-förmige Architektur genutzt, welche sowohl einen Decoder als auch Klassifikator an den Encoder anbindet. Diese Anbindung erfolgt über den latenten Raum. Da für das Pretaining auch AEs im Allgemeinen genutzt werden können, soll dadurch auch die Genauigkeit des Klassifikators von dieser Verbindung profitieren. Der Aufbau des VAE orientiert sich dabei an dem Tutorial auf der *Keras*-Website [19]. Der detaillierte Netzaufbau ist im Appendix in Abbildung A.1 mit allen Schichten zu sehen.

Die restlichen Komponenten (grau dargestellt) haben zum einen die Funktion den latenten Raum mit einer Distanzfunktion zu untersuchen und benachbarte bzw. ähnliche Bilder direkt im Trainingsset zu ermitteln. Zum anderen sollen Bilder hinsichtlich ihrer Struktur verglichen werden, um die Ähnlichkeit auf eine alternative Methode zu bestimmen.

Es gibt vier (grün dargestellt) *Interpretationskanäle*, die in Kapitel 4.3.1 genauer beschrieben werden.

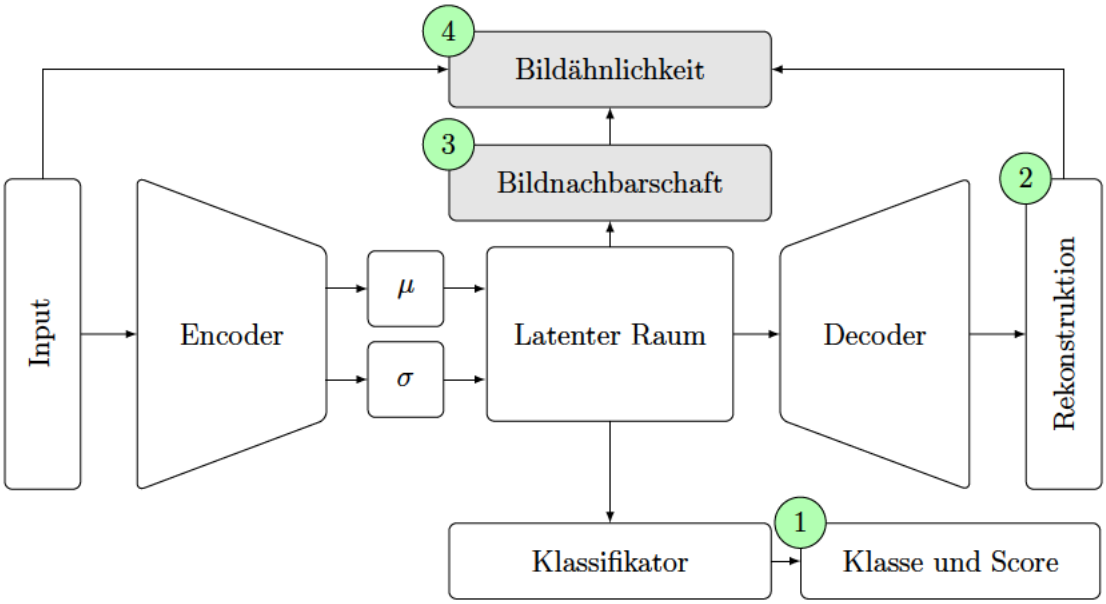


Abbildung 4.1: Der Aufbau des Modells unterteilt sich in ein Y-förmiges ML-Modell (weiß) sowie eigene Komponenten (grau). Die durchnummerierten Komponenten (grün) stellen die vier Interprationskanäle dar: **1)** Die vom Klassifikator ermittelten Klassen sowie dazugehörigem Score, **2)** der Rekonstruktion des Decoders, **3)** einer Liste von n im latenten Raum benachbarten Trainingsbildern und **4)** der Bildähnlichkeit zwischen Eingangsbild und entsprechender Rekonstruktion, sowie den ermittelten benachbarten Bildern.

Die in den Grundlagen vorgestellte Verlustfunktion 3.4 des VAEs wird mit der des Klassifikators L_K ergänzt. Dadurch wirkt dieser als zusätzlich regulierender Faktor auf das Training ein. Es ergibt sich somit folgende Gleichung:

$$L_{GESAMT} = \underbrace{(L_R + \beta \cdot L_{KL})}_{\text{Decoder}} + \underbrace{L_K}_{\text{Klassifikator}} \tag{4.1}$$

4.2 Implementierung

Die Implementierung und Bereitstellung des Modells erfolgt in zwei Schritten, wie in Abbildung 4.2 zu sehen. Zuerst wird es wie ein normales KNN trainiert bis die gewünschte

Performance erreicht ist. Anschließend wird nur der Encoder des Modells genommen und die Trainingsdaten ein weiteres mal über die Inferenz an das Netz gegeben. Die resultierenden Koordinaten werden gespeichert und stehen für Distanzberechnungen neuer Eingaben bereit.

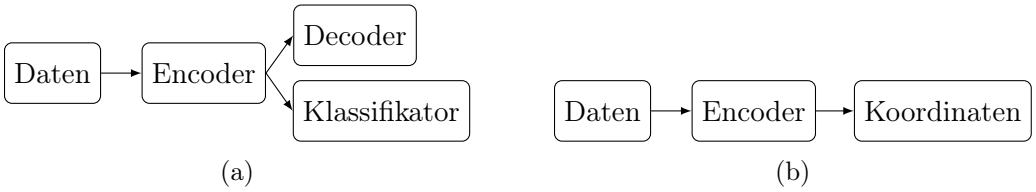


Abbildung 4.2: 2-Stufige Implementierung des Modells. In (a) wird das Modell trainiert, in (b) werden die Daten erneut an den Encoder gegeben und die Koordinaten im latenten Raum gespeichert.

4.2.1 Training

Das Training des Modells läuft bis die gewünschte Performance erreicht wurde. Die verwendeten Paramter sind der Tabelle 4.1 zu entnehmen und beziehen sich - soweit nicht anders angegeben - auf alle Versuche. Für einige Fragestellungen wird insbesondere der β-Wert in Kapitel 6 variiert.

Parameter	Wert
Train-Test-Split	60000:10000
Epochen	100
Batchsize	250
β	1
λ	1
Optimizer	Adam
Learning Rate	0.001

Tabelle 4.1: Parameter des Modells

4.2.2 Speicherung der latenten Koordinaten

Anschließend werden im zweiten Schritt die selben Trainingsdaten an die trainierte Inferenz gegeben und die resultierenden Koordinaten im latenten Raum inklusive Bild-ID

gespeichert[1]. Tabelle 4.2 zeigt die genutzte Datenstruktur. Dadurch können neue Eingangsbilder mit den genutzten Trainingsbildern mit Distanzfunktionen verglichen und die benachbarten direkt ermittelt werden.

ID	Dateipfad	Label-Nr.	Label-Name	Koordinaten
1	/bild1.jpg	0	T-shirt/Top	[0.4404, 1.4732]
2	/bild2.jpg	3	Dress	[-0.5584, -0.9886]
...	...	...	...	...

Tabelle 4.2: Beispielhafte Struktur der gespeicherte Trainingsdaten samt Koordinaten im latenten Raum

4.3 Evaluierungsmethodik

Die Evaluationsmethodik ist *Mensch-basiert* anhand von Beispielbildern. Hierfür wurden für die Ausgabe Interpretationskanäle als auch Evaluationskategorien definiert.

4.3.1 Interpretationskanäle

Unter dem Begriff *Interpretationskanal* wird eine Ausgabe beschrieben, die hinsichtlich der in Kapitel 3.1.1 definierten Interpretierbarkeit hilfreiche Ergebnisse darstellt. Die vier Kanäle wurden ausgewählt, da sie entweder bereits interner Bestandteil des Netzes sind (Klassifikation und Rekonstruktion), die Eigenschaften des latenten Raums nutzbar machen (Bildnachbarschaft) oder eine Modell-unabhängige Bildähnlichkeit ermöglichen. Nachfolgend sollen diese im Detail erläutert werden.

Klasse und Score

Ausgegeben wird hier eine Teilmenge der gegebenen Klassen inklusive des vom trainierten Netz ermittelten Scores der jeweiligen Klasse. In dieser Arbeit beschreibt dieser Kanal den Bezugspunkt für die anderen Kanäle. Das bedeutet, dass richtige Ergebnisse *(true positives)* bestärkt werden sollen, während falsche Ergebnisse *(false positives)* für den Nutzer erkennbar gemacht werden sollen. Nichtsdestotrotz bieten die Klassen-Scores an

[1] In dieser Arbeit als CSV-Datei

sich eine gewisse Interpretierbarkeit mit sich, wenn beispielsweise ähnliche Klassen einen ähnlichen Score erhalten.

Rekonstruktion

Die Rekonstruktion soll einen ersten Aufschluss darüber liefern, wie das Netz das Eingangsbild innerhalb des latenten Raumes einordnet. Da ein VAE ein generatives Netz ist, führen neue Datenpunkte zu einer Rekonstruktion, die ähnliche Features zu benachbarten Punkte aufweist. Die Rekonstruktion kann dann mit Eingangsbild sowie Klassifikation verglichen werden und somit als weiterer Interpretationskanal genutzt werden.

Bildnachbarschaft

Da die Trainingsdaten von ausschlaggebender Bedeutung für jedes neuronale Netz sind, sollen diese in die Analyse einbezogen werden. Das wird erreicht, indem die Koordinaten der Trainingsdaten zunächst im latenten Raum gespeichert werden und anschließend über Distanzfunktionen auf benachbarte Trainingsbilder zugegriffen werden kann.

Die Idee hinter diesem Kanal ist, dass der latente Raum ohne Verfahren zur Dimensionsreduktion (z.B. mit *PCA*) nicht über drei Dimensionen hinaus visuell dargestellt werden kann. Da die Klassen dank des VAEs vom Abstand her relativ gleichmäßig verteilt werden, können auch in höher dimensionierten Räumen Abstandsberechnungen zwischen Punkten durchgeführt werden und eine Anzahl an Bildnachbarn ermittelt werden. Diese nahegelegenen Bilder können dann dargestellt werden und somit einen kleinen Ausschnitt einer Punktewolke simulieren.

Bildähnlichkeit

Es kann passieren, dass das Eingangsbild mit allen zuvor genannten Kanälen einstimmig klassifiziert wurde, es aber de facto falsch ist (*false positive*). Um diese Problematik zu behandeln, sollen sowohl die Rekonstruktion als auch die benachbarten Trainingsbilder jeweils mit dem Eingangsbild verglichen und die Differenz mit der in Kapitel 3.4 vorgestellten Methode bestimmt werden.

Die Rekonstruktion soll per Definition möglichst identisch zum Eingangsbild sein. Falls das nicht der Fall ist, ist das ein Indikator für sich überlappende Cluster im latenten

Raum. Da es sich bei dem VAE-Anteil um ein probabilistisches Modell handelt und ein Punkt durch Zufall in einen falschen Cluster eingeordnet werden kann, sollen auch die Bildähnlichkeit zu benachbarten Bildern berechnet werden. Dadurch soll eine bessere gesamtheitliche Interpretation möglich sein.

4.3.2 Fashion-MNIST

Der Einfachheit halber wird das Training mit dem *Fashion-MNIST*-Datensatz [78] durchgeführt. Dieses wird oft als Benchmark-Datensatz verwendet und ist im Gegensatz zum normalen *MNIST*-Datensatz [45] anspruchsvoller. Des Weiteren kann man leichter Variationen in die Evaluationsbilder integrieren und ist dadurch näher an den Anwendungsfällen der realen Welt. In Abbildung 4.3 sieht man einen Teil des *Fashion-MNIST*-Datensatzes mit seinen zehn Klassen.

Abbildung 4.3: Beispielbilder des Fashion-MNIST Datensatzes. Jedes Bild hat eine Auflösung von 28x28 Pixeln.

4.3.3 Evaluationsbilder

Um das mit dem *Fashion-MNIST*-Datensatz trainierte Netz analysieren zu können, wurden *Evaluationsbilder* nach bestimmten Kriterien gewählt und in Kategorien eingeteilt, wie in Abbildung 4.4 zu sehen. Diese Kategorien sollen dabei stufenweise die Schwierigkeit erhöhen, mit der das Netz eine gute Klassifikation erreicht, um sowohl gute als auch schlechte Beispiele zu erlangen. Dabei ist anzumerken, dass die Evaluierung nach subjektiver Annahme des Autors durchgeführt wird. Das Ziel ist aber die Interpretationsfähigkeit an sich zu bewerten, auch wenn diese Evaluierung nicht präzise ist.

Abbildung 4.4: Beispiel der Evaluationskategorien an der Klasse *T-Shirt/Top*. Das Bild links (Kategorie A) ist sehr nah am Trainingsset. Das Bild in der Mitte (Kategorie B) weicht geringfügig ab, da die Ärmel etwas zusammengefaltet sind. Das letzte Bild (Kategorie C) beinhaltet sowohl eine Person, einen Hintergrund als auch Teile einer Hose, was bereits stark vom Trainingsset abweicht. (Quellen: Von links: Aeff Burroughs / Pixabay, Ryan Hoffman / Unsplash, Ian Dooley / Unsplash)

Kategorie A: Passend

Unter dieses Kriterium fallen einfache Beispiele (Abbildung 4.5), die dem Trainingsset sehr ähnlich sind und kaum Abweichungen aufweisen. Insbesondere der weiße Hintergrund ist hier ausschlaggebend. Es werden hier durchgehend gute Ergebnisse erwartet und dass alle *Interpretationskanäle* adäquate Resultate liefern.

Abbildung 4.5: Beispiele der Evaluationsbilder der Kategorie A (Quellen: Von links: Mostafa Mahmoudi, Aurelia Dubois, Kelly Sikkema, Maria Beatrice Alonzi, Santhosh Kumar, Luis Felipe / Unsplash)

Kategorie B: Geringfügig abweichend

Diese Bilder (Abbildung 4.6) sollen geringfügig abweichende Merkmale aufweisen wie beispielsweise Kleiderhaken, eine andere Hintergrundfarbe oder Rotation. Da das Trai-

ningsset an sich scheinbar wenig Variationen aufweist, könnte dieses Kriterium schwieriger erkannt werden.

Abbildung 4.6: Beispiele der Evaluationsbilder der Kategorie B (Quellen: Von links: Engin Akyurt, Raquel Gambin, Ryan Hoffman, Santhosh Kumar, Usama Akram, Luis Felipe / Unsplash)

Kategorie C: Stark abweichend

Unter diese Kategorie fallen Bilder mit Personen, Hintergründen und anderen Objekten (Abbildung 4.7) mit hohem Detailgrad. Die Frage bei dieser Kategorie ist, ob trotz der Schwierigkeit aussagekräftige Ausgaben der Interpretationskanäle möglich sind.

Abbildung 4.7: Beispiele der Evaluationsbilder der Kategorie C (Quellen: Von links: Paulina Milde Jachowska, R N, Mediamodifier, Rumman Amin, Ian Dooley, Christian Bolt / Unsplash)

5 Experimente

In diesem Kapitel werden auf Grundlage der Methodik aus Kapitel 4 verschiedene Experimente durchgeführt, um die aus Kapitel 1.2 definierten Fragestellungen evaluieren zu können. Die Ergebnisse bestehen dabei aus einer Auswahl an relevanten Beispielen. Die Evaluierung und Diskussion folgt dann in Kapitel 6.

5.1 Beispielinterpretationen

In diesem Unterkapitel werden die in Kapitel 4 vorgestellten Kategorien exemplarisch angewandt und die in 1.2 formulierte Teilfrage **F1** zur Interpretierbarkeit untersucht. Unterteilt sind die Beispiele in **leicht** und **schwierig** interpretierbar sowie jeweils **richtig** und **falsch** klassifiziert. Dabei geht es zunächst darum, welche Informationen aus der Methode gewonnen werden können. Die Unterabbildungen auf den Seiten sind - soweit nicht anders angegeben - folgendermaßen zu verstehen:

(a) Das Originalbild, die für das Netz aufbereitete Version sowie die Rekonstruktion inklusive dem NCC-Wert zwischen aufbereitetem Bild und Rekonstruktion.

(b) Die vom Klassifikator am höchsten gewichteten Klassen-Vorhersagen samt Score.

(c) Die über die euklidische Distanz ermittelten Nachbarbilder aus den Trainingsdaten im latenten Raum.

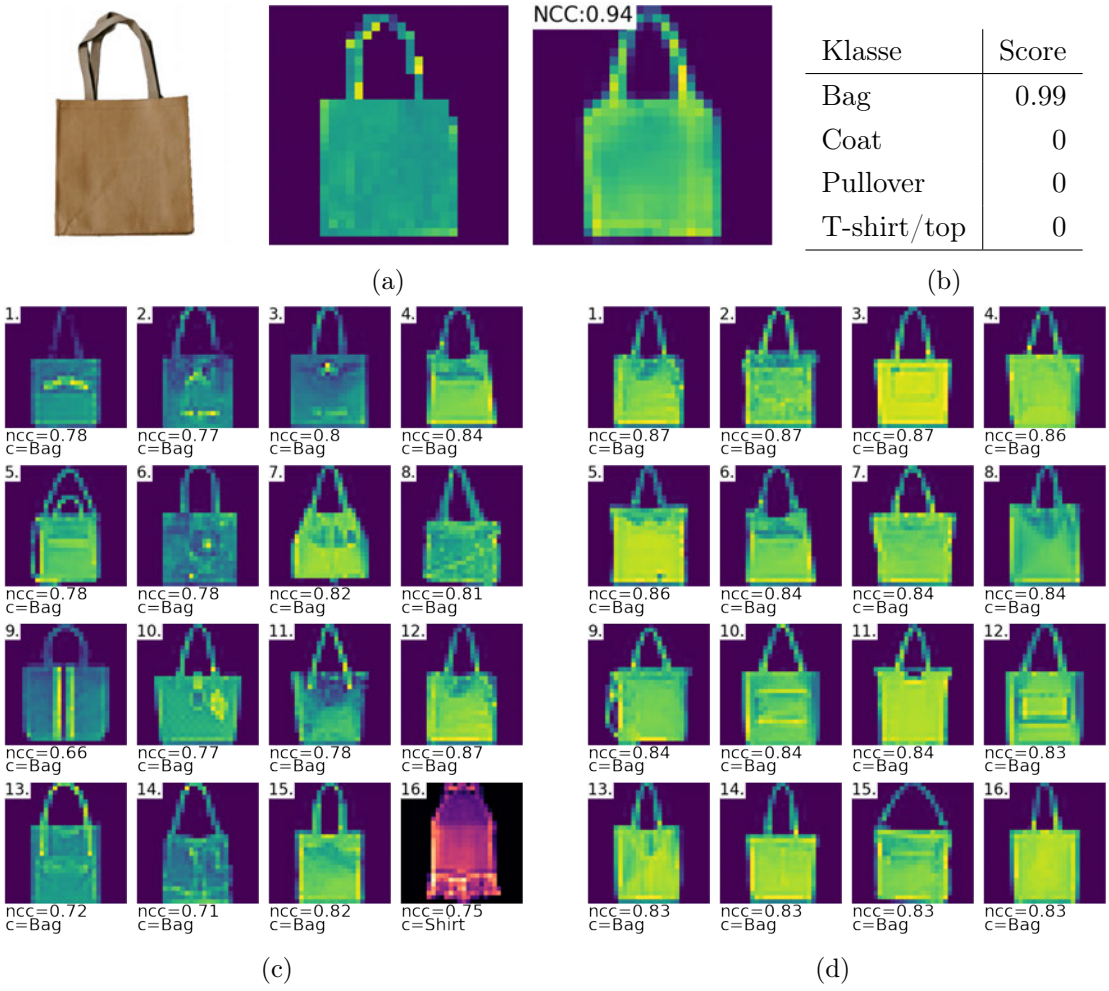
(d) Die über die NCC ermittelten ähnlichsten Bilder aus den Trainingsdaten.

Zusätzlich wird ein Punktdiagramm aller Trainingsdaten in Bezug auf das Eingangsbild dargestellt. Die Abbildungen (c) und (d) stellen eine Teilmenge dieser Repräsentation dar.

Hinweis: Für eine bessere Visualisierung sind die Nachbarn mit falscher Klasse rot gefärbt. In einem realen Kontext wäre das nicht der Fall.

5.1.1 Leicht interpretierbare Beispiele

Beispiel 1: Kategorie A, Klasse *Bag*, Dimensionen 20



Klasse	Score
Bag	0.99
Coat	0
Pullover	0
T-shirt/top	0

Abbildung 5.1: Beispielinterpretation 1: Kategorie A, *Bag*, 20D

In dem Beispiel, siehe Abbildung 5.1, wird ein Bild der Klasse *Bag* mit einem Score von 0.99 erkannt. Auch die Rekonstruktion in (a) ist sehr ähnlich und der NCC-Wert ist nahezu perfekt. Die Nachbarbilder stimmen sowohl über die euklidische Distanz als auch über NCC größtenteils mit dem Ergebnis überein. Einzig das benachbarte Bild Nr. 16 aus (c) ist nicht korrekt. Insgesamt wird dadurch der bereits sehr gute Score von den anderen Interpretationskanälen bestärkt.

In Abbildung 5.2 sieht man die Trainingsdaten in Bezug auf NCC und euklidischer Distanz zum Eingangsbild. Diese Euklid-NCC-Diagramme ermöglichen es den latenten Raum „aus der Sicht des Bildes“ zu betrachten. Das wird erreicht, indem die eindimensionale Größe der Distanz um eine weitere erweitert wird. Dadurch ist eine differenziertere und genauere Betrachtungsweise der Nachbarn möglich. Es fällt auf, dass der Cluster *Bag* optimale Werte für beide Bildähnlichkeiten hat, was in diesem Diagramm die nordwestliche Ecke darstellt. Allerdings gibt es bereits Überlagerungen mittig des Clusters mit mehreren anderen Klassen.

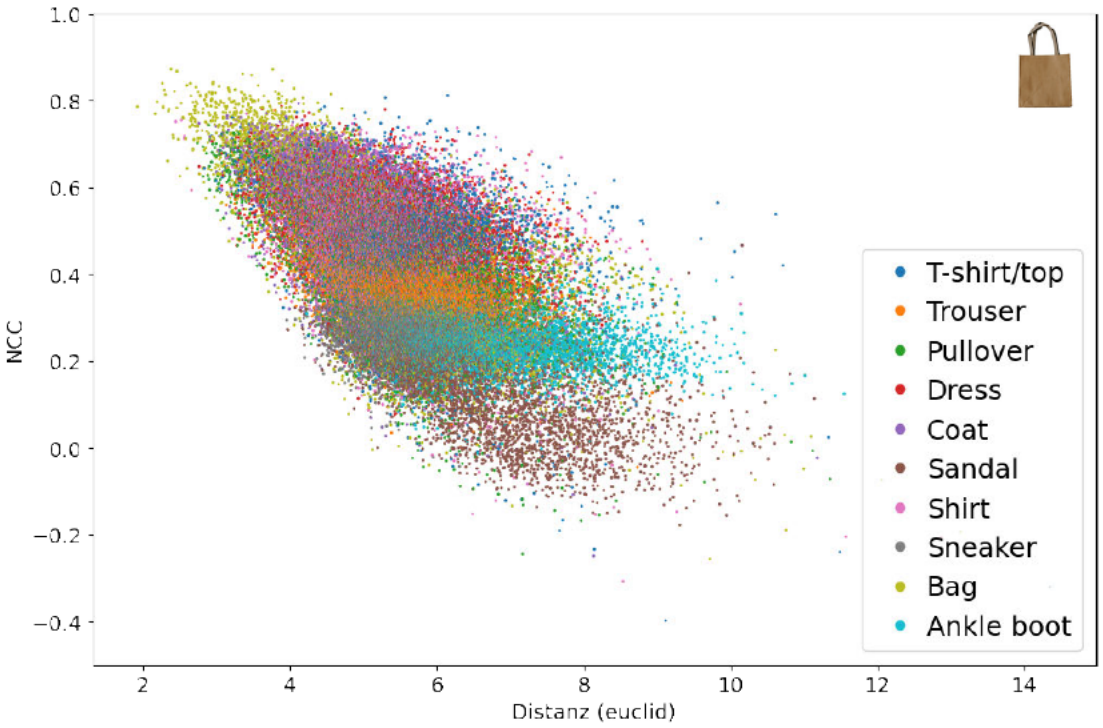
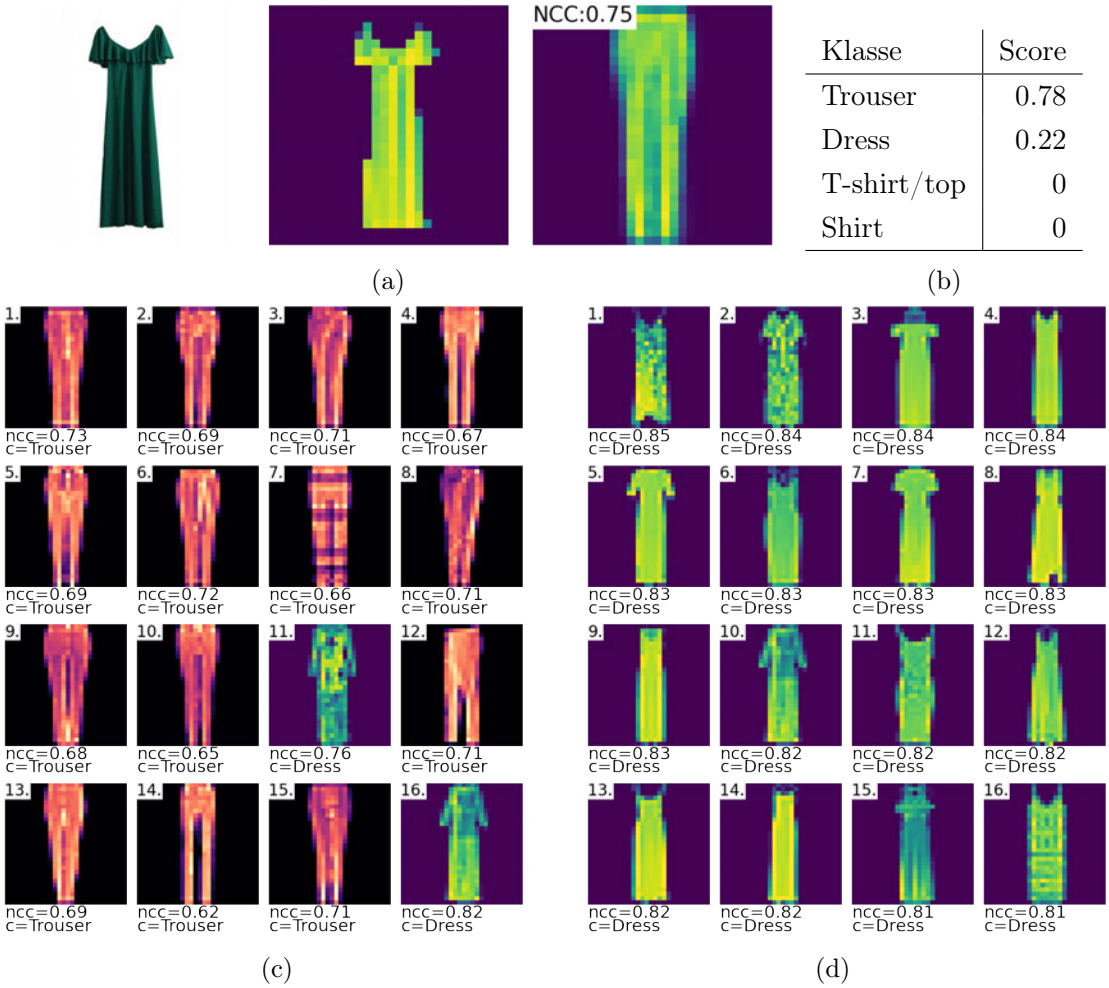


Abbildung 5.2: Euklid-NCC-Relation: Kategorie A, *Bag*, 20D

Somit lässt sich für diese Interpretation schlussfolgern, dass die sehr guten Ergebnisse zwar für das Beispielbild gelten, dies aber nicht unbedingt auf alle Bilder dieser Klasse zutreffen muss.

Beispiel 2: Kategorie A, Klasse *Dress*, Dimensionen 3



Klasse	Score
Trouser	0.78
Dress	0.22
T-shirt/top	0
Shirt	0

Abbildung 5.3: Beispielinterpretation 2: Kategorie A, *Dress*, 3D

Das nächste Beispiel (Abbildung 5.3) zeigt die Klasse *Dress* in einem dreidimensionalem latenten Raum, bei der die falsche Klasse *Trouser* klassifiziert (b) wurde. Die Rekonstruktion weicht von einer perfekten Übereinstimmung mit NCC ab. Bei den euklidischen Bildnachbarn werden nur zwei korrekte Klassen (Nr. 11 und Nr. 16) gezeigt, wobei diese bessere NCC-Werte haben als die falschen. Die NCC-Nachbarn an sich sind komplett richtig.

Betrachtet man die Abbildung 5.4, so sieht man das entsprechende Verhältnis der Cluster *Dress* und *Trouser*. Während ersterer Richtung gutem NCC tendiert, ist letzterer näher

bezüglich der Distanz. Zwar ist der latente Raum nur 3 Dimensionen groß, dennoch sieht man an diesem Beispiel gut, wie ähnlich diese beiden Klassen sind.

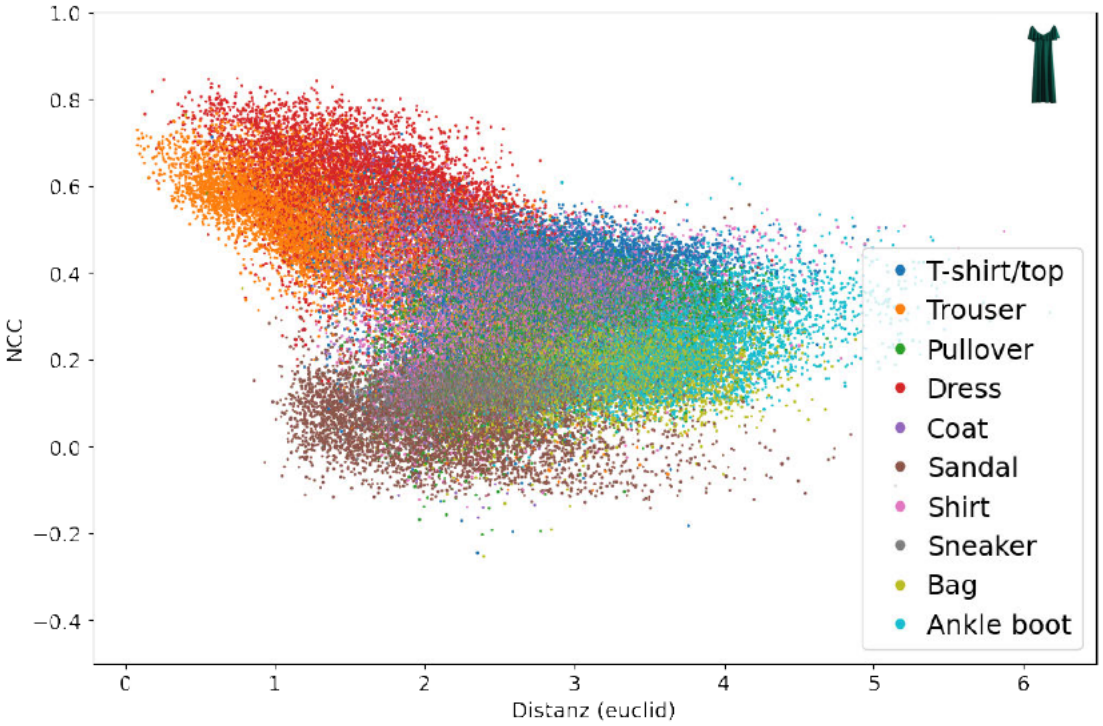
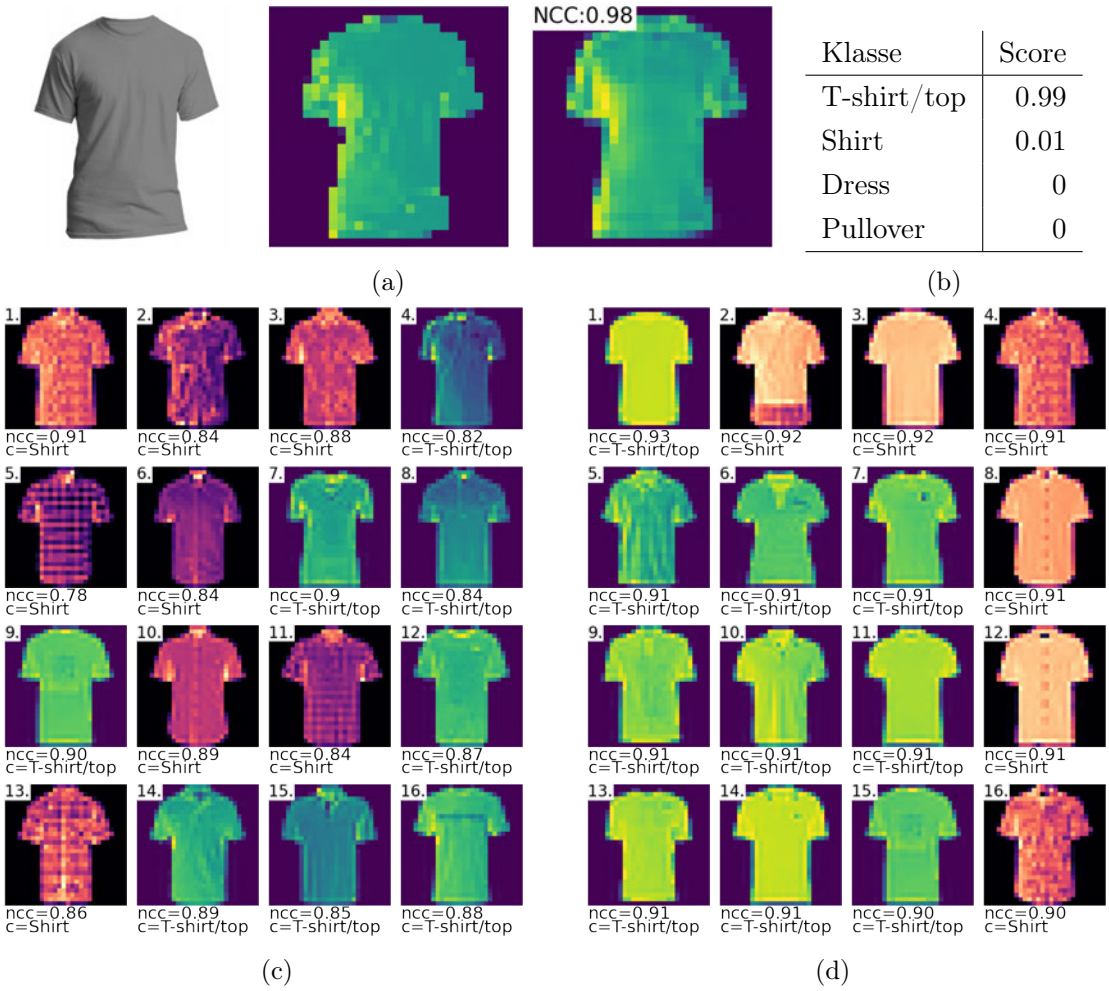


Abbildung 5.4: Euklid-NCC-Relation: Kategorie A, *Dress*, 3D

5.1.2 Schwierig interpretierbare Beispiele

Beispiel 3: Kategorie A, Klasse *T-Shirt/top*, Dimensionen 100



Klasse	Score
T-shirt/top	0.99
Shirt	0.01
Dress	0
Pullover	0

Abbildung 5.5: Beispielinterpretation 3: Kategorie A, *T-Shirt/top*, 100D

Im Beispiel aus Abbildung 5.5 wird die korrekte Klasse mit einem sehr guten Score (b) sowie nahezu identischer Rekonstruktion (a) ausgegeben. Allerdings zeigen besonders die Nachbarn über die euklidische Distanz (c) die ähnliche Klasse *Shirt*. An dieser Stelle lässt sich nicht so einfach erkennen, welches Merkmal für die Klassifizierung besonders ausschlaggebend ist. Besonders die falschen Nachbarn aus (c) könnten widersprüchliche Annahmen wecken.

Betrachtet man die Abbildung 5.6, so sieht man auch dort eine Überlappung der beiden Klassen. Die restlichen Klassen sind hingegen relativ weit entfernt, sowohl in der euklidischen Distanz als auch mit NCC.

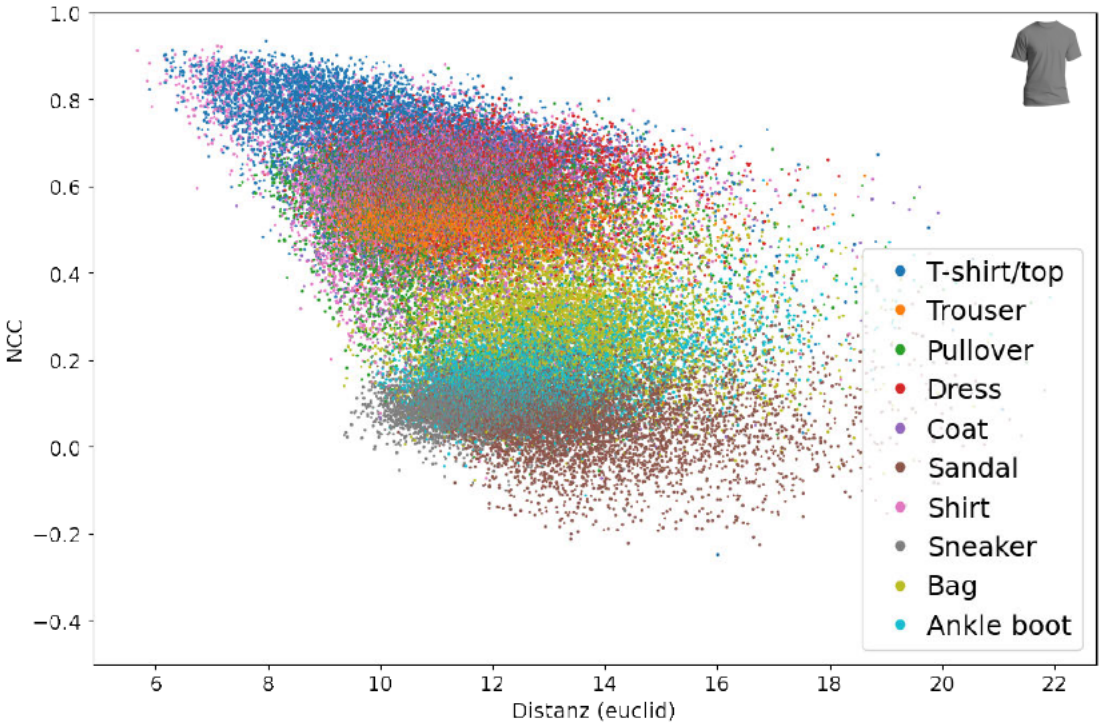
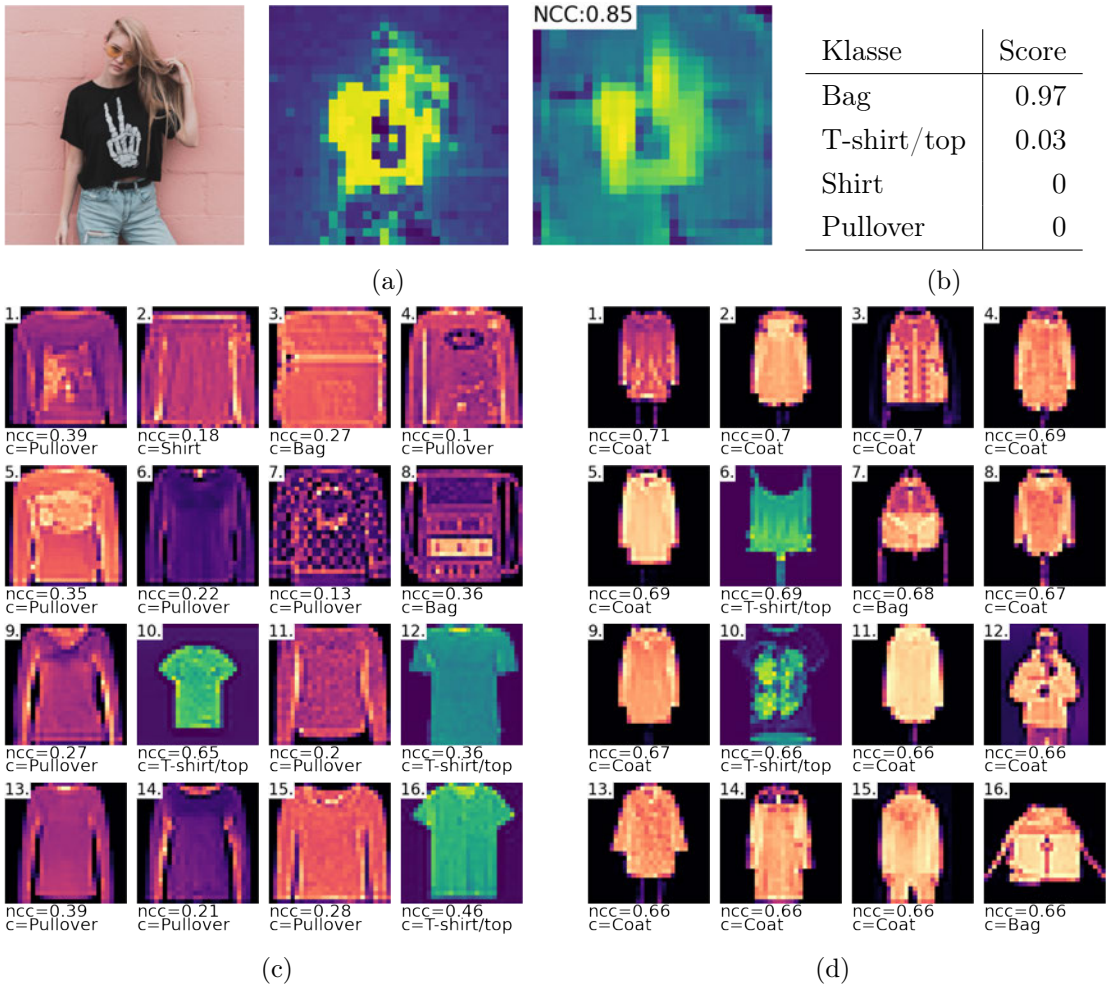


Abbildung 5.6: Euklid-NCC-Relation: Kategorie A, *T-Shirt/top*, 100D

Beispiel 4: Kategorie C, Klasse *T-Shirt/top*, Dimensionen 100



Klasse	Score
Bag	0.97
T-shirt/top	0.03
Shirt	0
Pullover	0

Abbildung 5.7: Beispielinterpretation 4: Kategorie C, *T-Shirt/top*, 100D

Abbildung 5.7 zeigt ein Beispiel der Kategorie C, wo mit großer Sicherheit die falsche Klasse *Bag* (b) erkannt wird. Die Rekonstruktion (a) hat zwar einen guten NCC-Wert, allerdings sieht die Rekonstruktion nicht aus wie die Klasse *T-Shirt/top*. Die euklidischen Nachbarn (c) sind zum größtenteil falsch, allerdings ist Bild Nr. 10 richtig und weist gleichzeitig den höchsten NCC-Wert auf. Die meisten NCC-Nachbarn (d) sind aus der Klasse *Coat*. Da sich die Interpretationskanäle so stark voneinander unterscheiden, ist es schwierig die richtige Klasse zu identifizieren. Nur der Einzelfall mit Bild Nr. 10 aus (c) scheint aufgrund seines höheren NCC-Werts einen Anhaltspunkt zu geben. Dieser wäre

in diesem Fall die übereinstimmende Skalierung des zu erkennenden Objekts der Klasse *T-shirt/top*.

Betrachtet man die entsprechende Abbildung 5.8, so fällt eine starke Überlagerung der Klassen auf. Tendenziell kann die richtige Klasse im nordwestlichen Bereich der Punktewolke gesehen werden. Die vereinzelten Punkte der Klasse *Pullover* westlich auf der Distanz-Achse, die auch bei den Nachbarn aus 5.7c zu sehen sind, haben das Cluster-Zentrum auf der ähnlichen Distanz wie die richtige Klasse. Gemeinhin lässt sich laut der Darstellung vermuten, dass die richtige Klasse einer Überschneidung aus euklidischen und NCC-Nachbarn ist.

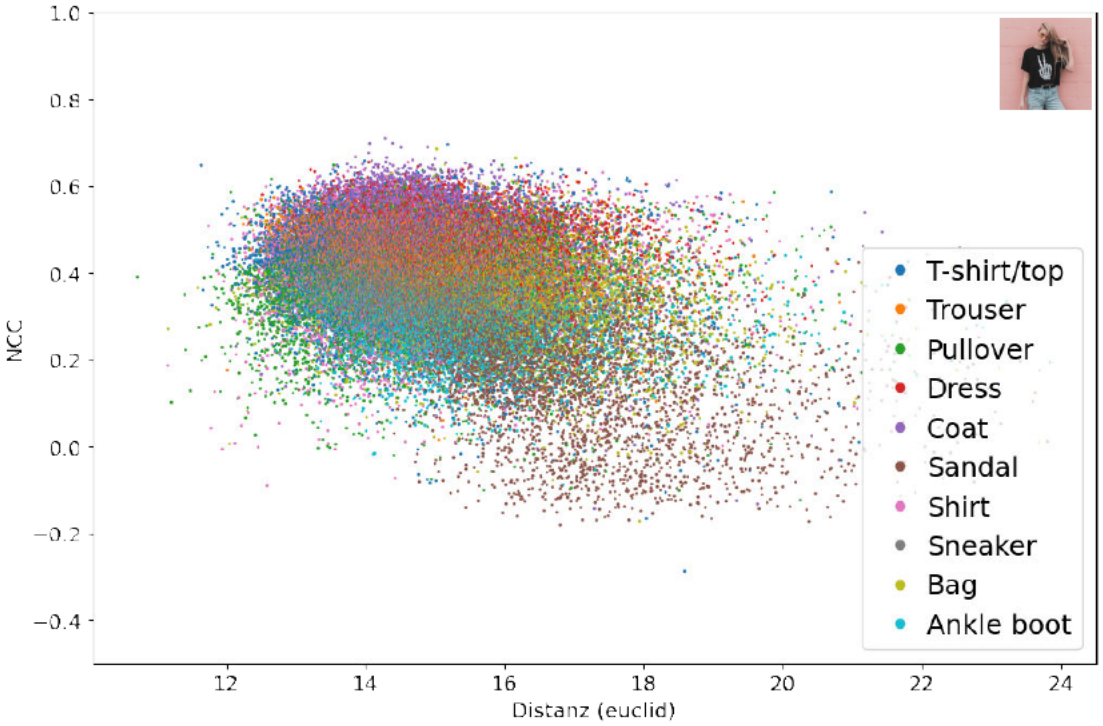
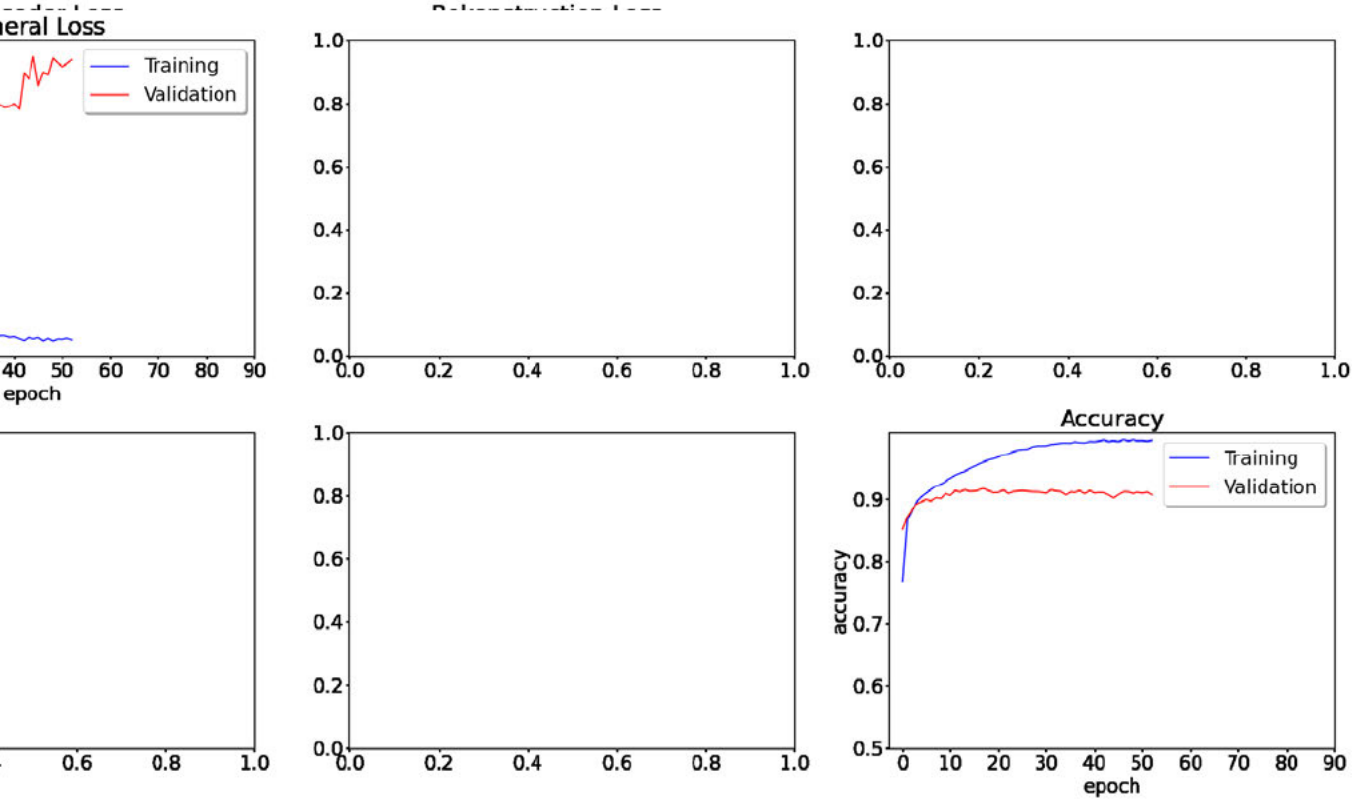


Abbildung 5.8: Euklid-NCC-Relation: Kategorie C, *T-Shirt/top*, 100D



(a) Genauigkeit im VAE+C

(b) Genauigkeit im C

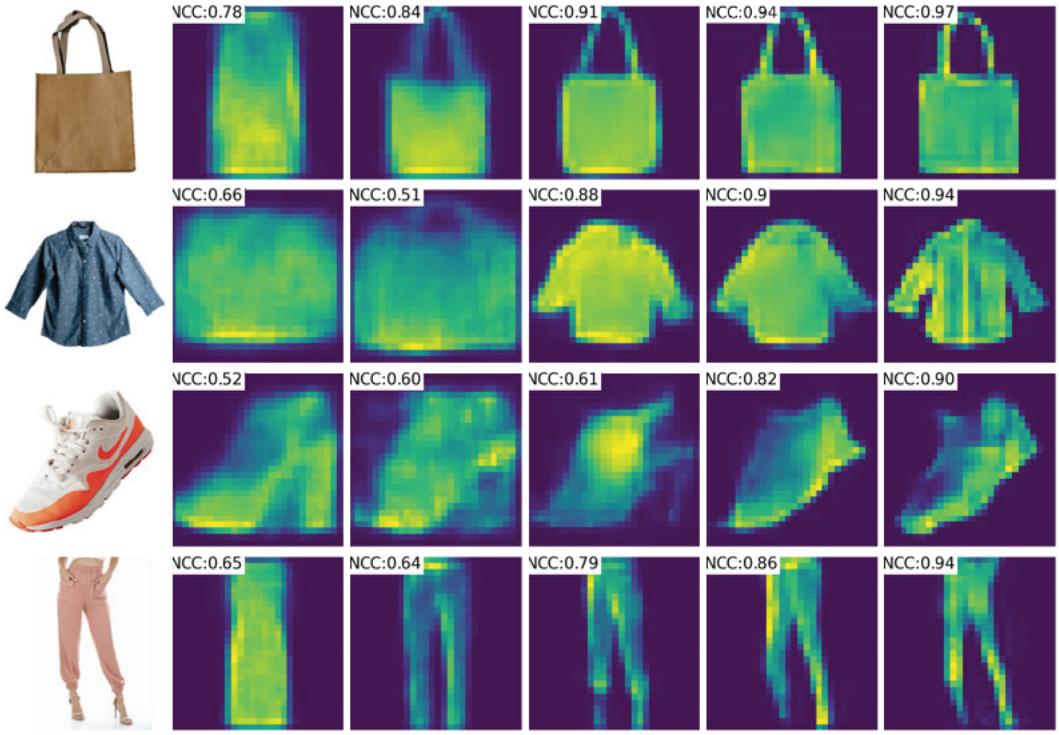
Abbildung 5.9: Vergleich VAEC zu reinem Klassifikator bei gleichem Netzaufbau (10D)

Abbildung 5.9 zeigt einen Vergleich der Genauigkeit zwischen *VAEC* und *C* beim Training. Diese ist bei *C* zwar höher, jedoch weist dieser auch *Overfitting* auf. Es scheint dahingehend auf den ersten Blick so, als ob der *VAEC* das *Overfitting* reguliere. Um an dieser Stelle aussagekräftige Vergleiche zu erlangen, müssten beide Netze verändert und gegeneinander optimiert werden. Da das auch die anderen Experimente beeinflussen würde, soll dieses Ergebnis daher zunächst so hingenommen werden.

5.3 Dimensionen

Da die Anzahl der Dimensionen insbesondere für die Berechnung der Distanz eine wichtige Rolle spielt, soll diese nachfolgend in Bezug auf Teilfrage **F3** untersucht werden. Hierbei werden einige Beispiele mit dem gleichen Bild verglichen und Visualisierungen des latenten Raums über die Dimensionen betrachtet.

5.3.1 Vergleich der Rekonstruktion

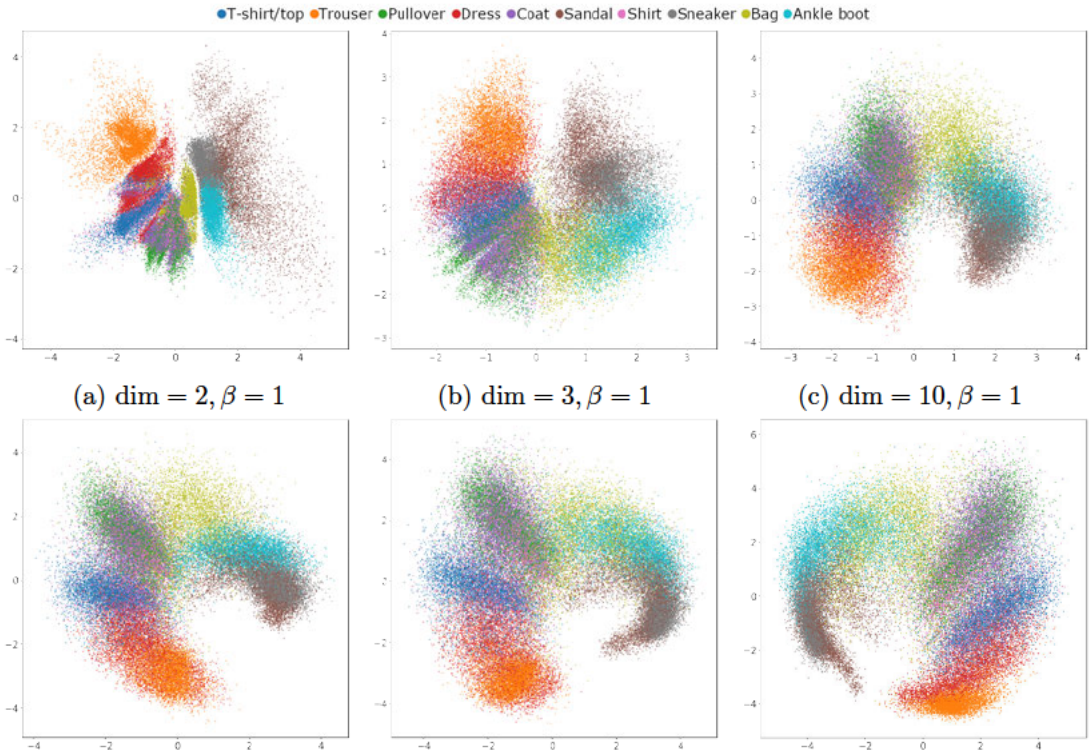


(a) Original (b) dim = 2 (c) dim = 3 (d) dim = 10 (e) dim = 20 (f) dim = 100

Abbildung 5.10: Vergleich steigender Dimensionen bei der Rekonstruktion ($\beta = 1$)

In Abbildung 5.10 ist zu sehen, dass mit steigender Anzahl der Dimensionen die Rekonstruktion besser wird. Das Bild in Zeile 2 erhält nach einer Dimension von dim > 20 den für die Klasse signifikanten Knopfverschluss. Die Klasse *Sneaker* in Zeile 3 wird dabei in (b) bis (d) als *Ankle boot* oder teilweise als *Sandal* rekonstruiert. Erst ab (e) wird ein *Sneaker* nahe des Originals daraus.

5.3.2 Betrachtung des latenten Raumes



(a) dim $= 2, \beta = 1$ (b) dim $= 3, \beta = 1$ (c) dim $= 10, \beta = 1$

(d) dim $= 20, \beta = 1$ (e) dim $= 50, \beta = 1$ (f) dim $= 100, \beta = 1$

Abbildung 5.11: Vergleich steigender Dimensionen mit PCA

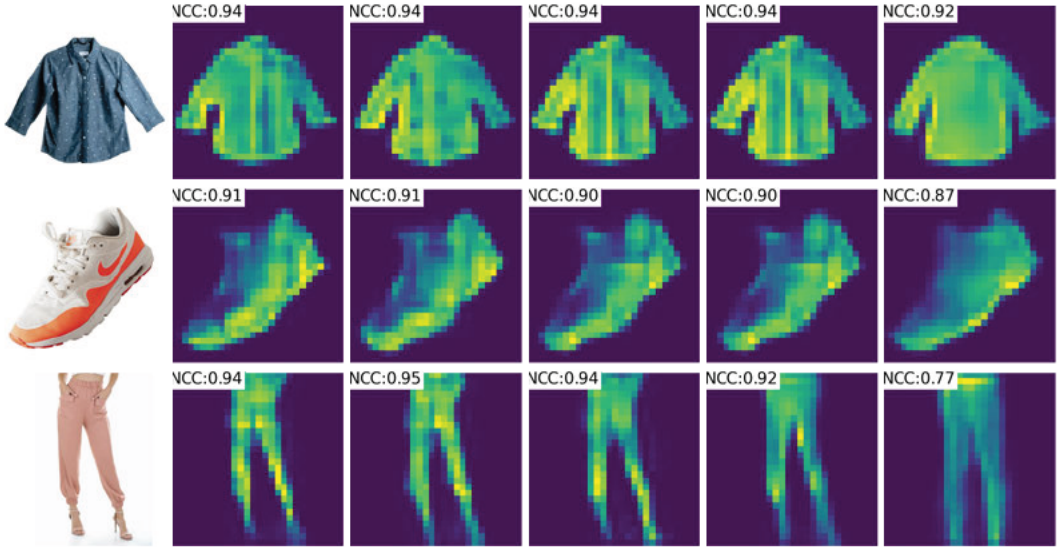
In Abbildung 5.11 ist der latente Raum bei steigender Anzahl der Dimensionen zu sehen, der mit PCA auf 2D reduziert wurde. In (a) sind *Cluster* sehr dicht beieinander, ähnlich einem klassischen AE. Bereits im dreidimensionalen Raum (b) sieht man eine deutlich homogenere und ringförmige Verteilung aller Punkte. Während bei (c) die *Cluster* kreisförmig sind, sind diese bei (d) und (e) teilweise oval. In (d) kann man erkennen, dass sich die Klassen *Dress* und *Trouser* stärker voneinander differenzieren.

5.4 β-Regulierung

Wie im vorherigen Kapitel 5.3 zu sehen, wird bereits durch eine höhere Anzahl an Dimensionen im latenten Raum sowohl eine bessere Rekonstruktion als auch ein besseres *Disentanglement* erreicht. Eine weitere Möglichkeit bietet der β-VAE, welcher in Bezug zu Teilfrage **F4** untersucht werden soll.

5.4.1 Vergleich der Rekonstruktion

Abbildung 5.12 zeigt die Rekonstruktion bei erhöhtem β-Wert bei dim = 100. Obwohl der β-Wert indirekt die Gewichtung des Rekonstruktionsverlustes verkleinert, sind diese bis zu einem Wert von $\beta = 100$ relativ nahe am Original. Allerdings merkt man bei diesem Wert auch einen Detailverlust. Bei der rekonstruierter Klasse *Shirt* fehlt im Gegensatz zu den vorherigen Rekonstruktionen der Knopfverschluss. Das ist scheinbar das entgegengesetzte Verhalten, wie es bei den Dimensionen in Abbildung 5.10 der Fall ist.

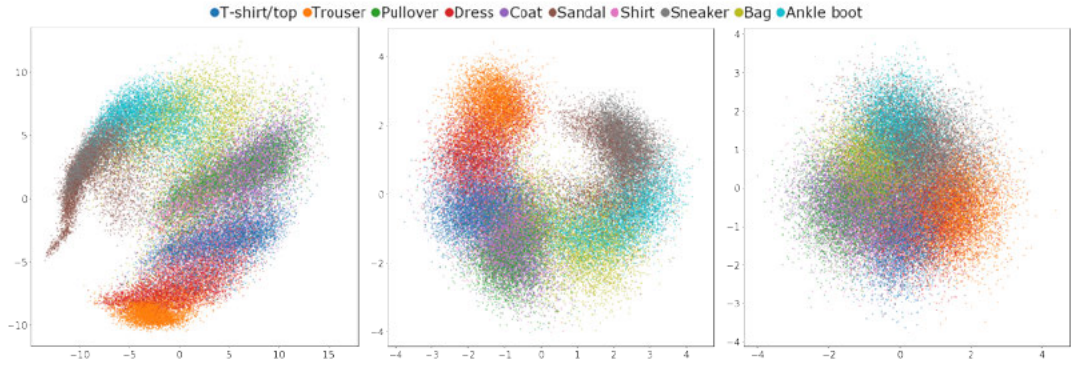


(a) Original (b) $\beta = 0.001$ (c) $\beta = 0.1$ (d) $\beta = 1$ (e) $\beta = 10$ (f) $\beta = 100$

Abbildung 5.12: Vergleich steigendem β-Wertes bei der Rekonstruktion (dim = 100)

5.4.2 Betrachtung des latenten Raumes

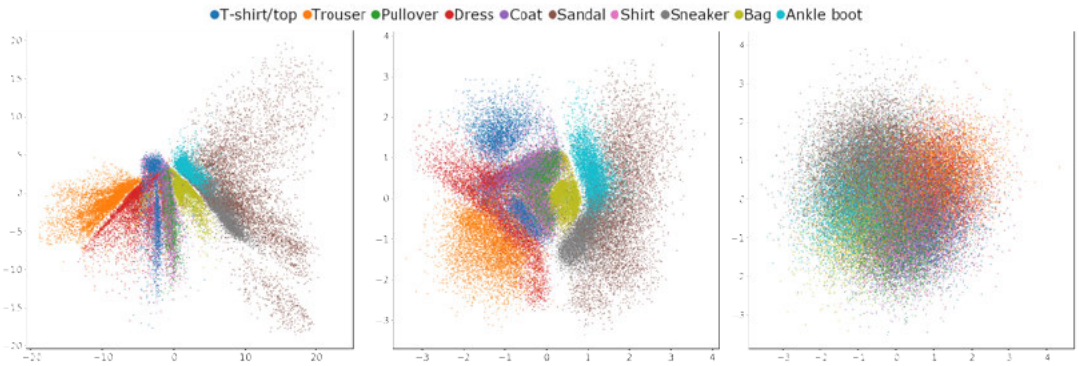
Abbildung 5.13 zeigt einen 100-dimensionalen latenten Raum und das Verhalten bei zunehmendem β-Wert. In (a) sind die Distanzen auf der Skala vergleichsweise groß und auch von der Verteilung her ist der latente Raum nahe an einem klassischen AE. Dennoch sind einige Bereiche der Cluster sichtbar voneinander differenziert, wie z.B. die Klassen *Trouser* und *Dress*. Im Gegensatz dazu zeigt (b) eine homogene Verteilung der Cluster, die in etwa den gleichen Radius aufweisen. Die ähnlichen Klassen sind dabei überlagert wie z.B. *Coat*, *Pullover* und *Shirt*. In (c) überlagern sich alle Cluster bei $\beta = 100$.



(a) $\beta = 0.01, \dim = 100$ (b) $\beta = 10, \dim = 100$ (c) $\beta = 100, \dim = 100$

Abbildung 5.13: Vergleich des β-Faktors in 100D mit PCA

Analog dazu ist der Effekt im zweidimensionalen Raum in Abbildung 5.14 zu sehen. Auch dort konvergieren die latenten Koordinaten zu einer homogenen Punktewolke.



(a) $\beta = 0.001, \dim = 2$ (b) $\beta = 10, \dim = 2$ (c) $\beta = 100, \dim = 2$

Abbildung 5.14: Vergleich des β-Faktors in 2D mit PCA

5.5 Varianz

Da ein VAE von der Architektur her ein probabilistisches Model ist und ein Punkt im latenten Raum zufällig gesetzt wird, ist das Resultat bei identischem Eingangsbild nicht deterministisch. Es soll daher in Bezug auf die Teilfrage **F5** untersucht werden, inwieweit diese Zufälligkeit Einfluss auf die Klassifikation hat. Je Evaluationsbild werden dabei 1000 Inferenzen angestoßen, die mit Hilfe von Box-Plots verteilt auf die Klassen dargestellt werden. Durch diese Methode sollen die Varianzen im Score in Abhängigkeit zum β-Wert ermittelt werden.

Bei den Versuchen konnten mit $\beta \leq 0.1$ keine nennenswerten Auffälligkeiten erkannt werden, da die Varianzen in den getesteten latenten Dimensionen bis dim = 100 nur geringfügig waren. Ebenso hatten diese Ergebnisse keinen negativen Einfluss auf die Klassifikation. Von daher werden diese nachfolgend nicht aufgeführt.

Tabelle 5.1 zeigt an einem fiktiven Beispiel, wie sich eine solche Varianz bei der Klassifizierung verhalten kann. Es ist zu sehen, dass sich die Klasse mit dem höchsten Score je Inferenz ändern kann. Dadurch können aufgrund des zufälligen *Samplings* falsche Klassifizierungen auftreten.

Klasse	Score
T-shirt/top	**0.7**
Shirt	0.2
Bag	0.1

(a) Inferenz 1

Klasse	Score
T-shirt/top	0.1
Shirt	**0.8**
Bag	0.1

(b) Inferenz 2

Klasse	Score
T-shirt/top	0.2
Shirt	0.3
Bag	**0.5**

(c) Inferenz 3

Tabelle 5.1: Beispielhafte Auswirkung der variierenden Klassifikation

Abbildung 5.15 zeigt am Beispiel der Klasse *Bag* der Kategorie B eine sehr große Varianz bei höheren β-Werten. Bei einem Wert von $\beta = 10$ schwanken die Klassifikationen zwischen den Klassen *Bag* und *Ankle boot* fast über die komplette Score-Skala hinsichtlich Minimum und Maximum. Auch die Quartile weisen Überlagerungen auf, sodass hier eine deterministische Klassifizierung nicht möglich ist. Bei einem Wert von $\beta = 100$ wird die richtige Klasse häufiger erkannt als mit $\beta = 10$, dennoch gibt es auch hier Varianzen. Diese überschneiden sich zwar nicht so stark, dennoch ist es möglich, dass die falsche Klasse *Dress* erkannt wird.

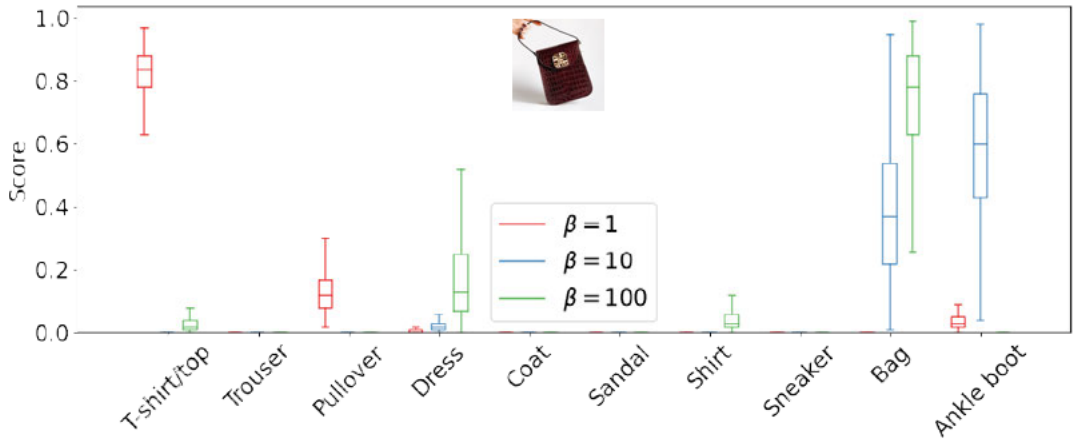


Abbildung 5.15: Varianz der Klasse *Bag* bei 1000 Inferenzen je β (100D)

Das nächste Beispiel in Abbildung 5.16 zeigt die Klasse *Tshirt/top* der Kategorie C. Während $\beta = 1$ einen annähernd perfekten Score hat, variieren die anderen Werte sehr stark. Es fällt auf, dass sich bei $\beta = 10$ der Score über mehrere Klassen verteilt. Obwohl die Wahrscheinlichkeit die korrekte Klasse zu erkennen am höchsten ist, sind die anderen nicht zu vernachlässigen. Bei $\beta = 100$ verschiebt sich die am häufigsten erkannte Klasse in Richtung *Shirt*. Diese Ähnlichkeit konnte bereits in Abbildung 5.6 beobachtet werden. Auch dort ist ein anderes Beispiel gleichen Klasse *T-shirt/top* zu sehen.

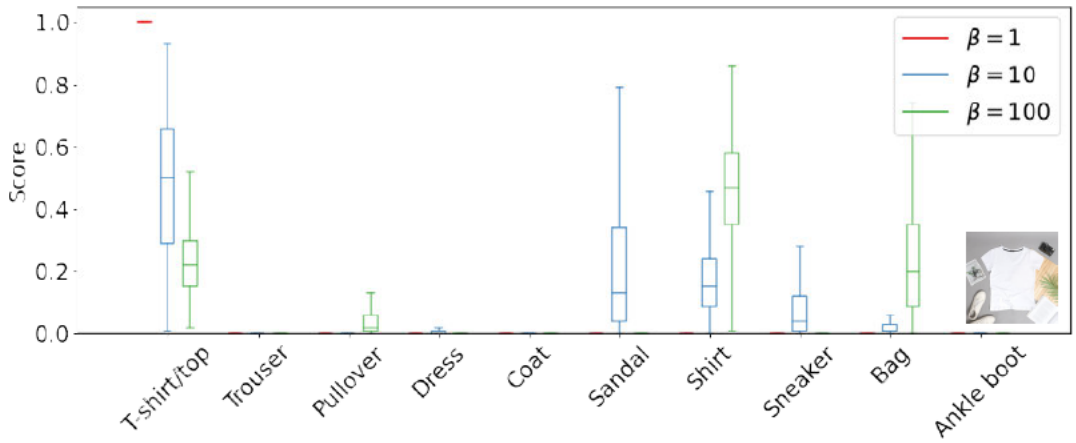


Abbildung 5.16: Varianz der Klasse *Tshirt/top* bei 1000 Inferenzen je β (100D)

Es gibt zwar mehrere solcher Ausreißer mit hoher Varianz, allerdings ist das nicht der Normalfall. Von den Versuchen her ist diese bei gut funktionierenden Bildern der Kategorie A in der Regel vernachlässigbar. Daher scheinen nur schwierige Bilder an der Grenze zu anderen Clustern betroffen zu sein, was einen Mehrwert für die Interpretation liefert.

6 Evaluierung und Diskussion

In Kapitel 5 wurden die Ergebnisse der Experimente vorgestellt. Diese werden nachfolgend evaluiert und die Teilfragen aus Kapitel 1.2 beantwortet. Abschließend wird die Forschungsfrage dieser Arbeit beantwortet.

6.1 Evaluierung der Beispielinterpretationen

In Kapitel 5.1 wurden die Ergebnisse in leicht sowie schwierig interpretierbare Beispiele unterteilt. Erstere beschreiben, wie das Modell nützliche Informationen geben kann, um eine bestimmte Problematik zu verstehen. Letztere, wo die Schwierigkeiten liegen. Diese Kategorien waren wiederum unterteilt in richtiger und falscher Klassifizierung.

Im Beispiel 1 spricht alles für die richtige Klassifizierung. Betrachtet man die Euklid-NCC-Relation aus Abbildung 5.2, so fällt eine starke Überlappung im Zentrum des *Bag*-Clusters auf. Daraus kann man schließen, dass dieses Beispiel der Klasse eher zu den besseren gehört. Ansonsten ist das ein Musterbeispiel und bietet dahingehend keine große Diskussionsgrundlage.

Für Beispiel 2 wurden 3 Dimensionen gewählt, was vergleichsweise wenig ist. Es lässt sich aber anhand des latenten Raumes sagen, dass weniger Dimensionen leichter zu interpretieren sind. Besonders an dem Beispiel sieht man sehr gut anhand der verschiedenen Nachbarn, dass sich die beiden Klassen aus „Sicht“ der Modells überschneiden. Ein Ansatz der Fehlerbehebung könnte das Hinzufügen von mehr Dimensionen sein.

Beispiel 3 ist hingehen schwieriger zu interpretieren. Obwohl Rekonstruktion und Klassifizierung nahezu perfekt sind, führen die jeweiligen Nachbarn (Euklid und NCC) zu gemischten Aussagen. Gerade in Euklid-NCC-Relation sieht man eine starke Überlagerung der Klassen, was jedoch nicht mit dem Ergebnis der *Interpretationskanäle* 1 und 2 korreliert. Das macht das wirkliche Verständnis des KNNs an diesem Beispiel schwierig.

Beispiel 4 ist ebenfalls schwierig zu interpretieren, da alle *Interpretationskanäle* widersprüchlich zueinander sind. Die Überlagerungen aus der Euklid-NCC-Relation deuten aber darauf hin, dass das Beispiel nicht richtig eingeordnet werden kann. Da dieses Bild als Kategorie C eingestuft wurde, ist dieses Verhalten zu erwarten. Dennoch können aus den Ergebnissen keine Handlungsempfehlungen gezogen werden, außer dass die Trainingsdaten um solche schwierigen Beispiele ergänzt werden sollten.

Insgesamt haben die Beispielinterpretationen gezeigt, dass das Modell teilweise eine gute Ergänzung für den Klassifikator ist, wenn der Bezug zu den gesamten Trainingsdaten betrachtet werden soll. Die verschiedenen *Interpretationskanäle* können die leicht interpretierbaren Beispiele gut beschreiben, bei den anderen fehlt der Detailgrad. Daher ist die Interpretierbarkeit abhängig vom Beispiel.

6.2 Kritik an den Evaluationskriterien

Das Ziel der Evaluationsbilder aus Kapitel 4.3.3 war es, den Schwierigkeitsgrad für das Netz sukzessive zu erhöhen. Die Beobachtungen haben ergeben, dass es mehr Einflussfaktoren zur Evaluierung des Models gibt. Daher liefern die Evaluationskategorien nur teilweise die erwartete Aussagekraft und unterliegen der subjektiven Wahrnehmung des Autors.

Nichtsdestotrotz ist Interpretierbarkeit laut der Definition in Kapitel 3.1.1 im Wesentlichen die subjektive Wahrnehmung eines Menschen. Daher ist die Qualität der Beispiele in dieser Arbeit zunächst zweitrangig. Um das zu verbessern, könnten entweder künstliche Daten (*Toy Data*) erzeugt werden, wie in [79], oder die Versuche an Daten der realen Welt durchgeführt werden.

6.3 Evaluierung der Bildähnlichkeit

Der Vergleich mit NCC wird in der Rekonstruktion, den Bildnachbarn im latenten Raum und im kompletten Trainingsset an sich angewandt. Jeder dieser Anwendungsfälle hat dabei seinen eigenen Interpretationsgehalt, der nachfolgend evaluiert wird.

NCC in Rekonstruktion

NCC wird bei der Rekonstruktion genutzt, um die VAE-eigene „Verschwommenheit“ auszugleichen und einen weiteren Kanal zur Interpretation zu geben. In den Versuchen hat sich herausgestellt, dass ein höheres β als auch niedrige Dimensionen Nachteile für den Klassifikator und die Bildnachbarschaft mit sich gebracht hat. Der Einsatz von NCC hat sich daher in der Rekonstruktion als sinnvoll erwiesen. Ausnahmen sind minimale Unterschiede, wie bei den Klassen *Trouser* oder *Dress*. Hierbei kann es schwierig werden, diese Unterschiede mit NCC zu interpretieren.

NCC bei Bildnachbarn

Der Einsatz von NCC bei Bildnachbarn ist größtenteils sinnvoll. Besonders im hochdimensionalen Raum können so Nachbarn der falschen Klasse besser erkannt werden. Gerade in Verbindung mit dem Euklid-NCC-Diagramm lassen sich so Erkenntnisse gewinnen. Jedoch kann es auch hier zu sehr ähnlichen Bildern kommen, die sowohl mit der euklidischen Distanz als auch mit NCC schwierig unterschieden werden können, wie z.B. *T-Shirt/top* mit *Shirt*.

NCC im Traingsset

Die Idee NCC unabhängig vom Modell auf die Trainingsdaten anzuwenden scheint zwar gut zu funktionieren, unterliegt aber strengen Richtlinien, die beachtet werden müssen. Zum einen sagt es nichts über das Modell an sich aus, außer dass es bei Abweichungen zu den anderen Interpretationskanälen vermutlich nicht gut genug trainiert wurde. Zum anderen werden Rotationen, Skalierung, Verzerrungen oder ähnliche Manipulationen des Bildes nicht gut erkannt. Eine Ausnahme bildet Abbildung 5.7c, bei der die Skalierung einen interpretierbaren Mehrwert bietet. Allerdings muss man hier darauf achten, dass die dunkle Farbe der Klasse *T-Shirt/top* auch einen relevanten Einfluss hat.

Es ist daher davon auszugehen, dass bei realen Daten mit besonders vielen Details diese Methode nicht funktioniert. Hierfür müsste die ursprüngliche Idee von NCC hinsichtlich der Bildabtastung mit *Template Matching* in das Modell integriert oder ein anderer Ansatz gewählt werden.

6.4 Rekonstruktion gegen Bildnachbarn

Die Qualität der Rekonstruktion als auch die Genauigkeit der Klassifizierung hängen stark von der Anzahl der Dimensionen im latenten Raum ab. Gleichzeitig nimmt dann aber die Qualität der Bildnachbarn ab. Von daher ist abzuwägen, inwieweit diese Faktoren zur Interpretierbarkeit beitragen.

Die Rekonstruktion mit Einberechnung des NCC ist bereits ein guter Indikator für eine gute oder falsche Klassifizierung, da diese Werte korrelieren. Jedoch ist das bei komplexeren Bildern teilweise schwierig zu erkennen, wie in Beispiel 4 zu sehen. Der NCC-Wert bringt dabei bedingt einen Mehrwert, da nicht bestimmt werden kann, wo ein guter Schwellenwert liegt. Ebenso ist dadurch die Idee der Arbeit, Eingangsbilder mit dem Trainingsset in Verbindung zu bringen, nur bedingt erfüllt. Aufgrund des generativen Modells ist zwar implizit eine Verbindung vorhanden, aber der Bezug zu den initialen Bilddateien müsste bei der Rekonstruktion stärker hervorgehoben werden. Das kann beispielsweise mit Interpolation erreicht werden.

Die Bildnachbarn auf der anderen Seite sind bei der passenden Anzahl der latenten Dimensionen ein guter Bezugspunkt zu den Trainingsdaten. Die Frage ist dann aber, ob die nächsten Bilder eben wirklich die einflussreichsten sind. Auch in Bezug auf entsprechende Bildbereiche lassen sich nicht viele Rückschlüsse ziehen. Zum einen könnte hier die eingangs genannte Arbeit [53] helfen, diese Invarianzen bzw. wichtigen Bereiche ausfindig zu machen. Zum anderen muss das probabilistische *Sampling* berücksichtigt werden, indem mehrere Inferenzen durchgeführt werden und so ein Mittelwert der Bildnachbarn gefunden wird.

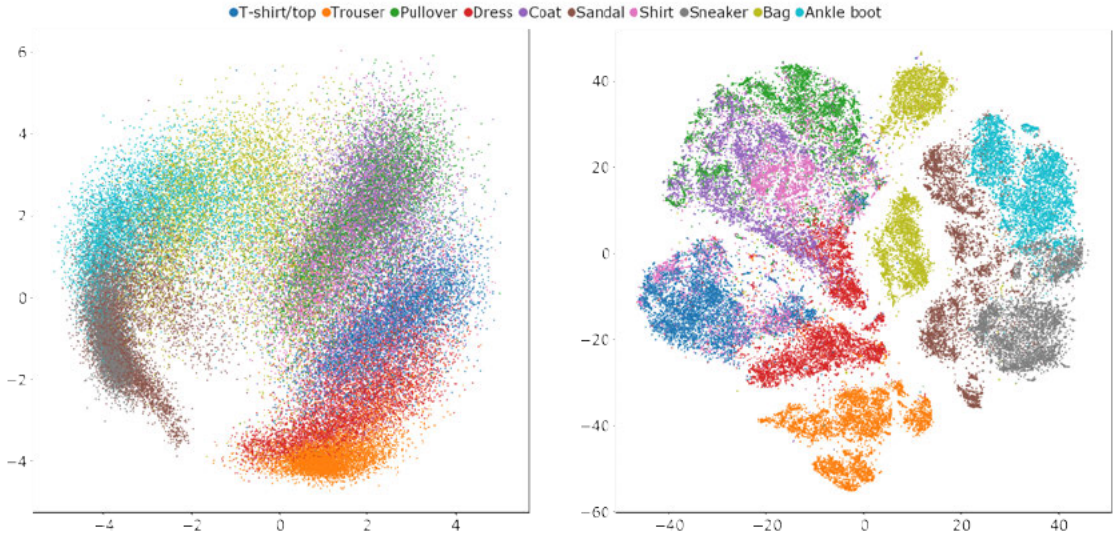
6.5 Einfluss auf den Klassifikator

Ein Anwendungsfall von AEs im Generellen ist es durch ein Pretraining eine bessere Genauigkeit zu erlangen. Auch mit einem VAE in Verbindung mit einem Klassifikator in einem *verbundenen* Training konnte dieser Vorteil genutzt werden. Der Klassifikator reguliert somit die Verteilung im latenten Raum, während der VAE ein Regulator gegen *Overfitting* zu sein scheint. In Anbetracht von XAI ist das Modell dadurch *intrinsisch* interpretierbar, ohne dass größere Einbußen in der Klassifizierung festzustellen sind.

Jedoch ist der Einfluss auf den Klassifikator noch nicht aussagekräftig genug, da hier die gleiche Netzarchitektur verglichen wurde. Aktuelle Netzwerke erreichen dabei mit *Fashion-MNIST* eine Genauigkeit von bis zu 96.91%[1]. Da hierbei oft spezielle Netzarchitekturen genutzt werden, sollte der VAE im Nachhinein in solche implementiert werden. Durch gesonderte Experimente mit diesen würde sich dann ein besserer Überblick ergeben, wann ein VAE als interpretierbarer Zusatz Sinn macht und wann nicht. Aufgrund dieser Faktoren wurde in dieser Arbeit der Einfluss nicht genauer untersucht.

6.6 Evaluierung der Dimensionen

Der Fluch der Dimensionalität [9] wird unter anderem in [10], [2] und [37] thematisiert und ist ein allgemein bekanntes Problem. Der Grund hierfür ist, dass der Zustandsraum für jede weitere Dimension exponentiell wächst. Das hat wiederum zur Folge, dass die Trainingsdaten oft nicht genügend Features aufweisen, um diesen „leeren“ Raum auszufüllen.



(a) PCA (b) t-SNE *(perplexity=50)*

Abbildung 6.1: Vergleich 2D-Repräsentation des 100D latenten Raums.

Die negativen Einflüsse hoher Dimensionen sind insbesondere bei der Nutzung der euklidischen Distanz in Kapitel 5 zu beobachten. Um den Sachverhalt besser verstehen

[1] https://paperswithcode.com/sota/image-classification-on-fashion-mnist

zu können, wurde der latente Raum mit PCA untersucht. Man sieht in der Abbildung 6.1 einen Vergleich der Visualisierung zwischen PCA und t-SNE. Darin erkennt man bei beiden Darstellungen insbesondere die Überlagerung bestimmter Klassen, darunter *Pullover*, *Coat* und *Shirt*. Abweichend dazu sind aber die Klassen *Sandal* und *Sneaker* mit PCA überlagert, während sie bei t-SNE entzerrt sind. Grund hierfür ist die Funktionsweise der Methoden. Während bei PCA auf die wichtigsten Komponenten reduziert wird, verteilt t-SNE die Daten neu.

Es kann auf den ersten Blick naheliegend sein, die *t-SNE*-Repräsentation für die Distanzfunktionen zu nutzen. Das ist jedoch eine Falschannahme, da die vom VAE ermittelten Koordinaten eines Eingangsbildes nicht denen der Repräsentation entsprechen. Das sieht man zum einen an der größeren Skala, zum anderen an der anders platzierten *Cluster*.

Des Weiteren hat die Reduktionen auf die Hauptkomponenten mit PCA den Nachteil, dass in der 2D-Darstellung zwar Überlagerungen zu sehen sind, diese aber hinsichtlich anderer Dimensionen nicht unbedingt zutreffen müssen. Um ein differenzierteres Bild zu bekommen, müssten mehrere Dimensionen gegeneinander verglichen werden. Dies wird beispielsweise in der eingangs erwähnten Arbeit [53] verwendet, um Invarianzen im latenten Raum zu ermitteln.

Insgesamt sollten diese Visualisierungen unter Vorwand als Erklärungsgrundlage genommen werden. Das liegt daran, dass *t-SNE* ein nicht-deterministisches Verfahren ist und die Nachbarschaft der Datenpunkte nur bedingt stimmt. Auch in [6] wird davor gewarnt, sich zu sehr auf entsprechende Methoden zur Dimensionsreduktion zu verlassen. In diesem Fall sind die Darstellungen aber nachvollziehbar in Bezug auf die Ergebnisse und tragen dadurch einen Teil dazu bei den *Fluch der Dimensionalität* zu verstehen.

6.7 Evaluierung der β-Regulierung

Die Ergebnisse aus Kapitel 5.4 haben gezeigt, dass die Regulierung des β-Wert vorsichtig gewählt werden muss. Einen Ansatz zur Balancierung bieten Asperti und Trentin in [5], wobei dort der Fokus auf einer guten Rekonstruktion liegt. Zwecks Distanzberechnung, Bildnachbarschaft als auch der Varianz scheinen hier jedoch kleinere Werte besser zu funktionieren. Da sich ein an 0 annähernder β-Faktor zunehmend wie ein klassischer AE verhält, stellt sich die Frage welchen Vorteil ein VAE bietet.

Dieser liegt zum einen an der Repräsentation des latenten Raumes. Die gezeigten Darstellungen, auch wenn sie über PCA reduziert wurden, bieten bereits auf den ersten Blick eine interpretierbare Verteilung der Daten. Es kann direkt gesehen werden, welche Klassen ähnlich sind oder sich überlagern. Anders als beim AE sind diese Cluster geordnet und wirken nicht willkürlich. Aufgrund der gleichmäßigen Abstände „springen“ die Ergebnisse von Distanzfunktionen weniger.

Zum anderen liegt ein Vorteil darin, die generativen Eigenschaften eines VAEs zu nutzen. Diese wurden in dieser Arbeit nicht behandelt, können aber interessante Möglichkeiten bieten, ähnlich zu den in Kapitel 2 erwähnten Arbeiten.

6.8 Signifikanz der Varianz

Die Beispiele aus Kapitel 5.5 haben gezeigt, dass die Varianz hauptsächlich Bilder nahe an den Grenzen der *Cluster* betrifft. Dadurch können diese zwischen zwei oder mehr Klassen variieren. Idealerweise könnte man bei den Ergebnissen davon ausgehen, dass die größte Verteilung zu einer Klasse hin auch die entsprechende Klassifizierung darstellt. Das ist jedoch am gezeigten Beispiel der Klasse *T-shirt/top* nicht pauschal zu behaupten. Das liegt daran, dass jeder β-Faktor ein neues Training sowie eine neue Inferenz darstellt und daher nicht deterministisch ist. Eine Schwankung bei den Ergebnissen, z.b. $\beta = 1$ mit richtiger Klassifizierung, $\beta = 10$ mit falscher und $\beta = 100$ wieder mit richtiger, wie am Beispiel in Abbildung 5.16 im vorherigen Kapitel zu sehen, kann durchaus eintreten. Das kann auf unterschiedliche Trainings zurückzuführen sein, weshalb auch diese Varianz zusätzlich untersucht werden müsste.

Peltola versucht in seiner Arbeit [56] die KL-Divergenz mit der Idee von *LIME* zu verbinden. Dabei ist das Ziel, Varianzen in die Erklärung einzubeziehen. In seinem Beispiel demonstriert er es an dem *MNIST*-Datensatz. Genauer, welche Kurven zwischen einer 8 und 3 besonders ausschlaggebend für die jeweilige Klassifizierung sind. Dadurch könnte der zuvor als Nachteil deklarierte Aspekt des Modells mit lokaler Interpretierbarkeit ausgebessert, als auch die Varianz an sich bei höherem β nutzbar gemacht werden. Anzumerken ist, dass es noch mehr Versuche braucht, um den beschriebenen Vorteil des *KL-LIME*-Ansatzes zu verifizieren.

Eine Möglichkeit die Varianz gering zu halten, könnte eine Intervention in das *Sampling* der Inferenz sein. Statt nach einer normal-verteilten Wahrscheinlichkeit einen zufälligen

Punkt im Cluster auszuwählen, kann direkt das Zentrum gewählt werden. Dadurch sollten zum einen die Nachbarn besser erkannt, als auch die Klassifizierung besser werden.

Auf der anderen Seite kann die Varianz in Anbetracht der Interpretierbarkeit jedoch positiv sein, was auch der Grund für die Verwendung eines VAEs ist. Das äußerst sich dadurch, dass erkannt werden kann, welche Klassen ähnlich sind. Dadurch kann entweder auf eine bessere Differenzierung dieser Klassen zwecks Fehleranalyse und -behebung angestrebt werden oder diese Ähnlichkeiten als fester Bestandteil des Netzes übernommen werden.

6.9 Beantwortung der Forschungsfrage

In Kapitel 1.2 wurde die Frage formuliert:

Kann mit Hilfe eines VAE die Entscheidung eines Klassifikators interpretiert werden?

Bedingt. Es können nützliche Informationen aus den *Interpretationskanälen* gezogen werden und das Bild kann im Kontext zu den anderen Trainingsdaten betrachtet werden. Auch die zu Hilfenahme von NCC zur Bestimmung eines zweiten Sets von Nachbarn sowie das Relations-Diagramm dieser zu den euklidischen Nachbarn kann den latenten Raum simulieren. Dieser ist sonst nur in 2D- und 3D-Visualisierungen oder in nachträglichen Verfahren wie PCA oder t-SNE für den Menschen ersichtlich.

Ein Problem stellt die unpräzise Interpretierbarkeit dar. Ein Beobachter muss die verschiedenen benachbarten Bilder vergleichen und versuchen das Merkmal zu identifizieren, was für die Klassifizierung ausschlaggebend war. Dadurch kann er die Entscheidungsgründe des Netzes erahnen, allerdings fehlt ein klarer Beweis dafür. Im Gegensatz dazu sind die lokalen XAI-Methoden aus Kapitel 3.1.4 wie LIME sehr gut darin punktuell die Gründe einer Falschklassifizierung zu zeigen. Daher kann eine Kombination mit einer solchen Methode dabei helfen, zunächst die Pixelbereiche einzugrenzen, die für die Klassifizierung am meisten beitragen, und diese dann in einem zweiten Schritt mit den Bildnachbarn zu vergleichen.

Die teilweise unzureichenden Ergebnisse der Bildähnlichkeit über NCC könnten mit einem Siamesischen Neuronalen Netz (SNN) verbessert werden. Oft werden diese genau zu diesem Zweck eingesetzt, wie in [3] und [18] gezeigt. Dabei wird unter anderem die

Contrastive Loss Function eingesetzt, um schon während des Trainings Bildähnlichkeiten einzubeziehen. In [12] kombinieren die Autoren die NCC-Bildähnlichkeit sogar mit einem SNN am Beispiel von Mikroskop-Bildern von Gehirnzellen.

Nichtsdestotrotz stellt sich die Frage, inwieweit zusätzliche KNNs zur Interpretierbarkeit beitragen, wenn diese wiederum selbst als „Blackboxen“ zu bezeichnen sind. Utkin et al. stellen in [74] eine entsprechende Methode vor, SNNs im Speziellen interpretierbar zu machen. Dennoch muss abgewogen werden, wie viele miteinander verbundene Systeme noch verhältnismäßig sind, um einen Klassifikator interpretierbar zu machen.

Abschließend kann behauptet werden, dass ein VAE im Verbund mit einem Klassifikator als ein *intrinsisches* auf Trainingsdaten basiertes Interpretationsmodell zwecks Verständnis, Fehleranalyse und Pretraining einen Mehrwert bieten kann. Mehrere *Interpretationskanäle* können wie eine Jury agieren und den „Entscheidungsprozess“ des Modells aus mehreren Sichtweisen erklären, ähnlich wie beim *Ensemble Learning* [29]. Wenn ein nicht-deterministischer Klassifikator für die Anwendung ausreicht und ein auf Wahrscheinlichkeit basiertes Modell geeignet ist, so ist der vorgestellte Ansatz vielversprechend. Gerade bei Systemen, die menschliche Experten in einer Domäne (z.B. Ärzte) in der Entscheidungsfindung unterstützen sollen, sind interpretierbare Ergebnisse wichtiger als die Genauigkeit. Die genannten Nachteile müssen dabei aber einbezogen oder zuvor verbessert werden.

7 Fazit

7.1 Zusammenfassung

In dieser Arbeit wird untersucht inwieweit die Kombination aus Klassifikator und VAE zur Interpretierbarkeit ersteren beiträgt. Zuerst werden in einem Y-förmigen Model Klassifikator und Decoder gleichzeitig in einem *verbunden* Training mit dem *Fashion-MNIST*-Datensatz trainiert. In einem zweiten Schritt wird die resultierende Inferenz genutzt, um die latenten Koordinaten für Distanzberechnungen der Trainingsdaten zu speichern.

Für die Evaluierung des Modells wurden Bilder nach bestimmten Schwierigkeitsgraden ausgewählt. Bei fast allen Evaluationsbildern zeigen die *Interpretationskanäle* eine sinnvolle Ergänzung zum Klassifikator. Bei einer korrekten Klassifizierung mit einem niedrigen Score bestärken die anderen Kanäle das Ergebnis. Bei falschen Klassifizierungen lässt sich der Grund für das Ergebnis meistens an den benachbarten Bildern direkt erkennen. Bei komplett falschen Ergebnissen mit besonders schwierigen Evaluationsbildern konnte der niedrige NCC-Wert einen Hinweis auf ein *Ausreißer*-Verhalten liefern. Besonders die Euklid-NCC-Diagramme geben aussagekräftige Erkenntnisse der Trainingsdaten in Bezug auf das Eingangsbild, wenn die jeweiligen Nachteile für Dimensionen und Bildähnlichkeit mit einbezogen werden.

Allerdings ist diese globale Methodik ungenau im Vergleich zu lokalen XAI-Methoden, die ein pixelgenaue Analyse der relevanten Bildbereiche ermöglichen. Gerade bei komplexeren Bildern kann es schwierig werden alleine durch den augenscheinlichen Vergleich Gemeinsamkeiten oder Unterschiede zu erkennen.

Bei höheren Dimensionen werden die Distanzfunktionen ungenauer, sodass die Bildnachbarn öfter falsch sein können. Allerdings kann dieser Nachteil bis zu einem gewissen Grad mit NCC ausgeglichen werden, indem so ein zweiter Faktor der Bildähnlichkeit unabhängig vom Model genutzt wird. Hierbei ist NCC aber auch mit Nachteilen verbunden, die mit einem neuronalen Bildvergleich über SNNs negiert werden könnten.

Die Versuche mit dem β-Wert und der damit verbundenen Varianz des VAEs haben gezeigt, dass ein zu hoher Wert problematisch ist. Das zeigt sich dadurch, dass sich die Klassen im latenten Raum stärker überlagern und dadurch unterschiedliche Ergebnisse entstehen können. Gleichzeitig führen niedrige Werte dazu, dass der VAE sich wie ein AE verhält. Eine Regulierung muss daher je nach Anwendung mit Bedacht vorgenommen werden.

Insgesamt ist das in dieser Arbeit vorgestellte Modell ein hilfreicher Ansatz, ein Eingangsbild in Relation zu den genutzten Trainingsbildern zu sehen. Dadurch lassen sich insbesondere schwierig zu unterscheidende Klassen besser ausmachen, um das Modell dahingehend zu verbessern.

7.2 Ausblick

Da in dieser Arbeit der Fokus auf eine breitere Untersuchung der Interpretierbarkeit von Klassifikatoren mit VAEs gelegt wird, gibt es dementsprechend mehrere Themen, die noch im Detail untersucht werden können. Nachfolgend sind Vorschläge und Ideen aufgelistet, die in zukünftigen Arbeiten behandelt werden können.

7.2.1 Neuronaler Bildvergleich

Wie bereits in Kapitel 6.9 diskutiert, könnte ein alternatives Verfahren zur Bestimmung der Bildähnlichkeit mit einem SNN eine bessere Unterscheidung der Daten mit sich bringen. Falls das entsprechende SNN nicht die gleiche Problematik mit den Dimensionen aufweist wie die euklidische Distanz, so kann das Ergebnis als Alternative genutzt werden. Ein Tutorial für einen solchen Bildvergleich kann unter [28] gefunden werden.

7.2.2 Evaluierung mit realen Daten

Fashion-MNIST ist ein vergleichsweise kleiner Datensatz, bei dem wenig Speicherplatz und Rechenleistung für die Berechnungen benötigt werden. Bei Problemstellungen mit mehr und größeren Bildern wird es mit den hier verwendeten Methoden definitiv zu Schwierigkeiten kommen. Um größere Datensätze wie *ImageNet* [22] zu werwenden, sollte das Modell deshalb hinsichtlich folgender Punkte neu optimiert werden:

- Die Speicherung der Bildreferenz zum latenten Raum erfolgt in dieser Arbeit in einer CSV-Datei. Bei sehr großen Datenmengen sollte hier auf eine Datenbank oder alternative Speicheroptionen gewechselt werden.
- *Data Augmentation*, d.h. die Menge der Trainingsdaten durch Rotation, Verzerrung, Generierung, etc. zu erhöhen, wurde hier nicht benutzt. Das sollte jedoch bei realen Problemen implementiert werden und es sollte darauf geachtet werden, die so erzeugten Bilder nur beabsichtigt für Distanzberechnungen zu nutzen.
- Die praktische Implementierung erfolgte über *JupyterHub*. Da diese Entwicklungsumgebung hinsichtlich Code-Wartbarkeit, Dateigrößen und Stabilität weniger für komplexere Anwendungen geeignet ist, sollte auch hier auf eine alternative Entwicklungsumgebung gewechselt werden, die dann die Rechenleistung per API nutzt.

7.2.3 Regulierung des Klassifikators

In dieser Arbeit wurde zwar der β-VAE hinsichtlich seines Einflusses untersucht, jedoch nicht der Klassifikator. Dieser kann durch einen λ-Faktor ähnlich zu [7] ebenfalls reguliert werden und im Zusammenspiel mit dem β-Faktor untersucht werden. Eine Frage dazu wäre, wie sich der latente Raum bei unterschiedlichen Gewichtungen verhält.

Eine entsprechende Gleichung sähe dann so aus:

$$L_{GESAMT} = \underbrace{(L_R + \beta \cdot L_{KL})}_{\text{Decoder}} + \underbrace{\lambda \cdot L_K}_{\text{Klassifikator}} \tag{7.1}$$

Weitergehend wurde auch ein allein stehender Klassifikator, wie in Kapitel 5.2 bereits herausgestellt, nicht genauer untersucht. Hier könnte tiefgreifender untersucht werden, welche Faktoren bei beiden Modellen eine Rolle spielen und mit welcher Metrik diese vergleichbar sind. Auch ein *Pretraining* kann interessant sein, indem zuerst der VAE und dann der Klassifikator mit Hilfe von *Transfer Learning* trainiert wird.

7.2.4 Visual Transformer

Alternativ zu der in dieser Arbeit verwendeten CNN-Architektur, könnten stattdessen *Visual Transformer* (ViT) [25] [14] eingesetzt werden. Diese haben aktuell, die große Menge an Trainingsdaten vorausgesetzt, eine bessere Performance als klassische CNNs.

Eine Verbindung mit dem VAE könnte zu interessanten Ergebnissen führen. Eine Inspiration zur Implementierung am Beispiel von *Layouts* kann in [4] gefunden werden.

7.2.5 Kombination mit lokalen XAI-Methoden

Wie bereits in der Diskussion unter 6.9 erwähnt, können lokale XAI-Methoden ergänzend verwendet werden. Dadurch kann das Defizit ausgebessert werden, die relevanten Bildbereiche zu erkennen. Der große Vorteil hierbei wäre, dass es eine Interpretierbarkeit von pixelgenauer Darstellung bis hin zur gesamtheitlichen Betrachtung aller Trainingsdaten möglich wäre. Auch die in Abschnitt 6.8 erwähnte Idee mit *KL-LIME* aus [56] kann hier einen lokalen Interpretationsansatz mit Hilfe der Varianz liefern.

7.2.6 Interpolation

Da VAEs generative Modelle sind, sind diese in der Lage zwischen Features zu interpolieren. In dieser Arbeit wurde diese Eigenschaft nicht behandelt. Jedoch nutzt [31] diese, um Bias in den Trainingsdaten zu finden, während [53] diese nutzt, um Invarianzen zu erkennen. Die Idee ist es einen weiteren (interaktiven) Interpretationskanal mit entsprechenden Variationen zu implementieren. Dadurch könnten nicht nur die n nächsten Nachbarn aus dem Trainingsset angezeigt werden, sondern auch die m nächsten Interpolationen mit einer bestimmten Distanz in eine bestimmte Richtung einer Feature-Dimension. Beispielhaft könnte man so den Übergang der schwierig differenzierbaren Klassen *T-shirt/top* und *Shirt* genauer untersuchen.

7.2.7 Evaluierung der Interpretierbarkeit

Interpretierbarkeit ist abhängig von der Domäne, der Anwendung und der interpretierenden Person selbst. In dieser Arbeit wurde diese nach menschlicher Intuition bewertet, was ein subjektiver Maßstab ist. Juri Zach hat in seiner Masterarbeit [80] eine neuronale Sprache entworfen, die eine Schnittstelle zwischen Mensch und Maschine bietet. Mit dieser kann die Qualität einer Interpretation gemessen werden kann. Der Fokus seiner Arbeit liegt dabei zwar auf visuellen Methoden und gemeinsamen *Konzepten* zwischen Mensch und KI, verteilt auf mehreren Netzschichten, jedoch könnte untersucht werden, ob eine Migration oder Anwendung auf diese Arbeit möglich ist.

Literaturverzeichnis

[1] Adadi, Amina ; Berrada, Mohammed: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). In: *IEEE Access* 6 (2018), S. 52138–52160

[2] Aggarwal, Charu C. ; Hinneburg, Alexander ; Keim, Daniel A.: On the Surprising Behavior of Distance Metrics in High Dimensional Spaces. In: *Proceedings of the 8th International Conference on Database Theory*. Berlin, Heidelberg : Springer-Verlag, 2001 (ICDT '01), S. 420–434. – ISBN 3540414568

[3] Appalaraju, Srikar ; Chaoji, Vineet: Image similarity using Deep CNN and Curriculum Learning. (2017), September

[4] Arroyo, Diego M. ; Postels, Janis ; Tombari, Federico: Variational Transformer Networks for Layout Generation. (2021), April

[5] Asperti, Andrea ; Trentin, Matteo: Balancing reconstruction error and Kullback-Leibler divergence in Variational Autoencoders. (2020), Februar

[6] Balestriero, Randall ; Pesenti, Jerome ; LeCun, Yann: Learning in High Dimension Always Amounts to Extrapolation. (2021), Oktober

[7] Bank, Dor ; Koenigstein, Noam ; Giryes, Raja: Autoencoders. (2020), März

[8] Barredo Arrieta, Alejandro ; Díaz-Rodríguez, Natalia ; Del Ser, Javier ; Bennetot, Adrien ; Tabik, Siham ; Barbado, Alberto ; Garcia, Salvador ; Gil-Lopez, Sergio ; Molina, Daniel ; Benjamins, Richard ; Chatila, Raja ; Herrera, Francisco: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. In: *Information Fusion* 58 (2020), S. 82–115. – ISSN 1566-2535

[9] Bellman, Richard: Dynamic programming. In: *Science* 153 (1966), Nr. 3731, S. 34–37

[10] Beyer, Kevin ; Goldstein, Jonathan ; Ramakrishnan, Raghu ; Shaft, Uri: When Is "Nearest Neighbor" Meaningful? In: Beeri, Catriel (Hrsg.) ; Buneman, Peter (Hrsg.): *Database Theory — ICDT'99*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1999, S. 217–235. – ISBN 978-3-540-49257-3

[11] Broniatowski, David A.: Psychological Foundations of Explainability and Interpretability in Artificial Intelligence. National Institute of Standards and Technology, apr 2021. – Forschungsbericht

[12] Buniatyan, Davit ; Macrina, Thomas ; Ih, Dodam ; Zung, Jonathan ; Seung, H. S.: Deep Learning Improves Template Matching by Normalized Cross Correlation. (2017), Mai

[13] Burgess, Christopher P. ; Higgins, Irina ; Pal, Arka ; Matthey, Loic ; Watters, Nick ; Desjardins, Guillaume ; Lerchner, Alexander: Understanding disentangling in β-VAE. (2018), April

[14] Caron, Mathilde ; Touvron, Hugo ; Misra, Ishan ; Jégou, Hervé ; Mairal, Julien ; Bojanowski, Piotr ; Joulin, Armand: Emerging Properties in Self-Supervised Vision Transformers. (2021), April

[15] Cetin, Irem ; Camara, Oscar ; Ballester, Miguel Angel G.: Attri-VAE: attribute-based, disentangled and interpretable representations of medical images with variational autoencoders. (2022), März

[16] Chen, Nutan ; Klushyn, Alexej ; Ferroni, Francesco ; Bayer, Justin ; Smagt, Patrick van der: Learning Flat Latent Manifolds with VAEs. In: *International Conference on Machine Learning 2020* (2020), Februar

[17] Chen, Ricky T. Q. ; Li, Xuechen ; Grosse, Roger ; Duvenaud, David: Isolating Sources of Disentanglement in Variational Autoencoders. (2018), Februar

[18] Chen, Xinlei ; He, Kaiming: Exploring Simple Siamese Representation Learning. (2020), November

[19] Chollet, François: *Convolutional Variational AutoEncoder (VAE) trained on MNIST digits*. Internet. Mai 2020. – URL https://keras.io/examples/generative/vae/

[20] Clough, James R. ; Oksuz, Ilkay ; Puyol-Anton, Esther ; Ruijsink, Bram ; King, Andrew P. ; Schnabel, Julia A.: Global and Local Interpretability for Cardiac MRI Classification. (2019), Juni

[21] Das, Arun ; Rad, Paul: Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. (2020), Juni

[22] Deng, Jia ; Dong, Wei ; Socher, Richard ; Li, Li-Jia ; Li, Kai ; Fei-Fei, Li: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition* Ieee (Veranst.), 2009, S. 248–255

[23] Dilokthanakul, Nat ; Mediano, Pedro A. M. ; Garnelo, Marta ; Lee, Matthew C. H. ; Salimbeni, Hugh ; Arulkumaran, Kai ; Shanahan, Murray: Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. (2016), November

[24] Doshi-Velez, Finale ; Kim, Been: Towards A Rigorous Science of Interpretable Machine Learning. (2017), Februar

[25] Dosovitskiy, Alexey ; Beyer, Lucas ; Kolesnikov, Alexander ; Weissenborn, Dirk ; Zhai, Xiaohua ; Unterthiner, Thomas ; Dehghani, Mostafa ; Minderer, Matthias ; Heigold, Georg ; Gelly, Sylvain ; Uszkoreit, Jakob ; Houlsby, Neil: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (2020), Oktober

[26] Dunn, Jack ; Mingardi, Luca ; Zhuo, Ying D.: Comparing interpretability and explainability for feature selection. (2021), Mai

[27] Ebers, Martin: Regulating Explainable AI in the European Union. An Overview of the Current Legal Framework(s). In: *SSRN Electronic Journal* (2021)

[28] Essam, Hazem ; Valdarrama, Santiago L.: *Image similarity estimation using a Siamese Network with a triplet loss*. Keras Website. März 2021. – URL https://keras.io/examples/vision/siamese_network/

[29] Ganaie, M. A. ; Hu, Minghui ; Malik, A. K. ; Tanveer, M. ; Suganthan, P. N.: Ensemble deep learning: A review. (2021), April

[30] Gao, Nicholas ; Wilson, Max ; Vandal, Thomas ; Vinci, Walter ; Nemani, Ramakrishna ; Rieffel, Eleanor: High-Dimensional Similarity Search with Quantum-Assisted Variational Autoencoder. (2020), Juni

[31] Gat, Itai ; Lorberbom, Guy ; Schwartz, Idan ; Hazan, Tamir: Latent Space Explanation by Intervention. (2021), Dezember

[32] Gatopoulos, Ioannis ; Tomczak, Jakub M.: Self-Supervised Variational Auto-Encoders. (2020), Oktober

[33] Ghojogh, Benyamin ; Ghodsi, Ali ; Karray, Fakhri ; Crowley, Mark: Stochastic Neighbor Embedding with Gaussian and Student-t Distributions: Tutorial and Survey. (2020), September

[34] Higgins, Irina ; Matthey, Loic ; Glorot, Xavier ; Pal, Arka ; Uria, Benigno ; Blundell, Charles ; Mohamed, Shakir ; Lerchner, Alexander: *Early Visual Concept Learning with Unsupervised Deep Learning*. 2016

[35] Hinton, Geoffrey ; Roweis, Sam: Stochastic Neighbor Embedding. In: *Advances in neural information processing systems* 15 (2003), S. 833–840. – URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.7959&rep=rep1&type=pdf

[36] Hisham, M.B. ; Yaakob, Shahrul N. ; Raof, R.A.A ; Nazren, A.B A. ; Wafi, N.M.: Template Matching using Sum of Squared Difference and Normalized Cross Correlation. In: *2015 IEEE Student Conference on Research and Development (SCOReD)*, 2015, S. 100–104

[37] Houle, Michael E. ; Kriegel, Hans-Peter ; Kröger, Peer ; Schubert, Erich ; Zimek, Arthur: Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? In: Gertz, Michael (Hrsg.) ; Ludäscher, Bertram (Hrsg.): *Scientific and Statistical Database Management*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010, S. 482–500. – ISBN 978-3-642-13818-8

[38] Hu, Huan ; Li, Jianzhong: Sublinear Time Nearest Neighbor Search over Generalized Weighted Manhattan Distance. (2021), April

[39] Ji, Tianchen ; Vuppala, Sri T. ; Chowdhary, Girish ; Driggs-Campbell, Katherine: Multi-Modal Anomaly Detection for Unstructured and Uncertain Environments. (2020), Dezember

[40] Kingma, Diederik P. ; Welling, Max: Auto-Encoding Variational Bayes. (2013), Dezember

[41] Kingma, Diederik P. ; Welling, Max: An Introduction to Variational Autoencoders. In: *Foundations and Trends in Machine Learning: Vol. 12 (2019): No. 4, pp 307-392* (2019), Juni

[42] Kommision, Europäische: *Neue Vorschriften für künstliche Intelligenz – Fragen und Antworten*. April 2021. – URL https://ec.europa.eu/commission/presscorner/detail/de/QANDA_21_1683

[43] Kong, Zhifeng ; Chaudhuri, Kamalika: Understanding Instance-based Interpretability of Variational Auto-Encoders. (2021), Mai

[44] Kumar, Abhishek ; Sattigeri, Prasanna ; Balakrishnan, Avinash: Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. (2017), November

[45] LeCun, Yann ; Cortes, Corinna ; Burges, CJ: MNIST handwritten digit database. In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010)

[46] Linardatos, Pantelis ; Papastefanopoulos, Vasilis ; Kotsiantis, Sotiris: Explainable AI: A Review of Machine Learning Interpretability Methods. In: *Entropy* 23 (2020), dec, Nr. 1, S. 18

[47] Lipton, Zachary C.: The Mythos of Model Interpretability. (2016), Juni

[48] Lundberg, Scott ; Lee, Su-In: A Unified Approach to Interpreting Model Predictions. (2017), Mai

[49] Mathieu, Emile ; Rainforth, Tom ; Siddharth, N. ; Teh, Yee W.: Disentangling Disentanglement in Variational Autoencoders. (2018), Dezember

[50] Maximilian Kohlbrenner, Sabbir Ahmmed Youssef K.: Pre-Training CNNs Using Convolutional Autoencoders. (2017)

[51] Maćkiewicz, Andrzej ; Ratajczak, Waldemar: Principal components analysis (PCA). In: *Computers and Geosciences* 19 (1993), Nr. 3, S. 303–342. – URL https://www.sciencedirect.com/science/article/pii/009830049390090R. – ISSN 0098-3004

[52] Meisel, Andreas: *Mustererkennung und Machine Learning*. Vorlesungsfolien. Oktober 2021

[53] Nguyen, An phi ; Martínez, María R.: Learning Invariances for Interpretability using Supervised VAE. (2020), Juli

[54] Oord, Aaron van den ; Vinyals, Oriol ; Kavukcuoglu, Koray: Neural Discrete Representation Learning. (2017), November

[55] Oza, Poojan ; Patel, Vishal M.: Active Authentication using an Autoencoder regularized CNN-based One-Class Classifier. (2019), März

[56] Peltola, Tomi: Local Interpretable Model-agnostic Explanations of Bayesian Predictive Models via Kullback-Leibler Projections. (2018), Oktober

[57] Philipsen, Mark P. ; Moeslund, Thomas B.: Distance in Latent Space as Novelty Measure. (2020), März

[58] Phillips, P. J. ; Hahn, Carina A. ; Fontana, Peter C. ; Yates, Amy N. ; Greene, Kristen ; Broniatowski, David A. ; Przybocki, Mark A.: Four Principles of Explainable Artificial Intelligence. National Institute of Standards and Technology, sep 2021. – Forschungsbericht

[59] Quinn, Thomas P. ; Gupta, Sunil ; Venkatesh, Svetha ; Le, Vuong: A Field Guide to Scientific XAI: Transparent and Interpretable Deep Learning for Bioinformatics Research. (2021), Oktober

[60] Quinn, Thomas P. ; Senadeera, Manisha ; Jacobs, Stephan ; Coghlan, Simon ; Le, Vuong: Trust and Medical AI: The challenges we face and the expertise needed to overcome them. (2020), August

[61] Rai, Arun: Explainable AI: from black box to glass box. In: *Journal of the Academy of Marketing Science* 48 (2019), dec, Nr. 1, S. 137–141

[62] Razzaki, Salman ; Baker, Adam ; Perov, Yura ; Middleton, Katherine ; Baxter, Janie ; Mullarkey, Daniel ; Sangar, Davinder ; Taliercio, Michael ; Butt, Mobasher ; Majeed, Azeem ; DoRosario, Arnold ; Mahoney, Megan ; Johri, Saurabh: A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis. (2018), Juni

[63] Ribeiro, Marco T. ; Singh, Sameer ; Guestrin, Carlos: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. (2016), Februar

[64] Schockaert, Cedric ; Macher, Vadim ; Schmitz, Alexander: VAE-LIME: Deep Generative Model Based Approach for Local Data-Driven Model Interpretability Applied to the Ironmaking Industry. (2020), Juli

[65] Schwalbe, Gesina ; Finzel, Bettina: A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts. (2021), Mai

[66] Selvaraju, Ramprasaath R. ; Cogswell, Michael ; Das, Abhishek ; Vedantam, Ramakrishna ; Parikh, Devi ; Batra, Dhruv: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. (2016), Oktober

[67] Seninge, Lucas ; Anastopoulos, Ioannis ; Ding, Hongxu ; Stuart, Joshua: VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. In: *Nature Communications* 12 (2021), sep, Nr. 1

[68] Shlens, Jonathon: A Tutorial on Principal Component Analysis. (2014), April

[69] Shrikumar, Avanti ; Greenside, Peyton ; Kundaje, Anshul: Learning Important Features Through Propagating Activation Differences. In: *PMLR 70:3145-3153, 2017* (2017), April

[70] Simonyan, Karen ; Vedaldi, Andrea ; Zisserman, Andrew: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. (2013), Dezember

[71] Singh, Amitojdeep ; Sengupta, Sourya ; Lakshminarayanan, Vasudevan: Explainable deep learning models in medical image analysis. (2020), Mai

[72] Tran, Loc ; Dolph, Chester ; Zhao, Derek: Enhancing Neural Network Explainability with Variational Autoencoders, 01 2021

[73] Utkin, Lev ; Drobintsev, Pavel ; Kovalev, Maxim ; Konstantinov, Andrei: Combining an Autoencoder and a Variational Autoencoder for Explaining the Machine Learning Model Predictions. In: *2021 28th Conference of Open Innovations Association (FRUCT)*, 2021, S. 489–494

[74] Utkin, Lev V. ; Kovalev, Maxim S. ; Kasimov, Ernest M.: An explanation method for Siamese neural networks. (2019), November

[75] Vahdat, Arash ; Kreis, Karsten ; Kautz, Jan: Score-based Generative Modeling in Latent Space. (2021), Juni

[76] Vilone, Giulia ; Longo, Luca: Explainable Artificial Intelligence: a Systematic Review. (2020), Mai

[77] Wang, Shuo ; Chen, Tianle ; Chen, Shangyu ; Rudolph, Carsten ; Nepal, Surya ; Grobler, Marthie: OIAD: One-for-all Image Anomaly Detection with Disentanglement Learning. (2020), Januar

[78] Xiao, Han ; Rasul, Kashif ; Vollgraf, Roland: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. In: *CoRR* abs/1708.07747 (2017). – URL http://arxiv.org/abs/1708.07747

[79] Yu, Ronald: A Tutorial on VAEs: From Bayes' Rule to Lossless Compression. (2020), Juni

[80] Zach, Juri: Entwicklung einer Qualitätsmetrik für Interpretationen von neuronalen Netzen. (2021), Mai. – URL https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/zach.pdf

[81] Zhou, Ding ; Wei, Xue-Xin: Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-VAE. In: *NeurIPS 2020* (2020), November

[82] Zhu, Qiuyu ; Zhang, Ruixin: A Classification Supervised Auto-Encoder Based on Predefined Evenly-Distributed Class Centroids. (2019), Februar

A TensorFlow/Keras-Model

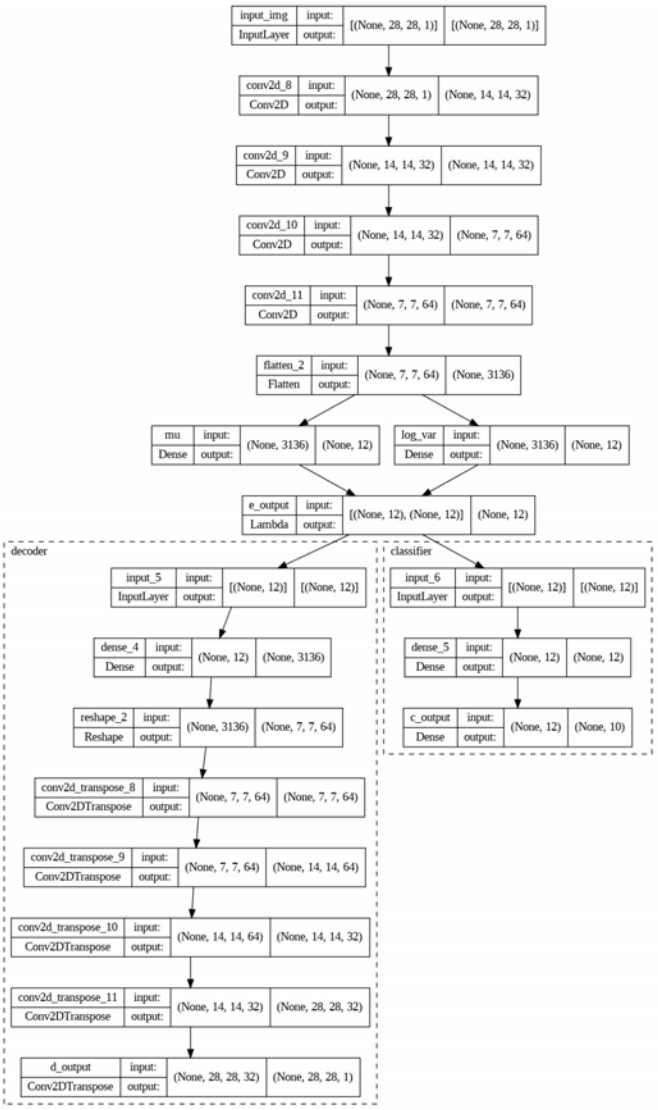


Abbildung A.1: Alle Versuche wurden mit dieser Architektur und variiertem Codelayer durchgeführt.

Glossar

AE Autoencoder.

CNN Convolutional neural network.

KI Künstliche Intelligenz.

KL-Divergenz Kullback-Leibler Divergenz.

KNN Künstliches neuronales Netz.

LIME Local Interpretable Model-agnostic Explanations.

ML Machine Learning.

NCC Normalized Cross Correlation.

PCA Principal Component Analysis.

SNN Siamesiches neuronales Netz.

t-SNE t-Distributed Stochastic Neighbor Embedding.

VAE Variational Autoencoder.

XAI Explainable Artificial Intelligence.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Masterarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Interpretierbarkeit neuronaler Klassifikatoren mit Hilfe von Variational Autoencodern

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original