

Bachelorarbeit

**Entwicklung eines Algorithmus für die
Wiedererkennung von Palettenklötzen
basierend auf maschinellem Lernen**

Britta Grimme
01. 2023

Gutachter:

Prof. Dr. Emmanuel Müller

Simon Klüttermann, MSc.

Technische Universität Dortmund

Fakultät für Informatik

Chair of Data Science and Data Engineering

<https://ls9-www.cs.tu-dortmund.de>

Kurzzusammenfassung

In dieser Arbeit werden maschinelle Lernverfahren eingesetzt, um einen bildgestützten Ansatz zur Verfolgung von Paletten zu entwickeln. Ziel der Arbeit ist es eine sichere Wiedererkennung von unbekanntem, also vom Modell noch nicht zuvor gesehenen Paletten anhand des Pressspannmusters der Palettenklötze zu ermöglichen. Als Benchmark-Verfahren wird der ORB-Algorithmus (Oriented Fast and Rotated BRIEF) zur Merkmalerkennung genutzt, welcher charakteristische Strukturen im Pressspannmuster erkennt und diese in einem zweiten Schritt untereinander abgleicht. Alle anderen in dieser Arbeit verwendeten Verfahren basieren auf neuronalen Netzen, insbesondere auf Siamesischen Neuronalen Netzen (SNN), einer Architektur basierend auf zwei oder mehr parallelen, identischen neuronalen Subnetzen. Solche Netze können aus den Daten automatisch wesentliche Unterscheidungsmerkmale erfassen und liefern durch den Vergleich dieser Merkmale eine Ähnlichkeitsmetrik, welche eine Re-Identifikation von unbekanntem Palettenklötzen ermöglicht. Neben der direkten Anwendung von SNNs auf den Palettenklotzbildern steht insbesondere ein Ansatz im Vordergrund, der kleinere Repräsentationen der Palettenklotzbilder verwendet. Diese enthalten Informationen über Anomalien bzw. Merkmale im Pressspannmuster (Anomaliekarten bzw. Merkmalskarten) oder sind der inneren Kernschicht eines Autoencoders (Latent-Space-Karten) entnommen. Die Ergebnisse zeigen, dass die vorherige Extraktion der wesentlichen Merkmale insbesondere in Form von Merkmalskarten und Latent-Space-Karten gegenüber dem direkten Einsatz eines SNNs auf Palettenklotzbildern einen deutlichen Vorteil bietet, wobei eine Kombination der Ausgaben beider Verfahren (Merkmalskarten und Latent-Space-Karten) nochmals zu einer Zunahme der Erkennungsrate führt.

Dies belegt, dass es grundsätzlich möglich ist, mittels unüberwachter Detektionsverfahren beliebige Palettenklötze zuverlässig wiederzuerkennen. Es ist insbesondere bemerkenswert, dass diese Erkennungsleistung bereits mit relativ einfachen neuronalen Architekturen kompetitiv zu jener von überwachten Verfahren ist.

Inhaltsverzeichnis

1	Einleitung	1
2	Hintergrund	5
2.1	Merkmalserkennung mit ORB (Oriented FAST and Rotated BRIEF)	5
2.2	Künstliche neuronale Netze	7
2.3	Convolutional Neural Networks (CNNs)	9
2.4	Autoencoder	10
2.5	Siamesische Neuronale Netze (SNN)	12
2.6	Datensatz	15
2.7	Gütemaße zur Beurteilung der Verfahren	16
2.8	Eingesetzte Technologien	17
3	Ansatz mit ORB (Oriented FAST and Rotated BRIEF)	19
3.1	Experimentelles Vorgehen	19
3.2	Ergebnisse des ORB-Verfahrens	20
4	Ansatz mit Anomaliekarten	25
4.1	Vorgehen zur Erzeugung der Anomaliekarten	25
4.1.1	Architektur des Autoencoders	26
4.1.2	Training des Autoencoders	27
4.1.3	Berechnung der Anomaliekarten	27
4.2	Iteratives Verfahren zum Vergleich von Anomaliekarten	28
4.2.1	Darstellung des Verfahrens	29
4.2.2	Ergebnisse des iterativen Vergleichs von Anomaliekarten	30
4.3	Anwendung eines SNNs auf Anomaliekarten	33
4.3.1	Architektur des Siamesischen Neuronalen Netzes	33
4.3.2	Triplet Loss als Verlustfunktion	34
4.3.3	Ergebnisse der Anwendung eines SNNs auf Anomaliekarten	35
5	Ansatz mit Merkmalskarten	37
5.1	Vorgehen zur Erzeugung der Merkmalskarten	37

5.2	Architektur des SNNs zur Re-Identifikation von Merkmalskarten	38
5.3	Ergebnisse des Ansatzes mit Merkmalskarten	39
6	Ansatz mit Latent-Space-Karten	43
6.1	Vorgehen zur Erzeugung der Latent-Space-Karten	43
6.2	Architektur des SNNs zur Re-Identifikation von Latent-Space-Karten	44
6.3	Ergebnisse des Ansatzes mit Latent-Space-Karten	44
7	Kombination von Modellen	47
7.1	Vorgehen zur Kombination der Modelle	47
7.2	Ergebnisse der Kombination von Modellen	48
8	Direkter Ansatz: SNN mit Palettenklotzbildern	51
8.1	Vorgehen zur direkten Anwendung eines SNNs auf Palettenklotzbilder	51
8.2	Ergebnisse des direkten Ansatzes	52
9	Zusammenfassung und Diskussion	55
	Literaturverzeichnis	59

Kapitel 1

Einleitung

Im Kontext des Supply Chain Managements [8], also der ganzheitlichen Betrachtung und Optimierung globaler Versorgungsketten, spielt der Vorgang des Transports eine essentielle Rolle. Um etwa Lieferwege von Produkten jederzeit nachvollziehen zu können, ist es wichtig, diese an verschiedenen Zwischenstationen sicher wieder auffinden zu können. Da Produkte etwa verbreitet auf Europaletten transportiert werden, ist es ausreichend die Transporteinheit der Europalette zu identifizieren. Ansätze einer eindeutigen Identifikation könnten dabei technologisch auf RFID-Tags oder Barcodes auf jeder Europalette basieren. Die Ausstattung aller Paletten mit diesen Erkennungsmerkmalen führt jedoch zu einem großen Aufwand, um eine globale Verbreitung sicherzustellen. Weiterhin können Barcodes schnell verschmutzt oder sogar zerstört werden, während RFID-Tags mit sehr hohen Installationskosten nicht nur für die Tags selbst, sondern insbesondere für die Lesegeräte und die Inventar-Software verbunden sind. Zudem sind beide Ansätze nicht abwärtskompatibel, d.h. alte Paletten müssten nachgerüstet werden.

Ein alternativer und universeller Ansatz zur Verfolgung von Paletten kann auf der bildgestützten Wiedererkennung von Paletten mit Hilfe moderner Verfahren der künstlichen Intelligenz basieren (siehe schematische Darstellung in Abb. 1.1). Rutinowski et al. [32] haben einen entsprechenden Ansatz basierend auf überwachten maschinellen Lernverfahren vorgeschlagen, bei denen die Pressspanmuster von Paletten erlernt wurden, um eine spätere Wiedererkennung zu realisieren. Während die Autoren einerseits gezeigt haben, dass damit eine zuverlässige Identifizierung von Paletten möglich ist, benötigt dieses Verfahren jedoch andererseits eine umfangreiche Trainingsphase für unbekannte Paletten.

Darüber hinausgehend sollen in dieser Arbeit Verfahren eingesetzt werden, die eine Wiedererkennung von unbekanntem, also vom Modell zuvor noch nicht „gesehenen“ Palettenklötzen ermöglichen. Ein wichtiges Ziel dabei ist es eine sichere Wiedererkennung zu gewährleisten, ohne dass das Modell fortlaufend aktualisiert werden muss.

Solche Verfahren werden auch „One-Shot-Learning“ genannt [20, 18, 16], da die einmalige Beobachtung eines neuen Objektes zu dessen Wiedererkennung genügt. Konzeption-

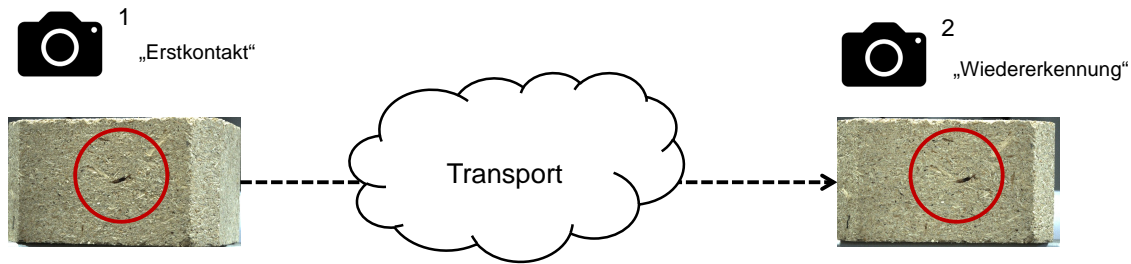


Abbildung 1.1: Problemstellung. Holzklötze von Europaletten sollen nach einmaligem Kontakt und anschließendem Transport wiedererkannt werden. Dazu soll das Pressspanmuster als sog. Fingerabdruck zur Wiedererkennung genutzt werden. Insbesondere Anomalien bzw. auffällige Strukturen im Pressspanmuster, wie im roten Kringel eingekreist, sind vielversprechende Marker, um eine gute Wiedererkennungsrate zu erreichen.

nell versuchen One-Shot-Learning-Verfahren die menschliche Fähigkeit des Erlernens und Erkennens neuer Muster zu reproduzieren. Lake et al. [18] haben einen Ansatz namens Hierarchical Bayesian Program Learning (HBPL) entwickelt, welcher visuelle Konzepte von neuen, d. h. zuvor unbekanntem Schriftzeichen lernen kann. Diese erlernten Konzepte können von nur einem Bild ausgehend generalisiert werden, wodurch Schriftzeichen mit nahezu menschlicher Genauigkeit wiedererkannt werden können. Dieser Ansatz beruht auf der Tatsache, dass Schriftzeichen aus wiederkehrenden elementaren Bausteinen – auch Primitive genannt – zusammengesetzt sind. Durch die Identifikation solcher Bausteine, können Schriftzeichen als eine Menge von Bausteinen und deren räumlichen Beziehungen beschrieben werden.

Auch in komplexeren Situationen sind Menschen in der Lage einmal gesehene Muster wiederzuerkennen. Beispielsweise ist das menschliche Gehirn in der Lage Unterschiede in Gesichtern auf wesentliche Informationen zu reduzieren und sich diese einzuprägen. Diese Fähigkeit haben etwa Schoff et al. [33] in ihrem FaceNet-Modell umgesetzt. Dabei setzen die Autoren einen derzeit sehr beliebten und erfolgreichen Ansatz für One-Shot-Learning ein, sogenannte Siamesische Neuronale Netze (SNNs). Ein SNN ist eine Architektur basierend auf zwei identischen und parallelen neuronalen (Sub-)Netzen, die jeweils eine andere Eingabe erhalten und deren Ausgaben kombiniert werden, um eine Vorhersage über die Ähnlichkeit beider Eingaben zu treffen. Ein solches Netz wurde erstmals 1993 von Bromley et al. [5] für die Verifizierung von Unterschriften vorgeschlagen. Neben Anwendungen in der Gesichtserkennung [33] wurden diese Netze erfolgreich etwa zur Erkennung von Schriftzeichen [16], bei der Objektverfolgung in Videos [27], der Identifizierung von Personen durch Gangerkennung und auch im Bereich der medizinischen Bildgebung [21] eingesetzt.

Als mögliche Alternative zu One-Shot-Learning-Verfahren können Ansätze zur Merkmalsextraktion wie SIFT (Scale-Invariant Feature Transform) [23], SURF (Speeded Up Robust Features) [4] und ORB (Oriented FAST and Rotated BRIEF) [31] genutzt werden. Diese werden traditionell in Kombination mit einem Klassifikationsverfahren zur Objekt-

erkennung eingesetzt (wie z. B. in [11]). Auch hier dient nur ein einziges Referenzbild als Ausgangspunkt für die Erkennung. Alle drei genannten Verfahren basieren auf einem einheitlichen dreistufigen Vorgehen. Als erstes werden die wichtigsten Merkmale eines Bildes, wie z. B. Ecken und Kanten von Objekten, gefunden (feature detector). Danach werden Informationen über diese Merkmale extrahiert (descriptor extractor), um diese anschließend miteinander zu vergleichen und gemeinsame Merkmale zu identifizieren (feature matcher). Nur die Umsetzung der Merkmalsdetektion und- extraktion unterscheidet sich in den drei Methoden [14, 31]. Alle drei Verfahren sind skalierungs-, rotations- und beleuchtungs-invariant. Da der ORB-Algorithmus der schnellste Algorithmus unter den drei genannten Ansätzen ist und im Gegensatz zu SIFT und SURF keine Lizenzbeschränkungen aufweist, wird ausschließlich dieses Verfahren im Kontext dieser Arbeit weiterhin betrachtet (Kapitel 2.1) und verwendet.

Es ist nun naheliegend, die vorgestellten Verfahren des One-Shot-Learnings (hier Siamesische Netzwerke) sowie der Merkmalserkennung (hier ORB) direkt auf Bilder des Pressspanmusters von Paletten anzuwenden. Dabei liegt die Annahme zugrunde, dass die Verfahren diskriminierende Features oder Merkmale selbstständig erlernen bzw. detektieren. Ein alternativer Ansatz besteht darin, Anomalien bzw. Merkmale im Pressspanmuster (siehe z.B. in Abbildung 1.1, innerhalb des roten Kreises) vorab herauszuarbeiten und diese in einem zweiten Schritt wiederzuerkennen. Dieser Ansatz erscheint insbesondere für die Verwendung innerhalb Siamesischer Netze vielversprechend, da die spezielle Fähigkeit dieser Netzwerke dabei darauf fokussiert wird, Ähnlichkeiten bezüglich Anomalien/Merkmalen oder sogar deren Anordnung innerhalb des Pressspanmusters zu bestimmen.

In der Arbeit soll ein besonderer Fokus auf der zweiten Betrachtungsart liegen. Dazu werden zuerst Anomalien bzw. Merkmale im Pressspanmuster herausgearbeitet und anschließend Ähnlichkeiten auf diesen im Vergleich zum Originalbild viel kleineren Repräsentationen bestimmt.

Aufbau der Arbeit

Die Arbeit gliedert sich in die folgenden Kapitel:

- Das Kapitel 2 befasst sich mit allen Grundlagen, die für das Verständnis der Arbeit von Bedeutung sind. Es stellt existierende Verfahren vor, die in dieser Arbeit zur Re-Identifikation von Pressspanmustern von Europaletten verwendet bzw. adaptiert werden. Des Weiteren werden einige Grundlagen künstlicher neuronaler Netze, insbesondere Autoencoder und Siamesische Neuronale Netze, behandelt, deren Anwendung einen großen Teil der Arbeit ausmacht. Zusätzlich wird der verwendete Datensatz vorgestellt und auf Maße zur Ermittlung der Klassifikationsgüte von Modellen eingegangen.

- Das Kapitel 3 behandelt die Anwendung von ORB (Oriented FAST and Rotated BRIEF), ein Algorithmus zur Erkennung von Bildmerkmalen, auf die Palettenklotzbilder. Auf den detektierten Merkmalen im Pressspanmuster wird ein direkter Ähnlichkeitsvergleich durchgeführt.
- In den Kapiteln 4, 5 und 6 werden auf verschiedene Weisen kleine Repräsentationen der Palettenklotzbilder erstellt, sogenannte Anomaliekarten (enthalten Informationen über die räumliche Anordnung, sowie über die Ausprägung von Anomalien im Pressspanmuster), Merkmalskarten (enthalten Informationen über Strukturen im Pressspanmuster) oder Latent-Space-Karten (enthalten die wesentlichen Informationen der Palettenklotzbilder, um deren Rekonstruktion zu ermöglichen). Allen Ansätzen ist gemeinsam, dass die erstellten Karten jeweils in ein Siamesisches Neuronales Netz gespeist werden, welches automatisch wichtige Unterscheidungsmerkmale erfassen kann und durch den Vergleich dieser Merkmale eine Ähnlichkeitsmetrik liefert.
- Im Kapitel 7 werden die Ergebnisse der Modelle mit Merkmalskarten und Latent-Space-Karten kombiniert, um zu überprüfen, ob eine gemeinsame Ausgabe zu besseren Ergebnissen führen kann.
- Im Kapitel 8 wird validiert, ob eine direkte Anwendung eines Siamesisches Neuronales Netzes auf Palettenklotzbilder kompetitiv zu den Ansätzen mit vorheriger Komprimierung ist.
- Das Kapitel 9 beinhaltet eine kurze Zusammenfassung und Diskussion aller Ergebnisse und schließt mit einem kurzen Ausblick für mögliche weiterführende Forschung.

Kapitel 2

Hintergrund

Das einleitend beschriebene Vorgehen zur Wiedererkennung von Pressspanmustern nur einmal gesehener Europaletten setzt die Verwendung und Adaption existierender Verfahren voraus. Auf diese und einige Grundlagen neuronaler Netze soll im Folgenden eingegangen werden. Das ORB-Verfahren wird in Abschnitt 2.1 als Repräsentant einer Verfahrensklasse beschrieben, welche es erlaubt charakteristische Merkmale in Bildern zu detektieren. Durch den Abgleich dieser Merkmale wird ein direkter Ähnlichkeitsvergleich ermöglicht.

Die weiteren Ansätze zur Wiedererkennung von Palettenklötzen basieren auf neuronalen Netzen, deren Grundlagen kurz im Abschnitt 2.2 vorgestellt werden. Im Abschnitt 2.3 wird auf Convolutional Neural Nets eingegangen, welche besonders für die Verarbeitung zweidimensionaler Daten (z. B. Bilder) geeignet sind und daher in dieser Arbeit zur Re-Identifikation von Palettenklotzbildern verwendet werden. Anschließend werden zwei spezielle Architekturen neuronaler Netze beschrieben, der Autoencoder (siehe Abschnitt 2.3) und Siamesische Neuronale Netze (siehe Abschnitt 2.5). Ersterer wird einerseits genutzt, um Anomalien im Pressspanmuster zu detektieren. Andererseits kann der Autoencoder mit seiner inneren Kernschicht eine Repräsentation wichtiger Merkmale von Pressspanmustern (Fingerprint) liefern. Siamesische neuronale Netze hingegen sind darauf spezialisiert, Ähnlichkeiten zwischen den Eingangsdaten zu erkennen. Sie sind somit hervorragend dafür geeignet, um Ähnlichkeiten bezüglich zuvor ermittelter Fingerprints oder bezüglich der räumlichen Anordnungen von Anomalien zu erkennen. Im letzten Abschnitt dieses Kapitels wird noch kurz auf den zugrundeliegenden Datensatz aus Palettenklötzen eingegangen.

2.1 Merkmalerkennung mit ORB (Oriented FAST and Rotated BRIEF)

ORB [31] ist ein Algorithmus zur Erkennung von Bildmerkmalen, der 2011 als effiziente Alternative zu SIFT (Scale-Invariant Feature Transform) [23] und SURF (Speeded Up

Robust Features) [4] entwickelt wurde und in Open Source Bibliotheken wie OpenCV¹ enthalten ist. Die Hauptvorteile von ORB gegenüber SIFT und SURF sind die schnellere Berechnung bei annähernd gleichbleibender Leistung bzgl. der wichtigsten Metriken, wie z. B. die Robustheit gegenüber Beleuchtungsschwankungen, sowie gegenüber Unschärfe [31]. Traditionell werden diese Algorithmen in Kombination mit Klassifikationsverfahren zur Objekterkennung eingesetzt [11, 3].

Alle drei genannten Verfahren basieren auf einem einheitlichen dreistufigen Vorgehen. Im ersten Schritt (feature detector) werden interessante Bildausschnitte (Merkmalspunkte oder „keypoints“), insbesondere Ecken und Kanten, detektiert, da diese das größte Wiedererkennungspotential haben. Nachdem die Erkennung abgeschlossen ist, folgt im zweiten Schritt die Merkmalsbeschreibung (descriptor extractor). Dabei wird insbesondere die nahe Umgebung um die zuvor identifizierten Merkmalspunkte betrachtet. Im dritten Schritt, dem sogenannten „Matching“, werden die Deskriptoren zweier Bilder verglichen, um gemeinsame Merkmalspunkte zu identifizieren.

ORB kombiniert die Stärken des FAST (Features from Accelerated Segment Test)-Detektors [30] und des BRIEF (Binary Robust Independent Elementary Features)-Deskriptors [6]. Merkmale werden mit dem FAST-Detektor folgendermaßen detektiert: Die Helligkeit eines Kandidatenpixels p wird mit den umliegenden Pixeln, die sich in einem kleinen Kreis um p befinden, verglichen. Wenn eine Gruppe von zusammenhängenden Pixeln im Kreis alle heller bzw. dunkler als die Helligkeit des Kandidatenpixels p sind, dann wird p als Merkmalspunkt klassifiziert. Merkmalspunkte beschreiben also Ecken und Kanten im Bild. Um skaleninvariante Merkmale zu identifizieren, verwendet der ORB-Algorithmus eine multiskalare Bildpyramide. Eine solche Bildpyramide eines einzelnen Bildes ist aus einer Reihe von Bildern aufgebaut, die alle Varianten desselben Bildes mit unterschiedlichen Auflösungen sind. Jede Ebene der Pyramide enthält die im Vergleich zur vorherigen Ebene verkleinerte Variante des Bildes. Mit dem FAST-Algorithmus können nun Merkmalspunkte auf jeder Ebene, also in unterschiedlichen Maßstäben erkannt werden. Des Weiteren haben FAST-Merkmale keine Orientierungskomponente. Um Rotationsinvarianz zu gewährleisten, wird das sogenannte Intensity Centroid-Verfahren [29] angewandt, wodurch jedem erfassten Merkmalspunkt eine Orientierung zugeordnet werden kann, um die mit dem BRIEF-Algorithmus berechneten Deskriptoren entsprechend auszurichten.

Nach der Bestimmung aussagekräftiger Merkmalspunkte, werden diese mittels BRIEF in binäre Merkmalsvektoren (sog. Deskriptoren) konvertiert. Dazu wird zunächst ein kleiner quadratischer Bildausschnitt q um den zu betrachtenden Merkmalspunkt herum aufgespannt. Innerhalb dieses geglätteten Bildausschnitts werden zufällig 256 Paare von Koordinatenpunkten (x,y) bestimmt mit denen binäre Intensitätsvergleiche gemäß Gleichung 2.1 durchgeführt werden:

¹https://docs.opencv.org/3.4/db/d95/classcv_1_1ORB.html

$$\tau(q, x, y) := \begin{cases} 1, & \text{wenn } q(x) < q(y) \\ 0, & \text{sonst} \end{cases} \quad (2.1)$$

wobei $q(x)$ die Intensität/Helligkeit von q an einem Punkt x ist. Somit entstehen binäre Merkmalsvektoren, auch Deskriptoren genannt, die als Fingerprint der Nachbarschaftsumgebung interpretiert werden können. Um die Vergleichbarkeit dieser Vektoren zu gewährleisten, müssen die gewählten Koordinatenpaare in jedem Bildausschnitt und insbesondere von Bild zu Bild identisch sein, d. h. dem Zufallszahlengenerator muss ein Startwert (seed) übergeben werden, so dass in jedem Bildausschnitt dieselben paarweisen Vergleiche gemacht werden.

Im letzten Schritt werden die Deskriptoren dazu verwendet, die Merkmalspunkte zweier Bilder zu vergleichen (matching). Dazu wird jeder Merkmalsdeskriptorvektor im ersten Bild mit allen Vektoren des zweiten Bildes verglichen. Das Paar mit dem geringsten Abstand zwischen den Deskriptorvektoren wird als beste Übereinstimmung für dieses Merkmal beibehalten. Als Abstandsmaß bietet sich hier die Hamming-Distanz an. Genauso wird mit allen weiteren Merkmalspunkten des Vergleichsbildes vorgegangen.

2.2 Künstliche neuronale Netze

In diesem Abschnitt wird ein kurzer Überblick über neuronale Netze als Grundlage für die folgenden Abschnitte über die speziellen neuronalen Netze, Convolutional Neural Networks, Autoencoder und Siamesische Neuronale Netze, gegeben.

Beim überwachten Lernen werden Feedforward-Netze häufig zur Klassifikation von Daten eingesetzt. Hier besteht das Ziel darin, eine Funktion $f^*(x) = y$ zu approximieren, welche eine Eingabe x einer Kategorie y zuordnet. Ein Feed-Forward-Netz definiert eine Abbildung $y = f(x; \theta)$ und lernt den Wert der Parameter θ die zur besten Funktionsannäherung führen [10].

Als Beispiel ist in Abbildung 2.1 ein einfaches Feed-Forward-Netz dargestellt. Solche Netze zeichnen sich dadurch aus, dass die Schichten lediglich mit der nächsthöheren Schicht verbunden sind und es insbesondere keine rückwärtsgerichteten Kanten gibt. Falls jedes Neuron mit allen Neuronen der folgenden Schicht verbunden ist, spricht man von vollständig verbundenen Netzen.

Ein neuronales Netz besteht aus 3 grundlegenden Schichten von künstlichen Neuronen. Die erste Schicht, welche für die Eingabe verantwortlich ist, wird als Input Layer bezeichnet. Darauf folgen eine oder mehrere verborgene Schichten (Hidden Layer) und eine Ausgabeschicht (Output Layer). Jedes Hidden Layer enthält eine Reihe von nicht verbundenen einzelnen Neuronen die vollständig mit allen Neuronen der vorhergehenden Schicht verbunden sind. Die Ausgabe des Output Layers, ist die eigentliche Vorhersage des Netzes für die gegebenen Eingabedaten.

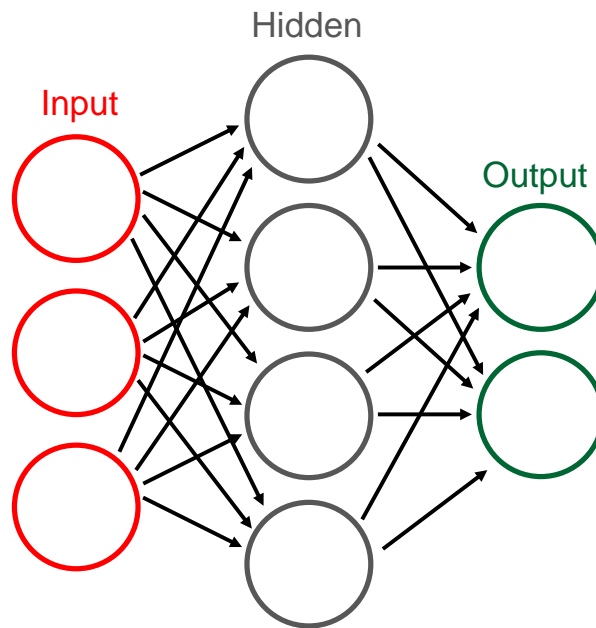


Abbildung 2.1: Architektur eines künstlichen neuronalen Netzes: Die kreisförmigen Knoten stellen künstliche Neurone dar. Die Pfeile beschreiben jeweils Verbindungen zwischen der Ausgabe eines Neurons und dem Eingang eines Neurons in der folgenden Schicht.

Die Funktionsweise des neuronalen Netzes basiert darauf, dass jedes Neuron von der Vorgängerschicht mehrere Inputwerte erhält. Diese Input-Werte werden mit Gewichten multipliziert und zusammen mit einem Bias-Wert aufsummiert. Die Gewichte sind ein Maß für die Intensität des Informationsflusses entlang der Verbindungen im neuronalen Netzwerk und führen dazu, dass die Neuronen unterschiedlich stark gewichtet werden. Der Bias-Wert verschiebt das Grundniveau der Aktivierung und bestimmt dadurch den Schwellenwert, wie stark ein Reiz sein muss, um das Neuron überhaupt anzuregen. In Abhängigkeit von der gewichteten Summe der Eingangssignale wird die Ausgabe des Neurons mit Hilfe einer Aktivierungsfunktion berechnet. Nichtlineare Aktivierungsfunktionen, wie z. B. die sigmoide Aktivierungsfunktion oder die ReLU (rectified linear unit) sorgen dafür, dass das neuronale Netz nichtlineare Funktion darstellen kann [26]. Die Aktivierungsfunktion ReLU (rectified linear unit) wurde von Nair und Hinton 2010 vorgeschlagen [25] und ist seither die am häufigsten verwendete Aktivierungsfunktion für Deep-Learning-Anwendungen, da sie im Vergleich zu anderen Aktivierungsfunktionen die bessere Leistung und Generalisierung bietet. Sie gibt, wie in Gleichung 2.2 dargestellt, für negative Eingabewerte eine Null zurück und für positive Eingabewerte die Identität.

$$f(x) = \max(0, x) \quad (2.2)$$

Im Falle einer kategorialen Zuordnung (“Klassifikation”) wird das neuronale Netz mit Eingaben x trainiert, die von einem Label $Y \approx f^*(x)$ begleitet sind. Die Trainingsbei-

spiele geben der Ausgabeschicht also direkt vor, in welche Kategorie die Eingabedaten einzuordnen sind.

Das Ziel besteht nun darin, eine Kombination aus Gewichten und Bias-Werten zu finden, die den geringsten Fehler erzeugt. Um diesen Fehler zu bewerten, wurden Verlustfunktionen entwickelt, welche die aktuelle Ausgabe des Netzes \hat{Y} mit der gewünschten oder richtigen Ausgabe Y vergleichen. Eine häufig genutzte Verlustfunktion ist die mittlere quadratische Abweichung (MSE), die wie folgt berechnet wird:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (2.3)$$

wobei n die Anzahl an Neuronen in der Output-Schicht, also die Anzahl an Vorhersagen ist.

Mittels Backpropagation und dem Gradientenabstiegsverfahren werden die Gewichte des Netzes nun nach und nach angepasst, indem der Fehler wieder über die Ausgabe zur Eingabeschicht zurück „propagiert“ wird. Dabei werden die Gewichtungen der Neuronenverbindungen abhängig von ihrem Einfluss auf den Fehler geändert. Dies garantiert bei einem erneuten Anlegen der Eingabe eine Annäherung an die gewünschte Ausgabe. Ausführliche Erklärungen des Backpropagation-Algorithmus, des Gradientenabstiegsverfahrens, sowie alternativer Verfahren zur Optimierung finden sich beispielsweise in dem Buch von Goodfellow et al. [10].

2.3 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks [19] sind von biologischen Prozessen inspiriert und so konzipiert, dass sie die neuronale Konnektivität im visuellen Kortex des Gehirns nachbilden. CNNs haben ein breites Anwendungsspektrum im Bereich der Computer Vision, wie z. B. die medizinische Bildanalyse, die Erkennung von Schriftzeichen und die Identifizierung von Objekten und Personen [28].

In diesem Abschnitt wird ein kurzer Überblick über CNNs gegeben. Sie sind ideal geeignet, um Daten mit einer gitterartigen Topologie (wie z. B. Bilder) zu verarbeiten, da die räumlichen Beziehungen zwischen den einzelnen Merkmalen bei den sogenannten Faltungsvorgängen (Convolution) in den folgenden Schichten des Netzes beibehalten werden. In dieser Arbeit werden CNNs daher auf den Palettenklotzbildern angewandt.

Grundsätzlich besteht die Struktur eines klassischen CNNs aus einer Reihung von einer oder mehreren Convolutional-Schichten, gefolgt von einer Pooling-Schicht.

Die Hauptaufgabe der Convolutional Layer ist es, Merkmale des Bildes zu extrahieren. Dies geschieht, indem Filterfenster (Kernel) schrittweise über jedes Pixel im Bild gezogen werden, um den Wert für das Pixel des neuen Bildes zu berechnen. Die Anwendung desselben Filters auf verschiedene Bereiche des Inputs führt zu einer Karte von Aktivierungen,

welche die Position und Ausprägung des erkannten Merkmals angibt. Darauf aufbauend führt die parallele Anwendung unterschiedlicher, erlernter und somit auf die Eingabe angepasster Filter, zu einer Vielzahl von Merkmalskarten. Das Zusammenspiel dieser Merkmalskarten, welche auf die Erkennung verschiedener Merkmale spezialisiert sind, bildet schließlich eine sehr aussagekräftige Repräsentation der Eingabe.

Die Pooling-Schicht dient zur Zusammenfassung der erkannten Merkmale. Im Falle des Max-Pooling wird beispielsweise aus jedem 2×2 Quadrat aus Neuronen des Convolutional Layers nur die Aktivität des aktivsten Neurons für die weiteren Berechnungsschritte beibehalten. Trotz der Datenreduktion (im Beispiel 75 %) verringert sich in der Regel die Performance des Netzwerks nicht durch das Pooling. Neben dem Vorteil der erhöhten Rechengeschwindigkeit, führt das Pooling implizit zu einem Wachstum der Größe der rezeptiven Felder in tieferen Convolutional Layers (ohne dass dafür explizit die Größe der Faltungsmatrizen erhöht werden müsste) [2].

Nachdem einige Convolutional- und Pooling-Schichten durchlaufen sind, folgt meist ein (oder mehrere) sogenanntes „Fully-connected-Layer“, welches die extrahierten Merkmale aus den Convolutional- und den Pooling Layern nutzt, um das Bild zu klassifizieren. Die Anzahl der Knoten der letzten Schicht stimmt mit der Anzahl der Zielklassen überein, so dass die „Softmax“-Aktivierungsfunktion [10] für jede Klasse eine Wahrscheinlichkeit ausgibt.

Um ein aussagekräftiges Modell mit sinnvollen Filtern zu erhalten, muss dieses vorher trainiert werden. Genauso wie bei herkömmlichen Neuronalen Netzen, wird dazu Backpropagation genutzt.

2.4 Autoencoder

Ein Autoencoder ist ein künstliches neuronales Netz, das darauf trainiert ist, seine Eingabedaten möglichst gut zu reproduzieren und auf die Ausgabedaten zu übertragen [10]. Da die Eingabedaten nicht klassifiziert vorliegen müssen, sondern ohne zusätzliches Wissen verarbeitet werden, sind Autoencoder bzw. das Training und Lernen der Parameter in den Bereich des unüberwachten Lernens einzuordnen. Mittels Backpropagation wird der Autoencoder so trainiert, dass das Ausgangssignal sich möglichst wenig vom Eingangssignal unterscheidet. Das wesentliche Element des Autoencoders stellt die versteckte Kernschicht (im englischen latent space) dar. In einem Autoencoder-Modell müssen die versteckten Schichten weniger Dimensionen haben als die der Eingabe- oder Ausgabeschichten. Andernfalls erhält das neuronale Netz zu viel Kapazität, um die Daten zu lernen. Im Extremfall könnte es einfach nur die Eingabewerte auf die Ausgabewerte übertragen, einschließlich des Rauschens, ohne wesentliche Informationen zu extrahieren. Die Abnahme der Dimension bis hin zur Kernschicht ermöglicht also die Extraktion der wesentlichen Informationen und Muster der Eingabewerte in diese Schicht. Konzeptionell lässt sich ein Autoencoder durch zwei

aneinandergereihte, gegensätzliche Trichter darstellen, siehe Abbildung 2.2. Beide Trichter sind als mehrschichtige neuronale Netze implementiert und bilden ein Eingangsbild x (Input) auf ein Ergebnisbild x' (Output) ab. Der erste Trichter stellt die Kodierfunktion, den sog. Encoder $h = f(x)$ dar, welcher das ursprüngliche Bild in eine niederdimensionale latente Repräsentation, den „Latent Space“ h abbildet. Der zweite Trichter entspricht der sogenannten Decoderfunktion $x' = g(h)$, welche meist symmetrisch zum Encoder aufgebaut ist und für die Zunahme der Dimension der Daten von der Kernschicht bis hin zur Ausgabeschicht sorgt, um das Originalbild zu rekonstruieren. Um dem Modell zu ermöglichen, korrekte Parameter für die Dekodierung der latenten Repräsentation zu erlernen, muss eine Verlustfunktion minimiert werden. Ein gängiger Ansatz zur Berechnung des Rekonstruktionsfehlers ist die Verwendung des mittleren quadratischen Fehlers (MSE) als Verlustfunktion:

$$\mathcal{L}(x, x') = \frac{1}{B} \sum_{j=1}^B \frac{1}{WHI} \sum_{w=1}^W \sum_{h=1}^H \sum_{i=1}^I (x_{whi}^j - x'_{whi}^j)^2 \quad (2.4)$$

wobei B , W , H und I die Batchgröße, die Bildbreite, -höhe und die Anzahl der Farbkanaäle bezeichnen.

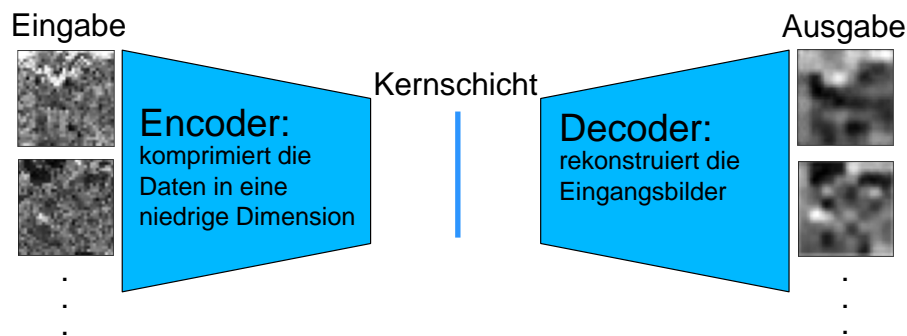


Abbildung 2.2: Autoencoder. Dargestellt sind Teilbilder eines Palettenklotzes als Eingabe und deren Rekonstruktion (Ausgabe).

Sobald das Modell trainiert ist, ist es in der Lage, Bilder x zu kodieren und zu rekonstruieren, die den Trainingsdaten ähneln. Im Falle eines Eingangsbildes, das den Trainingsdaten nicht ähnlich ist, schlägt die Rekonstruktion fehl und führt zu einem hohen Rekonstruktionsfehler, siehe Gleichung 2.4. Die starke Spezialisierung eines Autoencoders auf eine Klasse von Eingangsdaten kann ausgenutzt werden, um Bilder zu erkennen, die sich von denen unterscheiden, die beim Training verwendet wurden. Aus diesem Grund sind Autoencoder im Bereich der Erkennung von Anomalien sehr erfolgreich [17]. In dieser Arbeit wird der Autoencoder unter anderem auf vielen Teilbildern eines Palettenklotzes angewandt, so dass die Gleichung 2.4 für jedes Teilbild einen Anomaliewert liefert. Anschließend können die Anomaliewerte zu einer sogenannten Anomaliekarte zusammengesetzt werden, die darstellt in welchen Bereichen eines Palettenklotzes sich abnormale Strukturen befinden.

2.5 Siamesische Neuronale Netze (SNN)

Ein Siamesisches Netz ist eine Architektur basierend auf zwei oder mehr parallelen, identischen neuronalen (Sub-)Netzen. Die Netze² erhalten jeweils eine der zu vergleichenden Eingaben und wandeln diese in Merkmalsvektoren um. Im nächsten Schritt wird ein Abstandsmaß auf die Merkmalsvektoren angewandt, um Ähnlichkeiten zwischen den Eingaben zu erlernen. Solch ein Netz mit drei Eingängen ist in Abbildung 2.3 am Beispiel von Palettenklötzen dargestellt.

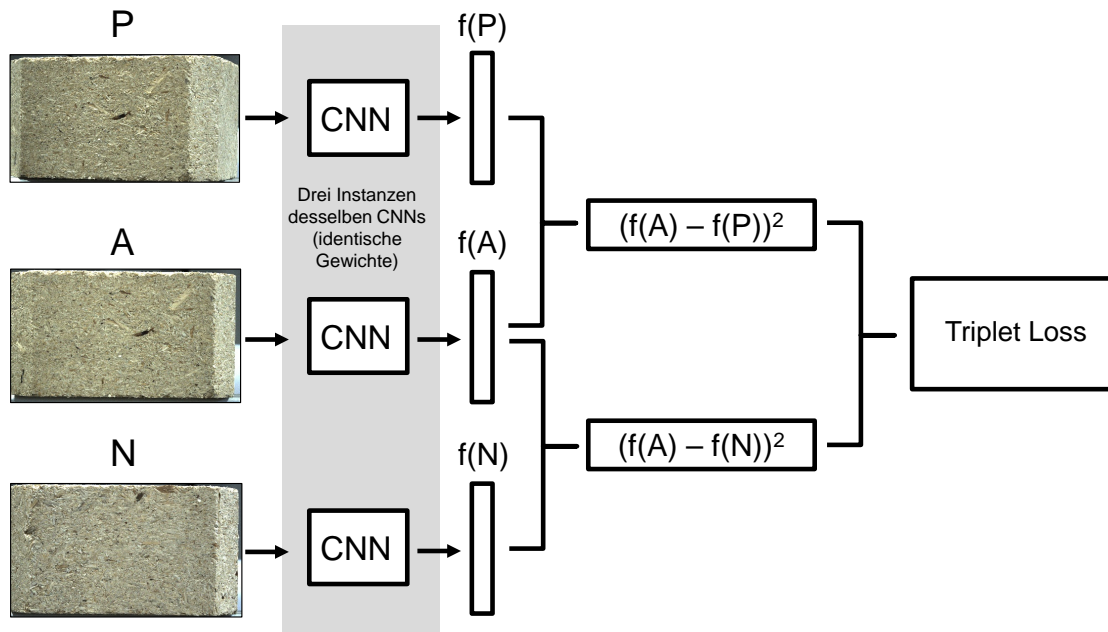


Abbildung 2.3: Siamesisches Neuronales Netz mit der Verlustfunktion „Triplet Loss“ zur Identifizierung von Pressspanmustern. Als Eingabe erhält das Netz drei Bilder von Palettenklötzen. Das Anker-Bild (A) dient als Referenz. Das „positive“ Bild (P) ist ein Bild desselben Palettenklotzes und das „negative“ Bild (N) ist entsprechend ein Bild eines anderen Palettenklotzes. Alle drei Bilder werden durch ein Convolutional Neural Net (CNN) geleitet, wodurch die Merkmalsvektoren $f(A)$, $f(P)$ und $f(N)$ erzeugt werden. Im nächsten Schritt wird die quadrierte euklidische Distanz der Merkmalsvektoren mit dem Merkmalsvektor des Referenzbildes gebildet, um den „Triplet Loss“ berechnen zu können.

Ein Siamesisches Netz kann, sobald es trainiert ist, aus den Daten automatisch wesentliche Unterscheidungsmerkmale erfassen. Durch den Vergleich dieser Merkmale liefert es eine Ähnlichkeitsmetrik anstelle der Vorhersage einer Klasse, wodurch die Vorhersagekraft des Netzwerks nicht nur auf neue Daten, sondern ebenso auf neue Klassen verallgemeinert werden kann.

²Da nicht nur die Architektur der Netze, sondern auch die Gewichtung übereinstimmt, genügt es in der Implementierung nur ein Netz zu generieren, welches nacheinander von den zu vergleichenden Eingaben durchlaufen wird.

Ein Siamesisches Netz wurde erstmals von Bromley et al. [5] für die Verifizierung von Unterschriften vorgeschlagen. Seitdem wurden Siamesische Netzwerke in einer Vielzahl von Anwendungen eingesetzt, die auf der Erkennung von Ähnlichkeiten beruhen. Im Jahre 2015 wurde ein SNN, welches tiefe Convolutional Neural Networks als Grundgerüst besaß, erfolgreich auf eine Schriftzeichenerkennungsaufgabe angewendet [16]. Die Autoren trainierten das Netzwerkmodell auf dem Omniglot-Datensatz für handgeschriebene Zeichen, um Ähnlichkeiten zwischen verschiedenen Eingabepaaren zu bewerten, die dann zur Generalisierung auf völlig unbekannte Klassen verwendet wurden. Im selben Jahr stellte Google ein Siamesisches Netz namens FaceNet vor, welches auf Gesichtsbildererkennung spezialisiert ist und zu einem State-of-the-Art-Modell der Gesichtserkennung wurde [33]. Das Netz erlernt eine Abbildung von Gesichtsbildern auf einen kompakten euklidischen Raum, in dem Abstände direkt mit einem Maß für die Ähnlichkeit von Gesichtern übereinstimmen. Neuere Studien setzen SNNs zum Beispiel bei der Objektverfolgung in Videos [27], der Identifizierung von Personen durch Gangerkennung und auch im Bereich der medizinischen Bildgebung ein. Li et al. [21] verwendeten ein Siamesisches Netzwerk, um den Schweregrad von Osteoarthritis anhand von Röntgenbildern des Knies zu bewerten. Die Ausgabe des Netzes liefert ein kontinuierliches Spektrum des Schweregrads der Erkrankung und kann gepaarte Bilder desselben Patienten zu zwei Zeitpunkten verwenden, um Veränderungen im Laufe der Zeit aufzuzeigen. Anwendungen von SNNs gibt es jedoch nicht nur im Bereich des Computer-Visions, sondern auch im Audibereich, wie z. B. der Suche nach ähnlichen Audioaufnahmen für eine bestimmte Audioclip-Anfrage, z. B. Tiergeräusche [24]. Velez et al. [36] haben ein Siamesisches Netzwerk trainiert, das erkennt, ob zwei Audioclips von demselben Sprecher stammen, auch bei völlig neuen Sprechern.

Die zentrale Idee hinter den SNNs beruht auf zwei wichtigen Eigenschaften:

- Die Konsistenz der Vorhersagen. Die Gewichtskopplung garantiert, dass zwei extrem ähnliche Eingaben (z. B. Bilder) von ihren jeweiligen Netzen an benachbarten Stellen im Merkmalsraum, auch „embedding“ genannt, abgebildet werden, da jedes Netz dieselbe Funktion berechnet. Unterschiedliche Eingaben hingegen werden an verschiedenen Stellen im Merkmalsraum abgebildet, wodurch die Differenz der Merkmalsvektoren groß ist.
- Die Symmetrie des Netzes: Werden dem Netz zwei unterschiedliche Eingaben vorgelegt, berechnet das Netz dieselbe Metrik, als würden die Eingaben in umgekehrter Reihenfolge, also jeweils dem anderen Zwilling vorlegt werden.

Um diese Ähnlichkeitsmetrik zu erreichen, werden beim Training der Siamesischen Netze spezielle Verlustfunktionen verwendet, die auf der Bestimmung des relativen Abstands zwischen den Merkmalsvektoren basieren, was oft als metrisches Lernen bezeichnet wird. Zwei häufig verwendete Verlustfunktionen sind „Contrastive Loss“ [7], welcher einen Ver-

lust für Eingabepaare konstruiert, und „Triplet Loss“ [33], der ein Tripel von Eingaben verlangt.

Zur Veranschaulichung wird die Eingabe im Folgenden als Bild angenommen: Ein Tripel besteht aus einem Vergleichsbild (Anker: A), einem Bild derselben Klasse (positiv: P) und einem Bild einer anderen Klasse (negativ: N). Um ein Tripel als Eingabe verarbeiten zu können, benötigt das Netzwerk, wie in Abbildung 2.3 zu sehen, drei parallele, identische neuronale Netze. Ziel dieser Verlustfunktion ist es, das positive (übereinstimmende) Paar von dem negativen um eine Distanzspanne zu trennen, wodurch kein Label vorhergesagt wird, sondern der relative Abstand zwischen beiden Eingaben bestimmt wird, was oft als metrisches Lernen bezeichnet wird. Genauer gesagt, muss die Distanzmetrik zwischen dem Positiv- und dem Negativpaar mindestens m betragen, um einen Verlustwert von 0 zu erreichen. Die Verlustfunktion kann mit Hilfe der euklidischen Abstandsfunktion beschrieben werden:

$$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + m, 0) \quad (2.5)$$

In der obigen Gleichung ist m ein sog. Margin-Term, der verwendet wird, um die Abstandsunterschiede zwischen ähnlichen und unähnlichen Paaren im Tripel zu „strecken“, $f(A)$, $f(P)$ und $f(N)$ sind die Merkmalsvektoren für das Anker-, Positiv- und Negativbild. Der Verlustwert ist 0, falls die Distanz zwischen A und N um mindestens m größer ist als die Distanz zwischen A und P . Im Gegensatz zum Vergleich von nur zwei Vergleichsbildern erlaubt der Triplet Loss somit, dass $f(A)$ und $f(P)$ auf einer Mannigfaltigkeit liegen und nicht nur auf einem Punkt. Die zu minimierende Verlustfunktion ist dann die Summe der Verlustwerte über alle n Tripel.

$$\mathcal{L} = \sum_{i=1}^n L(A^{(i)}, P^{(i)}, N^{(i)}) \quad (2.6)$$

In dieser Arbeit werden Siamesische Netze genutzt, um Pressspanmuster von Paletten zu identifizieren. Ähnlich wie bei der Gesichtsidentifizierung existieren pro Palettenklotz nur eine geringe Anzahl von Bildern mit unterschiedlichen Ausrichtungen und Helligkeiten. Einerseits kann das Siamesische Netzwerk direkt auf Bildern von Pressspanmustern angewandt werden, wie in Abbildung 2.3 gezeigt. Dabei liegt die Annahme zugrunde, dass diskriminierende Merkmale selbstständig erlernt bzw. detektiert werden. Andererseits sollen Anomalien und charakteristische Merkmale (Fingerprints) im Pressspanmuster vorab identifiziert werden und in einem zweiten Schritt ein Siamesisches Netz genutzt werden, um Ähnlichkeiten bezüglich der Fingerprints oder sogar bezüglich der räumlichen Anordnung von Anomalien innerhalb des Pressspanmusters zu bestimmen. Für alle Herangehensweisen ist das Vorgehen ähnlich, siehe Abbildung 2.3. Als Eingabe erhält das Netz ein Referenz- oder Anker-Bild (A), sowie ein Bild desselben Pressspanmusters (P) und ein Bild eines anderen Pressspanmusters (N). Alle drei Eingaben werden in ein SNN, aufgebaut aus drei

parallelen CNNs, gegeben. Anschließend wird die mittlere quadratische Distanz der resultierenden Merkmalsvektoren mit dem Merkmalsvektor des Referenzbildes gebildet und der „Triplet Loss“ berechnet. Allein der Aufbau des CNNs unterscheidet sich in den gewählten Ansätzen.

2.6 Datensatz

Der in dieser Versuchsreihe verwendete Datensatz besteht aus RGB-Bildern von 502 verschiedenen Klötzen aus Europaletten und wird daher „pallet-block-502“ [32] genannt. Von jedem Palettenklotz wurden Bilder aus fünf vordefinierten Kameraperspektiven aufgenommen: Zentralperspektive, Linksdrehung, Rechtsdrehung, linksseitige Verschiebung und rechtsseitige Verschiebung. Alle Fotos wurden sowohl unter natürlicher Beleuchtung, als auch unter künstlicher Beleuchtung aufgenommen, um die Struktur des Pressspanmusters hervorzuheben. Somit beläuft sich der gesamte Datensatz auf 5020 Bildern von 502 Palettenklötzen. Die Bilder wurden zugeschnitten, um den Hintergrund auszusparen. Die verbleibenden Bildausschnitte haben eine Größe von etwa 2100 x 1200 px für die ersten 102 Palettenklötze, welche im bmp-Format vorliegen, und etwa 1700 x 1000 px für die Palettenklötze 103 bis 502, die im jpg-Format vorliegen. In Abbildung 2.4 sind beispielhaft alle Bilder des Palettenklotzes mit der ID 103 dargestellt.

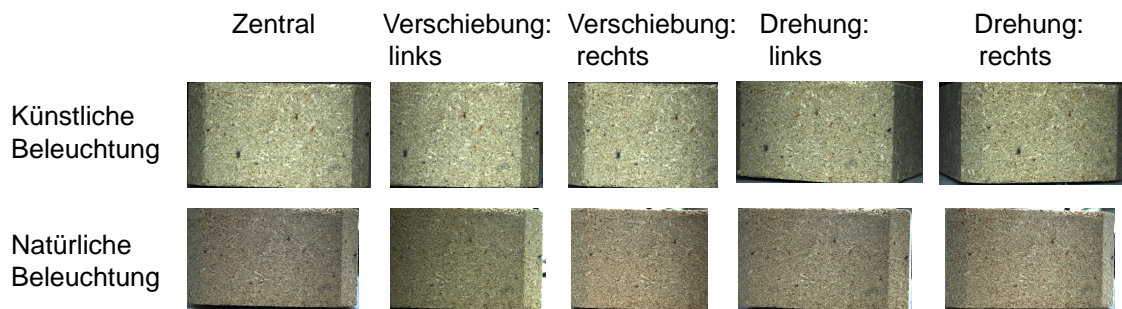


Abbildung 2.4: Palettenklotz ID 103. Dargestellt sind alle 10 Bilder des Palettenklotzes mit der ID 103 unter allen Kameraperspektiven und Beleuchtungsbedingungen.

Der Datensatz wurde für alle Verfahren einheitlich in einen Test- und einen Trainingsdatensatz unterteilt. Der Trainingsdatensatz setzt sich aus 375 pseudorandomisiert ausgewählten Palettenklötzen zusammen. Da der Datensatz wie oben beschrieben aufgrund der variierenden Kameraperspektiven und Beleuchtungsbedingungen aus jeweils 10 Bildern pro Palettenklotz besteht, setzt sich der Trainingsdatensatz somit aus 3750 Bildern zusammen. Der Test-Datensatz enthält entsprechend die restlichen 1270 Bilder.

2.7 Gütemaße zur Beurteilung der Verfahren

Die in dieser Arbeit benutzten Maße zur Ermittlung der Klassifikationsgüte bauen auf der sogenannten Konfusionsmatrix auf. Zur Erstellung einer Konfusionsmatrix wird der Klassifikator auf Objekte eines festgelegten Testdatensatzes angewandt, so dass die Klassifikationsprognose mit den bekannten wahren Klassenzugehörigkeiten verglichen werden kann. Im Falle der binären Klassifikation ergibt sich somit die in Tabelle 2.1 dargestellte 2×2 -Matrix. Die Anzahl der je Klasse korrekt klassifizierten Beobachtungen ist auf der Hauptdiagonalen zu finden (TP und TN). Die „True Positives“ (TP) ergeben sich dabei aus den korrekt als positiv klassifizierten Beobachtungen. Dementsprechend werden die korrekt als negativ klassifizierten Beobachtungen als „True Negatives“ (TN) bezeichnet. Die inkorrekt klassifizierten Objekte setzen sich hingegen aus den „False Positives“ (FP) und den „False Negatives“ (FN) zusammen.

Tabelle 2.1: Konfusionsmatrix zur Gegenüberstellung der tatsächlichen Klasse mit der prognostizierten.

	Klassifikationsergebnis: positiv	Klassifikationsergebnis: negativ
Realität: positiv	# TP	# FN
Realität: negativ	# FP	# TN

Aus der Konfusionsmatrix lassen sich mehrere Gütemaße zur Messung der Leistungsfähigkeit von Klassifikatoren ableiten. Im Folgenden wird nur auf die im Rahmen dieser Arbeit betrachteten Gütemaße AP (average precision) [34] und AUC (Area under the ROC Curve), sowie auf die Grundlagen, aus denen sich diese Maße zusammensetzen, eingegangen:

Die Precision (P) ist ein Maß dafür, wie gut der Klassifikator falsch positive Klassifizierungen (FP) vermeidet (Precision = 1 falls keine FP) und gibt den Anteil der korrekt als positiv klassifizierten Beobachtungen an der Gesamtheit der als positiv klassifizierten Beobachtungen an:

$$P = \frac{\#TP}{\#TP + \#FP}. \quad (2.7)$$

Der Recall (R) wird auch True-Positive-Rate (TPR) genannt und gibt den Anteil der korrekt als positiv klassifizierten Beobachtungen an der Gesamtheit der tatsächlich positiven Beobachtungen an:

$$R = TPR = \frac{\#TP}{\#TP + \#FN}. \quad (2.8)$$

Die False-Positive-Rate (FPR) entspricht dem Anteil der fälschlich als positiv klassifizierten Beobachtungen unter allen tatsächlich negativen Beobachtungen:

$$FPR = 1 - \frac{\#TN}{\#TN + \#FP}. \quad (2.9)$$

Receiver Operating Characteristic (ROC) und Area under the Curve (AUC):

Oft ist das Ergebnis einer Klassifikation nicht dichotom, d. h. es existieren mehr Ausprägungen als die Kategorien „positiv“ und „negativ“. Werden z. B. Resultate zwischen 0 und 1 ausgegeben, können die oben beschriebenen Gütemaße trotzdem berechnet werden, indem ein Schwellenwert $\gamma \in [0, 1]$ festgelegt wird, oberhalb dessen das Klassifikationsergebnis als positiv definiert wird. Entsprechend werden Resultate $< \gamma$ als negativ definiert.

Die ROC-Kurve stellt nun die Verteilung der TPR und der FPR bei Variation des Schwellenwertes dar. Die in Abhängigkeit von γ berechneten (TPR, FPR)-Paare werden in einem Graphen mit der FPR als Abszisse und der TPR als Ordinate aufgetragen. Der AUC (Area under the ROC Curve)-Wert beschreibt die Fläche unterhalb der ROC-Kurve. Er fasst die gesamte Kurve zu einem einzigen, vom Schwellenwert unabhängigen Gütemaß zusammen. Eine Möglichkeit zur Interpretation des AUC-Wertes ist die Wahrscheinlichkeit, dass das Modell ein zufällig positives Beispiel höher bewertet als ein zufälliges negatives Beispiel [12].

Der AP (average precision)-Wert³ approximiert die Fläche unterhalb der Precision-Recall-Kurve, welche den Kompromiss zwischen Precision und Recall für verschiedene Schwellenwerte aufzeigt. Er wird als das gewichtete Mittel der bei jedem Schwellenwert erreichten Precision berechnet, wobei der Anstieg der Recall-Werte gegenüber dem vorherigen Schwellenwert als Gewicht verwendet wird.

$$AP = \sum_n P_n(R_n - R_{n-1}), \quad (2.10)$$

wobei P_n und R_n die Precision und der Recall beim n-ten Schwellenwert sind.

2.8 Eingesetzte Technologien

Die in diesem Kapitel beschriebenen Ansätze zur Identifikation von Pressspanmustern wurden in der Programmiersprache Python (Version 3) in der Entwicklungsumgebung Jupyter Notebook realisiert. Diese Arbeit nutzt für die benötigten Methoden des maschinellen Lernens, wie der Erstellung der Siamesischen Netzwerke, die Frameworks Tensorflow (Version 2.8.0) und Keras (Version 2.8.0). Unterstützend wird unter anderem die Bibliothek OpenCV-python (Version 4.5.5.64) eingesetzt, welche Computer Vision-Verfahren für die Bildverarbeitung zur Verfügung stellt und insbesondere im Kontext von ORB zur Merkmalsextraktion genutzt wird. Zur Bewertung der Qualität von Vorhersagen der Modelle bietet die Bibliothek Scikit-learn (Version 1.0.2) viele nützliche Hilfsmittel. Des Weiteren werden die Bibliotheken matplotlib für die Evaluation und graphische Darstellung der Ergebnisse

³https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html

und numpy für mathematische Funktionen und Listenoperationen in Python eingesetzt. Die genutzte Rechner-Infrastruktur besteht aus einer Intel Workstation mit zwei NVIDIA RTX 3090 Grafikkarten.

Kapitel 3

Ansatz mit ORB (Oriented FAST and Rotated BRIEF)

In diesem Kapitel wird das ORB-Verfahren [31] zur Merkmalerkennung genutzt, um in der Pressspanstruktur der Palettenklötze charakteristische Merkmale zu detektieren und diese untereinander abzugleichen. Aufbauend auf diesem Verfahren wird ein direkter Ähnlichkeitsvergleich durchgeführt.

3.1 Experimentelles Vorgehen

Vor der Anwendung des ORB-Verfahrens werden die Bilder der Palettenklötze auf eine einheitliche Höhe gebracht. Da in Vorexperimenten die besten Resultate mit höheren Auflösungen erreicht wurden, wurde die Höhe auf 1000 Pixeln festgelegt, was in etwa der Auflösung der Bilder im Datensatz „pallet-block-502“ entspricht. Die Breite wird so gewählt, dass das ursprüngliche Seitenverhältnis erhalten bleibt.

Wie im Kapitel 2.1 beschrieben, berechnet der ORB-Algorithmus für jedes Bild einen Vektor von Merkmalspunkten (keypoints) und die dazugehörigen Deskriptoren (Fingerprints). Das ORB-Verfahren wird mittels der Bibliothek OpenCV-python (Version 4.5.5.64) mit den voreingestellten Standardparametern umgesetzt. Die Deskriptoren werden anschließend miteinander verglichen, um für jeden Deskriptor des ersten Bildes den nächstgelegenen Deskriptor des zweiten Bildes zu finden. Um konsistente Ergebnisse zu erhalten, wird dabei sichergestellt, dass nur die Übereinstimmungen (i,j) zurückgegeben werden, bei denen der i -te Deskriptor des ersten Bildes den j -ten Deskriptor des zweiten Bildes als beste Übereinstimmung hat und umgekehrt. Als Distanzmaß wird die Hammingdistanz verwendet, da die Deskriptoren binär sind, siehe Gleichung 2.1.

Auch wenn das ORB-Verfahren im Unterschied zu den Verfahren, die auf maschinellem Lernen basieren, keinen Trainingsdatensatz benötigt, wird auch diese Analyse auf dem einheitlichen Test-Datensatz bestehend aus 1270 Bildern von 127 Palettenklötzen

durchgeführt, siehe Kapitel 2.6. Das macht die untersuchten Verfahren und die berechneten Metriken vergleichbar.

3.2 Ergebnisse des ORB-Verfahrens

In Abbildung 3.1 ist beispielhaft das Ergebnis der Anwendung des ORB-Verfahrens auf zwei Palettenklötze derselben ID, aber unterschiedlicher Ausrichtung gezeigt. Die am besten zueinander passenden Merkmalspunkte (keypoints) zwischen beiden Pressspanmustern sind jeweils mit Punkten markiert und mit einer Linie verbunden.

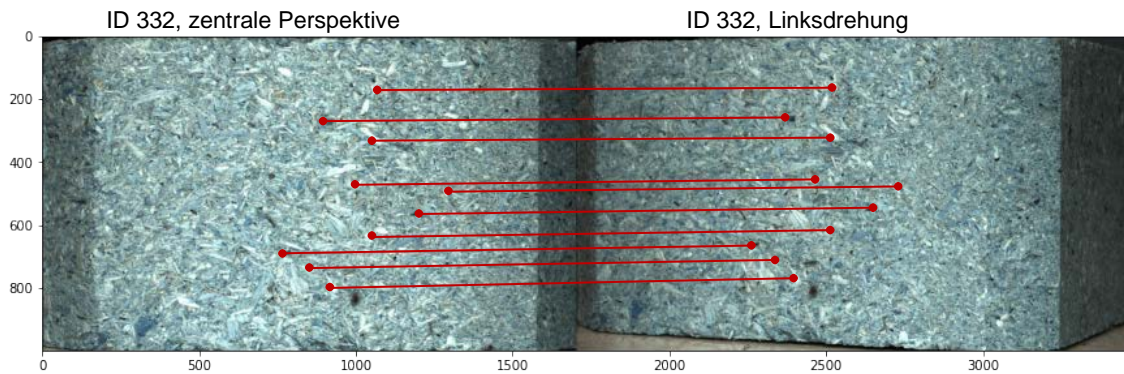


Abbildung 3.1: Ergebnis der Anwendung des ORB-Verfahrens auf zwei Palettenklötze derselben ID (332, künstliche Beleuchtung), aber unterschiedlicher Perspektive. Die 10 am besten übereinstimmenden Merkmale zwischen beiden Pressspanmustern sind jeweils mit einer Linie verbunden.

Das ORB-Verfahren wird in der Regel in Kombination mit einer Klassifizierungsmethode zur Objekterkennung eingesetzt. Im Gegensatz dazu wird in dieser Arbeit die Hamming-Distanz zwischen den Deskriptoren der am besten zueinander passenden Keypoints benutzt, um die Ähnlichkeit zweier Pressspanmuster zu berechnen. Dazu werden die Hamming-Distanzen zunächst aufsteigend geordnet. Nur die besten 5 Paare von Merkmalspunkten (keypoints) mit den größten Übereinstimmungen in den Deskriptoren (der kleinsten Hamming-Distanz) werden für die weitere Analyse benutzt. Als Distanzmaß (d) wird der normierte Mittelwert dieser 5 Hamming-Distanzen gebildet. Ein Maß für die Ähnlichkeit der Palettenklötze ergibt sich dann aus $s = 1 - d$.

Da jedes der $n = 1270$ Bilder mit allen anderen Bildern verglichen wird, werden insgesamt $n \cdot (n - 1) = 1270 \cdot 1269 = 1611630$ Deskriptor-Ähnlichkeiten berechnet. In Abbildung 3.2 sind die Häufigkeitsverteilungen der Deskriptor-Ähnlichkeiten von Palettenklötzen derselben ID und solcher unterschiedlicher IDs gegenübergestellt. Aus den Histogrammen geht hervor, dass Ähnlichkeitswerte $< 0,4$ in beiden Häufigkeitsverteilungen vorkommen. Diese Überschneidung in beiden Histogrammen führt zwangsweise zu Fehlern in der Klassifizierung, da die Wahl eines Schwellenwertes getroffen werden muss, oberhalb dessen die Palettenklötze als übereinstimmend klassifiziert werden. Würde der Schwellenwert auf $0,4$

festgelegt, so dass alle verglichenen Pressspanmuster-Paare, deren Deskriptor-Ähnlichkeit größer als 0,4 ist, als „Match“ gewertet werden, so ergibt sich eine False-Positive-Rate von $2,75 \cdot 10^{-5}$ und eine True-Positive-Rate von 0,793. Daraus geht hervor, dass mit dem ORB-Verfahren nahezu 80% der Palettenklötze richtig identifiziert werden können und zugleich der Anteil an fälschlicherweise als übereinstimmend klassifizierter Palettenklötze annähernd 0 ist. Dies soll aber nur ein grobes Gefühl für die Güte der Klassifizierung geben. Zum besseren Vergleich der in dieser Arbeit angewandten Verfahren werden zwei

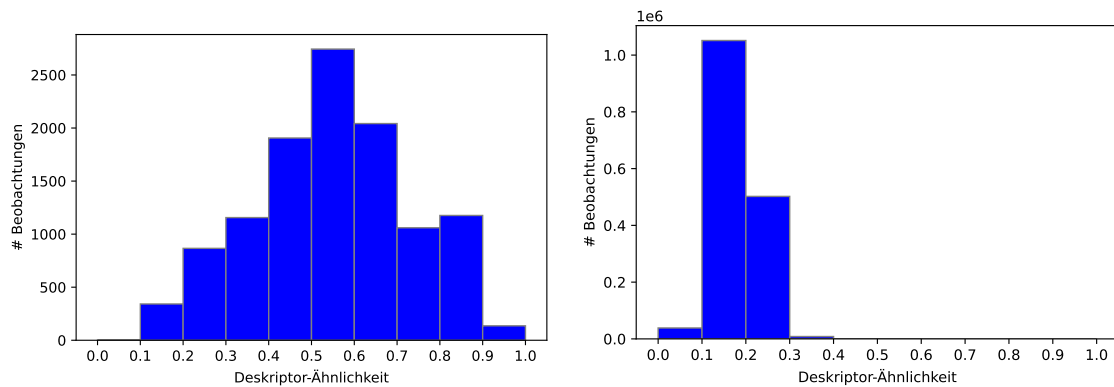
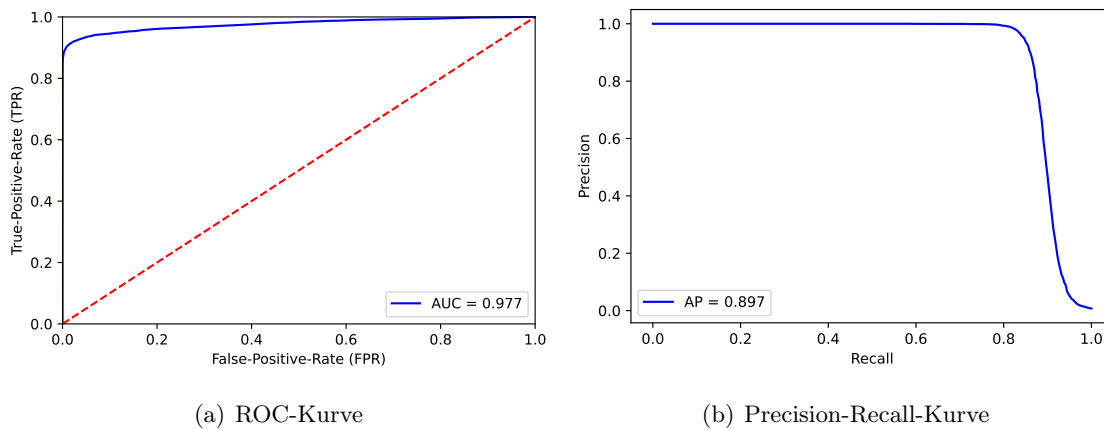


Abbildung 3.2: Histogramm über die Ähnlichkeit der Deskriptoren von Palettenklötzen derselben ID (links) und solcher unterschiedlicher ID (rechts) bei Anwendung des ORB-Verfahrens auf den Test-Datensatz.

vom Schwellenwert unabhängige Maße diagnostischer Güte herangezogen: der AUC-Wert, welcher die Fläche unterhalb der ROC-Kurve beschreibt und der Average Precision (AP)-Wert, der ein Maß für die Fläche unterhalb der Precision-Recall-Kurve ist. Beide Maße werden im Abschnitt 2.7 genauer beschrieben. Wird das ORB-Verfahren auf dem Test-Datensatz angewandt, so ergibt sich ein AUC-Wert von 0,977 und ein AP-Wert von 0,897. Zur Veranschaulichung sind die entsprechenden Kurven (ROC (a) und Precision-Recall (b)) in Abbildung 3.3 dargestellt.

Die durchschnittliche Laufzeit für die Berechnung der Keypoints und Deskriptoren beträgt 0,051 Sekunden pro Bild. Für die Berechnung des Ähnlichkeitsmaßes basierend auf den Hamming-Distanzen zwischen den Deskriptoren werden etwa 0,49 Sekunden für 1000 Vergleiche benötigt.

Da der Datensatz „pallet-block-502“ (siehe Abschnitt 2.6) mittels unterschiedlicher Kamerasysteme aufgenommen wurde, liegen Bilder in unterschiedlichen Datenformaten (bmp und jpg) vor. Es ist zudem anzunehmen, dass die Bilder aufgrund der Bauweise der Kameras und Ausstattung mit unterschiedlicher Sensorik voneinander abweichende Charakteristika haben. Um einen möglichen Einfluss des Dateiformats oder des Kamerasystems auf die Performance von ORB zu untersuchen, wird das ORB-Verfahren in einem nächsten Schritt nur auf den bmp-Bildern und anschließend nur auf jpg-Bildern angewandt. Dazu werden

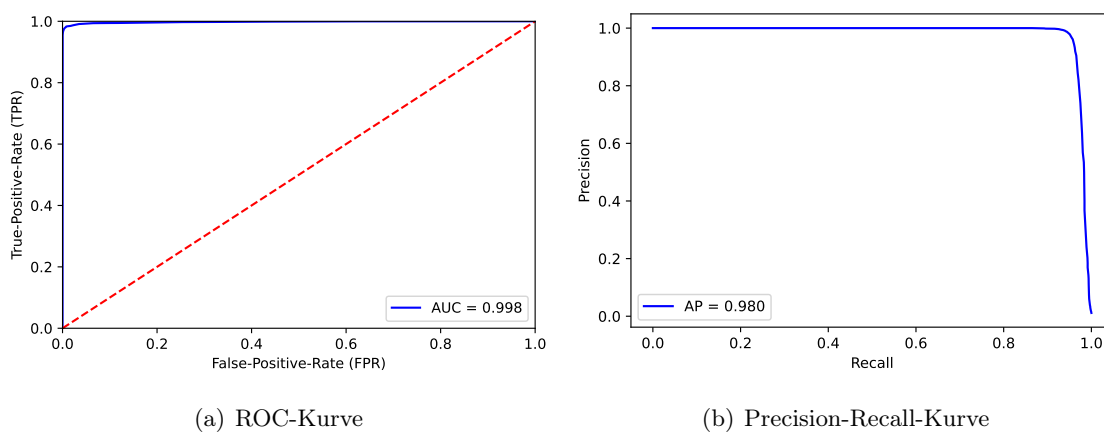


(a) ROC-Kurve

(b) Precision-Recall-Kurve

Abbildung 3.3: ROC-Kurve und Precision-Recall-Kurve bei Anwendung des ORB-Verfahrens auf den Test-Datensatz.

zunächst alle 1020 bmp-Bilder von 102 verschiedenen Palettenklötzen (ID 1 bis 102) zu einem Sub-Datensatz zusammengefasst. Die ORB-Analyse auf diesem Datensatz ergibt eine



(a) ROC-Kurve

(b) Precision-Recall-Kurve

Abbildung 3.4: ROC-Kurve und Precision-Recall-Kurve bei Anwendung des ORB-Verfahrens auf dem bmp-Datensatz.

deutlich bessere Performance mit einem AUC-Wert von 0,998 und einer AP von 0,980. Die entsprechenden Kurven sind in Abbildung 3.4 dargestellt.

Aus den Histogrammen in Abbildung 3.5 geht hervor, dass es kaum noch Überlappungen gibt zwischen der Deskriptor-Ähnlichkeit von übereinstimmenden Palettenklötzen und solcher unterschiedlicher Palettenklötze. Daraus lässt sich die gute Performance des ORB-Ansatzes auf dem bmp-Datensatz ableiten.

Zur Vollständigkeit wird das ORB-Verfahren schließlich auf einem nur aus jpg-Bildern bestehenden Datensatz von Palettenklötzen angewandt. Dieser jpg-Datensatz wird aus 1020 jpg-Bildern der Palettenklötze mit den IDs 103 bis 204 gebildet. Die daraus resul-

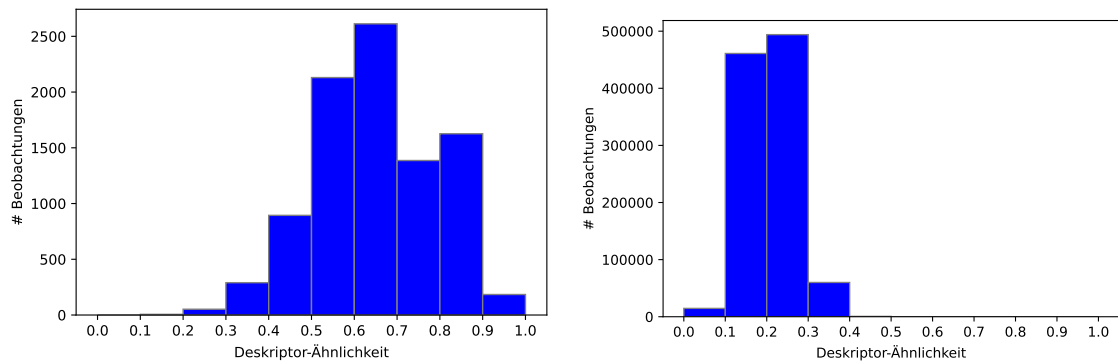


Abbildung 3.5: Histogramm über die Deskriptor-Ähnlichkeit von Palettenklötzen derselben ID (links) und solcher unterschiedlicher ID (rechts) bei Anwendung des ORB-Verfahrens auf den bmp-Datensatz.

tierenden Gütemaße betragen: $AUC = 0,965$ und $AP = 0,856$. Zusammenfassend sind die Gütemaße für alle untersuchten Datensätze in Tabelle 3.1 nochmals dargestellt. Der jpg-Datensatz schneidet im Vergleich zu den anderen Datensätzen deutlich schlechter ab.

Tabelle 3.1: Gütemaße als Resultat der Anwendung des ORB-Verfahrens auf verschiedene Datensätze von Palettenklotzbildern.

Datensatz	AUC	AP
Test-Datensatz	0,977	0,897
bmp-Datensatz	0,998	0,980
jpg-Datensatz	0,965	0,856

Aus den Ergebnissen lässt sich nicht trennscharf nach Bildformat¹ und Sensorik², aber zumindest allgemein feststellen, dass beide Teildatensätze deutliche Performanceunterschiede bezüglich des ORB-Verfahrens aufweisen. Diese Beobachtung wäre bei einem tatsächlichen Einsatz des Verfahrens im Anwendungskontext zu berücksichtigen. Dieser liegt aber hier nicht im Fokus der Arbeit, da das ORB-Verfahren als Referenzansatz genutzt wird.

¹vielleicht ausgelöst durch Artefakte der Kompression im jpg-Format

²eventuell bedingt durch Qualitätsunterschiede in der verbauten Elektronik

Kapitel 4

Ansatz mit Anomaliekarten

Die in diesem Kapitel beschriebenen Methoden zur Re-Identifikation von Palettenklötzen basieren grundlegend auf der Erkennung von Anomalien. Um diese Anomalien zu bestimmen, wird (wie einleitend dargestellt) ein Autoencoder mit bekannten Pressspanmustern trainiert. Dies führt zu dem Erlernen von typischen „Eigenschaften“ dieser Trainingsdaten. Dementsprechend können abweichende Strukturen vom Autoencoder nur unzureichend rekonstruiert werden. Der Rekonstruktionsfehler ist also ein geeignetes Mittel, um Anomalien im Vergleich zu einem „Standardpressspanmuster“ zu beschreiben. Wir können Anomalien bzgl. ihrer räumlichen Anordnung sogar als Fingerabdruck der identifizierten Unterschiede verstehen. Dieser Gedanke ist in sogenannten „Anomaliekarten“ fortgeschrieben. Diese Karten entstehen dadurch, dass die eingegebenen Palettenbilder in Teilbilder unterteilt werden. Diese werden in den Autoencoder gegeben und die Fehler der Teilbildrekonstruktion in einer Karte entsprechend der ursprünglichen räumlichen Anordnung der Teilbilder wieder zusammengesetzt. Damit ergibt sich eine grobe räumliche Repräsentation des Rekonstruktionsfehlers. Diese Konstruktion der Anomaliekarte und des zugrundeliegenden Autoencoders wird im Folgenden dargestellt (Abschnitt 4.1). Danach werden zwei Verfahren vorgestellt, die diese Anomaliekarten als zentrale Bestandteile der Re-Identifikation einsetzen: ein iteratives Verfahren direkt auf den Anomaliekarten (Abschnitt 4.2) und ein auf Siamesischen Neuronalen Netzen (SNN) basierender Ansatz (Abschnitt 4.3).

4.1 Vorgehen zur Erzeugung der Anomaliekarten

Im ersten Schritt zur Erzeugung einer Anomaliekarte wird das Bild eines Palettenklotzes in überlappende Teilbilder unterteilt, siehe Abbildung 4.1. Diese Teilbilder werden im zweiten Schritt mit Hilfe eines Autoencoders (siehe Abschnitt 2.4) komprimiert (Encoder) und in eine niederdimensionale Repräsentation abgebildet, welche die wichtigsten Merkmale der Eingabedaten beibehält. Im Folgenden sorgt der Decoder für die Rekonstruktion der Eingabedaten. Im Falle von Anomalien im Pressspanmuster, schlägt die Rekonstruktion

fehl und führt zu einem hohen Rekonstruktionsfehler. Dieser Rekonstruktionsfehler, hier auch Anomaliewert genannt, wird im letzten Schritt auf eine räumliche Karte abgebildet und als Fingerabdruck für die Wiedererkennung von Paletten verwendet.

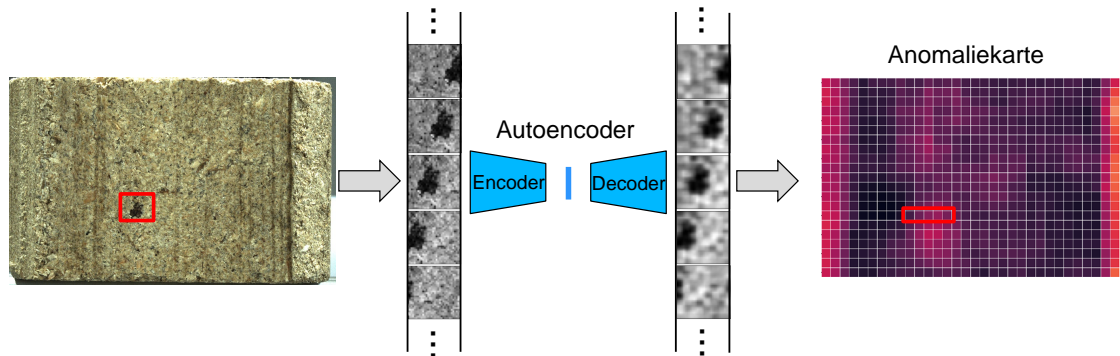


Abbildung 4.1: Vorgehen zur Erzeugung der Anomaliekarten. Das Bild des Palettenklotzes wird in kleine überlappende Bilder unterteilt (dargestellt sind Teilbilder aus dem rot markierten Bereich des Palettenklotzes). Die Teilbilder werden mit Hilfe des Autoencoders komprimiert und rekonstruiert. Aus den Rekonstruktionsfehlern wird die Anomaliekarte derart zusammengesetzt, dass die räumliche Darstellung genau dem Originalbild entspricht. Die Anomaliekarte ist in Form einer Heatmap dargestellt. Der Beitrag der gezeigten Teilbilder zur Anomaliekarte ist rot umrandet.

4.1.1 Architektur des Autoencoders

Es wurden mehrere Autoencoder-Architekturen getestet, die besten experimentellen Ergebnisse wurden jedoch mit der im Folgenden beschriebenen Architektur erzielt. Der Encoder umfasst zunächst ein Convolutional-Layer mit 16 Filtern der Größe 3×3 und einer Schrittweite (engl. strides) von 2. Danach wird eine MaxPooling-Schicht mit einer Kernelgröße von 2×2 angewendet, um die Dimensionalität effektiv um den Faktor 2 zu verkleinern. Die letzte Schicht bildet ein zweites Convolutional-Layer mit 8 Filtern der Größe 3×3 und einer Schrittweite von 2. Die Convolutional-Layer verwenden jeweils Same-Padding und die Rectified Linear Unit (ReLU)-Funktion (siehe Gleichung 2.2) als Aktivierungsfunktion.

Der Decoder ist folgendermaßen aufgebaut, um die Dimension der Kernschicht ($4 \times 4 \times 8$) auf die ursprüngliche Größe ($32 \times 32 \times 1$) zurückzurechnen: Die ersten beiden Schichten bilden zwei Transposed Convolutional-Layer mit 8 bzw. 16 Filtern der Größe 3×3 , einer Schrittweite von 2 und ReLU als Aktivierungsfunktion. Daraufhin schließt sich ein Upsampling-Layer mit einer Kernelgröße von 2×2 an. Als letzter Schritt wird ein Convolutional-Layer mit nur einem Filter der Größe 3×3 und einer Schrittweite von 1 angewendet, um wieder auf die Eingabegröße von $32 \times 32 \times 1$ zu kommen. Hier wird die sigmoide Aktivierungsfunktion [10] verwendet, um die Ausgabewerte zwischen 0 und 1 zu normieren. Die gesamte Architektur des Autoencoders ist in Abbildung 4.2 dargestellt.

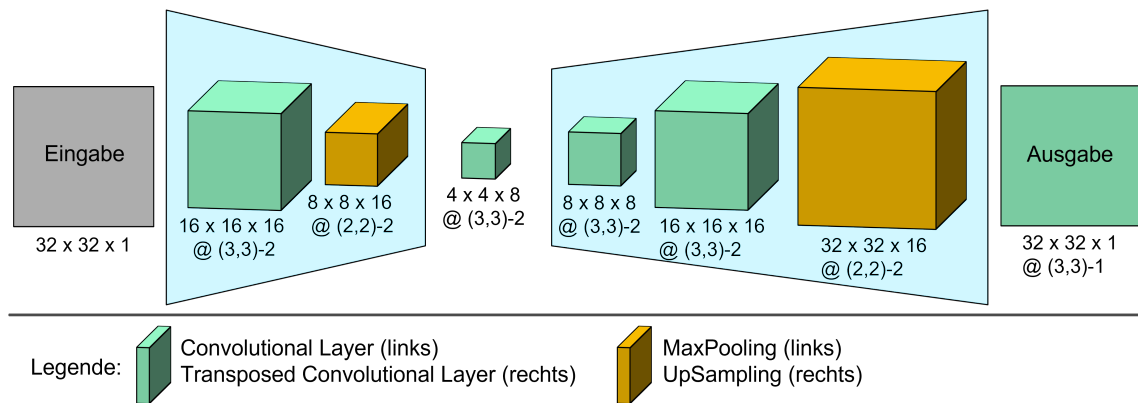


Abbildung 4.2: Architektur des Autoencoders für die Berechnung der Anomaliekarten. Die Zahlenangaben an den Layern haben folgende Bedeutung: Höhe x Breite x Tiefe @ (Filterhöhe x Filterbreite)-Stride. Alle Schichten verwenden Same-Padding und ReLU als Aktivierungsfunktion. Nur die letzte Schicht verwendet die sigmoide Aktivierungsfunktion.

4.1.2 Training des Autoencoders

Der Autoencoder wird auf dem einheitlichen Trainingsdatensatz bestehend aus 375 pseudo-randomisiert ausgewählten Palettenklötzen, siehe Kapitel 2.6, trainiert. Da der Datensatz aufgrund der variierenden Kameraperspektiven und Beleuchtungsbedingungen aus jeweils 10 Bildern pro Palettenklotz besteht, setzt sich der Trainingsdatensatz somit aus 3.750 Bildern zusammen. Der Test-Datensatz enthält entsprechend die restlichen 1.270 Bilder.

Zur gradientenbasierten Optimierung wird Adam [15] verwendet und als Verlustfunktion wird die mittlere quadratische Abweichung (MSE, siehe Gleichung 2.3) in Bezug auf das Eingabe-Bild und dessen Rekonstruktion eingesetzt. Die verwendete Batch-Größe ist 32. Da ein Bild aus 693 Teilbildern besteht und der Trainingsdatensatz 3.750 Bilder beinhaltet, enthält eine Epoche $\frac{693 \cdot 3.750}{32} = 81.211$ Batches. Aufgrund der großen Anzahl an Batches wird nur eine Epoche zum Training benötigt. Die Lernrate ist 0,001.

4.1.3 Berechnung der Anomaliekarten

Vor der Berechnung der Anomaliekarten werden die Bilder der Palettenklötze auf eine einheitliche Größe von 224 x 320 Pixeln gebracht. Jedes Bild wird dann in 32 x 32 Pixel große Teilbilder unterteilt, die sich sowohl vertikal als auch horizontal zu 75% überlappen. Da die Ränder der Palettenbilder häufig Hintergrund beinhalten, werden an allen Rändern zwei Reihen dieser Teilbilder in die Berechnung der Anomaliekarten nicht mit einbezogen. Insgesamt ergeben sich mit diesem Vorgehen für jedes Palettenbild $21 \cdot 33 = 693$ Teilbilder. Diese werden in Graustufenbilder umgewandelt, da wir davon ausgehen, dass nicht die Farbgebung, sondern die Struktur des Pressspanmusters für die Erkennung entscheidend ist. Die Grauwerte der Pixel werden auf Werte zwischen 0,0 (Grauwert 0) und 1,0 (Grauwert 255) normiert. Zur Berechnung der Anomaliekarten gelangen die Teilbilder einzeln in

den Autoencoder, so dass für jedes Teilbild basierend auf dem Rekonstruktionsfehler ein Anomaliewert berechnet werden kann.

Für die Berechnung der Anomaliekarten wird jedes der $1.270 \cdot 693 = 880.110$ Teilbilder aus dem Test-Datensatz in den zuvor trainierten Autoencoder gegeben. Dadurch wird jedes Bild zunächst encodiert und anschließend mit dem Decoder wieder hergestellt. Anschließend kann der Rekonstruktionsfehler gemäß Gleichung 2.4 als mittlere quadratische Abweichung zwischen dem 33×33 Eingabebild und der Vorhersage des Autoencoders berechnet werden. Mit diesem Verfahren erhält man für jedes der 693 Teilbilder eines Palettenbildes einen Anomaliewert. Die Anomaliekarte eines Palettenbildes wird anschließend aus den 693 Anomaliewerten der Teilbilder zusammengesetzt, wobei die ursprüngliche zweidimensionale Anordnung der Teilbilder der Anordnung der Anomaliewerte auf der 21×33 großen Anomaliekarte entspricht. Für die Normierung der Anomaliewerte wird die z -Standardisierung [9] angewendet. Dazu wird von jedem Anomaliewert x_{ij} der arithmetische Mittelwert (über alle Werte einer Karte) subtrahiert und die resultierende Differenz durch die Standardabweichung s_x dividiert:

$$z_{ij} = \frac{x_{ij} - \bar{x}}{s_x}. \quad (4.1)$$

Die resultierenden z -Werte geben in Einheiten von Standardabweichungen an, wie weit der Anomaliewert vom Mittelwert entfernt ist. Diese liegen im Allgemeinen zwischen -3 und 3. Deshalb werden z -Werte größer als 3 bzw. kleiner als -3 auf 3 bzw. -3 gesetzt. Um Anomaliewerte zwischen 0 und 1 zu erhalten, wird auf die so berechneten z -Werte ein Wert von 3 addiert und die resultierende Summe durch 6 dividiert.

Da Kanten von Palettenklötzen häufig zu hohen Anomaliewerten führen, wodurch innerhalb der Anomaliekarten gesamte Spalten stark aktiviert sind, wird die z -Standardisierung alternativ so durchgeführt, dass hohe Aktivierungen einer gesamten Spalte herausgerechnet werden. Dazu wird im Gegensatz zur obigen Berechnung zur Normierung nicht der arithmetische Mittelwert über alle Anomaliewerte einer Karte benutzt, sondern lediglich über die Werte innerhalb derselben Spalte wie der zu normierende Wert. Dementsprechend wird auch die Standardabweichung spaltenweise berechnet:

$$z'_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}. \quad (4.2)$$

Bei den im Folgenden vorgestellten Verfahren wird jeweils erläutert, welche Art der Normalisierung zur Berechnung der Anomaliekarten verwendet wird.

4.2 Iteratives Verfahren zum Vergleich von Anomaliekarten

Im Folgenden wird ein einfaches Verfahren entwickelt, welches Anomaliekarten vergleicht, indem es sie gegeneinander verschiebt, um die bestmögliche Übereinstimmung der Anomaliewerte zu erreichen. Dieses Verfahren wird als Baseline-Verfahren eingeführt und später

mit einem auf maschinellem Lernen basierenden Ansatz zur Wiedererkennung von Anomaliekarten verglichen. In den beiden folgenden Abschnitten werden das Verfahren und die Ergebnisse der Anwendung des Verfahrens auf den Datensatz „pallet-block-502“ vorgestellt.

4.2.1 Darstellung des Verfahrens

Wie in Abbildung 4.3 dargestellt, wird zum Vergleich zweier Anomaliekarten zunächst ein (15 x 11) großes Fenster zentriert über die Referenz-Anomaliekarte gelegt. Dieses Fenster, das die Anomaliewerte der Referenzkarte beinhaltet, wird dann (vertikal zentriert) in horizontaler Richtung über die zu vergleichende Karte bewegt und die Pearson-Korrelation [9] über die Anomaliewerte beider Karten innerhalb des Fensters berechnet. Die beste Ausrichtung mit maximalem Korrelationskoeffizienten legt die optimale Position des Fensters in horizontaler Richtung fest. Von dieser Position aus wird das Fenster im nächsten Schritt in vertikaler Richtung verschoben, um zusätzlich (unter Verwendung der Pearson-Korrelation) die optimale vertikale Ausrichtung des Fensters zu bestimmen. Im letzten Schritt wird das Fenster unter Berücksichtigung der zuvor ermittelten optimalen Ausrichtung in horizontaler und vertikaler Richtung vergrößert, bis die maximale Überlappung der beiden Anomaliekarten erreicht ist. Daraus ergeben sich zwei gleich große Ausschnitte aus den ursprünglichen Anomaliekarten. Der Pearson-Korrelationskoeffizient über die Anomaliewerte innerhalb dieser Ausschnitte wird als Maß für die Ähnlichkeit der zugrunde liegenden Palettenklötze verwendet.

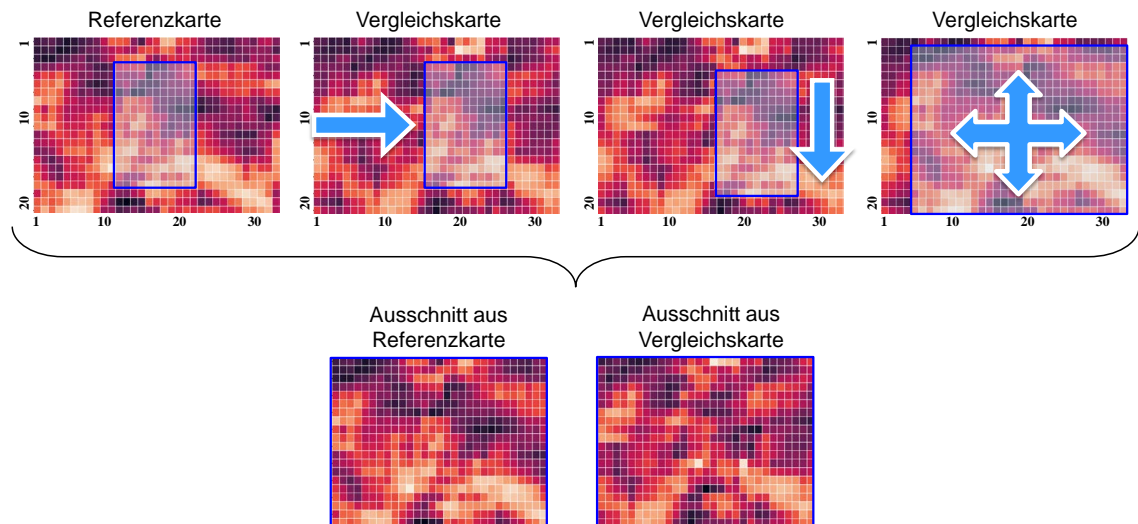


Abbildung 4.3: Iteratives Verfahren zum Vergleich zweier Anomaliekarten. Auf der Referenzkarte wird ein Fenster zentriert, das zunächst horizontal und anschließend vertikal auf der zu vergleichenden Anomaliekarte verschoben wird, um die Position mit der besten Übereinstimmung der Anomaliewerte zu finden. Von dieser optimalen Position aus, wird das Fenster so weit wie möglich vergrößert, so dass zwei Ausschnitte der zu vergleichenden Anomaliekarten entstehen (unten), deren Korrelation als Ähnlichkeitsmaß verwendet wird.

4.2.2 Ergebnisse des iterativen Vergleichs von Anomaliekarten

Zur Berechnung der Anomaliekarten werden die Palettenbilder, wie zu Beginn dieses Kapitels beschrieben, in viele Teilbilder unterteilt, wodurch über 80.000 Batches (siehe Abschnitt 4.1.1) entstehen, mit denen der Autoencoder trainiert wird. Aufgrund dieser großen Anzahl von Batches im Trainingsdatensatz wird nur eine Epoche zum Training benötigt.

Um die Entwicklung der Verlustwerte (engl. loss) während des Trainings zu veranschaulichen, zeigt Abbildung 4.4 ein Diagramm, in dem die Verlustwerte der Trainingsdaten gegen die Batches des Training-Datensatzes aufgetragen sind. Nach Abschluss des Trainings beträgt der Verlustwert des Trainingsdatensatzes 0.0086. Der Verlustwert des Test-Datensatzes erreicht einen ähnlich guten Wert von 0.0082.

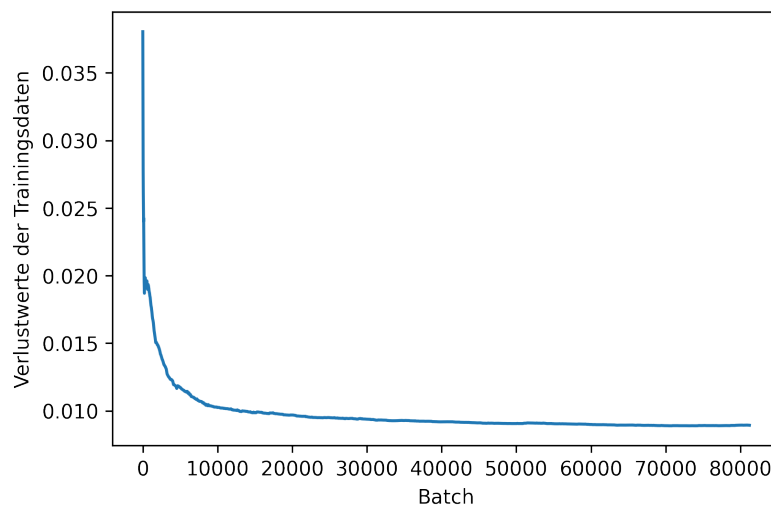


Abbildung 4.4: Verlustwerte des Trainings-Datensatzes im Verlauf des Trainings.

Der trainierte Autoencoder soll im Folgenden genutzt werden, um anhand des Rekonstruktionsfehlers Anomalien zu erkennen und Anomaliekarten zu erzeugen. Es soll aber zunächst anhand eines Beispiels betrachtet werden, wie gut der Autoencoder die Teilbilder des Testdatensatzes rekonstruieren kann. In Abbildung 4.5 sind Originalbilder und Rekonstruktionen des Palettenklotzes mit der ID 18 dargestellt. Man sieht, dass wichtige Merkmale des Palettenklotzes in der Rekonstruktion erhalten bleiben.

Im nächsten Schritt werden Rekonstruktionsfehler als mittlere quadratische Abweichung zwischen den 33 x 33 Pixeln des Eingangsbildes und der Ausgabe des Autoencoders berechnet. Die Anomaliekarte eines Palettenbildes setzt sich dann aus den Anomaliewerten der Teilbilder zusammen, wobei die ursprüngliche zweidimensionale Anordnung der Teilbilder der Anordnung der Anomaliewerte auf der 21 x 33 Anomaliekarte entspricht. In Abbildung 4.6 sind beispielhaft alle 10 Anomaliekarten eines Palettenklotzes als Heatmap visualisiert. Man sieht, dass bestimmte Bereiche in allen Anomaliekarten einheitlich aktiviert sind (helle Bereiche in der Heatmap). Diese Aktivierungen gehen auf Charakteristika des

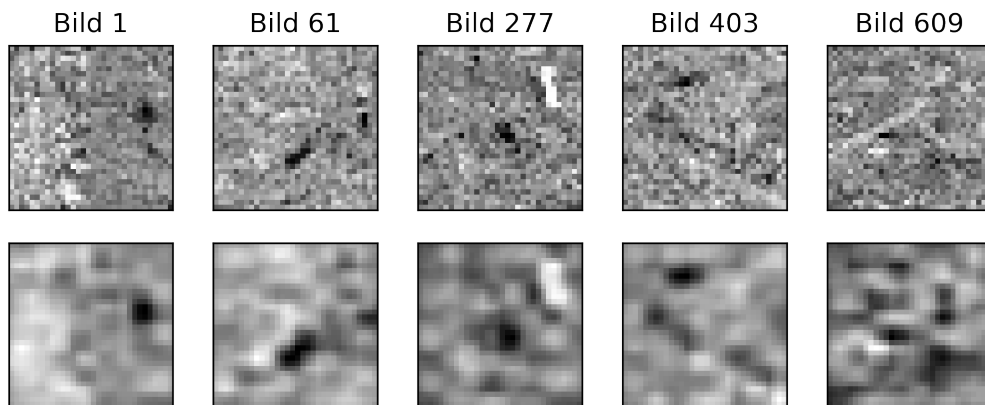


Abbildung 4.5: Rekonstruktion von Teilbildern des Palettenklotzes mit der ID 18. In der oberen Reihe sind die Originalbilder und in der unteren Reihe die zugehörigen Rekonstruktionen abgebildet.

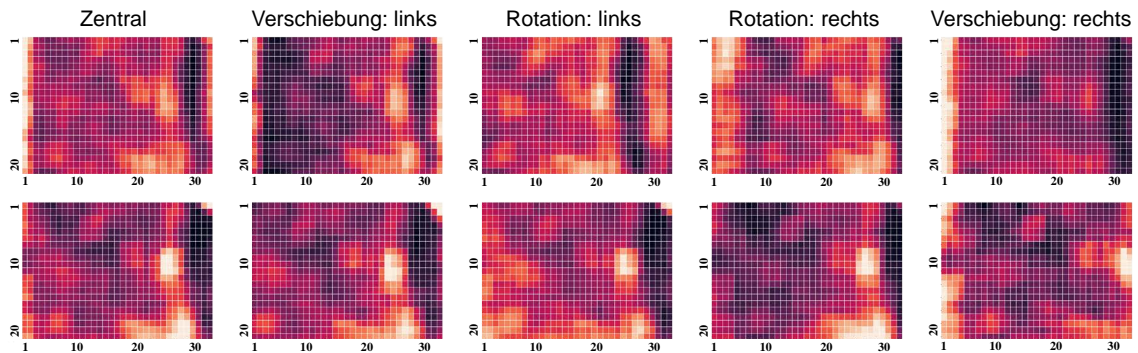


Abbildung 4.6: Anomaliekarten von Bildern eines Palettenklotzes (ID 6) aller 5 Kameraperspektiven, aufgenommen unter künstlicher Beleuchtung (oben) und mit natürlicher Beleuchtung (unten). Die Anomaliekarten sind als Heatmap visualisiert, wobei helle Bereiche Anomalien aufzeigen.

Palettenklotzes zurück und sollen für die Wiedererkennung genutzt werden. Darüber hinaus weisen die Anomaliekarten spaltenweise Aktivierungsunterschiede auf, die auf die unterschiedlichen Kameraperspektiven und die damit verbundenen Beleuchtungsveränderungen sowie auf die Kanten des Palettenklotzes zurückzuführen sind. Diese Aktivierungsunterschiede erschweren den hier durchzuführenden Vergleich von Anomaliekarten, welcher die Korrelation der Anomaliewerte als Ähnlichkeitsmaß verwendet. Die spaltenweise Normalisierung der Anomaliekarten, siehe Gleichung 4.2, soll die Anomaliekarten ungeachtet der Kameraperspektive einheitlicher machen. In Abbildung 4.7 sind die spaltenweise normalisierten Anomaliekarten des Palettenklotzes mit der ID 6 dargestellt. Im direkten Vergleich macht die spaltenweise Normalisierung die Anomaliekarten homogener, abgesehen von den durch die unterschiedlichen Kameraperspektiven verursachten Verschiebungen.

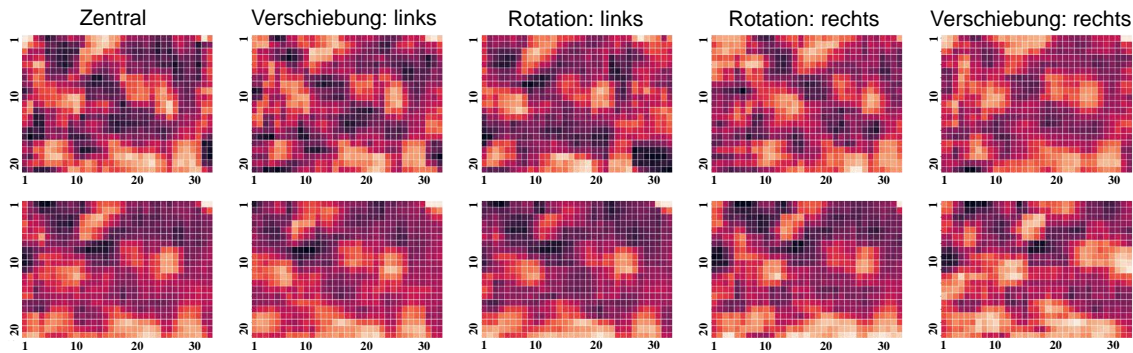


Abbildung 4.7: Spaltenweise normalisierte Anomaliekarten von Bildern eines Palettenklotzes (ID 6) aller 5 Kameraperspektiven, aufgenommen unter künstlicher Beleuchtung (oben) und mit natürlicher Beleuchtung (unten). Die Anomaliekarten sind als Heatmap visualisiert, wobei helle Bereiche Anomalien aufzeigen.

Im weiteren Vorgehen werden die Anomaliekarten als Fingerabdruck für die Wiedererkennung von Paletten genutzt. Wie im Abschnitt 4.2 genauer beschrieben, werden die zu vergleichenden Anomaliekarten dazu gegeneinander verschoben, um die bestmögliche Übereinstimmung der Anomaliewerte zu erreichen. Daraus ergeben sich zwei gleich große Ausschnitte mit Anomaliewerten, deren Pearson-Korrelation als Maß für die Ähnlichkeit der zugrunde liegenden Palettenklötze verwendet wird.

Wenn das iterative Verfahren zunächst auf die Anomaliekarten aus dem Testdatensatz angewandt wird, die gemäß der z -Standardisierung normalisiert wurden (Gleichung 4.1), erhält man einen AUC-Wert von 0,637 und einen Average Precision (AP)-Wert von 0,026. Durch die spaltenweise Normalisierung der Anomaliekarten wird ein AUC-Wert von 0,853 und ein AP-Wert von 0,223 erreicht, was die Güte des Verfahrens verbessert. Dies entspricht der visuellen Wahrnehmung.

Lässt man hingegen den letzten Schritt, d. h. die Vergrößerung des Fensters bis zur maximalen Überlappung (Bild oben rechts in Abbildung 4.3), weg und wendet die Pearson-Korrelation direkt auf die Anomaliewerte des in die optimale Position verschobenen 15×11 -Fensters an, so nimmt die Güte des Verfahrens ab. Man erhält einen AUC-Wert von nur 0,794 und einen AP-Wert von 0,129. Dies lässt sich durch zufällige Übereinstimmungen zwischen Anomaliekarten innerhalb des kleinen Fensters erklären. Schließlich wurde das Fenster so ausgerichtet, dass die Korrelation maximal ist. Wird das Fenster anschließend auf die maximale Überlappung der zu vergleichenden Karten vergrößert, fallen viele der zufälligen Übereinstimmungen weg, während Anomaliewerte von Palettenklötzen derselben ID auch im Bereich des vergrößerten Fensters gut korrelieren.

Die benötigte Laufzeit setzt sich aus der Zeit für die Berechnung der Anomaliekarten (etwa 0,1 Sekunden) und für den anschließenden iterativen Vergleich der Karten (1,57 Sekunden für 1.000 Vergleiche) zusammen.

Dieser einfache Ansatz zum Vergleich von Anomaliekarten schneidet im Vergleich zum ORB-Algorithmus deutlich schlechter ab, siehe auch Kapitel 3.1. Mit dem ORB-Algorithmus wurde auf dem Trainingsdatensatz ein AUC-Wert von 0,977 und ein AP-Wert von 0,897 erreicht. Dies sagt jedoch noch nichts über den Informationsgehalt der Anomaliekarten aus und darüber, ob dieser für die eindeutige Identifizierung von Palettenklötzen nicht ausreichend ist. Betrachtet man die Heatmaps visuell, so scheinen Anomaliekarten eine vielversprechende niedrigdimensionale Darstellung von Palettenklötzen zu sein. Daher stellt sich die Frage, ob maschinelle Lernverfahren bei der Identifikation von Anomaliekarten bessere Ergebnisse erzielen können. Für diese Aufgabe bieten sich Siamesische Netze an, die automatisch wesentliche Unterscheidungsmerkmale erfassen können und ebenso wie das iterative Verfahren eine Ähnlichkeitsmetrik bereitstellen [16]. Daher wird im nächsten Ansatz ein Siamesisches Netz verwendet, um die Ähnlichkeiten von Anomaliekarten zu bestimmen.

4.3 Anwendung eines SNNs auf Anomaliekarten

Siamesische Neuronale Netze sind darauf spezialisiert, aus den Eingangsdaten automatisch wichtige Unterscheidungsmerkmale zu erfassen und durch den Vergleich dieser Merkmale eine Ähnlichkeitsmetrik zu liefern. Sie erscheinen somit hervorragend dafür geeignet, Ähnlichkeiten bezüglich zuvor ermittelter Anomaliekarten, welche die räumliche Anordnung sowie die Ausprägung von Anomalien im Pressspanmuster kodieren, zu erkennen.

4.3.1 Architektur des Siamesischen Neuronalen Netzes

Im Vorfeld wurden verschiedene Architekturen von SNNs getestet. Die für die Re-Identifikation von Pressspanmustern am besten geeignete Architektur ist in Abbildung 4.8 dargestellt. Da das Netz aus drei identischen parallelen CNNs mit gekoppelten Gewichten besteht, wird nur eines gezeigt. Die CNNs haben eine sehr einfache Struktur und bestehen aus 3 Blöcken von Convolutional-Schichten mit jeweils 128 Filtern der Größe 3×3 . Alle Convolutional-Schichten verwenden ReLU als Aktivierungsfunktion und kein Padding. Die erste Schicht hat eine Schrittweite (engl. *strides*) von 2, alle anderen haben eine Schrittweite von 1. Zwischen zwei Standard-Convolutional-Schichten (im zweiten und dritten Block) wird jeweils eine zusätzliche 1×1 -Convolution hinzugefügt (inspiriert durch [22]), die dem Netz auf effiziente Weise Nichtlinearität hinzufügt und Merkmale über Merkmalskarten hinweg räumlich kombiniert. Die ersten beiden Blöcke enden mit einer Batch Normalization-Schicht [13]. Um im dritten Block von einer Convolutional- zu einer Fully-Connected Struktur überzugehen, müssen die Merkmalskarten zuvor in einen langen, „flachen“ Vektor transformiert werden. Die nachfolgende Fully-Connected-Schicht reduziert die Größe dieses Vektors auf eine Einbettungsgröße (engl. *embedding size*) von 900. In Anlehnung an das Face-Net [33] führt die letzte Schicht eine L2-Normalisierung

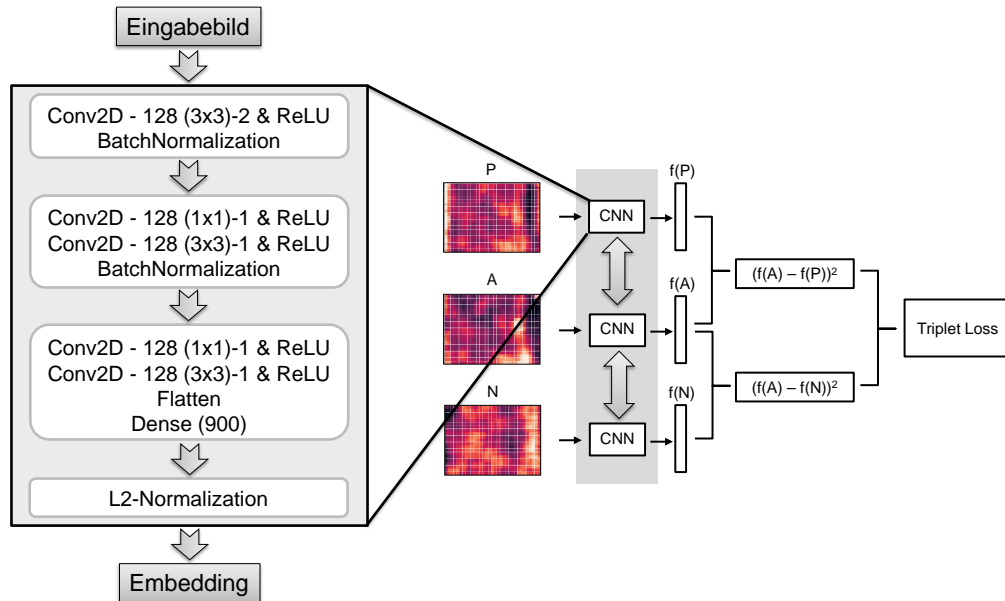


Abbildung 4.8: Architektur des SNNs. Dargestellt ist eines der drei CNNs. Die Zahlenangaben an den Convolutional-Schichten haben folgende Bedeutung: Anzahl Filter (Filterhöhe x Filterbreite)-Stride & Aktivierungsfunktion. Alle Schichten verwenden kein Padding. Die Dense-Schicht hat keine Aktivierungsfunktion.

durch. Der quadrierte euklidische Abstand zwischen normalisierten Vektoren ist proportional zu ihrer Kosinus-Ähnlichkeit.

Die Gewichte werden nach der Random Uniform¹-Methode initialisiert. Zusätzlich wird in allen Schichten eine L2-Regularisierung² ($\lambda = 0,0002$ bzw. $\lambda = 0,001$ in der letzten Dense-Schicht) durchgeführt, um einer Überanpassung des Modells entgegenzuwirken.

4.3.2 Triplet Loss als Verlustfunktion

Im Rahmen dieser Arbeit vergleicht die Triplet Loss-Verlustfunktion eine Referenz-Anomaliekarte eines Palettenklotzes (Anker) mit der Anomaliekarte eines Palettenklotzes übereinstimmender ID (Positiv) und eines Palettenklotzes unterschiedlicher ID (Negativ), siehe Abbildung 4.8 (rechts). Ziel dieser Verlustfunktion ist es, die Anomaliekarten in einen Merkmalsraum derart einzubetten, dass der quadratische euklidische Abstand für positive Paare (Anker, Positiv) klein ist, während der quadratische Abstand für negative Paare (Anker, Negativ) groß ist. Zusätzlich wird versucht, das positive (übereinstimmende) Paar von dem negativen um einen vorgegebenen Mindestabstand zu trennen. Dies wird in Gleichung 2.6 mit Hilfe eines sogenannten Margin-Terms m erreicht. Dieser erlaubt es, dass kleine Distanzen (Unterschiede) zwischen Palettenklötzen derselben ID durchaus zulässig sind, während der Abstand und damit die Unterscheidbarkeit zu anderen Identitäten erhal-

¹https://www.tensorflow.org/api_docs/python/tf/keras/initializers/RandomUniform

²https://www.tensorflow.org/api_docs/python/tf/keras/regularizers/L2

ten bleibt. Der Margin-Term wird in dieser Arbeit auf $m = 1$ gesetzt. Die zu minimierende Verlustfunktion ist dann, wie in Gleichung 2.5 dargestellt, die Summe der Verlustwerte über alle n Tripel.

Es stellte sich heraus, dass die Auswahl der zu verwendenden Tripel für das Erzielen einer guten Leistung sehr wichtig ist. Zu einfache Tripel sind nicht effektiv bei der Aktualisierung des Modells. Stattdessen werden sog. „harte Tripel“ gesucht, die Änderungen am Modell fördern. Die Generierung harter Tripel gelingt, wenn zunächst eine große Menge von Tripeln zufällig generiert wird und dann eine gezielte Auswahl von Tripeln mit hohen Verlustwerten aus dieser Menge getroffen wird [33]. Dieser Ansatz wird auch als „Online Triplet Mining“ bezeichnet und in dieser Arbeit verwendet. Dazu werden initial zufällig 200 Tripel aus dem Trainingsdatensatz generiert, von denen die 32 Tripel mit den höchsten Verlustwerten (mit dem aktuellen, bereits teilweise trainierten Modell berechnet) zum Trainieren des Modells ausgewählt werden. Für die Optimierung wird Adam verwendet. Die verwendete Batch-Größe beträgt 32 und die Lernrate 0.000006.

4.3.3 Ergebnisse der Anwendung eines SNNs auf Anomaliekarten

Das SNN wurde so lange trainiert bis nach 30.000 Epochen keine Verbesserung der Gütemaße (AUC und AP) des Testdatensatzes mehr zu beobachten war. Für die Evaluierung des Modells werden die 32 x 32-Anomaliekarten des Test-Datensatzes tupelweise in das Netz gegeben, um die Distanz der Merkmalsvektoren zu berechnen. Es wird jede Anomaliekarte mit allen anderen verglichen. Für die spaltenweise normalisierten Anomaliekarten ergibt sich ein AUC-Wert von 0,889 und ein AP-Wert von 0,260. Das Netzwerk schneidet also etwas besser ab, als das zuvor beschriebene iterative Verfahren.

Bei Anwendung des Siamesischen Netzes auf die Anomaliekarten, die gemäß der z -Standardisierung über alle Einträge der Karte normalisiert wurden (Gleichung 4.1) verbessert sich die Güte des Verfahrens deutlich. Es wird ein AUC-Wert von 0,922 und ein AP-Wert von 0,343 erreicht.

Im Gegensatz zum iterativen Verschiebungsverfahren, kann das SNN spaltenweise Helligkeitsunterschiede in den Anomaliekarten gut verarbeiten. SNNs sind darauf spezialisiert, wichtige Unterscheidungsmerkmale aus den Eingabedaten automatisch zu erfassen, während sie gleichzeitig unwichtige oder störende Informationen herausfiltern. Die spaltenweise Normalisierung von Anomaliekarten verzerrt Informationen über relative Aktivierungsunterschiede innerhalb von Zeilen. Daher kommt das Siamesische Netzwerk besser mit den unverfälschten z -standardisierten Anomaliekarten zurecht, indem es nur die für die Wiedererkennung relevanten Informationen aus den Karten extrahiert und störende Informationen wie z. B. die Aktivierungen ganzer Spalten automatisch herausfiltert. In kurzen Testreihen wurden verschiedene Änderungen an den Anomaliekarten, insbesondere bezüglich der Größe des Originalbildes, der Größe der Teilbilder und ihrer Überlappung, vorgenommen.

Diese konnten jedoch nicht zu einer wesentlichen Verbesserung des Unterscheidungsvermögens führen.

Kapitel 5

Ansatz mit Merkmalskarten

Die im vorherigen Abschnitt vorgestellten Anomaliekarten kodieren die räumliche Anordnung sowie die Ausprägung von Anomalien im Pressspanmuster, nicht jedoch die Strukturen im Pressspanmuster. Daher werden in diesem Kapitel Merkmalskarten erstellt, welche auf charakteristische Strukturen im Pressspanmuster beruhen. Die Merkmalskarten sollen, wie im vorherigen Kapitel für Anomaliekarten beschrieben, in ein Siamesisches Neuronales Netz gespeist werden, um eine Ähnlichkeitsmetrik für dessen Re-Identifikation zu erhalten.

5.1 Vorgehen zur Erzeugung der Merkmalskarten

Um die Merkmalskarten zu erstellen, werden die Bilder der Palettenklötze auf eine einheitliche Größe von 480×740 Pixel gebracht. Ähnlich wie bei der Berechnung der Anomaliekarten werden die Bilder in Teilbilder unterteilt. Die besten Ergebnisse werden mit $48 \times 48 \times 3$ großen Teilbildern erreicht, die sich sowohl vertikal als auch horizontal zu 60% überlappen. Mit diesem Verfahren wird jedes Palettenbild in $24 \cdot 38 = 912$ Teilbilder unterteilt. Analog zur Berechnung der Anomaliekarten soll jedes der $48 \times 48 \times 3$ großen Teilbilder durch nur einen Wert repräsentiert werden, welcher die Struktur des Pressspanmusters in diesem kleinen Ausschnitt widerspiegelt. Am einfachsten ist es, den Mittelwert der $48 \cdot 48 \cdot 3 = 6.912$ Pixel zu berechnen. Die Merkmalskarte wird dann erstellt, indem die 912 berechneten Mittelwerte entsprechend der ursprünglichen zweidimensionalen Anordnung der Teilbilder zu einer 24×38 Karte zusammengesetzt werden. Anschließend werden die Merkmalskarten mittels z -Standardisierung (siehe Gleichung 4.1) und anschließender Normalisierung, wie im Abschnitt 4.1.3 bzgl. der Anomaliekarten beschrieben, auf Werte zwischen 0 und 1 gebracht.

Ein anderer Ansatz zur Erstellung von Merkmalskarten, der in dieser Arbeit verfolgt wird, berechnet für jedes der 912 Teilbilder die mittlere quadratische Abweichung in Bezug

auf das dreidimensionale 48×48 RGB-Bild x^{rgb} und dessen Umwandlung in ein eindimensionales Graustufenbild¹ x^g , siehe Gleichung 5.1.

$$x(x^{rgb}, x^g) = \frac{1}{48 \cdot 48 \cdot 3} \sum_{w=1}^{48} \sum_{h=1}^{48} \sum_{c=1}^3 (x_{whc}^{rgb} - x_{wh}^g)^2, \quad (5.1)$$

wobei w , h und c die Bildbreite, -höhe und den Farbkanal des Teilbildes bezeichnen.

Die resultierenden Werte x werden anschließend, genau wie beim Mittelwert-Ansatz, zu Merkmalskarten zusammengesetzt und normiert.

5.2 Architektur des SNNs zur Re-Identifikation von Merkmalskarten

Die auf diese Weise gewonnenen Merkmalskarten werden in ein Siamesisches Netz eingespeist, welches dem im Abschnitt 4.3.1 beschriebenen Netz für Anomaliekarten sehr ähnlich ist. In Abbildung 5.1 ist eines von drei parallelen, identischen CNNs dargestellt. Im Unterschied zum SNN für Anomaliekarten werden Filter der Größe 7×7 (vorher: 3×3) eingesetzt und in allen Convolutional-Schichten (mit Ausnahme der 1×1 -Convolutions) wird Same-Padding verwendet. Unter Verwendung von Triplet Loss als Verlustfunktion und „Online

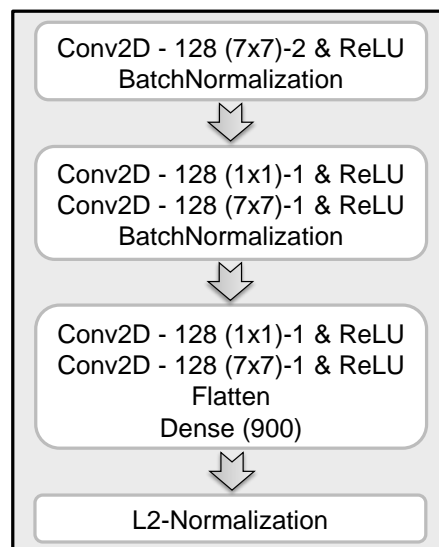


Abbildung 5.1: Architektur des SNNs für die Re-Identifikation von Merkmalskarten. Dargestellt ist eines von drei parallelen, identischen CNNs. Die Zahlenangaben an den Convolutional-Schichten haben folgende Bedeutung: Anzahl Filter (Filterhöhe x Filterbreite)-Stride & Aktivierungsfunktion. Alle Convolutional-Schichten verwenden Same-Padding.

Triplet Mining“ wird das Modell mit allen Einstellungen trainiert, die im Abschnitt 4.3.2 für Anomaliekarten beschrieben sind.

¹https://www.tensorflow.org/api_docs/python/tf/image/rgb_to_grayscale

5.3 Ergebnisse des Ansatzes mit Merkmalskarten

Die benötigte Laufzeit für die Berechnung der Merkmalskarten beträgt etwa eine Sekunde für den Farbdifferenz-Ansatz bzw. 0,3 Sekunden für den Mittelwert-Ansatz. In den Abbildungen 5.2 und 5.3 sind die Merkmalskarten beider Ansätze als Heatmap visualisiert.

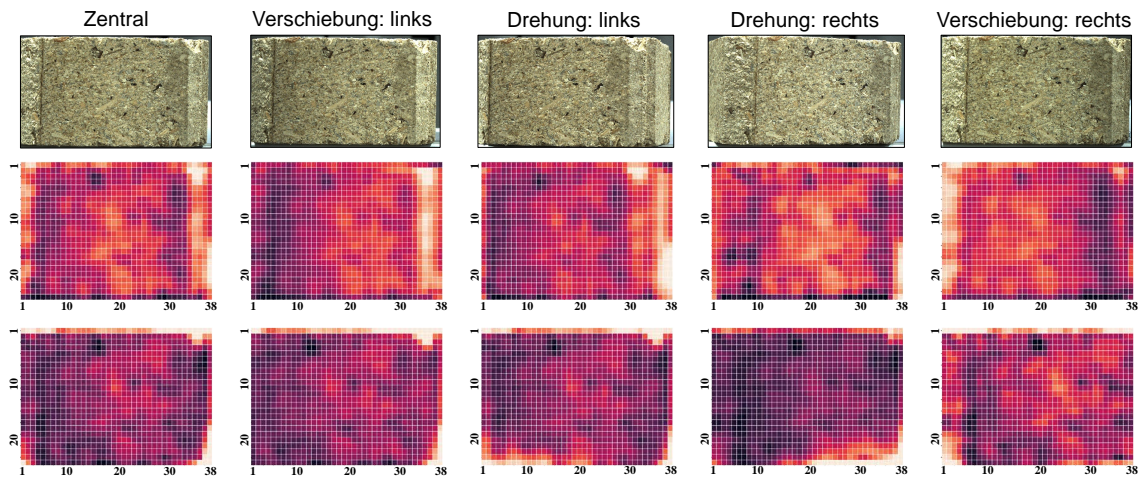


Abbildung 5.2: Merkmalskarten (Mittelwert-Ansatz). Die obere Reihe zeigt Bilder eines Palettenklotzes (ID 6), die aus fünf verschiedenen Kameraperspektiven bei künstlicher Beleuchtung aufgenommen wurden. In der mittleren Reihe sind die entsprechenden Merkmalskarten zu sehen. Die untere Reihe zeigt die Merkmalskarten von Palettenbildern, die mit natürlicher Beleuchtung aufgenommen wurden. Die Merkmalskarten werden als Heatmap visualisiert.

Um einen direkten Vergleich zu ermöglichen, sind jeweils alle 10 Merkmalskarten des Palettenklotzes mit der ID 6 dargestellt. Zusätzlich sind in Abbildung 5.2 die entsprechenden Palettenklötze abgebildet. Man sieht, dass die Merkmalskarten des Mittelwert-Ansatzes stärkere Aktivierungsunterschiede aufgrund von Kanten und Hintergrundinformationen der Palettenbilder aufweisen. Dies ist insbesondere am hellen Hintergrund im unteren Bereich der Palettenbilder zu erkennen, wodurch die mittelwertsbasierten Merkmalskarten an den entsprechenden Stellen helle Bereiche aufweisen. Diese sind in den farbdifferenzbasierten Merkmalskarten nicht sichtbar. Ebenso sind hier die Kanten des Palettenklotzes nicht ganz so stark hervorgehoben. Dennoch lassen sich trotz der geringen Auflösung der Merkmalskarten in beiden Darstellungen charakteristische Strukturen des Palettenklotzes erkennen. Am Beispiel des in den Abbildungen dargestellten Palettenklotzes mit der ID 6 hebt insbesondere die dunkle Struktur im oberen Bereich des Pressspannmusters deutlich eine fehlende Aktivierung in der Heatmap hervor, die fast wie ein Loch aussieht.

Das Training des Siamesischen Netzes auf den Merkmalskarten des Test-Datensatzes erfolgt für beide Ansätze separat. Dazu werden die Modelle jeweils über 50.000 Epochen trainiert, bis keine weitere Verbesserung der Gütemaße (AUC und AP) des Testdatensatzes zu beobachten ist.

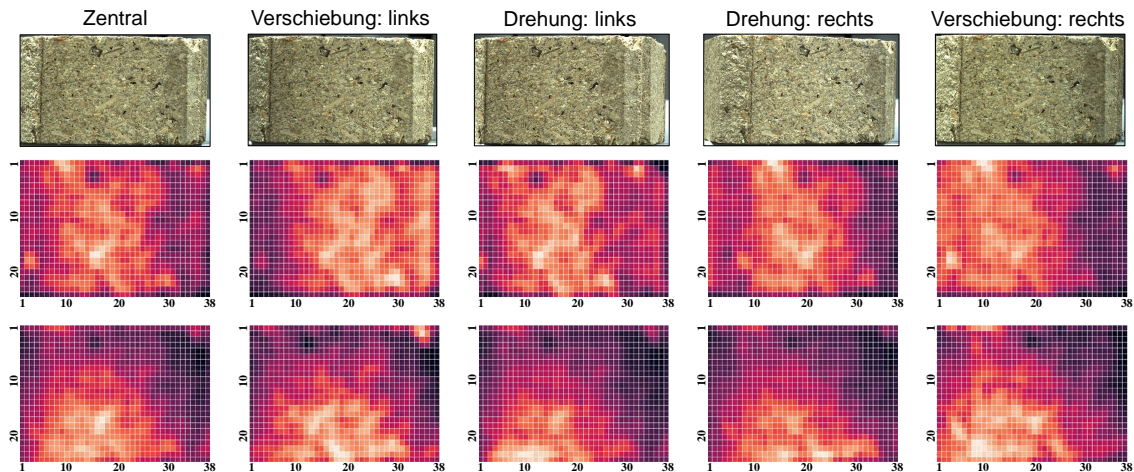


Abbildung 5.3: Merkmalskarten (Farbdifferenz-Ansatz). In der oberen Reihe sind Bilder eines Palettenklötzes (ID 6) aus fünf verschiedenen Kameraperspektiven bei künstlicher Beleuchtung dargestellt. Darunter sind die entsprechenden 10 Merkmalskarten aufgenommen unter künstlicher Beleuchtung (mittig) und mit natürlicher Beleuchtung (unten) abgebildet. Die Merkmalskarten werden als Heatmap visualisiert.

Für die anschließende Evaluierung der Modelle werden die Merkmalskarten des Test-Datensatzes tupelweise in das Netz gegeben, um die Distanz der Merkmalsvektoren zu berechnen. Es wird jede Merkmalskarte mit allen anderen verglichen. Für die mittelwertsbasierten Merkmalskarten ergibt sich ein AUC-Wert von 0,992 und ein AP-Wert von 0,856. Das SNN schneidet also auf den Merkmalskarten deutlich besser ab als auf den Anomaliekarten. Die Güte des Verfahrens lässt sich durch die farbdifferenzbasierten Merkmalskarten sogar nochmals steigern. Es wird ein AUC-Wert von 0,996 und ein AP-Wert von 0,925 erreicht. Zur Veranschaulichung sind in den Abbildungen 5.4 und 5.5 die ROC (a) und Precision-Recall (b)-Kurven beider Ansätze dargestellt. Die Güte des Modells auf den farbdifferenzbasierten Merkmalskarten ist demnach etwas besser als die Güte des ORB-Verfahrens.

Wie eingangs erwähnt, wurde die optimale Überlappung und Größe der Teilbilder in kurzen Vorexperimenten bestimmt. Dazu wurde die Größe des ursprünglichen Palettenbildes annähernd konstant gehalten (etwa 480 x 740 Pixel). Somit nimmt mit zunehmender Überlappung der Teilbilder automatisch deren Größe zu, um Merkmalskarten einheitlicher Größe (24 x 28 Pixel) zu erhalten. In Tabelle 5.1 ist dieser Zusammenhang zu sehen. Die besten Gütemaße werden mit 48 x 48 Teilbildern und einer Überlappung von 60% erreicht. Mit zunehmender und abnehmender Überlappung werden etwas schlechtere Ergebnisse erzielt.

Diese Ergebnisse zeigen, dass Merkmalskarten für Siamesische Netze aussagekräftigere Informationen enthalten als Anomaliekarten. Die alleinige Information über die räumliche Anordnung von Anomalien sowie über deren Ausprägung, so wie sie in den im Abschnitt 4

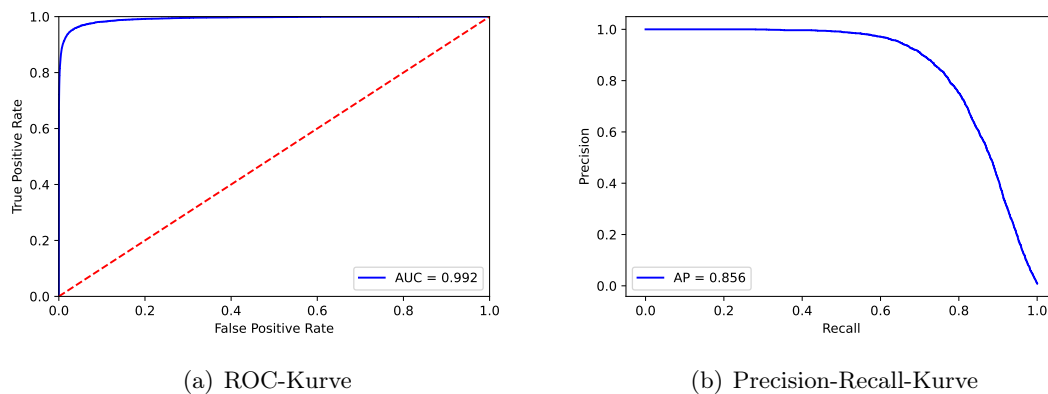


Abbildung 5.4: ROC-Kurve und Precision-Recall-Kurve bei Anwendung des Siamesischen Netzes auf die Merkmalskarten (Mittelwert) des Test-Datensatzes.

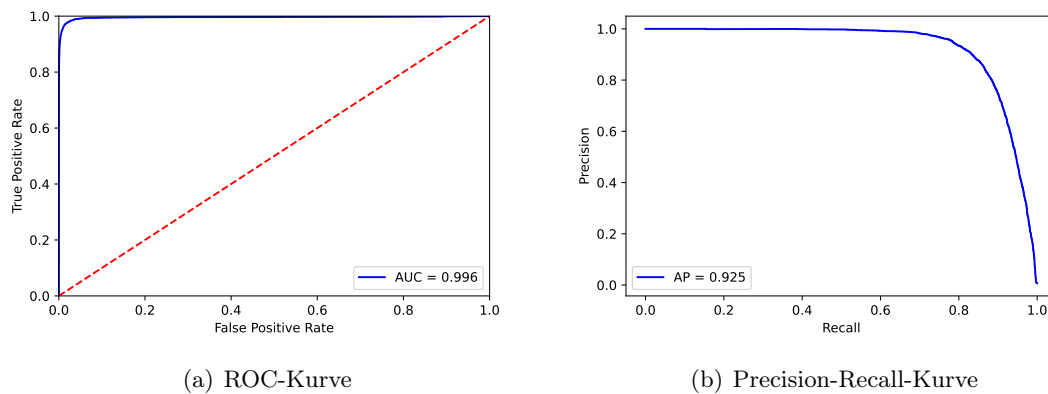


Abbildung 5.5: ROC-Kurve und Precision-Recall-Kurve bei Anwendung des Siamesischen Netzes auf die Merkmalskarten (Farbdifferenz-Ansatz) des Test-Datensatzes.

beschriebenen Anomaliekarten enthalten ist, reicht nicht aus, um Palettenklötze zuverlässig wiederzuerkennen. Merkmalskarten enthalten zusätzlich Informationen über die Strukturen des Pressspannmusters, was für die Re-Identifikation von entscheidender Bedeutung zu sein scheint. Zusätzlich ist bemerkenswert, dass solch eine kleine Repräsentation des Palettenklotzes ausreicht, um diesen wiederzuerkennen. Insbesondere diese Feststellung wird in dem Verfahren wieder aufgegriffen, welches im folgenden Kapitel 6 vorgestellt wird.

Tabelle 5.1: Gütemaße als Resultat der Anwendung des Siamesischen Netzes auf die farbdifferenzbasierten Merkmalskarten in Abhängigkeit von der Größe und Überlappung der Teilbilder bei einer Bildgröße des Palettenklotzes von etwa 480 x 740 Pixeln. Bei zunehmender Überlappung ergeben sich automatisch größere Teilbilder, um Merkmalskarten einheitlicher Größe (24 x 38 Pixel) zu erhalten.

Überlappung der Teilbilder	Größe der Teilbilder	AUC	AP
40%	32 x 32	0,995	0,903
50%	38 x 38	0,995	0,915
60%	48 x 48	0,996	0,925
70%	64 x 64	0,996	0,922
80%	92 x 92	0,993	0,877

Kapitel 6

Ansatz mit Latent-Space-Karten

Im vorhergehenden Kapitel wurde gezeigt, dass Merkmalskarten, die sehr komprimierte Informationen über die Strukturen des Pressspanmusters von Palettenklötzen enthalten, eine gute Repräsentation der Palettenklötze für deren Re-Identifikation mit einem Siamesischen Neuronalen Netz darstellen.

Eine Möglichkeit, um auf andere Weise eine ebenso kleine Repräsentation der Palettenklötze zu erhalten, ist die Verwendung der inneren Schicht des Autoencoders, auch Kernschicht oder „Latent Space“ genannt. Im Abschnitt 2.4 ist die Funktionsweise des Autoencoders bereits beschrieben worden. Die Kernschicht extrahiert die wesentlichen Strukturen des Palettenbildes, um eine gute Rekonstruktion des Originalbildes zu ermöglichen. Daher zielt dieser Ansatz darauf ab, zu überprüfen, ob die innere Kernschicht eines Autoencoders ebenfalls eine gute Repräsentation für die Re-Identifikation der Palettenklötze liefert. Zu diesem Zweck wird der Autoencoder so konstruiert, dass seine innere Kernschicht der Größe von 22 x 36 Pixeln entspricht. Da diese die Form einer zweidimensionalen Karte hat, wird im Folgenden von Latent-Space-Karten gesprochen.

Die Latent-Space-Karten werden, wie es zuvor für Anomaliekarten und Merkmalskarten beschrieben wurde, in ein Siamesisches Neuronales Netz gespeist, um eine Ähnlichkeitsmetrik für deren Re-Identifikation zu erhalten.

6.1 Vorgehen zur Erzeugung der Latent-Space-Karten

Die Bilder der Palettenklötze werden zunächst auf eine einheitliche Größe von 704 x 1152 Pixeln gebracht. Die Farbkanäle der Teil-Bilder (R, G, B) werden zwischen 0 und 1 normiert. Anschließend werden die Bilder in den zuvor auf dem Trainingsdatensatz trainierten Autoencoder gegeben. Die Architektur des Autoencoders, die in kurzen Vorexperimenten zu den besten Ergebnissen geführt hat, ist in Tabelle 6.1 dargestellt. Über mehrere Convolutional-Schichten mit einer Schrittweite (engl. strides) von 2 wird das Originalbild mit einer Größe von 704 x 1152 x 3 Pixeln in eine 22 x 36 x 1 Latent-Space-Karte

überführt. Der Decoder ist symmetrisch zum Encoder aufgebaut mit Conv2DTranspose-Schichten anstelle der Conv2D-Schichten. Die letzte Convolutional-Schicht mit 3 Filtern und einer Schrittweite von 1 sorgt dafür, dass die Eingabegröße wieder erreicht wird. In dieser Schicht wird die sigmoide Aktivierungsfunktion verwendet, um die Ausgabewerte zwischen 0 und 1 zu normieren.

Zum Trainieren des Autoencoders wird eine Batch-Größe von 32 und eine Lernrate von 0,001 verwendet. Anschließend wird nur der Encoder des Autoencoders auf den Test-Datensatz angewandt, wodurch die sogenannten Latent-Space-Karten mit einer Größe von 22 x 36 Pixeln entstehen. Diese werden mittels z -Standardisierung (siehe Gleichung 4.1) und anschließender Normalisierung, wie im Abschnitt 4.1.3 bzgl. der Anomaliekarten beschrieben, auf Werte zwischen 0 und 1 gebracht.

6.2 Architektur des SNNs zur Re-Identifikation von Latent-Space-Karten

Das verwendete SNN ist dem im Abschnitt 5.2 beschriebenen Netz für Merkmalskarten entnommen. Aufgrund der leicht geringeren Größe der Latent-Space-Karten (im Vergleich zu den Merkmalskarten) wurden zwei Änderungen durchgeführt: (1) im ersten Conv2D-Layer wird ein Stride von 1 (anstelle von 2) verwendet und (2) in allen Convolutional-Schichten wird kein Padding (anstelle von Same-Padding) durchgeführt.

Unter Verwendung von Triplet Loss als Verlustfunktion und „Online Triplet Mining“ wird das Modell mit allen Einstellungen trainiert, die im Abschnitt 4.3.2 für Anomaliekarten beschrieben sind.

6.3 Ergebnisse des Ansatzes mit Latent-Space-Karten

Der Autoencoder wird über 50 Epochen trainiert. Nach Abschluss des Trainings beträgt der Verlustwert des Testdatensatzes 0,0131. In Abbildung 6.1 sind beispielhaft alle 10 Originalbilder des Palettenklotzes mit der ID 6 mit den zugehörigen Rekonstruktionen dargestellt. Die wichtigsten Strukturen des Pressspanmusters, die Orientierung des Palettenklotzes, sowie Helligkeitsunterschiede aufgrund der Beleuchtung (künstlich bzw. natürlich) sind in den Rekonstruktionen zu erkennen.

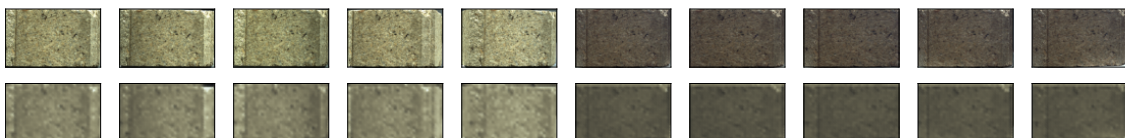


Abbildung 6.1: Rekonstruktion von Bildern des Palettenklotzes mit der ID 6. In der oberen Reihe sind die Originalbilder und in der unteren Reihe die zugehörigen Rekonstruktionen abgebildet.

Die Berechnung der Latent-Space-Karten erfordert durchschnittlich 0,027 Sekunden Laufzeit pro Karte. In Abbildung 6.2 sind alle 10 Latent-Space-Karten des Palettenklotzes mit der ID 6 als Heatmap visualisiert. Zum direkten Vergleich mit den originalen Palettenbildern, sind diese in der ersten Reihe zusätzlich abgebildet. Man kann auffällige Strukturen im Pressspanmuster, sowie auch die Kanten des Palettenklotzes in den Karten wiederfinden, wobei helle Strukturen im Originalbild in der Latent-Space-Karte dunkel dargestellt sind und umgekehrt. Insgesamt ähneln die Latent-Space-Karten den Merkmalskarten des Mittelwert-Ansatzes (jeweils mit umgekehrter Helligkeit), vergleiche mit Abbildung 5.2.

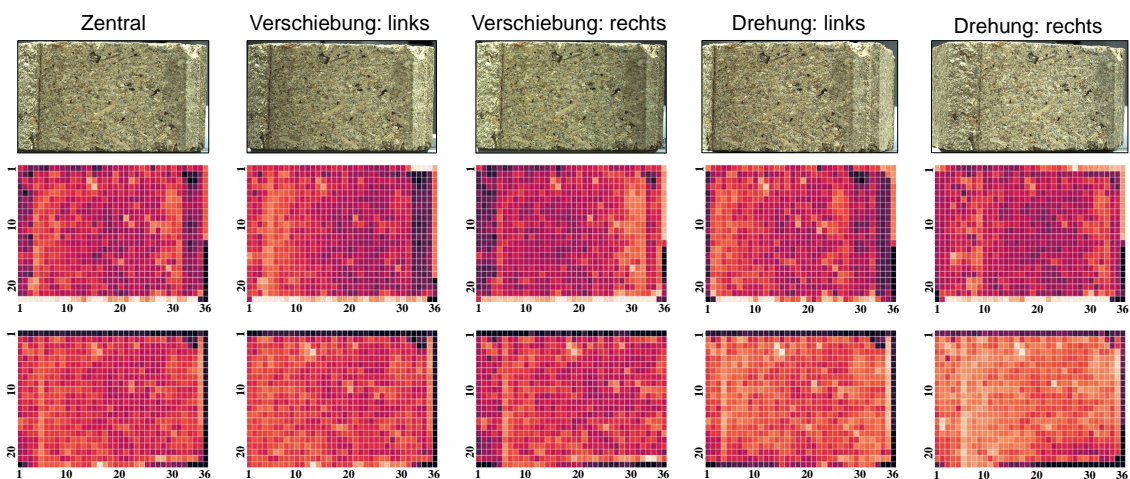


Abbildung 6.2: Latent-Space-Karten. Die obere Reihe zeigt Bilder eines Palettenklotzes (ID 6), die aus fünf verschiedenen Kameraperspektiven bei künstlicher Beleuchtung aufgenommen wurden. In der mittleren Reihe sind die dazugehörigen Latent-Space-Karten zu sehen. Die untere Reihe zeigt die entsprechenden Karten von Palettenbildern, die mit natürlicher Beleuchtung aufgenommen wurden. Die Latent-Space-Karten sind als Heatmap visualisiert.

Das SNN wird über 60.000 Epochen auf dem Test-Datensatz trainiert, bis keine weitere Verbesserung der Gütemaße (AUC und AP) des Testdatensatzes zu beobachten ist. Die Evaluierung des trainierten Modells auf dem Test-Datensatz ergibt einen AUC-Wert von 0,991 und einen AP-Wert von 0,845. Die Güte des Modells ist demnach vergleichbar mit der Güte des Mittelwert-Ansatzes (AUC = 0,992 und AP = 0,856), erreicht jedoch nicht die Güte des Farbdifferenz-Ansatzes (AUC = 0,996 und AP = 0,925).

Die Idee, einen Autoencoder zur Merkmalsextraktion und ein Siamesisches Neuronales Netz als Klassifikator hintereinanderschalten wurde bereits erfolgreich für die Online-Verifizierung von Unterschriften eingesetzt [1].

Tabelle 6.1: Autoencoder für den Latent-Space-Ansatz. Der Decoder ist symmetrisch zum Encoder aufgebaut mit Conv2DTranspose-Schichten anstelle der Conv2D-Schichten. Die Zahlenangaben an den Convolutional-Schichten haben folgende Bedeutung: Anzahl Filter (Filterhöhe x Filterbreite)-Stride & Aktivierungsfunktion. Alle Convolutional-Schichten verwenden Same-Padding.

Encoder	
Schicht	Ausgabe-Dimension
Batch Normalization	(704, 1152, 3)
Conv2D 32 (7 x 7)-2 & ReLU	(352, 576, 32)
Batch Normalization	(352, 576, 32)
Conv2D 16 (5 x 5)-2 & ReLU	(176, 288, 16)
Batch Normalization	(176, 288, 16)
Conv2D 8 (3 x 3)-2 & ReLU	(88, 144, 8)
Batch Normalization	(88, 144, 8)
Conv2D 4 (3 x 3)-2 & ReLU	(44, 72, 4)
Batch Normalization	(44, 72, 4)
Conv2D 2 (3 x 3)-2 & ReLU	(22, 36, 2)
Batch Normalization	(22, 36, 2)
Conv2D 1 (2 x 2)-1 & ReLU	(22, 36, 1)
Decoder	
Schicht	Ausgabe-Dimension
Batch Normalization	(22, 36, 1)
Conv2DTranspose 1 (2 x 2)-1 & ReLU	(22, 36, 1)
Batch Normalization	(22, 36, 1)
Conv2DTranspose 2 (3 x 3)-2 & ReLU	(44, 72, 2)
Batch Normalization	(44, 72, 2)
Conv2DTranspose 4 (3 x 3)-2 & ReLU	(88, 144, 4)
Batch Normalization	(88, 144, 4)
Conv2DTranspose 8 (3 x 3)-2 & ReLU	(176, 288, 8)
Batch Normalization	(176, 288, 8)
Conv2DTranspose 16 (5 x 5)-2 & ReLU	(352, 576, 16)
Batch Normalization	(352, 576, 16)
Conv2DTranspose 32 (7 x 7)-2 & ReLU	(704, 1152, 16)
Batch Normalization	(704, 1152, 16)
Conv2D 3 (3 x 3)-1 & sigmoid	(704, 1152, 3)

Kapitel 7

Kombination von Modellen

In den Kapiteln 5 und 6 wurden auf verschiedene Weisen kleine Repräsentationen eines Palettenklotzbildes erzeugt, um diese mittels eines Siamesischen Neuronalen Netzes zu vergleichen. Das SNN liefert für die zu vergleichenden Palettenklötze jeweils ein „Embedding“ als Ausgabe. Eine Ähnlichkeitsmetrik ergibt sich dann durch die Berechnung der mittleren quadratischen Abweichung der zu vergleichenden Embeddings.

Alle zuvor evaluierten Ansätze, die auf der Anwendung eines SNNs basieren, liefern gute Ergebnisse. Allerdings sind leichte Performance-Unterschiede zu beobachten, in denen die erlernten Modelle (in Grenzfällen) unterschiedliche Entscheidungen vorgeben. Ein möglicher Weg zur Behandlung solcher Entscheidungsunsicherheiten einzelner Modelle ist die Kombination der Ergebnisse verschiedener Modelle und die Bestimmung einer gemeinsamen Ausgabe.

Um dieses Potenzial der Modelle zu untersuchen, wird zum Abschluss der Betrachtung einzelner Modellvarianten der einfachste Weg zur Kombination jener Modelle verwendet. Dieser Weg besteht darin, den Mittelwert der normierten Ähnlichkeitswerte zu bilden. Dies kann insbesondere bei Modellkombinationen, in denen ein Modell eine klare Entscheidung vorgibt und eines oder mehrere Modelle Unsicherheiten nahe des Schwellenwertes aufweisen, eine Entscheidung hin zu einem gemeinsamen Wert oberhalb des Schwellenwertes bewirken.

7.1 Vorgehen zur Kombination der Modelle

Da ein Siamesisches Netz aus identischen Subnetzen aufgebaut ist, ist garantiert, dass zwei identische Eingaben auf identische Merkmalsvektoren abgebildet werden, deren Differenz 0 ist (siehe auch Abschnitt 2.5, Konsistenz der Vorhersagen). Somit gibt das Netz für identische Eingaben einen Wert von 0 aus. Unähnliche Eingaben führen entsprechend der Berechnung der Ähnlichkeitswerte (negative mittlere quadratische Abweichung der „Embeddings“) zu negativen Werten. Um empirisch abzuschätzen, in welchem Intervall die Werte

liegen, werden für jedes der drei Modelle alle 5020 Palettenklotzbilder (Trainings- und Testdatensatz) bzw. deren Repräsentationen als Merkmalskarten mit allen anderen verglichen. Daraus ergeben sich $\frac{5020 \cdot 5019}{2} = 12.597.690$ Ähnlichkeitswerte. Diese liegen im Intervall von $[-3.261, -0.045]$ für den Farbdifferenz-Ansatz, im Intervall von $[-3.134, -0.008]$ für den Mittelwert-Ansatz und im Intervall von $[-3.123, -0.009]$ für den Latent-Space-Ansatz.

Um eine normierte Ähnlichkeitsmetrik zu erhalten, werden die Ähnlichkeitswerte durch das Minimum aller für das jeweilige Modell gemessener Ähnlichkeitswerte geteilt und mit -1 multipliziert, um die Reihenfolge beizubehalten. Die Kombination zweier Modelle erfolgt nun durch die Berechnung des Mittelwerts der zuvor normierten Ähnlichkeitswerte beider Modelle.

7.2 Ergebnisse der Kombination von Modellen

Die Merkmalskarten-Ansätze (Mittelwert und Farbdifferenz) eignen sich besonders zur Kombination, da beide Ansätze auf eine Aufteilung des Originalbildes in überlappende Teilbilder basieren. Dieser Schritt kann daher einmalig für beide Ansätze durchgeführt werden. Die Berechnung des Mittelwerts der Teilbilder kostet kaum zusätzliche Rechenzeit. Die Kombination des Mittelwert- und des Farbdifferenzansatzes führt im Vergleich zur Einzelleistung beider Modelle zu einer deutlichen Verbesserung der Güte des Modells. Es wird ein AUC-Wert von 0,997 und ein AP-Wert von 0,981 erreicht. Auch die Kombination des Farbdifferenzansatzes mit dem Latent-Space-Ansatz übertrifft die Einzelleistung der Modelle. Insbesondere der AP-Wert nimmt deutlich zu. Hier wird ein AUC-Wert von 0,998 und ein AP-Wert von 0,986 erreicht. Die Kombination des Mittelwert-Ansatzes mit dem Latent-Space-Ansatz ergibt einen AUC-Wert von 0,997 und einen AP-Wert von 0,925. Alle drei Ansätze ergeben kombiniert einen AUC-Wert von 0,998 und einen AP-Wert von 0,987. In Tabelle 7.1 sind zur Veranschaulichung nochmals die Gütemaße der einzelnen Modelle, sowie der Zugewinn durch die Kombination der Modelle zu sehen.

Tabelle 7.1: Gütemaße als Resultat der Kombination von Modellen

Modell	AUC	AP
Mittelwert	0,992	0,856
Farbdifferenz	0,996	0,925
Latent-Space	0,991	0,845
Farbdifferenz & Mittelwert	0,997	0,981
Farbdifferenz & Latent-Space	0,998	0,986
Mittelwert & Latent-Space	0,997	0,925
Mittelwert & Farbdifferenz & Latent-Space	0,998	0,987

Die Zunahme der Gütemaße durch die Kombination von Modellen ist insbesondere dann zu erwarten, wenn die Modelle unterschiedliche Stärken haben und jeweils unterschiedliche Palettenklötze wiedererkennen können. Obwohl die Ansätze teilweise Gemeinsamkeiten aufweisen, führt eine Kombination zweier Modelle in allen Fällen zu einer Zunahme der Güte. Alle Verfahren weisen insbesondere die Gemeinsamkeit auf, dass sie das in überlappende Teilbilder unterteilte Originalbild benutzen, um daraus Merkmalskarten zu berechnen. Während die Merkmalskarten-Ansätze das Bild tatsächlich zerschneiden, übernehmen im Latent-Space-Ansatz die Filterfenster in den Convolutional-Schichten diese Aufgabe. Diese werden schrittweise (mit Überlappung) über das Bild bewegt, um entsprechende Merkmalskarten zu berechnen, die kombiniert letztlich die Latent-Space-Karte ergeben. So lässt sich erklären, warum sich beim Mittelwert- und Latent-Space-Ansatz die jeweiligen Karten so ähnlich sehen, abgesehen von einer Umkehrung der Helligkeitswerte. Der größte Zugewinn wird durch die Kombination des Farbdifferenzansatzes mit dem Latent-Space-Ansatz erreicht. Aber auch die Kombination des Farbdifferenz- mit dem Mittelwert-Ansatz erzielt sehr gute Werte. Die beste Leistung wird durch die Kombination aller drei Ansätze erreicht.

Kapitel 8

Direkter Ansatz: SNN mit Palettenklotzbildern

In den vorangegangenen Abschnitten wurde gezeigt, dass eine sehr kleine Repräsentation der Palettenklotzbilder in Kombination mit einem Siamesischen Neuronalen Netz ausreichend für eine zuverlässige Re-Identifikation von Palettenklötzen ist. Dennoch hat auch der ORB-Algorithmus, der direkt auf die Palettenklotzbilder angewandt wurde, hinreichend gute Ergebnisse geliefert, um die Frage aufzuwerfen, ob eine direkte Anwendung eines SNNs auf Palettenklotzbilder nicht auch zu guten Ergebnissen führt. Um dies zu validieren, wird die Architektur des SNNs so angepasst, dass eine direkte Eingabe von Palettenklotzbildern möglich ist.

8.1 Vorgehen zur direkten Anwendung eines SNNs auf Palettenklotzbilder

Zur Vorverarbeitung werden die Bilder der Palettenklötze auf eine einheitliche Größe von $200 \times 300 \times 3$ Pixel gebracht. Zusätzlich werden die Farbkanäle der RGB-Bilder zwischen 0 und 1 normiert.

Im Vorfeld wurden verschiedene Architekturen von Siamesischen Neuronalen Netzen getestet. Die besten Ergebnisse wurden mit der in Abbildung 8.1 gezeigten Architektur erzielt. Da das Netz aus drei identischen parallelen CNNs mit gekoppelten Gewichten besteht, wird nur eines gezeigt. Die CNNs bestehen aus 3 Blöcken von Convolutional-Schichten mit jeweils 32 Filtern der Größe 7×7 . Alle Convolutional-Schichten verwenden ReLU als Aktivierungsfunktion und kein Padding. Die erste Schicht hat eine Schrittweite (engl. strides) von 2, alle anderen haben eine Schrittweite von 1. Zwischen zwei Standard-Convolutional-Schichten (im zweiten und dritten Block) wird jeweils eine zusätzliche 1×1 -Convolution hinzugefügt. Die ersten beiden Blöcke verwenden MaxPooling und BatchNormalization.

Am Ende des dritten Blocks befinden sich eine Flatten- und eine Dense-Schicht mit einer „embedding size“ von 900.

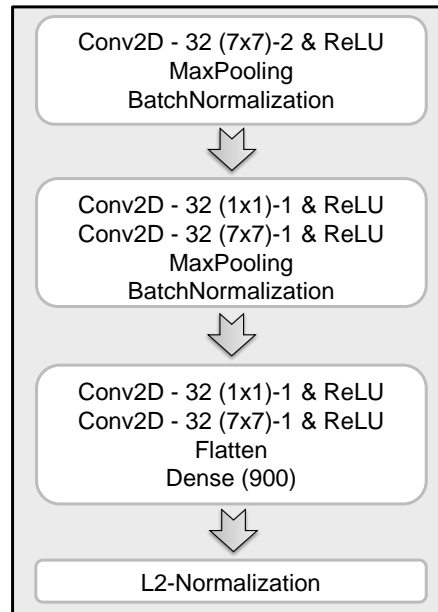


Abbildung 8.1: Architektur des SNNs zur direkten Anwendung eines SNNs auf Palettenklotzbilder. Dargestellt ist eines von drei parallelen, identischen CNNs. Die Zahlenangaben an den Convolutional-Schichten haben folgende Bedeutung: Anzahl Filter (Filterhöhe x Filterbreite)-Stride & Aktivierungsfunktion. Alle Convolutional-Schichten verwenden Same-Padding.

Unter Verwendung von Triplet Loss als Verlustfunktion und „Online Triplet Mining“ wird das Modell auf dem Trainings-Datensatz mit allen Einstellungen trainiert, die im Abschnitt 4.3.2 für Anomaliekarten beschrieben sind.

8.2 Ergebnisse des direkten Ansatzes

Im Vergleich zu den Ansätzen mit vorheriger Komprimierung der Palettenklotzbilder, nimmt das Training des SNNs deutlich mehr Zeit in Anspruch (Faktor 10). Für 100 Iterationen werden 3 Minuten benötigt. Das Training über 50.000 Epochen benötigt somit 25 Stunden (anstelle von 2,5 Stunden für Anomalie-, Merkmals- oder Latent-Space-Karten).

Für die anschließende Evaluierung des Modells werden die Palettenklotzbilder des Test-Datensatzes tupelweise in das Netz gegeben. Dabei wird jedes Bild mit allen anderen verglichen. Nach 50.000 Iterationen werden ein AUC-Wert von 0,979 und ein AP-Wert von 0,704 erreicht.

Somit übertreffen die Ansätze, die Merkmalskarten oder Latent-Space-Karten verwenden, den direkten Ansatz deutlich. Das gilt nicht nur für die Performance (AUC/AP), sondern zusätzlich für den Ressourcenaufwand zum Trainieren des Modells. Eine sinnvolle Komprimierung der Daten kann also die Leistungsfähigkeit des SNNs erhöhen. Natürlich

basiert dieser Ansatz genau wie die vorherigen Ansätze, auf einfachen, „selbst entworfenen“ CNNs. Speziell für die Re-Identifikation entwickelte CNNs, wie z. B. das PCB-Netzwerk [35], das Rutinowski et al. [32] für ein überwachtetes Lernverfahren zur Identifikation von Palettenklötzen verwendet haben, könnten zu weiteren Verbesserungen für alle Ansätze führen.

Kapitel 9

Zusammenfassung und Diskussion

Ziel dieser Arbeit war es einen bildbasierten Algorithmus zur Palettenerkennung unter Verwendung moderner Techniken der künstlichen Intelligenz zu entwickeln. Rutinowski et al. [32] haben bereits gezeigt, dass mittels eines überwachten maschinellen Lernverfahrens eine zuverlässige Identifizierung von Paletten möglich ist. Allerdings erfordert dieses Verfahren eine umfangreiche Trainingsphase für unbekannte Paletten. Ziel dieser Arbeit war es darüber hinaus, Verfahren zu entwickeln, die eine Wiedererkennung von unbekanntem, also vom Modell noch nicht zuvor gesehenen Palettenklötzen zu ermöglichen, so dass das Modell nicht fortlaufend aktualisiert werden muss, um eine zuverlässige Wiedererkennung zu gewährleisten. Dazu wurden verschiedene Ansätze entwickelt und evaluiert. In Tabelle 9.1 ist eine Übersicht aller Verfahren und der erzielten Gütemaße dargestellt.

Tabelle 9.1: Übersicht über alle Verfahren und die erzielten Ergebnisse auf dem Test-Datensatz.

Modell	AUC	AP
ORB	0,977	0,897
Anomaliekarten (iterativ)	0,853	0,026
Anomaliekarten (SNN)	0,922	0,343
Mittelwert	0,992	0,856
Farbdifferenz	0,996	0,925
Latent-Space	0,991	0,845
Farbdifferenz & Mittelwert	0,997	0,981
Farbdifferenz & Latent-Space	0,998	0,986
Mittelwert & Latent-Space	0,997	0,925
Mittelwert & Farbdifferenz & Latent-Space	0,998	0,987
Direkter Ansatz: SNN mit Palettenklotzbildern	0,979	0,704

Im Unterschied zu den anderen hier verwendeten Verfahren, basiert das ORB-Verfahren nicht auf maschinellem Lernen. Es ist ein Ansatz zur Merkmalsextraktion, welcher es er-

möglicht, charakteristische Merkmale in Bildern zu detektieren, siehe auch Kapitel 2.1. Da das Pressspanmuster der Palettenklötze ausschließlich aus charakteristischen Strukturen und Mustern besteht, ist das ORB-Verfahren ein geeignetes Benchmark-Verfahren. Es wird traditionell in Kombination mit einem Klassifikationsverfahren zur Objekterkennung eingesetzt. In dieser Arbeit wurde die Hamming-Distanz zwischen den Deskriptoren (Fingerprints) als Maß für die Ähnlichkeit zweier Pressspanmuster verwendet. Wird das ORB-Verfahren auf dem Test-Datensatz angewandt, so ergeben sich ein AUC-Wert von 0,977 und ein AP-Wert von 0,897.

Alle weiteren in dieser Arbeit untersuchten Ansätze zur Re-Identifikation von Palettenklötzen basieren auf neuronalen Netzen. Zuerst wurde die Idee verfolgt, Palettenklotzbilder anhand von Anomalien im Pressspanmuster zu re-identifizieren. Anomalien beschreiben dabei Rekonstruktionsfehler im Vergleich zum Standardpressspanmuster und werden mit einem Autoencoder für Teilbilder eines Palettenklotzes berechnet. Anschließend werden die Anomaliewerte in einer Karte entsprechend der ursprünglichen räumlichen Anordnung der Teilbilder wieder zusammengesetzt, wodurch sich eine grobe räumliche Repräsentation des Rekonstruktionsfehlers, die sogenannte Anomaliekarte, ergibt.

Obwohl die Anomaliekarten bei visueller Betrachtung vielversprechend aussehen, scheint die räumliche Anordnung, sowie die Ausprägung der Anomalien allein nicht ausreichend für eine zuverlässige Re-Identifikation von Palettenbildern zu sein. Beide untersuchten Verfahren, das iterative Verschiebungsverfahren und der Vergleich der Anomaliekarten mit Hilfe eines SNNs, führten nur zu einer geringen Re-Identifikationsleistung. Trotzdem führte die Verwendung eines Siamesischen Neuronalen Netzes im Vergleich zum iterativen Verfahren zu einer deutlichen Verbesserung der Diskriminationsleistung.

Diese Feststellung führte direkt zum nächsten Ansatz, der Konstruktion von Merkmalskarten, welche im Unterschied zu den Anomaliekarten Informationen über die Strukturen im Pressspanmuster enthalten. Genau wie die Anomaliekarten, werden auch die Merkmalskarten aus überlappenden Teilbildern eines Palettenklotzbildes zusammengesetzt. Die einfachste Möglichkeit zur Generierung der Merkmalskarten ist die Bildung des Mittelwerts über alle Pixel innerhalb der RGB-Teilbilder, so dass für jedes der Teilbilder ein einzelner Wert berechnet wird. Die Merkmalskarte setzt sich dann analog zur Anomaliekarte aus diesen berechneten Werten entsprechend der ursprünglichen räumlichen Anordnung der Teilbilder zusammen. Zusätzlich wurden Merkmalskarten durch die mittlere quadratische Abweichung der RGB-Teilbilder und deren Umwandlung in ein eindimensionales Graustufenbild (nach Gleichung 5.1) berechnet. Auch hier wurden die berechneten Werte anschließend zu einer zweidimensionalen Merkmalskarte zusammengesetzt. Im Unterschied zu den mittelwertsbasierten Merkmalskarten, sind die Kanten der Palettenklötze in den farbdifferenzbasierten Merkmalskarten deutlich weniger stark hervorgehoben. Trotz der geringen Auflösung der Merkmalskarten, lassen sich in beiden Darstellungen Strukturen des Pressspanmusters erkennen. Die auf diese Weise gewonnenen Merkmalskarten wurden

analog zu den Anomaliekarten in ein Siamesisches Neuronales Netz eingespeist. Es wurden AUC-Werte von 0,992 bzw. 0,996 und AP-Werte von 0,856 bzw. 0,925 für den Mittelwert- bzw. für den Farbdifferenz-Ansatz erreicht. Daraus lässt sich schließen, dass Merkmalskarten für Siamesische Netze aussagekräftigere Informationen enthalten als Anomaliekarten. Die zusätzlichen Informationen über die Strukturen im Pressspanmuster scheinen für die Re-Identifikation von entscheidender Bedeutung zu sein. Der Farbdifferenz-Ansatz übertrifft die Güte des ORB-Verfahrens, siehe Tabelle 9.1.

Anhand der Merkmalskarten wurde gezeigt, dass eine sehr kleine Repräsentation des Palettenklotzes für dessen Re-Identifikation ausreichend ist. Da die innere Schicht des Autoencoders, auch Latent-Space genannt, ebenso eine kleine Repräsentation des Eingangsbildes bereitstellt, welche die wesentlichen Informationen zur Rekonstruktion des Originalbildes enthält, wurden im nächsten Ansatz sogenannte Latent-Space-Karten erstellt. Diese wurden wie zuvor die Anomaliekarten und die Merkmalskarten in ein SNN gespeist, um eine Ähnlichkeitsmetrik zu erhalten. Die Evaluierung des trainierten Modells auf dem Test-Datensatz ergab einen AUC-Wert von 0,991 und einen AP-Wert von 0,845. Die Güte des Modells ist demnach vergleichbar mit dem Mittelwert-Ansatz.

Da die oben beschriebenen Ansätze mit Merkmalskarten und Latent-Space-Karten zu Modellen geführt haben, die eine ähnlich gute Diskriminationsleistung aufweisen, kann eine Kombination von Modellen zu einer zusätzlichen Verbesserung führen. Eine sehr einfache Möglichkeit zur Kombination besteht darin, den Mittelwert der normierten Ähnlichkeitswerte zu bilden, wie im Abschnitt 7.1 erläutert. Die Kombination zweier Modelle übertraf in allen Fällen die Einzelleistung der Modelle. Der jeweilige Zugewinn durch die Kombination kann in Tabelle 9.1 abgelesen werden. Die erreichten Gütemaße übertreffen den ORB-Ansatz deutlich und erreichen sogar die Leistung, die Rutinowski et al. [32] mittels eines überwachten maschinellen Lernverfahrens und dem Part-based Convolutional Baseline (PCB)-Netzwerk mit ResNet50 als grundlegendes Netzwerk, erreicht haben. Dieses Netzwerk wurde gezielt für die Aufgabe der Re-Identifikation von Personen entwickelt [35]. Letztendlich blieb zu überprüfen, ob eine direkte Anwendung eines SNNs auf Palettenklotzbilder ebenfalls zu einer guten Diskriminationsleistung führt. Die im Vergleich zu den Ansätzen mit vorheriger Komprimierung um das 10-fache erhöhte Zeit zum Trainieren des Modells, stellt jedoch für die Parameteroptimierung eine größere Herausforderung dar. Trotzdem wurden viele Modellkonfigurationen getestet, die letztendlich aber alle nicht kompetitiv zu den Ansätzen mit vorheriger Komprimierung (Merkmalskarten und Latent-Space-Karten) waren.

Die Ergebnisse der Arbeit zeigen, dass es grundsätzlich möglich ist, mittels unüberwachter Detektionsverfahren dem System unbekannte Palettenklötze zuverlässig wiederzuerkennen. Es ist insbesondere bemerkenswert, dass diese Erkennungsleistung bereits mit relativ einfachen neuronalen Architekturen jene von überwachten Verfahren erreichen kann.

Ausblick

Alle hier entwickelten Verfahren, die auf neuronalen Netzen beruhen, basieren auf einfachen CNNs. Im Rahmen der zeitlichen Grenzen und verfügbaren Ressourcen der Arbeit wurden lediglich die Parameter in einem iterativ-heuristischen Ansatz optimiert. Es ist davon auszugehen, dass mittels einer systematisch durchgeführten Hyperparameteroptimierung noch bessere Ergebnisse erzielt werden können. Das würde jedoch über die Zielsetzung der Arbeit hinausgehen, welche die Anwendbarkeit von verschiedenen Verfahren im Fokus hatte.

Darüber hinaus existieren neuronale Netze, die gezielt für die Re-Identifikation entwickelt wurden. So haben Rutinowski et al. [32] das Part-based Convolutional Baseline (PCB)-Netzwerk, welches ursprünglich für die Re-Identifikation von Personen entwickelt wurde [35], erfolgreich in einem überwachten Lernverfahren für die Re-Identifikation von Palettenklotzbildern eingesetzt. Eine Integration des PCB-Netzwerkes in ein Siamesisches Neuronales Netz führt zu einem unüberwachten Detektionssystem. Es wäre interessant zu überprüfen, ob solch ein Netzwerk unbekannte Palettenklötze noch besser re-identifizieren kann, als die in dieser Arbeit benutzten Netzwerke. Dabei wäre einerseits die direkte Anwendung auf den Palettenklotzbildern, andererseits aber auch die Anwendung auf eine komprimierte Repräsentation in Form von Merkmalskarten oder Latent-Space-Karten von Interesse. Allerdings müsste das Netzwerk in diesem Fall auf die geringe Größe der Karten angepasst werden.

Literaturverzeichnis

- [1] AHRABIAN, KIAN und BAGHER BABAALI: *Usage of autoencoders and Siamese networks for online handwritten signature verification*. *Neural Computing and Applications*, 31(12):9321–9334, 2019.
- [2] ARAUJO, ANDRE, WADE DAVENPORT NORRIS und JACK SIM: *Computing Receptive Fields of Convolutional Neural Networks*. *Distill*, 2019.
- [3] BANSAL, MONIKA, MUNISH KUMAR und MANISH KUMAR: *2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors*. *Multimedia Tools and Applications*, 80(12):18839–18857, 2021.
- [4] BAY, HERBERT, TINNE TUYTELAARS und LUC VAN GOOL: *Surf: Speeded up robust features*. In: *European conference on computer vision*, Seiten 404–417. Springer, 2006.
- [5] BROMLEY, JANE, ISABELLE GUYON, YANN LECUN, EDUARD SÄCKINGER und ROOPAK SHAH: *Signature Verification Using a Siamese Time Delay Neural Network*. In: COWAN, JACK D., GERALD TESAURO und JOSHUA ALSPECTOR (Herausgeber): *Advances in Neural Information Processing Systems 6, NIPS*, Seiten 737–744. Morgan Kaufmann, 1993.
- [6] CALONDER, MICHAEL, VINCENT LEPETIT, CHRISTOPH STRECHA und PASCAL FUA: *BRIEF: binary robust independent elementary features*. In: *Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10*, Seiten 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.
- [7] CHOPRA, S., R. HADSELL und Y. LECUN: *Learning a similarity metric discriminatively, with application to face verification*. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Band 1, Seiten 539–546 vol. 1, 2005.
- [8] CHOPRA, SUNIL und PETER MEINDL: *Supply chain management. Strategy, planning & operation*. Prentice Hall, 3 Auflage, 2006.

- [9] COTE, LINDA R, RUPA GORDON, CHRISLYN E RANDELL, JUDY SCHMITT und HELENA MARVIN: *Introduction to Statistics in the Psychological Sciences*. Open Educational Resources Collection, 2021.
- [10] GOODFELLOW, IAN, YOSHUA BENGIO und AARON COURVILLE: *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [11] GUPTA, SURBHI, MUNISH KUMAR und ANUPAM GARG: *Improved object recognition results using SIFT and ORB feature detector*. *Multimedia Tools and Applications*, 78(23):34157–34171, 2019.
- [12] HANLEY, JAMES A und BARBARA J MCNEIL: *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. *Radiology*, 143(1):29–36, 1982.
- [13] IOFFE, SERGEY und CHRISTIAN SZEGEDY: *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In: *International conference on machine learning*, Seiten 448–456. PMLR, 2015.
- [14] KARAMI, EBRAHIM, SIVA PRASAD und MOHAMED SHEHATA: *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*, 2017.
- [15] KINGMA, DIEDERIK P und JIMMY BA: *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
- [16] KOCH, GREGORY, RICHARD ZEMEL und RUSLAN SALAKHUTDINOV: *Siamese Neural Networks for One-shot Image Recognition*. In: *ICML deep learning workshop*, Band 2. Lille, 2015.
- [17] KOLBERG, JASCHA, MARCEL GRIMMER, MARTA GOMEZ-BARRERO und CHRISTOPH BUSCH: *Anomaly Detection With Convolutional Autoencoders for Fingerprint Presentation Attack Detection*. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):190–202, apr 2021.
- [18] LAKE, BRENDEN M, RUSS R SALAKHUTDINOV und JOSH TENENBAUM: *One-shot learning by inverting a compositional causal process*. *Advances in neural information processing systems*, 26, 2013.
- [19] LECUN, Y., L. BOTTOU, Y. BENGIO und P. HAFFNER: *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [20] LI, FEI-FEI, ROBERT FERGUS und PIETRO PERONA: *One-Shot Learning of Object Categories*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594–611, 2006.
- [21] LI, MATTHEW D, KEN CHANG, BEN BEARCE, CONNIE Y CHANG, AMBROSE J HUANG, J PETER CAMPBELL, JAMES M BROWN, PRAVEER SINGH, KATHARINA V

- HOEBEL, DENIZ ERDOĞMUŞ et al.: *Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging*. NPJ digital medicine, 3(1):1–9, 2020.
- [22] LIN, MIN, QIANG CHEN und SHUICHENG YAN: *Network in network*. arXiv preprint arXiv:1312.4400, 2013.
- [23] LOWE, DAVID G: *Distinctive image features from scale-invariant keypoints*. International journal of computer vision, 60(2):91–110, 2004.
- [24] MANOCHA, PRANAY, ROHAN BADLANI, ANURAG KUMAR, ANKIT SHAH, BENJAMIN ELIZALDE und BHIKSHA RAJ: *Content-based Representations of audio using Siamese neural networks*, 2017.
- [25] NAIR, VINOD und GEOFFREY E HINTON: *Rectified Linear Units Improve Restricted Boltzmann Machines*. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, Seite 807–814, Madison, WI, USA, 2010. Omnipress.
- [26] NWANKPA, CHIGOZIE, WINIFRED IJOMAH, ANTHONY GACHAGAN und STEPHEN MARSHALL: *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*, 2018.
- [27] PANG, HAIBO, QI XUAN, MEIQIN XIE, CHENGMING LIU, ZHANBO LI und XIAOXIAN YANG: *Research on Target Tracking Algorithm Based on Siamese Neural Network*. Mob. Inf. Syst., 2021, jan 2021.
- [28] PEDRYCZ, WITOLD und SHYI-MING CHEN: *Deep learning: Concepts and architectures*. Springer, 2020.
- [29] ROSIN, PAUL L: *Measuring corner properties*. Computer Vision and Image Understanding, 73(2):291–307, 1999.
- [30] ROSTEN, EDWARD und TOM DRUMMOND: *Fusing Points and Lines for High Performance Tracking*. In: *10th IEEE International Conference on Computer Vision, ICCV*, Seiten 1508–1515, Beijing, China, 2005. IEEE Computer Society.
- [31] RUBLEE, ETHAN, VINCENT RABAUD, KURT KONOLIGE und GARY BRADSKI: *ORB: An efficient alternative to SIFT or SURF*. In: *2011 International Conference on Computer Vision*, Seiten 2564–2571, 2011.
- [32] RUTINOWSKI, JÉRÔME, CHRISTIAN PIONZEWSKI, TIM CHILLA, CHRISTOPHER REINING und MICHAEL TEN HOMPEL: *Towards Re-Identification for Warehousing Entities - A Work-in-Progress Study*. In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Seiten 1–4, 2021.

- [33] SCHROFF, FLORIAN, DMITRY KALENICHENKO und JAMES PHILBIN: *FaceNet: A unified embedding for face recognition and clustering*. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 815–823, 2015.
- [34] SU, WANHUA, YAN YUAN und MU ZHU: *A Relationship between the Average Precision and the Area Under the ROC Curve*. In: *Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR '15*, Seite 349–352, New York, NY, USA, 2015. Association for Computing Machinery.
- [35] SUN, YIFAN, LIANG ZHENG, YI YANG, QI TIAN und SHENGJIN WANG: *Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)*. In: *Proceedings of the European conference on computer vision (ECCV)*, Seiten 480–496, 2018.
- [36] VÉLEZ, IVETTE, CALEB RASCON und GIBRÁN FUENTES-PINEDA: *One-shot speaker identification for a service robot using a cnn-based generic verifier*. arXiv preprint arXiv:1809.04115, 2018.

Eidesstattliche Versicherung

(Affidavit)

Name, Vorname
(surname, first name)

Matrikelnummer
(student ID number)

Bachelorarbeit
(Bachelor's thesis)

Masterarbeit
(Master's thesis)

Titel
(Title)

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem oben genannten Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present thesis with the above-mentioned title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution before.

Ort, Datum
(place, date)


Unterschrift
(signature)

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:

Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to EUR 50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the Chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, Section 63 (5) North Rhine-Westphalia Higher Education Act (*Hochschulgesetz, HG*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund University will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:*

Ort, Datum
(place, date)


Unterschrift
(signature)

***Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**