HTWK

Hochschule für Technik,
Wirtschaft und Kultur Leipzig

Bachelorarbeit
zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

im Bachelorstudiengang Medieninformatik
der Fakultät Informatik und Medien

Entwicklung eines Bilderkennungssystems zum finden von Duplikaten in einer Bilddatenbank bei der Registrierung neuer Bilder

vorgelegt von
Martino Thomann

Leipzig, den 13. September 2023

Erstprüfer:	Prof. Dr. Sibylle Schwarz
Zweitprüfer:	B.Sc. Marc Bellmann

Zusammenfassung

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Bei Spreadshirt werden täglich tausende Designs hochgeladen. Viele davon in der Partner-Area, wo Designer ihre Bilder zum Verkauf anbieten können. Dabei muss seitens Spreadshirt sichergestellt werden, dass die hochgeladenen Designs nicht verfassungsfeindlich sind und auch nicht gegen das Urheberrecht oder Spreadshirts eigene Richtlinien verstoßen. Die manuelle Überprüfung von so vielen Designs ist sehr aufwändig und häufig kommt es vor, dass die gleichen Designs mehrmals hochgeladen werden. Eine automatische Sperrung, von bereits verbotenen Designs, die erneut hochgeladen werden, kann den Überprüfungsprozess entlasten. Daher ist ein System notwendig, dass bei neu hochgeladenen Designs Duplikate in der Datenbank der verbotenen Designs erkennen kann.

1.2 Aufgabenstellung

Ziel dieser Bachelorarbeit ist die Entwicklung eines Systems, dass in der Lage ist, bei neuen Bildern zu erkennen, ob diese bereits in einer Datenbank mit bereits gespeicherten Bildern auftauchen. Ein neues Bild soll auch dann als Duplikat erkannt werden, wenn der Bildinhalt im Vergleich zum Original transformiert, also rotiert, skaliert, verschoben oder gespiegelt wurde. Fälle in denen das Bild anders gefärbt ist oder als Teil eines größeren Bildes auftaucht, sollen ebenfalls berücksichtigt werden.

Zur Implementierung des Systems soll ein "feature-based"(dt. merkmalbasierter) Algorithmus verwendet werden. Es gibt eine Vielzahl an merkmalbasierten Algorithmen, die jeweils ihre eigenen Stärken und Schwächen haben. Eine Auswahl dieser Algorithmen sollen innerhalb der Arbeit für den Anwendungsfall bei Spreadshirt getestet und miteinander verglichen werden.

1.3 Erfolgs- und Qualitätskriterien

Die Güte des Bilderkennungssystems soll anhand von Testdatensätzen ermittelt werden. Die Testdatensätze sind dabei in zwei Teilsätzen unterteilt. Der erste Teilsatz an Bildern stellt die Menge an gespeicherten Datenbankbildern dar. Der zweite Teilsatz enthält eine Untermenge an Duplikaten aus dem Datenbanksatz und eine Menge an neuen Bildern, die nicht im Datenbanksatz auftauchen. Getestet wird in verschiedenen Szenarien, die die Robustheit des Systems gegenüber bestimmter Sonderfälle testen soll. Je nach Szenario sind die Duplikate auf unterschiedliche Weise im Verglich zum Original verändert. Als Vergleich dient der pHash-Algorithmus, der momentan bei Spreadshirt zur Duplikatensuche verwendet wird.

Die verwendeten Metriken werden in den Grundlagen 2.3 erklärt. Am wichtigsten ist dabei ein hoher Recall, sodass möglichst viele Duplikate durch das System abgefangen werden. Da automatisch gesperrte Designs nochmal manuell geprüft werden, fällt eine niedrigere Spezifizität bei der Auswertung nicht so sehr ins Gewicht. Laufzeit- und Speicherkosten sollen ebenfalls innerhalb der Arbeit abgeschätzt werden.

Um für die Verwendung bei Spreadshirt in Frage zu kommen, muss das neue System zuverlässiger sein, als die momentan eingesetzte pHash-Implementaion. Dafür wird ein höherer Recall angestrebt. Um sicherzustellen, dass Spezifizität nicht zu sehr absinkt, wird auf eine höhere oder zumindest gleichbleibende Balancierte-Genauigkeit abgezielt.

2 Grundlagen

2.1 Merkmalbasierte Algorithmen

Ziel der "feature-based"(dt. Merkmalbasierten) Algorithmen ist es markante Punkte innerhalb von Bildern zu finden und zu beschreiben. Die Algorithmen bestehen dabei im Grunde aus zwei Schritten:

1. Schlüsselpunktsuche, bei der nach Koordinaten innerhalb eines Bildes gesucht wird, an denen sich markante Punkte befinden.
2. Erstellung von Deskriptoren, die den Bereich um die Schlüsselpunkte beschreiben. Diese sollen später mit Deskriptoren aus anderen Bildern verglichen werden, um gemeinsame Merkmale zu finden.

Wie genau diese beiden Schritte implementiert sind ist je nach Algorithmus unterschiedlich.

Meistens handelt es sich bei den gefundenen Merkmalen um Rand- und Eckpunkte oder Details auf einer Fläche.

2.1.1 SIFT: Scale-Invariant Feature Transform

Ziel des SIFT-Algorithmus ist es Merkmale in einem Bild zu finden und mit Deskriptoren zu beschreiben, die Robust gegenüber Rotation, Verschiebung und Skalierung sind. Auch der Einfluss durch Bildrauschen und Verzerrung soll möglichst gering sein. Dabei durchläuft der Algorithmus 4 Schritte. Schritt 1 bis 3 umfassen die Suche und Beschreibung der Schlüsselpunkte, während Schritt 4 sich mit Erstellung der Deskriptoren befasst. [Low99, S. 2]

1. Suche nach Extrempunkten SIFT sucht nach Extrempunkten innerhalb von Bildern. Extrempunkte sind dabei Punkte, an denen lokale Maxima oder Minima in der Helligkeit auftreten. Also helle Punkte innerhalb von dunklen Bereichen und dunkle Punkte innerhalb von hellen Bereichen.

Um Extrempunkte zu finden wird der "Difference of Gaussian"genutzt. Zuerst wird eine bestimmte Anzahl an Bildern, durch das Weichzeichnen des Originalbilds mit dem Gauschen Weichzeichner, erstellt. Der Grad der Unschärfe steigt dabei bei jedem Bild um einen konstanten Faktor. Ein Satz dieser weichgezeichneten Bilder wird als Oktave bezeichnet. Für jedes paar an aneinander grenzenden weichgezeichneten Bildern, wird die Differenz berechnet. Bei dieser Differenz handelt es sich um den "Difference of Gaussian". Durch den "Difference of Gaussian"werden Rand- und Eckpunkte, sowie andere Details im Bild hervorgehoben. Bei der Implementation von SIFT werden pro Oktave meist fünf weichgezeichnete Bilder verwendet, aus denen vier "Difference of Gaussian"Bilder entstehen. [Low04, S. 94]

Jeder Pixel der mittleren "Difference of Gaussian"Bilder wird mit seinen acht Nachbarn innerhalb desselben Bildes und seinen je neun Nachbarn im DOG darüber und darunter verglichen. Ein Pixel ist dann ein Schlüsselpunkt, wenn sein Helligkeitswert größer oder kleiner als der Wert all seiner Nachbarn ist. [Low04, S. 95]

Diese Extrempunktsuche wird dabei für mehrere Oktaven durchgeführt. Dabei wird für jede Oktave das Originalbild um den Faktor 2 herunter skaliert. Das ist notwendig, da manche Extrempunkte nur bei bestimmten Auflösungen auffindbar sind. [Low04, S. 94f] Durch die Sammlung von Schlüsselpunkten auf verschiedenen Auflösungen, sind diese bei Bildern in verschiedenen Auflösungen vergleichbar. Der Aufbau von "Difference of Gaussian"Bilder innerhalb von mehreren Oktaven wird als scale-space (dt. Bildpyramide) bezeichnet.

2. Akkurate Ortsbestimmung der Schlüsselpunkte Die Positionen der Schlüsselpunkte, die auf niedrigen Auflösungen gefunden wurden, entspricht gegebenenfalls nicht genau der Position in der Originalauflösung. Deshalb wird die Taylorreihe verwendet, um die tatsächliche Position zu interpolieren. [Low04, S. 97f]

Außerdem werden Schlüsselpunkte an Rändern und Schlüsselpunkte mit niedrigem Kontrast zu ihrer Umgebung herausgefiltert. Das ist nötig, da solche Schlüsselpunkte Deskriptoren hervorbringen, die nicht zuverlässig verglichen werden können. [Low04, 98f]

3. Bestimmung der Schlüsselpunkt Ausrichtung Um Schlüsselpunkte und die aus ihnen resultierenden Deskriptoren robust gegenüber Bildrotationen zu machen, wird jedem Schlüsselpunkt eine Ausrichtung zugewiesen. Damit diese Ausrichtung konsistent ist, wird diese anhand der lokalen Bildeigenschaften um den Schlüsselpunkt herum bestimmt. [Low04, S. 99]

Um das zu erreichen, wird ein 360 Grad umfassendes Histogramm mit 36 bins aufgebaut. Anhand der Skalierung des Schlüsselpunkts wird ein weichgezeichnetes Bild mit passender Skalierung gewählt. Innerhalb dieses Bildes werden Abtastpunkte um den Schlüsselpunkt herum genommen. Für jeden Abtastpunkt wird die Orientierung und Stärke des Gradienten bestimmt. Der Gradient beschreibt dabei die Änderung der Pixelhelligkeit innerhalb eines Bildes. Sowohl Stärke als auch Orientierung des Gradienten sind für alle Ebenen der Bildpyramide vorberechnet worden. [Low04, S. 99]

Jede Gradient-Orientierung wird in das Histogramm eingetragen und anhand der Gradient-Stärke gewichtet. Da das Histogramm 36 bins hat, werden Orientierungen in 10-Grad-Schritten zusammengefasst. Aus dem Histogramm wird die Orientierung des globalen Maximums, sowie die Orientierungen aller lokalen Maxima die mindestens 80 Prozent des globalen Maximums betragen, entnommen. Für jede entnommene Orientierung wird ein Schlüsselpunkt erstellt. Dadurch kann es mehrere Schlüsselpunkte mit der gleicher Position und Skalierung aber mit unterschiedlichen Orientierungen geben. [Low04, S. 100]

4. Deskriptoren erstellen In dem Bereich um den Schlüsselpunkt werden wie im Schritt 3 (Abschnitt 2.1.1) für jeden Abtastpunkt die Gradienten-Stärke und -Orientierung bestimmt. Der Bereich wird dabei in 4x4 große Unterbereiche aufgeteilt, für die ein Histogramm erstellt wird. Dabei werden für das Histogramm nur 8 bins verwendet, in denen Gradienten-Stärken mit ähnlicher Orientierung aufsummiert werden. Aus diesen Werten wird ein Deskriptor erstellt. Üblicherweise werden Abtastbereiche mit einer Größe von 16x16 verwendet. Daraus ergeben sich insgesamt 16 Unterbereiche der Größe 4x4 mit je 8 Orientierungs-bins. Ein typischer Deskriptor ist ein Vektor mit 16 * 8 = 128 Werten. [Low04, S. 101]

2.1.2 ORB: Oriented FAST and Rotated BRIEF

ORB basiert auf dem FAST-Algorithmus für die Schlüsselpunktsuche und dem BRIEF-Algorithmus für die Generierung von Deskriptoren

Schüsselpunktsuche mit FAST Der FAST-Algorithmus sucht nach Eckpunkten in einem Bild. Dazu wird um jeden Pixel im Bild ein Kreis mit 16 Randpixeln gezogen. Der betrachtete Kandidat-Pixel ist genau dann ein Eckpunkt, wenn eine bestimmte Anzahl an aneinander grenzenden Randpixel heller oder dunkler sind als der Kandidat-Pixel plus oder minus einem bestimmten Schwellenwert. [RD06, S. 4]

Bei der Implementation in ORB müssen 9 der 16 Randpunkte heller oder dunkler als der Kandidat-Pixel sein, damit dieser sich als Randpunkt qualifiziert. Zudem wird die "Harris corner measure"genutzt um Randpunkte herauszufiltern. Dazu wird jedem Schlüsselpunkt ein Wert zugeteilt, der aussagt, wie wahrscheinlich es ist, dass es sich bei dem Schlüsselpunkt um einen Eckpunkt handelt. Der Schwellenwert, um sich als Eckpunkt zu qualifizieren, wird niedrig genug gesetzt um eine bestimmte Mindestanzahl an Schlüsselpunkten zu erreichen. Die Mindestanzahl an Schlüsselpunkten, kann dabei je nach Anwendungsfall variieren. Es muss genügend Schlüsselpunkte geben, um das Bild zu beschreiben. Gleichzeitig möchte man aber Schlüsselpunkte vermeiden deren Deskriptoren nicht zuverlässig verglichen werden können. [RRKB11, S. 2565]

Die durch FAST gefundenen Schlüsselpunkte sind anders als bei SIFT noch nicht Robust gegenüber Rotations- und Auflösungsunterschieden. Um Invarianz gegenüber Auflösungsunterschieden entgegenzuwirken, generiert ORB ähnlich wie SIFT (Abschnitt 2.1.1 mehrere niedriger aufgelöste Varianten des Originalbilds. Für jede Auflösung wird die Schlüsselpunktsuche mit FAST und Randpunktfilterung mit "Harris corner measure"durchgeführt. Um die Orientierung des Schlüsselpunkts zu ermitteln, wird in dessen Umgebung nach dem Pixel mit dem höchsten Helligkeitswert gesucht. Die Orientierung ergibt sich dann aus dem Vektor zwischen diesen beiden Punkten. [RRKB11, S. 2565]

Erstellen der Deskriptoren mit BRIEF Der BRIEF-Algorithmus definiert um den Schlüsselpunkt einen Bereich P, der weichgezeichnet wird. Aus diesem Bereich werden n Pixelpaare (x,y) genommen und einem Binärtest T unterzogen. Für jedes Pixelpaar werden die Helligkeitswerte P(x) und P(y) verglichen. Je nachdem ob der Helligkeitswert P(x) größer als P(y) ist, ergibt der Binärtestt 0 oder 1. In der Implementation bei ORB werden 256 Pixelpaare genutzt. Der Deskriptor ist eine Aneinanderreihung der Testergebnisse aller 256 Pixelpaare. Somit ergibt sich ein 256 Bit Deskriptor pro gefundenen Schlüsselpunkt. [RRKB11, S. 2566]

Damit die durch BRIEF erstellten Deskriptoren robust gegenüber Rotationen sind, nutzt ORB die Orientierung die für die Schlüsselpunkte ermittelt wurden. Dazu wird eine Matrix S mit den Pixelpaaren für den Binärtest erstellt. Gemäß der Orientierung O des Schlüsselpunkts wird eine Rotationsmatrix Ro. Matrix S wird mit der Rotationsmatrix

Ro rotiert und ergibt So. Der Deskriptor für den Schlüsselpunkt wird anhand der rotierten Pixelpaare aus So erstellt. [RRKB11, S. 2566]

2.2 Match-Suche mit Deskriptoren

Im Abschnitt 2.1 wurden die Merkmalbasierten-Algorithmen erklärt, mit denen Deskriptoren für Merkmale in einem Bild generiert werden können. In diesem Abschnitt soll es darum gehen, wie diese Deskriptoren genutzt werden können, um die Merkmale aus zwei Bildern zu vergleichen und Übereinstimmungen zu finden. Übereinstimmungen werden auch als Match bezeichnet.

2.2.1 Deskriptoren Vergleichen

Um zwei Deskriptoren zu vergleichen, wird die Distanz zwischen den beiden berechnet. Die Distanz beschreibt in diesem Fall wie ähnlich sich Deskriptoren sind. Je kleiner die Distanz desto ähnlicher die Deskriptoren. Abhängig davon welche Art Deskriptor verglichen wird, wird ein anderer Distanzmaßstab verwendet.

Hamming-Distanz Die Hamming-Distanz kann verwendet werden, um Bitfolgen zu vergleichen. Dazu werden beide Bitfolgen Stelle für Stelle verglichen. Für jede Stelle, die nicht übereinstimmt, steigt die Distanz um 1.

$$distance(desc1, desc2) := |\{i \in \{1, ..., 256\} \mid desc1_i \neq desc2_i\}|$$

Die Hamming-Distanz eignet sich um Deskriptoren, die durch den ORB-Algorithmus errechnet werden. Bei diesen handelt es sich um Bitfolgen mit 256 Stellen. Auch bei dem Vergleich von pHash-Werten kommt die Hamming-Distanz zum Einsatz.

Euklidische-Distanz Bei Deskriptoren, die aus dem SIFT-Algorithmus resultieren, wird die Euklidische Distanz genutzt. Die Deskriptoren sind hierbei eine Folge von Gradient-Werten. Stelle für Stelle wird die Euklidische Distanz für alle Gradient-Werte der Deskriptoren berechnet und aufsummiert.

$$distance(desc1, desc2) := \sum_{i=1}^{128} \sqrt{(desc1[i] - desc2[i])^2}$$

SIFT-Deskriptoren bestehen in der Regel aus 128 Gradient-Werten.

2.2.2 Brute-Force Matcher

Um Übereinstimmungen von Merkmalen in zwei Bildern zu finden, werden alle Deskriptoren des ersten Bilds mit allen Deskriptoren des zweiten Bilds verglichen. [mat]

Der Vorteil dieser simplen Herangehensweise ist, dass garantiert die besten Übereinstimmungen gefunden werden. Allerdings ist diese Strategie mit $\mathcal{O}(n^2)$ auch sehr ineffizient, wodurch sie gerade bei Datensets mit einer großen Menge an Deskriptoren ins Gewicht fällt.

2.2.3 Filterung der Matching-Ergebnisse

Für jeden Deskriptor aus dem ersten Bild wird der erst- und zweitbeste Match aus dem zweiten Bild gespeichert. Ein Match besteht dabei aus den beiden Deskriptoren, die verglichen werden, und die resultierende Distanz (Siehe Abschnit 2.2.1). Daraus ergibt sich eine 2xN Matrix, wobei N die Anzahl der Deskriptoren im ersten Bild darstellt.

Es können Deskriptoren vorkommen, für die keine zuverlässiger Match gefunden werden kann. Das kann unter anderen passieren, wenn Schlüsselpunkte an Rändern von Objekten oder wirren Bereichen im Bild gefunden werden. Da mit solchen Deskriptoren keine gute Aussage über die Ähnlichkeit von zwei Bildern getroffen werden kann, müssen diese herausgefiltert werden.

Dazu schlägt Lowe den sogenannten "ratio test"vor. Dabei wird die Distanz des erst- und zweitbesten Matchs verglichen. Wenn das Verhältnis zwischen den beiden Distanzen kleiner als 0.8 ist, wird der erstbeste Match behalten. Ansonsten wird der Match verworfen. Dieses Vorgehen basiert auf der Annahme, dass die Distanz zum erstbesten Deskriptor deutlich niedriger als die Distanz zum Zweitbeste sein muss, damit die Übereinstimmung als zuverlässig gilt. [Low04, S.104]

2.3 Test-Metriken

Das Bilderkennungssystem soll die Bilder in zwei Klassen einordnet: Duplikate und Unikate. Da es in diesem Fall bei der Klassifizierung nur zwei mögliche Klassen gibt, kann man das Bilderkennungssystem als binären Klassifikator bezeichnen.

Die Performance von binären Klassifikatoren kann durch die Werte einer Wahrheitsmatrix quantifiziert werden. Die Wahrheitsmatrix gibt dabei an, wie viele richtige und falsche Entscheidungen das System bei der Klassifikation getroffen hat. [Tha20, S. 170]

System Klass.:	Duplikat	Unikat
Duplikat	Anzahl richtige Duplikate (TP)	Anzahl falsche Duplikate (FP)
Unikat	Anzahl falsche Unikate (FN)	Anzahl richtige Unikate (TN)

Tabelle 2.1: Wahrheitsmatrix für Duplikaten-Suche

Aus den Werten der Wahrheitsmatrix lassen sich weitere Metriken ableiten, die für die Bewertung eines binären Klassifikators nützlich sein können. Interessant für diese Arbeit sind Recall, Spezifizität und Balancierte-Genauigkeit.

Der Recall, oder auch true positive rate, gibt das Verhältnis zwischen den Duplikaten, die das System korrekt klassifiziert hat, und allen Duplikaten, die sich in dem Suchdatensatz befinden, an. [Tha20, S. 172]

$$Recall = \frac{TP}{TP + FN}$$

Die Spezifizität, oder auch true negative rate, gibt das Verhältnis zwischen den Unikaten, die das System korrekt klassifiziert hat, und allen Unikaten, die sich im Suchdatensatz befinden an. [Tha20, S. 172]

$$Specificity = \frac{TN}{FP + TN}$$

Die Accuracy (dt. Genauigkeit), gibt das Verhältnis zwischen den Bildern, die das System korrekt klassifiziert hat, und allen Bildern im Suchdatensatz an. Sie spiegelt die Fähigkeit des Systems Bilder richtig zu Klassifizieren wieder. [Tha20, S. 171]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{Recall * P + Specificity * N}{P + N}$$

Allerdings ist die Genauigkeit kein zuverlässiger Wert, wenn man mit unausgeglichenen Datensätzen arbeitet. Ein Datensatz gilt dann als unausgeglichen, wenn einer Klassifikation mehr Elemente angehören, als der anderen Klassifikation. [Tha20, S. 171] In den Testdatensätzen, die für diese Arbeit verwendet werden, befinden sich mehr Unikate als Duplikate. Dieses Ungleichgewicht kann die Genauigkeit stark beeinflussen. So kann zum

Beispiel eine gute Spezifizität einen schlechten Recall ausgleichen, wenn der Datensatz zum größten Teil aus Unikaten besteht.

Daher kommt bei unbalancierten Datensätzen die balanced-accuracy (dt. Balancierte-Genauigkeit) zum Einsatz. [Tha20, S. 175]

$$Balanced - Accuracy = \frac{Recall + Specificity}{2}$$

Da sie den Durchschnitt aus Recall und Spezifizität bildet, werden Unterschiede zwischen den beiden Werten ausgeglichen. Dadurch ist die Balancierte-Genauigkeit robust gegenüber unausgeglichenen Datensätzen.

3 Analyse

3.1 Baseline-Algorithmus (pHash)

Spreadshirt nutzt momentan einen Perceptual-Hashing-Algorithmus (kurz pHash) um neu hochgeladene Bilder zu erkennen, die bereits verboten wurden. Die pHash Implementierung bei Spreadshirt stützt sich auf den Blockeintrag [pHa] und lässt sich grob in drei Phasen unterteilen.

Vorverarbeitungsphase Zuerst kommt das Bild in eine Vorverarbeitungsphase. Falls das Bild einen transparenten Hintergrund hat, wird es so zugeschnitten, dass der Bildrand direkt Rand des Motivs liegt. Danach wird das Bild auf eine Auflösung von 32x32 herunter skaliert und in ein Graustufenbild umgewandelt.

Umwandlung in den Frequenzraum In dieser Phase wird das Bild vom Orts in den Frequenzraum überführt. Dazu wird die diskrete Kosinustransformation genutzt. Es entsteht eine 32x32 DCT-Matrix bestehend aus Koeffizienten, mit denen, in Kombination mit der Kosinusfunktion, das Originalbild rekonstruiert werden kann.

Für pHash wird obere linke 8x8 Bereich der Matrix genutzt. Dieser Bereich enthält die Niederfrequenzkomponenten und wird als reduzierte DCT-Matrix bezeichnet. Die Niederfrequenzkomponenten sind interessant für pHash, da diese die groben Strukturen und Muster des Bilds beschreiben. Zwei Bilder mit ähnlichen Strukturen und Mustern werden ähnliche Niederfrequenzkomponenten haben.

Hash-Berechnung Für die Hash-Berechnung wird der Median in der reduzierten DCT-Matrix ermittelt. In der reduzierten DCT-Matrix der Größe 8x8 befinden sich insgesamt 64 Werte. Für jeden Wert wird geprüft, ob er über oder unter dem Median liegt. Wenn der Wert größer ist, wird eine 1 und ansonsten eine 0 an den Hash-Wert angehängt. Damit ergibt sich ein 64-bit Hash-Wert für ein Bild.

pHash-Werte vergleichen Zum vergleichen von pHash-Werten, die Hamming-Distanz (Siehe Abschnitt 2.2.1)genutzt. Bei Spreadshirt gilt ein Bild als Duplikat, wenn beim Hash-Vergleich mit einem Bild aus der Datenbank eine Hamming-Distanz kleiner oder gleich 4 herauskommt.

4 Formel

5 Test-System

6 Versuche und Auswertung

7 Neues Bilderkennungs-System

8 Schluss

Literaturverzeichnis

[Low99] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[mat] Opencv feature matching dokumentation.

[pHa] Blockeintrag phash.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[Tha20] Alaa Tharwat. Classification assessment methods. *Applied computing and informatics*, 17(1):168–192, 2020.

Abbildungsverzeichnis

Tabellenverzeichnis

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Graduierungsarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommene Stellen sind als solche einzeln kenntlich gemacht.

Diese Arbeit ist bislang keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht worden.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Leipzig, 13. September 2023 Unterschrift