

Ostbayerische Technische Hochschule Regensburg
Fakultät für Informatik und Mathematik

Wissenschaftliche Abhandlung

Möglichkeiten zur Bestimmung der visuellen Ähnlichkeit zwischen zwei Bildern

Vorgelegt von: Andreas Huber <andreas.huber@st.oth-regensburg.de>
Matrikelnummer: 3370380
Studiengang: Master Informatik (Schwp. Software Engineering)
Betreuer: Prof. Dr. techn., Dipl.-Ing. Markus Kucera
Abgabedatum: 8. Dezember 2022

Inhaltsverzeichnis

Abbildungsverzeichnis	II
1 Einleitung	1
2 Übersicht inhaltsbasierter Bildvergleichsalgorithmen	1
2.1 Mean squared error (MSE)	1
2.2 Perceptual Hashing	2
2.3 Histogram Intersection	3
2.4 Scale-Invariant Feature Transform (SIFT)	4
2.5 Structural Similarity Index (SSIM)	7
2.6 Siamese Network (Deep Learning)	8
3 Fazit und Ausblick	9
Quellenverzeichnis	10

Abbildungsverzeichnis

1	MSE: Anwendung an verschiedenen Testbildern	1
2	PHASH: Anwendung an verschiedenen Testbildern	3
3	SIFT: Erzeugung der Bilderserie	4
4	SIFT: Anwendung Gaußscher Differenzfunktion	5
5	SIFT: Schlüsselpunkt-Lokalisierung	5
6	SIFT: Zuordnung der Orientierung	6
7	SIFT: Pro Schlüsselpunkt erzeugte Deskriptoren (Fingerabdruck)	6
8	SIFT: Anwendung an verschiedenen Testbildern	7
9	SSIM: Anwendung an verschiedenen Testbildern	8

1 Einleitung

Visuelle Ähnlichkeitsbestimmung von Bildern spielt in der heutigen Zeit eine immer wichtigere Rolle. Sie wird neben der Gesichtserkennung beispielsweise auch für Dublettenerkennung, Anwendung des Urheberrechts-Diensteanbieter-Gesetz, Qualitätsprüfung und vieles mehr verwendet. Da diese Problematik immer noch ein offenes Forschungsthema ist und es bisher keine allgemeingültige Lösung gibt, beschäftigt sich dieses Paper mit einer Übersicht aller aktuell gängigen Kategorien an „Content-Based Image Retrieval“-Algorithmen (CBIR).

2 Übersicht inhaltsbasierter Bildvergleichsalgorithmen

2.1 Mean squared error (MSE)

Ein naiver Ansatz zwei Bilder auf visuelle Ähnlichkeit zu bewerten, wäre es iterativ alle Pixelintensitäten miteinander zu vergleichen. Als Werkzeug zum Vergleich dient hierbei die mittlere quadratische Abweichung (Monsters, 2020). Je niedriger der Abstand, desto mehr stimmen die Bilder miteinander überein. Obwohl dieses Verfahren eine schnelle Performance aufweist, ist es nur für Fälle geeignet, bei denen das Referenz- und das Suchbild fast identisch sind. Bereits kleine Helligkeitsänderungen, bei gleichbleibendem Inhalt, ergeben nach Berechnung, des euklidischen Abstands, eine hohe mittlere quadratische Abweichung (siehe Abbildung 1). (Antoniadis, 2022)



Abbildung 1: MSE: Anwendung an verschiedenen Testbildern
Quelle: Eigene Darstellung

2.2 Perceptual Hashing

Neben der mittleren quadratischen Abweichung ist eine der einfachsten, aber auch anfälligsten Methoden zur Bestimmung der Ähnlichkeit von zwei Bildern das Hashing. Dabei gibt es viele verschiedene Ansätze und Vorgehensmodelle. Die gängigste Methode ist die Anwendung des pHashes, auch genannt Perceptual Hashing. (Ramos, 2022)

Beim Perceptual Hashing werden sowohl für Bild A als auch für Bild B ein Hash berechnet. Die daraus resultierende Hamming-Distanz zwischen den Hashes ergibt die Ähnlichkeit der Bilder. Je geringer die Distanz, desto ähnlicher. Das Verfahren ist nicht normiert und noch ein offenes Forschungsthema. (Klinger & Starkweather, 2010)

Im Folgenden wird ein beispielhaftes pHash-Verfahren erläutert. Zuerst wird sowohl das Referenz- als auch das Suchbild in eine Graustufen-Grafik umgewandelt. Schließlich wird die Grafik auf 32x32 Pixel skaliert. Auf das entstandene Grauwertbild folgen zwei Diskrete Kosinus Transformationen (1. Pro Zeile, 2. Pro Spalte). Die hochfrequenten Abschnitte befinden sich nun links oben in einer 8x8 Matrix. Daraufhin wird der Median-Grauwert der 64 Pixel berechnet. Jeder Pixel, dessen Grauwert unter dem Durchschnitt liegt wird weiß eingefärbt, der Rest schwarz. Daraus ergibt sich ein 64-Bit langer Hashwert (schwarz: 0, weiß: 1). Zuletzt wird durch eine Subtraktion der beiden generierten Bitfolgen die Hamming-Distanz berechnet. Mittels einem vorher festgelegten Schwellenwert wird jetzt auf Basis des Hamming-Abstands bestimmt, ob die Bilder als ähnlich eingestuft werden. Je niedriger der Abstand, desto größer die Übereinstimmung. (Ramos, 2022)

Vorteile

- Schnelle Suchperformance, wenn die Hashes der Referenzbilder bereits in einer dafür geeigneten Datenstruktur (z.B. k-d-Bäume, VP Bäume oder Kugelbäume) vorliegen (Christina, 2020)
- Einfach zu implementieren
- Robust gegen Wasserzeichen, Farbfilter, leichte Helligkeits- und Kontraständerungen, Gammakorrekturen, Skalierungen sowie Komprimierungen (siehe Abbildung 2) (Klinger & Starkweather, 2010)

Nachteile

- Nicht robust gegen Spiegelungen, Rotierungen und Verzerrungen (siehe Abbildung 2))
- Nicht robust gegen Zuschneidungen, neu eingefügten Elementen oder Änderungen des Blickwinkels

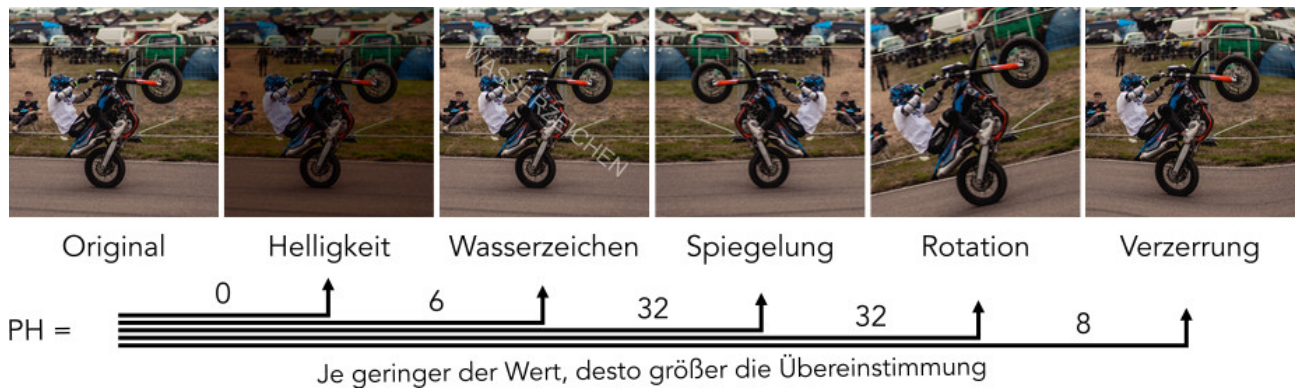


Abbildung 2: PHASH: Anwendung an verschiedenen Testbildern
Quelle: Eigene Darstellung

2.3 Histogram Intersection

Ein robusteres, aber auch rechenaufwendigeres Verfahren ist der Histogram Intersection Algorithmus von Swain und Ballard. Histogramme sind stabil gegenüber Verschiebungen und Rotationen um die Betrachtungsachse. Außerdem ändern sie sich bei Änderungen des Blickwinkels, des Maßstabs und bei Verdeckung von Elementen nur langsam. Gegen Belichtungsänderungen können stark reduzierte Farbhistogramme jedoch Probleme bereiten. (Swain & Ballard, 1991)

Im Standardfall wird die Histogram Intersection auf Farbhistogramme angewandt. Zuerst muss die Menge an zu diskretisierenden Farben im Histogramm festgelegt werden - zum Beispiel 200. Folglich gibt es 200 mögliche „Farbeimer“ in die wir jeden Pixel der Bilder einsortieren. Mit den durch das Referenz- und das Suchbild berechneten Histogrammen, kann schließlich die Überschneidung festgestellt werden. Je stärker sich die Histogramme überschneiden, desto ähnlicher sind die Bilder. (Lee et al., 2004) Zusätzlich wird wie beim Hashing ein gewisser Schwellenwert vorher definiert. Der Vorgang kann beispielsweise auch auf die einzelnen Farbkanaäle aufgeteilt werden. Bei RGB würde das bedeuten, dass man jeweils den Rot-, Grün-, und Blau-Kanal einzeln betrachtet. (Swain & Ballard, 1991)

Weitere Untersuchungen haben ergeben, dass sowohl die Wahl des Farbraums als auch die Festlegung des Quantization Levels (Anzahl der Farbeimer) eine große Rolle bei der Erfolgsschance dieses Vorgehensmodells spielen. Hierbei sorgte wohl der CIELab-Farbraum im Allgemeinen für die besten Ergebnisse. (Lee et al., 2004)

Eine zusätzliche Möglichkeit die Genauigkeit des Algorithmus zu verbessern ist das Hinzufügen und Vergleichen anderer Histogramme. Ein mögliches Beispiel wäre ein „Texture Direction“ oder „Texture Scale“ Histogramm. In diesen Fällen wird nicht versucht anhand der Farben eine Ähnlichkeit zu erkennen, sondern durch die Struktur des Bildes. Hierbei könnten

beispielsweise aus dem Bild extrahierte Ecken oder Kanten in ein Histogramm sortiert werden und schließlich auf Überschneidungen analysiert werden. (Simek, 2009) Offensichtlich wird jedoch jeder weitere zusätzlicher Berechnungsschritt die allgemeine Performance des Systems beeinträchtigen.

2.4 Scale-Invariant Feature Transform (SIFT)

David G. Lowe hat mit der Einführung des SIFT-Algorithmus einen für die Bildverarbeitung wertvollen Beitrag geleistet. Der SIFT-Algorithmus findet lokale Merkmale in Bildern. Diese Merkmale sind invariant gegenüber Bildskalierung und -drehung und teilweise invariant gegenüber Änderungen der Beleuchtung sowie des Betrachtungswinkels. Die Wahrscheinlichkeit einer Störung durch Verdeckung einzelner Elemente oder auftretenden Bildrauschens wird, durch räumliche Streuung und einem vielseitigen Frequenzbereich, erreicht. Im Folgenden wird die grobe Vorgehensweise zur Findung der lokalen Merkmale erläutert. (Lowe, 2004)

1. Scale-space extrema detection

Zu Beginn wird aus einem Bild durch repetetives Unschärfen (mittels Gaußschen Unschärfefilter) und Halbieren der Größe eine Serie an Bildern erzeugt (siehe Abbildung 3). Auf die Bilder der Bilderserie folgt die Anwendung der Gaußschen Differenzfunktion (siehe Abbildung 4). (Sinha, n. d. a)

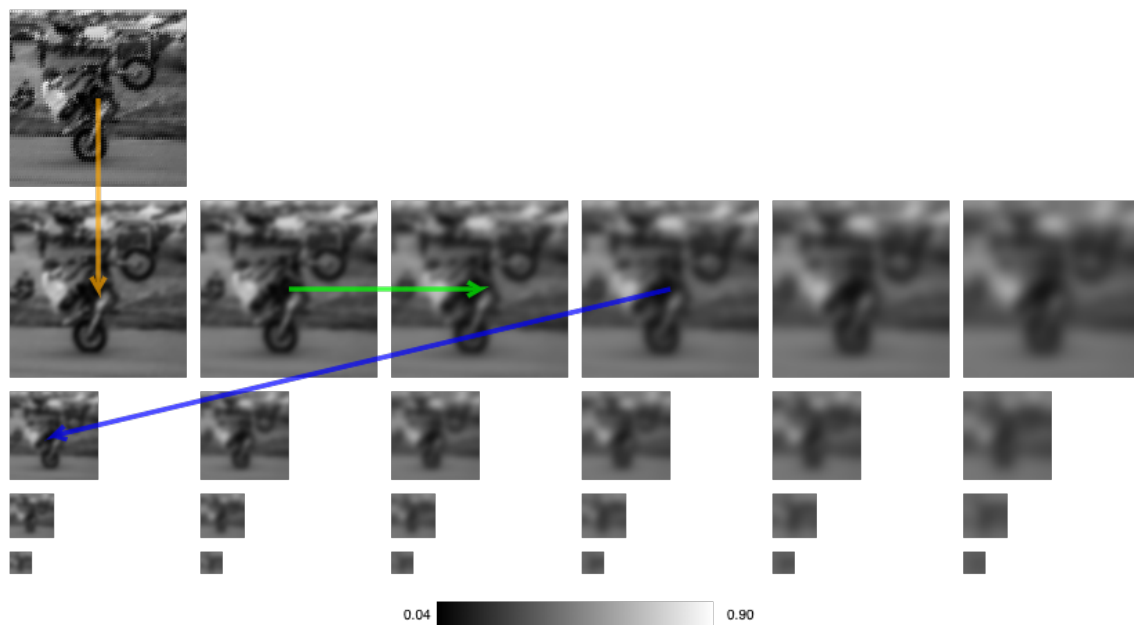


Abbildung 3: SIFT: Erzeugung der Bilderserie
Quelle: Eigene Darstellung; <http://weitz.de/sift/>

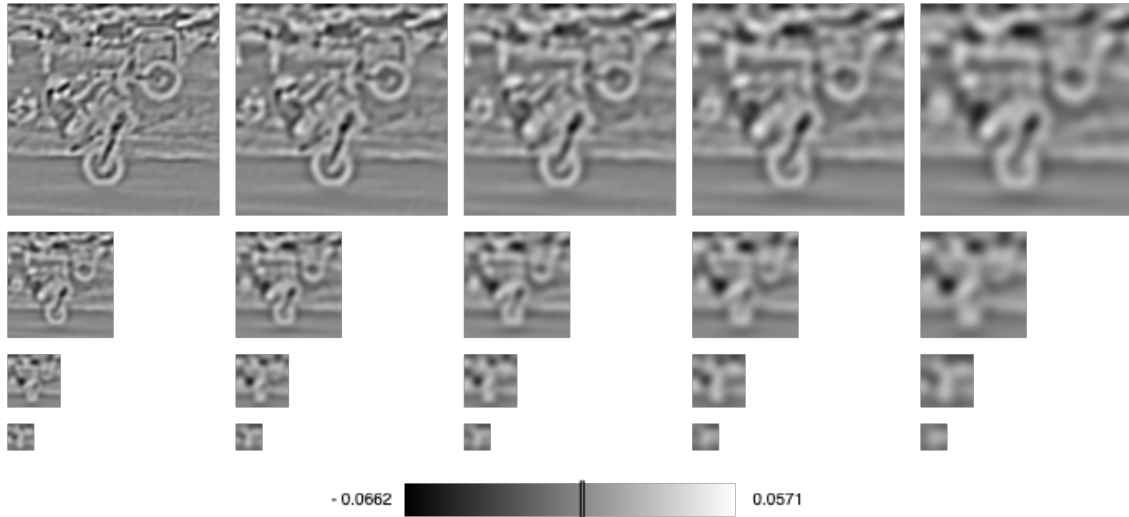
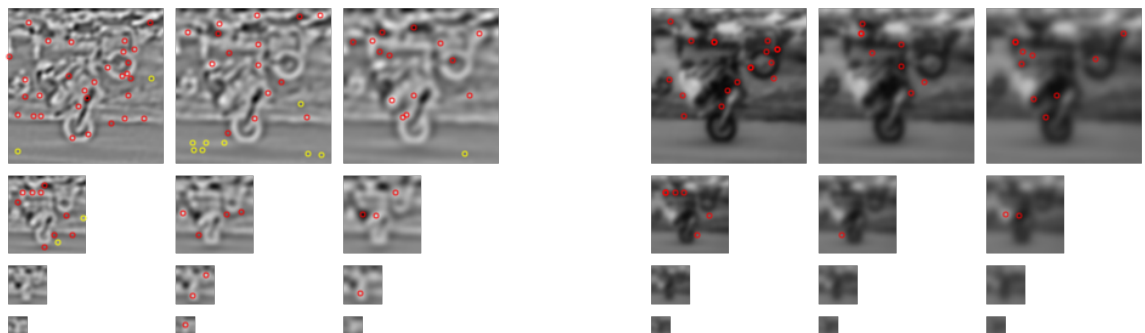


Abbildung 4: SIFT: Anwendung Gaußscher Differenzfunktion
Quelle: Eigene Darstellung; <http://weitz.de/sift/>

2. Keypoint localization

Aus den vorher differenzierten Grafiken werden jetzt mit dem Marr-Hildreth-Operator interessante Schlüsselpunkte bzw. Merkmale herausgefiltert (siehe Abbildung 5a). (Sinha, n. d. b) Eine dem Harris-Corner Detector ähnliche Prozedur wird angewandt, um Punkte auf Linien zwischen Ecken und Punkte mit wenig Kontrast (uninteressante Punkte) herauszusieben (siehe Abbildung 5b). (Sinha, n. d. c)



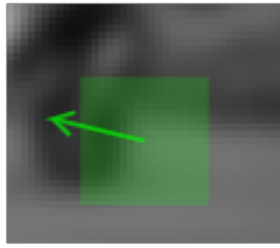
(a) Extrahierte Schlüsselpunkte

(b) Filterung wichtiger Schlüsselpunkte

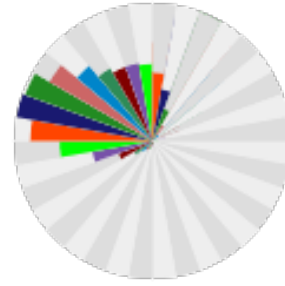
Abbildung 5: SIFT: Schlüsselpunkt-Lokalisierung
Quelle: Eigene Darstellung; <http://weitz.de/sift/>

3. Orientation assignment

Mithilfe eines Histogramms kann den Schlüsselpunkten nun eine Orientierung zugewiesen werden (siehe Abbildung 6). Dadurch werden die Punkte neben der Robustheit gegenüber Skalierung und Verschiebung auch gegen Rotation invariant (siehe Abbildung 8). (Sinha, n. d. d)



(a) Orientierung



(b) Zugehöriges Farbhistogramm (normalisiert)

Abbildung 6: SIFT: Zuordnung der Orientierung

Quelle: Eigene Darstellung; <http://weitz.de/sift/>

4. Keypoint descriptor

Zuletzt wird nach Anwendung einiger mathematischer Verfahren eine Art eindeutiger Fingerabdruck des Merkmals berechnet (siehe Abbildung 7). Schließlich wird dadurch die Stabilität gegenüber begrenzter lokaler Formverzerrungen und Beleuchtungsänderungen erreicht (siehe Abbildung 8). (Sinha, n. d. e)

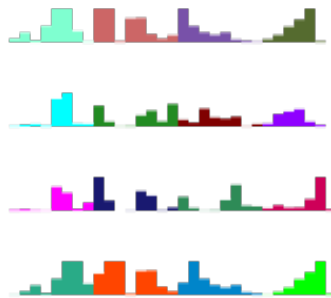


Abbildung 7: SIFT: Pro Schlüsselpunkt erzeugte Deskriptoren (Fingerabdruck)

Quelle: Eigene Darstellung; <http://weitz.de/sift/>

Nach Anwendung dieses Algorithmus muss man bei einem 500x500 Pixel großen Bild mit in etwa 2000 robusten Merkmalen rechnen. Werden die gefundenen Merkmale in einer Datenbank gespeichert, können sie schließlich in nahezu Echtzeit auf die Schlüsselpunkte eines Suchbilds verglichen werden. (Lowe, 2004)

Ein ähnlicher patentierter Algorithmus zur Erkennung von Bildmerkmalen ist der „Speeded up robust features (SURF)“ (Bay et al., 2006). Ferner gibt es viele ähnliche frei verfügbare Algorithmen der Gruppe „Bildmerkmalsfindung“.

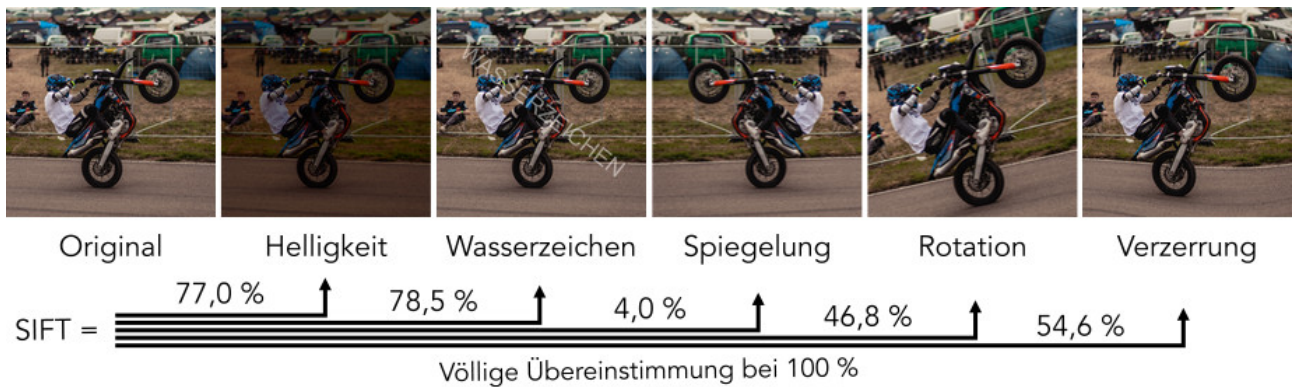


Abbildung 8: SIFT: Anwendung an verschiedenen Testbildern
Quelle: Eigene Darstellung

2.5 Structural Similarity Index (SSIM)

Anders als die meisten anderen Vorgehensmodelle versucht die menschliche Wahrnehmung nicht Fehler zu Quantifizieren, sondern strukturelle Informationen aus einer Szene wiederzuerkennen. Um dieses Verhalten nachzuahmen, haben Forschende Mitglieder der IEEE den Structural Similarity Index entworfen. (Wang et al., 2004)

Der Index extrahiert, vergleicht und kombiniert folgende drei Parameter:

- Leuchtdichte
- Kontrast
- Struktur

Im ersten Schritt wird die Leuchtdichte durch Mittelwertbildung über alle Pixelwerte gemessen. Anschließend wird der Kontrast durch Berechnung der Standardabweichung aller Werte ermittelt. Vereinfacht dargestellt wird die Struktur durch die Teilung des Eingangssignals durch seine Standardabweichung geteilt, sodass eine genormte Standardabweichung entsteht. (Datta, 2020)

Sobald die drei Werte berechnet wurden, werden diese anhand, im Original-Paper, veröffentlichten mathematischen Funktionen verglichen und schließlich kombiniert. (Datta, 2020)

Obendrein sei es nützlich den Index nicht global auf das Bild anzuwenden. Das Paper gibt an den Algorithmus vorzugsweise auf verschiedene lokale Stellen einzusetzen. Erstens seien für die Analyse interessante Objekte oft instationär und zweitens können Bildverzerrungen auch räumlich variabel sein. Um die Referenz zur Nachahmung der menschlichen Wahrnehmung zu ziehen, ist ebenfalls festzustellen, dass Menschen ebenfalls nur lokale Bereiche einer Grafik in hoher Auflösung wahrnehmen können. (Datta, 2020)

Auch wenn der SSIM durch seine Einfachheit, gegenüber anderen Verfahren, eine überragende Performance aufweist, ist der Algorithmus sehr anfällig für Skalierungen, Verschiebungen und Rotierungen (siehe Abbildung 9). (Zhang & Bull, 2014)

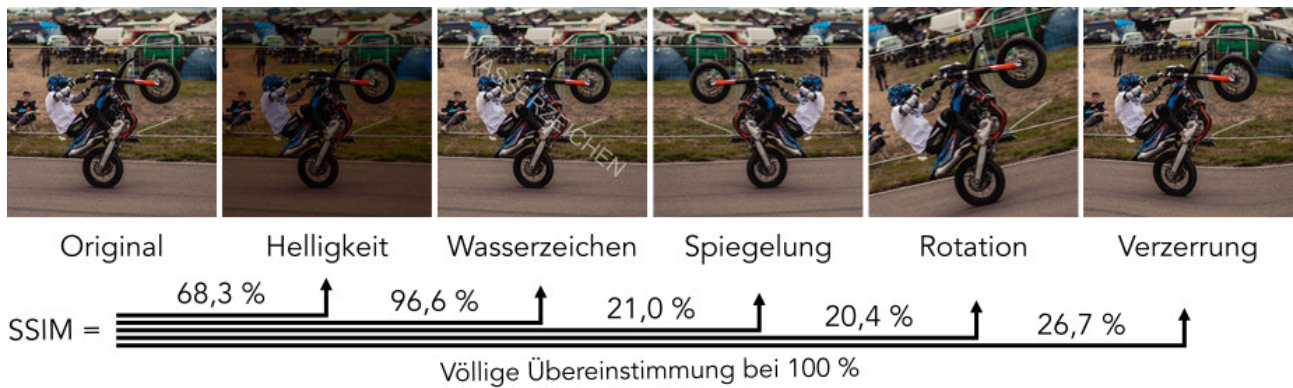


Abbildung 9: SSIM: Anwendung an verschiedenen Testbildern
Quelle: Eigene Darstellung

Aufbauend auf den SSIM-Algorithmus gibt es weitere optimierende Forschungen, wie zum Beispiel der Complex Wavelet Structural Similarity Index. Der CW-SSIM adressiert die vorher genannten Probleme und erzeugt durch Verwendung von Wavelet-Transformationen eine Robustheit gegenüber kleinen Verschiebungen und Rotierungen. (Sampat et al., 2019)

2.6 Siamese Network (Deep Learning)

Zu guter Letzt liefert der Image Matching Algorithmus auf Basis des Siamesischen Neuralen Netzwerks die besten Ergebnisse. Dabei besteht das neuronale Netzwerk aus zwei identischen Subnetzwerken. Meist kommt hierfür das sehr verbreitete Convolutional Neural Network als Subnetzwerk zum Einsatz, welches die Merkmalsvektoren aus den Bildern extrahiert. (Melekhov et al., 2016)

Die beiden entstehenden Vektoren dienen als Eingabe für die Contrastive Loss Function. Dieser Vorgang minimiert die euklidische Distanz der Vektoren im Falle eines Matches und maximiert diese bei inkomparablen Bilderpaaren. (Melekhov et al., 2016)

Wie bei vielen Machine Learning Ansätzen ist bei der Verwendung einer der größten Nachteile das Training. Damit das Siamese Network optimal funktioniert, muss das Netzwerk mit einem sehr großen Datensatz an klassifizierten Bilderpaaren trainiert werden. (Antoniadis, 2022)

3 Fazit und Ausblick

Zusammenfassend bietet dieses Paper einen groben Überblick über den aktuellen Forschungsstand bei der Ähnlichkeitsbestimmung von Bildern. Je nach Größe und Art des Projekts eignen sich manche Algorithmen mehr und manche weniger. Eine allgemeingültige Lösung gibt es bisher nicht.

Der feingranulare Vergleich zwischen Bildern wird auch sicherlich in der Zukunft noch ein wesentlicher Bestandteil der Forschung sein. Neben den in der Einleitung 1 erläuterten Anwendungsfälle spielen Image Matching Algorithmen in der Medizin eine immer größere Rolle. Dabei könnten Krebs und andere schwere Krankheiten schneller erkannt und behandelt werden.

Quellenverzeichnis

- Antoniadis, P. (2022). *Algorithms for Image Comparison*. Verfügbar 4. Dezember 2022 unter <https://www.baeldung.com/cs/image-comparison-algorithm>
- Bay, H., Tuytelaars, T. & Van Gool, L. (2006). SURF: Speeded Up Robust Features (A. Leonardis, H. Bischof & A. Pinz, Hrsg.).
- Christina. (2020). *Determining how similar two images are with Python + Perceptual Hashing*. Verfügbar 4. Dezember 2022 unter <https://lvngd.com/blog/determining-how-similar-two-images-are-python-perceptual-hashing/>
- Datta, P. (2020). *All about Structural Similarity Index (SSIM): Theory + Code in PyTorch*. Verfügbar 4. Dezember 2022 unter <https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e>
- Klinger, E. & Starkweather, D. (2010). *pHash - The open source perceptual hash library*. Verfügbar 4. Dezember 2022 unter <https://www.phash.org/>
- Lee, S. M., Xin, J. H. & Westland, S. (2004). Evaluation of Image Similarity by Histogram Intersection.
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints.
- Melekhov, I., Kannala, J. & Rahtu, E. (2016). Siamese network features for image matching. <https://doi.org/10.1109/ICPR.2016.7899663>
- Monsters, D. (2020). *A Quick Overview of Methods to Measure the Similarity Between Images*. Verfügbar 4. Dezember 2022 unter <https://medium.com/@datamonsters/a-quick-overview-of-methods-to-measure-the-similarity-between-images-f907166694ee>
- Ramos, A. (2022). *Introduction To Perceptual Hashes: Measuring Similarity*. Verfügbar 4. Dezember 2022 unter <https://apiumhub.com/tech-blog-barcelona/introduction-perceptual-hashes-measuring-similarity/>
- Sampat, M. P., Wang, Z., Gupta, S., Bovik, A. C. & Markey, M. K. (2019). Complex Wavelet Structural Similarity: A New Image Similarity Index.
- Simek, K. (2009). *Image comparison - fast algorithm*. Verfügbar 4. Dezember 2022 unter <https://stackoverflow.com/a/844113/17726654>
- Sinha, U. (n.d. a). SIFT: Theory and Practice. Verfügbar 4. Dezember 2022 unter <https://aishack.in/tutorials/sift-scale-invariant-feature-transform-scale-space/>
- Sinha, U. (n.d. b). SIFT: Theory and Practice. Verfügbar 4. Dezember 2022 unter <https://aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoints/>
- Sinha, U. (n.d. c). SIFT: Theory and Practice. Verfügbar 4. Dezember 2022 unter <https://aishack.in/tutorials/sift-scale-invariant-feature-transform-eliminate-low-contrast/>
- Sinha, U. (n.d. d). SIFT: Theory and Practice. Verfügbar 4. Dezember 2022 unter <https://aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/>
- Sinha, U. (n.d. e). SIFT: Theory and Practice. Verfügbar 4. Dezember 2022 unter <https://aishack.in/tutorials/sift-scale-invariant-feature-transform-features/>

- Swain, M. J. & Ballard, D. H. (1991). Color Indexing.
- Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity.
- Zhang, F. & Bull, D. R. (2014). Academic Press Library in signal Processing.