

gbsv Mini-Challenge 2

Alexander
Schilling

10. Januar 2024

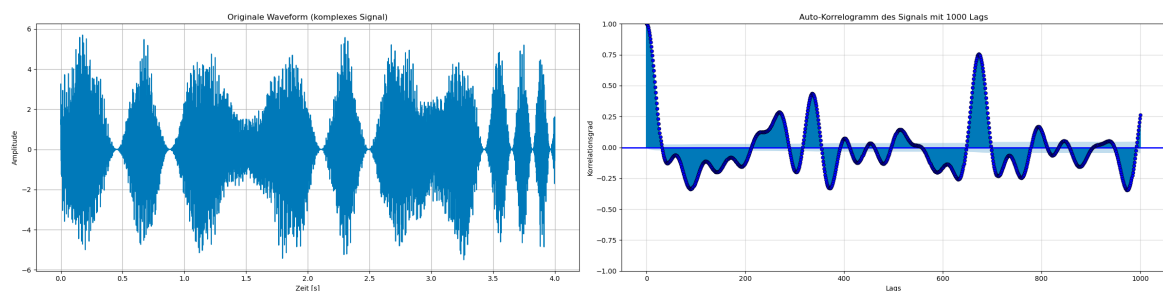
1 Wichtigste Resultate

TASK: Fasse die wichtigsten oder spannendsten Resultate zusammen. Hier kann eine Auswahl getroffen werden. Verwende ggf. Abbildungen. Ca. 300-500 Wörter.

In dieser Mini-Challenge habe ich mich auf die Mustersuche in Bild und Signal konzentriert. Durch die Anwendung verschiedener Techniken der Signalverarbeitung konnte ich interessante Ergebnisse erzielen:

Wiederkehrende Muster von Signalen:

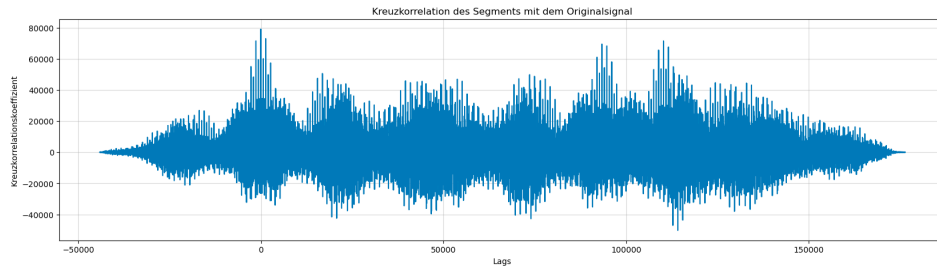
Die Auto-Korrelation misst, wie gut das Signal zu einer verschobenen Version von sich selbst passt und dient dazu, wiederkehrende Muster innerhalb des Signals zu identifizieren. Ich habe hier die Autokorrelation des komplexen Signals berechnet und angezeigt.



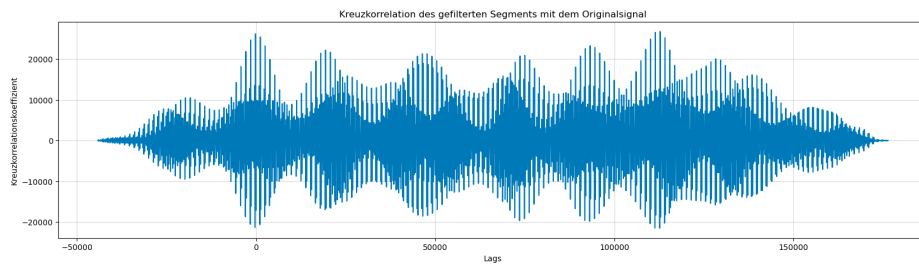
Es zeigt, wie das Signal zu verschiedenen Zeiten mit einer verzögerten Version von sich selbst korreliert. Die scharfen Spitzen im Korrelogramm deuten auf periodische Wiederholungen im Signal hin. Positive Werte nahe bei 1 zeigen eine starke positive Korrelation, während Werte nahe bei -1 eine starke negative Korrelation anzeigen.

Kreuzkorrelationen von Signalen:

Die Anwendung von Kreuzkorrelationen auf verschiedene Signale ermöglichte es mir, wiederkehrende Muster und Strukturen innerhalb der Daten zu identifizieren. Im untenstehenden Bild sieht man eine solche Kreuzkorrelation mit einem Signal und ein ausgeschnittenem Segment dieses Signals:

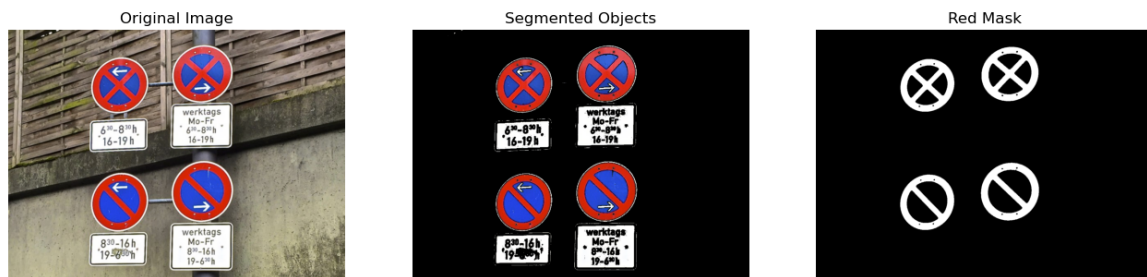


Dies erwies sich als besonders nützlich, um Ähnlichkeiten zwischen verschiedenen Teilen eines Signals zu finden und zu verstehen, wie sich Änderungen in einem Teil des Signals auf andere Teile auswirken. Diese Änderungen waren z.B. das Hinzufügen von Rauschen, eine Amplitudenerhöhung und eine Filterung. Durch das Experimentieren mit unterschiedlichen Lags konnte ich die periodische Natur und die wiederkehrenden Muster in den Signalen meistens klar hervorheben. Beim Experiment mit der Filterung wurde ein Tiefpassfilter angewandt und hier war es schwieriger, Ähnlichkeiten zu finden, da viele wichtige Informationen (hier hohe Frequenzen) verloren gingen. Wenn man das Bild unten mit dem oberen Bild vergleicht, ist dies gut zu sehen.



Segmentierung und Beschriftung von Wiederkehrende Muster (Verkehrsschilder):

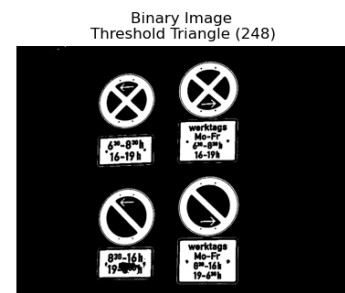
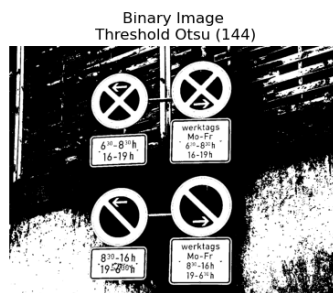
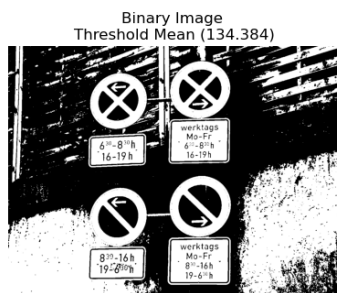
Hier habe ich die Schilder anhand der Farben erfolgreich segmentieren können. Die Visualisierung ganz rechts entspricht Maskierung der roten Farbe, welche ich später dem Originalbild hinzugefügt habe, um schlussendlich eine genauere Segmentierung der Bilder zu schaffen.



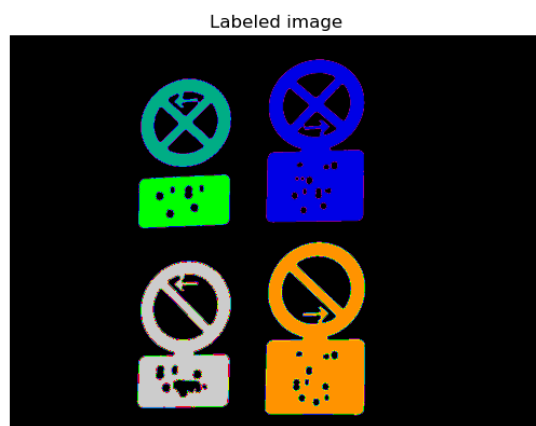
Das ursprüngliche Bild überlagert mit der Roten Maske ist hier zu sehen:



Danach habe ich verschiedene Ansätze zur Segmentierung von den Verkehrsschildern untersucht, wobei die Reduzierung von Hintergrundrauschen eine Schlüsselrolle spielte. Das Experimentieren mit verschiedenen Thresholding-Methoden von «skimage» wie Mean, Otsu und Triangle zeigte, wie empfindlich die Segmentierungsergebnisse auf die Wahl des Schwellenwertes reagieren. Mit diesen Methoden gelingt mir leider entweder eine zu schwache oder zu starke Segmentierung, wie im Bild unten zu sehen:



Durch das sorgfältige Einstellen des manuellen Schwellenwertes gelang es mir, die Schilder effektiver und schöner vom Hintergrund zu trennen, was für die weitere Analyse und Erkennung der Schilder unerlässlich war. Danach ging es darum, kleine Löcher zu füllen und Ränder definierter zu machen mittels morphologische Operationen wie Cleaning und Opening. Schlussendlich konnte ich die 5 verschiedene Schilder beschriften, wie im Bild unten zu sehen:



Keypoint-Erkennung mit ORB-Algorithmus:

Die Anwendung des ORB-Algorithmus zur Keypoint-Erkennung in Bildern ermöglichte es mir, wichtige Merkmale in den Bildern von meinem Lieblingsfussballers zu identifizieren und zu vergleichen.

Bild 1 - Bild 2 (96 Matches für 300 Keypoints) - (0.32 Genauigkeit)



Ein Challenge war es, die Parameter des Algorithmus zu verstehen und die optimalen Anzahl Keypoints auszuwählen. Darum habe ich experimentiert, wie die Auswahl der Parameter die Anzahl und Qualität der detektierten Keypoint-Matches zwischen den Bilder beeinflusst:

Anzahl Keypoints	Bild 1 - Bild 2 (Matches)	Bild 1 - Bild 3 (Matches)	Bild 1 - Bild 4 (Matches)	Bild 2 - Bild 3 (Matches)	Bild 2 - Bild 4 (Matches)	Bild 3 - Bild 4 (Matches)	Durchschnittliche Genauigkeit
25	13	11	7	9	12	10	0.4133
50	25	19	12	19	16	17	0.36
100	40	30	25	30	32	31	0.3133
200	68	41	36	61	57	54	0.2642
300	96	61	65	77	84	70	0.2517
500	122	99	104	127	129	108	0.2297
750	171	166	158	188	168	177	0.2285
1000	226	222	236	227	209	228	0.2247

2 Diskussion

TASK: Diskutiere deine Resultate explizit in Bezug auf deine Aufgaben bzw. Experimente. Mit Diskutieren ist gemeint, Vor- und Nachteile, Chancen und Risiken deines Lösungsweges bzw. der Resultate spezifisch für deine Aufgaben und Experimente zu besprechen. Welche Aufgaben/Fälle konnten gut gelöst werden bzw. welche nicht? Sind die verwendeten Metriken aufschlussreich bzw. brauchbar? Wo gibt es Limitierungen? Wie sind die Resultate im Vergleich zu anderen Arbeiten bzw. deinen Erwartungen? Ca. 300-500 Wörter.

Wiederkehrende Muster von Signalen (Autokorrelogramm):

Da der Lösungsweg hier nicht anspruchsvoll war, erkläre ich einfach einige Vorteile und Nachteile, die ich bei meinen Signalen bemerkt habe oder die bei einem anderen Signal vorkommen könnten:

Vorteile:

- Die Autokorrelation kann dabei helfen, die periodische Natur eines Signals zu identifizieren.
- Sie kann eingesetzt werden, um fundamentale Frequenzen in einem Signal zu finden.
- Die Autokorrelation kann dazu beitragen, zufälliges Rauschen zu identifizieren und zu reduzieren, da Rauschen in der Regel eine geringe Autokorrelation aufweist.

Nachteile:

- Bei Signalen ohne klare Periodizität oder bei rein zufälligen Signalen bietet die Autokorrelation eventuell keine nützlichen Informationen.
- Es könnte sein, dass mehrere überlagerte Frequenzen in einem Signal durch Autokorrelation nicht klar unterschieden werden können.

Kreuzkorrelationen von Signalen:

Der Einsatz von Kreuzkorrelationen zur Erkennung von wiederkehrenden Mustern in Signalen war effektiv. Die Herausforderung bestand jedoch darin, Ähnlichkeiten in gefilterten Signalen zu erkennen, vor allem nach der Anwendung eines Tiefpassfilters, was zu Informationsverlust führte. Dies zeigt die Limitierungen dieser Methode bei der Verarbeitung von stark modifizierten Signalen auf, aber dies macht auch Sinn, wenn das Signal fast nicht mehr identisch klingt wie das Originalsignal. Interessant

Segmentierung und Beschriftung von Wiederkehrenden Mustern (Verkehrsschilder):

Die Farbsegmentierungsmethode, die bei Punkt 1 erwähnt wurde, hat folgende Vor- und Nachteile:

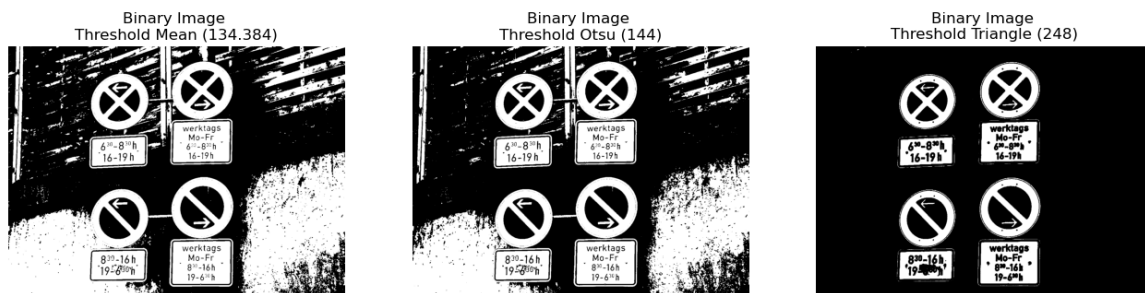
Vorteile:

- Durch die Definition spezifischer Farbbereiche im HSV-Farbraum können bestimmte Objekte, gezielt isoliert werden.
- Die Methode ist relativ einfach zu implementieren und benötigt keine komplexe Vorverarbeitung des Bildes.
- Farbsegmentierung ist in der Regel schnell und effizient, was für Echtzeitanwendungen nützlich sein kann.

Nachteile:

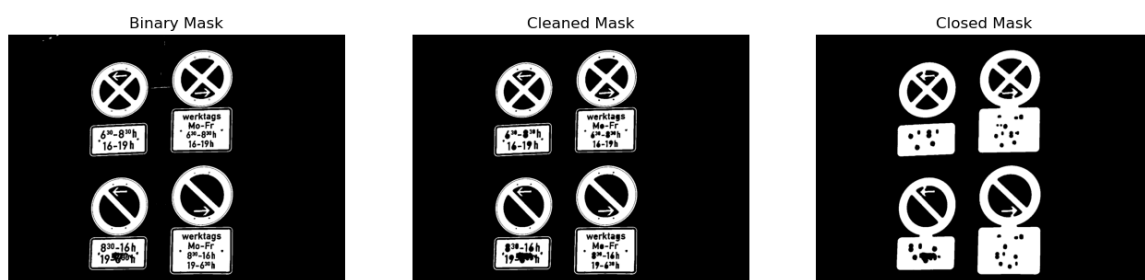
- Die Methode kann bei unterschiedlichen Lichtverhältnissen / qualitativ hohe Bilder mit viel Schatten unzuverlässig sein, da sich die Farbwahrnehmung mit der Beleuchtung ändert.
- Ähnliche Farbtöne können zu falsch-positiven Ergebnissen führen, wenn andere Objekte im Bild ähnliche Farben aufweisen.
- Die Methode ist auf die ausgewählten Farbbereiche beschränkt.

Die Segmentierung mit verschiedenen Thresholding-Methoden von skimage war ein kritischer Punkt und hat am meisten Zeit in Anspruch genommen, da ich recht viel experimentiert habe. Obwohl die manuelle Einstellung des Schwellenwertes zu besseren Ergebnissen führte, zeigten automatisierte Methoden eine hohe Sensitivität, was zu Über- oder Untersegmentierungen führte, wie im Bild unten zu sehen.



Dies ist definitiv ein Nachteil der Algorithmen von skimage und hebt die Notwendigkeit einer präzisen Parameterabstimmung hervor.

Die Kombination von morphologischen Operationen ermöglichte es mir dann die Segmentierung aufzuräumen. Ich habe kleine Objekte oder Rauschen aus der binären Maske entfernt, wobei Objekte kleiner als 150 Pixel ignoriert werden. Anschliessend wurde eine morphologische Schliessung auf der bereinigten Maske durchgeführt, um kleine Löcher innerhalb der Objekte zu füllen. Hier habe ich ein kreisförmiger Strukturelement mit einem Radius von 5 Pixeln verwendet. Dies führte zu grösseren, einheitlichen Formen, welche für die abschliessende Erkennung von den Objekte besser geeignet sind. Die resultierende Masken sind hier zu sehen:



Meine Experimente und mein Weg zur endgültigen Lösung der 5 erkennbare Strassenschilder zeigt, dass es sehr wichtig ist, nicht einfach darauf losschiessen und nur einfach bereits vorgegebene Algorithmen zu verwenden, da dies nicht immer zu den besten Ergebnissen führt. Es ist wichtig, die einzelnen Schritte zu verstehen und zu wissen, wie die Algorithmen funktionieren. Diese Resultate stimmen mit meinen Erwartungen überein und sind genau das, was ich mir zu Beginn dieser Aufgabe vorgestellt habe (eine erfolgreiche Segmentierung der Verkehrsschilder)

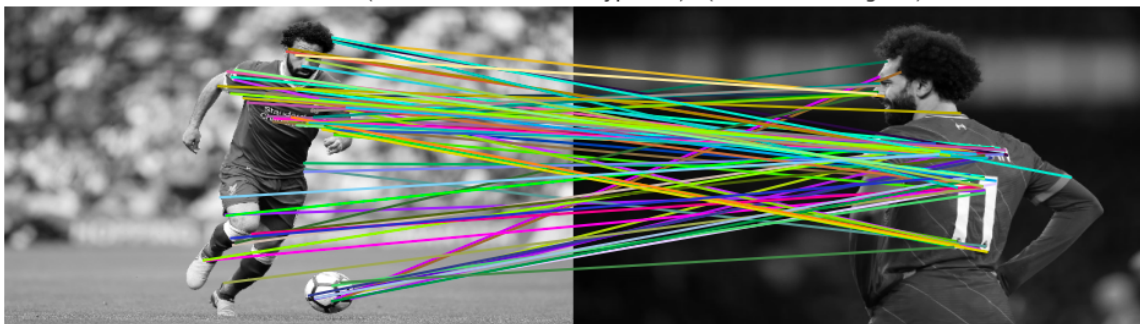
Keypoint-Erkennung mit ORB-Algorithmus:

Die Segmentierung mit verschiedenen Thresholding-Methoden war ein kritischer Punkt und hat am meisten Zeit in Anspruch genommen, da ich recht viel experimentiert habe. Die manuelle Einstellung des Schwellenwerts führte zwar zu besseren Resultaten, doch die automatisierten Methoden von skimage waren sehr empfindlich, was oft zu Über- oder Untersegmentierungen führte. Dies ist definitiv ein Nachteil der Algorithmen von skimage und unterstreicht die Notwendigkeit einer feinen Abstimmung der Parameter.

Die Keypoint-Erkennung mit dem ORB-Algorithmus brachte nicht die erhofften Resultate. Oft basierten die Matches zwischen zwei Bildern auf komplett anderen Keypoints. Dies war enttäuschend und scheint ein Nachteil des ORB-Algorithmus zu sein, der Keypoints auf Basis der Pixelhelligkeit in ihrer Umgebung identifiziert. Ähnliche Helligkeitsbereiche können zu ungenauen Matches führen, was die Effektivität der Methode in bestimmten Szenarien einschränkt und die Wichtigkeit einer umfassenden Validierung der Ergebnisse hervorhebt. In zukünftigen Projekten wäre es notwendig, die Parameteranpassung und -validierung weiter zu verfeinern oder alternative Algorithmen zu betrachten, die möglicherweise weniger anfällig für solche Probleme sind.

Wie im Bild zu sehen, z.B. wird der Ball bei Bild 2 mit dem Stirn des Spielers gematched.

Bild 2 - Bild 3 (77 Matches für 300 Keypoints) - (0.2567 Genauigkeit)



3 Reflexion

TASK: Reflektiert abschliessend. Was ist in dieser Mini-Challenge gut gelaufen? Was würdest du das nächste Mal anders machen? Was würdest du ggf. in der Aufgabenstellung ändern? Ca. 100-150 Wörter.

In dieser Mini-Challenge lief vieles gut. Besonders zufrieden bin ich mit der manuellen Einstellung des Schwellenwertes bei der Bildsegmentierung, die zu den klar identifizierten 5 Verkehrsschilder geführt hat.

Nächstes Mal würde ich mir allerdings mehr Zeit für die Signalverarbeitungsphase nehmen und eine breitere Palette von Veränderungen an den Signalen vornehmen, um die Robustheit der Kreuzkorrelation weiter zu testen. Solche Änderungen wären beispielsweise Zeitliche Dehnung, Phasenverschiebung, Frequenzmodulation. Das würde helfen, ein tieferes Verständnis für die Stabilität dieser Methode unter verschiedenen Bedingungen zu entwickeln. Sonst denke ich, dass ich in dieser Challenge vieles abgedeckt habe.

Für die Aufgabenstellung wäre es noch interessant, mehrere Keypoint-Erkennung Algorithmen zu testen und miteinander zu vergleichen.

4 Code

TASK: Zugang zu aufgeräumtem Git-Repository. Fachexpertinnen Accounts: susuter (GitHub), susanne.suter@fhnw.ch (GitLab)).

<https://github.com/alexschillingfhnw/gbsv>

5 Optional: Lerntagebuch

TASK: Optional darf der Abgabe ein Lerntagebuch beiliegen, welches regelmässig dokumentiert, wie der Lernfortschritt war. Bspw. kurz ein paar Fragen beantworten, analog zu einem Scrum Daily. Was hast du an diesem Tag gemacht? Was ist gelungen? Wo gibt es aktuell Probleme? Wer könnte bei diesen Problemen helfen?