

# gbsv Mini-Challenge 1

Anonymous

3. November 2023

## 1 Wichtigste Resultate

Ein spannendes Ergebnis war das verändern der Helligkeit mit Hilfe einer Gamma Korrektur. Ich habe immer gedacht, der einzige Weg ein Bild aufzuhellen sei durch das gleichmässige Verschieben der Pixelwerte. Das würde allerdings grosse überbelichtete Flecken erzeugen, welche ja dann auch Informationen aus dem Bild verwerfen würden. Die Gamma Korrektur kann aber die meisten der Informationen behalten. Auch wenn man es mit der Aufhellung ein bisschen übertreibt, kann man immer noch fast alle Details erkennen. Interessant finde ich auch wie sich die Korrektur im Histogramm widerspiegelt: Beim Aufhellen werden die weniger intensiven Bins gestreckt (also verteilt; grösserer Abstand dazwischen) und nach rechts geschoben. Die mittleren Bins werden auch nach rechts geschoben und die hellen Bins werden zusammengeschoben.



Abbildung 1: stark aufgehelltes Bild

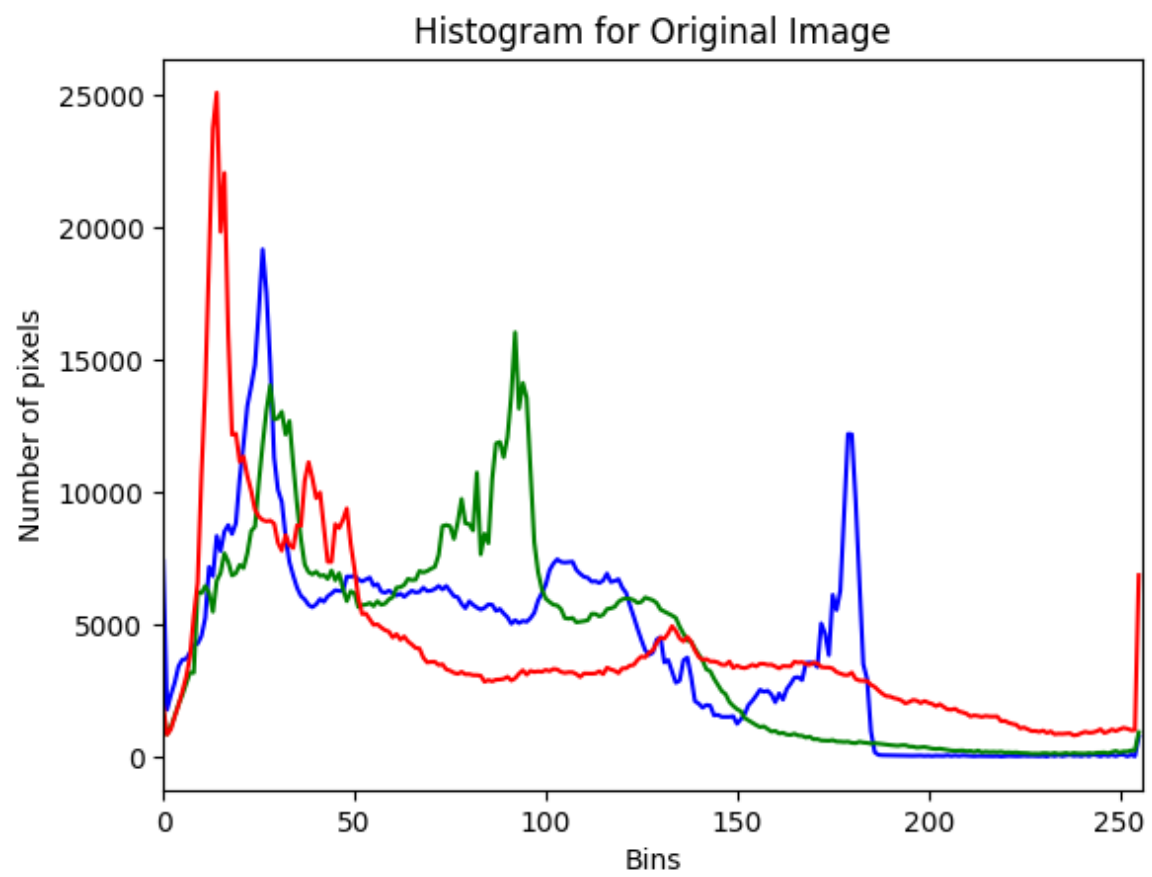


Abbildung 2: Histogramm des Originalbild

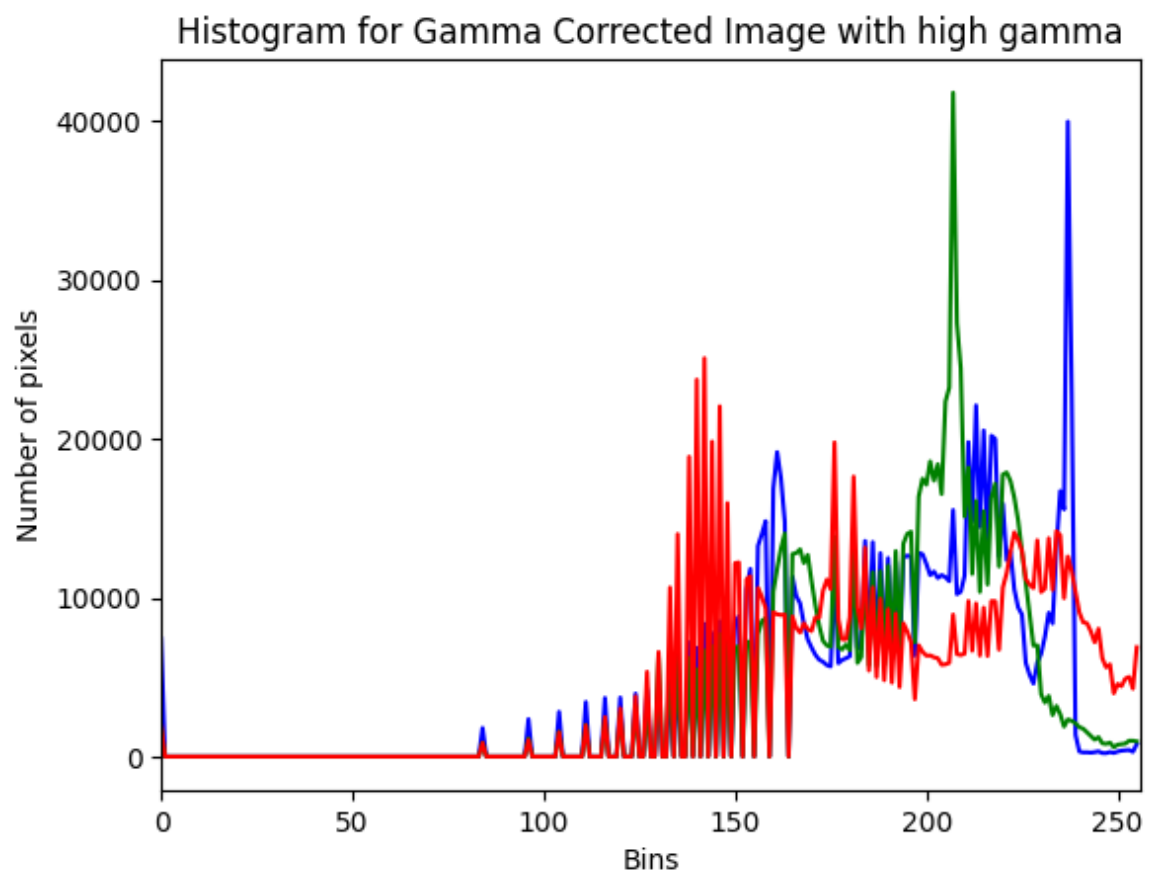


Abbildung 3: Histogramm des stark aufgehellten Bild

Ein weiteres spannendes Experiment fand ich das Transformieren in einen anderen Farbraum und anschliessend in diesem neuen Farbraum eine Veränderung durchzuführen und wieder zurück transformieren. Konkret habe ich ein Bild vom Meer (was viele Pixel mit der Farbe Blau enthält) vom RGB Farbraum, der ein Bild mit den Farben Rot, Grün und Blau darstellt, in den HSV Farbraum transformiert. Der HSV Farbraum stellt sich auch aus drei Komponentne zusammen: Hue (einen Wert zwischen  $0^\circ$  und  $360^\circ$  der zu einer Farbe auf einem Kreis korespondiert), Saturation (einen Wert zwischen 0 und 100 der die Sättigung darstellt (von nicht farbig bis ganz farbig) und dem Value (auch ein Wert zwischen 0 und 100, der etwas von der Helligkeit hat und ein Kontinuum zwischen Schwarz und nicht Schwarz darstellt). Ich habe dann den Hue Wert um  $120^\circ$  gedreht um das mehrheitlich blaue Meer in die Farbe Grün umzuwandeln. Dann habe ich das Bild wieder in den RGB Raum zurück transformiert. Wenn man sich nun die Histogramme des Originalbildes und des veränderten Bildes ansieht, dann stellt man fest, dass sich drei Paare finden lassen, bei denen die Kurven exakt gleich aussehen (Rot wird zu Blau, Blau wird zu Grün und Grün wird zu Rot). Dies ist weil durch das Rotieren der Hue-Wertes die Farben so vertauscht wurden.

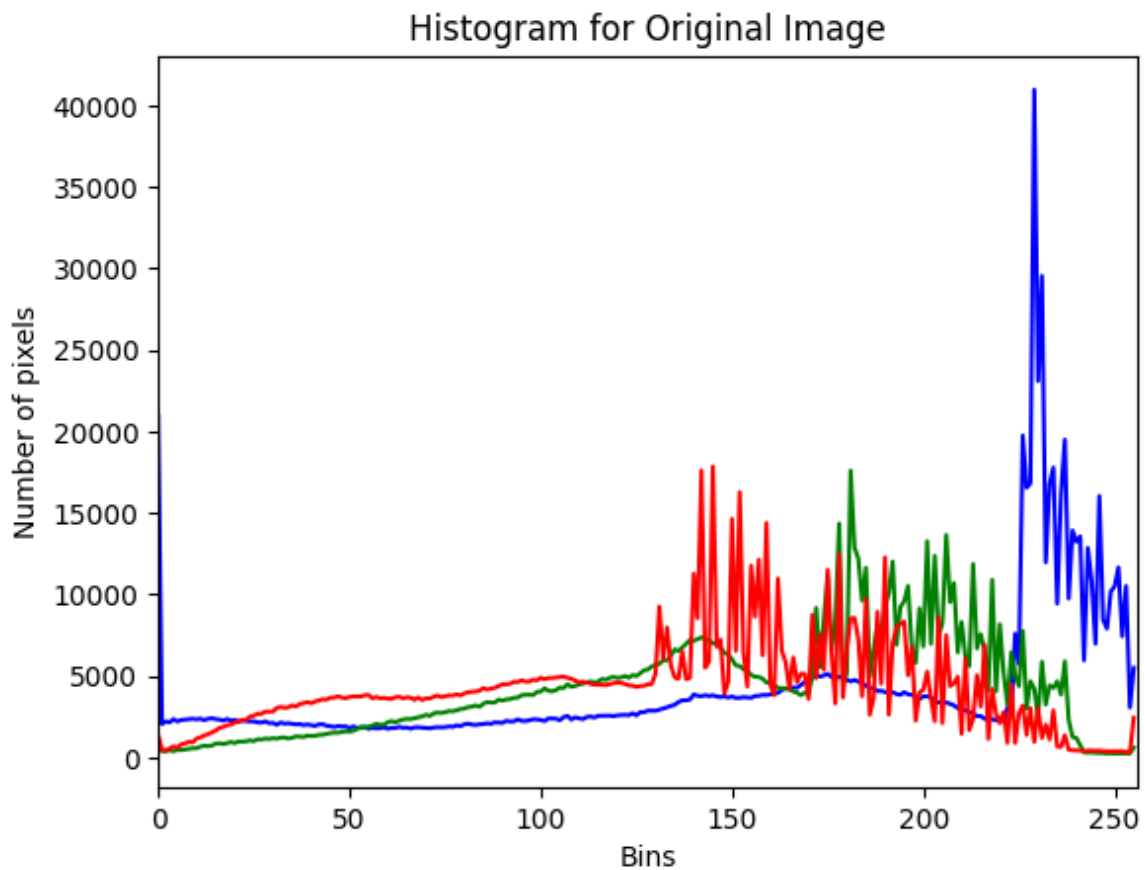


Abbildung 4: Histogram des Bild des Meer

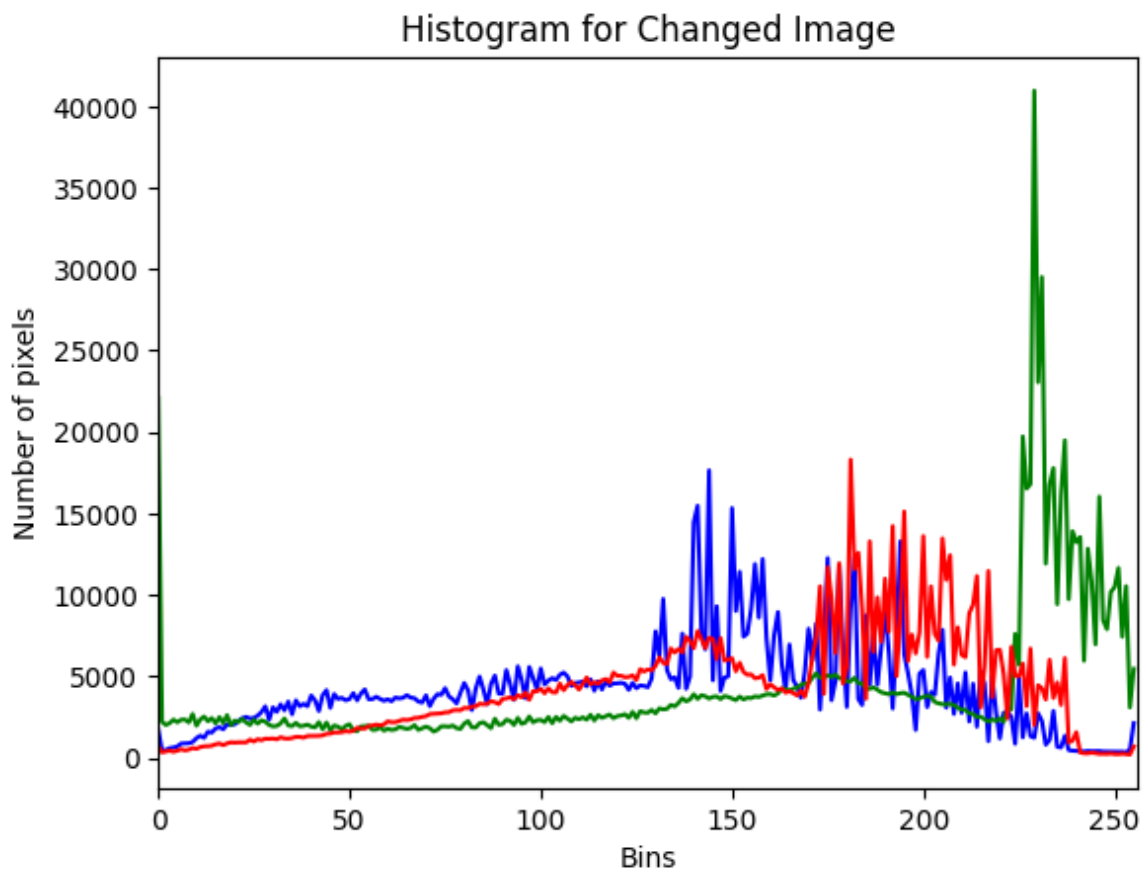


Abbildung 5: Histogram des Bild des Meer nach dem Hue-Shift

Auch ein spannendes Experiment fand ich das erkennen von Linien in einem Bild. Denn das ist etwas, was wir on blossen Auge sehr gut können aber uns nicht direkt vorstellen können wie eine Maschine dies macht. Ich habe das mit dem Hough-Algorithmus gemacht. Und mit ein bisschen herumspielen mit den Parametern konnte ich dann auch nur die wichtigsten Linien der Banden erkennen.

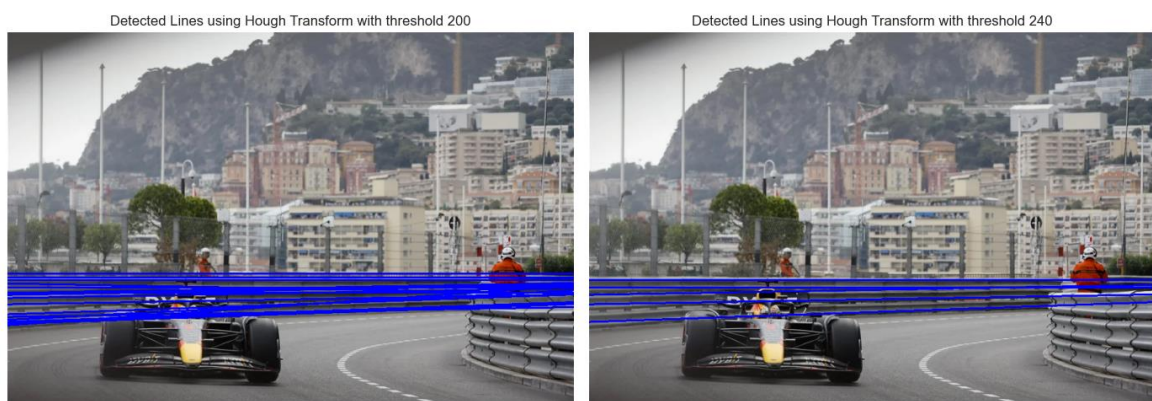


Abbildung 6: Erkannte Linien eines F1-Tracks

## 2 Diskussion

Aufgrund der sehr umfangreichen Mini-Challenge und des begrenzten Platzes habe ich hier auch eine Auswahl getroffen:

- Wertebereiche von Pixeln: Beim Clipping der Werte auf einen bestimmten Wert gehen Daten verloren. Es entstehen weiße und schwarze Flecken. Besser wäre hier das Verschieben mit Auseinanderziehen sowie Zusammenstossen am anderen Ende ähnlich wie das bei der Gamma-Korrektur gemacht wird. Dennoch ist dieser Ansatz einfach und intuitiv.

- Die Gamma-Korrektur ist eine effektive und gute Methode um ein Bild aufzuhellen oder zu verdunkeln. Das Problem ist, dass der Wertebereich an einem Ort stark gestreckt wird und dort nicht mehr viel Informationen enthält. Das kann man umgehen indem man die Farbintensität höher auflöst. 16bit Farbtiefe statt 8bit zum Beispiel. Das muss aber schon beim Aufnehmen des Bildes passieren.
- Beim Konvertieren der Wellenform des Yachthorns nach Rechteckwellen, war von Anfang an das Problem, dass das Audio keine schönen Sinuswellen darstellt. Das hat dann zu Rechteckwellen geführt, die nicht alle die gleiche Frequenz hatten. Besser wäre hier wohl gewesen, dass Audio in die einzelnen Frequenzen zu zerlegen und statt mit Sinuskurven das Ding mit Rechteckwellen wieder zurück transformieren.
- Beim Filtern der Paukenschläge habe ich einfach die tieferen Frequenzen aus dem Spektrogramm gefiltert. Problem ist hier, dass sich mit dem Spektrogramm nicht die Paukenschläge isolieren lassen, sondern lediglich die tiefen Frequenzen. Das Risiko ist, dass man dann auch noch andere Instrumente rausfiltert.
- Die Convolutionfunktionen habe ich mithilfe von For-Loops gelöst. For-Loops sind aber in Python nicht effizient, weshalb hier die bessere Variante gewesen wäre, die einsum Funktion von Numpy zu verwenden. Damit wird die Convolution im viel schnelleren C-Code ausgeführt. An der Laufzeitkomplexität würde das allerdings nichts ändern.
- Bei der Bonusaufgabe habe ich versucht in der räumlichen und spektralen Domäne Rauschen zu filtern. Das ist mir auch gelungen. Allerdings gibt es hier auch einfachere Methoden wie nur in der räumlichen Domäne mit Hilfe eines Gaussian-Kernel das Rauschen rauszufiltern.
- Die Auswahl der Metriken war oft schwierig. Oftmals musste ich mich mit einer Metrik zufrieden geben, die entweder keinen quantitativen Wert ausspuckt (wie das Histogramm oder das Spektrogramm) oder dann weil die Metrik nur einen Unterschied quantifizierte (z.B: Peak Signal to Noise Ratio) und dann ein Referenzwert (was ist jetzt gut, was ist schlecht) schwierig zu ermitteln war. Dort habe ich mich dann daran orientiert, was bei der Bildkomprimierung akzeptabel war und somit ausgemacht, was akzeptable Referenzwerte sind.
- Die Resultate waren zum Teil ernüchternd, weil die Methode/Lösungsweg nicht eine Manipulation auf dem Verständnislevel erlaubt wie wir uns Menschen das vorstellen/produzieren können (z.B: ein Instrument aus der Nationalhymne entfernen). Ansonsten bin ich mit den Resultaten zufrieden, da sie z.B. die menschliche Wahrnehmung verbessern (z.B. Aufhellen, filtern von Muster in der spektralen Domäne).

### 3 Reflexion

Eine Kernherausforderung in dieser Mini Challenge war es die richtigen Bilder/Experimente zu finden und gleichzeitig viel Spass dabei zu haben. Hier sehe ich zwei Wege:

- Bottom-Up: Überlegen, was will ich verändern? welches Bild verwende ich? Wie kann ich das anstellen? Wie messe ich das?
- Top-down: Wie kann ich eine Veränderung im Bild messen? Wie kann ich etwas im Bild verändern? Welches Bild eignet sich um das zu demonstrieren.

Ich habe meistens den zweiten Ansatz verwendet, da ich das Gefühl hatte, dass er ein bisschen schneller und effektiver ist und die Mini-Challenge sehr umfangreich war. Allerdings ging dabei der Spassfaktor ein bisschen unter. Dennoch habe ich mit dem Top-down Ansatz immer noch viel Zeit investiert.

Das schwierigste fand ich immer genau herauszulesen/finden welches Experiment/Metrik jetzt geeignet ist. Ich würde die Metrik weglassen oder optional machen, da nicht immer eine existiert.

## 4 Code

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

## 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?*