

Blatt 1

Louis Franzke 3705548

Aufgabe 1: Farbbilder bearbeiten

```
import matplotlib.pyplot as plt
import numpy as np

an_image = plt.imread('test.png')
# plt.imshow(an_image)

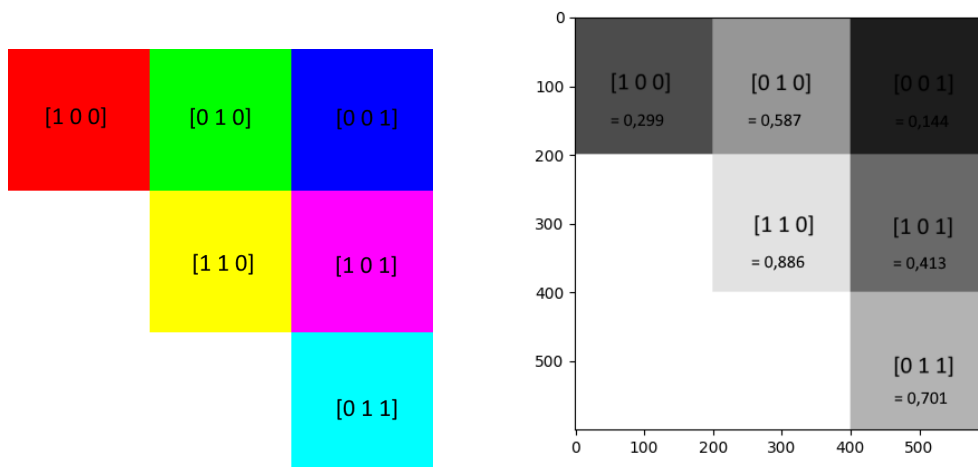
weights = [0.2989, 0.5870, 0.1140]
weights2 = [1.4, 0.5870, 0.1140]

grayscale_image = np.dot(an_image[..., :3], weights)
plt.imshow(grayscale_image, cmap=plt.get_cmap("gray"))
plt.show()
```

a)

Der Code wandelt ein farbiges Bild in ein Graustufenbild mittels Gewichtung der Farbkanäle um. In Zeile 9 wird zunächst die Anzahl der Farbkanäle (tiefste Dimension) auf drei beschränkt, um die Form den Gewichten anzupassen: `an_image[..., :3]` ist bei einem zweidimensionalen Bild dabei identisch zu `an_image[:, :, :3]`.

Aus der Numpy Dokumentation¹ geht hervor, dass wenn das erste Argument ein 3D array und das zweite ein 1-D array, dann wird das Summenprodukt aus den letzten Axen (Dimensionen) gebildet. In diesem Fall bedeutet das, dass die Farbkanäle mit den Gewichten skalar multipliziert werden und somit ein zweidimensionales Array entsteht. Dieses wird in der nächsten Zeile als Graustufen Bild visualisiert.



¹<https://numpy.org/doc/stable/reference/generated/numpy.dot.html>

b)

Verändert man die Gewichte, so ändert sich das Graustufenbild entsprechend. Das heißt, die Gewichte bestimmen, wie stark die einzelnen Farbkanäle zur Bildung der Graustufen beitragen. Insbesondere steht der erste Wert für den roten Farbkanal, der zweite für den grünen und der Dritte für den blauen. An den Randfällen wie $weights = [1.0, 0, 0]$ wird nur ein Farbkanal interpretiert und diejenigen Pixel, die kein rot haben ausgeschwärzt.

Es sollte dabei darauf geachtet werden, dass die Summe der Gewichte 1 nicht übersteigt und keine negativen Gewichte verwendet werden. Sollte dies der Fall sein, so wird das angezeigte Bild dementsprechend angepasst, das bedeutet, dass wenn die Graustufen nun im Bereich $[min, max]$ liegen, min schwarz ist und max weiß.

Aufgabe 2: Interpolieren in Farbräumen

a)

Im RGB Farbraum werden die Farben als additive Mischung der Kanäle für Blau, Grün und Rot beschrieben. Dieser wird häufig in der Bildbearbeitung genutzt, da der Farbraum für Bildschirme verwendet wird. Auch die sechsstelligen Hexadezimalcodes für Farben basieren auf dem RGB-Raum. Im Gegensatz dazu ist das HSV-System auf die menschliche Wahrnehmung der Farbe spezifiziert mit Farbton, Sättigung und Helligkeit. Vorstellen kann man sich dieses Modell als Kegel bei dem die Höhe, die Helligkeit darstellt, die Sättigung durch den Abstand vom Radius definiert wird und der Farbton durch den Winkel im Kreis bestimmt wird. Die Spitze ist somit Helligkeit 0%, Sättigung 0% und stellt schwarz dar. Helligkeit 100% und Sättigung 0% ist reines weiß und befindet sich im Mittelpunkt der Grundfläche.

Die Systeme sind jedoch nicht völlig verschieden und können ineinander umgerechnet werden, wobei die Umrechnung recht komplex ist.

b)

Um in RGB zu interpolieren könnten die drei Farbkanäle separat linear interpoliert werden. Jedoch wirft das Ergebnis oft ungewünschte Farben in der Mitte des Gradienten auf, denn die Wahrnehmung der Farben, insbesondere die Helligkeit und Sättigung, wird in diesem Ansatz nicht berücksichtigt. Ein anderer Ansatz wäre es die Farben zuerst in den HSV Farbraum umzuwandeln um anschließend Farbton (den kleineren Winkelabstand), Sättigung und Helligkeit linear zu interpolieren. Manche argumentieren ², dass diese Interpolation der menschlichen Wahrnehmung dennoch nicht ganz genügt und die Farben deshalb zuerst in die Farbräume HCL oder CIE umgewandelt werden sollten, jedoch kommt es hier auch auf die erforderliche Exaktheit des Kontexts an.

²<https://stackoverflow.com/questions/13488957/interpolate-from-one-color-to-another>