

## 1. Übung

### Farbbilder und Rotation

In dieser Übung sollen Sie sich mit den Python Bibliotheken Pillow und Numpy vertraut machen. Inhalt der Übung ist neben der Einarbeitung in die IDE PyCharme, das Einlesen eines Bildes mithilfe von Pillow und die Implementierung einer ersten eigenen Funktion zur Rotation eines Bildes.

---

1. Starten Sie die PyCharme IDE und legen Sie ein neues Projekt auf dem Z-Laufwerk an.
2. Lesen Sie ein Bild mit Hilfe von Pillow ein und lassen Sie sich das Bild anzeigen.
3. Untersuchen Sie den Datentyp des eingelesenen Bildes und konvertieren Sie es zu einem Numpy Array. Wie ist ein Numpy Array aufgebaut?
4. Lassen Sie sich die drei Farbkanaäle eines Bildes getrennt anzeigen.
5. Implementieren Sie eine Funktion, die ein eingelesenes Bild wahlweise horizontal oder vertikal spiegelt. (Verwenden Sie hierfür keine Numpy Funktionen.)

### Abgabe

Die Aufgaben werden per Mail an [tkocher@htwg-konstanz.de](mailto:tkocher@htwg-konstanz.de) vor der nächsten Übungsstunde abgegeben. Außerdem werden die Lösungen nächstes Mal mündlich präsentiert.

### Bild Referenzen

<http://weburbanist.com/2011/09/21/incredible-rgb-art-shifts-as-lighting-colors-change/>  
<http://www.manuela-kerkhoff.de/hidden-in-the-woods-rgb-technique/>  
<http://100-beste-plakate.de/plakate/ikea-rgb-billboard/>

```

0 from PIL import Image, ImageOps
1
2 import numpy as np
3
4
5 def print_info(img):
6     print(img.format)
7     print(img.mode)
8     print(img.size)
9     print(img.width)
10    print(img.height)
11    print(img.palette)
12    print(img.info)
13
14
15 def load_image(filename):
16     img = Image.open(filename)
17     img.load()
18
19     return img
20
21
22 def make_array(img):
23     return np.array(img)
24
25
26 def split_channels(np_array):
27     np_array_red = np_array.copy()
28     np_array_blue = np_array.copy()
29     np_array_green = np_array.copy()
30
31     np_array_blue[:, :, 0] *= 0
32     np_array_blue[:, :, 1] *= 0
33     blue = Image.fromarray(np_array_blue)
34
35     np_array_green[:, :, 1] *= 0
36     np_array_green[:, :, 2] *= 0
37     green = Image.fromarray(np_array_green)
38
39     np_array_red[:, :, 0] *= 0
40     np_array_red[:, :, 2] *= 0

```

```

41 red = Image.fromarray(np_array_red)
42
43 return blue, green, red
44
45
46 def transform(img, mode):
47     if mode is "vertical":
48         return ImageOps.flip(img)
49     elif mode is "horizontal":
50         return ImageOps.mirror(img)
51     else:
52         return img
53
54 if __name__ == "__main__":
55     # 2.Bild einlesen
56     image = load_image("hidden.png")
57
58     # 3.Bild zu Numpy-Array konvertieren
59     array = make_array(image)
60
61     # 4. Bild nach Farb-Kanälen getrennt ausgeben
62     blue, green, red = split_channels(array)
63
64     # 5.Funktion zum vertikal/horizontal spiegeln
65     t_image = transform(image, "vertical")
66
67     t_image.show()
68     blue.show()
69     green.show()
70     red.show()

```