

Aufgabenblatt 1

Einführung in die Bildverarbeitung

Christian Wilms und Simone Frintrop
SoSe 2020

Ausgabe: 24. April 2020 - **Abgabe bis:** 8. Mai 2020, 10:00

abgegeben am 6. Mai 2020 von:

Abdulssatar Khateb (6976879), Merle Hoffmann (7031673), Felix Swimmer (7162123)

Aufgabe 1 — NumPy und Matplotlib Tutorial

1. Wie kann die Anzahl der Zeilen und Spalten eines NumPy-Arrays bestimmt werden?

Mit `a.shape` wird ein Tupel zurückgegeben, in dem die Anzahl der Zeilen und Spalten des Arrays `a` stehen.

Beispiel:

```
import numpy as np

a = np.array([[1, 2, 3], [4, 5, 6]])
print(a.shape)      # Schreibt (Zeilen, Spalten) auf die Konsole
print(a.shape[0])   # Schreibt die Anzahl der Zeilen auf die Konsole
print(a.shape[1])   # Schreibt die Anzahl der Spalten auf die Konsole
```

Konsole:

```
(2, 3)
2
3
```

2. Was unterscheidet die Funktionen `numpy.array` und `numpy.zeros`?

Mit `numpy.array` wird aus einer übergebenen Liste ein Array erzeugt. Die Elemente der Liste werden ins Array überführt, wobei weitere Listen in der Liste die Zeilen darstellen.

Mit `numpy.zeros` wird aus einem übergebenen Tupel, welches die Anzahl der Zeilen und Spalten darstellt, ein Array erzeugt. Das Array hat die selbe Anzahl an Zeilen und Spalten, wie im Tupel angegeben und als Elemente nur Nullen.

Beispiel:

```
import numpy as np

a = np.array([[1, 2, 3], [4, 5, 6]])
print(a)      # Schreibt das Array a auf die Konsole
b = np.zeros((2, 3))
print(b)      # Schreibt das Array b auf die Konsole
```

Konsole:

```
[[1 2 3]
 [4 5 6]]
[[0. 0. 0.]
 [0. 0. 0.]]
```

3. Wie kann der Datentyp eines gegebenen NumPy-Arrays ermittelt werden?

Mit `a.dtype` wird der Datentyp des Arrays `a` zurückgegeben.

Beispiel:

```
import numpy as np

a = np.array([1, 2, 3])
print(f'{a} hat den Datentyp {a.dtype}')
b = np.array([1, 2, 3], dtype = np.float64)
print(f'{b} hat den Datentyp {b.dtype}')
```

Konsole:

```
[1 2 3] hat den Datentyp int64
[1. 2. 3.] hat den Datentyp float64
```

4. Was bedeutet `.T` hinter dem Variablennamen eines NumPy-Arrays? Beispielsweise: `a.T`

Das `T` steht für Transponieren. Mit `a.T` wird das Array `a` transponiert.

Bedeutet: Die Zeilen des Arrays werden zu Spalten und die Spalten zu Zeilen.

Beispiel:

```
import numpy as np

a = np.array([[1, 2, 3], [4, 5, 6]])
print(f'{a} \n wird transponiert zu \n {a.T}')
```

Konsole:

```
[[1 2 3]
 [4 5 6]]
wird transponiert zu
[[1 4]
 [2 5]
 [3 6]]
```

5. Was machen die Funktionen `xlabel` und `ylabel` aus `matplotlib.pyplot`?

Mit der Funktion `xlabel` lässt sich die x-Achse und mit `ylabel` die y-Achse einer zweidimensionalen Darstellung beschriften. Die Beschriftung der x-Achse steht unterhalb und die Beschriftung der y-Achse links neben der Darstellung.

Aufgabe 2 — Erste Schritte in NumPy

Siehe *blatt01_2.py*

Aufgabe 3 — Basisoperationen mit Bildern

Siehe *blatt01_3.py*

Hinweis: `plt.imshow`-Funktionen sind auskommentiert.

Aufgabe 4 — Fortgeschrittene Bearbeitung von Bildern

Siehe *blatt01_4.py*

Hinweis: `plt.imshow`-Funktionen sind auskommentiert.

Aufgabe 5 — Broadcasting vs. Schleifen in NumPy

Siehe *blatt01_5.py*

Aufgabe 6 — Bildstatistiken in NumPy

Siehe *blatt01_6.py*

Hinweis: `print`-Funktionen sind auskommentiert.