

Aufgabenblatt 1

Übung zu Einführung in die Bildverarbeitung

Laszlo Korte, Thang Le, Juri Schalück, Maris Hillemann, Tim Rolff, Christian Wilms
SoSe 2023

Ausgabe: 21. April 2023 - **Abgabe bis: 1. Mai 2023, 10:00**

Abgabe per Moodle!

Juri Schalück (Gruppe 2)	0schalue@informatik.uni-hamburg.de
Thang Le (Gruppe 3)	phuoc.thang.le@uni-hamburg.de
Laszlo Korte (Gruppe 4)	9korte@informatik.uni-hamburg.de
Maris Hillemann (Gruppe 5, 6)	maris.nathanael.hillemann@studium.uni-hamburg.de
Tim Rolff (Gruppe 1)	tim.rolff@uni-hamburg.de
Christian Wilms (Gruppe 1)	christian.wilms@uni-hamburg.de

Das Vorrechnen dieser Aufgaben in der nächsten Übung (1./2. Mai) zählt noch **nicht** für die Scheinkriterien, da es sich noch nicht um BV-Inhalte handelt. Die Punkte des Aufgabenblattes zählen schon für die Scheinkriterien.

Aufgabe 1 — Python und NumPy Tutorial - $5+5+5+5 = 20$ Punkte - Theorieaufgabe

Geht das kurze Tutorial¹ von Justin Johnson zu Python und NumPy durch. Bearbeitet dabei nur die Teile zu Python und NumPy (bis Numpy Documentation). Testet anschließend euer Wissen und beantwortet folgende Fragen kurz schriftlich.

1. Wie kann man den vierten bis zehnten Wert einer Liste in einem Befehl setzen?
2. Welche Möglichkeiten gibt es in einer `for`-Schleife über eine Liste auch den Index zu bekommen? Nenne zwei Möglichkeiten.
3. Wie kann die Anzahl der Zeilen und Spalten eines zweidimensionalen NumPy-Arrays bestimmt werden?
4. Was unterscheidet die Funktionen `numpy.array` und `numpy.full`?

Aufgabe 2 — Codewort dechiffrieren - $3+5+5+4+3=20$ Punkte - Programmieraufgabe

Ihr habt ein merkwürdiges Codewort bekommen: toarmumiknif. Dechiffriert dieses nun mit eurem möglicherweise ersten Pythonskript, indem ihr dieser Anleitung folgt. Wichtig: Erzeugt nur einmal eine Liste explizit, die weiteren Listen sollen aus dieser initialen Liste extrahiert werden bzw. aus den Einzelteilen wieder zusammengefügt werden.

1. Erzeugt euch die Nachricht als Liste aus 12 einzelnen Buchstaben jeweils als String.
2. Zerlegt die Liste in drei Listen. Die erste Liste `a` soll die ersten 5 Elemente bestehen. Die Liste `b` soll die nächsten beiden Elemente und die Liste `c` die letzten 5 Elemente beinhalten.
3. Fügt nun die Listen `c` und `a` in dieser Reihenfolge zusammen. Extrahiert anschließend zwei Listen aus dem Ergebnis: Die Liste `d` mit den Elementen an den geraden Indizes (0, 2, ...) und die Liste `e` mit den Elementen an den ungeraden Indizes (1, 3, ...).
4. Dreht nun die Liste `e`.
5. Fügt zuletzt die Liste `d` und die gedrehte Liste `e` sowie die Liste `b` in dieser Reihung zusammen. Wie lautet das Codewort?

¹<https://cs231n.github.io/python-numpy-tutorial/> ↗

Aufgabe 3 — Informationen zu Listen - 5+5+8+8+14=40 Punkte - Programmieraufgabe

Wir wollen nun in Vorbereitung auf die Verarbeitung von Bildern Informationen über Listen in Python ermitteln. Zur Lösung dieser Aufgabe sind **keine** NumPy-Funktionen erlaubt. Außerdem ist die Benutzung der Funktion `sum()` ebenfalls nicht erlaubt.

1. Erzeugt zunächst eine Liste `l` mit den 99 Werten von 101 bis einschließlich 199. Nutzt dazu die Funktion `range()`.
2. Erstellt nun eine Funktion, welche die Summe über alle Elemente einer Liste ermittelt und zurückgibt. Ruft diese Funktion mit der Liste `l` auf. Zurückgeben heißt hierbei immer, dass `return` benutzt werden soll und nicht `print()`.
3. Schreibt nun eine Funktion, die den Mittelwert (arithmetische Mittel) einer Liste zurückgibt und ruft diese mit der Liste `l` auf.
4. Nun wollen wir die Anzahl der durch 5 teilbaren Elemente einer Liste bestimmen. Schreibt dazu eine entsprechende Funktion, die eine Liste nimmt und die Anzahl zurückgibt. Ruft die Funktion mit der Liste `l` auf.
Tipp: Der Modulo-Operator in Python lautet `%`.
5. Zuletzt wollen wir die Anzahl der Primzahlen in einer Liste feststellen. Schreibt auch dazu eine entsprechende Funktion, die eine Liste nimmt und die Anzahl der Primzahlen zurückgibt. Wie viele Primzahlen befinden sich in der Liste `l`?

Aufgabe 4 — Erste Schritte in NumPy - 3+3+4+3+3+4=20 Punkte - Programmieraufgabe

Erlaubte (Sub-)Pakete: `numpy`

In dieser Aufgabe sollen Matrizen und Vektoren aus NumPy-Arrays erzeugt und manipuliert werden. Alle Schritte sollen in Python programmiert und in einem Python-Skript festgehalten werden. Im Rahmen dieser Aufgabe ist nur die Nutzung von NumPy erlaubt.

1. Erzeugt zunächst einen Vektor `u` als 1D-Array mit 42 Fünfen.
2. Erzeugt zudem einen Vektor `v` als 1D-Array aus der Liste: `[0,1,2,3,4,5,6,7,8,9]`. Welchen Elementtyp hat das Array Ergebnis?
3. Formt den Vektor `v` nun in eine Matrix `m` (2D-Array) um, die aus fünf Zeilen und zwei Spalten besteht.
Tipp: NumPy kennt eine Funktion, um die Form (engl. *shape*) eines Arrays zu ändern.
4. Multipliziert alle Einträge der Matrix `m` mit dem konstanten Faktor 3.1. Welchen Elementtyp hat das Ergebnis?
5. Ändert nun den Elementtyp der Matrix `m` auf `np.int`. Tipp: Nutzt die Methode `astype` eines NumPy-Arrays.
6. Berechnet das Hadamard-Produkt² (elementweise Multiplikation) der Matrix `m` mit sich selbst (`m⊙m`).

²<https://de.wikipedia.org/wiki/Hadamard-Produkt> ↗