



# Will it blend?! Integration digitaler Tools online und in Präsenz

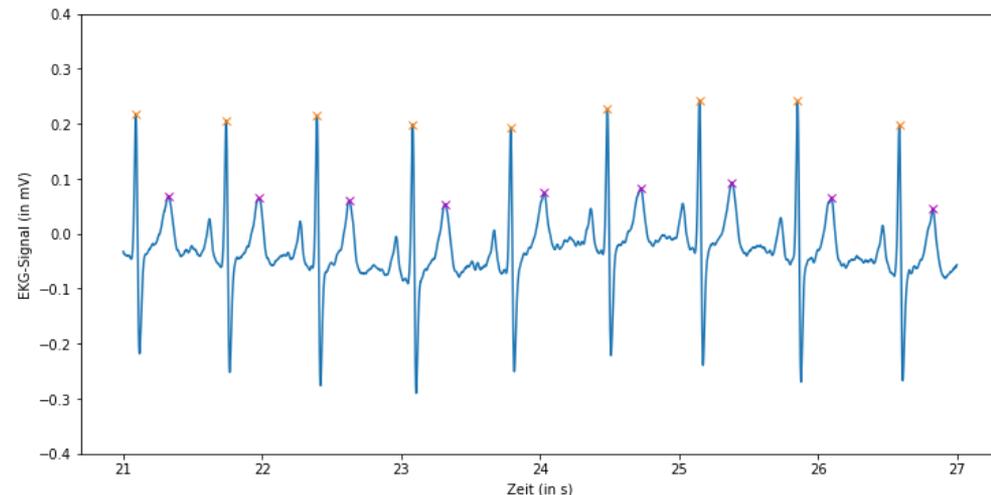
E-Learning Tag 2023

# Ausgangssituation

Seminar im Nebenfach Medical Data Science:  
„Analyse medizinischer Daten und Signale“

Komplementär zu Vorlesung / Messpraktikum  
(Herkunft von Daten)

Greenfield Projekt



# Problemstellung

Vorwissen (Erstsemester)

Thematische Fülle

Komplexe theoretische Grundlagen

(unklare) pandemische Lage

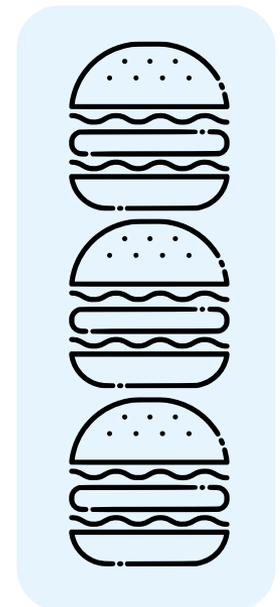
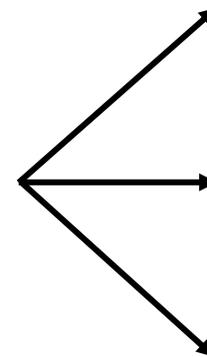
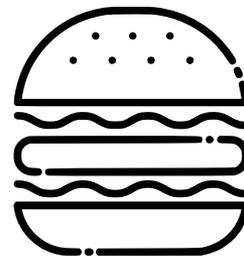
# Veranstaltungskonzept – Aufbau

Themenblöcke ~ 30min

klar kommunizierte Lernziele

Sandwichprinzip

- Input / Impulsvortrag
- **Aktivierung**
- Auswertung / Diskussion
- ...



# Jupyter to the Rescue!

## Jupyter Notebooks kombinieren

- Text
- Programmcode
- Ausgaben

## Ausführung im Browser

- Lokal im PC-Pool
- Remote via Server (+ Zoom)





# Jupyter Notebooks

Das aktuelle Feld beinhaltet formatierten Text. Das folgende Feld beinhaltet Code. Dieser kann direkt ausgeführt werden. Die Ausgabe erscheint darunter.

```
[1]: # Hier steht code; Kommentare beginnen mit "#"
var = 42

print("Dies ist Beispieltext.\nDie Antwort auf die endgültige Frage",
      "nach dem Leben, dem Universum und dem ganzen Rest lautet:", var)
```

Dies ist Beispieltext.

Die Antwort auf die endgültige Frage nach dem Leben, dem Universum und dem ganzen Rest lautet: 42

## Weitere Überschrift

```
[3]: import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0,10, 200)
plt.plot(x, np.sin(x))
plt.title("Grafiken sind toll!")
plt.show()
```



# Veranstaltungskonzept – Integration

Impulsvorträge

Synchrone Aktivierung

- Miniaufgaben, Lückentext, ausprobieren, ...
- Intensive Betreuung

Asynchrone Aktivierung

- Hausaufgaben

Abschlussaufgabe

- Projekt + Ausarbeitung



# Beispiel: synchron (ausprobieren)



## Short Term Fourier Transformation

Eine etwas robustere Variante zum ausprobieren

Lassen Sie sich nicht verunsichern. Der Code ist nur zu Funktionen zusammengefasst und es werden Jupyter-Interna verwendet, dass sich die Variablenänderung angenehmer ausprobieren lässt.

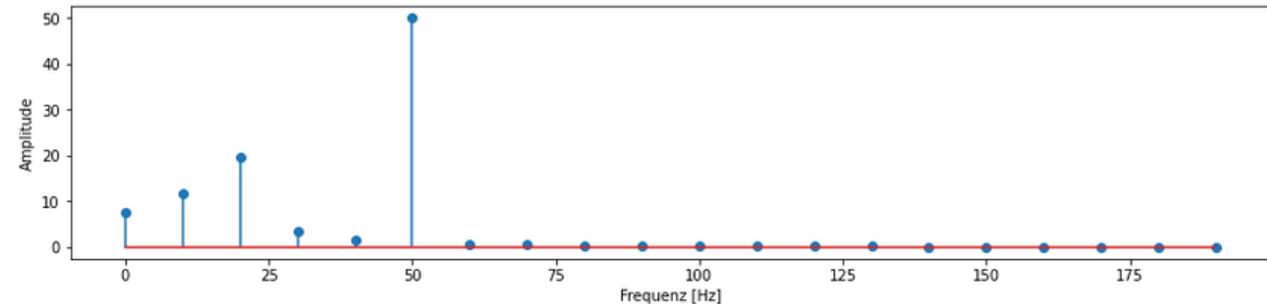
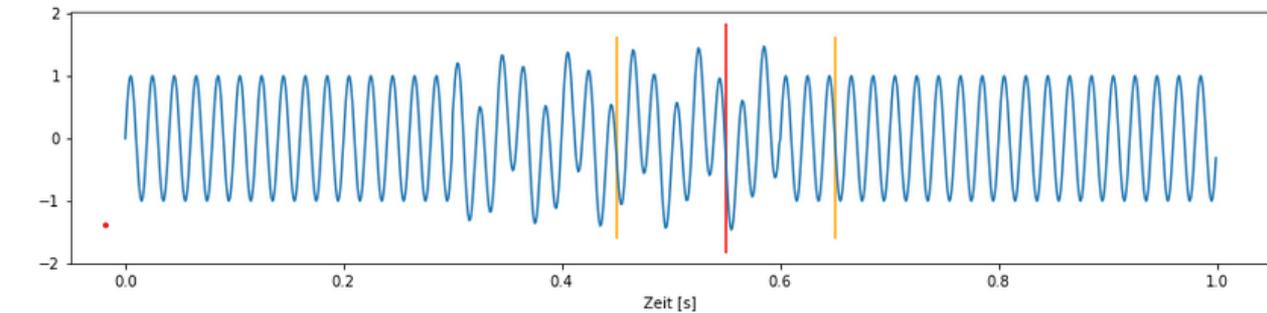
...

pos  550

win\_width  100

f\_abtast  1000

freq\_max\_idx  20



# Beispiel: synchron

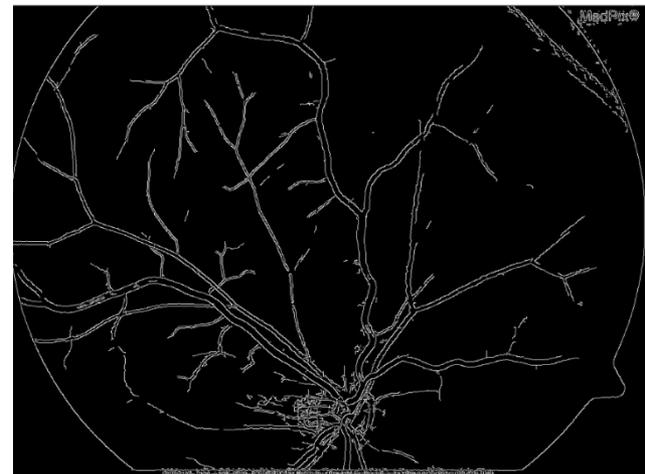
```
jupyter [ ]: img = cv.imread('retina_1.jpg',0)
plt.imshow(img, cmap='gray', vmin=0, vmax=255)
plt.show()

[ ]: ## TODO: Apply the canny edge detector to the retina image "img"

## START

edges =
## END

plt.imshow(edges, cmap='gray', vmin=0, vmax=255)
```



# Beispiel: Asynchron (Hausaufgabe)



This exercise shows the connection between element-wise products in Fourier space and convolutions in image space. Moreover, it visualises the fact that convolutions of images with certain masks correspond to lowpass or highpass filtering. Remember that the lower frequencies are represented around the origin (center) in the frequency domain, while the higher frequencies lie close to the edges.

```
[ ]: import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

x = cv.getGaussianKernel(3,0.5)
gaussian = x*x.T

sobel_x= np.array([[1, 0, -1],
                  [2, 0, -2],
                  [1, 0, -1]])

sobel_y= np.array([[1, 2, 1],
                  [0, 0, 0],
                  [-1, -2, -1]])

laplacian=np.array([[0,1,0],
                  [1,-4,1],
                  [0,1,0]])

filters = [gaussian, laplacian, sobel_x, sobel_y]
filter_name = ['gaussian', 'laplacian', 'sobel_x', 'sobel_y']

## Compute the 2D Fourier transform of all 4 filters (for the shape parameter "s" of numpy.fft.fft
## Afterwards, create a list "mag_spectrum" that contains the 4 resulting arrays in the same order
## by the list "filters"
## Try to interpret the outputs (Which frequency ranges are enhanced/diminished by the filters?)

## START

mag_spectrum =
## STOP
```

## Lessons Learned

Konzept funktioniert!

Studierende neigen dazu

- den kürzesten Weg zu nehmen
- Hausaufgaben zu „vergessen“

Gute Infrastruktur des Rechenzentrums

Curse of Knowledge

# Vielen Dank für die Aufmerksamkeit!

Fragen?