Code zu Auswertung MRT

July 4, 2024

[ ]:
```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
cmap = mpl.colormaps.get_cmap('tab10').colors
plt.style.use('../l3.mplstyle')
import pydicom
```

1 SNR

[ ]:
```
with pydicom.dcmread('C:/Users/steph/Downloads/MRT/T1WEIGHTED_IMAGE_TSE_0002/902.
 ↪MR.PRAKTIKUM_SCAN_PROTOCOL.0002.0011.2024.06.26.11.29.54.49682.156687664.IMA')␣
 ↪as image:
    arr = image.pixel_array
    # Use same vmin, vmax for low and high SNR image for better comparison
    vmin, vmax = image.SmallestImagePixelValue, image.LargestImagePixelValue
# Determine SNR:
x_on, y_on = 153, 217
x_off, y_off = 10,10
width = 80
roi_on = arr[y_on:y_on+width, x_on:x_on+width]
roi_off = arr[y_off:y_off+width, x_off:x_off+width]
snr = np.mean(roi_on)/np.mean(roi_off)
print(f'SNR: {snr}')
# Plotting
fig,ax = plt.subplots()
im = ax.imshow(arr, cmap='gray', vmin=vmin, vmax=vmax)
rect_on = plt.Rectangle((x_on,y_on),width,width, edgecolor=cmap[0],␣
 ↪facecolor='none')
rect_off = plt.Rectangle((x_off,y_off),width,width, edgecolor=cmap[1],␣
 ↪facecolor='none')
ax.add_patch(rect_on)
ax.add_patch(rect_off)
ax.set(xticks=[], yticks=[])
cbar = fig.colorbar(im, ax=ax)
cbar.set_label('Intensität [ADU]')
fig.tight_layout()
fig.savefig('plots/t1_high_snr.pdf', bbox_inches='tight')
```

```
# ------------------------------------ Low SNR␣
↪-------------------------------------

with pydicom.dcmread('C:/Users/steph/Downloads/MRT/T1_WEIGHTED_LOWRES_0003/902.
↪MR.PRAKTIKUM_SCAN_PROTOCOL.0003.0011.2024.06.26.11.29.54.49682.156689714.IMA')␣
↪as image:
    arr = image.pixel_array
# Determine SNR:
x_on, y_on = 210, 310
x_off, y_off = 10,10
width = 100
roi_on = arr[y_on:y_on+width, x_on:x_on+width]
roi_off = arr[y_off:y_off+width, x_off:x_off+width]
snr = np.mean(roi_on)/np.mean(roi_off)
print(f'SNR: {snr}')
# Plotting
fig,ax = plt.subplots()
im = ax.imshow(arr, cmap='gray', vmin=vmin, vmax=vmax)
rect_on = plt.Rectangle((x_on,y_on),width,width, edgecolor=cmap[0],␣
↪facecolor='none')
rect_off = plt.Rectangle((x_off,y_off),width,width, edgecolor=cmap[1],␣
↪facecolor='none')
ax.add_patch(rect_on)
ax.add_patch(rect_off)
ax.set(xticks=[], yticks=[])
cbar = fig.colorbar(im, ax=ax)
cbar.set_label('Intensität [ADU]')
fig.tight_layout()
fig.savefig('plots/t1_low_snr.pdf', bbox_inches='tight')
```

2 T1 vs T2

[ ]:
```
with pydicom.dcmread('C:/Users/steph/Downloads/MRT/T1WEIGHTED_IMAGE_TSE_0002/902.
↪MR.PRAKTIKUM_SCAN_PROTOCOL.0002.0011.2024.06.26.11.29.54.49682.156687664.IMA')␣
↪as image_t1:
    arr_t1 = image_t1.pixel_array
# Determine Michelson-contrast:
x_bright, y_bright = 150, 220
x_dark, y_dark = 210, 330
width = 80
roi_bright = arr_t1[y_bright:y_bright+width, x_bright:x_bright+width]
roi_dark = arr_t1[y_dark:y_dark+width, x_dark:x_dark+width]
contrast = (np.mean(roi_bright)-np.mean(roi_dark))/(np.mean(roi_bright)+np.
↪mean(roi_dark))
print(f'Contrast t1: {contrast}')
```

```
# Plotting:
fig,ax = plt.subplots()
im_t1 = ax.imshow(arr_t1, cmap='gray')
ax.set(xticks=[], yticks=[])
cbar = fig.colorbar(im_t1, ax=ax)
cbar.set_label('Intensität [ADU]')
# Plot rects for contrast
rect_bright = plt.Rectangle((x_bright,y_bright),width,width, edgecolor=cmap[0],␣
 ↪facecolor='none')
rect_dark = plt.Rectangle((x_dark,y_dark),width,width, edgecolor=cmap[1],␣
 ↪facecolor='none')
ax.add_patch(rect_bright)
ax.add_patch(rect_dark)
# coordinate system
ax.arrow(10,501, 80,-80, color='red', head_width=10, head_length=10)
ax.text(100, 440, 'cor', color='red', ha='center', va='center', rotation=45)
ax.arrow(10,501, 0,-140, color='red', head_width=10, head_length=10)
ax.text(35, 380, 'trans', color='red', ha='center', va='center', rotation=90)
ax.arrow(10,501, 140,0, color='red', head_width=10, head_length=10)
ax.text(140, 475, 'sag', color='red', ha='center', va='center')
fig.tight_layout()
fig.savefig('plots/t1.pdf', bbox_inches='tight')

# ------------------------------------------------ T2 weighted image␣
 ↪------------------------------------------------

with pydicom.dcmread('C:/Users/steph/Downloads/MRT/T2_WEIGTHED_0004\902.MR.
 ↪PRAKTIKUM_SCAN_PROTOCOL.0004.0011.2024.06.26.11.29.54.49682.156691663.IMA') as␣
 ↪image_t2:
    arr_t2 = image_t2.pixel_array
# Determine Michelson-contrast:
x_bright, y_bright = 45,86
x_dark, y_dark = 26,44
width = 28
roi_bright = arr_t2[y_bright:y_bright+width, x_bright:x_bright+width]
roi_dark = arr_t2[y_dark:y_dark+width, x_dark:x_dark+width]
contrast = (np.mean(roi_bright)-np.mean(roi_dark))/(np.mean(roi_bright)+np.
 ↪mean(roi_dark))
print(f'Contrast t2: {contrast}')

# Plotting:
fig,ax = plt.subplots()
im_t1 = ax.imshow(arr_t2, cmap='gray')
ax.set(xticks=[], yticks=[])
cbar = fig.colorbar(im_t1, ax=ax)
cbar.set_label('Intensität [ADU]')
```

```
# plot rects for contrast
rect_bright = plt.Rectangle((x_bright,y_bright),width,width, edgecolor=cmap[0],␣
 ↪facecolor='none')
rect_dark = plt.Rectangle((x_dark,y_dark),width,width, edgecolor=cmap[1],␣
 ↪facecolor='none')
ax.add_patch(rect_bright)
ax.add_patch(rect_dark)
ax.set(xticks=[], yticks=[])
# coordinate system
ax.arrow(2.5,125, 20,-20, color='red', head_width=2.5, head_length=2.5)
ax.text(25, 110, 'cor', color='red', ha='center', va='center', rotation=45)
ax.arrow(2.5,125, 0,-35, color='red', head_width=2.5, head_length=2.5)
ax.text(9, 95, 'trans', color='red', ha='center', va='center', rotation=90)
ax.arrow(2.5,125, 35,0, color='red', head_width=2.5, head_length=2.5)
ax.text(35, 118, 'sag', color='red', ha='center', va='center')
fig.tight_layout()
fig.savefig('plots/t2.pdf', bbox_inches='tight')
```

3 Artefacts

[ ]:
```
with pydicom.dcmread('C:/Users/steph/Downloads/MRT/
 ↪T1WEIGHTED_IMAGE_ARTIFACT_0005\902.MR.PRAKTIKUM_SCAN_PROTOCOL.0005.0008.2024.
 ↪06.26.11.29.54.49682.156693796.IMA') as image:
    arr = image.pixel_array
fig,ax = plt.subplots()
im = ax.imshow(arr, cmap='gray')
ax.set(xticks=[], yticks=[])
cbar = fig.colorbar(im, ax=ax)
cbar.set_label('Intensität [ADU]')
fig.tight_layout()
fig.savefig('plots/artifact.pdf', bbox_inches='tight')
```

4 Vegetables

[ ]:
```
with pydicom.dcmread('C:/Users/steph/Downloads/MRT/T1WEIGHTED_IMAGE_TSE_0017/902.
 ↪MR.PRAKTIKUM_SCAN_PROTOCOL.0017.0012.2024.06.26.11.29.54.49682.156724687.IMA')␣
 ↪as image:
    arr = image.pixel_array
fig,ax = plt.subplots()
im = ax.imshow(arr, cmap='gray')
ax.set(xticks=[], yticks=[])
cbar = fig.colorbar(im, ax=ax)
cbar.set_label('Intensität [ADU]')
# Coordinate system
ax.arrow(10,501, 80,-80, color='red', head_width=10, head_length=10)
```

```
ax.text(100, 440, 'sag', color='red', ha='center', va='center', rotation=45)
ax.arrow(10,501, 0,-140, color='red', head_width=10, head_length=10)
ax.text(35, 370, 'cor', color='red', ha='center', va='center', rotation=90)
ax.arrow(10,501, 140,0, color='red', head_width=10, head_length=10)
ax.text(140, 475, 'trans', color='red', ha='center', va='center')
fig.tight_layout()
fig.savefig('plots/veggie.pdf', bbox_inches='tight')
```

[ ]:

[ ]: