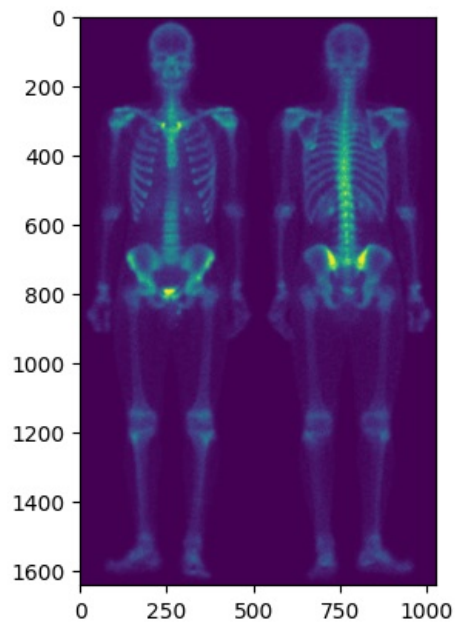


```
In [1]: import matplotlib.pyplot as plt
import tiff as tiff
import numpy as np
import os
```

```
In [ ]: # Załadowanie pliku .tiff
if os.name == 'nt':
    file_name = "./src/"+input("Podaj nazwe pliku z danymi: ")
elif os.name == 'posix':
    file_name = "./src//"+input("Podaj nazwe pliku z danymi: ")
else:
    print("Nieznany system")
img = tiff.imread(file_name)
```

```
In [ ]: #Wyświetlenie załadowanego obrazu
plt.figure()
plt.imshow(img)

plt.show()
```



```
In [ ]: if len(img.shape)==3:
    img = np.mean(img,axis=2).astype(np.uint8)
mode = input("Podaj :")

if( mode == "pozioma"):
    line_num = int(input("Podaj: "))
    gray_val = img[line_num,:]

elif(mode == "pionowa"):
    line_num = int(input("Podaj: "))
    gray_val = img[:,line_num]

else:
    print("Eee")

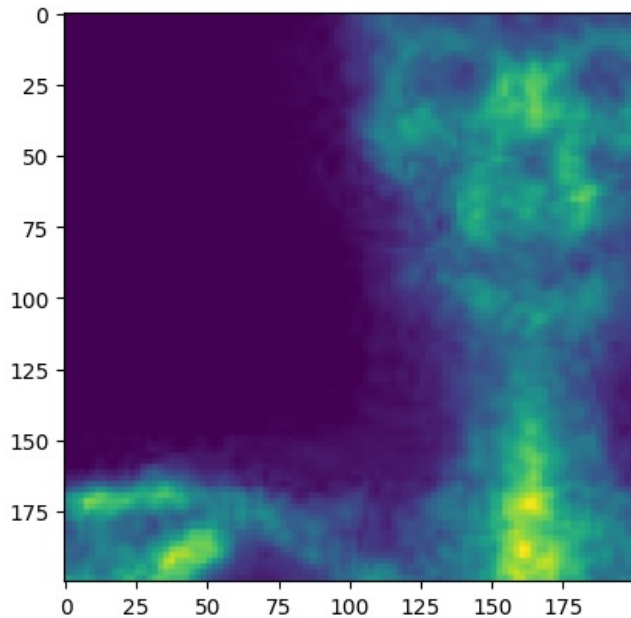
plt.figure()
plt.plot(gray_val)
plt.title('Histogram skali szarości')
plt.xlabel('Wartość pikseli')
plt.ylabel('Częstość')
plt.show()
```



```
In [12]: try:
    x, y, w, h = map(int, input("Podaj współrzędne X, Y, szerokość i wysokość prostokąta (oddzielone spacjami):
    if w <= 0 or h <= 0:
        print("Szerokość i wysokość muszą być dodatnie.")
except ValueError:
    print("To nie są liczby całkowite. Proszę spróbować ponownie.")
region_of_intrest = img[y:y+h , x:x+w]

plt.figure()
plt.imshow(region_of_intrest)

plt.show()
```



```
In [35]: import matplotlib.pyplot as plt
import tifffile as tiff
import numpy as np
import os
```

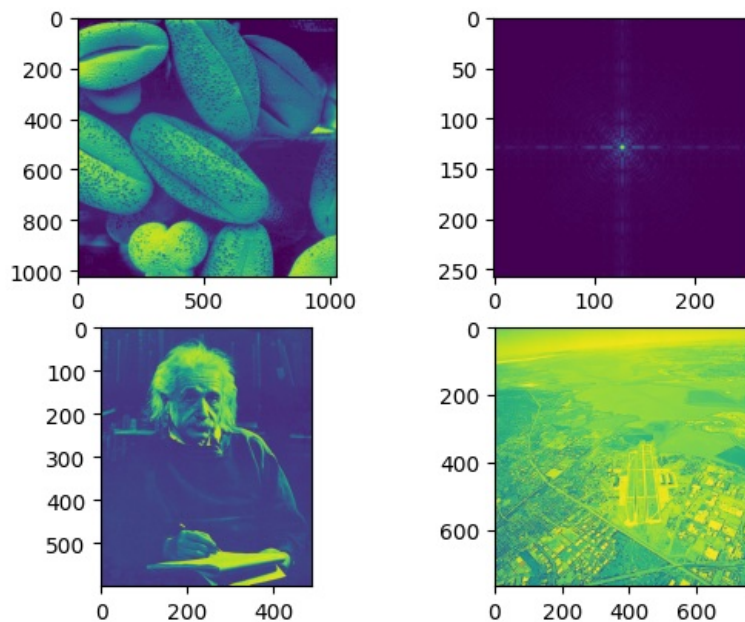
```
In [36]: # Załadowanie pliku .tiff
img_a = tiff.imread("src/pollen-dark.tif")
img_b = tiff.imread("src/spectrum.tif")
img_c = tiff.imread("src/einstein-low-contrast.tif")
img_d = tiff.imread("src/aerial_view.tif")
```

```
In [37]: #Wyświetlenie załadowanego obrazu
plt.figure()
plt.subplot(2,2,1)
plt.imshow(img_a)

plt.subplot(2,2,2)
plt.imshow(img_b)

plt.subplot(2,2,3)
plt.imshow(img_c)

plt.subplot(2,2,4)
plt.imshow(img_d)
plt.show()
```



```
In [39]: m = 0.45
e = 8
gamma = 0.2
plt.figure()
#pollen-dark
plt.subplot(2,2,1)
c = int(input("Podaj"))
img_a = img_a*c
plt.imshow(img_a)
#spectrum
plt.subplot(2,2,2)
img_b = c * np.log(1+img_b)
plt.imshow(img_b)
#einstein-low-contrast
plt.subplot(2,2,3)
img_c = 1/(1+pow(m/img_c,e))
plt.imshow(img_c)
#aerial-view
plt.subplot(2,2,4)
if(gamma>0 and c>0):
    img_d = c * pow(img_d,gamma)
    plt.imshow(img_d)

plt.show()
```

/tmp/ipykernel_21695/1819212428.py:12: RuntimeWarning: invalid value encountered in log
img_b = c * np.log(1+img_b)

