

Krzysztof Krupicki – raport lab 8

1. Wyszukaj **obraz** w formacie jpg lub png, który jest obrazem medycznym w odcieniach szarości. Sprawdź tryb i przekonwertuj do trybu ‘L’.

```
im = Image.open('rentgen.png')
t_im = np.array(im)
r, g, b = im.split()
szary = r
print(f'im.mode: {im.mode}')
im_L = im.convert('L')
print(f"im_L.convert('L').mode: {im_L.mode}")
```

2. Wypisz statystyki tego obrazu i skomentuj słownie te statystyki. Wyświetl histogram (zrzut ekranu wklej do raportu) i skomentuj słownie odnosząc się również do statystyk.

```
def statystyki(im):
    s = stat.Stat(im)
    print("extrema ", s.extrema) # max i min
    print("count ", s.count) # zlicza
    print("mean ", s.mean) # średnia
    print("median ", s.median) # mediana
    print("stddev ", s.stddev) # odchylenie standardowe

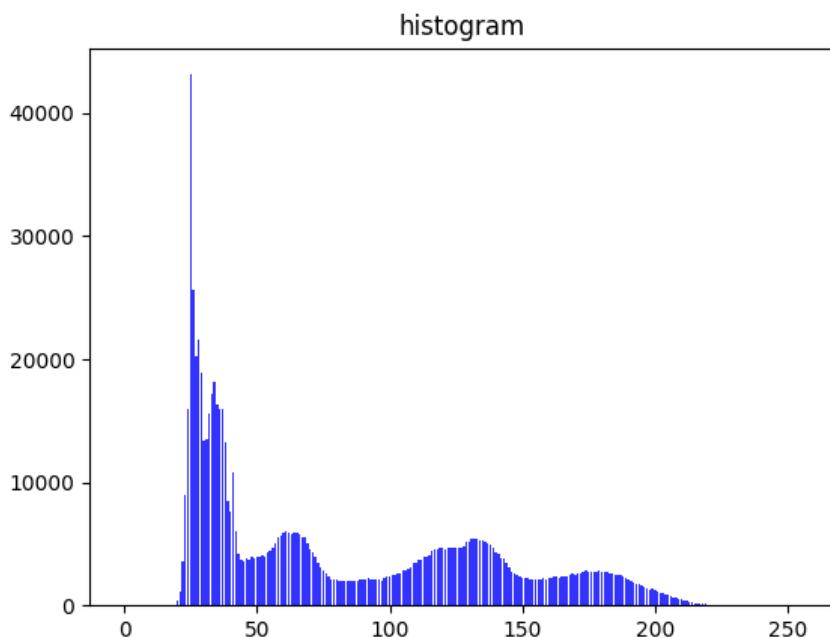
    print('===== obraz oryginal =====')
    statystyki(im)
    print('===== obraz convert L =====')
    statystyki(im_L)

hist = im_L.histogram()
plt.title("histogram")
plt.bar(range(256), hist[:,], color='b', alpha=0.8)
plt.savefig('hist.png')
```

===== obraz oryginal =====
extrema [(19, 242), (19, 242), (19, 242)]
count [861120, 861120, 861120]
mean [82.58639330174657, 82.58639330174657, 82.58639330174657]
median [64, 64, 64]
stddev [54.37005925188605, 54.37005925188605, 54.37005925188605]
===== obraz convert L =====
extrema [(19, 242)]
count [861120]
mean [82.58639330174657]
median [64]
stddev [54.37005925188605]

Obraz nie korzysta z pełnego zakresu odcieni szarości, minimalna wartość to 19, a maksymalna to 242. Średnia wartość pikseli wynosi 82.59 co mówi, że obraz jest stosunkowo ciemny. Mediana wynosi 64, co oznacza, że połowa pikseli ma wartość poniżej 64.

Odchylenie standardowe wynosi 54.37, co wskazuje na umiarkowaną różnorodność wartości pikseli w obrazie.

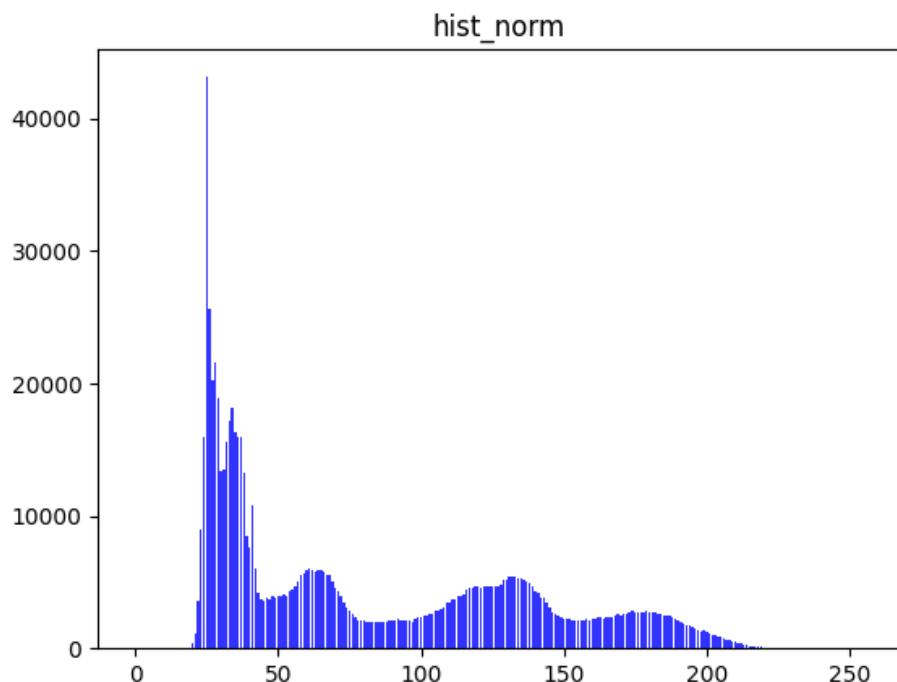


Krzysztof Krupicki – raport lab 8

3. Napisz program `histogram_norm(obraz)`, który na wyjściu daje histogram znormalizowany obrazu. Zastosuj tę funkcję do obrazu i przedstaw histogram w postaci diagramu plt a następnie wklej do raportu.

```
def histogram_norm(obraz):
    hist = obraz.histogram()
    pixels = sum(hist)
    hist_norm = [i / pixels for i in hist]
    plt.title("hist_norm")
    plt.bar(range(256), hist_norm, color='b', alpha=0.8)
    plt.savefig('hist_norm.png')

histogram_norm(im_L)
```

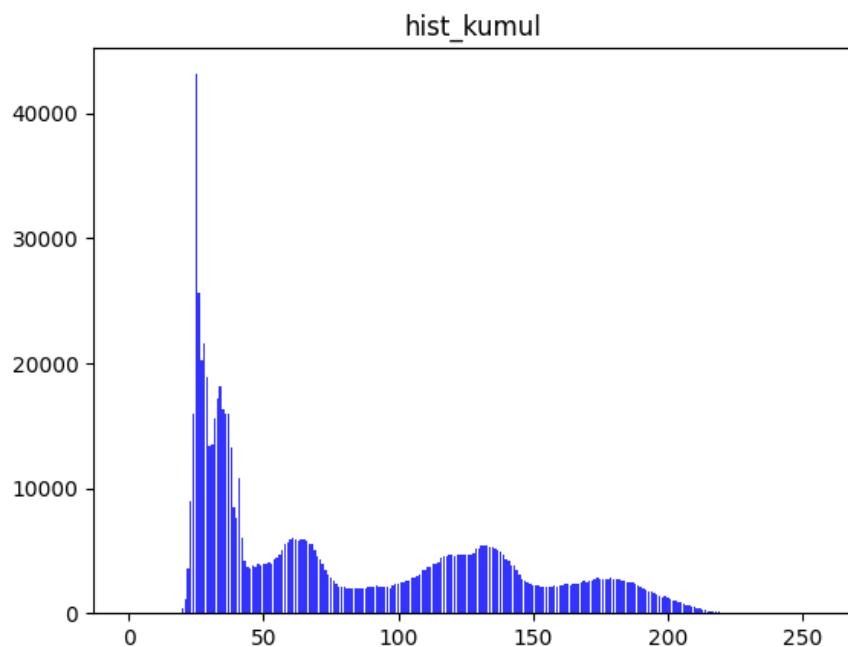


Krzysztof Krupicki – raport lab 8

4. Napisz program `histogram_kumul(obraz)`, który na wyjściu daje histogram skumulowany obrazu. Zastosuj tę funkcję do obrazu i przedstaw histogram w postaci diagramu plt a następnie wklej do raportu.

```
def histogram_kumul(obraz):
    hist = obraz.histogram()
    pixels = sum(hist)
    hist_norm = [i / pixels for i in hist]
    hist_kumul = [sum(hist_norm[:i]) for i in range(256)]
    plt.title("hist_kumul")
    plt.bar(range(256), hist_kumul, color='b', alpha=0.8)
    plt.savefig('hist_kumul.png')

histogram_kumul(im_L)
```



Krzysztof Krupicki – raport lab 8

5. Napisz program `histogram_equalization(obraz)`, który na wyjściu daje obraz powstały po wyrównaniu histogramu obrazu. Zastosuj tę funkcję do obrazu `obraz` i zapisz jako obraz `equalized.png`

```
def histogram_equalization(obraz):
    hist = obraz.histogram()
    pixels = sum(hist)
    hist_norm = [i / pixels for i in hist]
    hist_kumul = [sum(hist_norm[:i]) for i in range(256)]
    hist_equal = [int(255*hist_kumul[p]) for p in range(256)]
    t_obraz = np.array(obraz)
    t_eq = np.zeros(t_obraz.shape, dtype=np.uint8)
    for i in range(256):
        t_eq[t_obraz == i] = hist_equal[i]

    return Image.fromarray(t_eq)

equalized = histogram_equalization(im_L)
equalized.save('equalized.png')
```

6. Zastosuj metodę `ImageOps.equalize` do obrazu `obraz`, zapisz obraz jako `equalized1.png`.

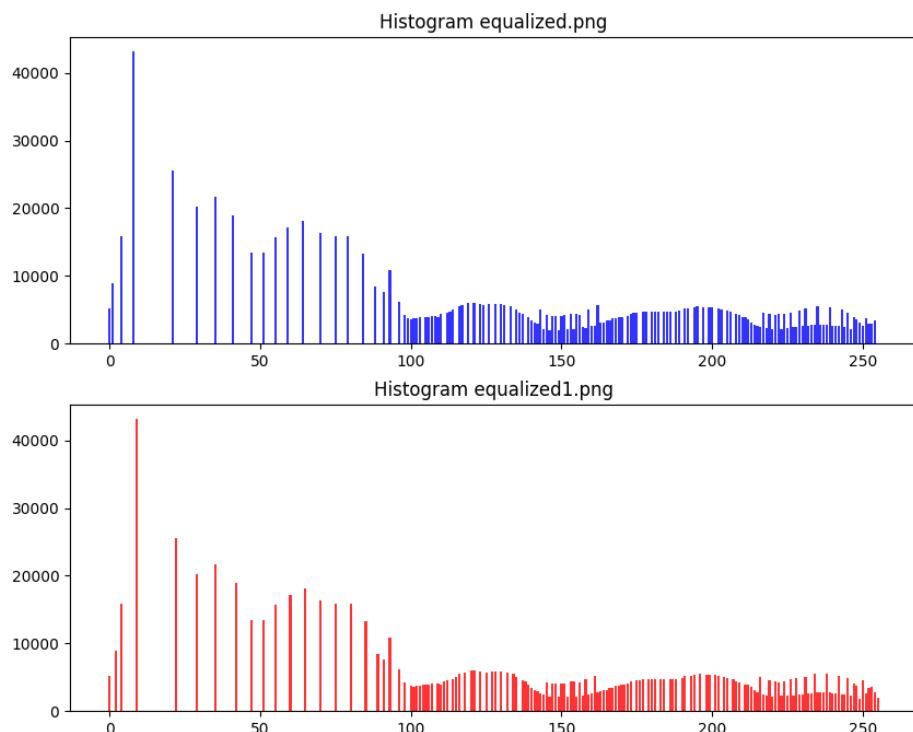
```
equalized1 = ImageOps.equalize(im_L)
equalized1.save('equalized1.png')
```

Następnie

- porównaj obrazy z pkt 5. i 6. i skomentuj

Odpowiedź: Obrazy są do siebie bardzo podobne, gołym okiem trudno zauważyc różnice.

- przedstaw histogramy na jednym diagramie plt, wklej do raportu i skomentuj różnice



Odpowiedź: Histogramy są do siebie bardzo podobne, widać bardzo niewielkie różnice.

Krzysztof Krupicki – raport lab 8

- c. pobierz statystyki obu obrazów i skomentuj różnice

```
print('===== equalized.png =====')
statystyki(equalized)

print('===== equalized1.png =====')
statystyki(equalized1)

===== equalized.png =====
extrema [(0, 254)]
count [861120]
mean [125.5111436269045]
median [126]
stddev [74.7452035962131]
===== equalized1.png =====
extrema [(0, 255)]
count [861120]
mean [126.0698160535117]
median [127]
stddev [74.7701200991852]
```

Odpowiedź: Statystyki obu obrazów są do siebie bardzo zbliżone. Różnica na ekstremach wynosi 1, wartość średnia różni się o 0.5, mediana o 1, a odchylenie standardowe o 0.025. Co potwierdza, że obrazy są bardzo podobne, niemalże identyczne.