



Zadanie 2 -Histogramy

Otwarto: środa, 6 marca 2024, 00:00

Polecenie wstępne. Wczytywanie gotowych obrazów

W katalogu instalacyjnym Scikit-image, przykładowo

```
c:\Python\Lib\site-packages\skimage\data\
```

są dostępne przykładowe obrazy. Obrazy te można czytać przy pomocy instrukcji biblioteki scikit image, zob. tutaj:

<http://scikit-image.org/docs/stable/api/skimage.data.html>

W laboratorium będziemy korzystać z tych obrazów.

ZADANIE 1

a) Wczytaj czarno-biały obraz *camera* i korzystając z polecenia *np.histogram* wykonaj histogram tego obrazu dla 256 "kubelków". Przedział dla kubelków [0:257). Histogram zaprezentuj na wykresie *matplotlib* jako linię (użyj *plt.plot*). Obok zaprezentuj obraz.

Uwaga: istnieją funkcje tworzenia histogramów w bibliotece scikit-image, moduł exposure, lecz nie pozwalają na określenie wartości minimalnej i maksymalnej dla kubelków.

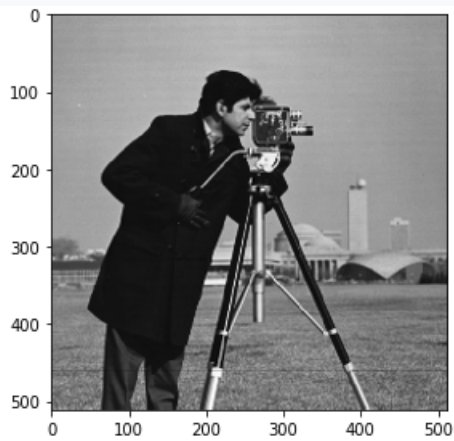
b) Dla tego samego obrazu przygotuj histogram dla 8 kubelków i tego samego zakresu. Histogram zaprezentuj na wykresie matplotlib jako słupki (*plt.bar*)

c) dla tego samego obrazu wykonaj histogram skumulowany i zaprezentuj go jako linię. Wskazówka Dla zwykłego histogramu użyj funkcji *np.cumsum*

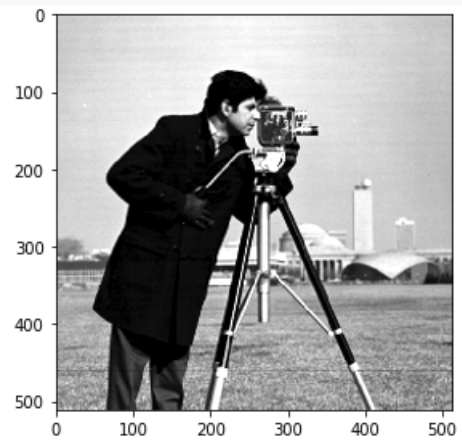
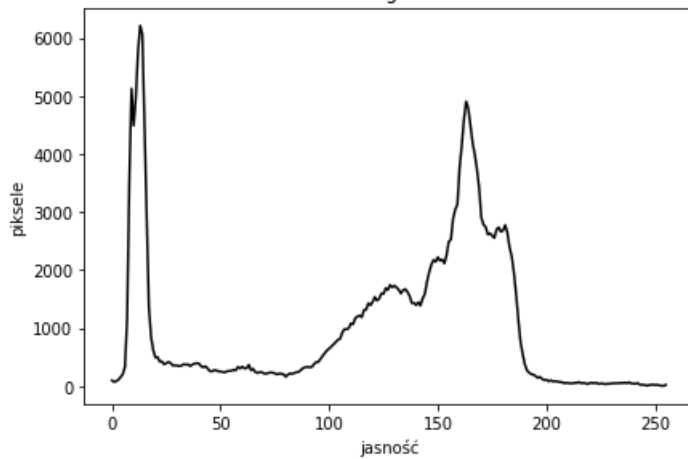
ZADANIE 2

a) Na pierwszym wykresie histogramu zaobserwuj miejsce bliskie zeru, dla którego wykres nieznacznie się podnosi. Zanotuj tę wartość jako *a*. Podobnie, metodą obserwacji znajdź liczbę *b* bliską 255, poniżej której wykres wyraźnie się podnosi.

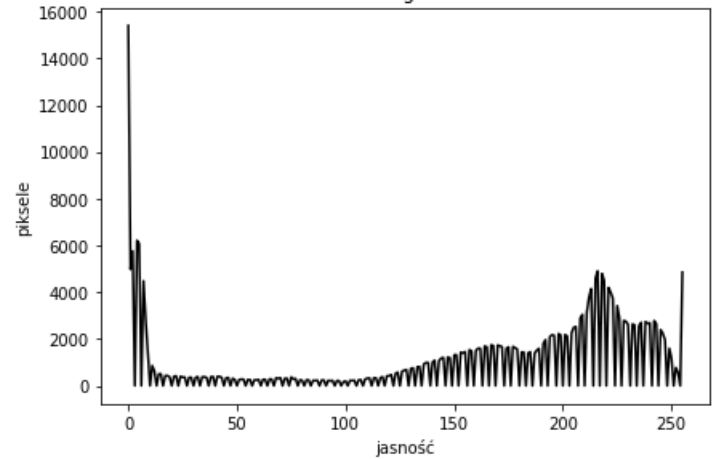
Na stronie <http://scikit-image.org/docs/dev/api/skimage.exposure.html> jest lista komend dla zmiany jasności i kontrastu obrazu. Wykorzystaj *rescale_intensity* z parametrami *a* i *b*. Ponownie wyświetl obraz i jego histogram. Może on wyglądać tak:



histogram



histogram



Jeśli na brzegach są bardzo wysokie słupki oznacza to, że wartość a jest zbyt duża, a wartość b , za mała. Skoryguj te wartości. Wyjaśnij, dlaczego po takiej operacji dla niektórych kubeków histogramu pojawiły się zera?

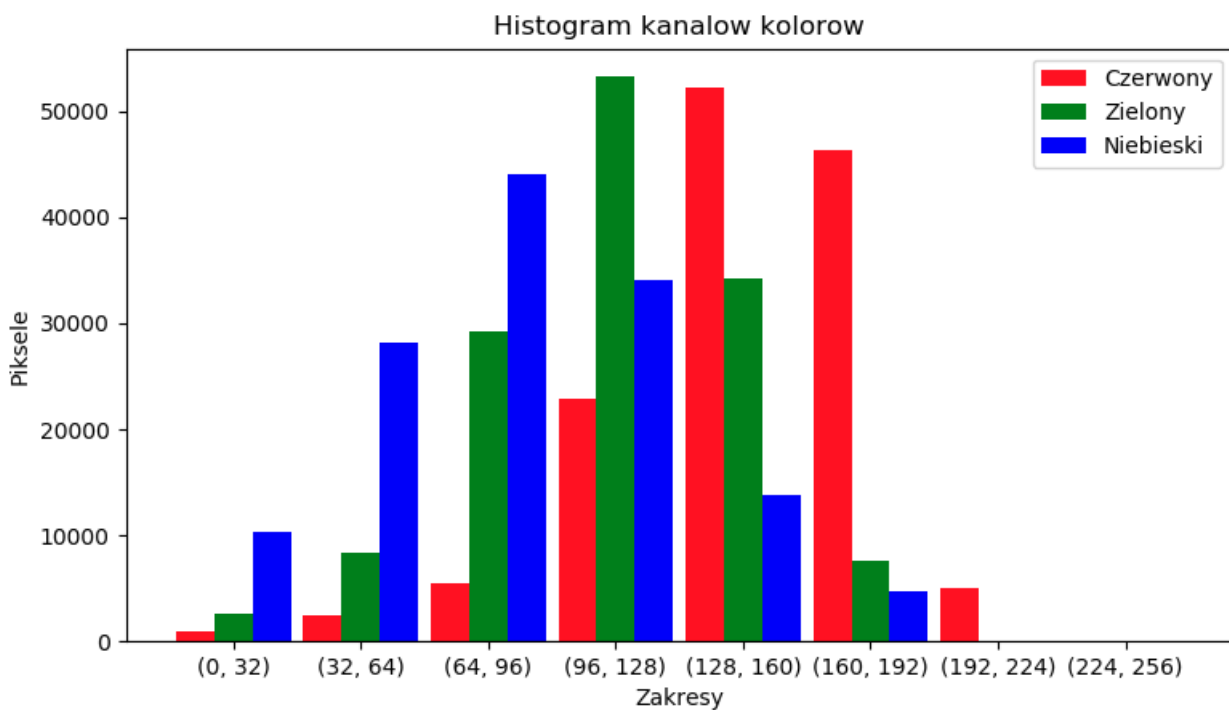
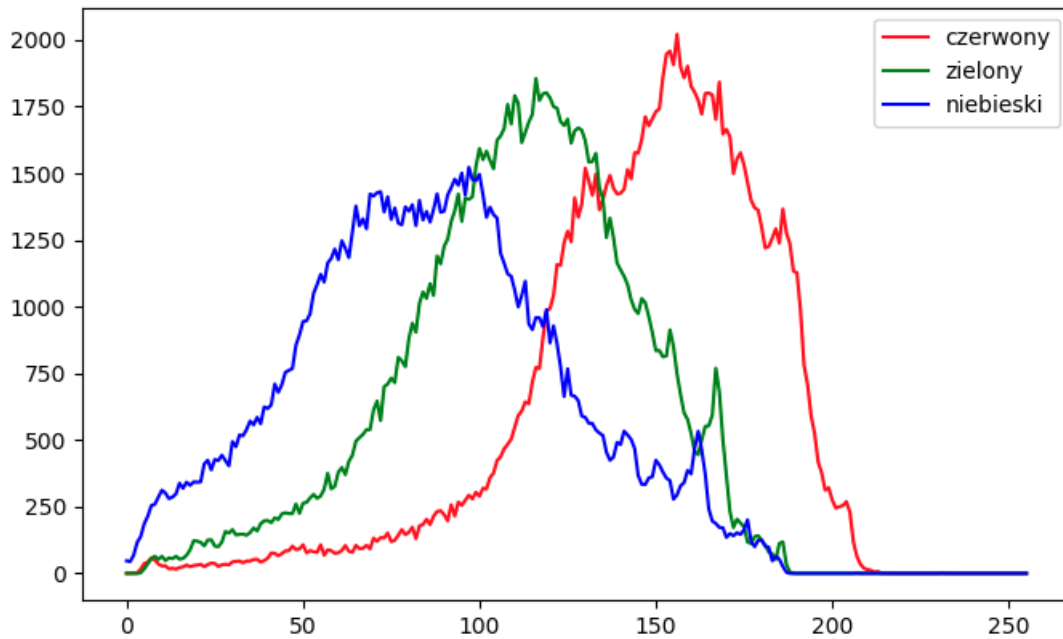
b) powyższe operacje wykonaj dla załączonego obrazu *dark_image.png*. Chcemy rozjaśnić obraz tą samą metodą. Jakie ustaliłeś a i b ? Czy można osiągnąć zadowalający efekt? Następnie spróbuj użyć funkcji *adjust_gamma*. Jaki parametr (>1 czy <1) powinien być zastosowany, aby uzyskać rozjaśnienie obrazu? Jaka wartość daje najładniejszy efekt? A może najpierw użyć korekcji gamma a następnie zmienić skalę?



ZADANIE 3

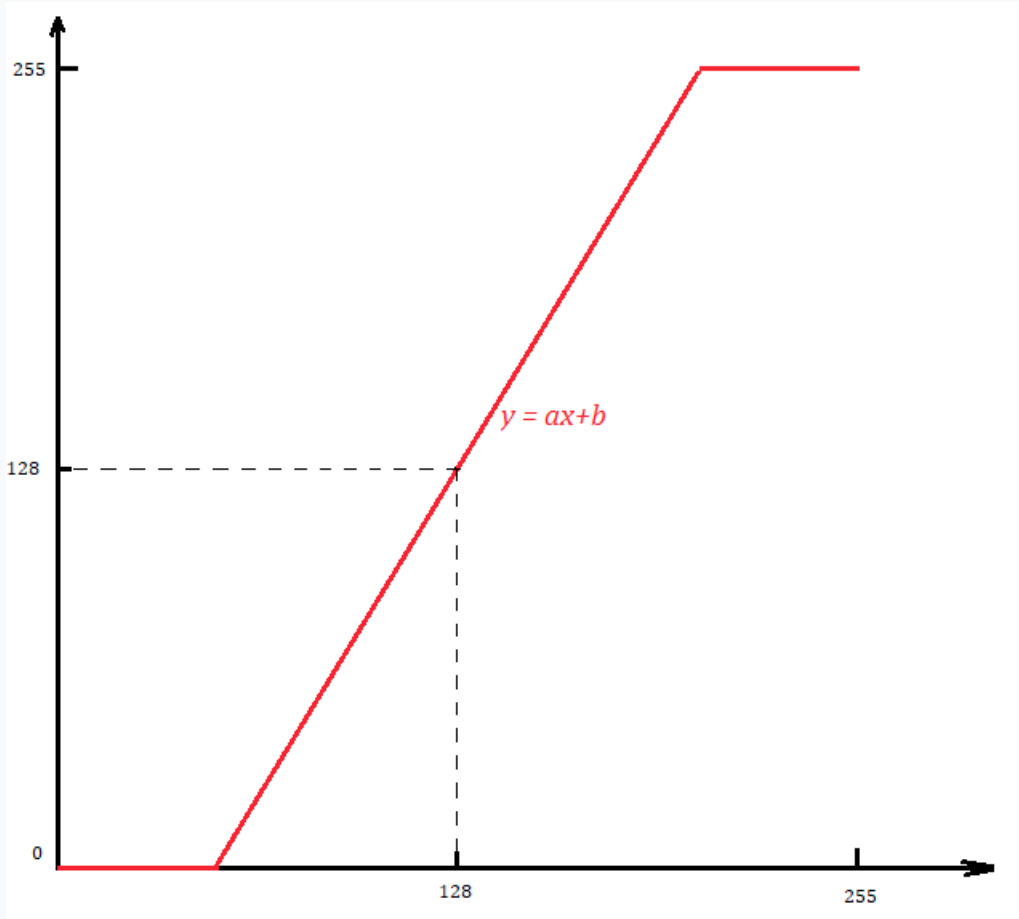
Wczytaj barwny obraz *skimage.data.chelsea()* lub *skimage.data.coffee()*

Wykonaj dla histogramu wykresy liniowy i słupkowy podobnie, jak w Zadaniu 1, ale dla kanałów R, G, B koloru, 3 wykresy na jednej figurze.



ZADANIE 4 - Tablica LUT i ręczna zmiana ekspozycji obrazu

a) przygotuj w Pythonie funkcję liniową, która dla dowolnej liczby z przedziału [0:256] będzie tworzyła wykres, jak poniżej.



Funkcję określ tak, że przechodzi przez punkt 128,128 oraz ma zakres wartości ograniczony do przedziału [0:256]. Niech dodatkowym parametrem tej funkcji (oprócz x) będzie a , która jest nachyleniem funkcji liniowej. Na powyższym wykresie $a > 1$. Funkcja ta powinna zwracać wartość zaokrągloną do całkowitej.

Funkcja ta przekształci jasność każdego punktu obrazu w nową jasność.

Jednak zwykle nie robi się tego bezpośrednio, tylko poprzez tzw. LUT

LUT jest tablicą, która pod indeksem i przechowuje więc wartość $fun(i)$. Np.

```
a = 1.2
```

```
LUT = [fun(x, a) for x in range(256)]
```

Stosując LUT wykonaj zmianę ekspozycji dowolnego obrazu czarno-białego z `skimage.data`

