

Sprawozdanie z laboratorium nr 4 z przedmiotu WMM

Wylosowany przeze mnie obraz: barbara.png:



Rozpoczęcie zadania rozpocząłem od importu potrzebnych bibliotek, stworzenie ścieżek do plików i przygotowaniu potrzebnych funkcji:

Biblioteki

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Ścieżki do obrazów

```
standar_img = "./images/barbara_col.png"  
inoise_img = "./images/barbara_col_inoise.png"  
inoise2_img = "./images/barbara_col_inoise2.png"  
noise_img = "./images/barbara_col_noise.png"
```

Podstawowe funkcje

```
def save_image(name, image):  
    cv2.imwrite(change_img+name, image)  
  
def plt_imshow(img, img_title="image"): # funkcja służąca do rysowania obrazu  
    plt.figure()  
    plt.title(img_title)  
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # bez tej linijki obraz staje się niebieski  
    plt.imshow(img, cmap="gray", vmin=0, vmax=255)  
    plt.xticks([]), plt.yticks([])  
    plt.show()  
  
def calcPSNR(img1, img2): # funkcja służąca do obliczania PSNR  
    imax = 255.**2  
    mse = ((img1.astype(np.float64)-img2)**2).sum()/img1.size  
    return 10.0*np.log10(imax/mse)  
  
def print_histogram(image): # funkcja służąca do rysowania histogramu  
    histogram = cv2.calcHist([image], [0], None, [256], [0, 256])  
    histogram = histogram.flatten()  
    plt.figure()  
    plt.plot(histogram)  
    plt.xlim([0,256])  
    plt.show()
```

1. Zadanie nr 1

Zrealizować operację filtracji barwnego obrazu cyfrowego.

Do realizacji zadania wykorzystać obrazy zaszumione (szumem gaussowskim oraz impulsowym).

Każdy z obrazów wejściowych poddać przetwarzaniu filtrem wygładzającym (Gausa) i filtrem medianowym. Każdy obraz wynikowy wyświetlić i obliczyć dla niego PSNR (w stosunku do obrazu oryginalnego, nie zaszumionego!), funkcja do obliczania PSNR dostępna jest w przykładowym skrypcie). Ocenić działanie filtrów dla masek o rozmiarach: 3x3, 5x5, 7x7.

Przygotowałem dwie funkcję, które mają spełnić wymagania zadania odnośnie do działania filtrów dla różnej liczby masek oraz typów filtrów:

Karol Kasperek
Nr albumu: 311294

```
def zad_1_Gauss(which_image): # funkcja zwracająca PSNR i obraz dla podanego obrazu z zastosowaniem filtru Gaussa
    image = cv2.imread(which_image, cv2.IMREAD_UNCHANGED)
    for i in [3, 5, 7]: # pętla sprawdzająca wyniki dla mask 3, 5 oraz 7
        print(f'Maska: {i} x {i}')
        gauss_blur = cv2.GaussianBlur(image, (i,i), 0) # stosowanie filtru Gaussa
        print(calcPSNR(unchanged_img, gauss_blur)) # liczenie PSNR
        plt.imshow(gauss_blur) # rysowanie obrazu
```

Powyższa funkcja generuje wartość PSNR oraz obraz po zastosowaniu filtra Gaussa dla podanego przy wywołaniu obrazu. Otrzymałem następujące obrazy:

Filtr Gaussa:

Inoise1_maska_3x3



Inoise_maska_5x5

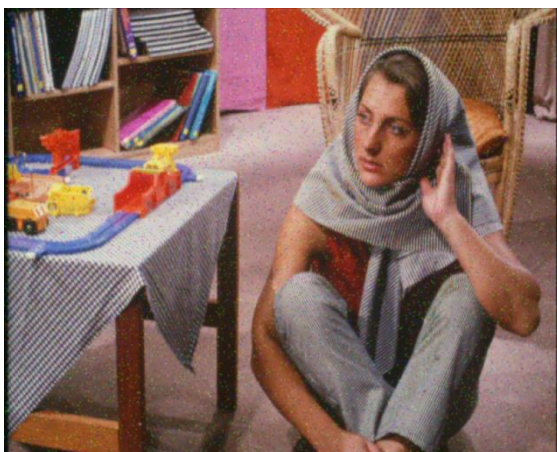


Inoise1_maska_7x7

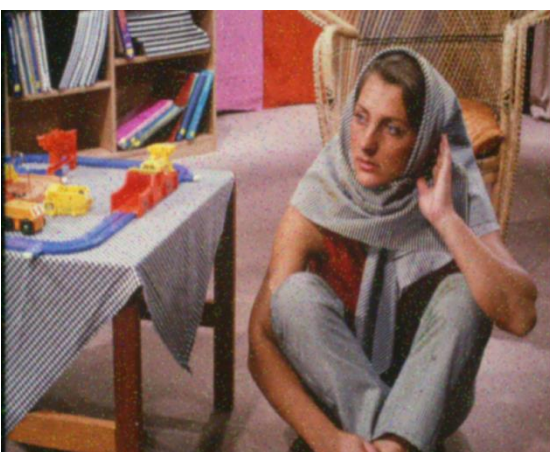


Karol Kasperek
Nr albumu: 311294

Inoise2_maska_3x3



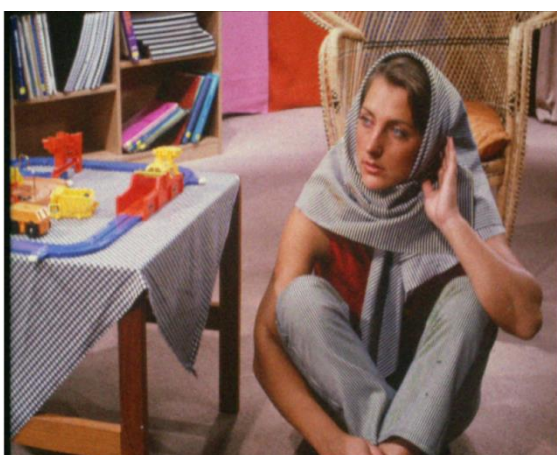
Inoise2_maska_5x5



Inoise2_maska_7x7



Noise_maska_3x3



Noise_maska_5x5



Noise_maska_7x7



Filtr medianowy:

```
def zad_1_median(which_image): # funkcja zwracająca PSNR i obraz dla podanego obrazu z zastosowaniem filtru medianowego
    image = cv2.imread(which_image, cv2.IMREAD_UNCHANGED)
    for i in [3, 5, 7]: # pętla sprawdzająca wyniki dla mask 3, 5 oraz 7
        print(f'Maska: {i} x {i}')
        median_blur = cv2.medianBlur(image, i) # stosowanie filtru medianowego
        print(calcPSNR(unchanged_img, median_blur)) # liczenie PSNR
        plt.imshow(median_blur) # rysowanie obrazu
```

Inoise_maska_3x3



Inoise_maska_5x5



Inoise_maska_7x7



Karol Kasperek
Nr albumu: 311294

Inoise2_maska_3x3

Inoise2_maska_5x5



Inoise2_maska_7x7



Noise_maska_3x3

Noise_maska_5x5



Noise_maska_7x7



Tabele PSNR:

Bez filtrów		
Inoise_1	Inoise_2	noise
21,5116	20,882	27,5834

	Filtr Gaussa		
Maska	Inoise_1	Inoise_2	noise
3x3	25,9067	25,6297	27,5827
5x5	25,1477	25,0139	25,8812
7x7	24,279	24,2121	24,6365

	Filtr medianowy		
Maska	Inoise_1	Inoise_2	noise
3x3	26,3346	26,3029	25,7469
5x5	23,8678	23,8547	23,7392
7x7	23,4586	23,4584	23,3564

Pytania i wnioski:

Jaki wpływ na skuteczność filtracji i na zniekształcenie obrazu ma rozmiar maski filtru?

Generalnie większa maska powoduje lepsze usunięcie szumów, jednak jest to powiązane również z gorszą jakością obrazów, na których widać niechciane plamy lub są rozmywane.

Czy ocena subiektywna uzyskanych obrazów wynikowych, jest zgodna z PSNR (lepszą jakość – większy PSNR)?

Porównując otrzymane obrazy z ich wartościami PSNR można stwierdzić, że PSNR jest dość mocno skorelowany z jakością obrazów, chociaż nie jest to wiążąca ocena danego obrazu. Zdecydowanie najlepiej wyszły obrazy z maską 3x3 co również zgadza się z otrzymanymi w tabeli wynikami.

2. Zadanie nr 2

Zrealizować operację wyrównania histogramu dla obrazu barwnego i zapisać obraz wynikowy do pliku. UWAGA: operację wyrównania histogramu należy wykonać wyłącznie dla składowej odpowiadającej za jasność, w tym celu należy wejściowy obraz RGB skonwertować do innej przestrzeni (np. YCbCr/YUV), a po wyrównaniu histogramu dla właściwej składowej powrócić do pierwotnego formatu.

Porównać uzyskane obrazy i ich histogramy (w szczególności: histogram dla składowej, dla której wykonano operację wyrównywania histogramu).

Czy obraz po wyrównaniu histogramu jest subiektywnie *lepszey* jakości?

```
def zad2(image):  
    image_YCrCb = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)  
    image_YCrCb[:, :, 0] = cv2.equalizeHist(image_YCrCb[:, :, 0])  
    image_end = cv2.cvtColor(image_YCrCb, cv2.COLOR_YCrCb2BGR)  
    plt.imshow(image)  
    plt.imshow(image_end)  
    print_histogram(image)  
    print_histogram(image_end)  
    save_image("normal_hist.png", image_end)
```

Standardowy obraz:

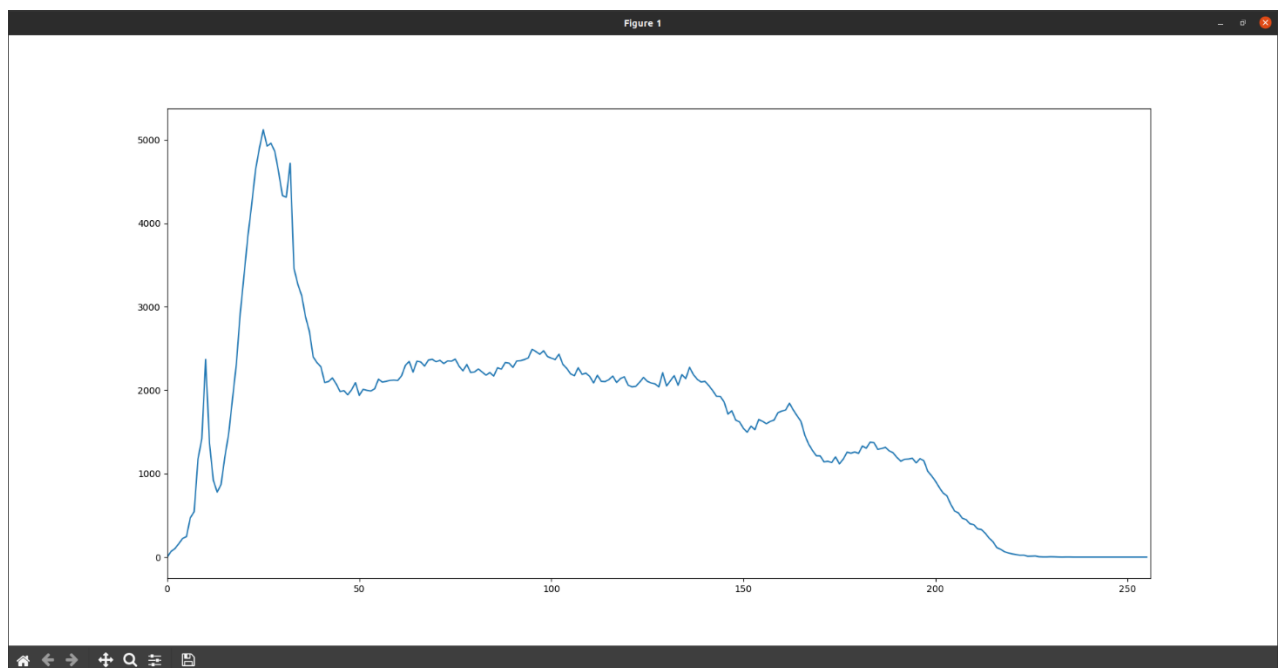


Karol Kasperek
Nr albumu: 311294

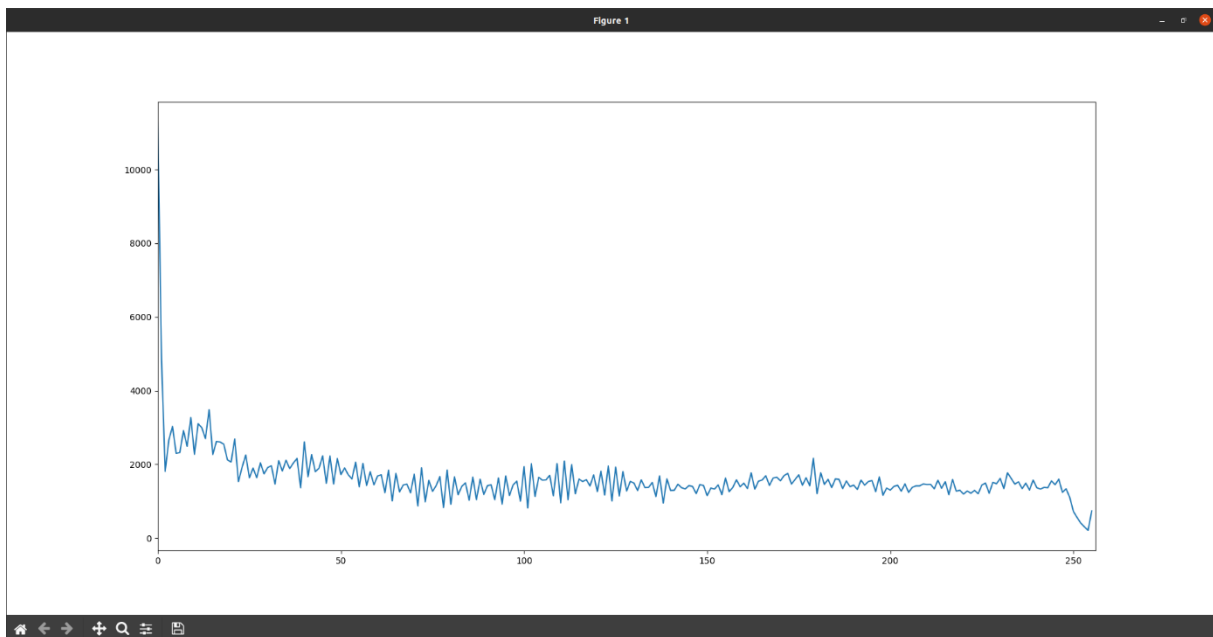
Obraz po wyrównaniu histogramem:



Wykres przed wyrównaniem:



Wykres po wyrównaniu:



Wnioski i pytania:

Czy obraz po wyrównaniu histogramu jest subiektywnie *lepszego* jakości?

Moim zdaniem obraz po wyrównaniu jest gorszej jakości. Jest znacznie bardziej prześwietlony, kontrast pomiędzy jasnymi i ciemnymi powierzchniami jest nienaturalny i nieprzyjemny dla oka.

3. Zadanie nr 3

Korzystając z filtru Laplace'a do wyznaczenia wysokoczęstotliwościowych składowych obrazu dokonać *wyostrzenia* obrazu:

```
img_out = img_in + W*img_laplace.
```

Jaki jest wpływ *wagi* składowej wysokoczęstotliwościowej na postać obrazu wynikowego?

Dla jakich wartości tej *wagi* uzyskuje się *dobrye, przyjemne* dla oka wyniki?

Uwaga: należy pamiętać, że wyostanie obrazu powoduje również uwydatnienie szumu w obrazie, w niektórych przypadkach (niezbyt dobrej jakości obrazów oryginalnych) przydatne może być wstępne wygładzenie obrazu filtrem dolnoprzepustowym (np. filtrem Gaussa).

Wskazówka: wykorzystać funkcję `cv2.addWeighted()` do sumowania obrazów i równoczesnej konwersji wyniku do 8-bitowych liczb całkowitych z przedziału `[0, 255]`.

Uwaga: ze względu na sposób wyliczania laplasjanu w bibliotece OpenCV w celu uzyskania 'dobrych' wyników należy odjąć 'obraz wysokoczęstotliwościowy', czyli zastosować ujemne wartości *wagi* \bar{W} .

```
def zad3():  
    W = -10  
    image = unchanged_img  
    gauss_image = cv2.GaussianBlur(image, (3,3), 0)  
    laplacian_image = cv2.Laplacian(gauss_image, cv2.CV_64F)  
    img = np.asarray(image, np.float64)  
    img_out = cv2.addWeighted(img, 1, laplacian_image, W, 0)  
    cv2.imwrite(change_img+"laplacian-10.png", img_out)
```

Zastosowałem tutaj różne wartości składowej wysokoczęstotliwościowej:

W = 5



W = 2

Karol Kasperek
Nr albumu: 311294



W = 1



W = 0

Karol Kasperek
Nr albumu: 311294



W = -1



W = -2

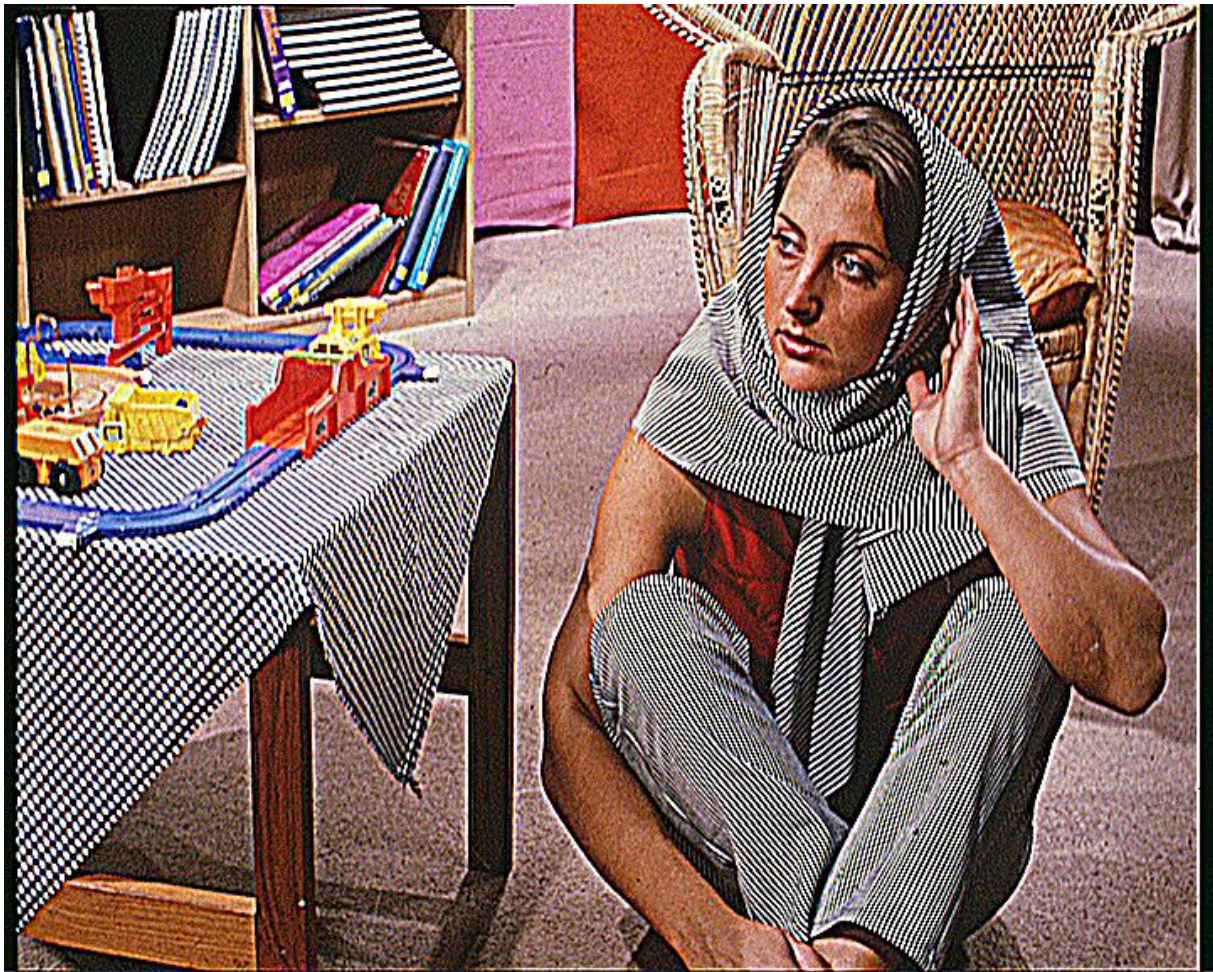
Karol Kasperek
Nr albumu: 311294



W = -5



W = -10



Wnioski i pytania:

Jaki jest wpływ *wagi* składowej wysokoczęstotliwościowej na postać obrazu wynikowego?

- Im niższa wartość *wagi* składowej tym bardziej uwydatniane są krawędzie wszystkich obiektów na obrazie
- Dodatnie wartości powodują rozmycie się obrazu – efekt przeciwny

Dla jakich wartości tej *wagi* uzyskuje się *dobre, przyjemne* dla oka wyniki?

Najprzyjemniejsze dla oka wydają się obrazy, dla których wartość jest z zakresu [0,5].