

**Sprawozdanie Andrzej Żaba**  
**nr indeksu: 401490**  
**Inżynieria mechatroniczna L3**

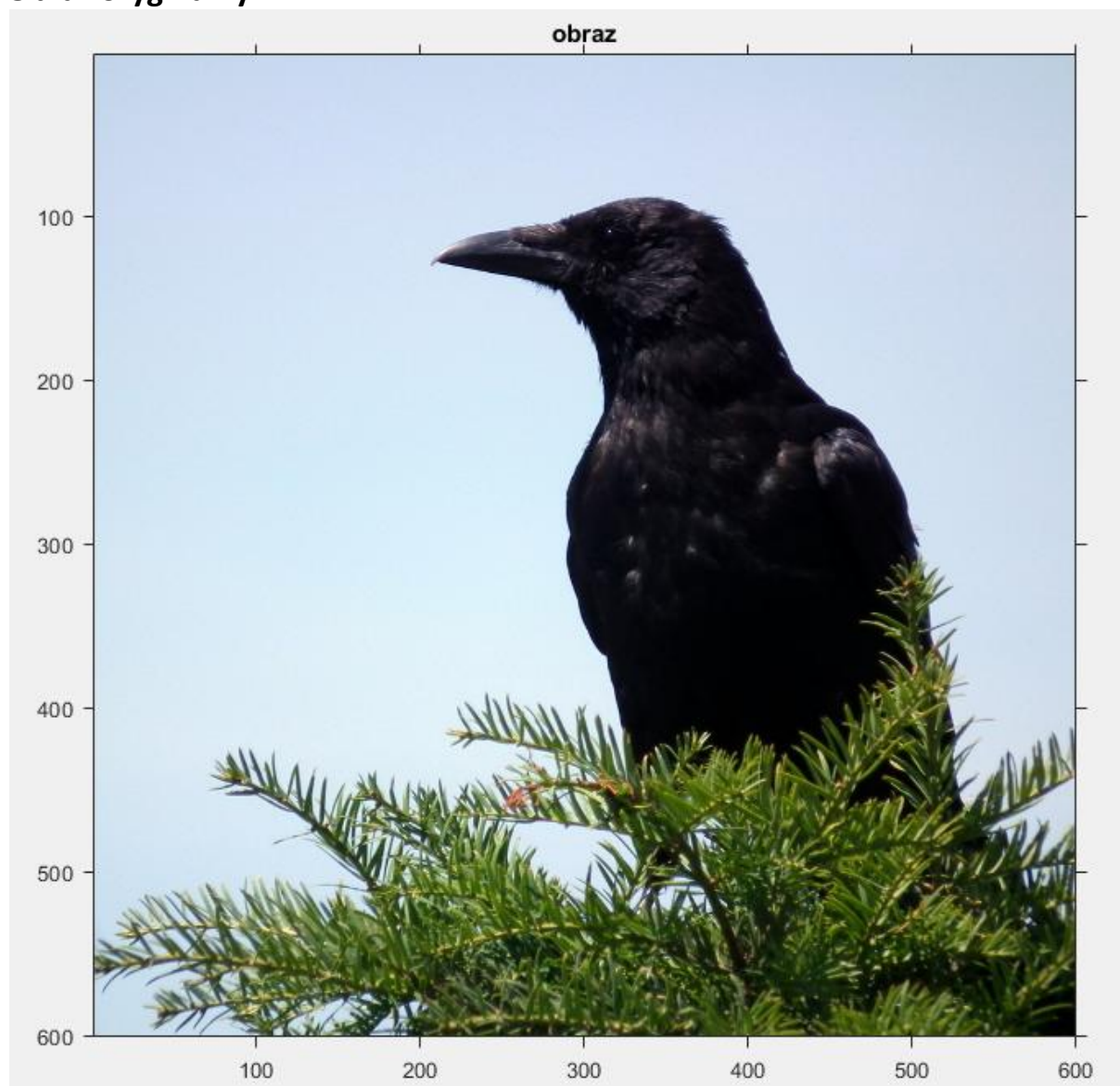
**Zad1.**

```
clc
close all
clear all

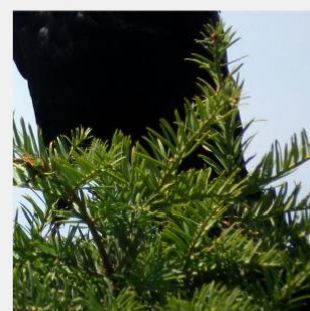
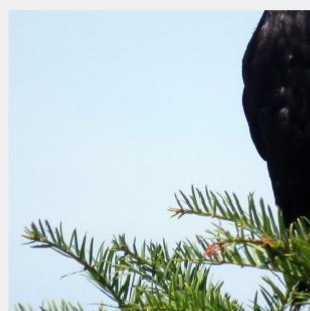
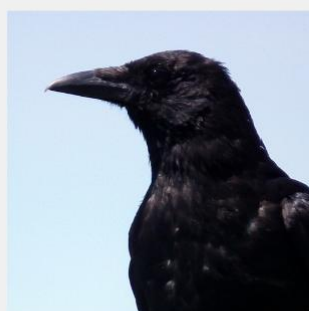
x = imread('Images/Crow.jpg');
imshow(x), title('obraz'), axis on
ONE = imcrop(x, [170 50 300 300]);
TWO = imcrop(x, [60 220 300 300]);
THREE = imcrop(x, [300 300 300 300]);
%% PLOT
figure(1)
subplot(1,3,1)
imshow(ONE)
subplot(1,3,2)
imshow(TWO)
subplot(1,3,3)
imshow(THREE)
%% Converting to gray scale
ONE_gray = rgb2gray(ONE);
TWO_gray = rgb2gray(TWO);
THREE_gray = rgb2gray(THREE);
%% Histograms

figure(2)
subplot(1,3,1)
imhist(ONE_gray)
subplot(1,3,2)
imhist(TWO_gray)
subplot(1,3,3)
imhist(THREE_gray)
```

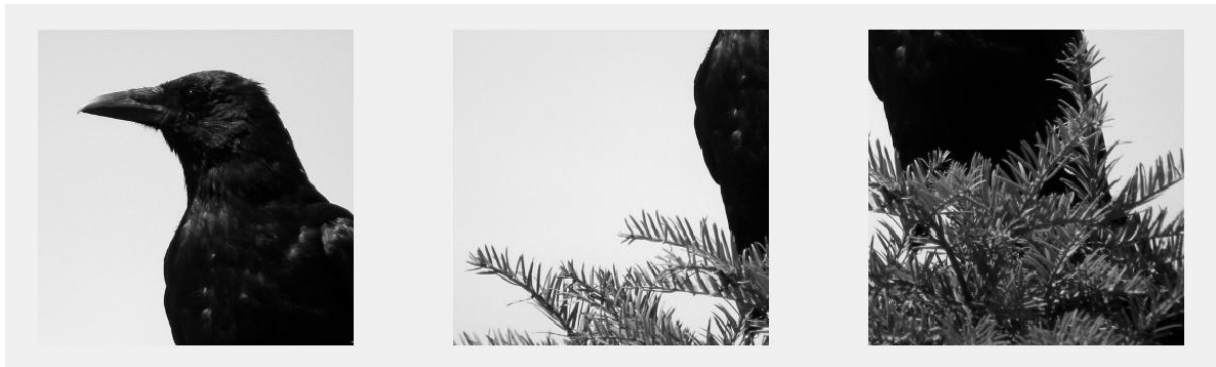
**Obraz oryginalny:**



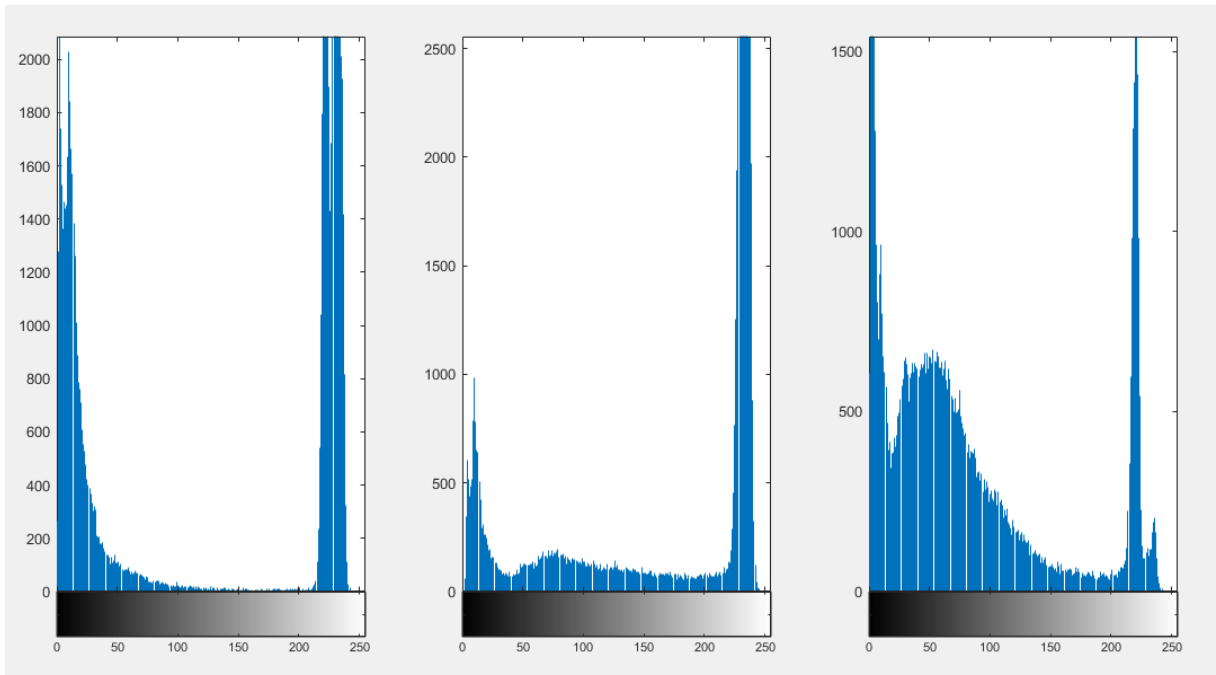
**Fragmenty obrazu:**



Fragmenty obrazu w odcieniach szarości:



Histogramy:

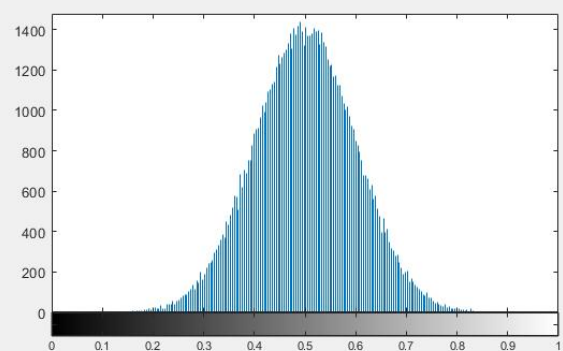
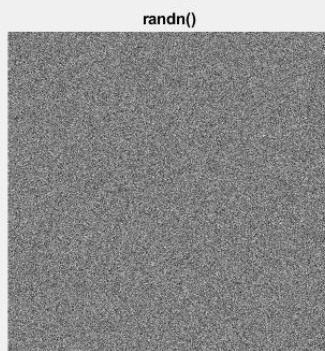
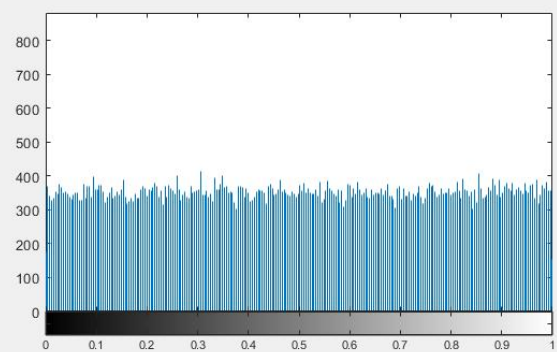
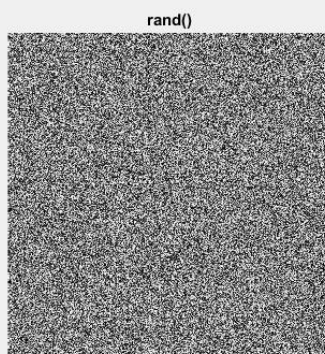


Wszystkie trzy histogramy charakteryzują się wysokim pikiem na końcu skali szarości oraz na jej początku (może poza rys. 2). Oznaczają one dużą ilość pikseli o skrajnych wartościach - mocna czerń (blisko wartości zero) oraz wyraźnie białe (blisko wartości 255). Owe skrajne wartości opisują piksele tła (białe) oraz piór kruka (czarne). Na histogramie rys trzeciego widzimy nasilenie ilości pikseli o wartości w przedziale 50 -100. Są to piksele przedstawiające gałązki drzewa.

## Zad2.

```
clc;
clear all
close all
A=[];
for k =1:1:300
    for i=1:1:300
        A(k,i) = rand();
    end
end
B=[];
for k =1:1:300
    for i=1:1:300
        B(k,i) = (1+randn())/5)/2;
    end
end
figure(1)
subplot(2,2,1)
imshow(A);
title('rand()')
subplot(2,2,2)
imhist(A)

subplot(2,2,3)
imshow(B)
title('randn()')
subplot(2,2,4)
imhist(B)
```

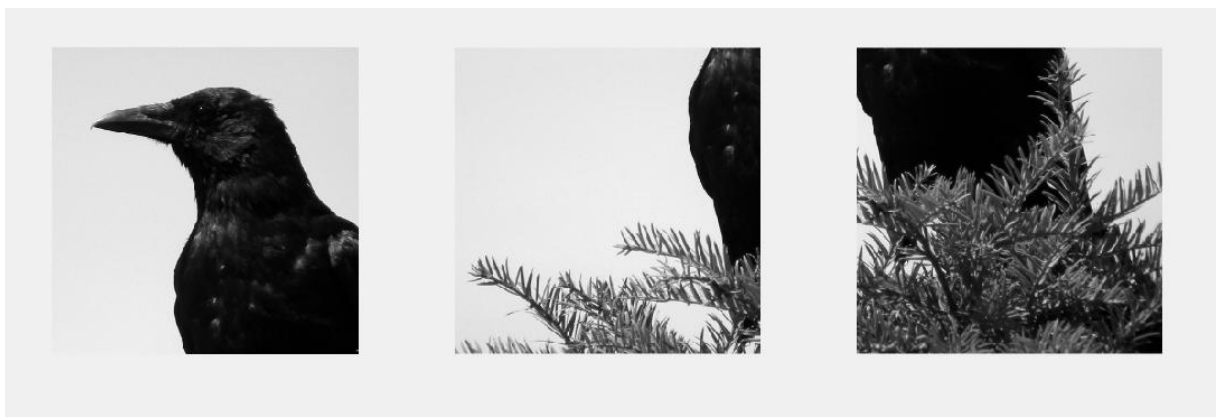


Wyniki histogramów były do przewidzenia. Funkcja `rand()` generuje losowe liczby z przedziału 0:1, oznacza to, że prawdopodobieństwo na wylosowanie każdej wartości było takie samo. Dlatego ilości pikseli o poszczególnych wartościach są podobne.

Natomiast funkcja `randn()` generuje losowe liczby z przedziału 0:1, jednak z zachowaniem proporcji rozkładu normalnego. Stąd najwięcej pikseli przyjmuje wartości w połowie skali szarości.

Zad3.

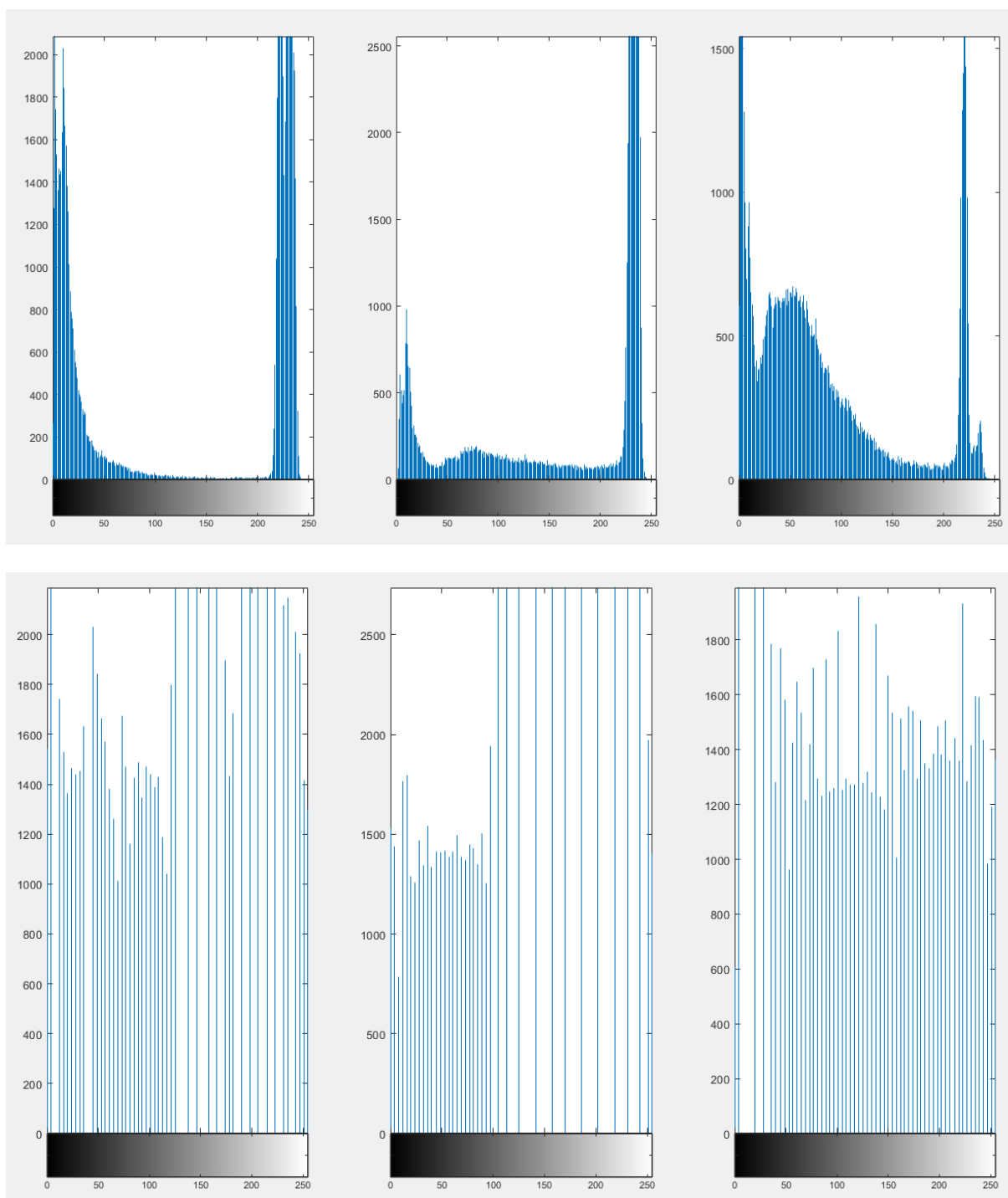
Fragmenty obrazu w skali szarości:



Fragmenty obrazu w skali szarości po wyrównaniu histogramów:



## Porównanie histogramów:



Histogramy po wyrównaniu są jakby "rzadsze". Oznacza to, że częściej otrzymujemy konkretne wartości i dlatego otrzymujemy ich więcej. Natomiast różnorodność otrzymywanych wartości przyjmowanych przez piksele zmalała.



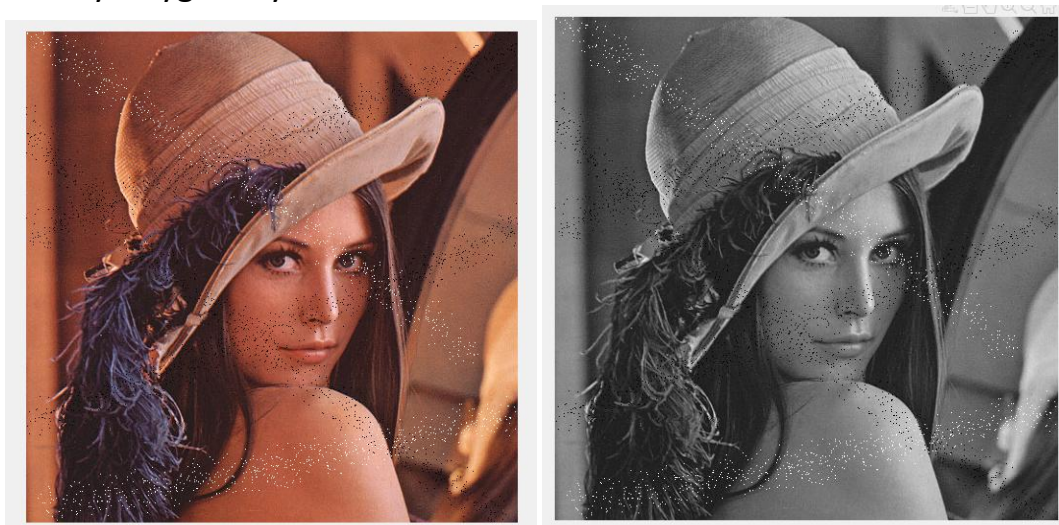
## Instrukcja 2

### Zad1.

```
clc
close all
clear all
figure(1)
im = imread('Images/Lena_dots.jpg');
imshow(im)
img = rgb2gray(im)
figure(5)
imshow(img)
%% mask filter
mask = [0.17 0.67 0.17; 0.67 -2.33 0.67; 0.17 0.67 0.17];
Fim1 = conv2(double(img), double(mask));
figure(4)
imshow(uint8(Fim1))
%% Lowpass filter
h1 = fspecial('average', [3 3]);
Fim2A = imfilter(im,h1,'replicate');
h2 = fspecial('gaussian',[3 3], 1000);
Fim2B = imfilter(im,h2,'replicate');
figure(2)
subplot(1,2,1)
imshow(uint8(Fim2A))
subplot(1,2,2)
imshow(uint8(Fim2B))

%% Nonlinear filter
imNoise = imnoise(img,'salt & pepper',0.02);
Fim3 = medfilt2(imNoise);
figure(3)
subplot(1,2,1)
imshow(imNoise);
subplot(1,2,2)
imshow(Fim3);
```

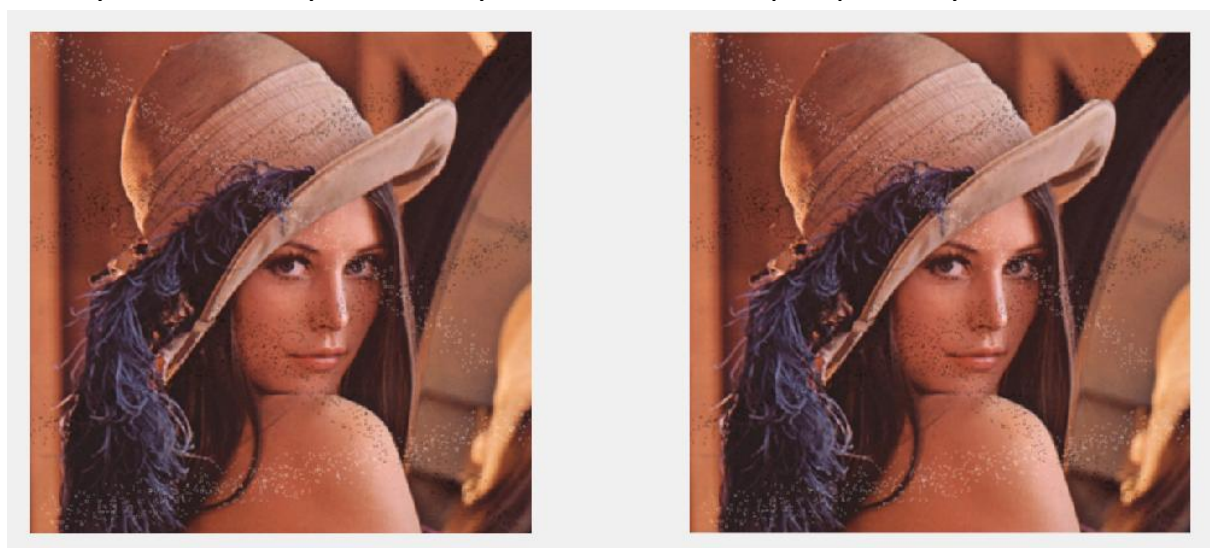
Obrazy: oryginalny i w odcieniach szarości



Obraz uzyskany ze splotu obrazu i maski:



Obrazy kolorowe uzyskane z użyciem filtrów dolnoprzepustowych:





Obrazy w skali szarości uzyskane dzięki filtrom nieliniowym:



Filtr z użyciem maski spowodował lekkie rozmazanie obrazu i wyostrenie poszczególnych krawędzi – zwłaszcza krawędzi szumu.

Filtry dolnoprzepustowe wyłącznie rozmazały obraz.

Natomiast w ostatnim zestawieniu polecenie `imnoise()` dodało więcej szumu do obrazu jednak funkcja `medfilt(imNoise)` prawie doskonale oczyściła wcześniej stworzony obraz z wszelkich szumów.

## Zad2.

```
clc
close all
clear all

im = imread('Images/Lena_dots.jpg');
imshow(im)
img = rgb2gray(im)
%% Lowpass filter
h1 = fspecial('average', 6);
h2 = fspecial('gaussian', 6, 3);
h3 = fspecial('disk', 6);
Fim2A = imfilter(img, h1, 'replicate');
Fim2B = imfilter(img, h2);
Fim2C = imfilter(img, h3);

%% Nonlinear filter
imNoise = imnoise(img, 'salt & pepper', 0.02);
Fim3 = medfilt2(imNoise);

figure(2)
subplot(3, 2, 1)
imshow(img);
title("Original")
subplot(3, 2, 2)
imshow(imNoise);
title("Noise")

subplot(3, 2, 3)
imshow(uint8(Fim2A))
title('Rectangular')
subplot(3, 2, 4)
imshow(uint8(Fim2B))
title('Gaussian')

subplot(3, 2, 5)
imshow(uint8(Fim2C))
title('Radius')

subplot(3, 2, 6)
imshow(uint8(Fim3))
title('Mediana')
```

**Original**



**Noise**



**Rectangular**



**Gaussian**



**Radius**



**Mediana**



Maska obrazu przy okazji zastosowanych filtrów określa jak mocno rozmaże się obraz.

Wartość replicate odpowiada za przyjmowanie skrajnych wartości maski przez wartości leżące poza granicami maski.

Najlepiej ze wszystkich zastosowanych filtrów sprawdził się filtr Medianowy.

### Zad3.

```
clc
close all
clear all

im = imread('Images/Lena_dots.jpg');
imshow(im)
img = rgb2gray(im)
%%
% Fim - Filtered image
% h1, h2,... - masks
h1 = fspecial('prewitt');
h2 = fspecial('sobel');

%h3 = fspecial('laplacian',alpha);
%h4 = fspecial('log',hsize,sigma);
h3 = fspecial('laplacian',0.01);
h4 = fspecial('log',3,0.3);

h5 = fspecial('unsharp',0.99);

Fim1 = imfilter(img,h1);
Fim2 = imfilter(img,h2);

Fim3 = imfilter(img,h3);
Fim4 = imfilter(img,h4);

Fim5 = imfilter(img,h5);

Fim6 = edge(img,'sobel');
Fim7 = edge(img,'Canny');
Fim8 = edge(img,'Prewitt');

figure(1)
subplot(1,2,1)
imshow(Fim1)
title('Prewitt');

subplot(1,2,2)
imshow(Fim2)
title('sobel')

figure(2)
subplot(1,2,1)
imshow(Fim3)
title('Laplace');

subplot(1,2,2)
imshow(Fim4)
title('LoG')

figure(3)
subplot(2,2,1)
imshow(Fim5)
```



```

title('unsharp')

subplot(2,2,2)
imshow(Fim6)
title('edge - sobel')

subplot(2,2,3)
imshow(Fim7)
title('edge - Canny')

subplot(2,2,4)
imshow(Fim8)
title('edge - Prewitt')

```

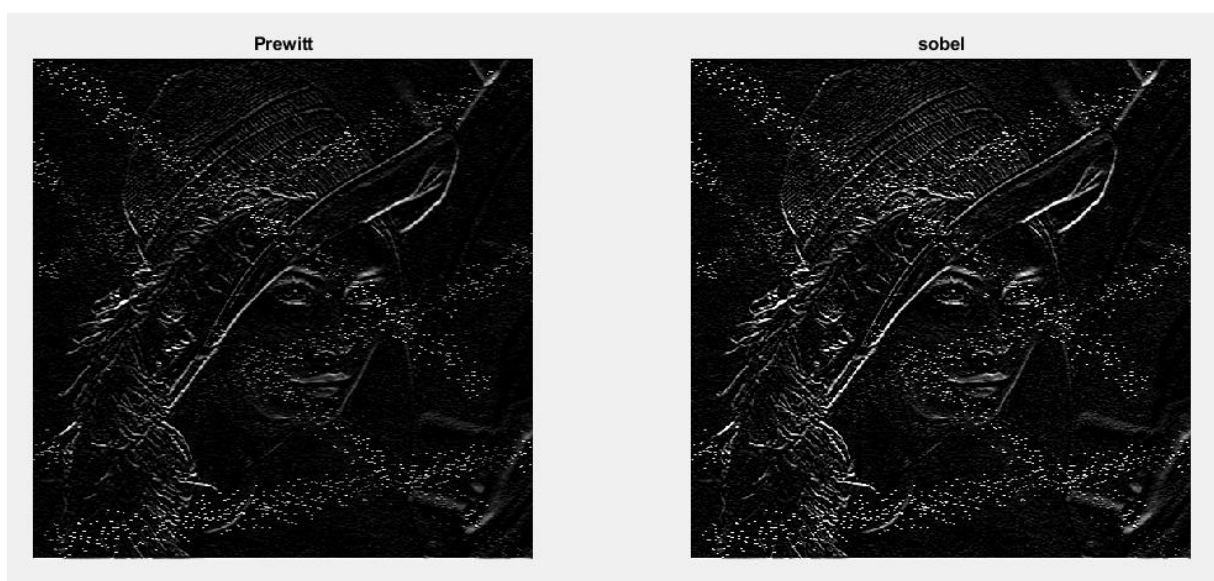
Maski:

Prewitt:

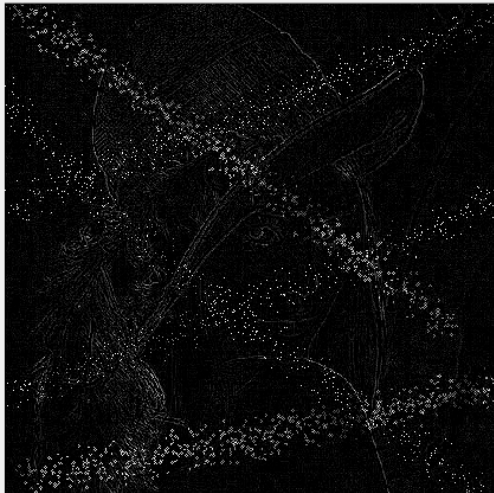
	1	2	3	4
1	1	1	1	
2	0	0	0	
3	-1	-1	-1	
4				

Sobel:

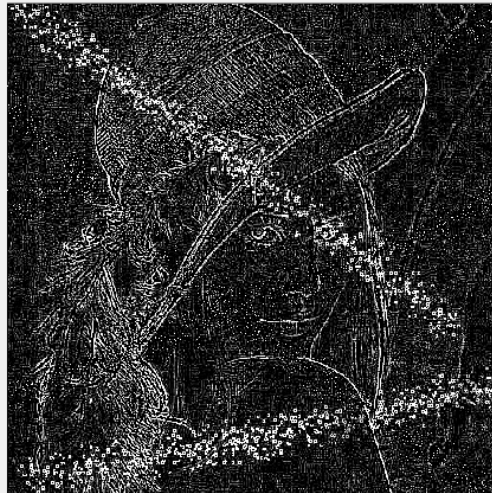
	1	2	3	4
1	1	2	1	
2	0	0	0	
3	-1	-2	-1	
4				



Laplace



LoG



Laplasjan mocno zarysowuje szumy a słabo pozostałe kształty. LoG natomiast mocno zarysował krawędzie Leny, część szumów zlała się z tłem, jednak mocniejsze szumy również zostały uwytatnione.

unsharp



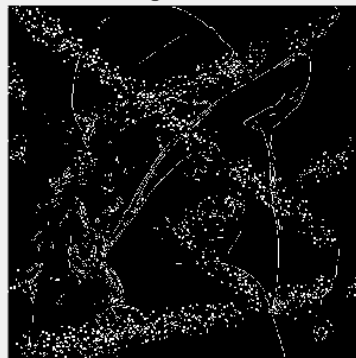
edge - sobel



edge - Canny



edge - Prewitt



Funkcja unsharp wpływa lekko na szumy. Zmienia wartości poszczególnych pikseli szumu.

Przy zastosowaniu funkcji edge jasno widać przewagę detektora krawędzi Cnny'ego. Wykrywa on po prostu zdecydowanie więcej krawędzi niż metoda Prewitta czy Sobela.

Można też zauważyć, że Prewitt, działa minimalnie lepiej od Sobela i wykrywa więcej krawędzi.