UNIVERZITET U BEOGRADU – ELEKTROTEHNIČKI FAKULTET
Katedra za Signale i sisteme
Istraživačka grupa za Biomedicinsku Instrumentaciju i Tehnologije

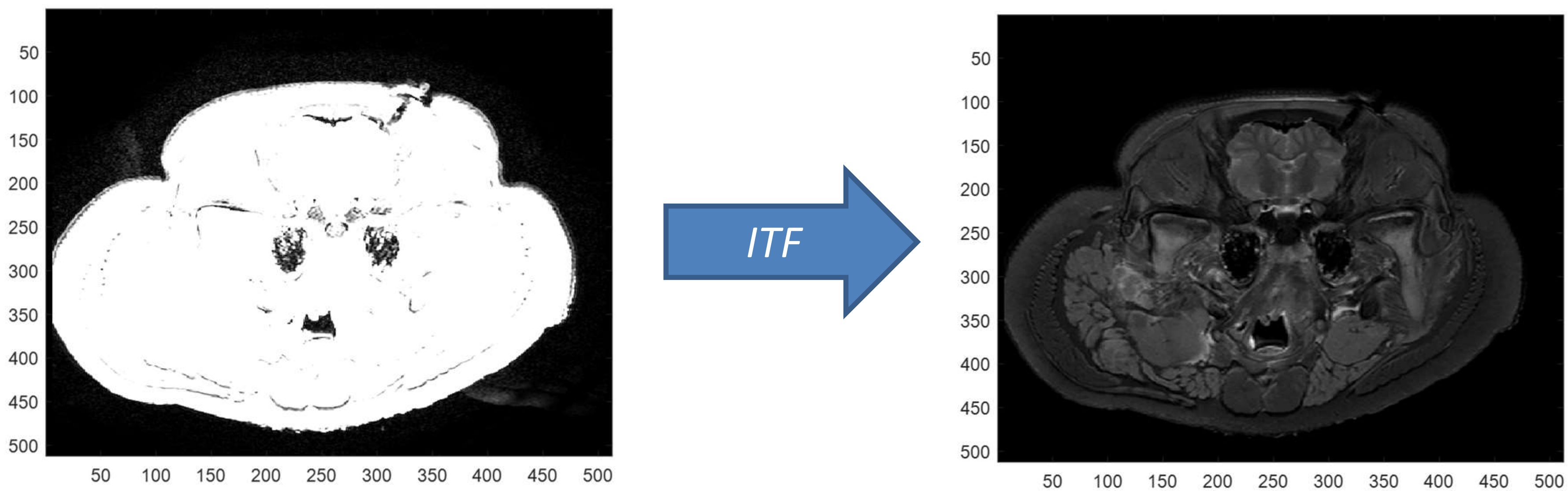
13E054ABS
Analiza biomedicinske slike

Prostor intenziteta. Filtriranje.

Predavanje je bazirano na knjizi
Wolfgang Birkfellner "Applied Medical Image Processing", CRC Press, 2014.

2022/2023

Transformacija intenziteta
Intensity Transfer Function (ITF)



Linearna transformacija

$$\rho' = \frac{\rho - \rho_{\min}}{\rho_{\max} - \rho_{\min}} \omega_{\text{target}} + \rho'_{\min}$$

ρ' ... Transformisani intenzitet

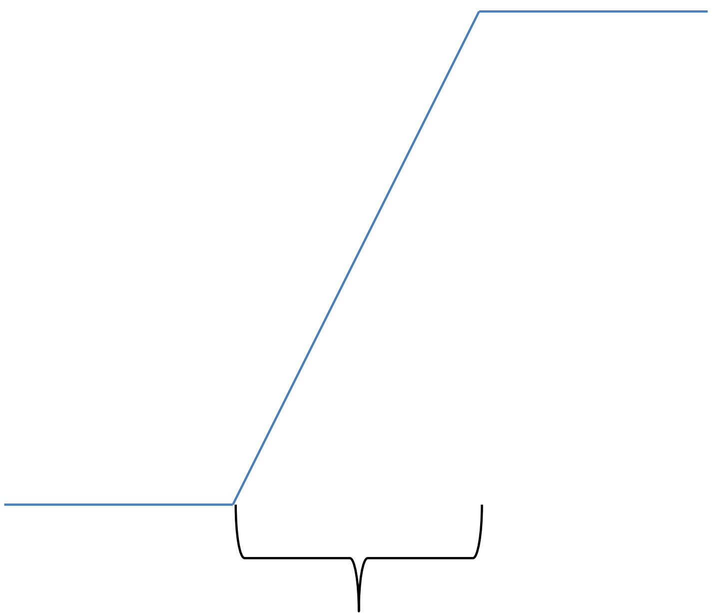
ρ ... Originalni intenzitet

$\rho_{\min}, \rho_{\max}$ ... Minimalna, maksimalna vrednost sivog na originalnog slici

ω_{target} ... Broj nivoa sivog u novom prostoru (uzimajući u obzir ciljanu bitsku dubinu)

Pitanje: ako će nova slika da ima elemente čije su vrednosti -127 do 127, koliko treba zadati da bude ω_{target}?

Transformacija intenziteta
Intensity Transfer Function (ITF)



prozor

Prozorovanje:
Primer primene
“prozora” na CT sken

Tkivo	[HU]
Vazduh	-1000
Pluća	-900 ... -200
Voda	0
Jetra	20 ... 60
Kosti	50 ... 3072
Bubrezi	40 ...50

Transformacija intenziteta
Intensity Transfer Function (ITF)

Sigmoid funkcija (Logistička funkcija)

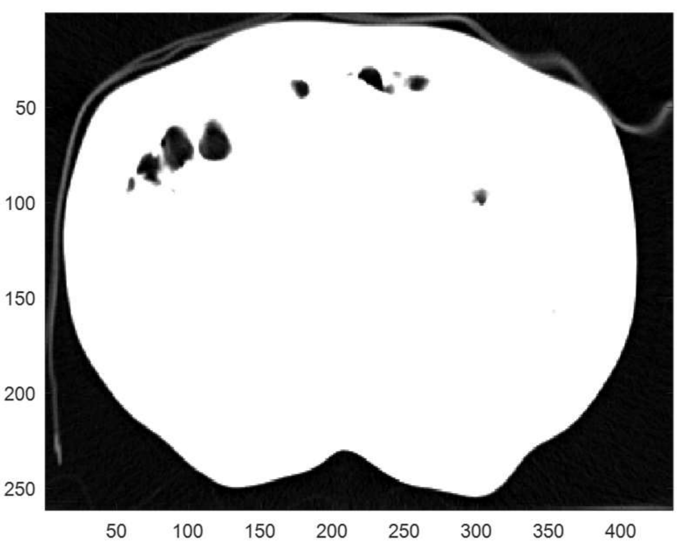
$$S(\rho) \quad = \quad 255\frac{1}{1+e^{-\frac{\rho-\omega}{\sigma}}}$$

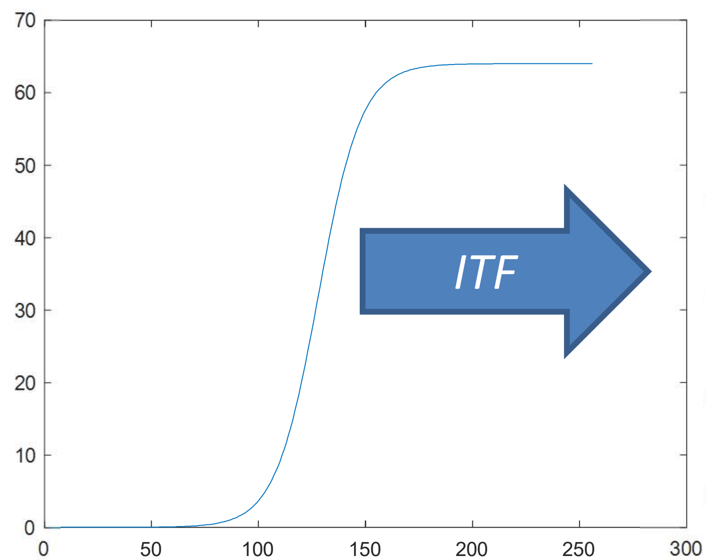
σ - sa porastom nagib se smanjuje
ω - pomeranje levo-desno

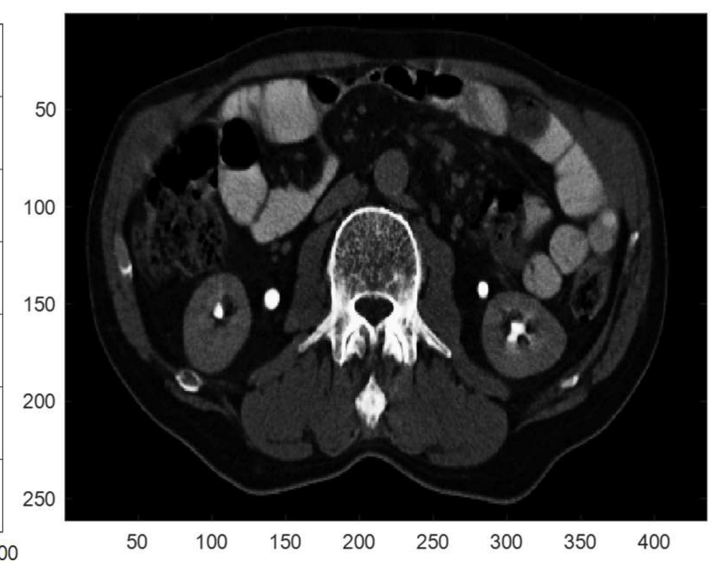
$\rho \quad \ldots$ Originalna vrednost intenziteta

$\omega \quad \ldots$ Centar distribucije vrednosti intenziteta (za 8-bitnu sliku je 127)

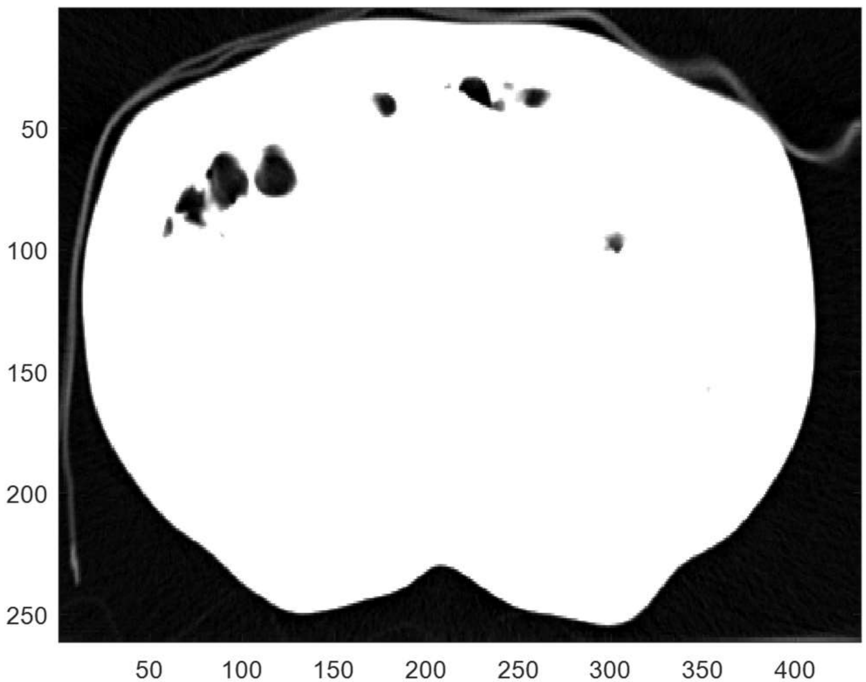
$\sigma \quad \ldots$ Širina distribucije vrednosti intenziteta

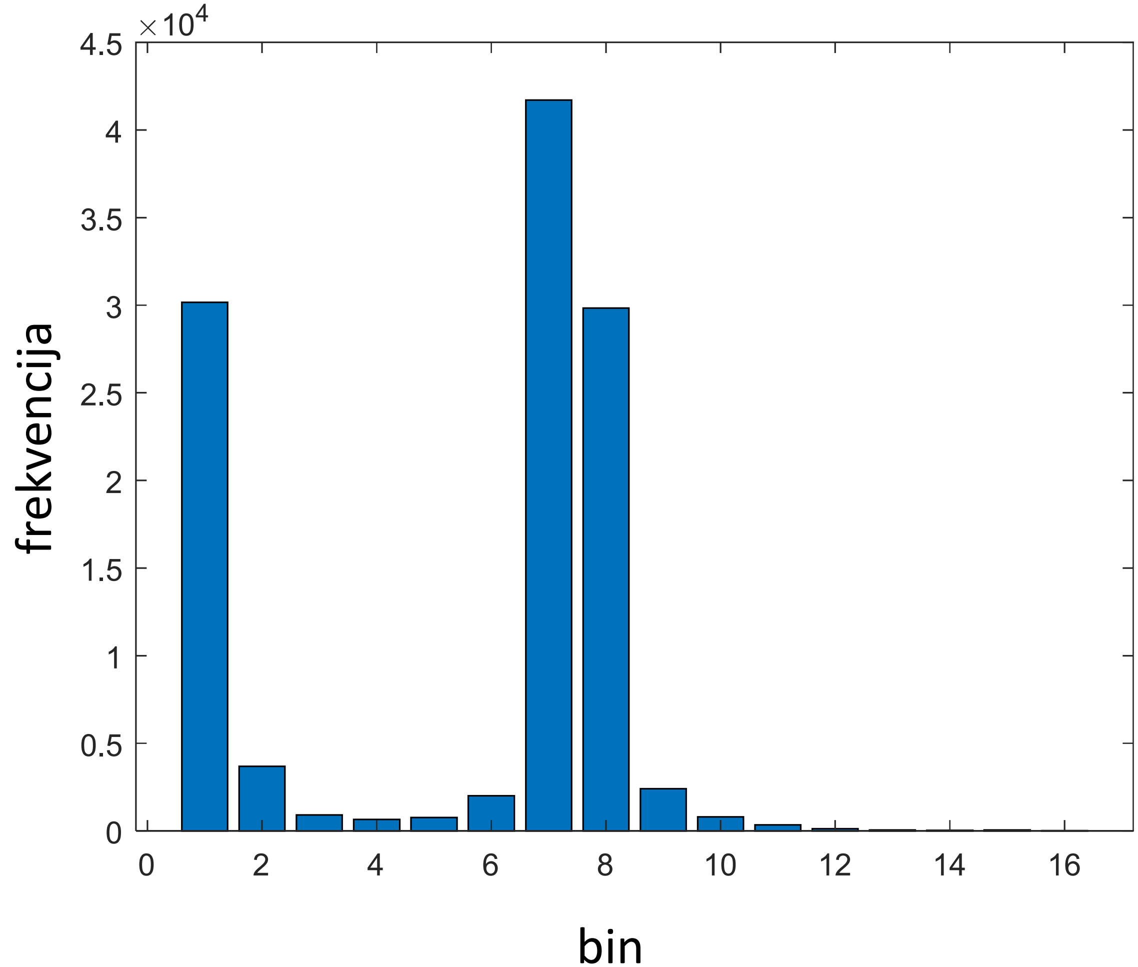






Histogram slike



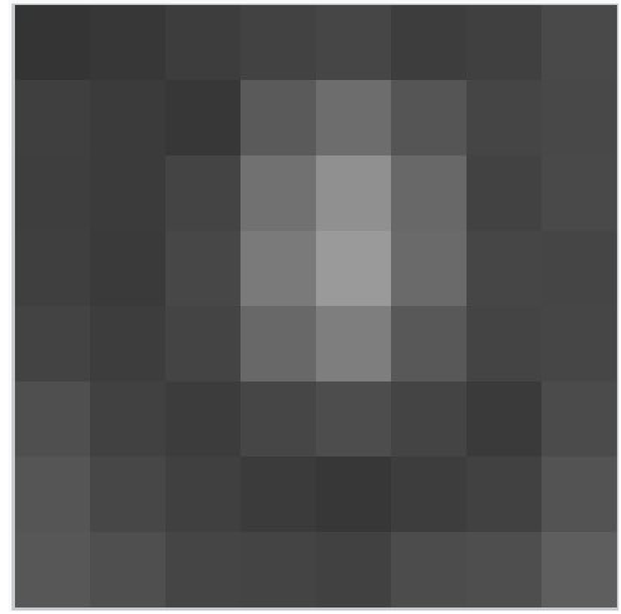


Širina bina = broj nivoa sivog / broj binova

Ekvalizacija histograma

8-bit grayscale image

$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$



Korak 1

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Piksel vrednosti 52 se pojavljuje 1 put
Piksel vrednosti 55 se pojavljuje 3 puta
Piksel vrednosti 58 se pojavljuje 2 puta
...
Piksel vrednosti 154 se pojavljuje 1 put

Preuzeto sa linka

Ekvalizacija histograma

Korak 2 Korak 3

1+2+3=6

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

v, Pixel Intensity	cdf(v)	h(v), Equalized v
52	1	0
55	4	12
58	6	20
59	9	32
60	10	36
61	14	53
62	15	57
63	17	65
64	19	73
65	22	85
66	24	93
67	25	97
68	30	117
69	33	130
70	37	146

...

$$h(v) = \mathrm{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

cdf –cumulative distribution function
(funkcija raspodele)
cdf_{min} – minimum različit od nule
M, N – dimenzije matrice (ovde: M=N=8)
L=broj nivoa sivog (obično L=256)

Ekvalizacija histograma

8-bit grayscale image

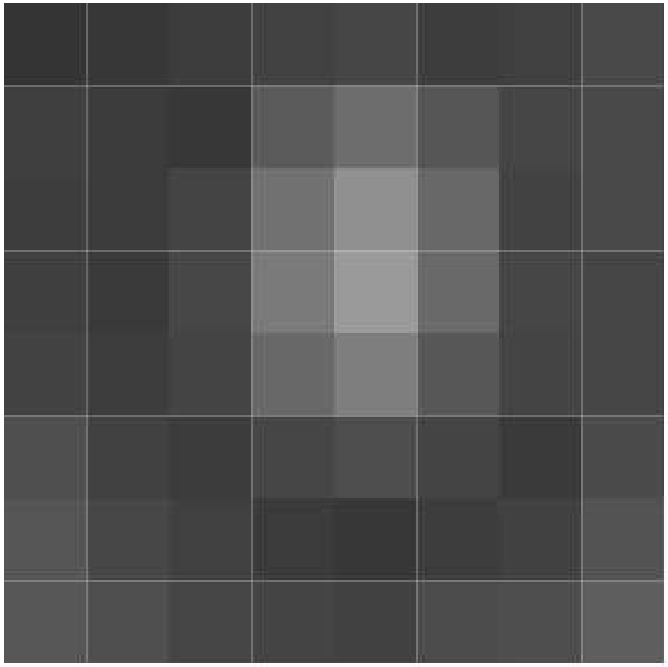
$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

Korak 4

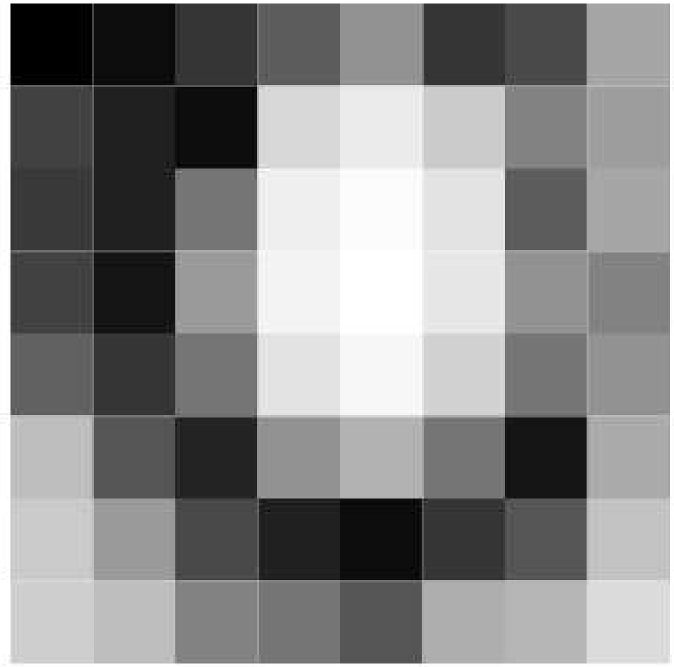
$$\begin{bmatrix} 0 & 12 & 53 & 32 & 146 & 53 & 174 & 53 \\ 57 & 32 & 12 & 227 & 219 & 202 & 32 & 154 \\ 65 & 85 & 93 & 239 & 251 & 227 & 65 & 158 \\ 73 & 146 & 146 & 247 & 255 & 235 & 154 & 130 \\ 97 & 166 & 117 & 231 & 243 & 210 & 117 & 117 \\ 117 & 190 & 36 & 190 & 178 & 93 & 20 & 170 \\ 130 & 202 & 73 & 20 & 12 & 53 & 85 & 194 \\ 146 & 206 & 130 & 117 & 85 & 166 & 182 & 215 \end{bmatrix}$$

v, Pixel Intensity	cdf(v)	h(v), Equalized v
52	1	0
55	4	12
58	6	20
59	9	32
60	10	36
61	14	53

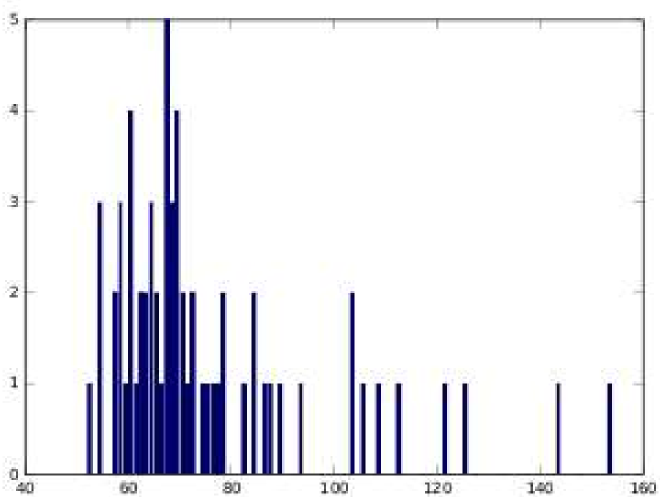
Ekvalizacija histograma



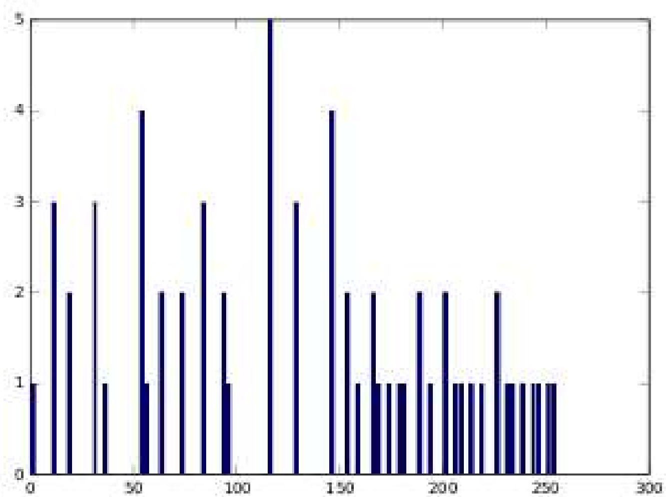
Original



Equalized



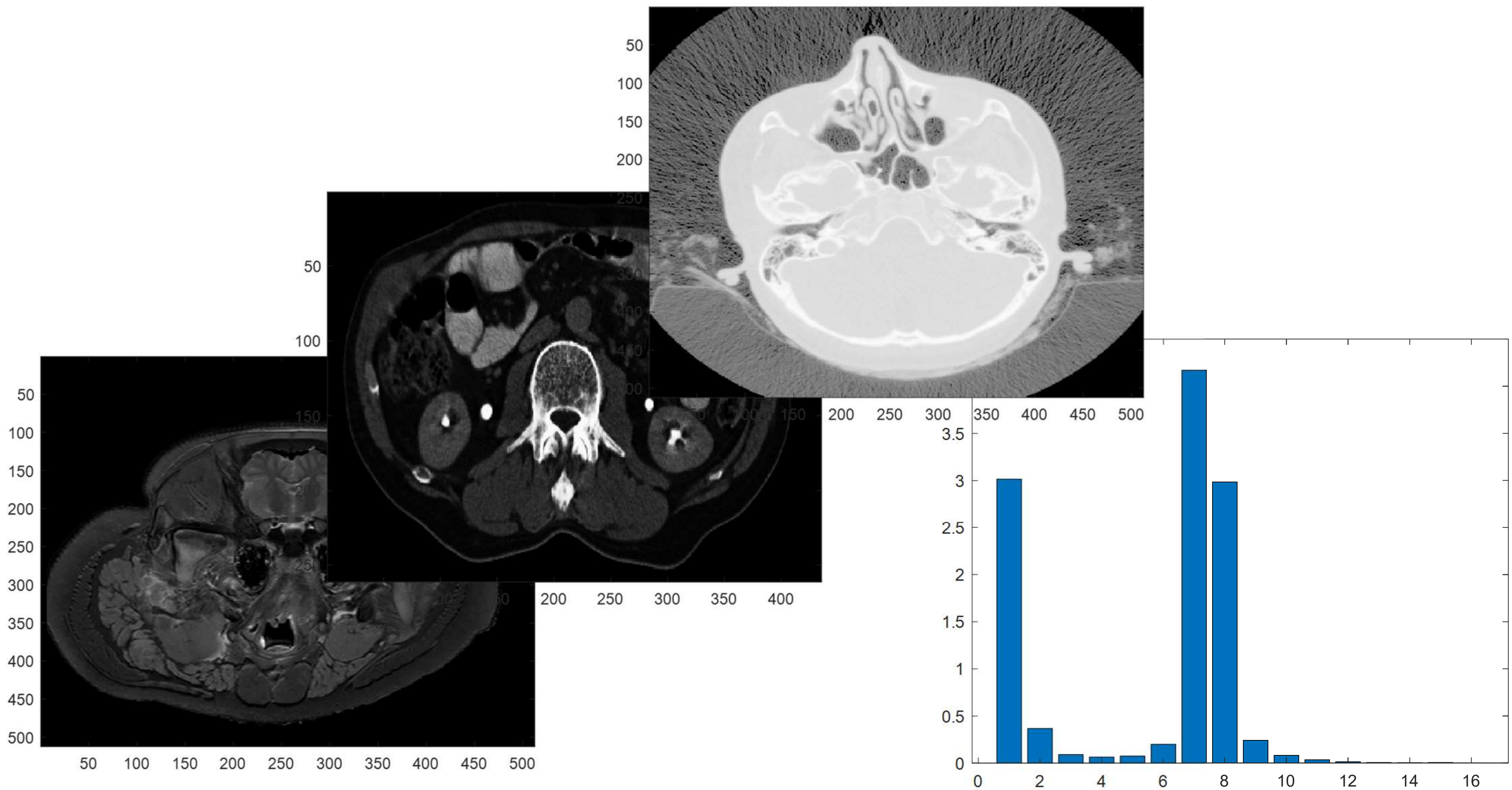
Histogram of Original image



Histogram of Equalized image

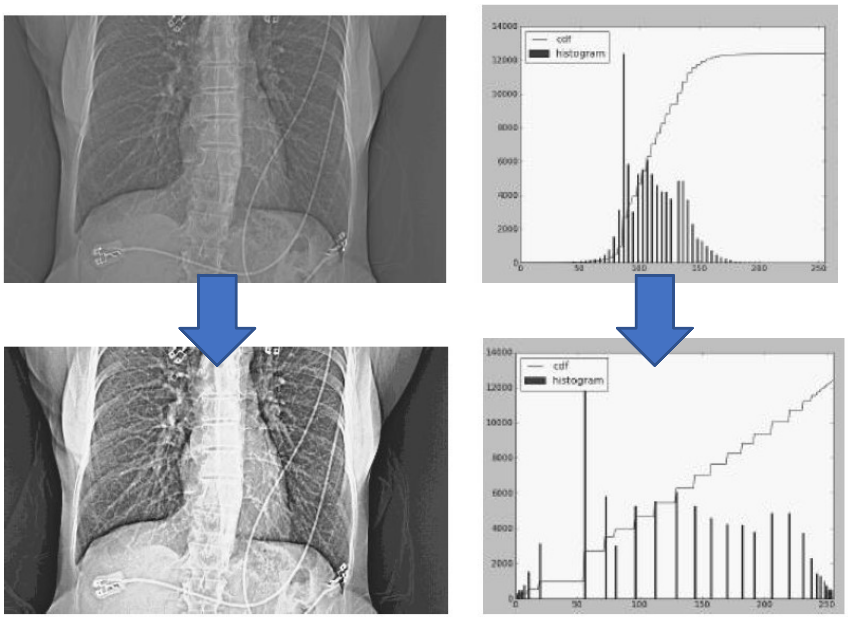
Rad prema uputstvu u Matlab/Octave

- Slediti korake date u uputstvo_Prostor_intenziteta.pdf iz direktorijuma primeri_Matlab_prostor_intenziteta.



Ekvalizacija – Python zadatak 1/1

Implementirati ekvalizaciju histograma u Python-u (mogu se koristiti pomoćne instrukcije sa ovog slajda) i primeniti je na rendgenski snimak pluća hequalization_input.png:
direktorijum prakticni_deo_za_samostalni_rad_Python.



Početak:

```
import numpy as np
import imageio
from PIL import Image
from matplotlib import pyplot as plt

img = Image.open('hequalization_input.png').convert('L')
img.show()

img1 = np.asarray(img)
```

Pomoćne instrukcije:

np.flatten() – transformacija matrice u niz

np.histogram() – kreiranje histograma sa 256 binova, opseg 0 do 255

cumsum() – računanje funkcije raspodele
np.ma.masked_equal() – maskiranje elemenata niza koji imaju vrednost 0

np.ma.filled().astype('uint8') – popunjavanje nulom maskiranih vrednosti

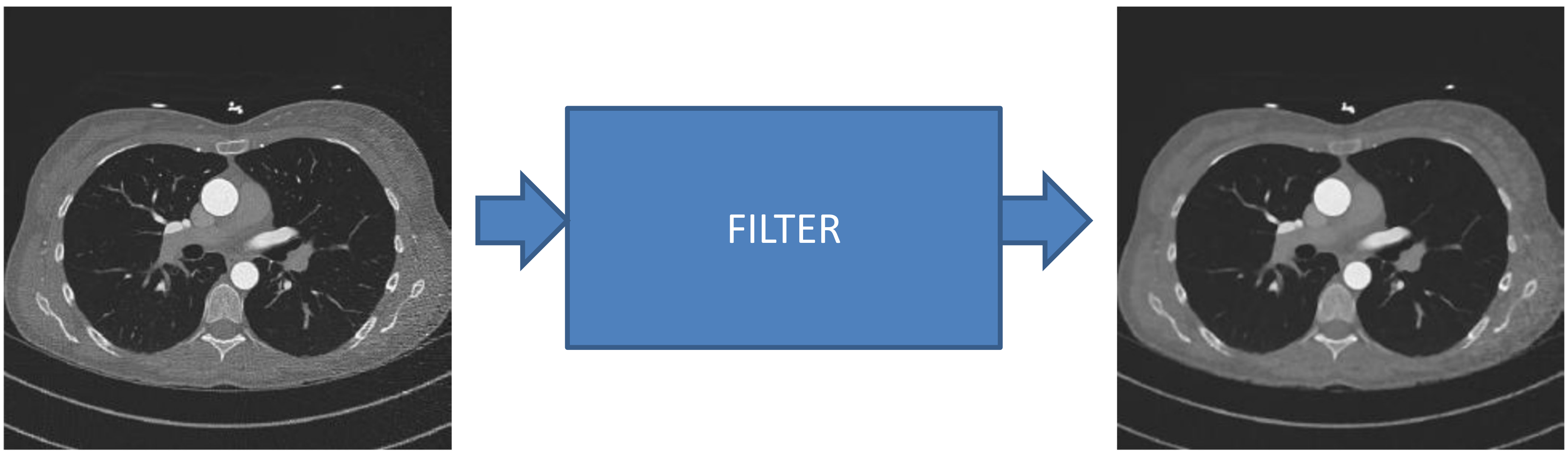
np.reshape() – promena dimenzija matrice

np.asarray() – konverzija 2D u image

show() – prikaz slike

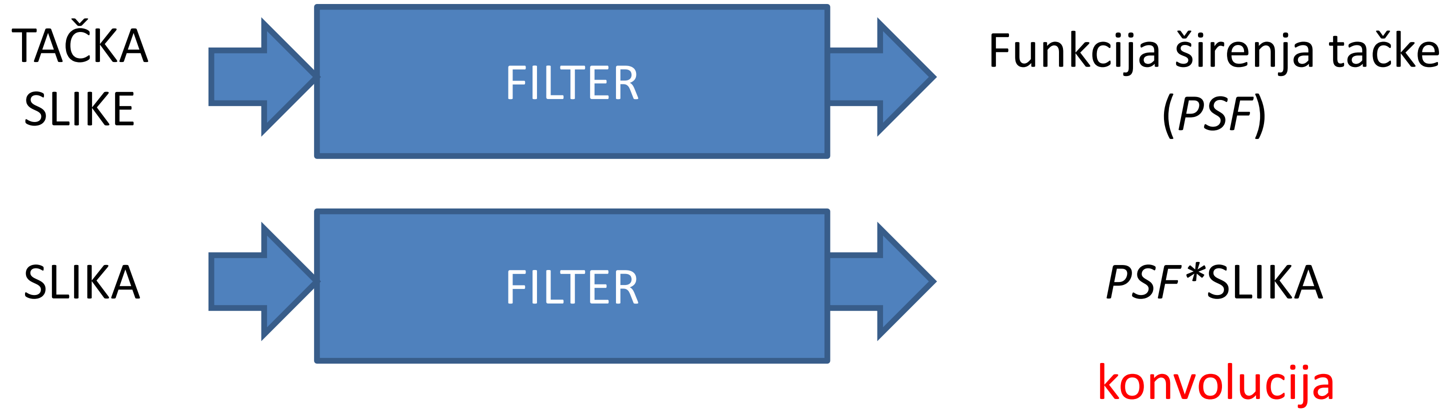
Filtriranje

- Filter je funkcija koja modifikuje ulazni signal



- Prostorno i frekvencijsko filtriranje
- Linearni i nelinearni filtri

Funkcija širenja tačke (*Point Spread Function, PSF*)



- Primena *PSF* na svaki piksel pojedinačno, utiče i na okolne piksele
- Kernel – mala matrica (*PSF*)

Prostorno filtriranje

- Konvolucija u prostornom domenu

F_1	F_2	F_3
F_4	F_5	F_6
F_7	F_8	F_9

kernel (filter)

$I(i-1,j-1)$	$I(i-1,j)$	$I(i-1,j+1)$
$I(i,j-1)$	$\boxed{I(i,j)}$	$I(i,j+1)$
$I(i+1,j-1)$	$I(i+1,j)$	$I(i+1,j+1)$

deo slike oko *I (i, j)* piksela

$$I_{new}(i,j) = F_1 * I(i-1,j-1) + F_2 * I(i-1,j) + F_3 * I(i-1,j+1)$$
$$+ F_4 * I(i,j-1) + F_5 * I(i,j) + F_6 * I(i,j+1)$$
$$+ F_7 * I(i+1,j-1) + F_8 * I(i+1,j) + F_9 * I(i+1,j+1)$$

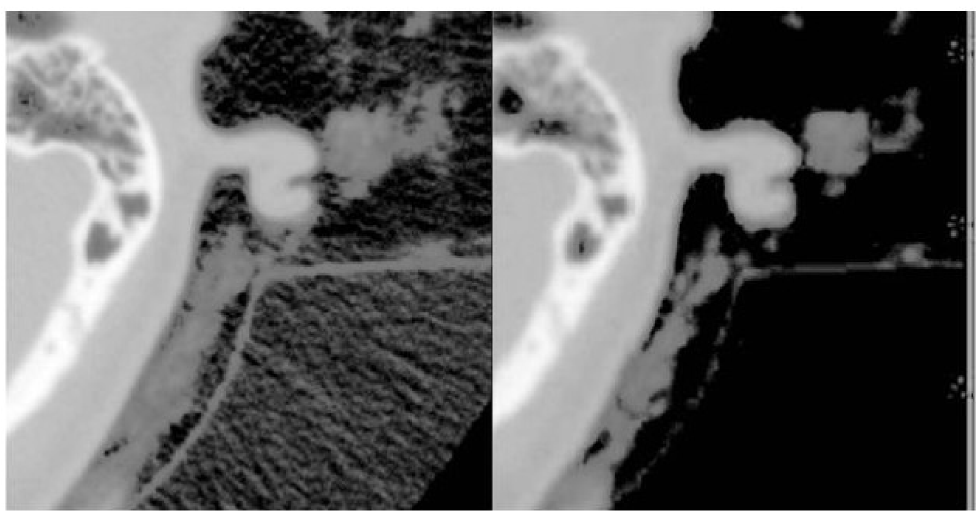
Prostorno filtriranje – linearni filtri

- Zamagljivanje (*smoothing*) – “*moving average*”

$$K_{\text{blur}} = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- Numerička aproksimacija dvodimenzionalne Gausove krive
- Koristi se za supresiju šuma

ulazna slika



“zamagljena” slika

Prostorno filtriranje – linearni filtri

- Zamagljivanje (*smoothing*) - *Gaussian blur*

$$K_{5\times5\,\text{Gauss}} = \frac{1}{256}\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 16 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- Izoštravanje slike (*Unsharp masking*)

$$K_{\text{Unsharp Mask}} = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Unity operator}} - w * K_{\text{blur}}$$

Adaptivno filtriranje: sama slika utiče na filtriranje

- Izoštravanje slike (*Sharpening*)

$$K_{\text{sharp}} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Koliko povezanih suseda koristi K_{sharp}?
8 (*eight-connected*)

Prostorno filtriranje – linearni filtri

- Numeričko diferenciranje – detekcija ivica

Forward diferenciranje

$$\frac{df(x)}{dx} = \frac{\rho_{i+1} - \rho_i}{\Delta x} \Rightarrow K_{\text{x-forward}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Centralno diferenciranje:

$$\frac{df(x)}{dx} = \frac{1}{2}\left(\underbrace{\rho_{i+1} - \cancel{\rho_i}}_{\text{Forward }\Delta} + \underbrace{\cancel{\rho_i} - \rho_{i-1}}_{\text{Backward }\Delta}\right)$$

$\Delta x = 1$

$$\Rightarrow K_{\text{x-central}} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\Rightarrow K_{\text{y-central}} = \frac{1}{2}\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Totalno diferenciranje:

$$\Rightarrow \sqrt{(K_{\text{x-central}} \star I(x,y))^2 + (K_{\text{y-central}} \star I(x,y))^2}$$

Sobel kernel: među najboljima za detekciju ivica

$$\underbrace{\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}}_{\text{1D Gaussian filter}} * \underbrace{\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}}_{\text{x-derivative}} \Rightarrow K_{\text{Sobel}_x} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\underbrace{\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}}_{\text{y-derivative}} * \underbrace{\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}}_{\text{1D Gaussian filter}} \Rightarrow K_{\text{Sobel}_y} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Prostorni filtri – granične vrednosti?

Dopuna za 3x5 filter?

- Dopuna nulama
- Dopuna konstantom
- Dopuna najbližim susedom
- Reflektovanje poslednjeg reda ili kolone

0	2	5	7	3	10	9
11	1	4	6	8	2	0
0	12	10	9	7	4	5
1	9	7	8	13	11	0
5	10	14	6	2	1	1
7	6	11	3	13	8	4
3	9	6	12	7	10	5

(a) A 7-by-7 input image.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	5	7	3	10	9	0	0
0	0	11	1	4	6	8	2	0	0	0
0	0	0	12	10	9	7	4	5	0	0
0	0	1	9	7	8	13	11	0	0	0
0	0	5	10	14	6	2	1	1	0	0
0	0	7	6	11	3	13	8	4	0	0
0	0	3	9	6	12	7	10	5	0	0
0	0	0	0	0	0	0	0	0	0	0

(b) Padding with zeros.

5	5	5	5	5	5	5	5	5	5	5
5	5	0	2	5	7	3	10	9	5	5
5	5	11	1	4	6	8	2	0	5	5
5	5	0	12	10	9	7	4	5	5	5
5	5	1	9	7	8	13	11	0	5	5
5	5	5	10	14	6	2	1	1	5	5
5	5	7	6	11	3	13	8	4	5	5
5	5	3	9	6	12	7	10	5	5	5
5	5	5	5	5	5	5	5	5	5	5

(c) Padding with a constant.

0	0	0	2	5	7	3	10	9	9	9
0	0	0	2	5	7	3	10	9	9	9
11	11	11	1	4	6	8	2	0	0	0
0	0	0	12	10	9	7	4	5	5	5
1	1	1	9	7	8	13	11	0	0	0
5	5	5	10	14	6	2	1	1	1	1
7	7	7	6	11	3	13	8	4	4	4
3	3	3	9	6	12	7	10	5	5	5
3	3	3	9	6	12	7	10	5	5	5

(d) Padding with nearest neighbor.

2	0	0	2	5	7	3	10	9	9	10
2	0	0	2	5	7	3	10	9	9	10
1	11	11	1	4	6	8	2	0	0	2
12	0	0	12	10	9	7	4	5	5	4
9	1	1	9	7	8	13	11	0	0	11
10	5	5	10	14	6	2	1	1	1	1
6	7	7	6	11	3	13	8	4	4	8
9	3	3	9	6	12	7	10	5	5	10
9	3	3	9	6	12	7	10	5	5	10

(e) Padding with reflect option.

Prostorno filtriranje – linearni filtri

- Primer 01 - Implementirati sledeće filtre:
  - 5x5 *Gaussian blur* filter
  - K_{sharp} – izoštravanje
  - K_x – *forward* diferenciranje
  - $K_{sobel\ x}$, $K_{sobel\ y}$, $K_{sobel\ centralno}$ – detekcija ivica
  - $K_{unsharp}$ – izoštravanje

i primeniti ih na slici u SKULLBASE.dcm datoteci.

Kompletno rešenje je u: *p01_linearni_prostorni_filtri.m*

```
clear all
close all

fp=fopen('SKULLBASE.DCM', 'r'); %%%%Može se koristiti i dicomread funkcija u Matlabu (za Octave je neophodno instalirati dodatnu dicom biblioteku)
fseek(fp,1622,'bof');
img=zeros(512,512);
img(:)=fread(fp,(512*512),'short');
img=transpose(img);
fclose(fp);

figure(1); imshow(img,[]); title ('Original')

%% 5x5 Gaussian blur
Kern_Gaus = [1 4 6 4 1; 4 16 24 16 4; 6 24 36 24 16; 4 16 24 16 4; 1 4 6 4 1]/256;
img_Gaus = zeros(512,512);
for i = 3:510
    for j = 3:510
        for cnt1 = -2:2
            for cnt2 = -2:2
                img_Gaus(i,j) = img_Gaus(i,j) + img(i+cnt1,j+cnt2)*Kern_Gaus(cnt1+3,cnt2+3);
            end
        end
    end
end
figure(2); imshow(img_Gaus,[]); title('5x5 Gaussian blur')
```

$$K_{5\times5\,\text{Gauss}} = \frac{1}{256}\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 16 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Prostorno filtriranje – nelinearni filtri

- Median filter

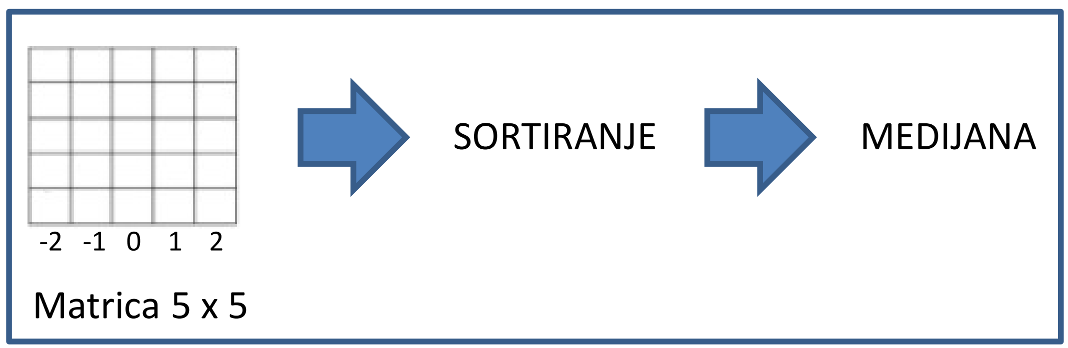
Šta je medijana za niz: 1, 9, 6, 3, 7, 3, 8?
Sortirati niz: 1, 3, 3, **6**, 7, 8, 9. Medijana je 6.

Šta je medijana za niz: 1, 9, 3, 8, 2, 4, 6, 5?
Sortirati niz: 1, 2, 3, **4, 5**, 6, 8, 9. Medijana je (4+5)/2=4.5

- Primer 02 - Implementirati median filtar i primeniti ga na slici u SKULLBASE.dcm datoteci.

```
mfimg=zeros(512,512);
rhovect = zeros(25,1);
for i=3:510
for j=3:510
idx = 1;
for k = -2:2
for l = -2:2
rhovect(idx)=img((i+k),(j+l));
idx = idx + 1;
end
end
rhovect=sort(rhovect);
mfimg(i,j) = rhovect(13,1);
end
end
```

Kompletno rešenje je u:
p02_nelinearni_prostorni_median_filter.m



Furijeova transformacija

2D Furijeova transformacija:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i2\pi(ux+vy)} dx\, dy$$

Inverzna 2D Furijeova transformacija:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{i2\pi(ux+vy)} du\, dv$$

Diskretna 2D Furijeova transformacija:

$$F(u,v) = \frac{1}{LK}\sum_{x=0}^{L-1}\sum_{y=0}^{K-1} f(x,y) e^{-i2\pi\left(\frac{ux}{L}+\frac{vy}{K}\right)}$$

Diskretna inverzna 2D Furijeova transformacija:

$$f(x,y) = \sum_{u=0}^{L-1}\sum_{v=0}^{K-1} F(u,v) e^{i2\pi\left(\frac{ux}{L}+\frac{vy}{K}\right)}$$

Magnituda 2D Furijeove transformacije:

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

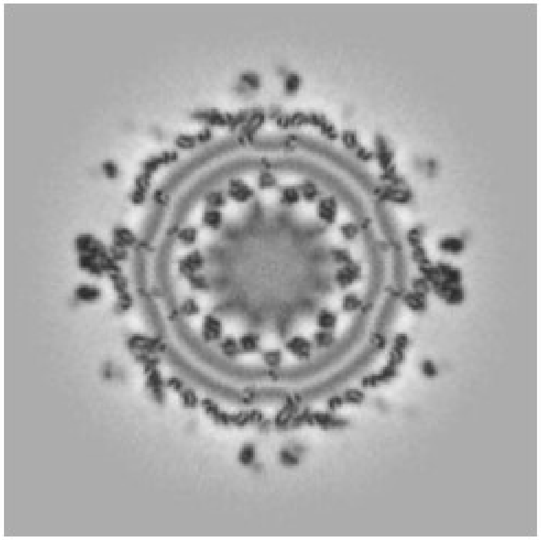
Faza 2D Furijeove transformacije:

$$\theta(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

Snaga 2D Furijeove transformacije:

$$P(u,v) = R^2(u,v) + I^2(u,v) = |F(u,v)|^2$$

Sindbis virus

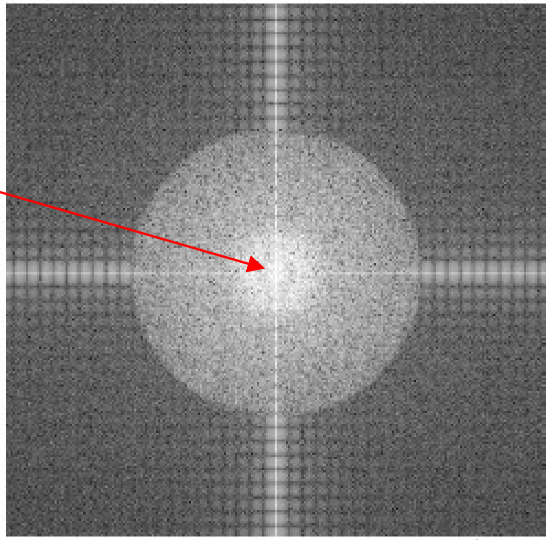


(x,y)

FFT, logaritamski skaliran

F (0,0)

Srednja vrednost svih piksela (najsvetlija tačka)



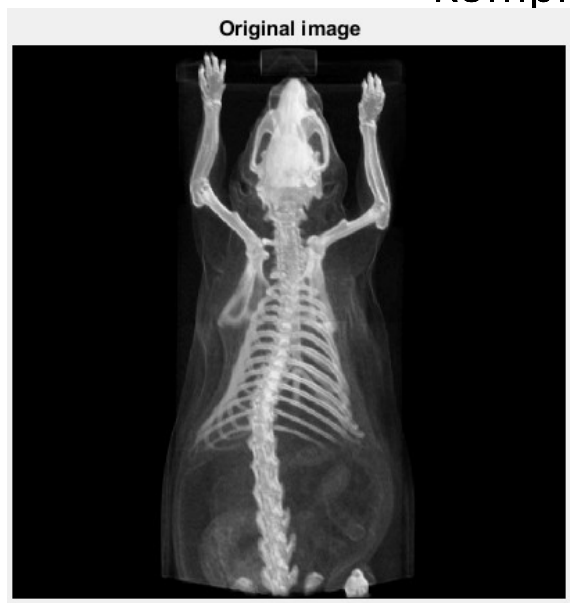
(u,v)

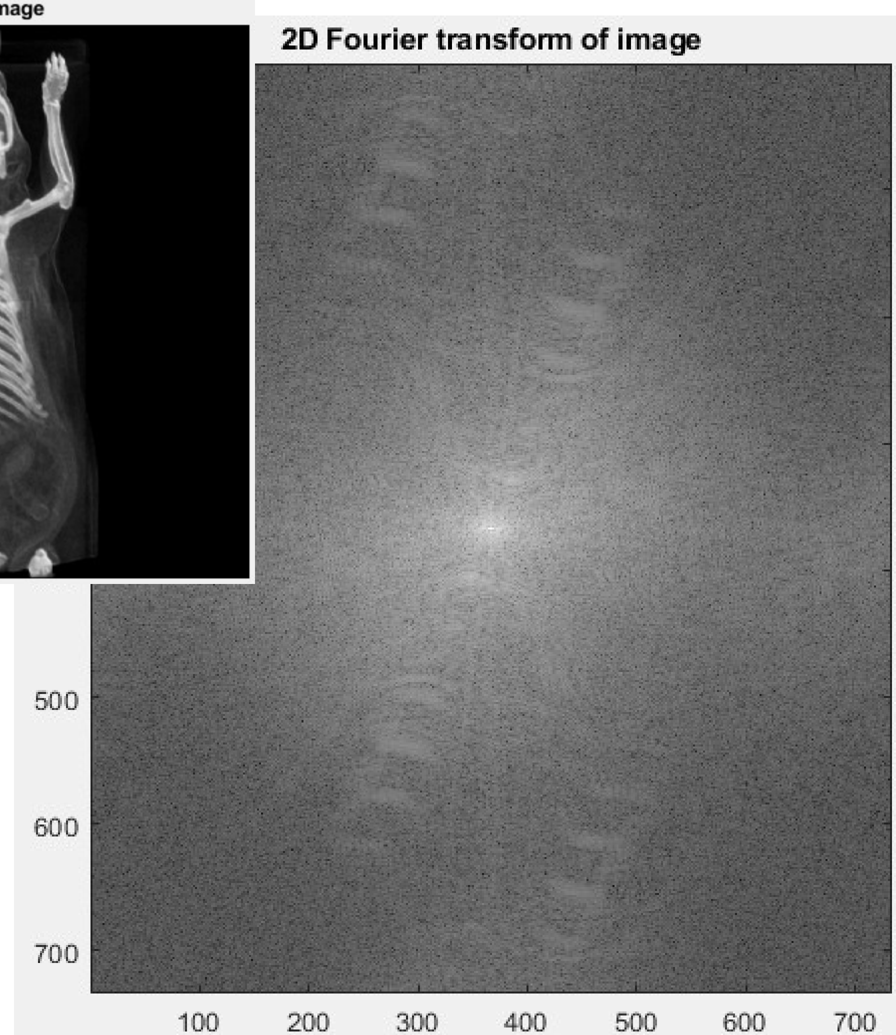
Furijeova transformacija

- Primer 03 – Za sliku *MouseCT.jpg* prikazati izgled Furijeove transformacije (logaritamski skaliran).

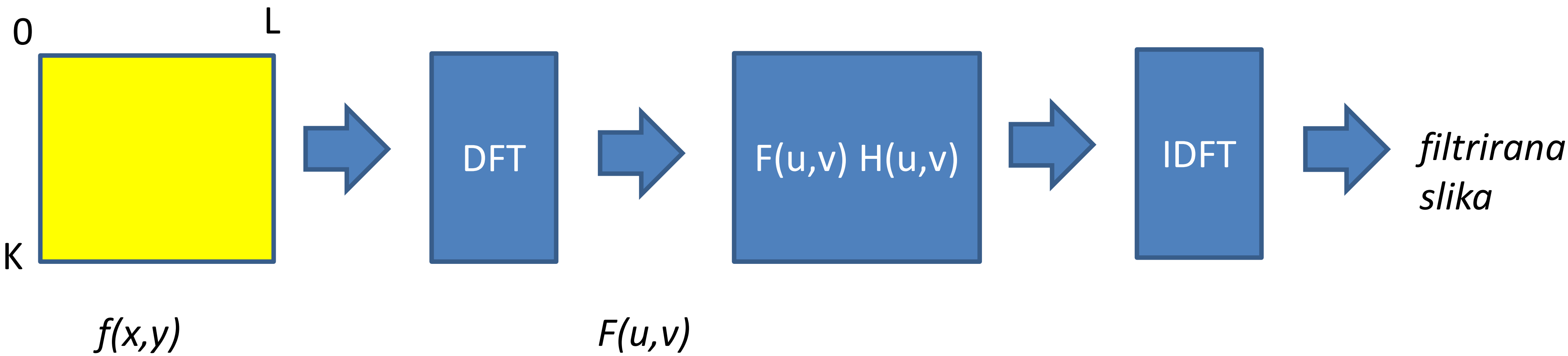
Kompletno rešenje je u: *p03_fft.m*

```
close all
clear all
img = imread('MouseCT.jpg');

figure(1)
subplot(1,2,1);
imshow(img)
title ('Original image')

fimg = fft2(img);
fimg_abs=abs(fimg);
fimg_shift = fftshift(fimg_abs);
f=log(fimg_shift);

minint=min(min(f));
f=f-minint;
maxint=max(max(f));
f=f/maxint*64;

subplot(1,2,2);
image(f)
title ('2D Fourier transform of image')
colormap gray
```





Frekvencijsko filtriranje



Niskopropusni filtri:

Idealni filter:

Euklidsko rastojanje od koordinatnog početka

$$d(u,v) = \sqrt{(u-L/2)^2 + (v-K/2)^2}$$

$$H(u,v) = \begin{cases} 1, & \text{if } d(u,v) \le d_0 \\ 0, & \text{else} \end{cases}$$

Batervortov filter:

$$H(u,v) = \frac{1}{1 + \left(\frac{d(u,v)}{d_0}\right)^2}$$

Gausov filter:

$$H(u,v) = e^{\frac{-d^2(u,v)}{2d_0^2}}$$

Visokopropusni filtri: $H_{VF} = 1 - H_{NF}$

Idealni filter:

$$H(u,v) = \begin{cases} 0, & \text{if } d(u,v) \le d_0 \\ 1, & \text{else} \end{cases}$$

Batervortov filter:

$$H(u,v) = \frac{1}{1 + \left(\frac{d_0}{d(u,v)}\right)^{2n}}$$

Gausov filter:

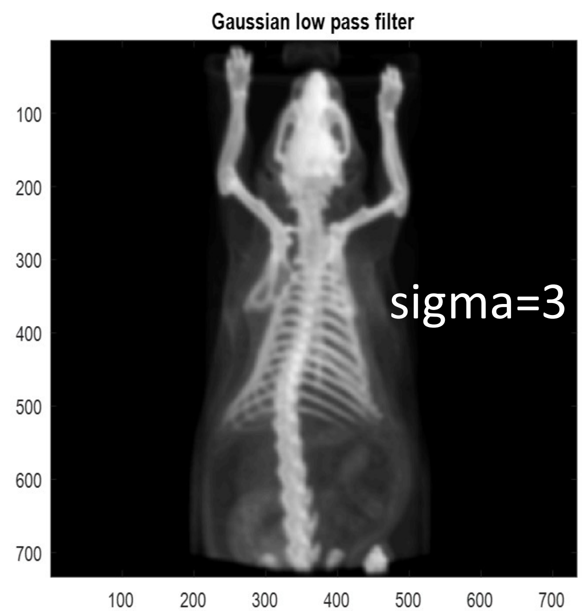
$$H(u,v) = 1 - e^{\frac{-d^2(u,v)}{2d_0^2}}$$

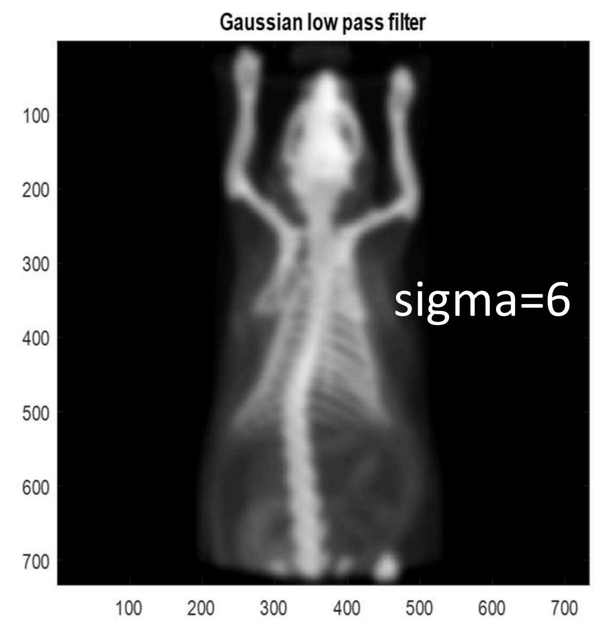
Frekvencijsko filtriranje

- Primer 04 – Implementirati Gausov niskofrekvencijski filter u frekvencijskom domenu i primeniti ga na *MouseCT.jpg* sliku za različite vrednosti *sigma*.

Kompletno rešenje je u:
p04_frekvencijsko filtriranje.m

```
fimg = fftshift(fft2(img));
              L    K
gs=zeros(733,733);
sigma=3;
for j=1:733
for k=1:733
gs(j,k)=exp(-((j-366)^2+(k-366)^2)/(2*sigma^2));
end
end

gs=fftshift(fft2(gs));

fimg_filt=gs.*fimg;

cimg=ifftshift(ifft2(fimg_filt));
```

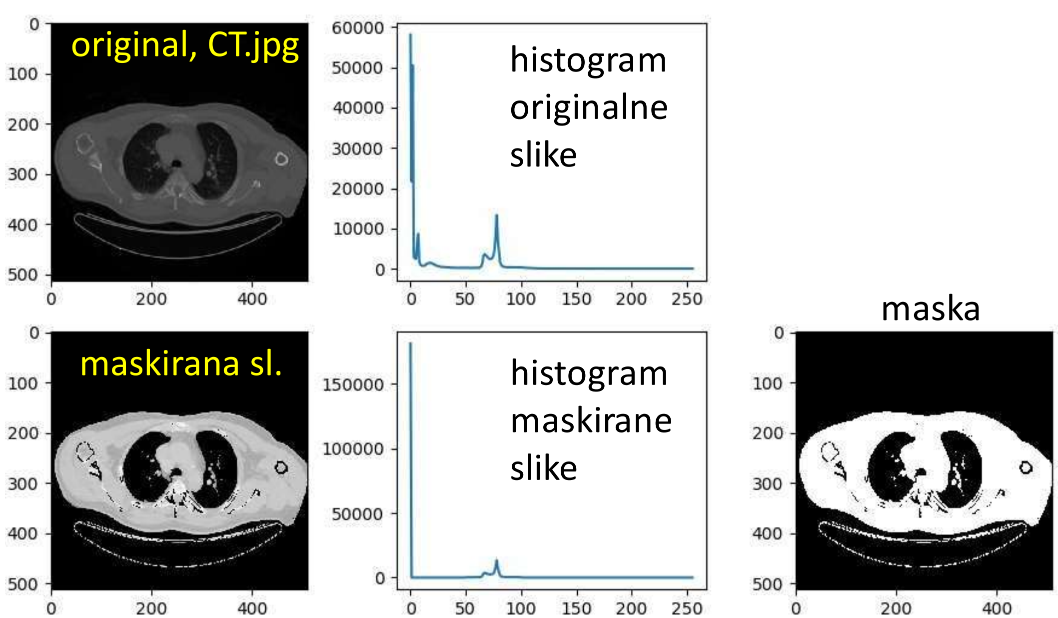




Filtriranje – Python zadatak 1/3

```
import imageio
import matplotlib.pyplot as plt
import numpy as np
import scipy.ndimage as ndi
```

Zadatak 1 Maskirati piksele intenziteta izmedju 50 i 100. Prikazati histograme originalne slike i maskirane slike. Prikazati sliku maske.



Čitanje

imageio.imread()

Histogram

ndi.histogram()

Maskiranje

mask= logički uslov

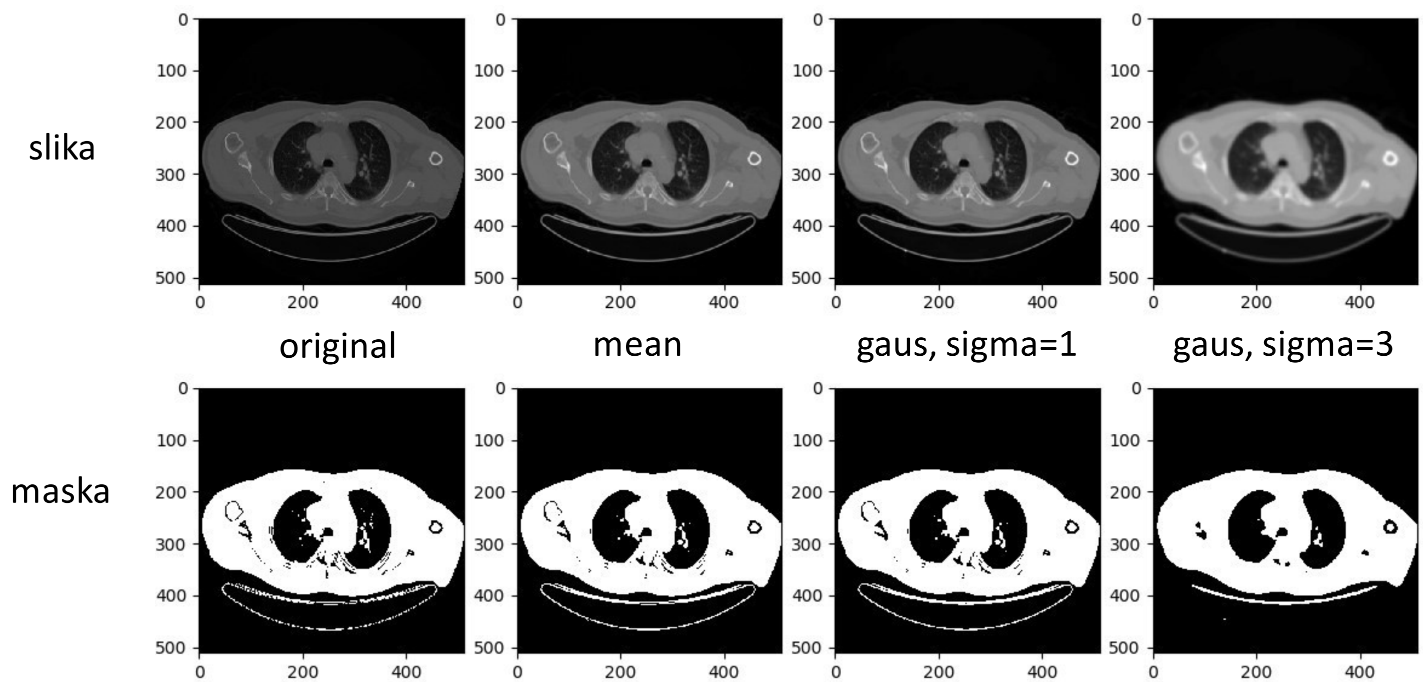
im_mask=np.where (mask,im,0)

Filtriranje – Python zadatak 2/3

Zadatak 2 Prikazati originalnu sliku "CT.jpg", kao i slike nastale filtriranjem *mean* filtrom i gausovim filtrom (za sigma=1 i za sigma=3). Prikazati i odgovarajuće maske definisane kao u ***Zadatku 1***.

Filtriranje: ndi.convolve (svi težinski koeficijenti su 1/9=0.11)

ndi.gaussian_filter



Filtriranje – Python zadatak 3/3

Zadatak 3 Detektovati horizontalne ivice, vertikalne ivice, gradijent magnitude (Pitagorina teorema primenjena na horizontalne i vertikalne ivice) na masku dobijenu gausovim filtrom (za sigma=3) u ***Zadatku 2***.

Filtriranje: ndi.convolve, ndi.sobel

Podsetnik: težinski koeficijenti filtra za horizontalne ivice

```
[[+1, +1, +1],
 [0, 0, 0],
 [-1, -1, -1]]
```