**DERS 8**

**Noise\_example**

clear; clc;

I = imread("afghan\_girl.jpeg");

J = I;

p3 = 0.09; %default

b = J;

x = rand(size(b));

d = x < p3/2;

b(d) = 0; % Minimum value

d = find(x >= p3/2 & x < p3);

b(d) = 1; % Maximum (saturated) value

I2=b;

I3=b;

[M,N] = size(I2);

for i=2:M-1

for j=2:N-1

%ort = (J(i-1,j-1)+J(i-1,j)+J(i-1,j+1)+...

% J(i,j-1)+J(i,j+1)+...

% J(i+1,j-1)+J(i+1,j)+J(i+1,j+1)/8);

ort = uint8(mean([I2(i-1,j-1),I2(i-1,j),I2(i-1,j+1),I2(i,j-1),I2(i,j+1),I2(i+1,j-1),I2(i+1,j),I2(i+1,j+1)]));

I3(i,j)=ort;

end

end

subplot(1,3,1);

imshow(I);

subplot(1,3,2);

imshow(b);

subplot(1,3,3);

imshow(I3);

**Noise\_w\_median**

clear; clc;

I = imread("afghan\_girl.jpeg");

J = I;

p3 = 0.09; %default

b = J;

x = rand(size(b));

d = x < p3/2;

b(d) = 0; % Minimum value

d = find(x >= p3/2 & x < p3);

b(d) = 1; % Maximum (saturated) value

I2=b;

I3=b;

[M,N] = size(I2);

for i=2:M-1

for j=2:N-1

%ort = (J(i-1,j-1)+J(i-1,j)+J(i-1,j+1)+...

% J(i,j-1)+J(i,j+1)+...

% J(i+1,j-1)+J(i+1,j)+J(i+1,j+1)/8);

ortanca = uint8(median([I2(i-1,j-1),I2(i-1,j),I2(i-1,j+1),I2(i,j-1),I2(i,j),I2(i,j+1),I2(i+1,j-1),I2(i+1,j),I2(i+1,j+1)]));

I3(i,j)=ortanca;

end

end

subplot(1,3,1);

imshow(I);

subplot(1,3,2);

imshow(b);

subplot(1,3,3);

imshow(I3);

**Noise\_w\_functions**

clear; clc;

I = imread("afghan\_girl.jpeg");

I2=gurultule(I,0.09);

I3 = medianFilter(I2);

I4= meanFilter(I2);

subplot(1,4,1);

imshow(I);

subplot(1,4,2);

imshow(I2);

subplot(1,4,3);

imshow(I3);

subplot(1,4,4);

imshow(I4);

Gurultule

function [b] = gurultule(I,noise)

p3=noise; %noise oranýný p3'e aldý.

b = I;

x = rand(size(b));

d = find( x < p3/2 );

b(d) = 0; % Minimum value

d = find( x >= p3/2 & x < p3 );

b(d) = 1; % Maximum (saturated) value

end

**MeanFilter**

function [I3] = meanFilter(J)

I3 = J;

[M,N] = size(J);

for i=2:M-1

for j=2:N-1

%ort = (J(i-1,j-1)+J(i-1,j)+J(i-1,j+1)+...

% J(i,j-1)+J(i,j+1)+...

% J(i+1,j-1)+J(i+1,j)+J(i+1,j+1)/8);

ort = uint8(mean([J(i-1,j-1),J(i-1,j),J(i-1,j+1),J(i,j-1),J(i,j),J(i,j+1),J(i+1,j-1),J(i+1,j),J(i+1,j+1)]));

I3(i,j)=ort;

end

end

end

**MedianFilter**

function [I3] = medianFilter(J)

I3 = J;

[M,N] = size(J);

for i=2:M-1

for j=2:N-1

%ort = (J(i-1,j-1)+J(i-1,j)+J(i-1,j+1)+...

% J(i,j-1)+J(i,j+1)+...

% J(i+1,j-1)+J(i+1,j)+J(i+1,j+1)/8);

orta = uint8(median([J(i-1,j-1),J(i-1,j),J(i-1,j+1),J(i,j-1),J(i,j),J(i,j+1),J(i+1,j-1),J(i+1,j),J(i+1,j+1)]));

I3(i,j)=orta;

end

end

end

**DERS 10**

**Fog\_reducehaze**

clear; clc;

I = imread("fog.jpeg");

I2 = imreducehaze(I);

subplot(1,2,1);

imshow(I);

subplot(1,2,2);

imshow(I2);

**DERS 11**

**Video\_reader**

clear; clc;

v= VideoReader("keman.mp4");

H=v.Height;

W=v.Width;

s = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

s2 = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

k=1;

while hasFrame(v)

s(k).cdata = readFrame(v);

k=k+1;

end

kk=k;

%Ters Oynatma

for i=1:k-1

s2(kk-i).cdata = s(i).cdata;

end

movie(s);

**video\_reader2**

clear; clc;

v= VideoReader("keman.mp4");

H=v.Height;

W=v.Width;

s = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

s2 = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

k=1;

while hasFrame(v)

s(k).cdata = readFrame(v);

k=k+1;

end

kk=k;

%Ters Oynatma

for i=1:k-1

s2(kk-i).cdata = s(i).cdata;

end

movie(s);

**Gurultule**

function [b] = gurultule(I,noise)

p3=noise; %noise oranýný p3'e aldý.

b = I;

x = rand(size(b));

d = find( x < p3/2 );

b(d) = 0; % Minimum value

d = find( x >= p3/2 & x < p3 );

b(d) = 1; % Maximum (saturated) value

end

**MeanFilter**

function [I3] = medianFilter(J)

I3 = J;

[M,N] = size(J);

for i=2:M-1

for j=2:N-1

ort = int8(median([J(i-1,j-1),J(i-1,j),J(i-1,j+1),J(i,j-1),J(i,j),J(i,j+1),J(i+1,j-1),J(i+1,j),J(i+1,j+1)]));

I3(i,j) = ort;

end

end

end

**DERS 12**

**Video\_reader**

clear; clc;

v= VideoReader("flashTV.mp4");

H=v.Height;

W=v.Width;

s = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

s2 = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

s3 = struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);

k=1;

while hasFrame(v)

s(k).cdata = readFrame(v);

s2(k).cdata = s(k).cdata(50:150,50:150,:);

s3(k).cdata = meanFilter(s2(k).cdata,21);

k=k+1;

end

movie(s3);

**meanFİlter**

function [K] = meanFilter(J,Mask)

yMask = uint8((Mask+1)/2);

xMask = uint8( yMask-1);

K = J;

[M,N,C] = size(J);

for i=yMask : M-yMask

for j=yMask : N-yMask

ort = uint8(mean(mean(J(i-xMask:i+xMask,j-xMask:j+xMask,:))) );

K(i,j,:)=ort;

end

end

end

**ODEV 6**

**Histeq\_prog**

clear; clc;

I=imread('faded.jpeg');

I2 = imread('dark.jpeg');

I3=histeqFunc(I);

I4=histeqFunc(I2);

subplot(2,2,1);

imshow(I);

subplot(2,2,2);

imshow(I3);

imwrite(I3,"faded\_histeq.jpeg");

subplot(2,2,3);

imshow(I2);

subplot(2,2,4);

imshow(I4);

imwrite(I4,"dark\_histeq.jpeg");

**histeqFunc**

function [I2] = histeqFunc(I)

numofpixels=size(I,1)\*size(I,2);

I2=uint8(zeros(size(I,1),size(I,2)));

freq=zeros(256,1);

probf=zeros(256,1);

probc=zeros(256,1);

cum=zeros(256,1);

output=zeros(256,1);

%freq counts the occurrence of each pixel value.

%The probability of each occurrence is calculated by probf.

for i=1:size(I,1)

for j=1:size(I,2)

value=I(i,j);

freq(value+1)=freq(value+1)+1;

probf(value+1)=freq(value+1)/numofpixels;

end

end

sum=0;

no\_bins=255;

%The cumulative distribution probability is calculated.

for i=1:size(probf)

sum=sum+freq(i);

cum(i)=sum;

probc(i)=cum(i)/numofpixels;

output(i)=round(probc(i)\*no\_bins);

end

for i=1:size(I,1)

for j=1:size(I,2)

I2(i,j)=output(I(i,j)+1);

end

end

end

**ODEV 5**

**İmage\_noising**

clear; clc;

I = rgb2gray(imread("afghan\_girl.jpeg"));

I2 = noisingFunc(I,0.3); %noised image I2

I3 = meanFilter(I2); %denoised with meanFilter

I4 = gaussianFilter(I2);

subplot(2,2,1);

imshow(I);

subplot(2,2,2);

imshow(I2);

imwrite(I2,"noisedImage.jpeg");

subplot(2,2,3);

imshow(I3);

imwrite(I3,"meanFiltered.jpeg");

subplot(2,2,4);

imshow(I4);

imwrite(I4,"gaussianFiltered.jpeg");

**noisingFunc**

function [b] = noisingFunc(I,noise)

p3=noise; %noise oranýný p3'e aldý.

b = I;

x = rand(size(b));

d = find( x < p3/2 );

b(d) = 0; % Minimum value

d = find( x >= p3/2 & x < p3 );

b(d) = 1; % Maximum (saturated) value

end

**MeanFilter**

function [I3] = meanFilter(J)

I3 = J;

[M,N] = size(J);

for i=2:M-1

for j=2:N-1

ort = uint8(mean([J(i-1,j-1),J(i-1,j),J(i-1,j+1),J(i,j-1),J(i,j),J(i,j+1),J(i+1,j-1),J(i+1,j),J(i+1,j+1)]));

I3(i,j)=ort;

end

end

end

**GaussianFilter**

function [I5] = gaussianFilter(J)

I5 = J;

[M,N] = size(J);

for i=2:M-1

for j=2:N-1

gaussianOrt = uint8((J(i-1,j-1)+(2\*J(i-1,j))+J(i-1,j+1)+(2\*J(i,j-1))+(4\*J(i,j))+(2\*J(i,j+1))+J(i+1,j-1)+(2\*J(i+1,j))+J(i+1,j+1))/16);

I5(i,j) = gaussianOrt;

end

end

end