|  |  |  |  |
| --- | --- | --- | --- |
| **Class: M.Sc. (CS) Part1 SEM-I**  **Subject: Digital Signals and Image Processing**  **Index** | | | |
| **Sr.**  **No.** | **Title** | **Date** | **Signature** |
| 1.a. | 2D Linear Convolution |  |  |
| 1.b. | Circular Convolution between two 2D matrices |  |
| 2.a | DFT of 4x4 Gray scale image |  |  |
| 2. b. | Compute discrete cosine transform |  |
| 3. a. | Brightness enhancement of an image |  |  |
| 3. b. | Brightness suppression of an image |  |
| 4.a. | Contrast Manipulation |  |  |
| 4. b. | Image negative |  |
| 5. a. | Perform threshold operation |  |  |
| 5. b. | perform gray level slicing without background |  |
| 6.a. | Basic transformation  Log transformation |  |  |
| 6.b. | Power law transformation |  |
| 7.a. | WAP to display 3 images using figure () function with title. |  |  |
| 7.b | WAP to display 3 images using subplot () function with title. |  |
| 8 | Binary Image Processing |  |  |
| 8. a. | Image Dilation |  |
| 8. b. | Image Erosion |  |
| 8.c | Image Opening |  |  |
| 8.d | Image Closing |  |  |

Practical 1A

Aim: Implement 2D Linear Convolution in scilab

Code:

clc;

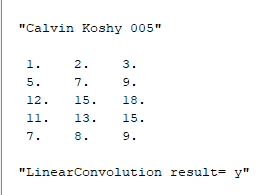
x=[1,2,3;4,5,6;7,8,9]

y=[1;1;1]

y=conv2(x,y);

disp('Calvin Koshy 005');

disp(y,'LinearConvolution result= y'



Practical 1B

implement Circular Convolution between two 2D matrices in scilab.

clc;

x=[1,2;3,4];

h=[5,6;7,8];

X= fft2(x);

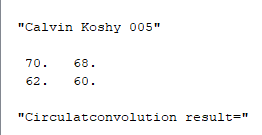
H= fft2(h);

Y=X.\*H;

y= ifft(Y);

disp('Calvin Koshy 005')

disp(y,'Circulatconvolution result=')



Practical 2A

Find DFT of 4x4 signal in scilab

clc;

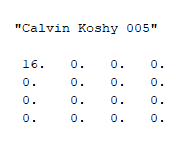
f=[1 1 1 1;1 1 1 1;1 1 1 1;1 1 1 1];

t= fft2(f);

disp('Calvin Koshy 005')

disp(t,'Display')

output:



Practical 2b:

Compute discrete cosine transform

clc;

f1=[2 4 4 2;4 6 8 3;2 8 10 4;3 8 6 2];

F1=dct(f1);

disp(F1,'Discrete Cosine Transform of f(m,n) using dct function');

subplot(221);

imshow(f1);

title('Image in spatial domain')

subplot(223);

imshow(F1);

title('Image in frequency domain')

output:

Practical 3a

Brightness enhancement of an image

clc;

a = imread("

f1 = f + 50

f2 = f + 50

f3 = f + 80

subplot(2,2,1);

imshow(f);

title("Original Image; (Calvin Koshy)");

subplot(2,2,2);

imshow(f1);

title('Brightness Increased by 50;(Calvin Koshy)');

subplot(2,2,3);

imshow(f2);

title('Brightness Increased by 60;(Calvin Koshy)');

subplot(2,2,4);

imshow(f1);

title('Brightness Increased by 80;(Calvin Koshy)');

Practical 3b

AimBrightness suppression of an image

clc;

a = imread("

f1 = f - 50

f2 = f - 50

f3 = f - 80

subplot(2,2,1);

imshow(f);

title("Original Image; (Calvin Koshy)");

subplot(2,2,2);

imshow(f1);

title('Brightness decreased by 50;(Calvin Koshy)');

subplot(2,2,3);

imshow(f2);

title('Brightness decreased by 60;(Calvin Koshy)');

subplot(2,2,4);

imshow(f1);

title('Brightness decreased by 80;(Calvin Koshy)');

Practical 4A

Contrast Manipulation(Contrast Streching)

a=imread('D:\KC IP Code\cameraman.png');

r1=100;

r2=140;

s1=150;

s2=240;

l=s1/r1;

m=(s2-s1)/(r2-r1);

n=(255-s2)/(255-r2);

s=size(a);

for i=1:s(1)

for j=1:s(2)

if ((a(i,j) > 0) && (a(i,j) < r1))

b(i,j) = a(i,j)\*l;

end

if ((a(i,j) > r1) && (a(i,j) < r2))

b(i,j) = (m\*(a(i,j)-120))+s1;

end

if ((a(i,j) > r2) && (a(i,j) < 256))

b(i,j) = (n\*(a(i,j)-150))+s2;

end

end

end

subplot(1,2,1),imshow(a),title('Original Image');

subplot(1,2,2),imshow(uint8(b)),title('Contrast Image');

Practical 4B

Image Negative

clc;

a1=imread('D:\KC IP Code\img.jpeg');

[m,n]=size(a1);

for i=1:m

for j=1:n

b1(i,j)=255 - a1(i,j);

end

end

subplot(1,2,1),imshow(a1),title('Original;(Cavin Koshy)');

subplot(1,2,2),imshow(b1),title('Negation;(Calvin Koshy)');

output:

Practical 5A:

Perform threshold Operation

a=imread("

[m,n]=size(a);

for i=1:m

for j=1:n

x=a(i,j);

if x >= 128

b(i,j)=255;

else

b(i,j)=0;

end

end

end

subplot(2,2,1),imshow(a),title('Original; Calvin Koshy')

subplot(2,2,2),imshow(b),title('Threshold Output; Calvin Koshy')

Practical 5B

perform gray level slicing

a=imread('D:\KC IP Code\cameraman.png');

[m,n]=size(a);

min = 100;

max= 200;

for i=1:m

for j=1:n

x=a(i,j);

if x > min && x < max

b(i,j)=a(i,j);

elseif x > max

b(i,j)=255;

else

b(i,j)=0;

end

end

end

subplot(1,2,1),imshow(a),title('Original');

subplot(1,2,2),imshow(b),title('Gray Slicing')

output:

Practical No.6-A : Basic transformation

a)Log transformation

a=imread('D:\KC IP Code\cameraman.png');

[m,n]=size(a);

for i=1:m

for j=1:n

x=a(i,j);

b(i,j)=20\*log(1+double(x));

end

end

subplot(1,2,1),imshow(a),title('Original;(Calvin Koshy)');

subplot(1,2,2),imshow(b),title('Log Transform,(Calvin Koshy)');

b) Power Law Transformation

a=imread('D:\KC IP Code\cameraman.png');

[m,n]=size(a);

for i=1:m

for j=1:n

x=double(a(i,j));

b(i,j)=20\*(x^0.4);

c(i,j)=20\*(x^0.6);

d(i,j)=20\*(x^0.9);

end

end

subplot(2,2,1),imshow(a),title('Original;(Calvin Koshy)');

subplot(2,2,2),imshow(b),title('Power Law Transform : Gamma = 0.4; (Calvin Koshy)');

subplot(2,2,3),imshow(c),title('Power Law Transform : Gamma = 0.6');

subplot(2,2,4),imshow(d),title('Power Law Transform : Gamma = 0.9');

Practical 7

WAP to display 3 images using figure () function with title

img1 = imread('D:\KC IP Code\cameraman.png');

img2 = imread('D:\KC IP Code\color.png');

img3 = imread('D:\KC IP Code\img.jpeg');

figure(1);

title('Image 1; Calvin Koshy');

imshow(img1);

figure(2);

title('Image 2'; Calvin Koshy);

imshow(img2);

figure(3);

title('Image 3'; Calvin Koshy);

imshow(img3);

practical 7b

WAP to display 3 images using subplot () function with title.

img1 = imread('D:\KC IP Code\cameraman.png');

img2 = imread('D:\KC IP Code\color.png');

img3 = imread('D:\KC IP Code\img.jpeg');

subplot(2,2,1);

title('Image 1'; Calvin Koshy);

imshow(img1);

subplot(2,2,2);

title('Image 2'; Calvin Koshy);

imshow(img2);

subplot(2,2,3);

title('Image 3'; Calvin Koshy);

imshow(img3);

Practical 8

Binary Image Processing operations 1)Erosion 2) Dilation 3) Opening and 4) Closing

a1=imread('D:\KC IP Code\color.png');

b1 = testmatrix('square',11);

outputimage = imerode(a1,b1);

subplot(1,2,1),imshow(a1),title('Original');

subplot(1,2,2),imshow(b1),title('Erosion Image');

Practical 8b

Dialation

a1=imread('D:\KC IP Code\color.png');

b1 = testmatrix('square',11);

outputimage = imdil(a1,b1);

subplot(1,2,1),imshow(a1),title('Original');

subplot(1,2,2),imshow(b1),title('Dilation');

Practical 8c

Opening

img=imread('D:\KC IP Code\cameraman.png');

se1 = testmatrix('square',11);

im2 = imerode(img,se1);

im3 = imdilate(im2,se1);

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

subplot(1,2,2),imshow(im3),title('opening image');

Practical 8d

Closing

img=imread('D:\KC IP Code\cameraman.png');

se1 = testmatrix('square',11);

im2 = imerode(img,se1);

im3 = imdilate(im2,se1);

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

subplot(1,2,2),imshow(im3),title('opening image');