

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

DIP LAB FILE

**Name: Akash Rajput**

**University Rollno. : 2115000099**

**Subject Name: Digital Image Processing Lab**

**Subject Code: BCSE 0101**

**Course: B.Tech.**

**Year: III**

**Semester: V**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Exp.**  **No.** | **Program** |  |
| 1 | Introduction with the MATLAB Command. |  |
| 2 | Read the image and display it. |  |
| 3 | Create the horizontal and vertical strips. |  |
| 4 | Create chessboard. |  |
| 5 | Read the colored image and check attributes. |  |
| 6 | Perform Log Transformation on Image. |  |
| 7 | Perform Negative Transformation on Image. |  |
| 8 | Perform Power Log Transformation on Image. |  |
| 9 | Perform Gray Level Slicing with and without background. |  |
| 10 | Perform Histogram Equalization on image. |  |
| 11 | Find Four, Diagonal and Eight Neighbours of image. |  |
| 12 | Perform Dilation on Image. |  |
| 13 | Perform Erosion on Image. |  |
| 14 | Perform Perforn Closing(Dilation Followed by Erosion) on Image. |  |

**Experiment -1**

**Objective:-** Introduction with the MATLAB Command.

**Commands:-**

1. clc - Clear Command window

2. clearall - clear the workspace

3. date - displays current date

4. linspace - creates regularly spaced rectory

5. Mat – returns largest element

6. Min - returns smallest element

7. size – computes array size

8. ones – creates an array of ones

9. zeros – creates an array of zeros

10. subplot – creates plot in subwindows.

**Experiment -2**

**Objective:-** Read the image and display it.

**Implement:-**

image = imread("cameraman.tif");

imshow(image)

**Output:-**



**Experiment -3**

**Objective:-** Create the horizontal and vertical strips.

**Implement:-**

arr = zeros(10,20);

%vertical strip

vertical = uint8(arr);

vertical(:,2:2:20) = 255;

subplot(1,2,1),imshow(vertical)

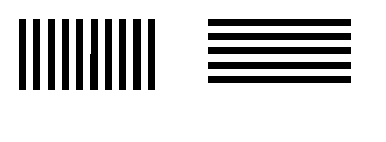
%horizontal strips

horizontal = uint8(arr);

horizontal(2:2:10,:) = 255;

subplot(1,2,2),imshow(horizontal)

**Output:-**



**Experiment -4**

**Objective:-** Create chessboard.

**Implement:-**

arr = zeros(8,8);

horizontal = uint8(arr);

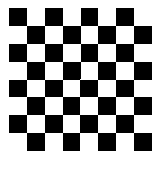
horizontal(2:2:8,:) = 255;

horizontal(:,2:2:8) = 255;

horizontal(2:2:8,2:2:8) = 0;

subplot(1,2,1),imshow(horizontal)

**Output:-**



**Experiment -5**

**Objective:-** Read the colored image and check attributes.

**Implement:-**

image = imRead ("pears.png");

R = image (:,:,1);

G = image(:,:,2);

B = image(:,:,3);

subplot(2,2,1);

imshow (image);

subplot(2,2,2);

imshow (R);

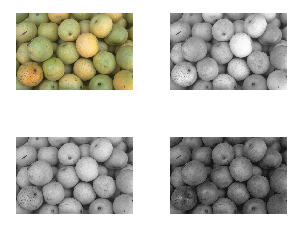
subplot(2,2,3);

imshow (G);

subplot(2,2,4);

imshow (B);

**Output:-**

****

**Experiment -6**

**Objective:-** Perform Log Transformation on Image.

**Implement:-**

image = imread('cameraman.tif');

subplot(1,2,1),

imshow(image),

title("original image");

r = double(image);

c = 40;

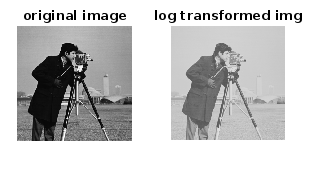
s = c \* log(1 + r);

image1 = uint8(s);

subplot(1,2,2), imshow(image1),

title("log transformed img");

**Output:-**



**Experiment -7**

**Objective:-** Perform Negative Transformation on Image.

**Implement:-**

image = imread('cameraman.tif');

subplot(1,2,1),

imshow(image),

title("original image");

image1=image;

for row=1:size(image,1)

for col=1:size(image,2)

image1(row,col)=255-image(row,col);

end

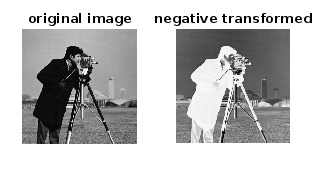
end

subplot(1,2,2),

imshow(image1),

title("negative transformed");

**Output:-**



**Experiment -8**

**Objective:-** Perform Power Log Transformation on Image.

**Implement:-**

image = imread('cameraman.tif');

subplot(1,2,1),

imshow(image),

title("original image");

r = double(image);

c = 100;

g = 0.5;

s = c \* log(1 + r)^g;

t = 255/(c\*log(256)^g);

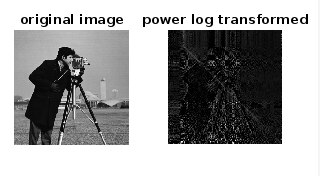
image1 = uint8(t\*s);

subplot(1,2,2),

imshow(image1),

title("power log transformed");

**Output:-**



**Experiment -9**

**Objective:-** Perform Gray Level Slicing with and without background.

**Implement:-**

image = imread('cameraman.tif');

LT = 50;

HT = 150;

image1 = image;

image1(image < HT & image > LT) = 255;

without\_back= image;

without\_back(image > HT | image < LT) = 255;

subplot(2, 2, 1),

imshow(image),

title('Original Image');

subplot(2, 2, 2),

imshow(image1),

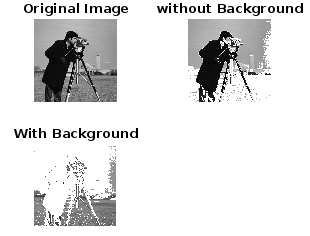
title('without Background');

subplot(2,2,3),

imshow(without\_back),

title('With Background');

**Output:-**



**Experiment -10**

**Objective:-** Perform Histogram Equalization on image.

**Implement:-**

image=imread('cameraman.tif');

[n,m]=size(image);

for i=1:256

t(i,1)=i-1;

t(i,2)=0;

end

for i=1:n

for j=1:m

pix=image(i,j)+1;

t(pix,2)=t(pix,2)+1;

end

end

for i=1:256

t(i,3)=t(i,2)/(256\*256);

end

t(1,4)=t(1,3)

for i=2:256

t(i,4)=t(i-1,4)+t(i,3);

end

for i=1:256

t(i,5)=t(i,4)\*255;

end

for i=1:256

t(i,6)=round(t(i,5));

end

for i=1:256s(i,1)=i-1;

s(i,2)=0;

end

for i=1:256

pix=t(i,6)+1;

s(pix,2)=s(pix,2)+t(i,2);

end

image2=image;

for i=1:n

for j=1:m

image2(i,j)=t(image(i,j)+1,6);

end

end

subplot(3,2,1),

imshow(image),

title('Original');

subplot(3,2,2),

imshow(image2),

title('Manual Transform');

subplot(3,2,3),

bar(t(:,1),t(:,2)),

title('Histogram');

subplot(3,2,4),

bar(s(:,1),s(:,2)),

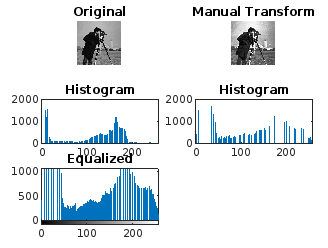
title('Histogram');

subplot(3,2,5),

imhist(image2),

title("Equalized");

**Output:-**

****

**Experiment -11**

**Objective:-** Find Four, Diagonal and Eight Neighbours of image.

**Implement:-**

image = [1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12; 13, 14, 15, 16];

x = input('X-axis: ');

y = input('Y-axis: ');

[n, m] = size(image);

fprintf('n: %d, m: %d\n', n, m);

FN = NaN(1, 4);

EN = NaN(1, 8);

DN = NaN(1, 4);

if x > 1

FN(1) = image(x - 1, y);

end

if y > 1

FN(2) = image(x, y - 1);

end

if x < n

FN(3) = image(x + 1, y);

end

if y < m

FN(4) = image(x, y + 1);

end

if x > 1 && y > 1

DN(1) = image(x - 1, y - 1);

end

if x < n && y < m

DN(2) = image(x + 1, y + 1);

end

if x < n && y > 1

DN(3) = image(x + 1, y - 1);

end

if x > 1 && y < m

DN(4) = image(x - 1, y + 1);

end

fprintf('Four Neighbors: %s\n', mat2str(FN));

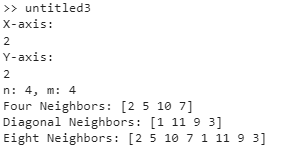
fprintf('Diagonal Neighbors: %s\n', mat2str(DN));

EN(1:4) = FN(1:4);

EN(5:8) = DN(1:4);

fprintf('Eight Neighbors: %s\n', mat2str(EN));

**Output:-**



**Experiment -12**

**Objective:-** Perform Dilation on Image.

**Implement:-**

image = imread('cameraman.tif')

image = im2bw(image);

image1 = logical(ones(3, 3));

[n, m] = size(image);

image2 = false(n, m);

for r = 2:n-1

for c = 2:m-1

if any(any(image(r-1:r+1, c-1:c+1) & image1))

image2(r, c) = true;

end

end

end

figure;

subplot(2, 2, 1),

imshow(image);

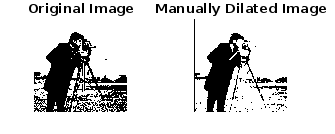
title('Original Image');

subplot(2, 2, 2),

imshow(image2);

title('Manually Dilated Image');

**Output:-**



**Experiment -13**

**Objective:-** Perform Erosion on Image.

**Implement:-**

image = imread('cameraman.tif');

image = im2bw(image);

image1 = logical(ones(3, 3));

[n, m] = size(image);

image2 = true(n, m);

for r = 2:n-1

for c = 2:m-1

if all(all(image(r-1:r+1, c-1:c+1) & image1))

image2(r, c) = true;

else

image2(r, c) = false;

end

end

end

subplot(1, 2, 1),

imshow(image);

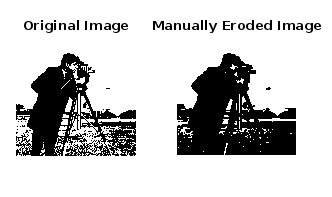
title('Original Image');

subplot(1, 2, 2),

imshow(image2);

title('Manually Eroded Image');

**Output:-**



**Experiment -14**

**Objective:-** Perform Perforn Closing(Dilation Followed by Erosion) on Image.

**Implement:-**

image = imread('cameraman.tif')

image = im2bw(image);

se = logical(ones(3, 3));

[n, m] = size(image);

D\_image = false(n, m);

subplot(2, 2, 1),

imshow(image),

title('Original Image');

for r = 2:n-1

for c = 2:m-1

if any(any(image(r-1:r+1, c-1:c+1) & se))

D\_image(r, c) = true;

end

end

end

subplot(2, 2, 2),

imshow(image),

title('Dilated Image');

image1 = true(n, m);

for r = 2:n-1

for c = 2:m-1

if all(all(D\_image(r-1:r+1, c-1:c+1) & se))

image1(r, c) = true;

else

image1(r, c) = false;

end

end

end

subplot(2, 2, 3),

imshow(image1),

title('Erosion Binary Image');

**Output:-**

