1. Write a program for adding and subtraction of two images.

**clc;**

**clear;**

**close;**

**img1 = imread('cameraman.tif');**

**img2 = imread('coins.png');**

**img3 = imresize(img2,[256,256]);**

**add = img1+img3;**

**sub = img1-img3;**

**figure(1)**

**imshow(add)**

**title('Addition of two image')**

**figure(2)**

**imshow(sub)**

**title('Subtraction of two image')**

**2. Write a program to negative an image.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**neg = 255-img;**

**figure**

**subplot(211)**

**imshow(img);**

**title('Original image')**

**subplot(212)**

**imshow(neg)**

**title('Negetive image')**

**3. Write a program to convert color image to gray-level image.**

**clc;**

**clear;**

**close;**

**img = imread('onion.png');**

**R = img(:,:,1);**

**G = img(:,:,2);**

**B = img(:,:,3);**

**gray = R\*0.2989+G\*0.5870+B\*0.1140;**

**figure**

**subplot(211)**

**imshow(img)**

**title('Color image');**

**subplot(212)**

**imshow(gray)**

**title('Gray Formation of Color image');**

4. Write a program to convert an image into binary /Monochrome using proper threshold.

**clc;**

**clear;**

**close;**

**img = imread('onion.png');**

**%if tell Color image to binay converstion then**

**R = img(:,:,1);**

**G = img(:,:,2);**

**B = img(:,:,3);**

**gray = R\*0.2989+G\*0.5870+B\*0.1140;**

**Th = 128;**

**%if tell gray to digital then**

**%binary = img>Th;**

**binary = gray>Th;**

**figure**

**subplot(211)**

**imshow(img)**

**title('Gray image')**

**subplot(212)**

**imshow(binary)**

**title('Binary formation of Gray image');**

**5.Write a program for zooming and shrinking image.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**[width,height] = size(img);**

**factor = 3;**

**new\_width = ceil(width\*factor);**

**new\_height = ceil(height\*factor);**

**new\_img = uint8(zeros(new\_width,new\_height));**

**for i=1:new\_width**

**for j=1:new\_height**

**new\_img(i,j) = img(ceil(i\*width/new\_width),ceil(j\*height/new\_height));**

**end**

**end**

**factor1 = 0.5;**

**new\_width1 = ceil(width\*factor1);**

**new\_height1 = ceil(height\*factor1);**

**new\_img1 = uint8(zeros(new\_width1,new\_height1));**

**for i=1:new\_width1**

**for j=1:new\_height1**

**new\_img1(i,j) = img(ceil(i\*width/new\_width1),ceil(j\*height/new\_height1));**

**end**

**end**

**figure(1)**

**imshow(img)**

**title('Original image')**

**figure(2)**

**imshow(new\_img)**

**title('Zooming image')**

**figure(3)**

**imshow(img)**

**title('Original image')**

**figure(4)**

**imshow(new\_img1)**

**title('Shrinking image')**

**6.Write a program to show the histrogram of an image.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**[rows,cols] = size(img);**

**histogram = zeros(1,256);**

**for i=1:rows**

**for j=1:cols**

**gray\_val = img(i,j);**

**histogram(gray\_val+1) = histogram(gray\_val+1)+1;**

**end**

**end**

**figure**

**subplot(211)**

**imshow(img)**

**title('Original image')**

**subplot(212)**

**bar(histogram)**

**title('Histogram of image')**

**7.Write a program to show the effect of log transformation of an image.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**new\_img = im2double(img);**

**c = 1;**

**trans\_img = c \* log(1+new\_img);**

**figure**

**subplot(211)**

**imshow(img)**

**title('Original image')**

**subplot(212)**

**imshow(trans\_img)**

**title('Log Transformation of an image')**

**8.Write a program to show the effect of power-law(or Gamma correction) for image enhancement.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**new\_img = im2double(img);**

**c = 1;**

**gamma = 0.5;**

**trans\_img = c\*(new\_img.^gamma);**

**figure**

**subplot(211)**

**imshow(img)**

**title('Original Image')**

**subplot(212)**

**imshow(trans\_img)**

**title('Gamma Correction of an image')**

**9.Write a program to show the effect of histrogram equalization of an image.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**[rows,cols] = size(img);**

**histogram = zeros(1,256);**

**for i=1:rows**

**for j=1:cols**

**intensity = img(i,j);**

**histogram(intensity+1) = histogram(intensity+1)+1;**

**end**

**end**

**pdf = histogram/(rows\*cols);**

**cdf = zeros(1,256);**

**cdf(1) = pdf(1);**

**for k=2:256**

**cdf(k) = cdf(k-1)+pdf(k);**

**end**

**eq\_intensity = round(cdf\*255);**

**eq\_img = uint8(zeros(rows,cols));**

**for i=1:rows**

**for j=1:cols**

**eq\_img(i,j) = eq\_intensity(img(i,j)+1);**

**end**

**end**

**eq\_histogram = zeros(1,256);**

**for i=1:rows**

**for j=1:cols**

**intensity = eq\_img(i,j);**

**eq\_histogram(intensity+1) = eq\_histogram(intensity+1)+1;**

**end**

**end**

**figure**

**subplot(221)**

**imshow(img)**

**title('Original image')**

**subplot(222)**

**bar(histogram)**

**title('Histogram of an image')**

**subplot(223)**

**imshow(eq\_img)**

**title('Enhanced Image')**

**subplot(224)**

**bar(eq\_histogram)**

**title('Histogram of enhanced image')**

**10.Write a program for bit plane slicing of 8 bit image.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**I = double(img);**

**bit1 = I-floor(I/2)\*2;**

**bit2 = floor(I/2) - floor(I/4)\*2;**

**bit3 = floor(I/4) - floor(I/8)\*2;**

**bit4 = floor(I/8) - floor(I/16)\*2;**

**bit5 = floor(I/16) - floor(I/32)\*2;**

**bit6 = floor(I/32) - floor(I/64)\*2;**

**bit7 = floor(I/64) - floor(I/128)\*2;**

**bit8 = floor(I/128) - floor(I/256)\*2;**

**p1 = bit1\*2^0;**

**p2 = bit2\*2^1;**

**p3 = bit3\*2^2;**

**p4 = bit4\*2^3;**

**p5 = bit5\*2^4;**

**p6 = bit6\*2^5;**

**p7 = bit7\*2^6;**

**p8 = bit8\*2^7;**

**re\_img = bit0 + bit1 + bit2 + bit3 + bit4 + bit5 + bit6 + bit7;**

**re\_img = uint8(re\_img);**

**% Display the bit planes and reconstructed image**

**figure**

**sgtitle('Bit Plane Slicing','Color','red','FontSize',18)**

**subplot(421)**

**imshow(p0)**

**title('Bit plane 1')**

**subplot(422)**

**imshow(p1)**

**title('Bit plane 2')**

**subplot(423)**

**imshow(p2)**

**title('Bit plane 3')**

**subplot(424)**

**imshow(p3)**

**title('Bit plane 4')**

**subplot(425)**

**imshow(p4)**

**title('Bit plane 5')**

**subplot(426)**

**imshow(p5)**

**title('Bit plane 6')**

**subplot(427)**

**imshow(p6)**

**title('Bit plane 7')**

**subplot(428)**

**imshow(p7)**

**title('Bit plane 8')**

**figure(2)**

**subplot(121)**

**imshow(img)**

**title('Original image');**

**subplot(122)**

**imshow(re\_img)**

**title('Reconstructed image');**

**11.Write a program for smoothing of an image by mean filter.**

**clc;**

**clear;**

**close;**

**img = imread('cameraman.tif');**

**noise\_img = imnoise(img,'salt & pepper',0.01);**

**[rows,cols] = size(noise\_img);**

**new\_img = zeros(rows,cols);**

**for i=2:rows-1**

**for j=2:cols-1**

**intensity = [img(i-1,j-1),img(i-1,j),img(i-1,j+1),...**

**img(i,j-1),img(i,j),img(i,j+1),...**

**img(i+1,j-1),img(i+1,j),img(i+1,j+1)];**

**avg = sum(intensity)/9.0;**

**new\_img(i,j)=avg;**

**end**

**end**

**new\_img = uint8(new\_img);**

**figure(1)**

**imshow(img)**

**title('Original image')**

**figure(2)**

**imshow(noise\_img)**

**title('Noisy image')**

**figure(3)**

**imshow(new\_img)**

**title('Filtered image')**

**12.Write a program for smoothing of an image using min,max and median filter.**

**clc;**

**close;**

**clear;**

**img = imread('cameraman.tif');**

**noise\_img = imnoise(img,'salt & pepper',0.01);**

**[rows,cols] = size(noise\_img);**

**new\_img = zeros(rows,cols);**

**for x = 2 : rows-1**

**for y = 2 : cols-1**

**intensity = [img(x-1,y-1),img(x-1,y),img(x-1,y+1),...**

**img(x,y-1),img(x,y),img(x,y+1),...**

**img(x+1,y-1),img(x+1,y),img(x+1,y+1)];**

**n = length(intensity);**

**for i = 1:n-1**

**for j = 1 : n-i**

**if intensity(j)>intensity(j+1)**

**temp = intensity(j) ;**

**intensity(j) = intensity(j+1);**

**intensity(j+1)=temp;**

**end**

**end**

**end**

**median = intensity(5);**

**new\_img(x,y) = median;**

**%for min = intensity(1)**

**%for max = intesity(9);**

**end**

**end**

**new\_img = uint8(new\_img);**

**figure(1)**

**imshow(img)**

**title('Original image')**

**figure(2)**

**imshow(noise\_img)**

**title('Noisy image')**

**figure(3)**

**imshow(new\_img)**

**title('Filtered image')**

**13.Write a program to sharpen an image by a Laplacian operator.**

**clc;**

**clear;**

**close;**

**img = imread('moon.tif');**

**img = im2double(img);**

**laplacian\_filter = [0 -1 0; -1 4 -1; 0 -1 0];**

**[rows, cols] = size(img);**

**laplacian\_img = zeros(rows, cols);**

**sharpened\_img = zeros(rows, cols);**

**for i = 2:rows-1**

**for j = 2:cols-1**

**local\_neighborhood = double(img(i-1:i+1, j-1:j+1));**

**filtered\_pixel = sum(sum(local\_neighborhood .\* laplacian\_filter));**

**laplacian\_img(i, j) = filtered\_pixel;**

**sharpened\_img(i, j) = img(i, j) + laplacian\_img(i,j);**

**end**

**end**

**subplot(1, 3, 1);**

**imshow(img);**

**title('Original Image');**

**subplot(1, 3, 2);**

**imshow(laplacian\_img);**

**title('Laplacian Image');**

**subplot(1, 3, 3);**

**imshow(sharpened\_img);**

**title('Sharpened Image');**

**14.Write a program to sharpen an image by a sobel operator.**

**clc;**

**clear;**

**close;**

**img = imread('moon.tif');**

**img = im2double(img);**

**sobel\_x = [-1 0 1; -2 0 2; -1 0 1];**

**sobel\_y = [-1 -2 -1; 0 0 0; 1 2 1];**

**[rows, cols] = size(img);**

**sobel\_img = zeros(rows, cols);**

**sharpened\_img = zeros(rows, cols);**

**for i = 2:rows-1**

**for j = 2:cols-1**

**local\_neighborhood = img(i-1:i+1, j-1:j+1);**

**Gx = sum(sum(local\_neighborhood .\* sobel\_x));**

**Gy = sum(sum(local\_neighborhood .\* sobel\_y));**

**sobel\_img(i, j) = sqrt(Gx^2 + Gy^2);**

**sharpened\_img(i, j) = img(i, j) + sobel\_img(i, j);**

**end**

**end**

**subplot(1, 3, 1);**

**imshow(img);**

**title('Original Image');**

**subplot(1, 3, 2);**

**imshow(sobel\_img);**

**title('Sobel Filtered Image');**

**subplot(1, 3, 3);**

**imshow(sharpened\_img);**

**title('Sharpened Image');**

**15.Write a program to sharpen an image by a prewitt operator.**

**clc;**

**clear;**

**close;**

**img = imread('moon.tif');**

**img = im2double(img);**

**prewitt\_x = [-1 0 1; -1 0 1; -1 0 1];**

**prewitt\_y = [-1 -1 -1; 0 0 0; 1 1 1];**

**[rows, cols] = size(img);**

**prewitt\_img = zeros(rows, cols);**

**sharpened\_img = zeros(rows, cols);**

**for i = 2:rows-1**

**for j = 2:cols-1**

**local\_neighborhood = img(i-1:i+1, j-1:j+1);**

**Gx = sum(sum(local\_neighborhood .\* prewitt\_x));**

**Gy = sum(sum(local\_neighborhood .\* prewitt\_y));**

**prewitt\_img(i, j) = sqrt(Gx^2 + Gy^2);**

**sharpened\_img(i, j) = img(i, j) + prewitt\_img(i, j);**

**end**

**end**

**subplot(1, 3, 1);**

**imshow(img);**

**title('Original Image');**

**subplot(1, 3, 2);**

**imshow(prewitt\_img);**

**title('Prewitt Filtered Image');**

**subplot(1, 3, 3);**

**imshow(sharpened\_img);**

**title('Sharpened Image');**