**Image processing code:**

1. **To generate an image:**

clc; clear all; close all;

img = imread('cameraman.tif');

imshow(img);

* **Output:**

****

1. **To convert grey scale to black and white image:**

clc; clear all; close all;

img = imread('cameraman.tif');

% converting image to black and white scale

for i = 1:size(img, 1)

for j = 1:size(img, 2)

if img(i, j) >= 127

img(i, j) = 255;

else

img(i, j) = 0;

end

end

end

imshow(img);

* **Output:**

|  |  |
| --- | --- |
| **Original Image** | **Black and White Image** |
|  |  |

1. **To convert grey scale into image negative:**

% Problem Statement:

% To convert a grey scale image to digital negative

% Formula used:

% S = (L - 1) - r

% Here, S = pixel values of output image

% L = maximum intensity level (=2^n)

% r = pixel values of input image

% Variables Used:

% o\_img = input image

% i\_img = output image

% l = max intensity

% n = max power of 2

clc; clear all; close all;

% pixel matrix of original image

o\_img = imread('cameraman.tif');

% finding value of L:

max\_val = double(max(o\_img(:))); % max intensity

n=0;

while true

if max\_val <= 2^n

l=2^n;

break;

end

n=n+1;

end

r\_img = (l - 1) - o\_img;

imshow(r\_img)

* **Output:**

|  |  |
| --- | --- |
| **Original Image** | **Digital Negative Image** |
|  |  |

1. **To convert grey scale into threshold image:**

% Problem Statement:

% To convert a grey scale image to threshold

% Formula used:

% S = (L - 1) if r >= given pixel value

% = 0 if r < given pixel value

% Here, S = pixel values of output image

% r = pixel values of input image

% Variables Used:

% o\_img = input image

% i\_img = output image

% l = max intensity

clc; clear all; close all;

% pixel matrix of original image

o\_img = imread('cameraman.tif');

% pixel matrix of resulting image

r\_img = zeros(size(o\_img));

% finding value of L:

l = double(max(o\_img(:)));

% assuming r = 128

for i = 1:size(o\_img, 1)

for j = 1:size(o\_img, 2)

if o\_img(i, j) >= 128

r\_img(i, j) = l - 1;

else

r\_img(i, j) = 0;

end

end

end

imshow(r\_img)

* **Output:**

|  |  |
| --- | --- |
| **Original Image** | **Threshold Image** |
|  |  |

1. **To convert grey scale into logarithmic transform:**

% Problem Statement:

% To convert a grey scale image to logarithmic transform

% Formula used:

% S = clog(1+r)

% Here, S = pixel values of output image

% c = constant term

% r = pixel values of input image

% Variables Used:

% o\_img = original image

% r\_img = resulting image

% l = max intensity

% n = max power of 2

clc; clear all; close all;

% pixel matrix of original image

o\_img = imread('cameraman.tif');

% finding value of L:

max\_val = double(max(o\_img(:))); % max intensity

n=0;

while true

if max\_val <= 2^n

l=2^n;

break;

end

n=n+1;

end

% taking c = 1 in given formula

o\_img = double(o\_img);

[M, N] = size(o\_img);

r\_img = zeros(M, N);

% Apply log transformation (step 1: log(1 + r))

for i = 1:M

for j = 1:N

r\_img(i,j) = log10(1 + o\_img(i,j));

end

end

% final image

c = l/(log10(1+l));

r\_img = round(c \* r\_img);

imshow(uint8(r\_img)) % uint8 - unsigned 8-bit integers

* **Output:**

|  |  |
| --- | --- |
| **Original Image** | **Logarithmic Transform Image** |
|  |  |

1. **To convert grey scale into power transform:**

% Problem Statement:

% To convert a grey scale image to power transform

% Formula used:

% S = cr^Y

% Here, S = pixel values of output image

% c = constant term

% r = pixel values of input image

% Y(gamma) = constant term = 0.4

% Variables Used:

% o\_img = original image

% r\_img = resulting image

clc; clear all; close all;

% pixel matrix of original image

o\_img = imread('cameraman.tif');

c = 1;

Y = 0.4;

o\_img = im2double(o\_img);

% Apply log transformation (step 1: log(1 + r))

r\_img = c \* (o\_img .^ Y);

r\_img = uint8(255 \* r\_img);

imshow(r\_img);

* **Output:**

|  |  |
| --- | --- |
| **Original Image** | **Power Transform Image** |
|  |  |