FUNDAMENTALS OF IMAGE PROCESSING

(2181102)

Laboratory Manual

(Semester – VIIIth- EC)

Department of Electronics and Communication Engineering



Vishwakarma Government Engineering College, Chandkheda

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of Experiment** | **Sign** |
| **1** | **Introduction to Image Processing Toolbox** |  |
| **2** | **Digital Image Fundamentals** |  |
| **3** | **Image Enhancement In The Spatial Domain** |  |
| **4** | **Filtering In The Spatial Domain** |  |
| **5** | **Image Enhancement In The Frequency Domain** |  |
| **6** | **Image Restoration** |  |
| **7** | **Colour Image Processing** |  |
| **8** | **Image Compression** |  |
| **9** | **Morphological Image Processing** |  |
| **10** | **Image Segmentation** |  |

**PRACTICAL NO: 1**

**Introduction to Image Processing Toolbox**

**Introduction**

Image Processing Toolbox™ provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development. You can perform image segmentation, image enhancement, noise reduction, geometric transformations, and image registration using deep learning and traditional image processing techniques. The toolbox supports processing of 2D, 3D, and arbitrarily large images.

Image Processing Toolbox apps let you automate common image processing workflows. You can interactively segment image data, compare image registration techniques, and batch-process large datasets. Visualization functions and apps let you explore images, 3D volumes, and videos; adjust contrast; create histograms; and manipulate regions of interest (ROIs).

You can accelerate your algorithms by running them on multicore processors and GPUs. Many toolbox functions support C/C++ code generation for desktop prototyping and embedded vision system deployment.

**Key Features**

* Image analysis, including segmentation, morphology, statistics, and measurement
* Apps for image region analysis, image batch processing, and image registration
* 3D image processing workflows, including visualization and segmentation
* Image enhancement, filtering, geometric transformations, and deblurring algorithms
* Intensity-based and non-rigid image registration methods
* Support for CUDA enabled NVIDIA GPUs (with Parallel Computing Toolbox™)
* C-code generation support for desktop prototyping and embedded vision system deployment.

**PRACTICAL NO 2.**

**DIGITAL IMAGE FUNDAMENTALS**

**Aim:** Write MATLAB program to

* Read and display an image
* Write the image
* Display information of image
* Display an image with and without a border
* Flip the image
* Resizing an image to 1.25 times the original image
* Rotating an image 35 degrees counterclockwise.

**Code:**

%=========================Practical 2===========================

%% -------------------------------------------------------------

% 1.Read and display an image

% --------------------------------------------------------------

I = imread('images/cameraman.tif');

figure;

imshow(I);

title('Read and display an image');

%% -------------------------------------------------------------

% 2.Write the image

% --------------------------------------------------------------

imwrite(I,'images/cameraman.png');

%% -------------------------------------------------------------

% 3.Display information of image

%---------------------------------------------------------------

imageinfo('images/cameraman.tif');

%% -------------------------------------------------------------

% 4.Display an image with and without a border

% --------------------------------------------------------------

figure;

imshow(I,'Border','loose');

title('With Border');

figure;

imshow(I,'Border','tight');

%% -------------------------------------------------------------

% 5.Flip the image

% --------------------------------------------------------------

flip\_I = I(end:-1:1, 1:1:end);

figure;

imshow(flip\_I);

title('Flip the image');

%% -------------------------------------------------------------

% 6.Resizing an image to 1.25 times the original image

% --------------------------------------------------------------

resize\_I = imresize(I,1.25);

figure;

subplot(1,2,1)

imshow(I);

title('Original')

subplot(1,2,2)

imshow(resize\_I);

title('1.25 Resized')

%% -------------------------------------------------------------

% 7.Rotating an image 35 degrees counterclockwise

% --------------------------------------------------------------

rotate\_I = imrotate(I,35);

figure;

subplot(1,2,1)

imshow(I);

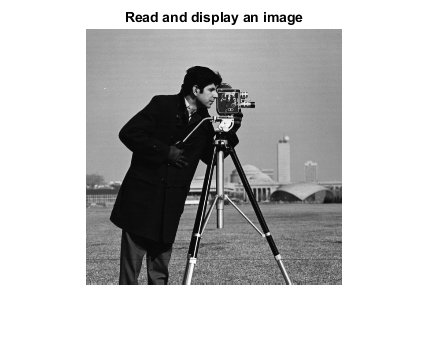
title('Original')

subplot(1,2,2)

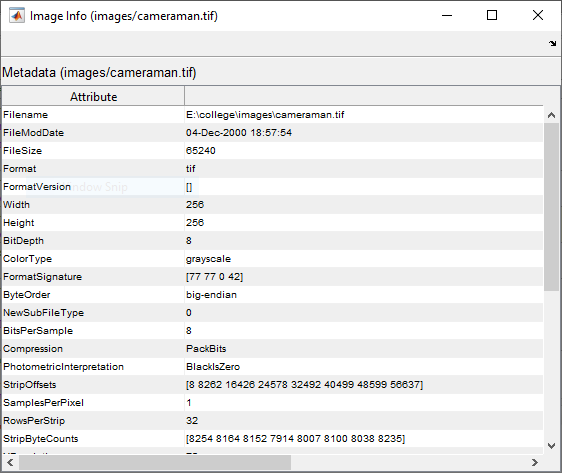
imshow(rotate\_I);

title('35 ccw rotated')

**Output:**

****

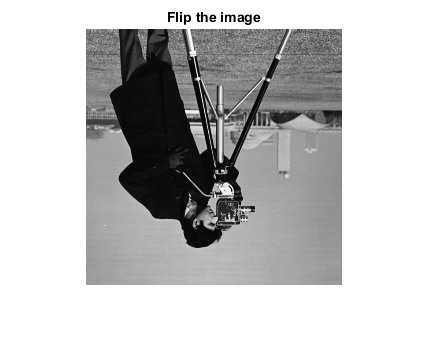
**Fig 2.1** Display an Image

****

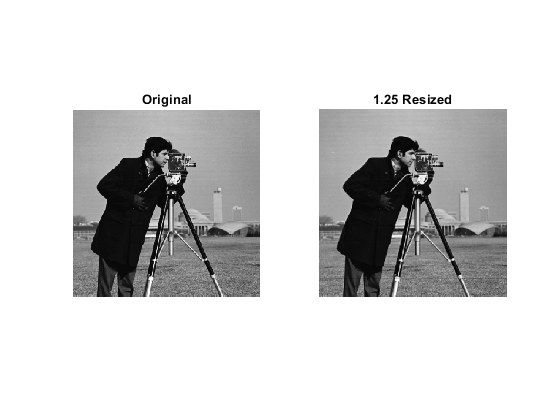
**Fig 2.2** Display Image Information

|  |  |
| --- | --- |
| **Fig 2.3** | **Fig 2.4** |

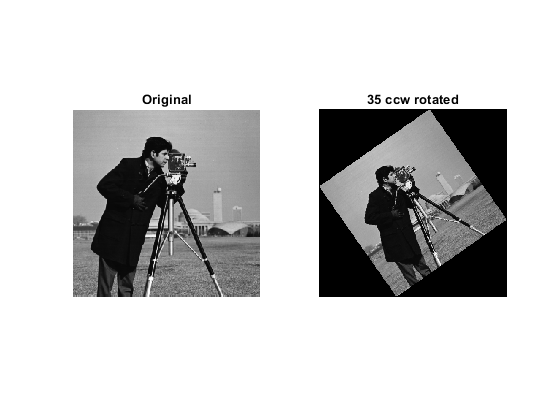
Display image with and without border

****

**Fig 2.5** Flip the image

****

**Fig 2.6** Resizing image to 1.25 times original size

****

**Fig 2.7** Rotate image 35 degrees in counterclockwise direction

**PRACTICAL NO 3.**

**IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN**

**Aim:** Write matlab program to

* produce image negative from an image
* to increase the contrast in a low contrast grayscale image and display the histogram of the output image. Compare the histogram before and after intensity adjustments
* display images before and after gamma correction. Compare the histogram before and after gamma correction
* performHistogram Equilization

**Code:**

%=============================Practical 3=================================

%% -----------------------------------------------------------------------

% 1.Produce image negative from an image

% ------------------------------------------------------------------------

I = imread('images/cameraman.tif');

I\_inv = 255 - I;

figure;

subplot(1,2,1);

imshow(I);

title('Original');

subplot(1,2,2);

imshow(I\_inv);

title('Inverted');

%% -----------------------------------------------------------------------

% 2.To increase the contrast in a low contrast grayscale image and display

% the histogram of the output image. Compare the histogram before and

% after intensity adjustments.

% ------------------------------------------------------------------------

I = imread('images/pout.tif');

contrast\_I = imadjust(I, [0.3 0.7], []);

figure;

subplot(2,2,1);

imshow(I);

title('Original');

subplot(2,2,2);

imhist(I);

title('Original Histogram');

subplot(2,2,3);

imshow(contrast\_I);

title('Contrast Enhanced');

subplot(2,2,4);

imhist(contrast\_I);

title('Contrast Histogram');

%% -----------------------------------------------------------------------

% 3.Display images before and after gamma correction. Compare the

% histogram before and after gamma correction.

% ------------------------------------------------------------------------

I=imread('eight.tif');

gamma\_I=imadjust(I,[],[],0.3);

figure;

subplot(2,2,1);

imshow(I);

title('ORIGINAL');

subplot(2,2,3);

imshow(gamma\_I);

title('GAMMA=0.3');

subplot(2,2,2);

imhist(I);

title('Original Histogram');

subplot(2,2,4);

imhist(gamma\_I);

title('GAMMA=0.3 Histogram');

%% -----------------------------------------------------------------------

% 4.Perform Histogram Equilization

% ------------------------------------------------------------------------

I = imread('images/pout.tif');

histeq\_I=histeq(I);

figure;

subplot(2,2,1);

imshow(I);

title('ORIGINAL');

subplot(2,2,2);

imhist(I);

title('HISTOGRAM ORIGINAL');

subplot(2,2,3);

imshow(histeq\_I);

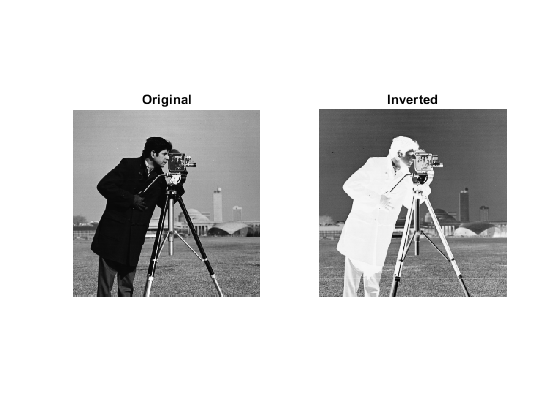
title('HISTEQ IMAGE');

subplot(2,2,4);

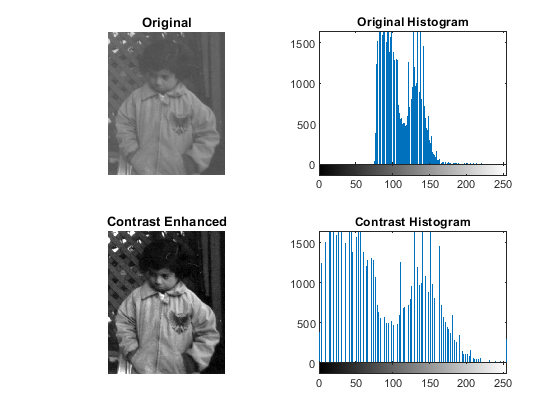
imhist(histeq\_I);

title('HISTOGRAM EQUALIZED');

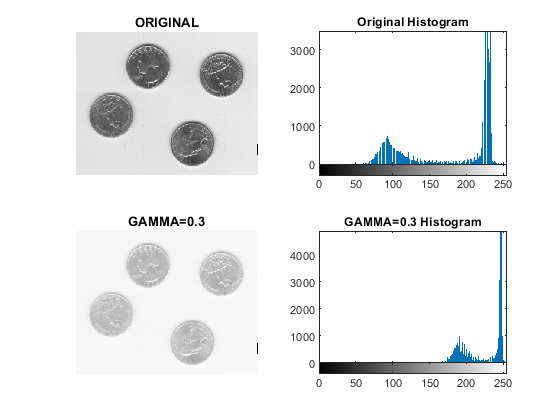
**Output:**



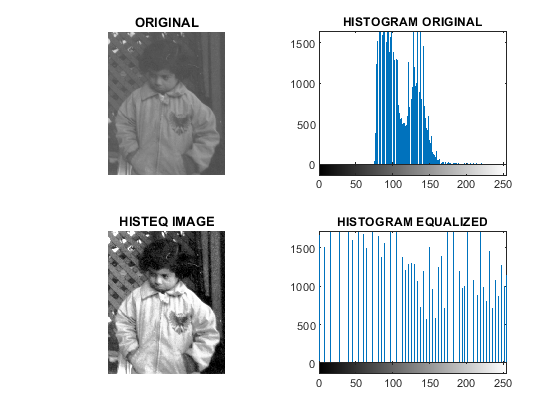
**Fig 3.1** Produce Image Negative



**Fig 3.2** Increase contrast in greyscale image and compare Histogram



**Fig 3.3** Display and compare Image before and after gamma correction.



**Fig 3.4** Perform Histogram Equalization

**PRACTICAL NO 4.**

**FILTERING IN THE SPATIAL DOMAIN**

**Aim:** Write a matlab program to

* illustrate the effects of smoothing with averaging filter masks of sizes 8,16,64,128
* filter the image with median filter and averaging filter when the image is corrupted by salt and pepper noise.

**Code:**

**Output:**

**PRACTICAL NO 5.**

**IMAGE ENHANCEMENT IN THE FREQUENCY DOMAIN**

**Aim:** Write a matlab program to perform Low Pass Filtering and High Pass Filtering with cut of frequencies set at radius of 5,15,30,80 for the following filters. Plot input image, the frequency spectrum of input image & filter and output image.

* Ideal low pass filter
* Gaussian low pass filter
* Butterworth low pass filter

**Code:**

**Output:**

**PRACTICAL NO 6.**

**IMAGE RESTORATION**

**Aim:** Write a matlab program to

* demonstrate the impact of following noises on an image and plot the corresponding histograms.
* Create a simulated blur in the image and deblur the image with a Wiener Filter
* Create a simulated blur in the image, add Gaussian noise to it and deblur the image with a Regularized Filter.

**Code:**

**Output:**

**PRACTICAL NO 7.**

**COLOUR IMAGE PROCESSING**

**AIM:** Write a MATLAB program to

A. Illustrate conversion between the following image types:

1. RGB image or colormap to Grayscale
2. Indexed image to RGB image and viceversa
3. Indexed image to grayscale image and viceversa
4. Grayscale or Binary image to indexed image
5. Grayscale image/Indexed image/Truecolor image to binary image

B. Read an RGB image and separate the R,G and B components from it. Convert the RGB image to HSV image and separate out the H,S and V components from it.

**Code:**

**Output:**

**PRACTICAL NO 8.**

**IMAGE COMPRESSION**

**AIM:** Write a MATLAB program to compress an image and display the corresponding compression ratio.

**Code:**

**Output:**

**PRACTICAL NO 9.**

**MORPHOLOGICAL IMAGE PROCESSING**

**AIM:** Write a MATLAB program to

 Creating a structuring element

a) Flat Diamond shaped structuring element

b) 3-by-3 square structuring element

c) Disk shaped structuring element with radius 10

d) Rolling ball structuring element

e) Vertical line structuring element

 Dilate a binary image using the above structuring elements

 Erode a binary image using the above structuring elements

 Illustrating Morphological open by erosion followed by dilation

 Illustrating Morphological open by using the function bwmorph

 Illustrating Morphological close by dilation followed by erosion

 Illustrating Morphological close by using the function bwmorph

**Code:**

**Output:**

**PRACTICAL NO 10.**

**IMAGE SEGMENTATION**

**AIM:** Write a MATLAB program to

 Detection of line in an image

a) -45 degree line

b) Horizontal line

c) Vertical line

d) 45 degree line

 Edge detection of an image

a) Edge detection using prewitt operator

b) Edge detection using Sobel operator

**Code:**

**Output:**