

**Implementation of Digital Image Basic and Editing functions using MATLAB**Lalit Kumar Saini<sup>1</sup>, Deepak Verma<sup>2</sup><sup>1</sup>Department of CSE, JECRC UDML College of Engineering, Jaipur, Rajasthan, India<sup>2</sup>Department of CSE, JECRC UDML College of Engineering, Jaipur, Rajasthan, India

**Abstract** — Digital image is the big multimedia medium of Information gathering and save data. Videos are also the combination of digital images put into the frame. As the technology grows developers needs to develop techniques that can help photographers and Scientist to process digital images easily. In this paper the implementation of different-2 image processing and editing are shown. MATLAB is the scientific tool for engineering applications. There are so many functions related to digital images present in MATLAB[1].

**Keywords-** Digital Image, MATLAB, cropping, Histogram.

**I. INTRODUCTION**

Eyes is one of the powerful senses of human being for gathering information. Large parts of the activity of human brain involved in processing image from eyes. Some of images are of low contrast than human cannot read it properly such as satellite images and medical X-ray images. In a single digital image one can hide so much of information. So processors and developers needs to develop some algorithms and predefined functions that can help in enhancement of images. Image enhancement is used for image processing and video processing applications[2].

In this paper we discuss about the basic image processing functions of MATLAB. A digital image is the matrix of pixel values and we know that MATLAB processing basic platform is matrix also. Due to this MATLAB is the best tool for image processing[3]. User can easily read every pixel of image and if they want to edit it, they will do it easily.

**1.1 Brief History of Digital Image**

Digital imaging starts with the invention of CCD (charge-coupled device). Boyle & Smith of Bell Labs used photons of light to create a Digital image using charged-coupled device (CCD) in 1969. They both were the grandfathers of the digital imaging revolution. Their invention was first implemented commercially in television cameras in 1975. Steve Sasson created a prototype of digital camera in 1975 at Kodak. The first digital camera was Sony's Mavica B&W camera (1981) commercially. Digital imaging matured in 1994 with the development of the scanback by Mike Collette using a Kodak tri-liner CCD array.

**1.2 Types of Digital Images :**

- ❖ **Binary:** In binary image each pixel has only two values just black or white. If considered the values, it have only two values for each pixels(0,1).
- ❖ **Grayscale:** Each pixel has the value from 0 (black) to 255 (white). So that every pixel represented by 8 bit (exactly 1 byte) pictures is shaded like grey in that image type.
- ❖ **True Color, or RGB:** In RGB image there are 3 colors present in each pixel, red, green, blue. The values for all color is vary from 0-255. So in that type of image 3 matrices of each color values for each pixel is stacked[4].

**1.3 Basic Image functions :****1.2.1 Processing Functions:**

- ❖ Browse image
- ❖ Image addition
- ❖ Gray conversion
- ❖ Noise addition
- ❖ Histogram
- ❖ B/W conversion

### 1.2.2 Editing functions

- ❖ Edge detection
- ❖ Image rotate
- ❖ Image Resize
- ❖ Crop

## II. IMPLEMENTATION OF THE DIGITAL IMAGE PROCESSING FUNCTIONS IN MATLAB.

Here we present the process of implement the imaging functions in MATLAB 2014b. So at first we have shown the some basics of MATLAB.

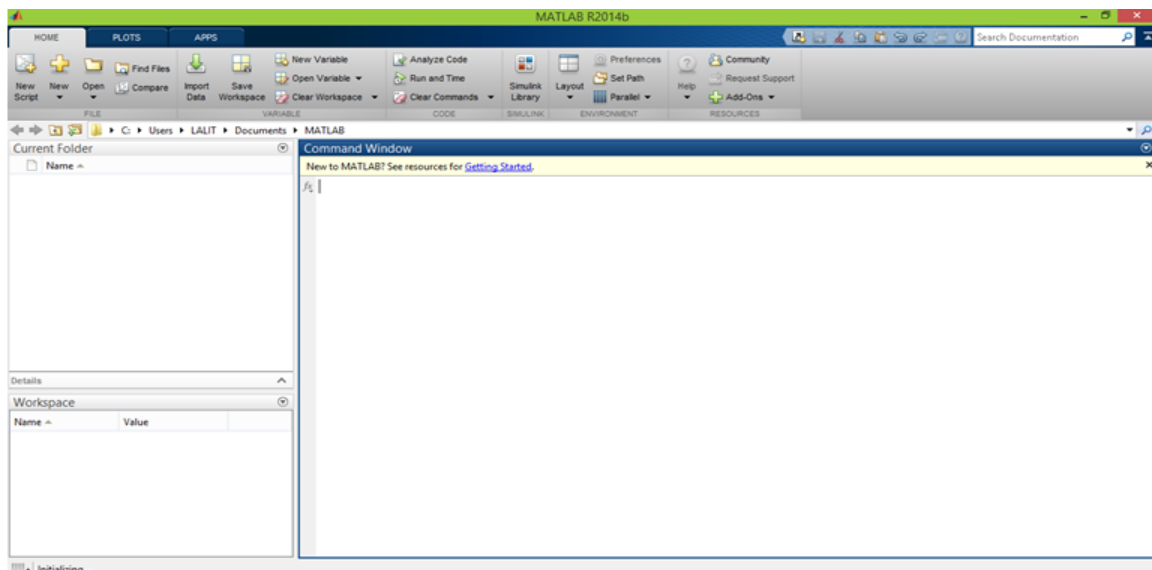
### 2.1. Opening MATLAB

- a. Access the Start Menu, Proceed to Programs, Select MATLAB 2014b
- OR--
- b. Open through C:\Program Files\MATLAB\R2014b\bin\matlab.exe

### 2.2. MATLAB

When MATLAB opens, the screen should look something like what is pictured in figure 1. below.

**Figure 1. MATLAB 2014b window**



All the commands are written in the command window at the right side of the Figure 1.

### 2.3. Implementing Basic image functions

#### 2.3.1 Open Image and Browse Image

- i. Browse an image:  

```
[filename1, pathname]=uigetfile ('*.*','select the image');  
image1=imread (num2str (filename1));  
imshow(image1);  
title ('scene.jpg');
```
- ii. Open an Image  

```
I= imread('scene.jpg');  
figure (1);  
imshow(I);
```

### 2.3.2 Image addition

Image addition will superimpose or overlay an image against another image or control the brightness of an image.

#### i. Superimpose Two Images

```
I1=imread('scene.jpg');  
I2=imread('FIC.jpg');  
I3=imadd(i1,i2);  
imshow(i3);
```

#### ii. Control Brightness of Image

```
i1=imread('scene.jpg');  
i2=imadd(i1,50);  
subplot(2,1,1),imshow(i1);  
subplot(2,1,2),imshow(i2);
```

### 2.3.3 Gray Conversion

This conversion converts a RGB image to gray scale, pixel ranging from 0 to 255.

```
i1=imread('scene.jpg');  
i2=rgb2gray(i1);  
subplot(2,1,1),imshow(i1);  
subplot(2,1,2),imshow(i2);
```

### 2.3.4 Noise Addition:

When we take pictures through any medium different-2 types of noise added to an image.

Noises can be of various types such as Poisson, Salt and pepper, Gaussian and Speckle.

```
i1=imread('scene.jpg');  
i2=imnoise(i1,'salt & pepper',0.04)  
subplot(2,1,1),imshow(i1);  
subplot(2,1,2),imshow(i2);
```

### 2.3.5 Histogram:

Histograms are a method to show the intensities of an image. As a definition, image histograms are the graph plot of the image where the x axis shows the intensity value and the y axis shows the number of pixels with that intensity value.

```
i1=imread('scene.jpg');  
i2=rgb2gray(i1);  
Figure, imhist(i2)
```

### 2.3.6 Convert to Binary Image:

Binary image had only two possibilities of pixel values either (0) or (1).

```
i1=imread('scene.jpg');  
i2=im2bw(i1);  
imshow(i2);
```

## 2.4 Implementing Image editing functions:

### 2.4.1 Edge Detection:

Edge detection technique is applicable only to binary images, so in case of an RGB or gray image it has to be first converted to a binary image and then edge detection technique has to be applied. RGB image is not directly converted to B/W image.

```
i1=imread('scene.jpg');  
i2=rgb2gray(i1);  
ED1 = edge(i2,'sobel');  
ED2 = edge(i2,'canny');  
figure,imshow(ED1);  
figure,imshow(ED2);
```

#### **2.4.2 Image rotate**

Image rotate is used to rotate the image to a specified degree.

```
i1=imread('scene.jpg');  
i2=imrotate(i1,90);  
figure,imshow(i2);
```

#### **2.4.3 Image Crop**

Image cropping is used to select any particular portion of the whole image.

```
I1 = imread('scene.jpg');  
I2 = imcrop(I1,[0 0 530 512]);  
imshow(I1), figure, imshow(I2)
```



#### **2.4.4 Image Resize**

Image resize is being used to resize the actual image to certain multiples. The syntax is 'imresize'.

```
I1 = imread('scene.jpg');  
I2 = imresize(i1,.5)  
imshow(I2)
```

### **III. INPUT /OUTPUT AND RESULTS.**

#### **3.1 Input images:**

S.no	Image Name	Image
1.	Scene.jpg	
2.	FIC.jpg	

*Table 1. Input Images*

#### **3.2 Outputs Images:**

S.no	Function Name	Effect of function on image
1.	Imadd():superimpose	
2.	Imadd:brightness	
3.	rgb2gray	
4.	Imnoise:salt&pepper	
5.	Imhist: histogram	
6.	im2bw()	
7.	Edge:sobel algorithm	
8.	Edge:canny algorithm	
9.	Imrotate: 90 degree	
10.	imcrop	

**Table 2. Output Images with their respective functions**

#### **IV. CONCLUSION AND FUTURE WORK**

Image processing and editing functions available in MATLAB are under one common platform. So according to the above article everyone easily understands that the implementation of image functions in MATLAB is very easy. As the need grows many of the new functions and tools are added to MATLAB library for helping of developers. As a future of above work is implement that functions in MATLAB with GUI.

## **V. REFERENCES**

- [1] Rafael C. Gonzalez (University of Tennessee), Richard E. Woods (MedData Interactive) and Steven L. Eddins (The MathWorks, Inc.), in 'Digital Image Processing Using MATLAB' Second Edition, 2009 by Gatesmark , LLC.
- [2] lukas krasula, Milos Klima, Eric Rogard, Edouard Jeanblanc, "Matlab-based applicationsfor image processing and image quality assessment Part ii: experimental results", RadioEngineering, vol. 21, no. 1, april 2012
- [3] Hudian Cao, Yuming Shen, " Application of MATLAB Image Processing Technology in Sewage Monitoring System" Electronic Measurement & Instrumentations, 2009.ICEMI' 09'th International Conference , Aug ,2009
- [4] Nandgaonkar, S.; Jagtap, R.; Anarase, P., " Image mining of textual images using low-level image features" , Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference, 2010
- [5] Guerrero j., "Tutorial III: Image Processing and analysis with MATLAB", Electrical, communication and computers, 2009. CONIELECOMP, Feb, 2009.