

Design of Image Editor & Edge Detector in Matlab GUI Environment

^[1] A.Sanyasi Rao, ^[2] K.Radha Krishna

Dept. of Electronics & Communication Engg, Balaji Institute of Technology & Science
Dept. of Electronics & Communication Engg, Balaji Institute of Technology & Science

Abstract: - Digital Image Processing techniques can enhance or distort an image, highlight certain features of an image create a new image from portions of other images, restore an image that has been degraded during or after the image acquisition. This paper envisages the implementation of basic features of Image Processing like viewing the red, green and blue components of a colour image separately, converting image to gray, black & white, negative; image addition & joining, comparison, crop, resize, increase/decrease brightness, rotate and edge detection using different algorithms etc. Also, it deals with accessing Webcam and getting a snapshot of it to process.

Keywords: - Image Processing, MATLAB, GUI, Edge Detection.

I. INTRODUCTION

Digital Image Processing is a subset of the electronic domain wherein the image is converted to an array of small integers, called pixels, representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer. MATLAB based image processing is a very suitable platform and very simple to build an algorithm. An image is a matrix of pixel values. MATLAB considers every input as a matrix. For this reason MATLAB provides an easy tool for image Processing as a user can easily access each and every pixel value from the image matrices and edit it. Generally users deal with three types of images, hence three different matrices. Black and white or binary image matrix consists of only zero and one, one being the brighter portion and zero being the darker portion. Images are of 8bit and corresponding image matrix is 256 x 256. Gray scale image is also a 2 dimensional matrix with each element value varying from 0 to 256. Similar to gray scale image RGB image can also be denoted by matrix with each pixel values varying from 0 to 256. In case of RGB image, three separate matrices for each red, green and blue components overlap to form a RGB image of 256x256x3 dimension.

II IMAGE EDITOR

The image editor is prepared by the Graphical User Interface (GUI) option existing in MATLAB . The editor consists of three axis and the following options:

- Image input using web camera

- Browse image
- Image addition
- Image joining
- Image comparison
- Black & White Conversion
- Gray Conversion
- Image Negative
- Brightness adjustment
- Image Rotate with user input
- Image Cropping with user input
- Slider to adjust Image Brightness
- Image Resize with user input
- Image Reset to Original
- Panel to view RGB format and each component separately
- Clearing all axes
- Edge detection

While creating the GUI, push button, slider, radio button, pop-up menu, panel and button group options available in MATLAB are used, (Figure. 2) Now each component is discussed separately.

Panels - In this 5 panels are created. First panel is to load the image from either browser or from webcam. Second panel is to take second image from browser or webcam. In this panel image addition, joining and comparison options are included. Third panel has editing options like gray image, black & white, image negative, brightness control using slider, resize, crop, rotate, clear, reset to original etc. Fourth panel is to have Original RGB format and individual primary colours separately. Fifth panel is to give different edge detection methods.

Axes - Three axes are used in the GUI. The first axis is mainly for a first input image. The second axis is mainly for second input image and the third one is to display the effect. (fig. 2)

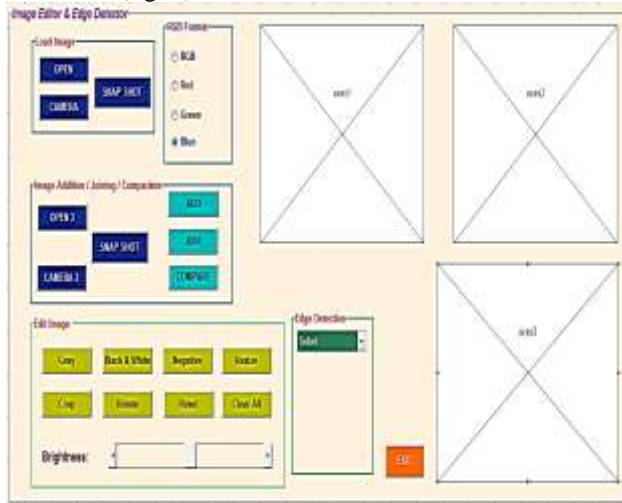


Fig.1:MATLAB GUI Layout with Axes



Fig. 2: Axes

Browse Image - The Load image panel helps in selecting an image from hard disk or an image taken directly from web camera for further processing.

Image Addition - An inbuilt function 'imadd' available in MATLAB is used to perform addition of two images in which pixel values of first image are added to the second image.

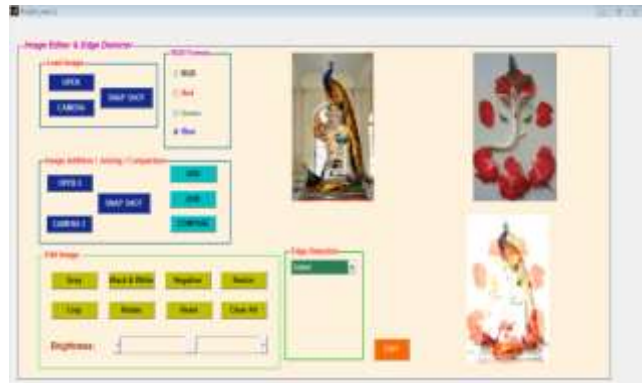


Fig. 3: Image Addition

Image Joining

Two images can be joined after converting both the image sizes into same size.

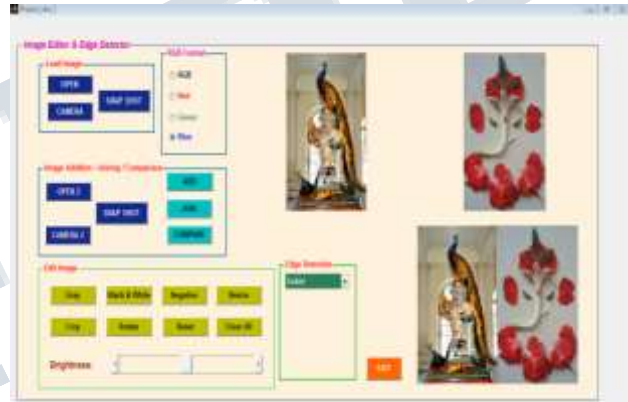


Fig. 4 Image Joining

Image Comparison - Two images are compared and tell whether both the images are perfectly same or almost same or not same. To do this histogram difference between two images will be calculated. If both images are same then the histogram difference is zero.



Fig. 5: Image Comparison

Gray Conversion - Gray conversion is done mainly to convert an RGB image (three dimensional matrix) to gray

scale (two dimensional matrix) having pixel values ranging from 0 to 255.

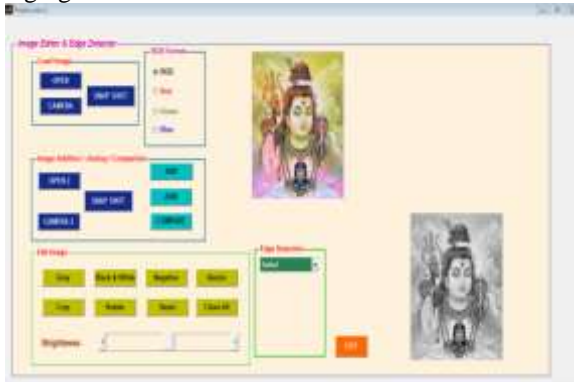


Fig. 6: RGB to Gray Conversion

Black and White Image - Binary image (black and white) image is a two dimensional image with pixel values either 0(black) or 1(white).

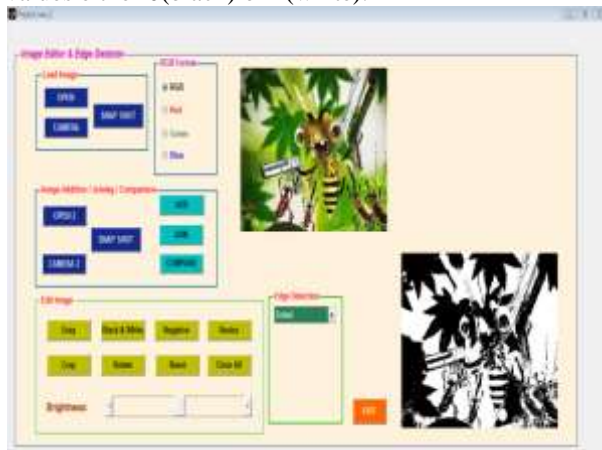
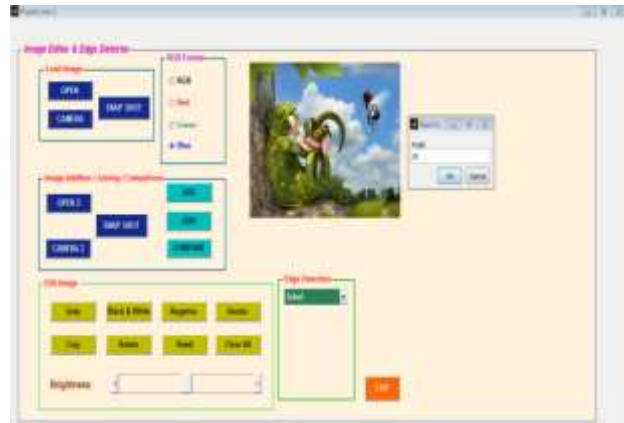


Fig. 7: RGB to B & W Conversion

Image rotate - Image rotate is used to rotate the image to a specified degree. 'imrotate' command is used to perform this operation. Axes 1 is the actual image and axes 3 is the rotated image. Here user can give the angle to which the image has to be rotated.



g.8(a): Image Rotate with user input

Fi

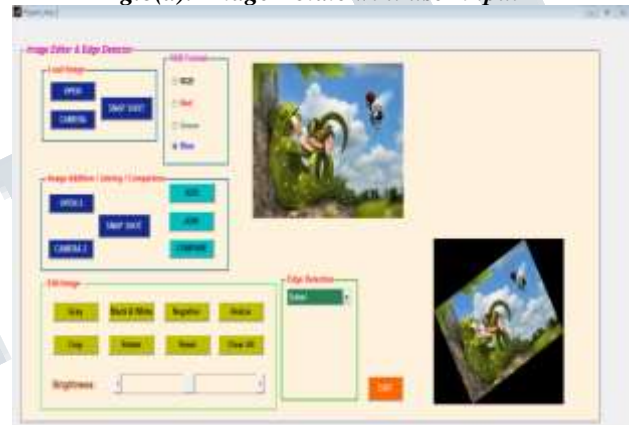


Fig.8(b): Image Rotate with user input

Image Crop - Image cropping is used to select any particular portion of the whole image. The syntax is 'imcrop'. Axes 1 shows the actual image and axis 3 shows the crop portion. This is done by taking input from the user while simulation is going on.



Fig.9(a): Original Image size



Fig.9(b): Setting dimensions for Image cropping



Fig.10(b): Image Resize with user input

Brightness - Here one slider is used for brightness adjustment. The axis 1 is the original image and axis 3 shows a change in brightness.



Fig.9: Image cropping with user input

Image Resize -Image resize is being used to resize the actual image to certain multiples. The syntax is 'imresize'. This is done by taking input from the user while simulation is going on.



Fig. 11: Brightness adjustment

RGB Format - RGB image is overlap of three two dimensional matrix. This RGB panel is used to view the red, green and blue components of the image separately. The RGB color model is an preservative color model in which red, green and blue light are added together in different ways to replicate a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.



Fig.10(a): Image Resize with user input

**International Journal of Engineering Research in Electronics and Communication
 Engineering (IJERECE)
 Vol 5, Issue 2, February 2018**

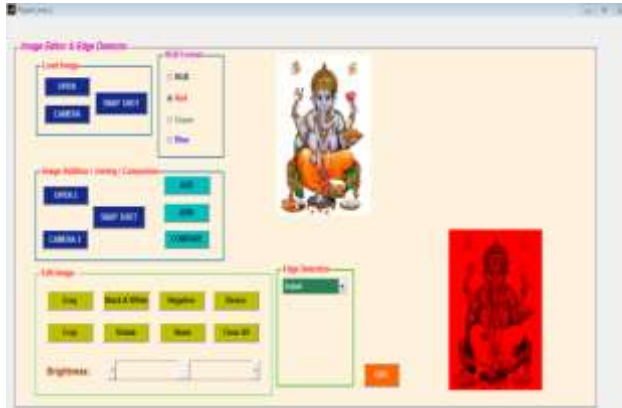


Fig.12(a): Red Component

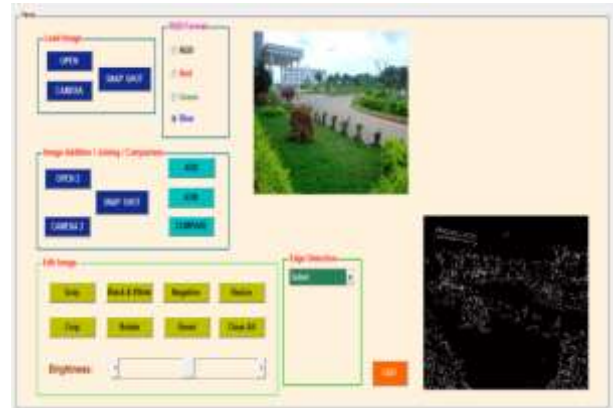


Fig.13(a): Sobel Edge detection technique



Fig.12(b): Green Component

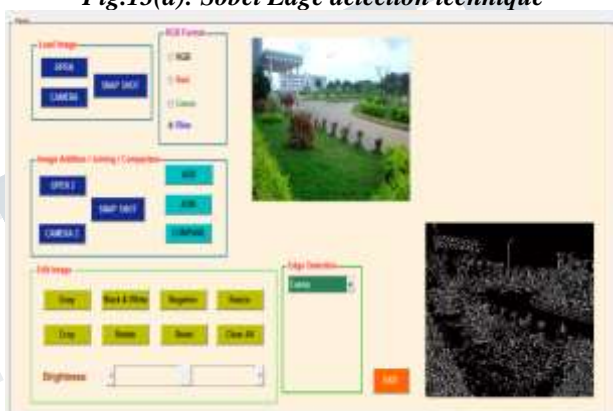


Fig.13(b): Canny Edge detection technique



Fig.12(c): Blue Component

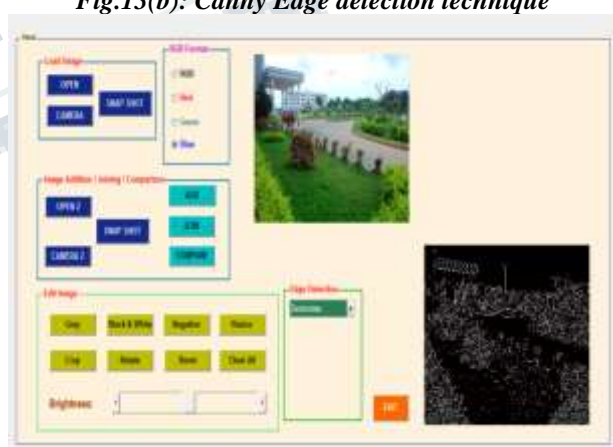


Fig.16: Zerocross Edge detection technique

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. The various types of edge detection techniques are: Sobel, Prewitt, Roberts, LoG, Zerocross and Canny.

Image Loading using Webcam - Webcam is accessed to load the image and this image is further processed.

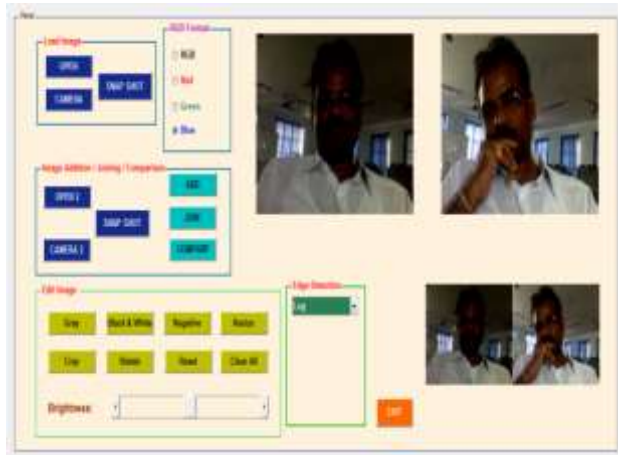


Fig. 17: Images from webcam

III CONCLUSION

In this paper a variety of components of this MATLAB based image editor have been discussed. Tried to make the image editor as user friendly as possible. The purpose of the image editor is to get the various image editing functions existing in MATLAB under one common platform and to make it easier for the understanding of any user.

REFERENCES

1. Digital Image Processing - Ikvinderpal Singh (1st Edition – 2015).
2. Digital Image Processing using MATLAB - Rafael C.Gonzalez, Richard E.Woods& Steven L.Eddins.
3. Digital Image Processing And Analysis – B.Chandra & D.Dutta Majumder.
4. Digital Image Processing (Implementation using MATLAB) – Ashish Jain
5. Alasdair McAndrew, in ‘An Introduction to Digital Image Processing with Matlab, Notes for SCM2511 Image Processing 1’, School of Computer Science and Mathematics ,Victoria University of Technology
6. Justyna Ingot, ‘Advanced Image Processing with Matlab’, in Bachelor’s Thesis Information Technology, May 2012, Date of the bachelor's thesis 07.05.2010 ,Mikkeli University of Applied Sciences.