**INTEGRATED IMAGE PROCESSING FUNCTIONS USING MATLAB GUI**

T. Karan#1, S. Pavan#2, V.G Adithya#3

==

***Abstract*—** **Graphic User Interface (GUI) was designed to integrate many functions in image processing field to call back such that it can perform actions of Image processing functions such as image segmentation, skin detection, level set technique, object extraction, separating colour image components and combining gray images, adding different noises and pre-processing operations such as different filtering to remove these noises.**

**A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. The designed GUI is easy to use and program. The results finally after executing may show that many image processing functions can be integrated in one file to be called and then get actions. The level set technique and extraction an object method can give very accurate and clear results. Finally it will be easy using MATLAB programming to get useful image processing toolbox.**

***Keywords***— **Edge Detection, Erosion, Region of interest.**

1. Introduction

Many functions in image processing field were integrated to call back. For example; vehicle number plate segmentation and extraction, where according to the image segmentation, the shape of an object can be described either in terms of its boundary or in terms of the region it occupies.

Gives the user a better view about each operation at the click of the button. This GUI can be used for any general image. The same GUI can be used for other operations by altering the call backs.

The objective of segmentation technique is to partition a given image into regions or components for extracting objects from an image based on examining an image on a pixel by-pixel basis and on utilizing the image information in a prescribed neighbourhood to established boundaries between regions in order to obtain good segmentation results

GUI eliminates the need to learn a language or type commands to run the application, by providing point-and-click control of software applications.

1. LITERATURE REVIEW

**Gerald K. Ijemaru1 , Augustine O. Nwajana2 , Emmanuel U. Oleka3 , Richard I. Otuka4 , Isibor,K.Ihianle5 ,Solomon H. Ebenuwa6Emenike Raymond Obi7**

Owing to recent technological advancement, computers and other devices running several image editing applications can be further exploited for digital image processing operations. This paper evaluates various image processing techniques using matrix laboratory (MATLAB-based analytics). Compared to the conventional techniques, MATLAB gives several advantages for image processing. MATLAB-based technique provides easy debugging with extensive data analysis and visualization, easy implementation and algorithmic-testing without recompilation. Besides, MATLAB's computational codes can be enhanced and exploited to process and create simulations of both still and video images. Moreover, MATLAB codes are much concise compared to C++, thus making it easier for perusing and troubleshooting. MATLAB can handle errors prior to execution by proposing various ways to make the codes faster. The proposed technique enables advanced image processing operations such as image cropping/resizing, image denoising, blur removal, and image sharpening. The study aims at providing readers with the most recent MATLAB-based image processing application tools. We also provide an empirical-based method using two-dimensional discrete cosine transform (2D-DCT) derived from its coefficients. Using the most recent algorithms running on MATLAB toolbox, we performed simulations to evaluate the performance of our proposed technique. The results largely present MATLAB as a veritable approach for image processing operations

**Nassir H. Salmana, Gullanar M. Hadib , Faculty of Computer science, Cihan university,Erbil, Iraq Faculty of Engineering-Software Engineering, Salaheldeen Univ., Erbil, Iraq**

paper many functions in image processing field were integrated to call back. For example; vehicle number plate segmentation and extraction, where according to the image segmentation, the shape of an object can be described either in terms of its boundary or in terms of the region it occupies. The objective of segmentation technique in this paper is to partition a given image into regions or components for extracting objects from an image (i.e vehicle number plate segmentation and extraction) based on examining an image on a pixel by-pixel basis and on utilizing the image information in a prescribed neighbourhood to established boundaries between regions in order to obtain good segmentation results (Pitas,2000)

**Nayana R. Shenoy1, Kamala C2 ,Vindhya K3**

Image processing is the manipulation and analysis of information contained in images. Image processing techniques help to improve the image characteristics. The basic operation that is usually performed on an image are edge detection, histogram, region of interest, bit planes, morphological operation, segmentation. Digital image processing is rapidly growing technologies and finds importance in various fields. A graphical user interface (GUI) is a set of techniques and mechanisms used to create interactive communication between a program and a user. GUI eliminates the need to learn a language or type commands to run the application ,by providing point-and-click control of software applications.

**Claudimar Pereira Da Veiga, Cássia Rita Pereira Da Veiga, Anderson Catapan, Wesley Vieira da Silva , November 2017 [4]**

The complex structure of skin tissue can make the analysis of highthroughput data manually inconvenient and leads to inaccurate analysis and time consumption. Therefore, automated system that can segment and detect features which might provide critical information for interesting phenotype is required. User friendly graphical user interface GUI in MATLAB can provide facilities to create a tool to enhance, segment and analyse images without having expert skills in image processing, this can be used in the study of skin morphology phenotyping to find interesting morphological and metabolic phenotypes. Using image processing capability facilitates to develop a tool to analyse a range of different images in term of intensity and quality because of the variation in histology performed in different laboratory. Consequently, develop of automated high-throughput bioimaging tool is considered to be a very important topic in disease diagnosis and drug development. Significant assessment of the morphological features in H&E skin section through the use of GUI MATLAB tool by quantifying all of epidermal and dermal thickness and number and size of adipocyte in subcutaneous. Using our developed tool, we were able to detect interesting epidermis, dermis and adipocyte phenotypes in mice skin sections. The Morphological Bio-imaging Tool provides facilities in the highthroughput analysis of H&E skin section to understand genetic basis of diseases

**Ahmet Selman Bozkir,** **Ebru Akcapinar Sezer, 2020 [5]**

Image processing is the manipulation and analysis of information contained in images. Image processing techniques help to improve the image characteristics. The basic operation that is usually performed on an image are edge detection, histogram, region of interest, bit planes, morphological operation, segmentation. Digital image processing is rapidly growing technologies and finds importance in various fields. A graphical user interface (GUI) is a set of techniques and mechanisms used to create interactive communication between a program and a user. GUI eliminates the need to learn a language or type commands to run the application by providing point-and-click control of software applications.

1. Proposed Methodology

A MATLAB GUI is a figure window to which we add user-operated controls . GUI is designed to integrate many image processing functions. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like buttons, list boxes .We can select, resize and position the components at any location. Using callbacks the components perform the required task when the user clicks or manipulates them with keystrokes. We can build MATLAB GUIs in two ways: i) Use GUIDE (GUI Development Environment), an interactive GUI construction kit. ii) Create code files that generate GUIs as functions or scripts (programmatic GUI construction). The first approach starts with a figure that we populate with components from within a graphic layout editor. GUIDE creates an associated code file containing callbacks for the GUI and its components. GUIDE saves both the figure (as a FIG-file) and the code file. Opening either one also opens the other to run the GUI. In the second approach, programmatic GUI-building approach, we create a code file that defines all component properties and behaviours; when a user executes the file, it creates a figure, populates it with components, and handles user interactions. The figure is not normally saved between sessions because the code in the file creates a new one each time it runs. IV. In our work we have developed a GUI using guide. The components used are text boxes, push button, pop-up menu, static text, axes. Push buttons generate an action when clicked. For example, an OK button might apply settings and close a dialog box. When you click a push button, it appears depressed; when you release the mouse button, the push button appears raised .Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively .Pop-up menus open to display a list of choices when users clicks the arrow. Axes enable your UI to display graphics such as graphs and images. Like all graphics objects, axes have properties that you can set to control many aspects of its behaviour and appearance.

1. RESULTS AND DISCUSSIONS
2. *RGB to grey conversion*

Converts the true colour image RGB to the grayscale image I. The rgb2gray function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. If you have Parallel Computing Toolbox installed, rgb2gray can perform this conversion on a GPU.

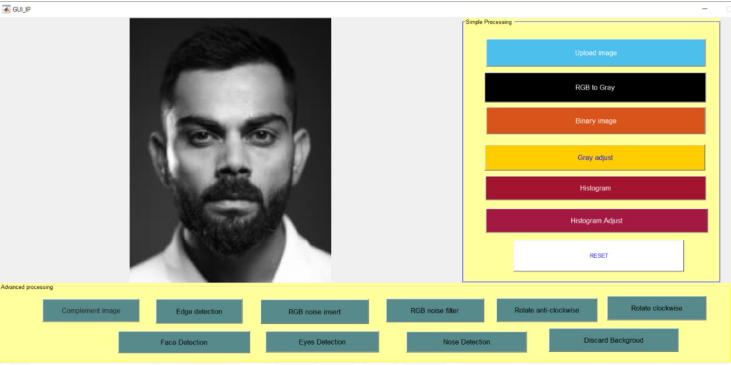


Fig 1.1 Rgb to grey conversion

1. Binary

Binary Image data are stored as an m-by-n logical matrix in which values of 0 and 1 are interpreted as black and white, respectively. Some toolbox functions can also interpret an m-by-n numeric matrix as a binary image, where values of 0 are black and all nonzero values are white.

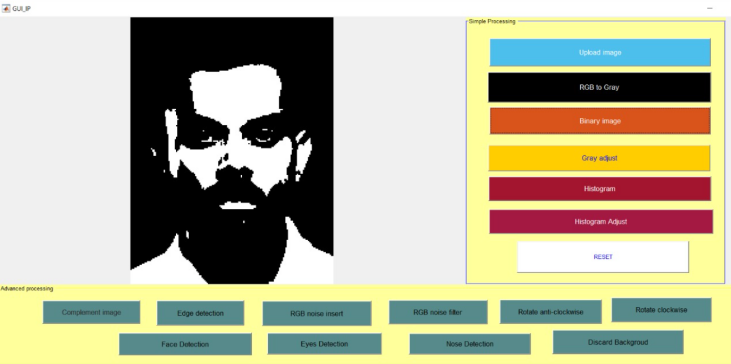


Fig 1.2 Binary

1. Gray adjust

A grayscale image is a data matrix whose values represent intensities of one image pixel. While grayscale images are rarely saved with a color map, MATLAB uses a color map to display them. You can obtain a grayscale image directly from a camera that acquires a single signal for each pixel.

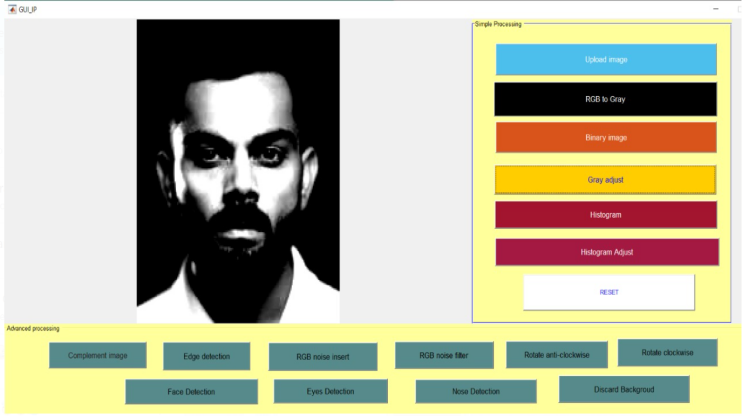


Fig 1.3 Gray adjust

1. *Histogram*

Histograms are a type of bar plot for numeric data that group the data into bins. After you create a Histogram object, you can modify aspects of the histogram by changing its property values. This is particularly useful for quickly modifying the properties of the bins or changing the display

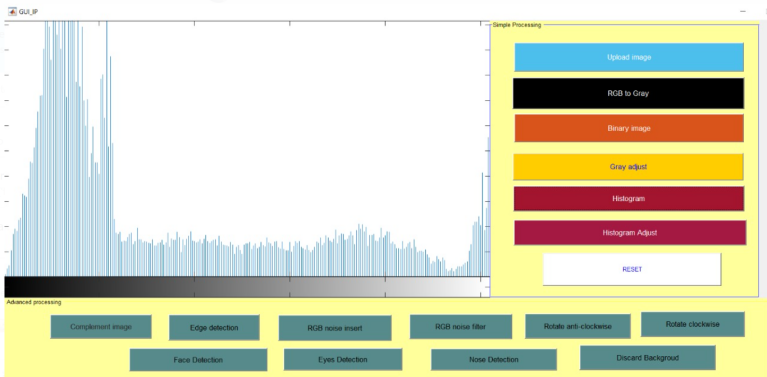


Fig 1.4 Histogram

1. *Histogram Adjust*

Histograms are a type of bar plot for numeric data that group the data into bins. After you create a Histogram object, you can modify aspects of the histogram by changing its property values. This is particularly useful for quickly modifying the properties of the bins or changing the display



Fig 1.5 Histogram Adjust

1. *Complement image*

Displays the binary image BW in a figure. For binary images, im show displays pixels with the value 0 (zero) as black and 1 as white. Imshow ( X , map ) displays the indexed image X with the colormap map . im show( filename ) displays the image stored in the graphics file specified by filename.

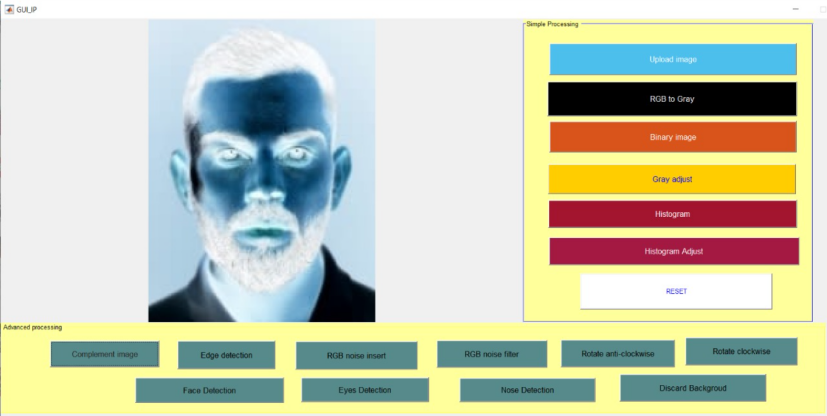


Fig 1.6 complete image

1. *Edge Detection*

Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.them are briefed in Module description session. Evaluation metric is the vital part of building a effective model as we get feedback from this metrics to improve the model further.

The edge function calculates the gradient using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges, including weak edges in the output if they are connected to strong edges.

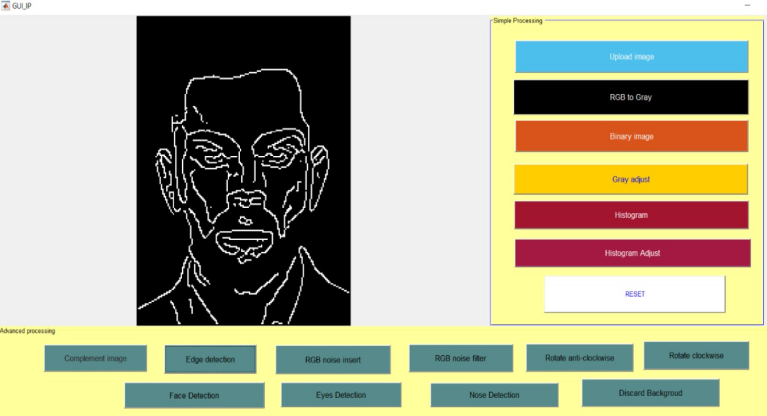


Fig 1.7 Edge Detection

1. *Face detection*

It is possible to achieve face recognition using MATLAB code. The built-in class and function in MATLAB can be used to detect the face, eyes, nose, and mouth. The object vision. Cascade Object Detector System of the computer vision system toolbox recognizes objects based on the Viola-Jones face detection algorithm.The KLT algorithm tracks a set of feature points across the video frames. Once the detection locates the face, the next step in the example identifies feature points that can be reliably.

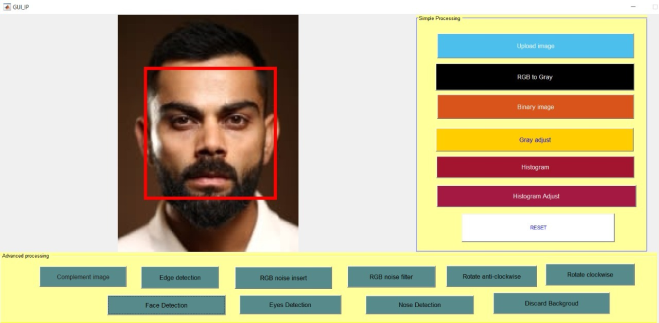


Fig 1.8 Face Detection

1. *Eye Detection*

First the command window is cleared and then the workspace is cleared. A video recording is started from which a snapshot is taken at regular intervals. This snapshots are then passed to the viola jones algorithm which is used for detection body parts from which we detected the Pair of Eyes and after getting the image we resized it so as to get only one eye this image is colored one, futher to just detect the eye pupil the image is converted into grayscale image and then passed on to be converted into black and white in which there are little amount of white holes, in order to eliminate this holes the medfilt2 and find functions are used. At the end both the results are plotted one grayscale image and other black-white image with the help of subplot function.

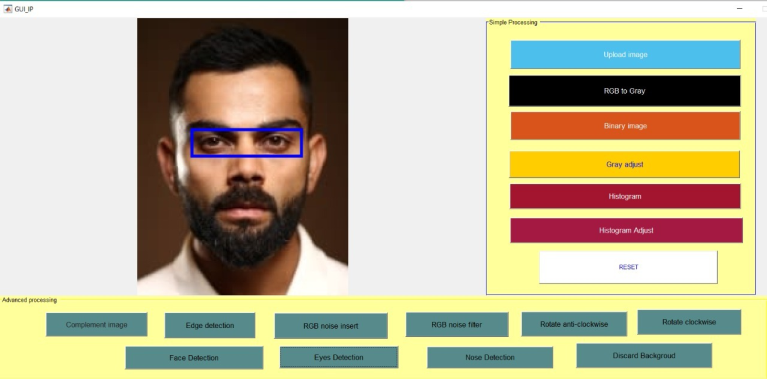


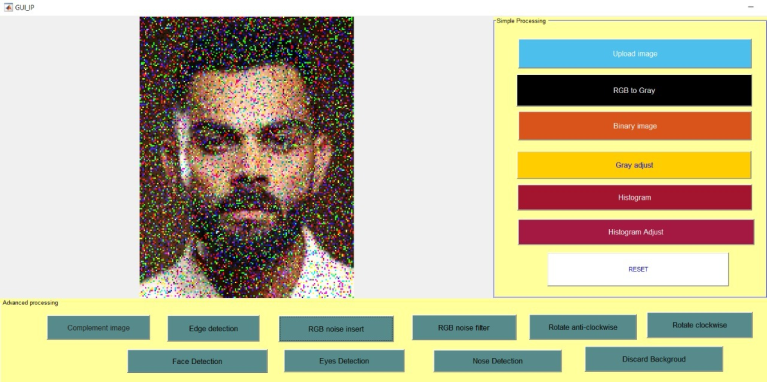
Fig 1.9 Eye detection

1. *RGB noise insert filter*

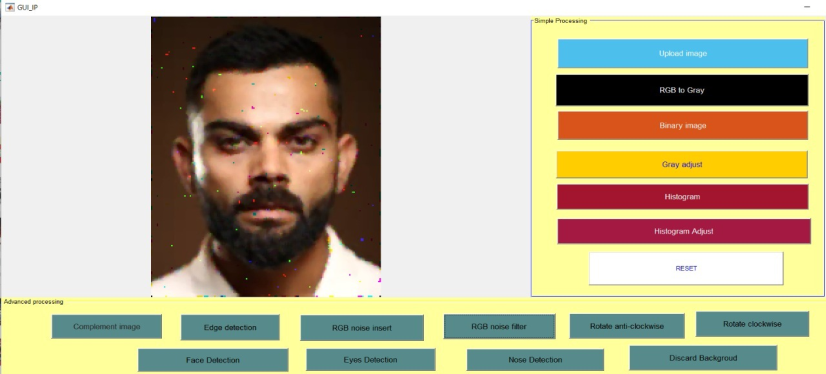
The RGB Noise filter adds a normally distributed noise to a layer or a selection. It uses the RGB color model to produce the noise (noise is added to red, green and blue values of each pixel). A normal distribution means, that only slight noise is added to the most pixels in the affected area, while less pixels are affected by more extreme values. (If you apply this filter to an image filled with a solid grey color and then look at its histogram, you will see a classic bell-shaped Gaussian curve.)

The result is very naturally looking noise.

This filter does not work with indexed images.

Fig 2.0 RGB noise insert filter

1. *RGB noise filter*

 Fig 2.1 RGB noise filter

1. *Nose detection*

Detection of facial features such as eye, nose, and mouth is an important step for many subsequent facial image analysis tasks such as face recognition. In this paper, we introduce a method to detect eye and nose fields from gray scale facial images. The Independent Components Analysis (ICA) is utilized to learn the appearance and shape of the facial feature. The regions with high response to the ICA basis vectors are chosen as the target facial features. For improving the performance further, the local characteristics of eye and nose are adapted besides ICA. Experiments on different databases show the promising results of the proposed method.

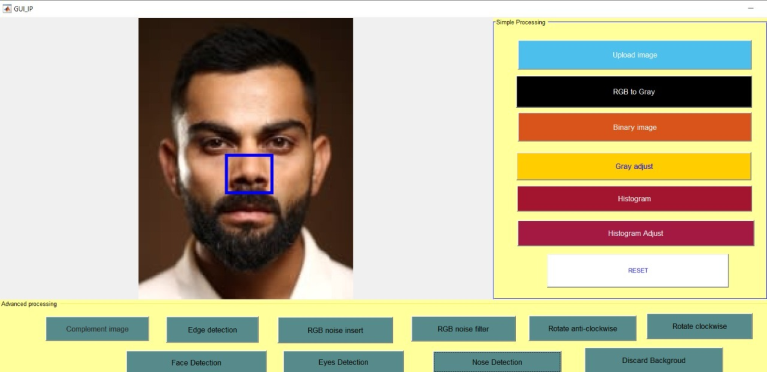


Fig 2.2 Nose detection

1. *Rotate clockwise*

The rotation operator performs a geometric transform which maps the position IMG_256 of a [picture element](https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixel.htm) in an input image onto a position IMG_257 in an output image by rotating it through a user-specified angle IMG_258 about an origin IMG_259. In most implementations, output locations IMG_260 which are outside the boundary of the image are ignored. Rotation is most commonly used to improve the visual appearance of an image, although it can be useful as a preprocessor in applications where directional operators are involved. Rotation is a special case of [affine transformation](https://homepages.inf.ed.ac.uk/rbf/HIPR2/affine.htm).

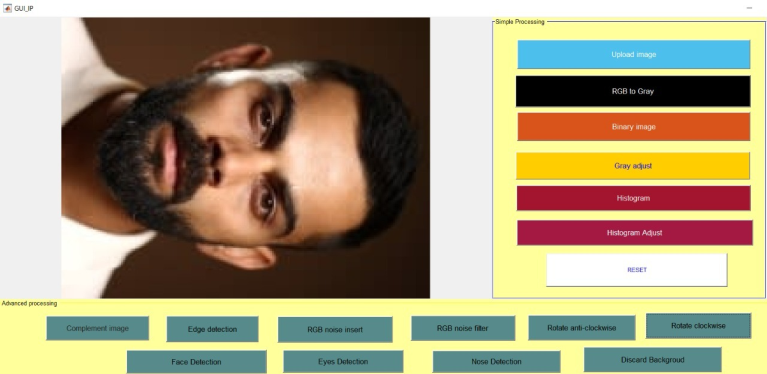


Fig 2.4 rotate clockwise

1. *Rotate anti-clockwise*

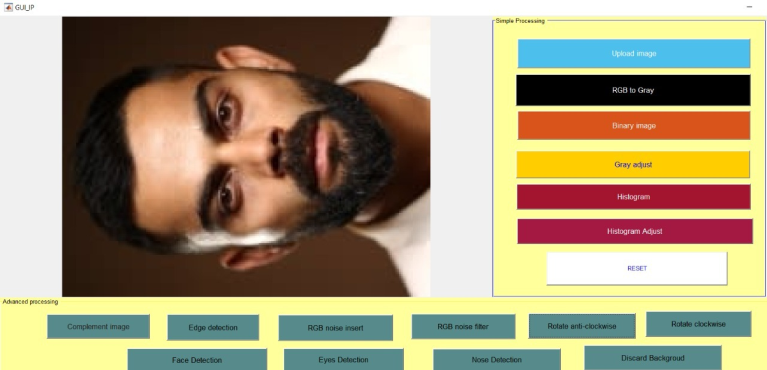


Fig 2.3 rotate anti-clockwise

1. Discard back ground

The idea of background removal is not a modern one; instead, it has existed in the field since the dawn of modern technology. Hence, there have been numerous techniques for background removal in the domain of computer vision. Nevertheless, classical computer vision techniques are constantly being overthrown by modern, artificial intelligence-based ones by lodging more innovative ideas giving more precise and accurate results.

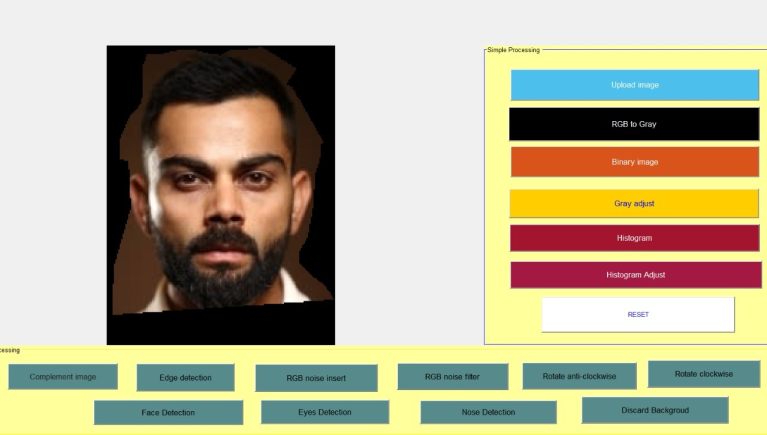


Fig 2.4Discard background

1. Conclusion

Basic important concepts of image processing are briefly presented in this paper. The GUI developed performs the basic operations. It gives the user a better view about each operation at the click of the button. This GUI can be used for any general image .The same GUI can be used for other operations by altering the callbacks. Most of the functions presented in this paper can be further investigated and their performance can be improved. Future work aims at expanding the set of applications, calculating the statistical improvement after the application of the image processing techniques.

A graphic user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. The designed GUI is easy to use and program. The results show that many image processing functions can be integrated in one file to be called and then get actions. There are very clear results, where the origin image and the processed result appeared in one figure to compare easy. The level set technique and extraction an object method give very accurate and clear results. Finally it is easy using MATLAB programming to get useful image processing toolbox.

1. References

[1] Milan Sonka, Vaclav Hlavac, Roger Boyle, “Image Processing , Analysis, and Machine Vision”Cengage learning,2019

[2] R. C. Gonzalez and R. E. Woods, “Digital Image Processing” Gatesmark Publishing, 2020.

[3] Raquib Buksh, Soumyajit Routh, Parthib Mitra, Subhajit Banik, Abhishek Mallik, Sauvik Dass Gupta, ”Matlab based image editing and color detection” ,International journal of Science and Research publications, Vol 4 ,issue 1,Jan 2016, 1-6.

[4] Thabit Sultan Mohammed,Wisam F.Al-Azzo ,Khalid Mohammed Al Mashani,”Image processing development and implementation :A software simulation using Matlab”,ICIT ,Mar2019

[5] NassirH Salman,Gullanar M Hadi,”Integrated image processing functions using Matlab GUI”,Journal of advanced computer science and technology research,vol 3 no 1,March 2017,31-38.

[6] www.mathworks.in.