

A PROJECT REPORT

ON

PYTHON GUI FOR IMAGE PROCESSING APPLICATIONS

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

Aadhya Bajpai (171075)

UNDER THE SUPERVISION OF

Dr. Nishant Jain

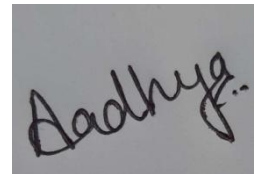


JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

MAY, 2021

DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled “**Python GUI for Image Processing Applications**” submitted at **Jaypee University of Information Technology, Waknaghat, India** is a true full proved record of the work I carried out under the supervision of **Dr. Nishant Jain**. I have not submitted my work anywhere else for some other degree.



Aadhya Bajpai

171075

This is to bring in notice that the above mentioned statements made by the candidate are correctly verified by the concerned authorities and there are no copyright issues.



Dr. Nishant Jain
Date: 24th May 2021

Head of the Department/Project Coordinator

ACKNOWLEDGEMENT

I firstly thank God for giving me the strength and courage which were key features for the completion of the project. It acknowledges all the people who gave me their extreme support in making the project successful. I would like to thank Head of the E.C.E. Department, Prof. Rajeev Kumar for his constant encouragement throughout the making of the project. My special thanks to my Project guide Dr. NISHANT JAIN, Associate Professor, Electronics and Communications engineering, for his constant support and guidance regarding the project work. I am also thankful to all the faculty members of Electronics and Communications Engineering department for their valuable advice that helped me to complete the project. In the end, I would like to extend my thanks to my beloved parents for their support at each and every step of progress I made in the project work and continuous cooperation in any way possible for the betterment of the project.

PYTHON GUI FOR IMAGE PROCESSING APPLICATIONS

ABSTRACT

One picture is equivalent to thousand words, as there is a lot of information hidden in an image. A picture has an important role in many fields like medicine (ultrasound, X-ray), infrastructural projects and many other areas. In today's scenario, there are many tools and devices to acquire a picture. For example, obtaining an X-ray image from the X-ray machine. Also, today every person uses a smart phone or a camera to click pictures of various objects present around them.

While capturing an image through different devices, we sometimes notice issues in the captured image, most commonly blurring which might occur due to the device used for acquisition of an image. It is possible that the images blurred were part of some important project, and the person may not have the choice of clicking that picture again. Even on retaking an image through the same device, the pixel resolution will not change and thus the desired information is not obtained from an image. In such situations, Image Processing is very helpful.

INDEX

CONTENTS

ABSTRACT

1. INRODUCTION.....	09
1.1. Problem Statement.....	09
1.1.1. What is GUI.....	09
1.1.2. Why python GUI.....	09
1.2. What is tKinter.....	10
1.2.1. Why tKinter.....	10
2. LITERATURE SURVEY.....	11
2.1. ImageJ GUI.....	11
2.2. Matlab GUI.....	11
2.3. 3 tKinter geometry managers.....	11
2.3.1. Pack.....	12
2.3.2. Place.....	12
2.3.3. Grid.....	12
3. DIGITAL IMAGE PROCESSING.....	13
3.1. Digital Image Processing.....	13
3.2. Fundamental Step In Image Processing.....	14
3.3. Elements of Digital Image Processing Systems.....	15
3.3.1. A Simple Image Formation Model.....	15
3.4. Image Enhancement.....	16
3.4.1. Methods of Image Enhancement.....	17
3.4.1.1. Histogram Equalization.....	17

3.4.1.2. Median Filtering.....	17
3.4.1.3. Unsharp Mask Filtering.....	18
3.4.2. Methods of Image Segmentation.....	18
3.4.2.1. Edge Based Segmentation.....	18
4. BUILDING A GUI USING PYTHON.....	20
4.1. Steps for creating GUI.....	20
4.1.1. Import the module.....	20
4.1.2. Creating buttons.....	20
4.1.3. Implementing pack manager.....	20
4.1.4. Layout improvement.....	21
4.1.5. Adding some widgets.....	22
4.1.5.1. Buttons.....	22
4.1.5.2. Canvas.....	22
4.1.5.3. Checkbuttons.....	23
4.1.5.4. Entry.....	23
4.1.5.5. Frame.....	23
4.1.5.6. Listbox.....	24
4.1.5.7. Menu.....	24
4.1.5.8. Message.....	24
4.1.5.9. Radiobuttons.....	25
4.1.5.10. Text.....	25
4.1.6. Python Libraries Used.....	25
4.1.6.1. Tkinter.....	25
4.1.6.2. Pillow Library.....	25

4.1.6.3. Image Module in Pillow Library.....	25
4.1.6.4. ImageTK Module in Pillow Library	25
4.1.6.5. ImageOps Module in Pillow Library	26
4.1.6.6. ImageEnhance Module in Pillow Library	26
4.1.6.7. Tkinter Import Filedialog	26
4.1.6.8. Numpy	26
4.1.6.9. ImageFilter Module in Pillow Library	26
5. APPLICATION PROTOTYPE.....	27
6. CONCLUSIONS.....	39
REFERENCES.....	40

LIST OF FIGURES

1.	Fundamental Steps in Digital Image Processing.....	14
2.	Elements of Digital Image Processing.....	15
3.	Enhancing grayscale images with the help of Histogram Equalization.....	17
4.	Median Filtering.....	17
5.	Unsharp Mask Filtering.....	18
6.	Edge Based Segmentation.....	19

CHAPTER – 1

INTRODUCTION

1.1. PROBLEM STATEMENT

One picture is equivalent to thousand words, as there is a lot of information hidden in an image. A picture has an important role in many fields like medicine (ultrasound, X-ray), infrastructural projects and many other areas. In today's scenario, there are many tools and devices to acquire a picture. For example, obtaining a X-ray image from the X-ray machine. Also, today every person uses a smart phone or a camera to click pictures of various objects present around them.

While capturing an image through different devices, we sometimes notice issues in the captured image, most commonly blurring which might occur due to the device used for acquisition of an image. It is possible that the images blurred were part of some important project, and the person may not have the choice of clicking that picture again. Even on retaking an image through the same device, the pixel resolution will not change and thus the desired information is not obtained from an image. In such situations, Image Processing is very helpful.

1.1.1. WHAT IS GUI?

Graphical User Interface is a type of user interface which allows users to work with various electronic devices in a friendly environment. It uses different graphical icons and features, instead of text based interface. Thus, a user can simply select different widgets and has nothing to write a code or commands for functioning it. The user has multiple options on a screen and this gives a sense of joy. The GUIs are used not only for computer systems, but also mobile devices. The objects present in GUI are buttons, labels, list box, etc.

1.1.2. WHY PYTHON GUI?

Python has many options for GUI development. Also python programming is easy to read and understand. The code length is small. It allows multiple programming paradigms. It has a vast and robust library for standard functions. Also, it offers many tools and frameworks for open

source, and simplifies complex software developments. It supports many major systems and platforms, hence GUI can be functioned on various devices easily.

1.2. WHAT IS TKINTER?

Tkinter is one of the most commonly used python library for developing GUI. It comes with Tk GUI toolkit. It doesn't require any license for the users to use, hence it is a free software. Tkinter is compatible with many major platforms like Unix, Linux, Microsoft windows, Mac OS, etc. It has many built in features to create various widgets like buttons, check buttons, menu button, allows user to write a text on or around an image and many other options.

1.2.1. WHY TKINTER?

Although Python has several options available for GUI designing for applications like PySimple GUI, Wax, PySide, Libavg, tKinter, PyQt, Kivy, etc; out of these tKinter is the fastest and easiest way to create GUI applications. Creating a GUI using tKinter is an easy task. Tkinter is pre installed in Python 3, so the user don't have to install it separately.

CHAPTER – 2

LITERATURE SURVEY

2.1. IMAGEJ

ImageJ is an open source JAVA GUI. It is particularly an image processing platform. In this, the data acquisition, analysis and image processing part are done on its editor and compiler. The plugins used are custom or self developed that can be used for analyzing image, from live imaging of cell [1] to radiological image processing, [2] multiple data comparisons of images [3] to automated haematology systems. [4] The plugins used in imageJ's are extremely popular for image processing techniques. [5] ImageJ can be easily downloaded on various mobile devices, and on any computer with pre-installed Java 5 and latest virtual machine.

Some of the image processing features in imageJ are Image Acquisition, filtering techniques like median filtering and Gaussian blur, segmentation techniques to separate the intersecting or touching particles in an image. Some segmentation involves converting an original image into a binary image.

2.2. MATLAB GUI

MATLAB matrix laboratory is a software for computing matrices. MATLAB GUI is a language for high level programming and provides a user friendly environment which allows you to perform computational tasks faster and more efficient than that with traditional programming languages such as C++, JAVA and FORTRAN. MATLAB is mainly used for numerical computation and applications like deep learning, machine learning, signal and image processing and many others. It was developed by Te Math Work. MATLAB allows data and function plotting, implementation of algorithms, creating user interfaces, connecting and sharing with programs in other languages [6].

2.3. TKINTER GEOMETRY MANAGERS

TKinter provides the user with geometrical managers of the widgets to properly organize and arrange widgets in the parent window. A single window or frame uses a single geometry manager. Thus, tKinter geometry managers controls the overall layout of the GUI. There are 3 main geometry managers in tKinter:

2.3.1. PACK

This method basically uses the packing algorithm for placing the widgets in a frame in a specified manner or a particular block. It firstly calculates the area in the rectangular block with blank space. By default, it puts the widget at the center location, until user defines the location for widget. The user can pass arguments like fill, side and expand.

The fill argument is used to give the direction to the frames. The frames can be filled in a horizontal (tk.X) or vertical direction (tk.Y), and to fill in both the directions, we use tk.BOTH. the side argument is to pack the widgets to a particular side (top, bottom, left and right). The expand argument is used to indicate that whether the widgets should be expanded to fill the space that is extra in a block, if not FALSE is passed as an argument.

2.3.2. PLACE

It organizes the widgets in X and Y coordinates (pixels). (0,0) marks the top left corner of a frame. Y tells the number of pixels from top of the window and X tells the number of pixels from left of the window.

2.3.3. GRID

It is the most commonly used manager as it provides all the powers of pack() function, in a simpler and efficient way. The user can simply give the arguments to rows and columns to specify the location of the widget.

CHAPTER – 3

DIGITAL IMAGE PROCESSING

3.1 DIGITAL IMAGE PROCESSING

The pillars of digital image processing are mainly built from 2 principal areas of interest:

1. High picture quality for predicting the humans correctly.
2. Processing is done on train dataset for our model to perform well.

In the latter, interest follows on performing the list of steps for extraction the desired information from the image in a suitable processing format.

Image:

An image is nothing but a 2D light intensity function $f(x, y)$, which signifies the brighter or darker levels in an image at that point, for different spatial coordinates x and y . A digital image is encoded in discrete values for both spatial coordinates and intensity level. We thereby form an array of digital image, which comprises of image elements or pixel values.

A simple image model:

For computing processor suitability, the image is digitally encoded in both spatial coordinates and function height. Such a digitalization done for coordinates is called image sampling. Digitalization done for the intensity function is called gray-level quantization.

With the increase of picture quality resolution or pixel value, and gray levels, the storage also increases which demands high processor for it.

Example: A gray-scale image of pixel value size 256×256 requires a space of 64k bytes of memory.

3 Types of image processing

- Low level
- Medium level
- High level

A lower level processing is responsible for carrying out primary tasks like image reading, resizing, rotating an image, converting color form (RGB to gray), drawing histograms etc. We obtain the raw image after this processing. Medium level processing is used for extraction of important features from the output of the previously formed image. It correctly identifies the boundaries (edges) in an image, which is called segmentation. High level processing then adds artificial intelligence concepts for making our system intelligent and smart to learn new things by it.

3.2. FUNDAMENTAL STEPS IN IMAGE PROCESSING

1. Acquiring a raw digital image
2. Image pre-processing: enhancing the features of image such that the probability of success of further steps performed increases.
3. Image segmentation: to segment the various parts present in an input image.
4. Representing image: For applying the processing techniques, raw image is converted into suitable format.
5. Image description: extracting such feature that differentiates the classes for objects.

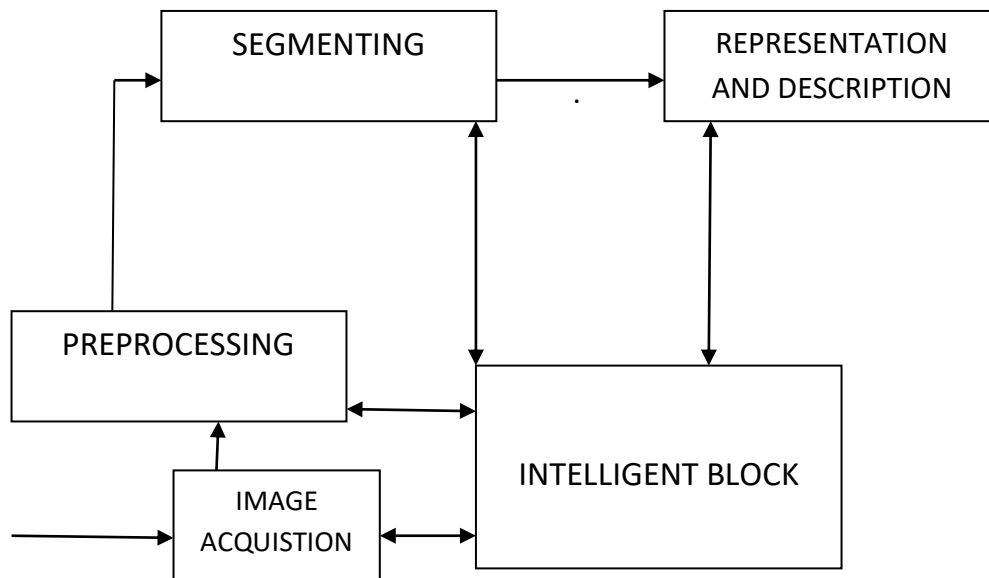


FIG.3.1. FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING

3.3. ELEMENTS OF DIGITAL IMAGE PROCESSING SYSTEMS

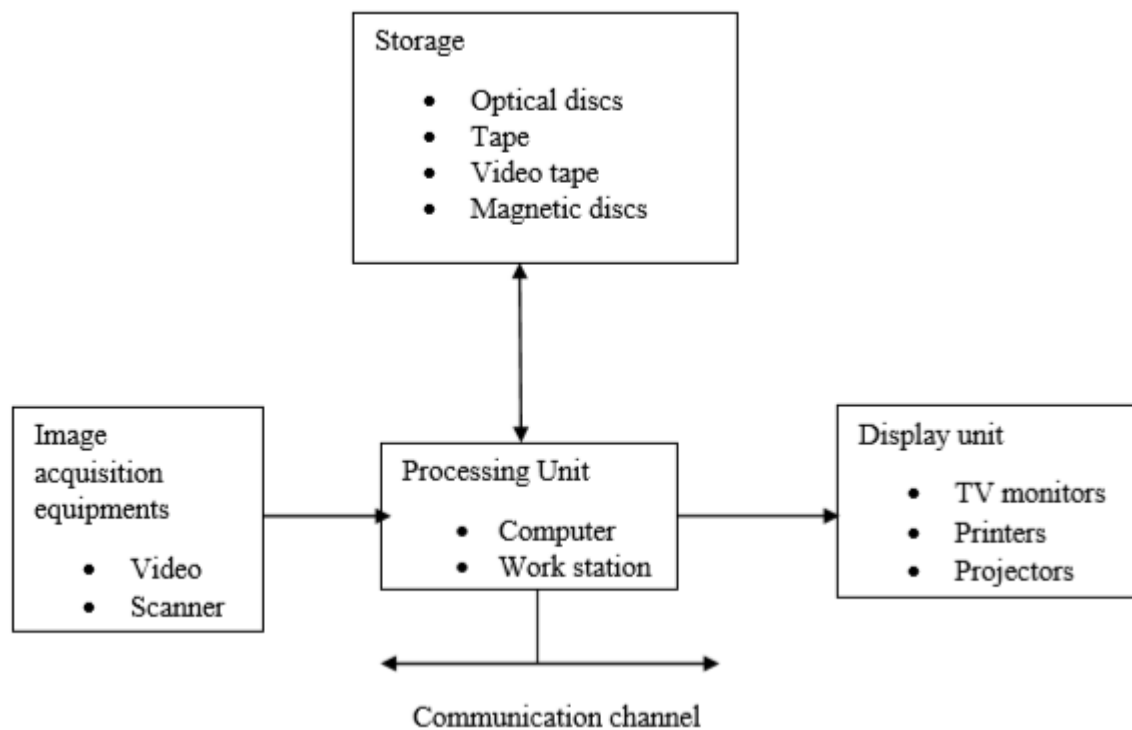


FIG.3.2. ELEMENTS OF DIGITAL IMAGE PROCESSING

The important stages involved in processing of the digital image are:

1. Image acquired
2. Storing capacity
3. Processor required
4. Channel for communicating
5. Displaying the final image

3.3.1. A SIMPLE IMAGE FORMATION MODEL

Image is represented as a 2-D function $f(x, y)$, which consists of 2 spatial components x and y :

1. The intensity of the source light $I(x, y)$ incident on an image.
2. The intensity of reflected light $R(x, y)$ from the objects of an image.
3. $f(x, y) = I(x, y)R(x, y)$; where $0 < I(x, y) < 1$ and $0 < R(x, y) < 1$

Format name	Full name	Description	Recognized extensions
TIFF	Tagged Image File Format	A flexible file format supporting a variety image compression standards including JPEG	.tif, .tiff
JPEG	Joint Photographic Experts Group	A standard for compression of images of photographic quality	.jpg, .jpeg
GIF	Graphics Interchange Format	Frequently used to make small animations on the internet	.gif
BMP	Windows Bitmap	Format used mainly for simple uncompressed images	.bmp
PNG	Portable Network Graphics	Compresses full color images with transparency(up to 48bits/p	.png

TABLE 3.3. DIFFERENT IMAGE FORMATS

3.4. IMAGE ENHANCEMENT

Image Enhancement is the methodology of improving the quality and data content of the actual information or data prior to processing. It is a way towards changing or adjusting of digital images so that the outcomes which we obtain are more reasonable to show or to do further image analysis.

For example-

- It can remove unwanted noise and disturbances.
- It makes it simpler to distinguish key highlights.
- It sharpens up the image.
- It brightens up an image.

3.4.1. METHODS OF IMAGE ENHANCEMENT

3.4.1.1. HISTOGRAM EQUALIZATION

Histogram Equalization is a method for changing image intensities to improve contrast of the image. This strategy typically builds or increases the global contrast of numerous images, particularly when the usable information of the image is addressed by close contrast values.



Fig 3.3- Enhancing grayscale images with the help of Histogram Equalization.

3.4.1.2. MEDIAN FILTERING

Median Filtering is a technique which is usually used to eliminate noise and disturbance from images. This type of decrease in noise and disturbances is a typical pre processing step to enhance the consequences of later processing.

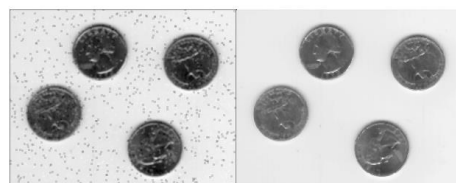


Fig 3.4- Median Filtering

3.4.1.3. UNSHARP MASK FILTERING

The unsharp Mask Filtering Technique is a basic technique which hones or sharpens the operator which further gets its name from that it improves, sharpens and enhances edges through a methodology which deducts smoothed, unsharpened redintion of an image from its original picture.



Fig 3.5- Unsharp Mask Filtering

3.4.2. METHODS OF IMAGE SEGMENTATION

3.4.2.1. EDGE BASED SEGMENTATION

Edge detection is the way toward finding edges in a picture which is a vital advance towards understanding picture highlights. It is accepted that edges comprise of significant highlights and contain huge data. It altogether diminishes the size of the picture that will be prepared and sift through data that might be viewed as less applicable, safeguarding and zeroing in exclusively on the significant underlying properties of a picture for a business issue.

Edge-based segmentation calculations work to identify edges in a picture, in view of different discontinuities in grey level, shading, surface, brilliance, immersion, contrast and so on To additional upgrade the outcomes, advantageous preparing steps should follow to link every one of the edges into edge chains that relate better with borders in the picture.



Fig 3.6- Edge Based Segmentation

CHAPTER – 4

BUILDING A GUI USING PYTHON

4.1. STEPS FOR CREATING GUI

There are several steps involved in creating GUI for an application such as importing a desired module which is best fit for the application, then adding various widgets like button, list box, etc and implementing the geometry managers for improving the GUI layout. And finally, we can ensemble our model by adding key elements from other GUIs.

4.1.1. IMPORT THE MODULE

The first step for starting our work in an application is importing the desired module. It can be done by importing the module in user's namespace. It allows user to classes and constants by their names. Therefore, we can label our own text instead of using the default tkinter label. It is relatively easy and time consuming approach.

User can use `>> from Tkinter import` to import the necessary module.

Thus, now user can start building its GUI by following further steps.

4.1.2. CREATING BUTTONS

Now, once the Tkinter starts running, the user can add buttons to its GUI. Buttons are the most simplest of all widgets in the GUI applications.

```
>>b = Button(win, text="Hello")
```

The window created previously acts as a parent for the button added now. But after writing the code, the buttons aren't available on the screen because of absence of geometry managers (pack, grid and place). Therefore, next step would be implementing these geometrical managers.

4.1.3. IMPLEMENTING PACK MANAGER

Now, we will use the pack manager for adding buttons to the parent window. The user can define that where he wants his child widget to be packed in a parent widget by using functions

TOP, BOTTOM, LEFT, RIGHT. By default, the program considers the function TOP, until mentioned by the user.

```
>> b.pack()
```

The window adjusts itself according to the button size. The pack manager places the widgets in horizontal or vertical direction only. For placing the widget in any other way, we use GRID manager that uses x and y coordinates for placing the widget.

4.1.4. GRID MANAGER

Grid manager allows the parent class to divide into rows and columns. The child widget can be placed in parent grid's cell. The modifications on the length and the width of every row and column can be done as per the user's choice. The grid manager gives user the complete freedom in designing the widgets.

```
>> b = Button(win, text="HELLO")
```

```
>> b.grid(row=1, column=1)
```

4.1.4. IMPROVING LAYOUT

The frame contains various widgets. A user can pack many widgets in a single frame. Also, many frames can be packed inside a single frame. Thus, with these features, advance GUI layouts can be created.

The buttons created before are unresponsive; they do not function until the user assigns them some particular task to perform.

```
>> b.configure(text="highlight")
```

```
>> def bx()
```

```
print("Button one was pushed")
```

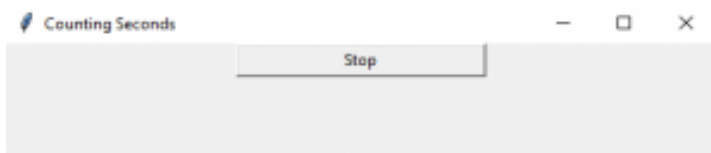
```
>> b.configure(command=bx)
```

So, when the user clicks the button, the respective message will be displayed in the window.

4.1.5. ADDING SOME WIDGETS

4.1.5.1. BUTTONS

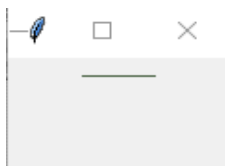
```
import tkinter as tk
r = tk.Tk()
r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=25, command=r.destroy)
button.pack()
r.mainloop()
```



4.1.5.2. CANVAS

It is used in drawing pictures and some complex layouts like texts, graphics and widgets.

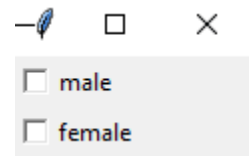
```
from tkinter import *
master = Tk()
w = Canvas(master, width=400, height=600)
w.pack()
canvas_height=200
canvas_width=200
y = int(canvas_height / 2)
w.create_line(0, y, canvas_width, y )
mainloop()
```



4.1.5.3. CHECKBUTTONS

The user can select any option from variety of options available as a toggle buttons.

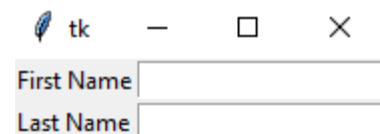
```
from tkinter import *
master = Tk()
var1 = IntVar()
Checkbox(master, text='male', variable=var1).grid(row=0, sticky=W)
var2 = IntVar()
Checkbox(master, text='female', variable=var2).grid(row=1, sticky=W)
mainloop()
```



4.1.5.4. ENTRY

This widget allows the user to input the single line text.

```
from tkinter import *
master = Tk()
Label(master, text='First Name').grid(row=0)
Label(master, text='Last Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
mainloop()
```



4.1.5.5. FRAME

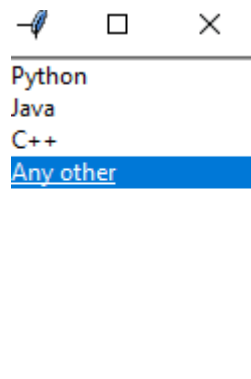
It acts a main window that comprises of all the widgets. It is also used for organizing and arranging the widgets.

4.1.5.6. LISTBOX

It provides user list of options to choose from.

```
from tkinter import *
```

```
top = Tk()
Lb = Listbox(top)
Lb.insert(1, 'Python')
Lb.insert(2, 'Java')
Lb.insert(3, 'C++')
Lb.insert(4, 'Any other')
Lb.pack()
top.mainloop()
```



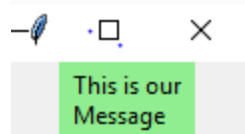
4.1.5.7. MENU

It is used to create all kinds of menu options used in an application.

4.1.5.8. MESSAGE

It is a multi line text. It cannot be edited once declared by the user.

```
from tkinter import *
main = Tk()
ourMessage = 'This is our Message'
messageVar = Message(main, text = ourMessage)
messageVar.config(bg='lightgreen')
messageVar.pack( )
main.mainloop( )
```



4.1.5.9. RADIOBUTTON

The user can choose one option from the several options available.

4.1.5.10. TEXT

It is used to format or edit the multi line text.

4.1.6. PYTHON LIBRARIES USED

4.1.6.1 TKINTER

Tkinter is one of the most commonly used python library for developing GUI. It doesn't require any license for the users to use, hence it is a free software. Tkinter is compatible with many major platforms like Unix, Linux, Microsoft windows, Mac OS, etc. It has many built in features to create various widgets like buttons, check buttons, menu button, allows user to write a text on or around an image and many other options. Although Python has several options available for GUI designing for applications like PySimple GUI, Wax, PySide, Libavg, tKinter, PyQt, Kivy, etc; out of these tKinter is the fastest and easiest way to create GUI applications. Creating a GUI using tKinter is an easy task. Tkinter is pre installed in Python 3, so the user don't have to install it separately.

4.1.6.2 PILLOW LIBRARY

Pillow library has all the basic image processing functionalities. It can perform functions like image transformation, image rotation, image resizing and many more. This module permits to haul a few statistical data out of the image utilizing histogram strategy, which further moving on can be utilized for automatic contrast enhancement and statistical analysis.

4.1.6.3 IMAGE MODULE IN PILLOW LIBRARY

Image Module in Pillow Library gives a class a similar same which is utilized to address a PIL Image. This module additionally gives various factory functions, including functions to create new images and to load images from files.

4.1.6.4 IMAGETK MODULE IN PILLOW LIBRARY

ImageTK Module in Pillow Library contains support to make and adjust Tkinter BitmapImage and PhotoImage objects from PIL images. Filedialog is utilized for the dialog box to appear whenever you are opening file from anyplace in system or saving file at a specific location or spot.

4.1.6.5 IMAGEOPS MODULE IN PILLOW LIBRARY

ImageOps Module in Pillow Library contains various instant image processing operations. This module to some degree is trial or experimental, and mainly operators work on L and RGB images.

4.1.6.6 IMAGEENHANCE MODULE IN PILLOW LIBRARY

ImageEnhance Module in Pillow Library contains various classes which can be utilized for image enhancement.

4.1.6.7 TKINTER IMPORT FILEDIALOG

Python Tkinter offers a bunch of dialogs that can be utilized while working on with the documents. By utilizing these we do not need to plan standard dialog exchanges by ourselves. File dialogs helps in opening of the files, saving of the files or even directories. It allows us to open a single file, a directory, save as file and substantially much more.

4.1.6.8 NUMPY

NumPy is one of the most important or in other words one of the main libraries in Python as it provides support for arrays. An image is essentially a standard NumPy array containing pixels of data points. Therefore, pixel values of an image can be modified by using basic NumPy operations, such as slicing, masking, and fancy indexing.

4.1.6.9 IMAGEFILTER MODULE IN PILLOW LIBRARY

ImageFilter Module in Pillow Library contains definitions of pre-defined set of filters, which can be utilized with Image.filter() method.

The current adaptation of the library provides the accompanying arrangement of predefined image enhancement filters-

- Detail
- Contour
- Blur
- Edge_Enhance
- Edge_Enhance_More
- Smooth_More
- Smooth
- Emboss
- Sharpen
- Find_Edges

This strategy permits you to apply practically any shading change in constant time with the use of pre-determined decimated tables.

CHAPTER – 5

APPLICATION PROTOTYPE

5.1 GUI INTERFACE OF IMAGE EDITOR

This is the main GUI interface for Image Editor.

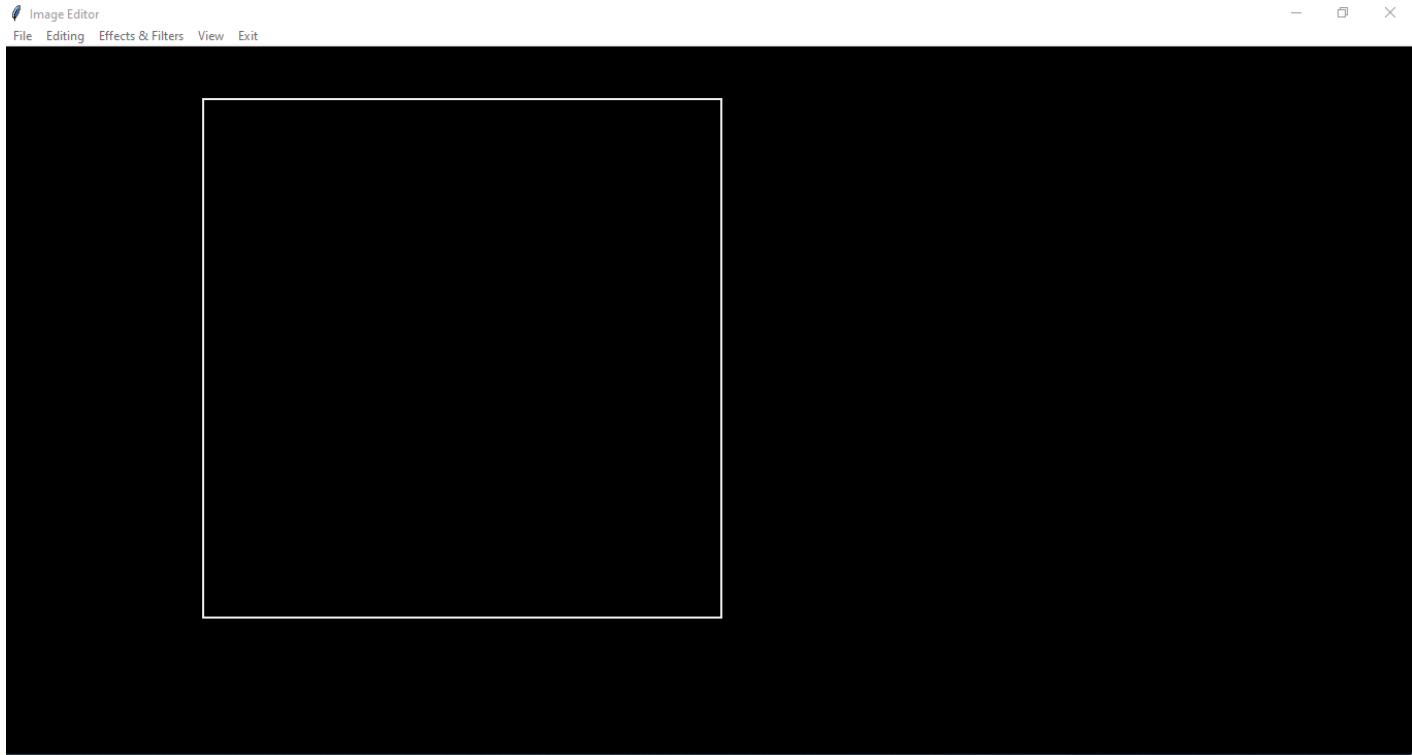


Fig 5.1

5.2 DROPDOWN MENU

This Image Editor can edit photos by selecting the images present in the secondary storage of the device.

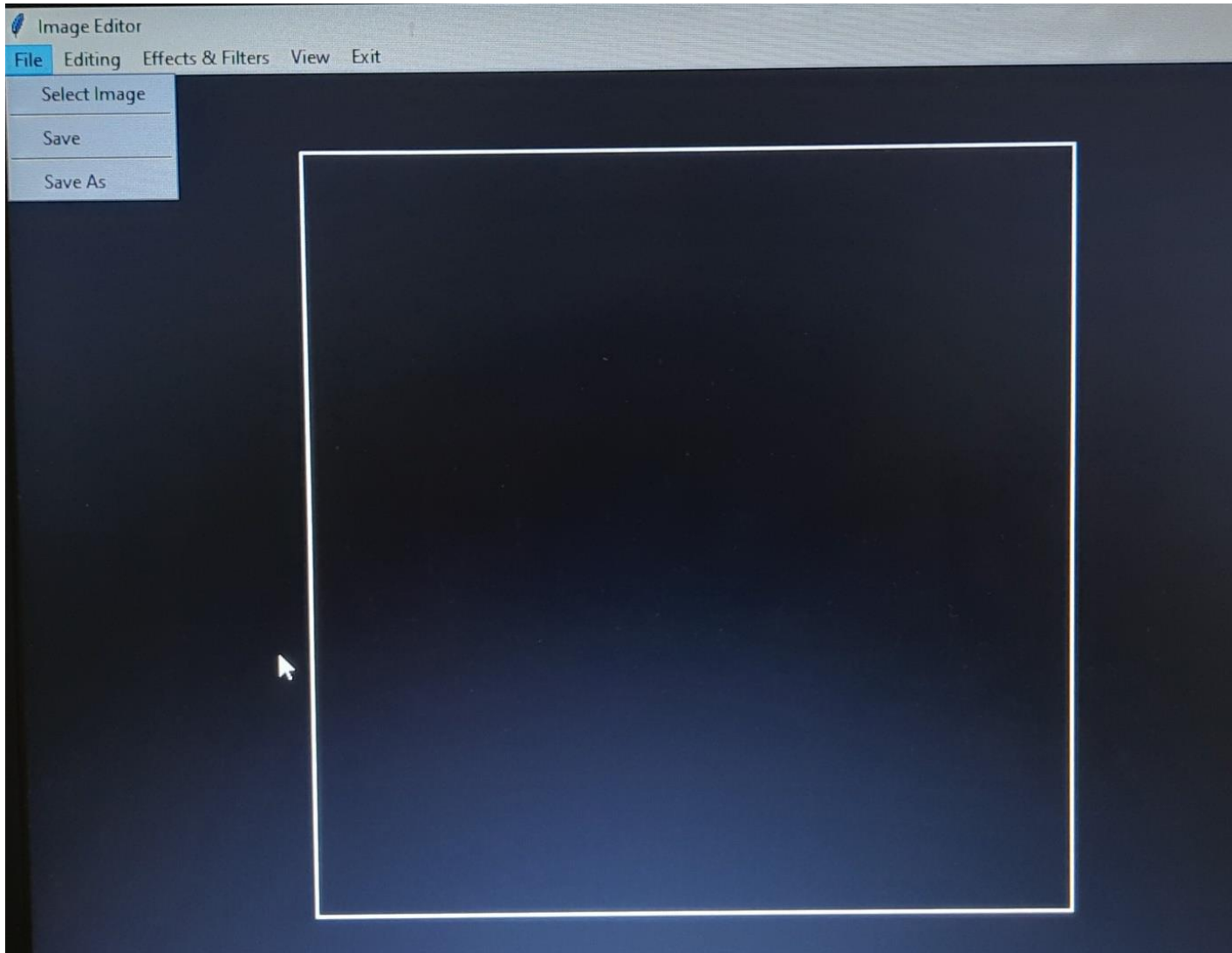


Fig 5.2

5.3 SELECTED AN IMAGE

After clicking on “Select Image” from the dropdown menu the image will be selected.



Fig 5.3

5.4 CROPPED IMAGE AND OTHER EDITING FEATURES

We can crop the selected image by clicking on “Editing” button and a dropdown menu will appear from which we can further select “Crop” button and it will automatically crop the image So that the image fits perfectly in the canvas of the Image Editor.



Fig 5.4

5.5 APPLIED FILTERS IN THE IMAGE

Now moving on, if we want to change the enhancements of the image then we can select the Automatic Mode and it will directly change filters and brightness of picture and enhance the image. If we want to enhance the image by using more filters then custom filters options is also available and we can select other filter and effects.



Fig 5.5

5.6 NEW ADDITIONS IN THE GUI

I have added to new buttons in the Image Editor GUI which are-:

- Image Segmentation
- Image Enhancement

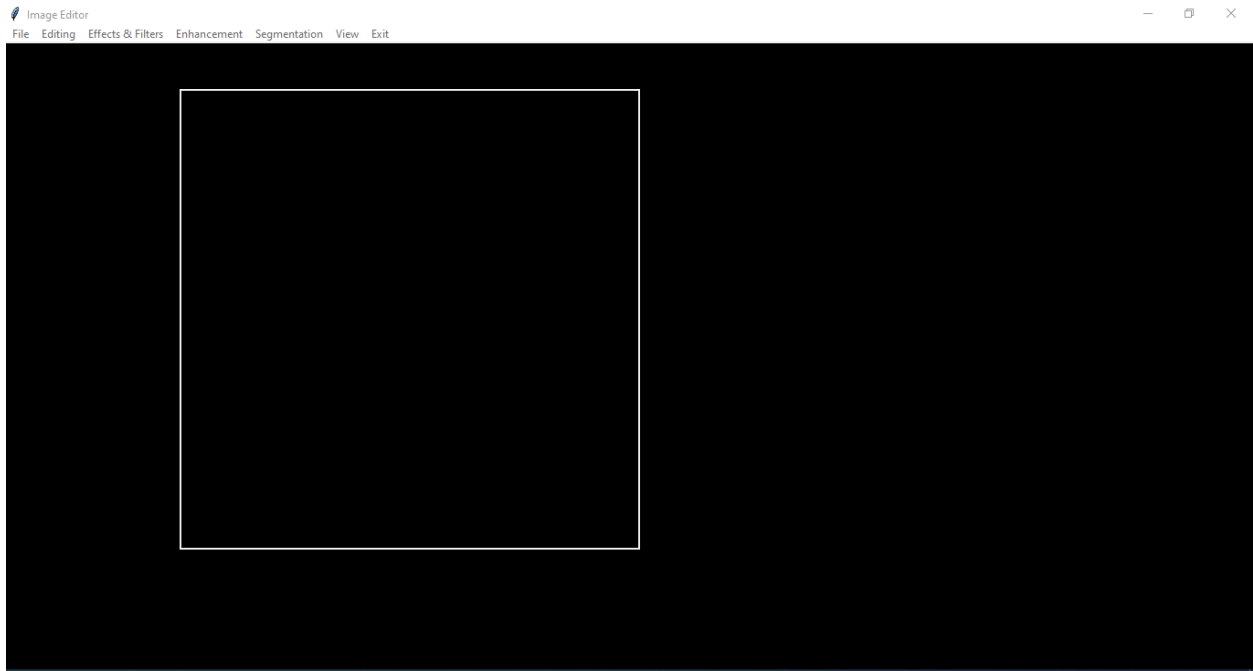


Fig 5.6

5.7 ENHANCEMENT FILTERS

On clicking the Enhancement button three image enhancement techniques will appear which are-

- Histogram Equalization
- Median Filtering
- Unsharp Mask Filtering

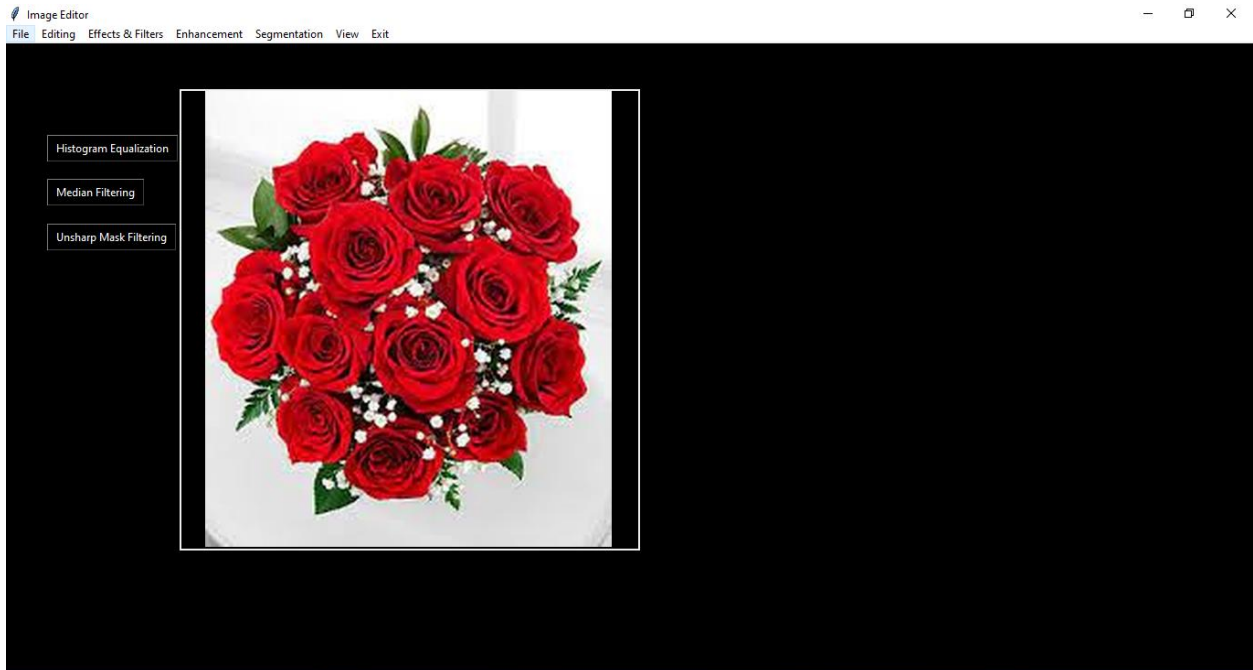


Fig 5.7

5.8 HISTOGRAM EQUALIZATION

On clicking Histogram Equalization a slider will appear which will be numbered 1 to 10, and using which we can apply this technique.

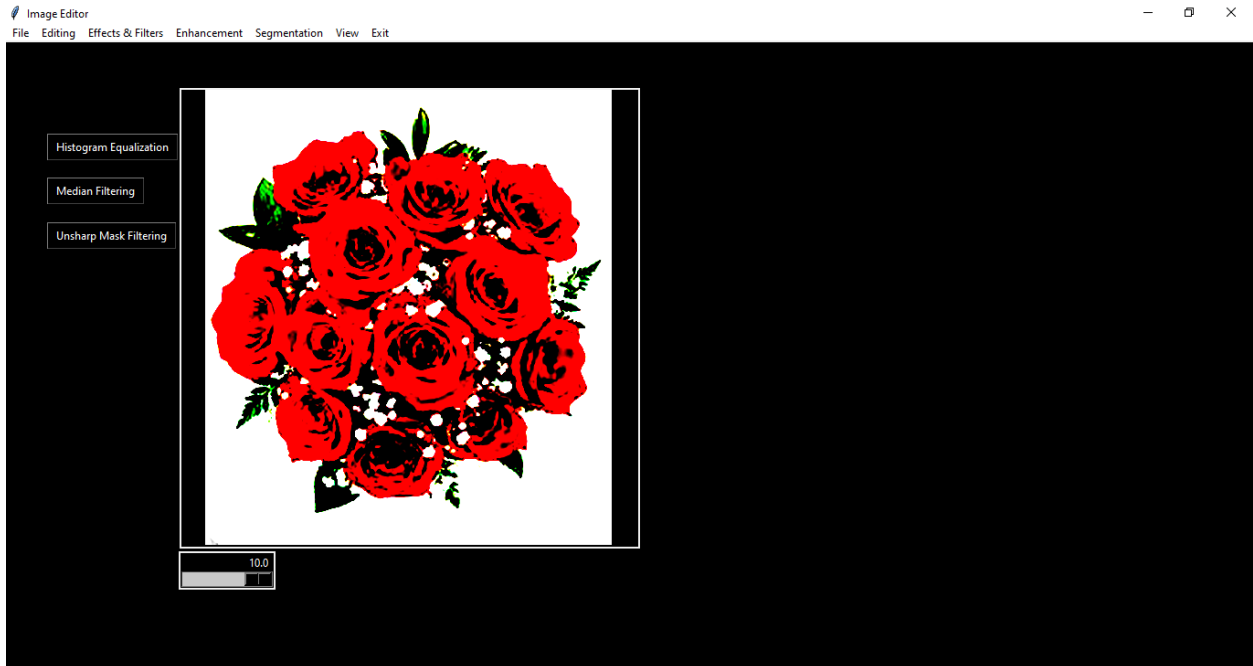


Fig 5.8

5.9 MEDIAN FILTERING

On clicking Median Filtering a slider will appear which will be numbered 1 to 10, and using which we can apply this technique.

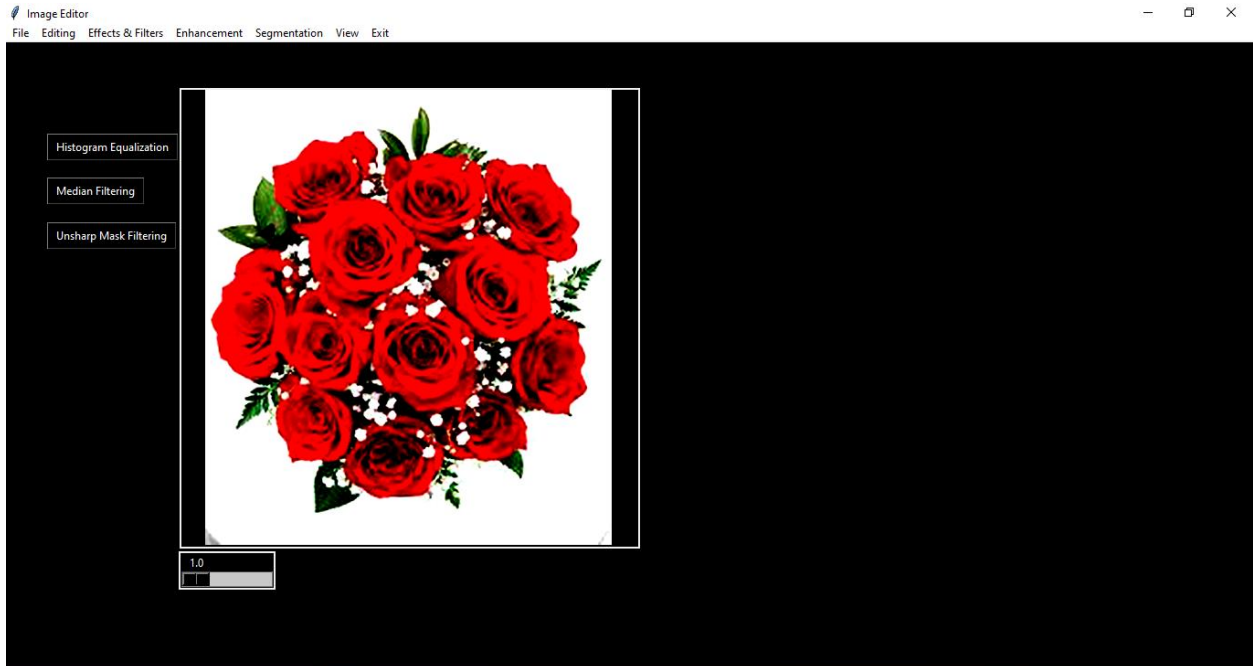


Fig 5.9

5.10 UNSHARP MASK FILTERING

On clicking Unsharp Mask Filtering we can apply this technique.

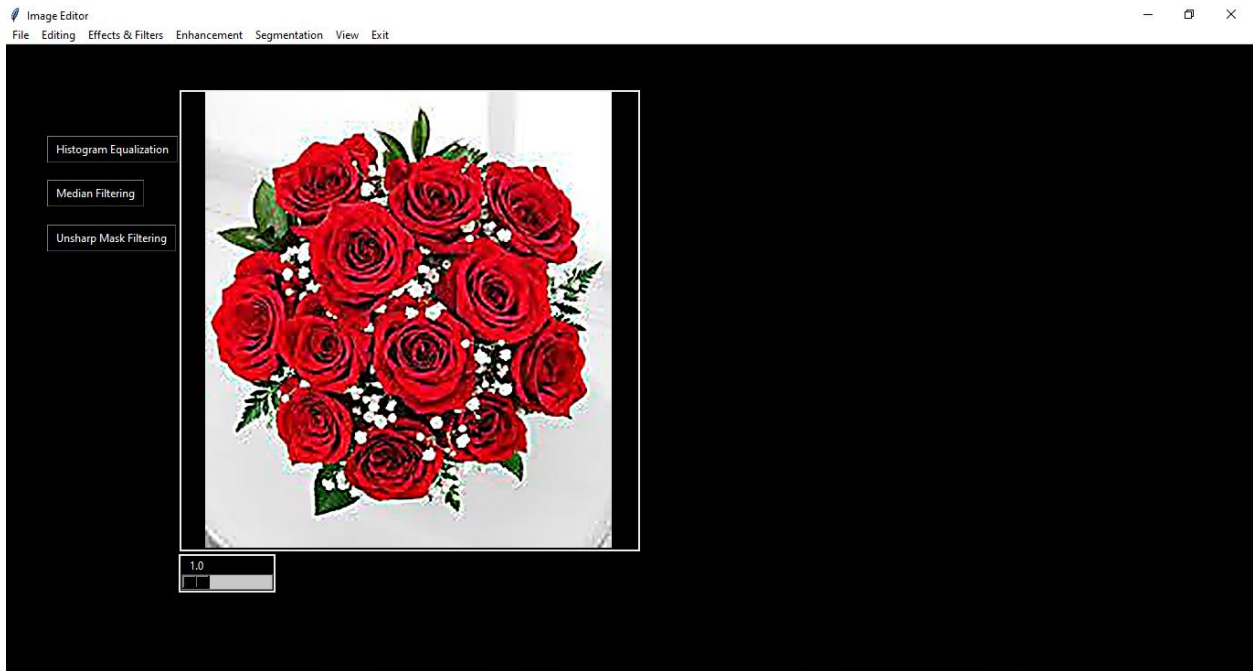


Fig 5.10

5.11 SEGMENTATION FILTER

On clicking Segmentation edge based segmentation technique button will appear.

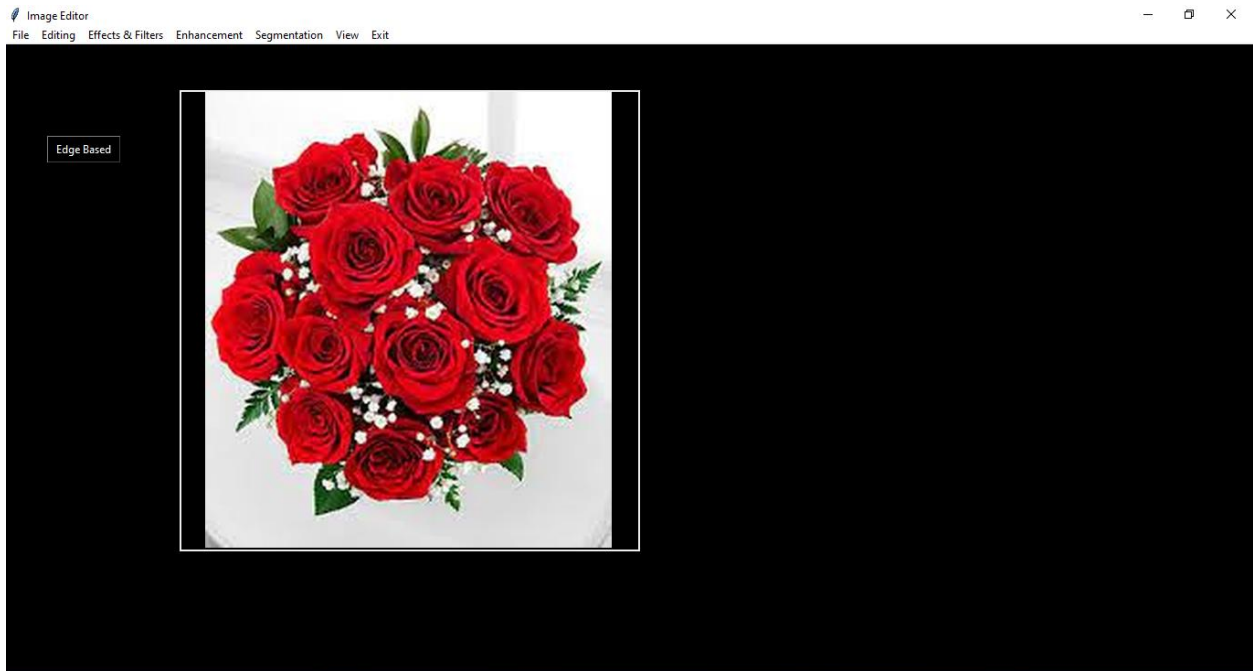


Fig 5.11

5.12 EDGE BASED SEGMENTATION

On clicking Edge based we can apply this technique.



Fig 5.12

CHAPTER – 6

CONCLUSIONS

- Image processing is exceptionally advantageous and cost effective way to deal with the different actual properties estimation e.g. size, color composition, shape and texture of particles.
- Different platforms are accessible for the image processing and a considerable lot of those are in open space domain. One of those is imageJ which is in open area domain and functions admirably.
- tKinter doesn't require any license for the users to use, hence it is a free software. It is compatible with many major platforms like Unix, Linux, Microsoft windows, Mac OS, etc. It has many built in features to create various widgets like buttons, check buttons, menu button, allows user to write a text on or around an image and many other options.
- Although Python has several options available for GUI designing for applications like PySimple GUI, Wax, PySide, Libavg, tKinter, PyQt, Kivy, etc out of these tKinter is the fastest and easiest way to create GUI applications.
- Creating a GUI using tKinter is an easy task. Tkinter is pre installed in Python 3, so the user don't have to install it separately.

REFERENCES

- [1] Schneider CA, Rasband WS, Eliceiri KW (2012). "NIH Image to ImageJ: 25 years of image analysis". *Nat Methods* 9 (7), 671–675.
- [2] Collins TJ (2007). "ImageJ for microscopy". *BioTechniques* 43 (1): 25–30.
- [3] Girish, V., & Vijayalakshmi, A. (2004). Affordable image analysis using NIH Image/ImageJ. *Indian journal of cancer*, 41(1), 47.
- [4] Eliceiri, K. W., & Rueden, C. (2005). Tools for visualizing multidimensional images from living specimens. *Photochemistry and photobiology*, 81(5), 1116-1122.
- [5] Barboriak, D. P., Padua, A. O., York, G. E., & MacFall, J. R. (2005). Creation of DICOM—aware applications using ImageJ. *Journal of digital imaging*, 18(2), 91-99.
- [6] Rajesh C. patil, A.S. Bhalchandra, "Brain tumorextraction from MRI images Using MAT Lab", *IJECSCSE*, ISSN: 2277-9477, Volume 2, issue1.