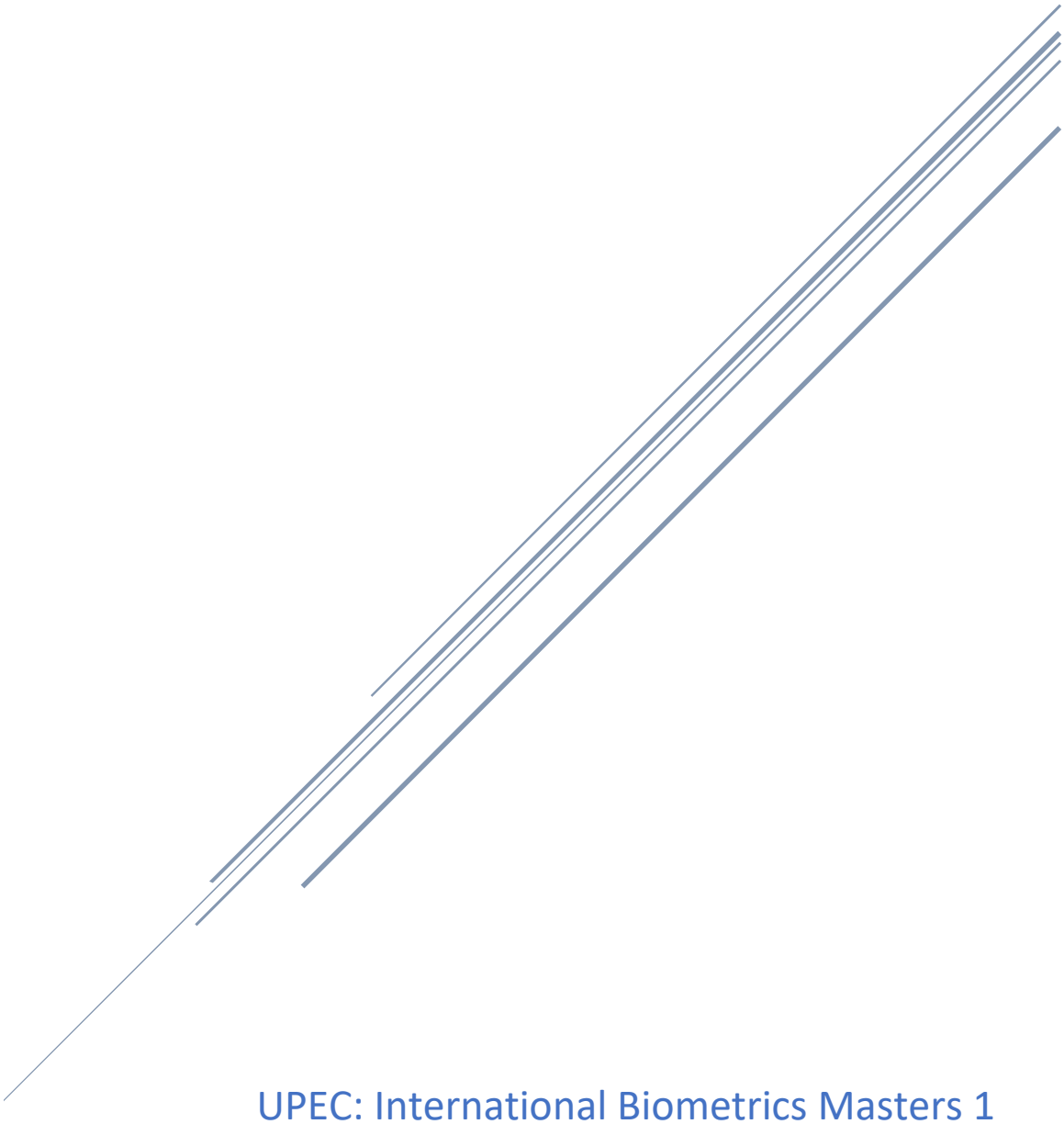
IMAGE PROCESSING

PROJECT



UPEC: International Biometrics Masters 1
Mounika Reddy PATIMIDHI

Contents

EXECUTIVE SUMMARY

The Image processing is an application that allows the users to upload the pictures to perform some operation on the image, in order to get an enlargement or to extract some useful information in it. This a type of signal processing in which input is an image and output may be image features associated with that image.

Users simply enter the image they want to analyses or you can allow to upload directly from the systems. The main functionality of this project is to provide the user to detect the attributes in the image is to perform some filters on the image.

The development of the project is done by one student currently at the UPEC as a part of the Master's in international biometrics(M1) program. The project start data is **17/12/2018** and the submission date is set to **4/02/2019**.This compasses approximately **58 hours** of work utilizing existing available equipment.

Total the project was done in 2 phases:

1. Image Processing Using with MATLAB

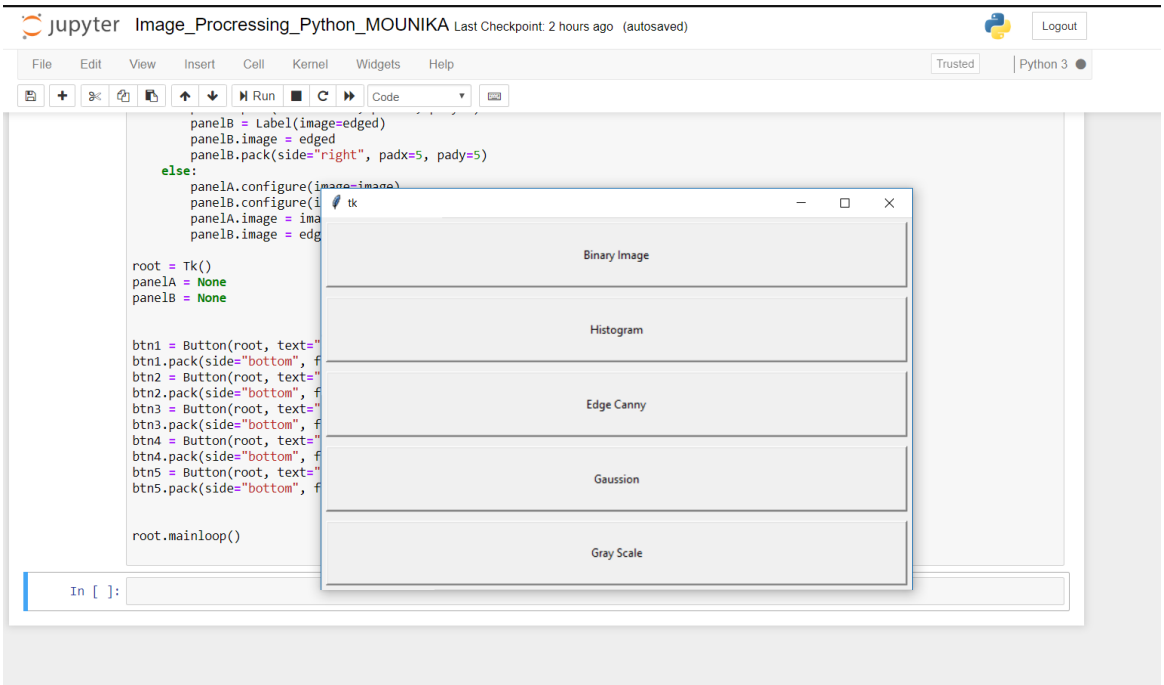
2. Image Processing using with Python

Introduction

The **Image processing** Procedures such as **Image** enhancement and restoration are used to process degraded or blurred images. In computer science, **digital image processing** is the use of computer algorithms to perform image processing on **digital images**. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during **processing**.

Raster images have a finite set of digital values, called picture elements or pixels. The digital image contains a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding antiquated values there present the brightness of a given color at any specific point.

Introduction to Graphical User Interface (GUI) MATLAB and Python:



Digital Image process

Image processing is among rapidly growing technology in core research area within engineering and computer science.

Image processing basic operation steps to following three steps:

- Importing the image via image acquisition tools;
- Analyzing and manipulating the image;
- Output in which result can be altered image or report that is based on image analysis

Constantly images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have various types of noise. In image improvement, the goal is to emphasize certain image features for following analysis or for image array, Examples include contrast and edge enhancement, noise filtering, Grayscale filtering, Histogram Plotting, Edges detection, colure changing, Gaussian filtering and Black and white Image improvement is useful in feature extraction, image analysis and an image display. The improvement process itself does not increase the inherent information content in the data. It simply highlights certain specified image components. Improvements in algorithms are generally collective and operation dependent.

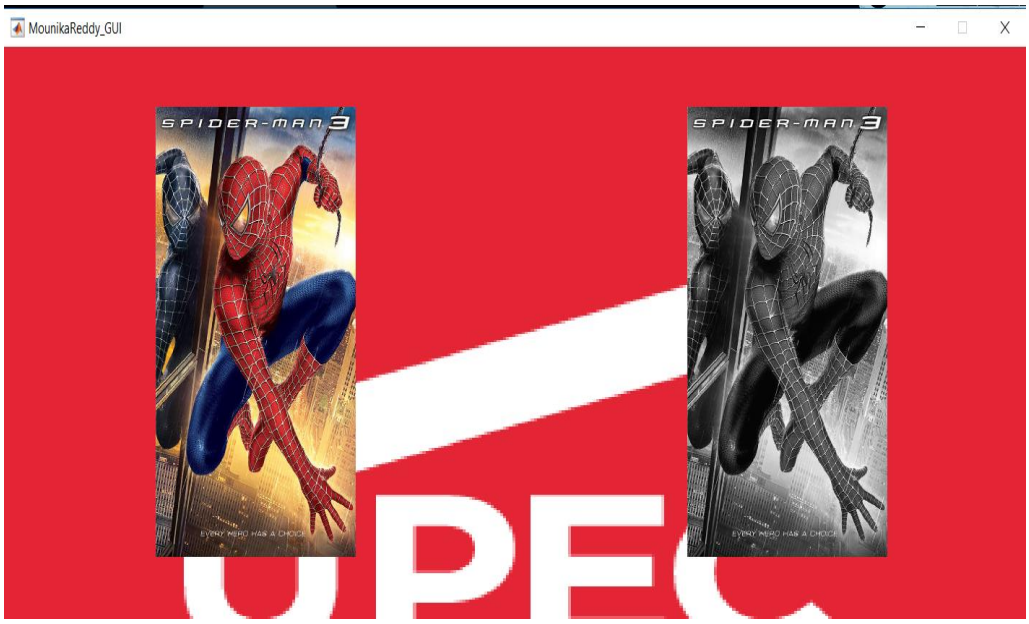
Gray scale

In digital photography, computer-generated imagery, and colorimetry, a grayscale or greyscale image is one in which the value of each pixel is a single sample representing only an amount of light, that is, it carries only intensity information.

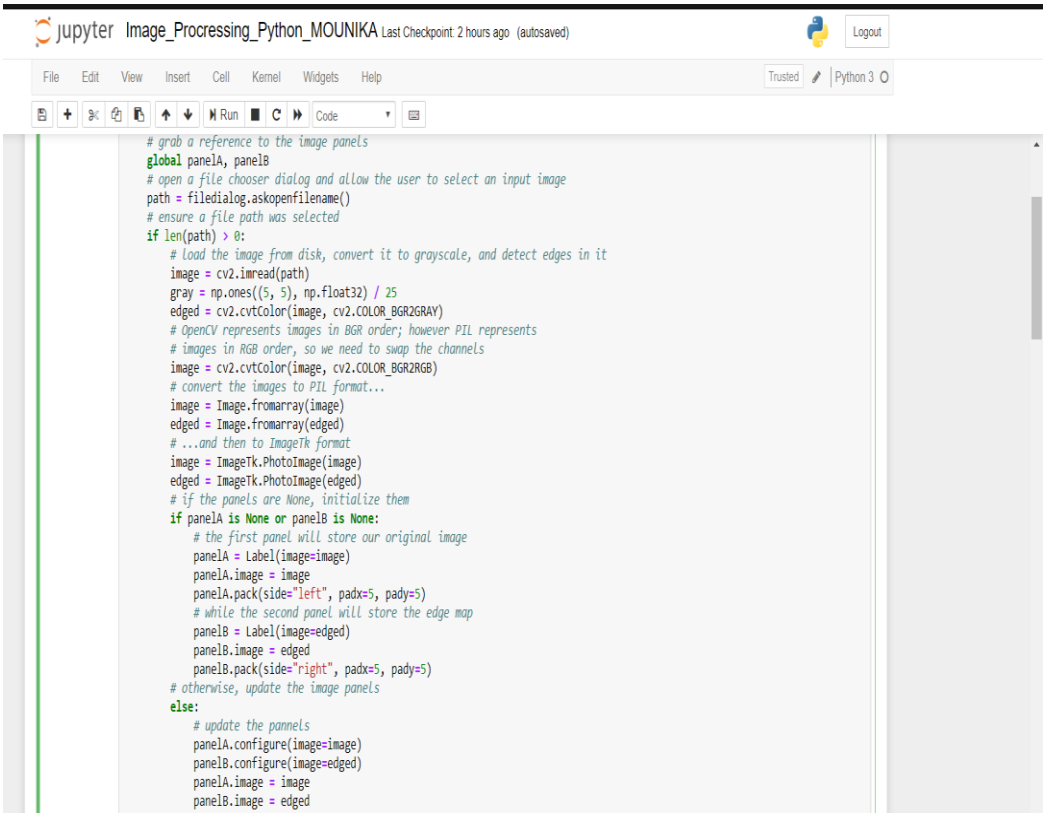
MATLAB CODE AND OUTPUT:

```
a = getappdata(0,'a');

a = rgb2gray(a);

BW2 = edge(a,'sobel');

axes(handles.axes2);

imshow(BW2);
```

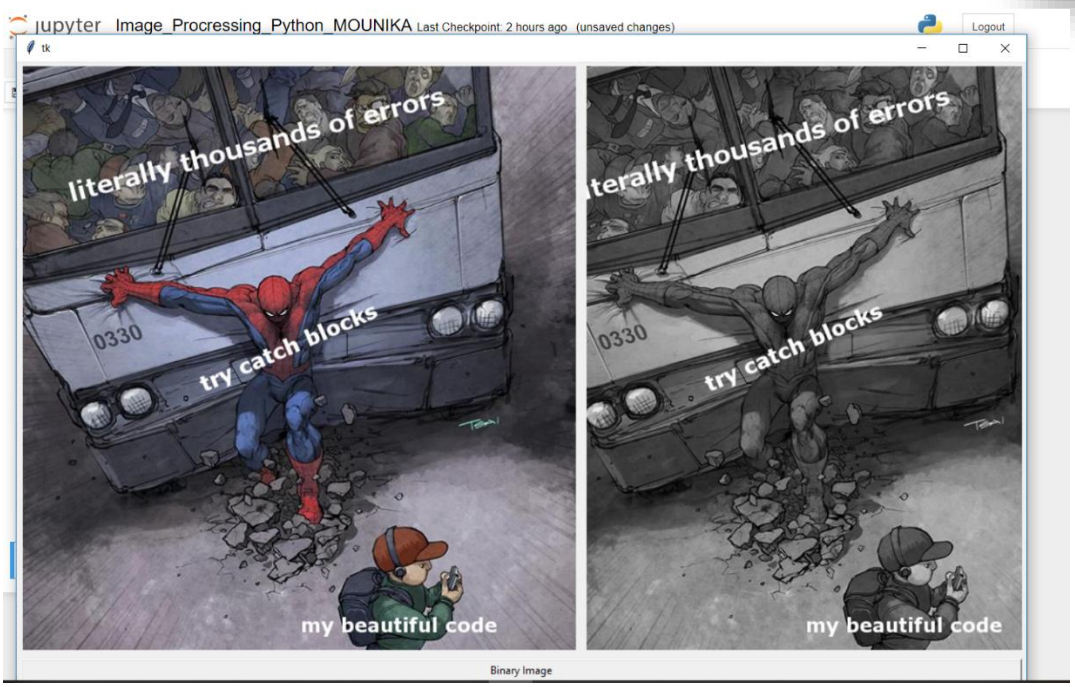
MATLAB INPUT AND OUTPUT SCREENSHOT



PYTHON CODE AND THE OUTPUT




```
# grab a reference to the image panels
global panelA, panelB
# open a file chooser dialog and allow the user to select an input image
path = filedialog.askopenfilename()
# ensure a file path was selected
if len(path) > 0:
    # load the image from disk, convert it to grayscale, and detect edges in it
    image = cv2.imread(path)
    gray = np.ones((5, 5), np.float32) / 25
    edged = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # OpenCV represents images in BGR order; however PIL represents
    # images in RGB order, so we need to swap the channels
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    # convert the images to PIL format...
    image = Image.fromarray(image)
    edged = Image.fromarray(edged)
    # ...and then to ImageTk format
    image = ImageTk.PhotoImage(image)
    edged = ImageTk.PhotoImage(edged)
    # if the panels are None, initialize them
    if panelA is None or panelB is None:
        # the first panel will store our original image
        panelA = Label(image=image)
        panelA.image = image
        panelA.pack(side="left", padx=5, pady=5)
        # while the second panel will store the edge map
        panelB = Label(image=edged)
        panelB.image = edged
        panelB.pack(side="right", padx=5, pady=5)
    # otherwise, update the image panels
    else:
        # update the pannels
        panelA.configure(image=image)
        panelB.configure(image=edged)
        panelA.image = image
        panelB.image = edged
```

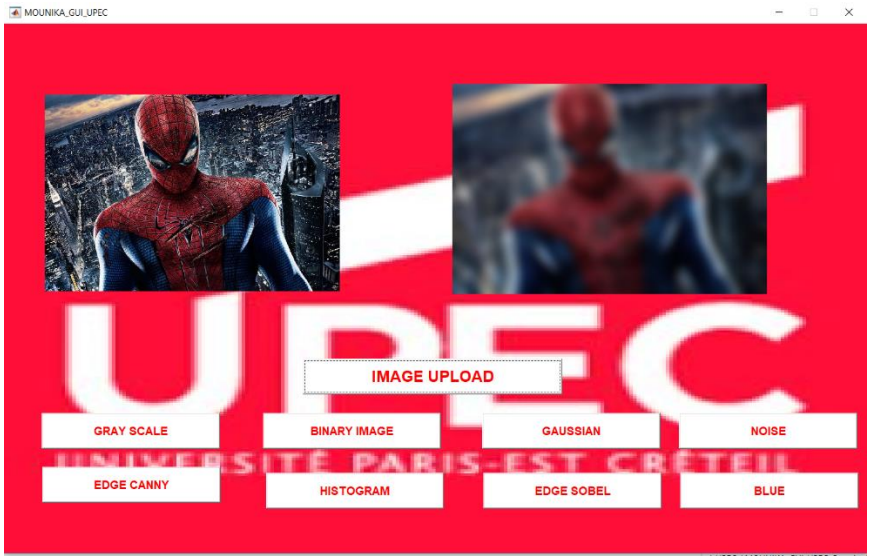


GAUSSIAN FILTERS

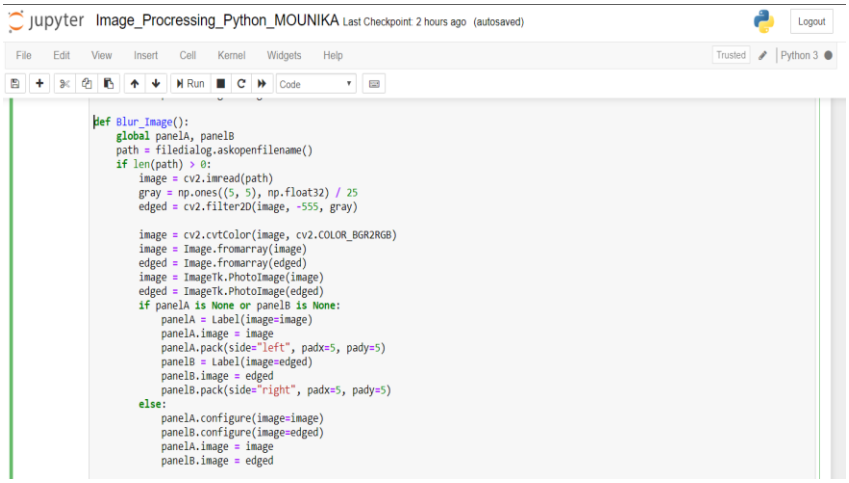
It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components a Gaussian blur is thus a low pass filter.

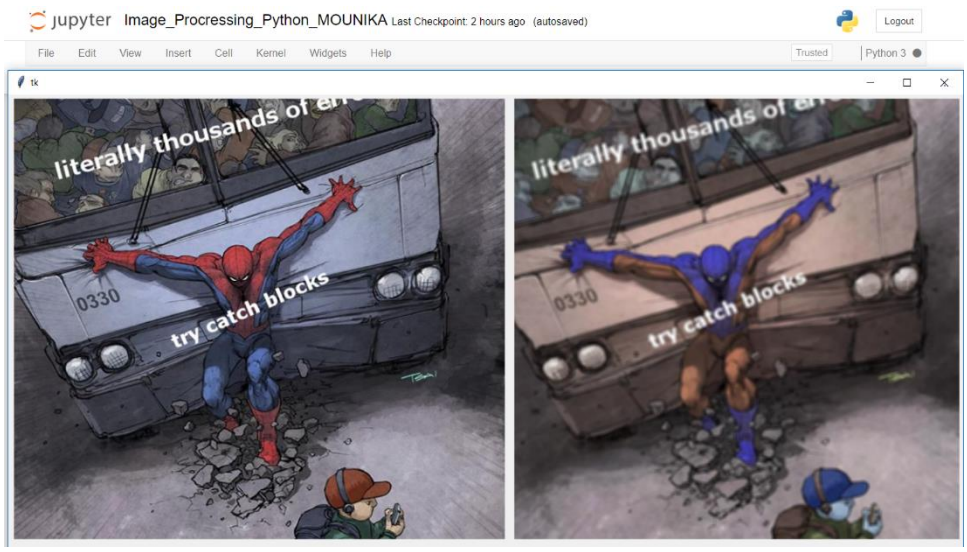
Output of Gaussion in Matlab

```
a = getappdata(0,'a');

d = imgaussfilt(a,20);

setappdata(0,'filename',d);

axes(handles.axes2);

imshow(d);
```



Output of Gaussian in Python



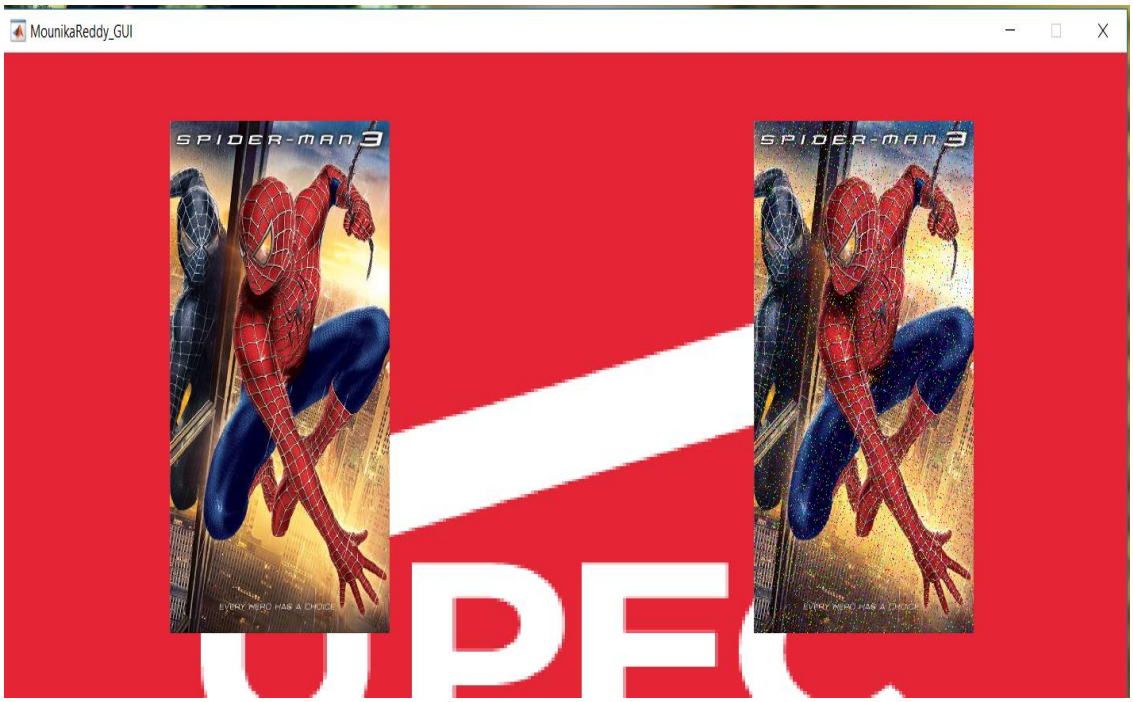


NOISE FILTERS

Noise Filtering is used to filter the needless data from an image. It is also used to extract different types of noises from the images. Mostly these characters are interactive. Different filters like low pass, high pass, mean, median etc.,

Noise Filter MATLAB Code and output

```
a = getappdata(0,'a');

e = imnoise(a,'salt & pepper');

setappdata(0,'filename',e);

axes(handles.axes2);

imshow(e);
```

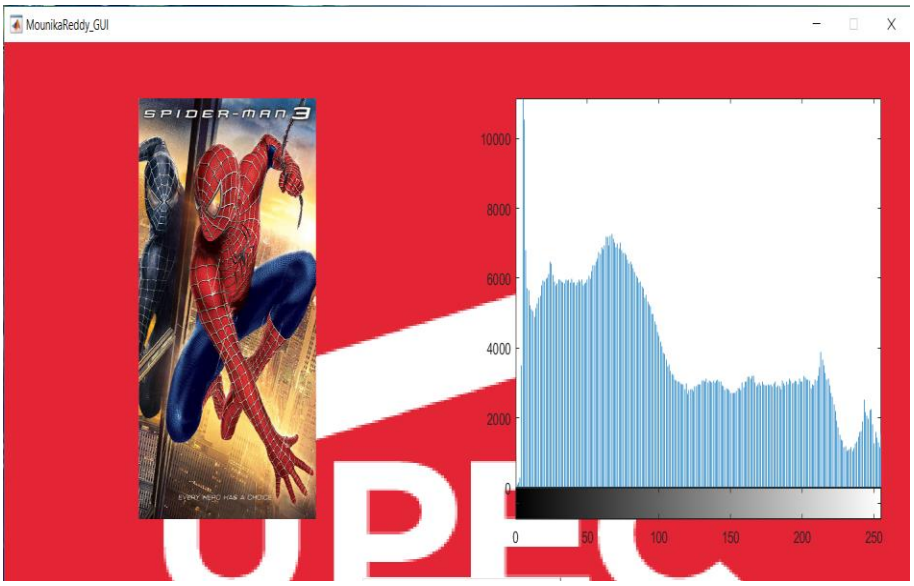


HISTOGRAM

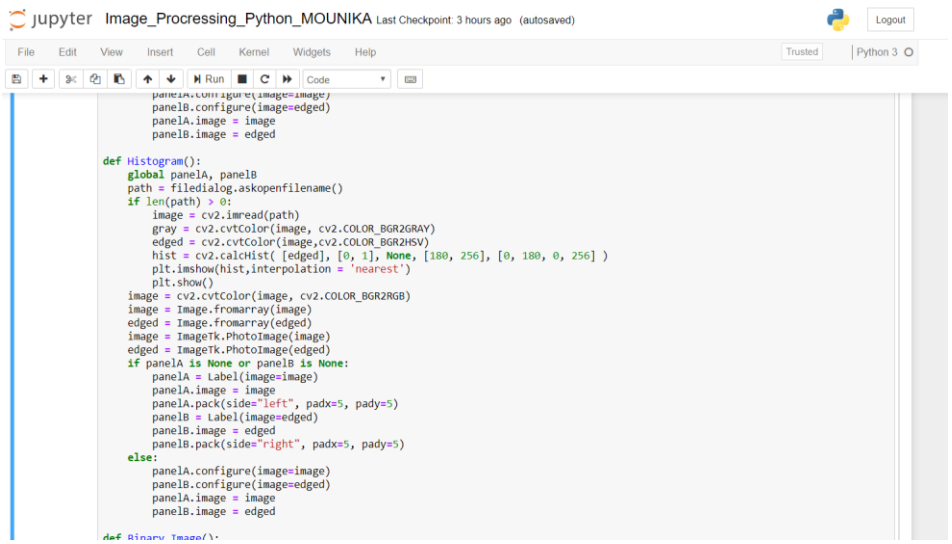
A histogram is a definitive representation of the distribution of numerical data. It is an estimate of the probability distribution of a continuous variable and was first introduced by Karl Pearson. It differs from a bar graph, in the sense that a bar graph relates two variables, but a histogram relates only one.

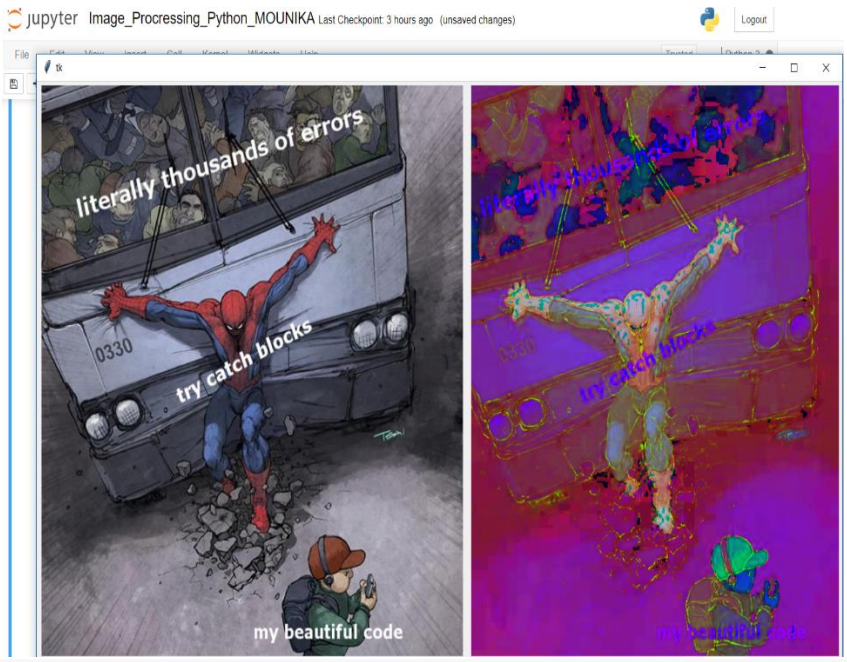
Histogram MATLAB code and Output

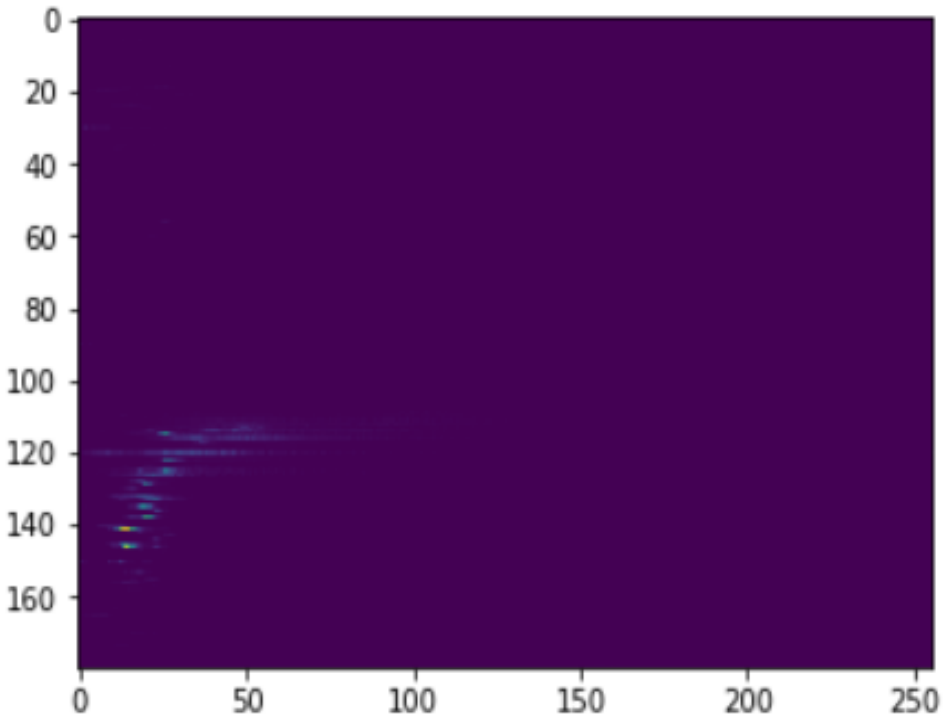
```
a= getappdata(0,'a');
input = a;
input = rgb2gray(input);
axes(handles.axes2);
setappdata(0,'filename',input);
imhist(input);
```



Output of Histogram in Python




jupyter Image_Procressing_Python_MOUNIKA Last Checkpoint: 3 hours ago (unsaved changes)
Logout
literally thousands of errors
0330
try catch blocks
my beautiful code

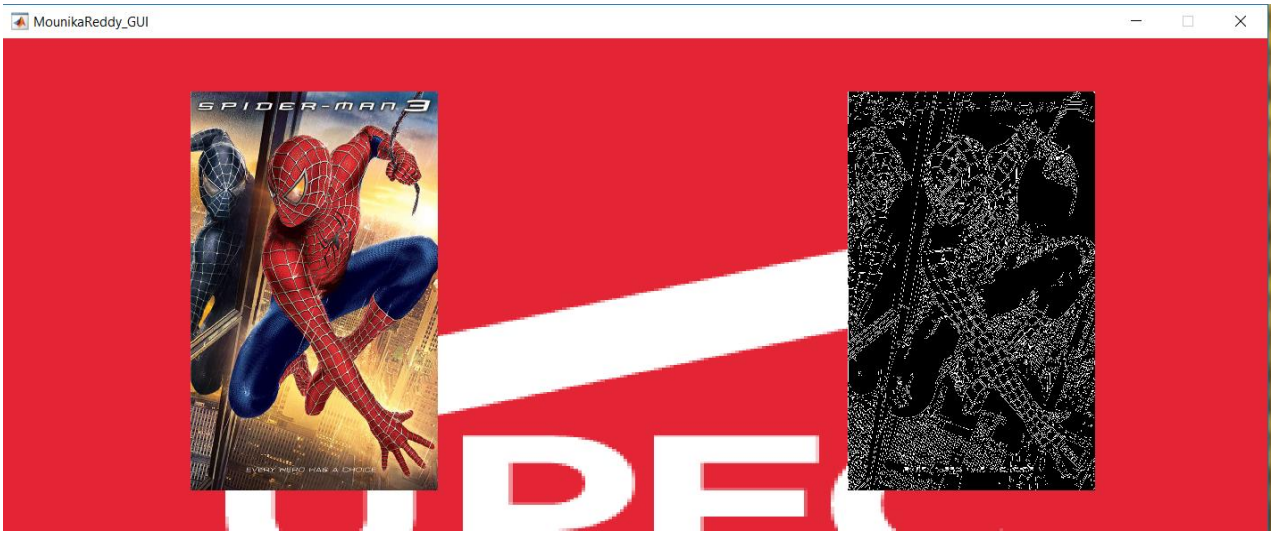

0
20
40
60
80
100
120
140
160
0
50
100
150
200
250

EDGE CANNY

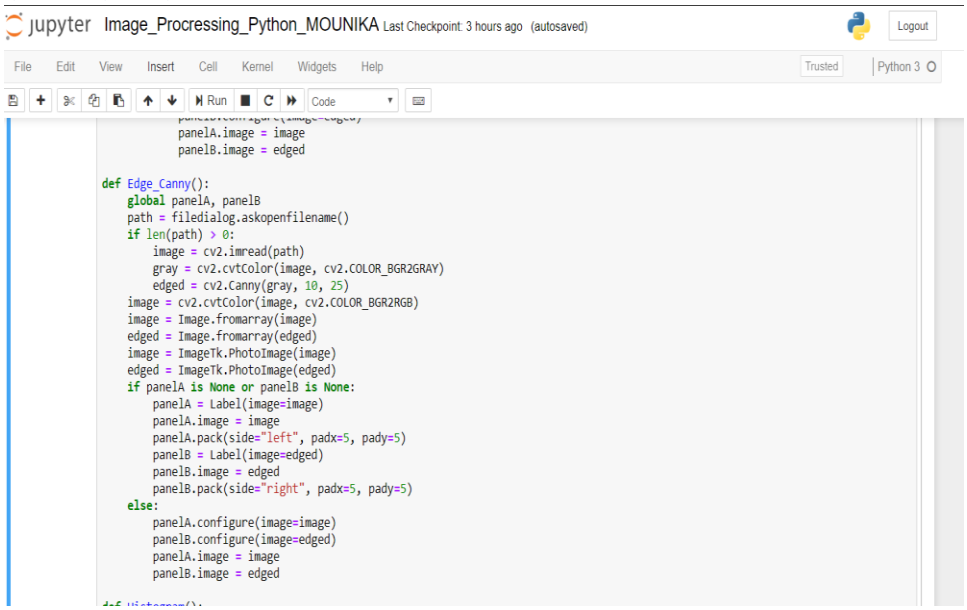
The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works

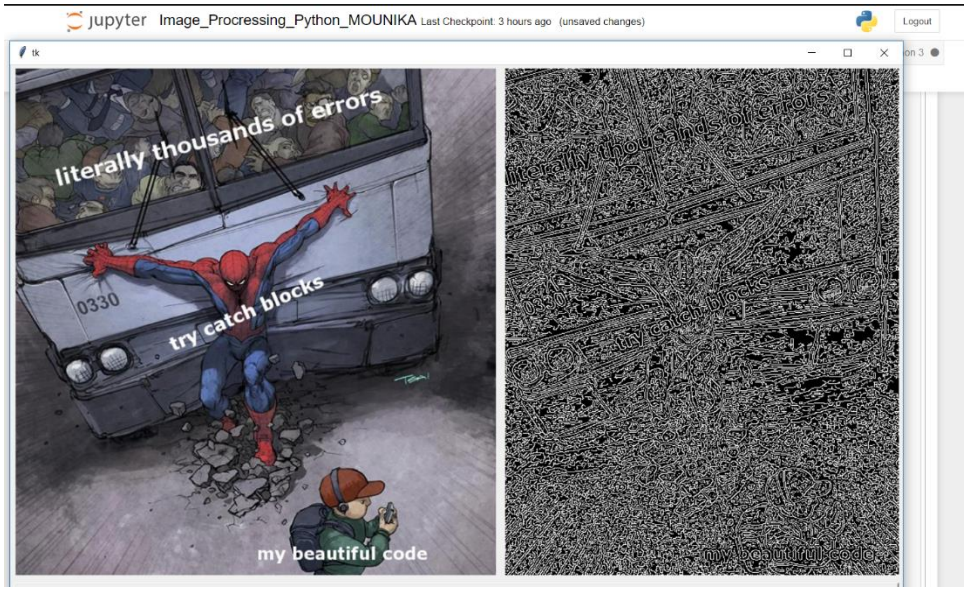
EDGE CANNY MATLAB Code and Output

```
a = getappdata(0,'a');

a = rgb2gray(a);

BW2 = edge(a,'canny');

axes(handles.axes2);

imshow(BW2);
```



Output of Edge Canny in Python





Binary Image

Each pixel is stored as a single bit (0 or 1)

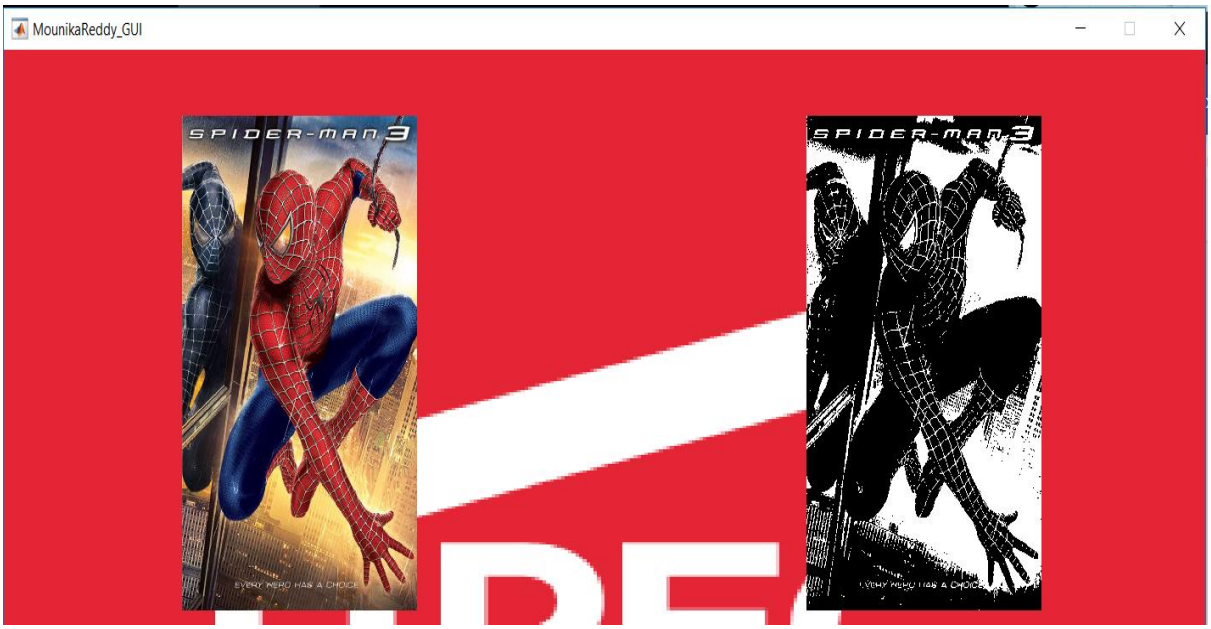
GRAT LEVEL IMAGE: Each pixel has a gray value between 0 and 255.

- Each pixel is represented by single bytes for example a dark pixel might have a value of 10 and bright one might be 230
- A640*480 gray scale image requires 300kB of storage (640 x 480 = 307,200)

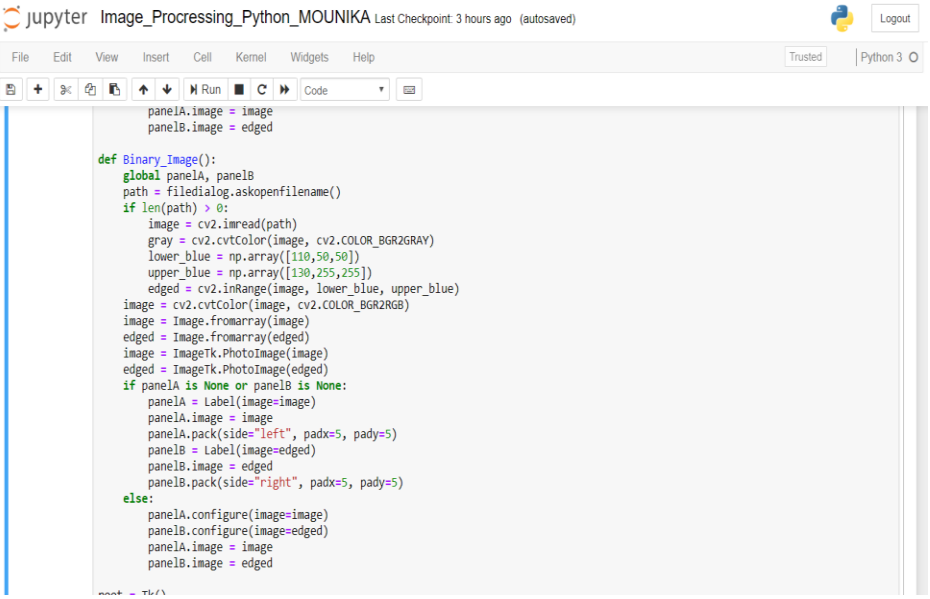
I = im2bw(RGB) converts the TrueColor image RGB to the grayscale intensity image. The im2bw function converts RGB images to black and white by eliminating the hue and saturation information while retaining the luminance.

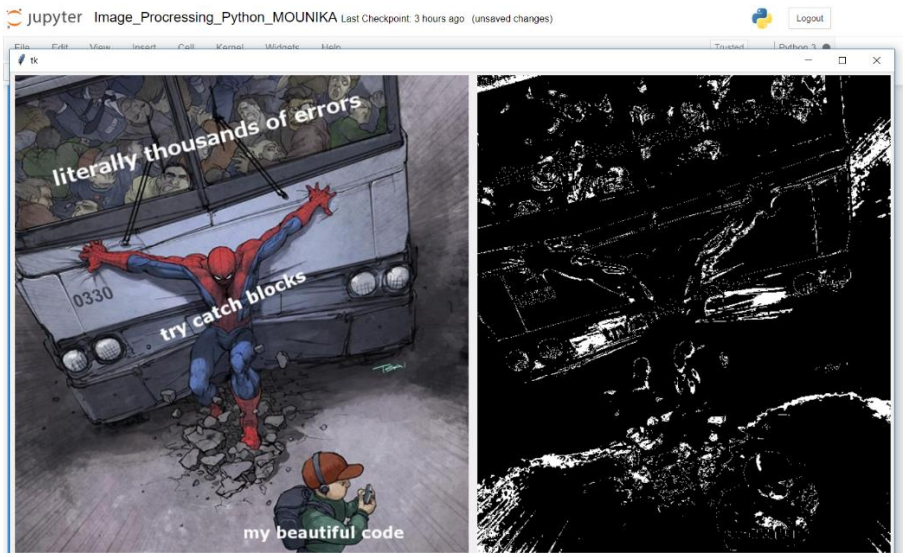
Binary Image MATLAB code and Output

```
a = getappdata(0,'a');

c= im2bw(a);

setappdata(0,'filename',c);

axes(handles.axes2);

imshow(c)
```



Output of Binary Image in Python





Color Image

In a color 24-bit image each pixel is represented by three bytes usually representation RGB.

This format support **256x256x256** possible combined color or total of 16,777m216 possible colors.
However, such flexibility does result in a storage penalty: a 640x480 24-bit color image would require 921.6 KB of storage without any compression. but many systems can make use of 8 bits of color information in producing a screen image , such image files use the concept of a lookup table to store color information basically, the image stores not color but instead just a set of bytes

each of eliches actually an index in to table with 3-bytes values that specify the color for a pixel with that lookup table index

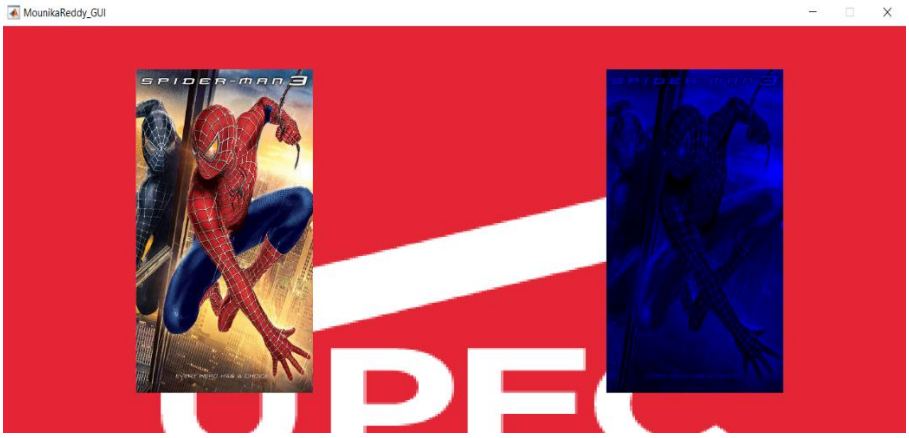
NOTE THAT THE GREAT SAVING IN SPACE FPR 2=BIT IMAGE OVER 24-BIT ONES:

a**640x480x 8**-bit color image only requires 300kb of storage compressed to 921.6kb for a color image (again without any compression applied).

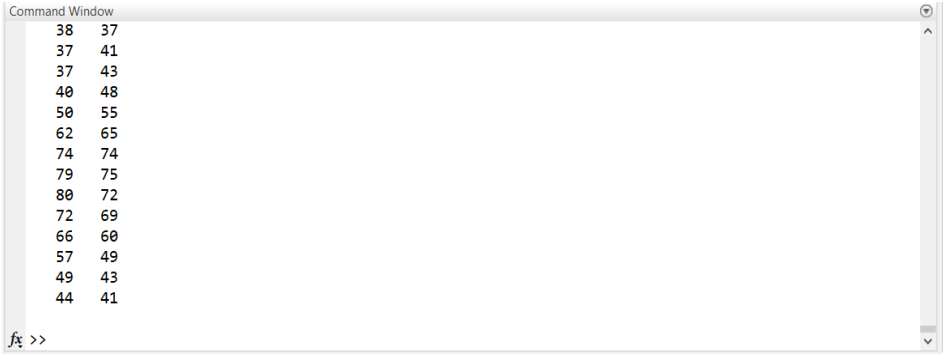
A color image can encode as a combination of three images (channels). if each channel is encoded on 8 bits the RGB color image which combines three images will be called 24bit color image.

Color Image MATLAB code and output

```
a =getappdata(0,'a');

blue = a;

blue(:,:,1:2)=0

setappdata(0,'filename',blue);

axes(handles.axes2);

imshow(blue);
```



Below image represent the bytes of the image to turn the color in blue.



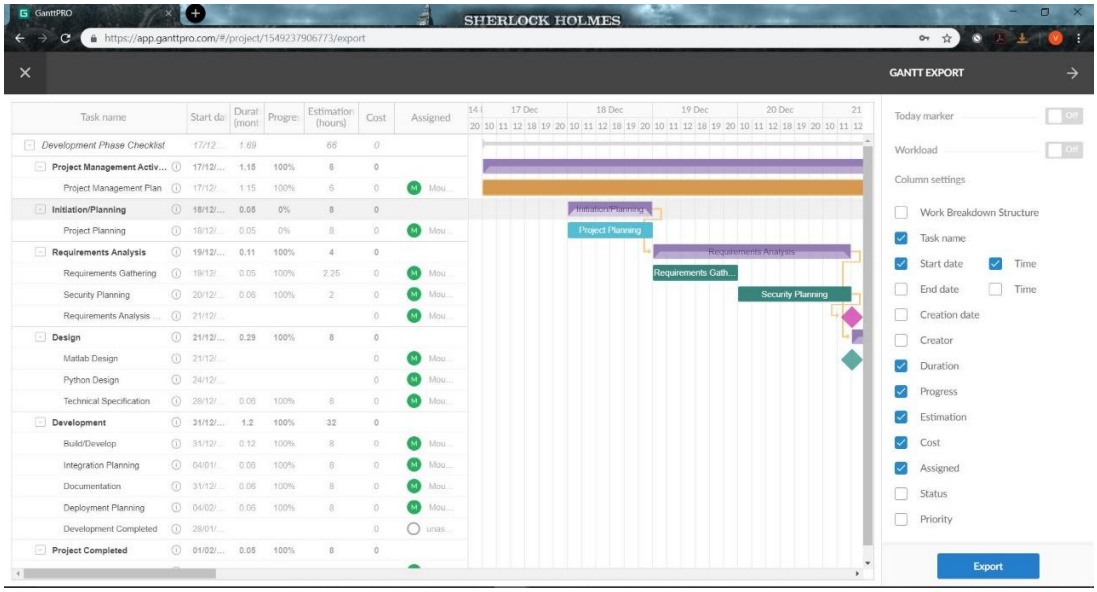
GANTT CHART

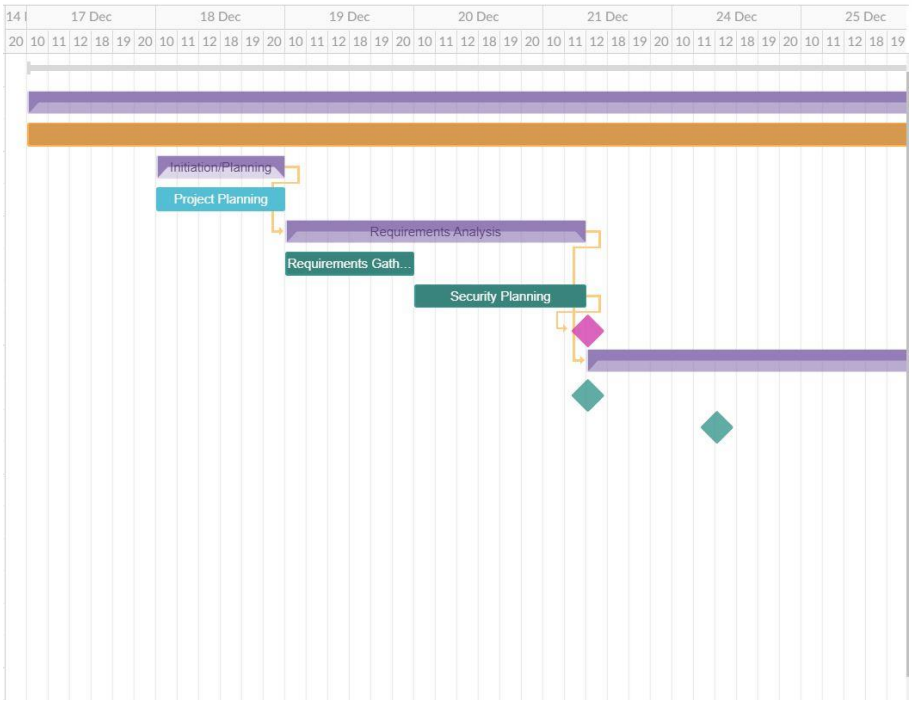
Gantt chart is a type of bar cart that illustrates a project schedule. Chart lists shows tasks to be performed on the vertical axis and time interval on the horizontal axis. Along with that we can keep the milestones of our project. In the project Image Processing I have 4 milestones in my project in various stages.

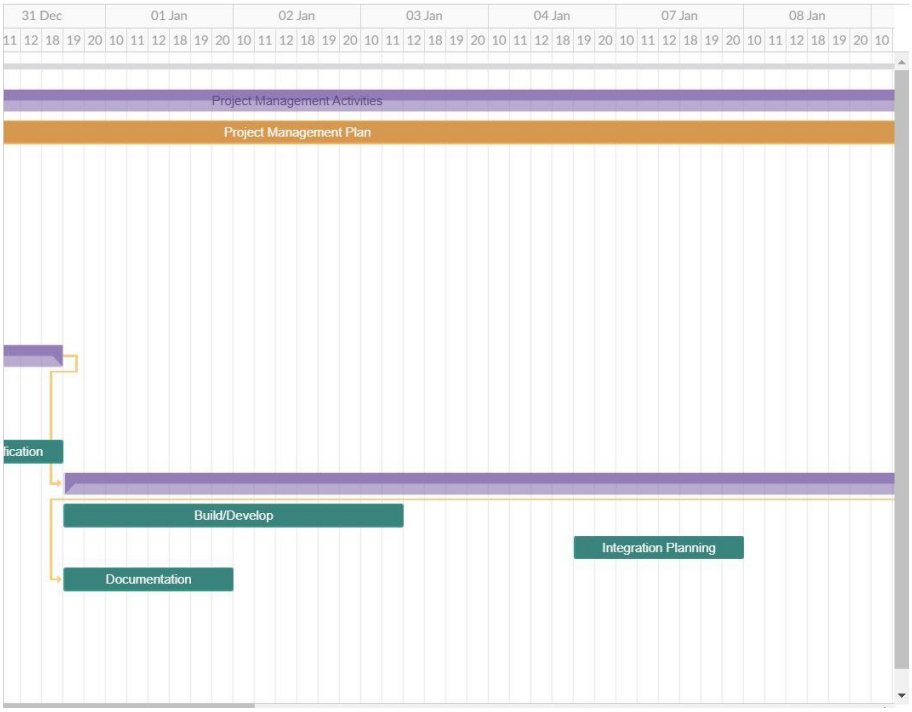
I have used Ganttpro for my project.

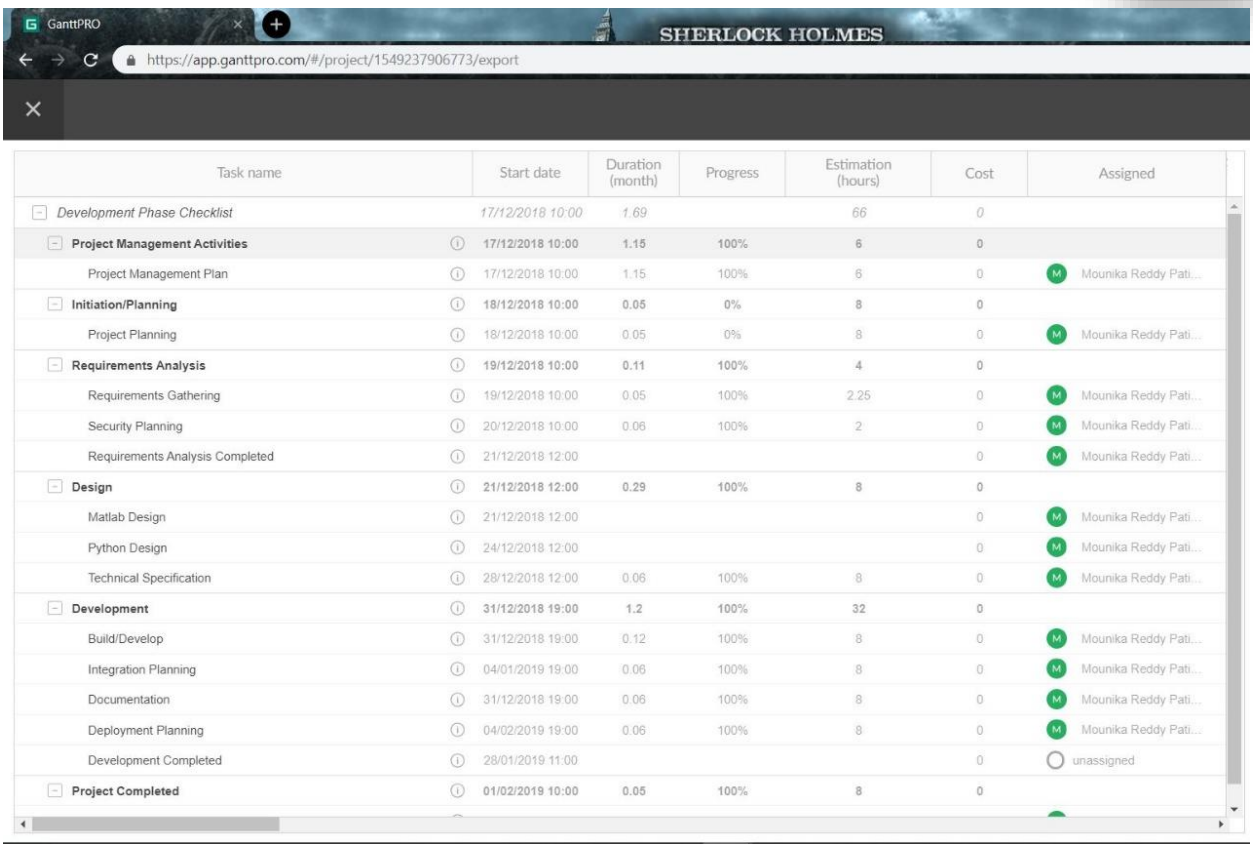
Tasks name that I have taken in this project:

1. Project Management Activities
2. Initiation/Planning
3. Requirements Analysis.
4. Design
5. Development
6. Project Deployment (Done)




17 Dec
18 Dec
19 Dec
20 Dec
21 Dec
24 Dec
25 Dec
Initiation/Planning
Project Planning
Requirements Analysis
Requirements Gath...
Security Planning


31 Dec
01 Jan
02 Jan
03 Jan
04 Jan
07 Jan
08 Jan
Project Management Activities
Project Management Plan
Build/Develop
Integration Planning
Documentation



Task name	Start date	Duration (month)	Progress	Estimation (hours)	Cost	Assigned
Development Phase Checklist	*17/12/2018 10:00*	*1.69*		*66*	*0*	
Project Management Activities	**17/12/2018 10:00**	**1.15**	**100%**	**6**	**0**	
Project Management Plan	17/12/2018 10:00	1.15	100%	6	0	Mounika Reddy Pati...
Initiation/Planning	**18/12/2018 10:00**	**0.05**	**0%**	**8**	**0**	
Project Planning	18/12/2018 10:00	0.05	0%	8	0	Mounika Reddy Pati...
Requirements Analysis	**19/12/2018 10:00**	**0.11**	**100%**	**4**	**0**	
Requirements Gathering	19/12/2018 10:00	0.05	100%	2.25	0	Mounika Reddy Pati...
Security Planning	20/12/2018 10:00	0.06	100%	2	0	Mounika Reddy Pati...
Requirements Analysis Completed	21/12/2018 12:00				0	Mounika Reddy Pati...
Design	**21/12/2018 12:00**	**0.29**	**100%**	**8**	**0**	
Matlab Design	21/12/2018 12:00				0	Mounika Reddy Pati...
Python Design	24/12/2018 12:00				0	Mounika Reddy Pati...
Technical Specification	28/12/2018 12:00	0.06	100%	8	0	Mounika Reddy Pati...
Development	**31/12/2018 19:00**	**1.2**	**100%**	**32**	**0**	
Build/Develop	31/12/2018 19:00	0.12	100%	8	0	Mounika Reddy Pati...
Integration Planning	04/01/2019 19:00	0.06	100%	8	0	Mounika Reddy Pati...
Documentation	31/12/2018 19:00	0.06	100%	8	0	Mounika Reddy Pati...
Deployment Planning	04/02/2019 19:00	0.06	100%	8	0	Mounika Reddy Pati...
Development Completed	28/01/2019 11:00				0	unassigned
Project Completed	**01/02/2019 10:00**	**0.05**	**100%**	**8**	**0**	

Conclusion:

To conclude, Project Image Processing works like a analysis which can access all the images. It overcomes the many limitations.

Easy Implementation, Generate the Processed output

Scope for future development

This Project has a very vast scope in future. The project can be implemented on different cases in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed GUI function and backend code the users is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

- Will involve scanning the heavens for other intelligent life out in space.
- Also new intelligent, digital species created entirely by research scientists in various nations.
- Diagnosing Medical Conditions, Performing increasing power and sophistication of modern computing.
- Reprograming defects in human DNA


Thank You
Merci
Obrigado
Asante
Arigato
Multumesc
Spasibo
Juspaxar
Grazie
Dank Je
Matondo
Maake
Welalin
Salamat
Kia Ora
Kiitos
Vinaka
Mamana
Terima Kasih
Mochchakkeram
Cholrane
theleamingclinic.com.au