



# **MAJOR PROJECT: IMAGE PROCESSING INTERFACE**

**SEM: 8<sup>TH</sup>**

**BRANCH:ECE**

**BATCH:2017-2021**

**GROUP MEMBERS :SAJAL CHANDRA (04314802817)**

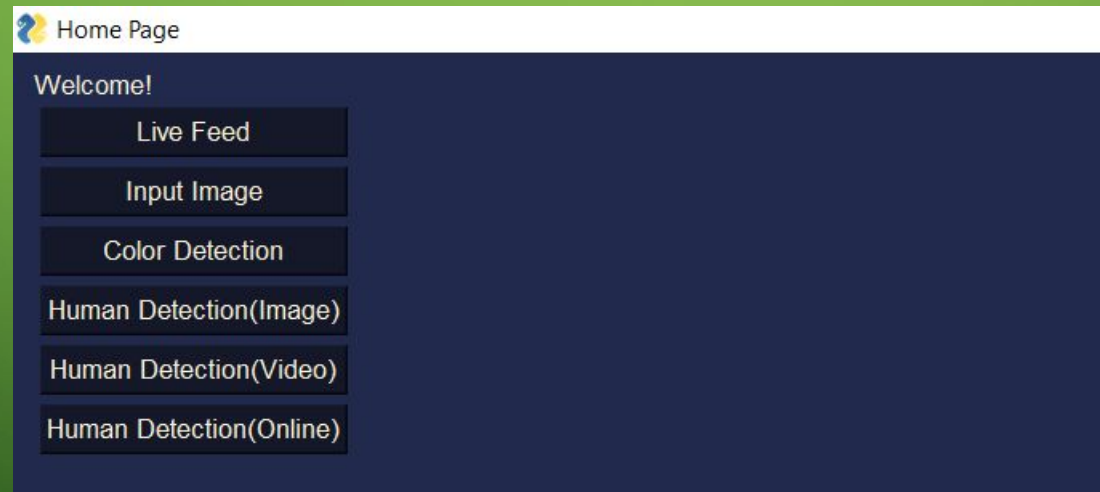
**OJAS (35914802817)**

**ROHIT KUMAR (04214802817)**

**MENTOR: DR. R.K CHOUDHARY**

# INTRODUCTION

- **AIM:** We aim to create a interface that allows one to select an image processing function (for eg: cartoonify image , color detection etc.) , and then demonstration of the function automatically displays.



# DIFFERENT LIBRARIES USED

- 1.NUMPY : It is used for working with arrays . It also helps for scientific computing in python.
- 2.PANDAS: It is an open source python package that is most widely used for data analysis and ML tasks . It is built on top of another package named Numpy.
- 3.CV2: OpenCV-Python is a library of python bindings designed to solve computer vision related tasks . It provides us with functions like `cv2.imread( )` which loads an image directly from the specified file.
- 4.PILLOW: It is a Python Imaging Library (PIL), which adds support for opening, manipulating and saving images.
- 5.PySimpleGUI: It is a Python Package that enables Python programmers of all levels to create GUIs. We specify our GUI window using a “layout” which contains widgets or elements.

# OPENCV VS MATLAB

OPENCV	MATLAB
Faster in execution	Convenient in developing and data representation
Harder to learn	Comparatively easy to learn
Useful for rapid prototyping	Its program are not portable
It is open source	It is not open source

# COLOR DETECTION

- Colors are made up of 3 primary colors; red, green, and blue. In computers, we define each color value within a range of 0 to 255. So in how many ways we can define a color? The answer is  $256 * 256 * 256 = 16,581,375$ .
- We will be using a dataset that contains RGB values with their corresponding names.

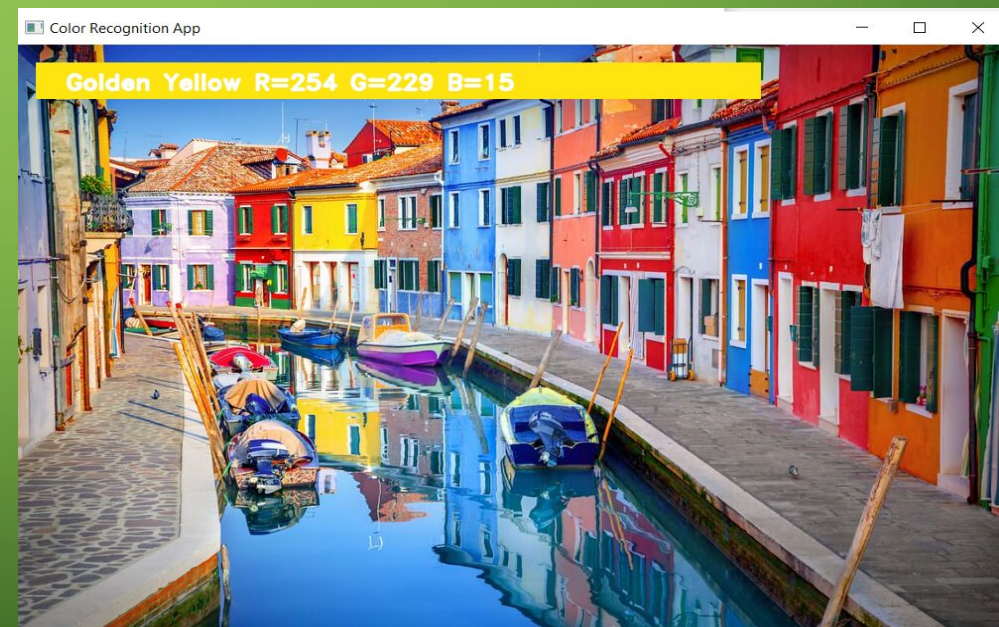
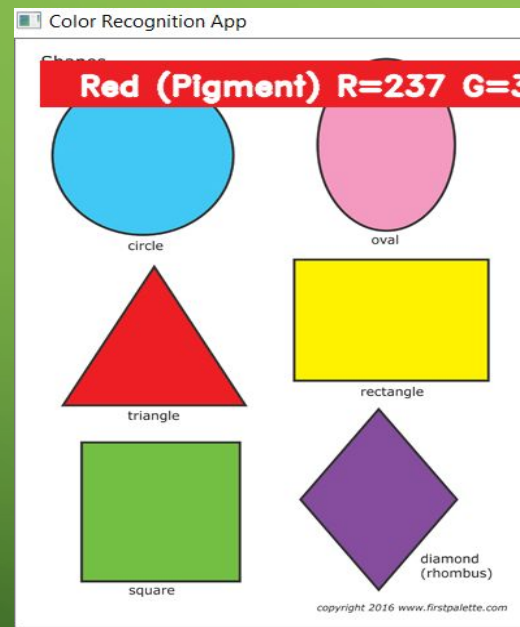
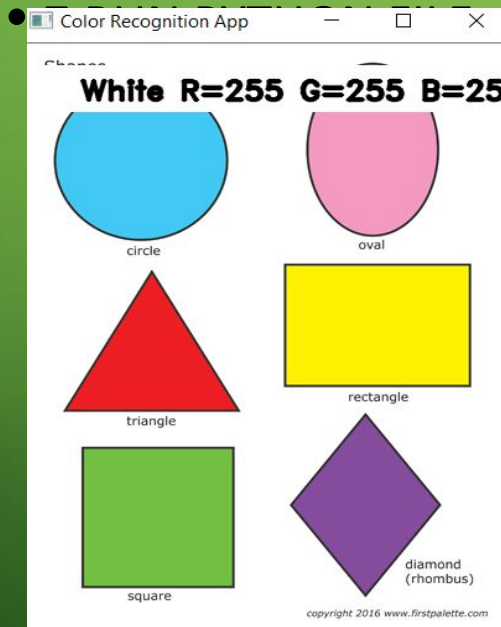
## STEPS

1.TAKING IMAGE FROM USER

2.READ CSV FILE



- 3. SET A MOUSE CALLBACK EVENT ON A WINDOW
- 4.CREATE DRAW FUNCTION
- 5.CALCULATE DISTANCE TO GET COLOR NAME
- 6.DISPLAY IMAGE ON WINDOW



# REAL TIME HUMAN DETECTION

- 1.Import the libraries
- 2. Model building for detecting humans

(i) We have used HOGDescriptor with SVM

(ii) `cv2.HOGDescriptor_getDefaultPeopleDetector()` calls the pre-trained model for Human detection of OpenCV and then we will feed our support vector machine with it.

- 3. Detect() Method

Video: A video combines a sequence of images to form a moving picture. We call these images as Frame. So in general we will detect the person in the frame. And show it one after another that it looks like a video.

That is exactly what our Detect() method will do. It will take a frame to detect a person in it. Make a box around a person , show the frame and return the frame with person bounded by a green box.

- ***detectMultiScale()***

*It returns 2-tuple.*

*List containing Coordinates of bounding Box of person.*

*Coordinates are in form X, Y, W, H.*

*Where x,y are starting coordinates of box and w, h are width and height of box respectively.*

*Confidence Value that it is a person.*

*4.HumanDetector() method:There are two ways of getting Video.*

*(i)Web Camera*

*(ii)Path of file stored*



5. DetectByCamera() method :cv2.VideoCapture(0) passing 0 in this function means we want to record from a webcam. video.read() read frame by frame. It returns a check which is True if this was able to read a frame otherwise False.

6. DetectByPathVideo() method

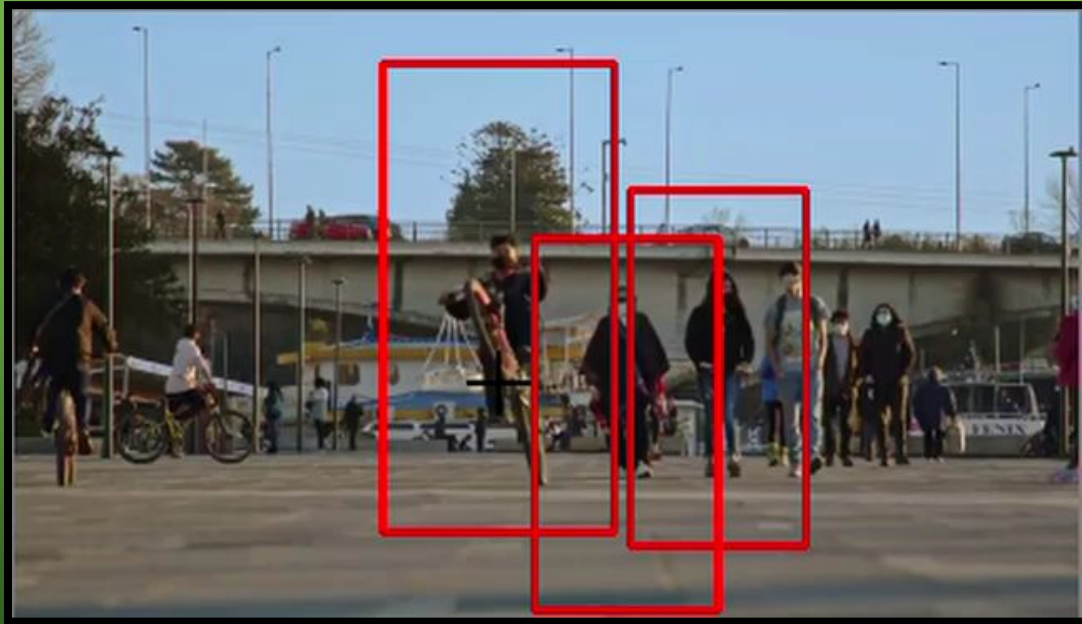
7. DetectByPathimage() method: This method is used if a person needs to be detected from an image.

8. Argparse() method

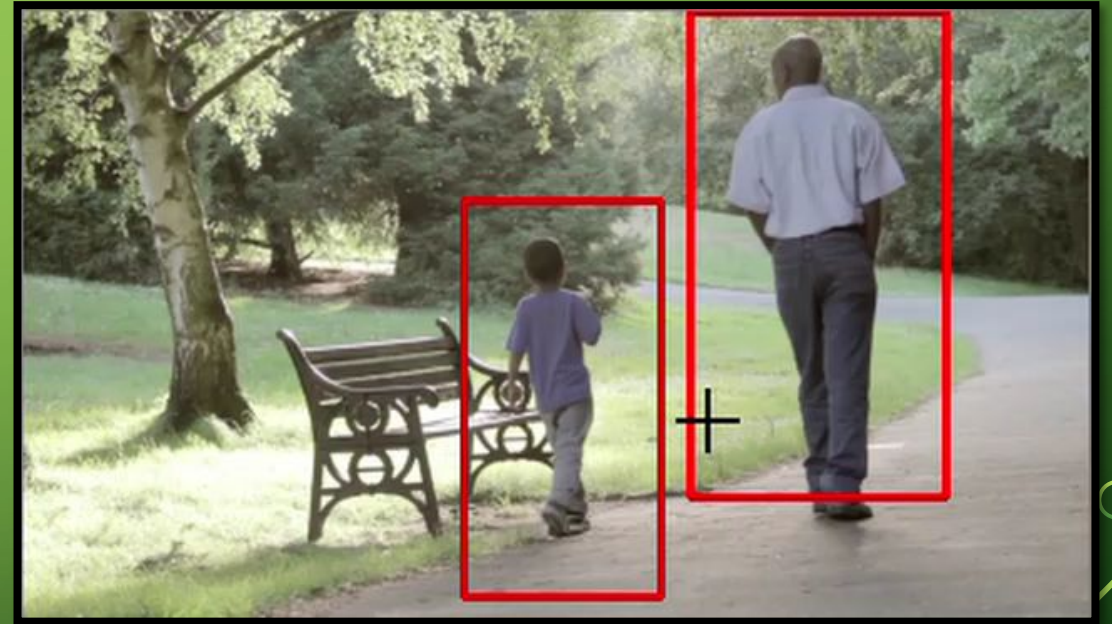
The function argparse() simply parses and returns as a dictionary the arguments passed through your terminal to our script. There will be Three arguments within the Parser:

- ☐ Image: The path to the image file inside your system
- ☐ Video: The path to the Video file inside your system
- ☐ Camera: A variable that if set to 'true' will call the cameraDetect() method.

# HUMAN DETECTION IN VIDEO

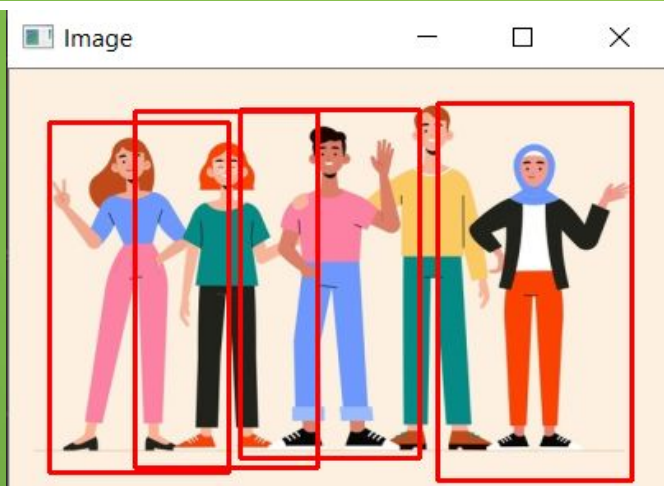


Imported from YouTube

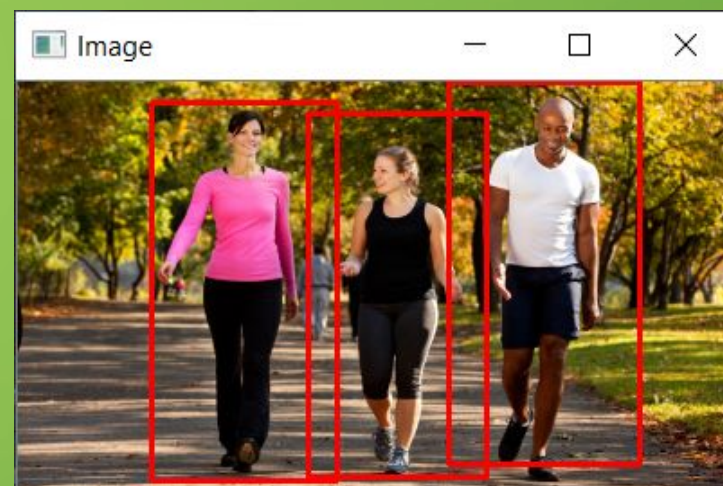


Imported from the System

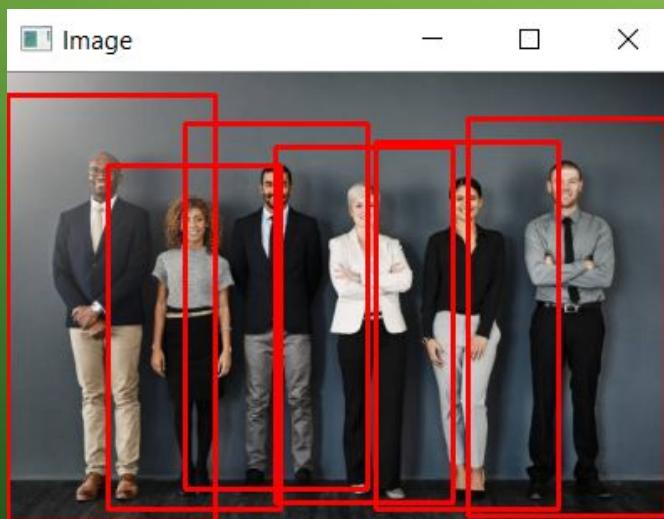
# HUMAN DETECTION IN IMAGES



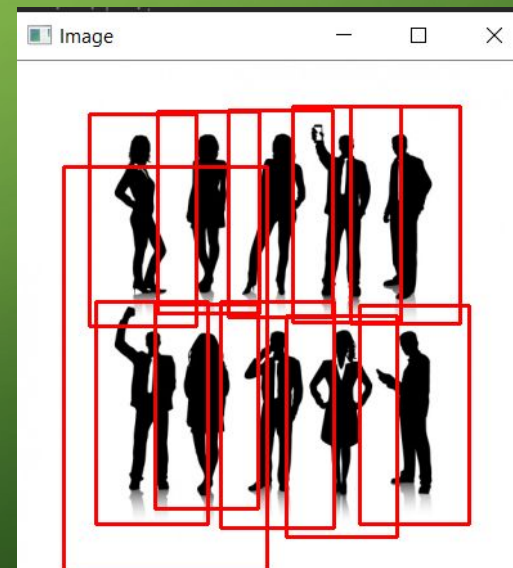
4/5 correct classifications  
1 human missed



3/3 correct classifications



6/6 correct classification



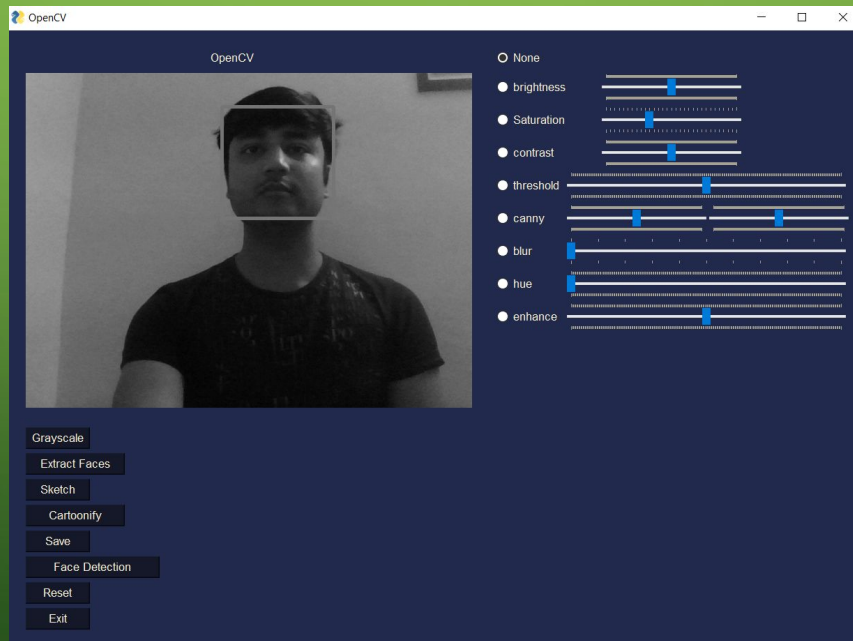
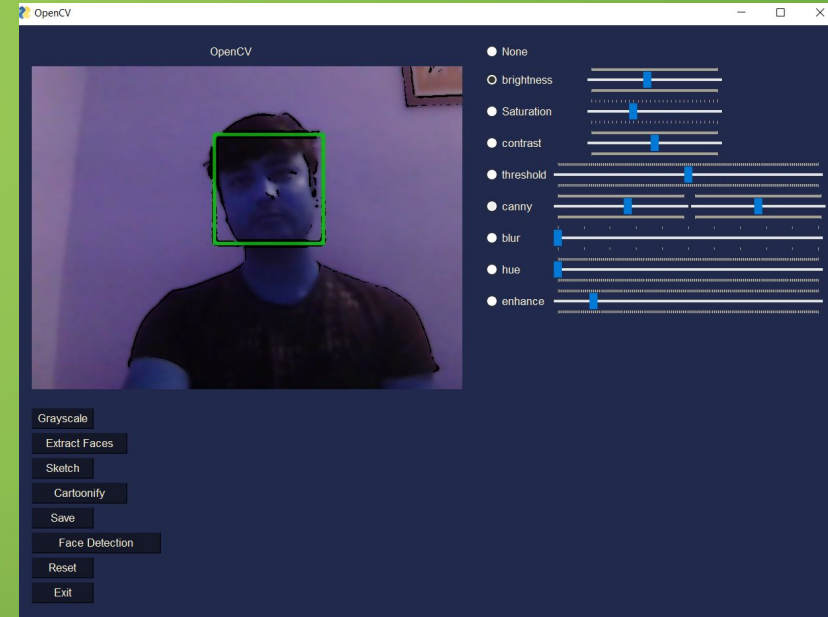
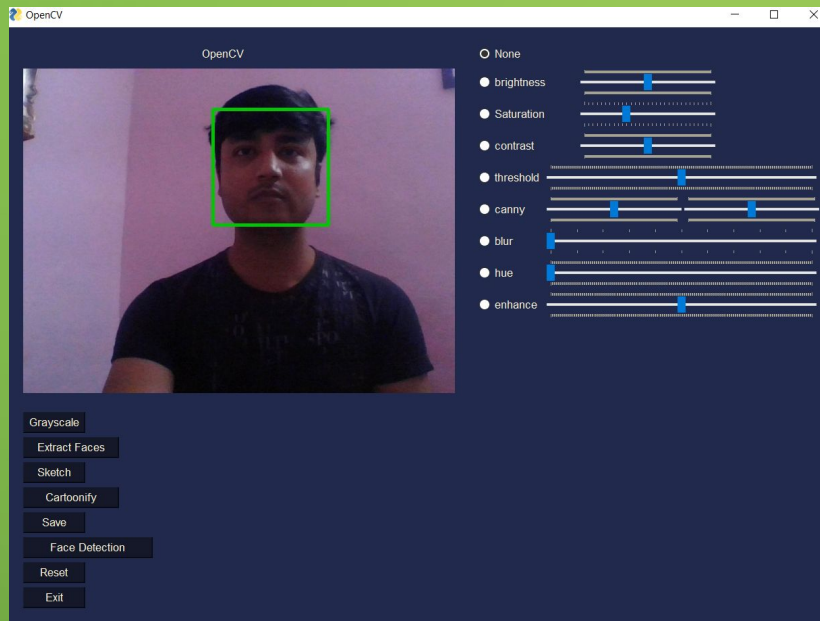
10/10 correct classifications  
1 mis-classification



# APPLICATIONS

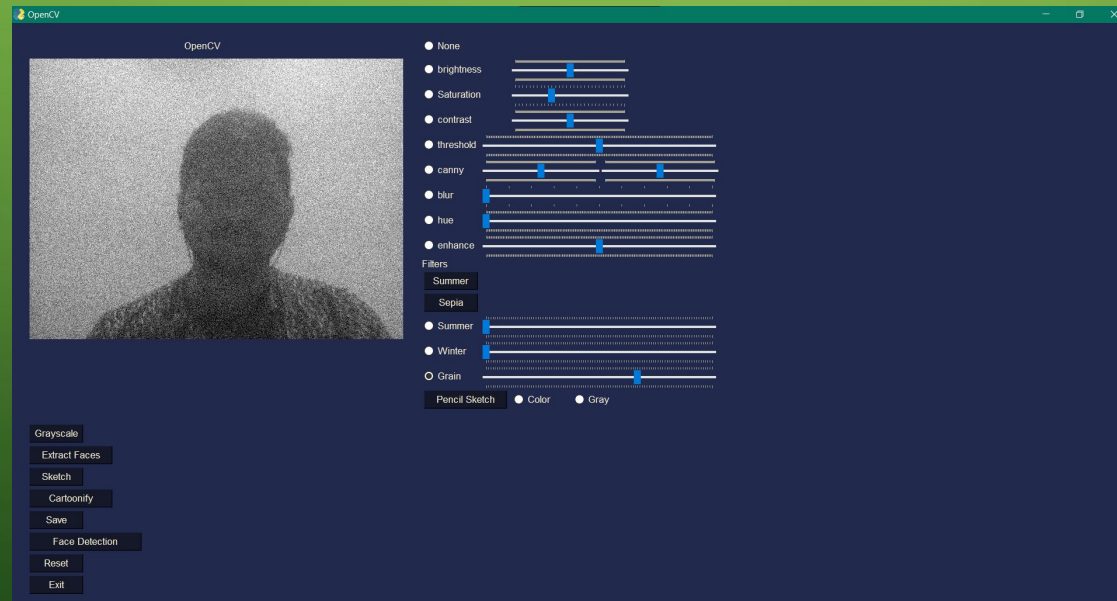
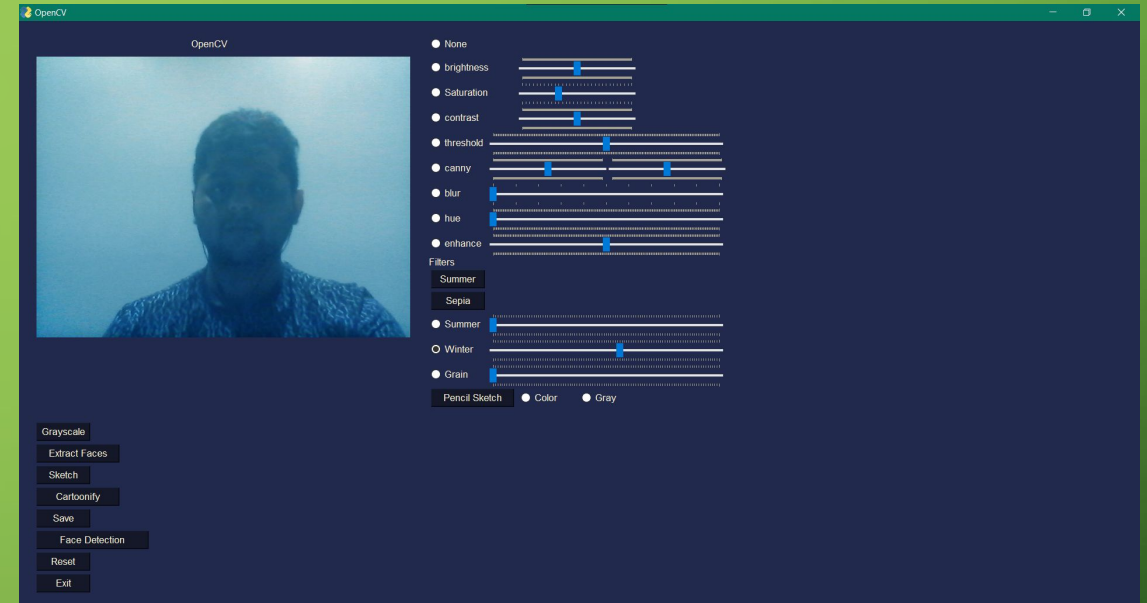
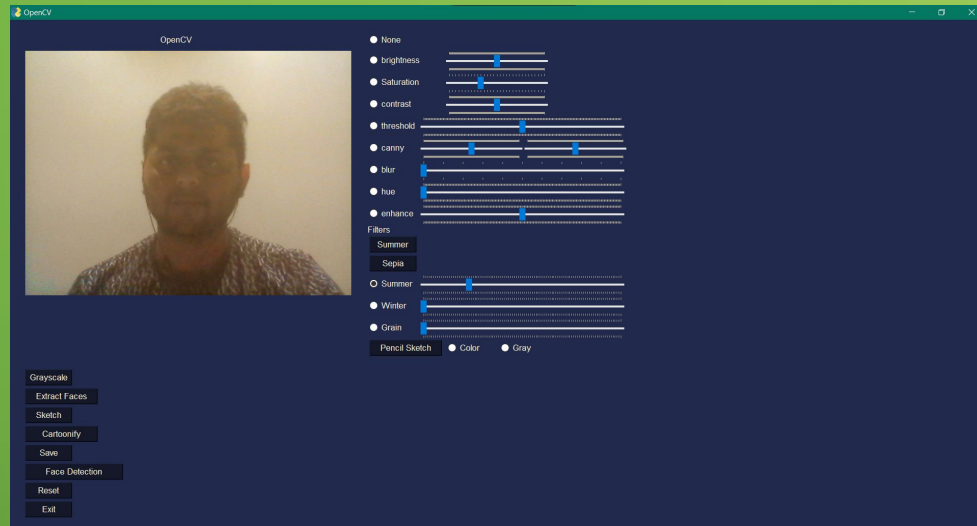
- 1.This color detection functionality can recognize and detect colors which are adequate for applications such as medical diagnosis ,color printing , computer color monitor calibration etc.
- 2.This Basic application can be further incorporated in applications which require human detection in a visual surveillance .It can help in areas like person identification , human gait characterization , congestion analysis etc.

# RESULTS



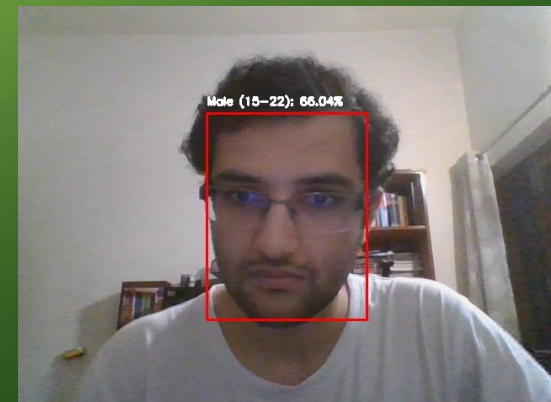


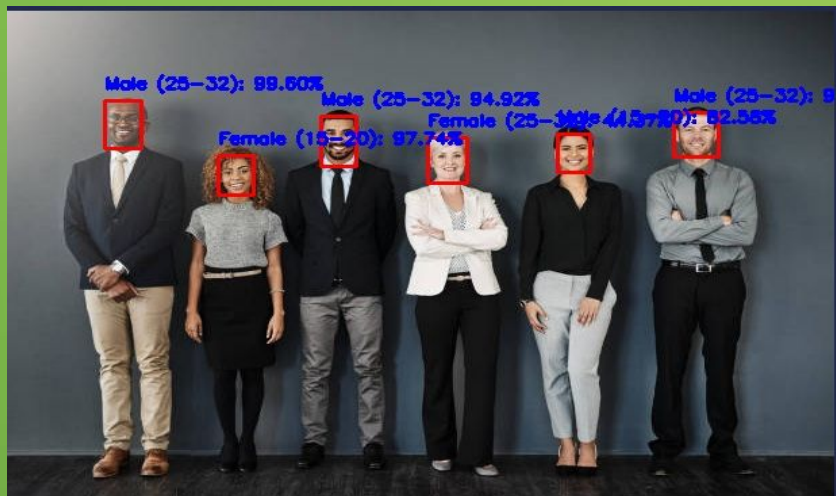
# ADDITIONAL FILTERS (UPDATE)



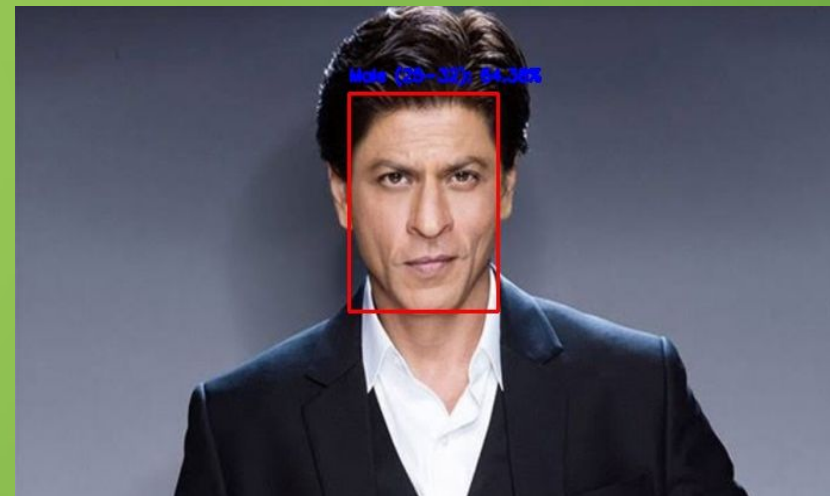
# AGE AND GENDER CLASSIFICATION

- Prototxt and CaffeModel
- Age divided in buckets=>(0-2), (4-6), (8-12), (15-22), (25-34), (35-43), (44-59), (60+)
- 





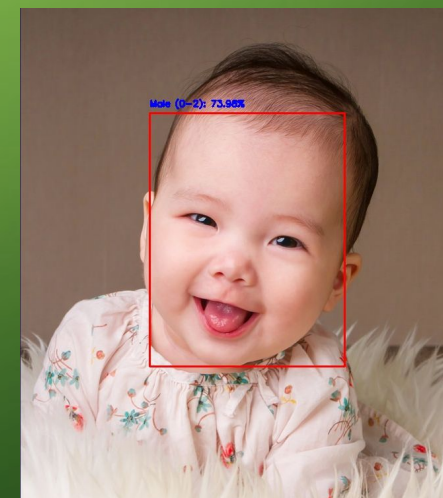
10/12 Age groups and Genders  
Classified Correctly



Gender identified correctly  
Algorithm deceived by an actor's looks



Both Age and Gender  
identified correctly



Both Age and Gender  
identified correctly



