**Acknowledgement**

We hereby declare that the work presented in this project report entitled, ”Colour Detection System” in partial fulfilment for the Degree of “Bachelor Of Computer Science” .

Our extreme gratitude to sir Dr. Kashif Hussain and Miss Soomal Fatima(Lab Incharge) who guided us throughout the project. Without her willing Disposition, spirit of accommodation, frankness, timely clarification and above all faith in us. They helped us to understand & remember important details of the project. Our project has been a success only because of their guidance.

We are especially intended & we are beholden to my friends. Finally we thanks to the member of my family for their support & encouragement.

This task expended enormous measure of work, research and commitment. In any case, execution would not have been conceivable in the event that we didn't have a help of a few people. In this manner, we might want to stretch out our genuine appreciation to every one of them.

We are also grateful to Bahria University for provision of expertise, and technical support in the implementation. Without their superior knowledge and experience, the Project would like in quality of outcomes, and thus their support has been essential.

# **Contents**

[**Contents** 2](#_Toc94643511)

[**1.1** **ABSTRACT** 2](#_Toc94643512)

[**CHAPTER 2: INTRODUCTION** 2](#_Toc94643513)

[**2.1 INTRODUCTION:** 2](#_Toc94643514)

[**2.2 PROJECT IDEA** 2](#_Toc94643515)

[**2.3 PROJECT OVERVIEW** 2](#_Toc94643516)

[ **PURPOSE** 2](#_Toc94643517)

[ **SCOPE** 2](#_Toc94643518)

[**2.4 BACKGROUND** 2](#_Toc94643519)

[**2.5 PROBLEM STATEMENT** **.**](#_Toc94643520)

[**2.6 ADVANTAGE** 2](#_Toc94643521)

[**CHAPTER 3: DATASET** 2](#_Toc94643522)

[**3.1 INTRODUCTION:** 2](#_Toc94643523)

[**3.2 DATASET** 2](#_Toc94643524)

[**CHAPTER 3: ANALYSIS** 2](#_Toc94643525)

[**3.1. PROPOSED SYSTEM:** 2](#_Toc94643526)

[**CHAPTER 4: SYSTEM IMPLIMENTATION** 2](#_Toc94643527)

[**4.1. INRODUCTION:** 2](#_Toc94643528)

[**4.2 CODE:** 2](#_Toc94643529)

[**CHAPTER 5: USER INTERFACE** 2](#_Toc94643530)

[**5.1. SAMPLE SCREEN SHOTS:** 2](#_Toc94643531)

[**8. REFERENCE** 2](#_Toc94643532)

# **ABSTRACT**

The Proposed system “Color Detection System” is supposed the essential goal of this utility is the technique for figuring out the shades of colors with a specific prediction with their names. A take a look at says, a ordinary human can capable of genuinely become aware of almost 1 million shades of colors. But within side the case of human having “enchroma”, will be capable of see simplest 1% (i.e.10,000 colors) from the ordinary humans. While portray pictures, a painter wishes to become aware of the shadeation styles precisely in any other case the truth of photo isn't always clear..

Color detection is generally a primary stage in most imaging applications when the application relies on color information, such as color. Such as traffic sign recognition, face recognition, skin color recognition, objects and object tracking, etc. modules in an image processing application is adversely affected by the above modules, high-performance color recognition accuracy inevitably becomes crucial in some applications. This project presents a method for efficient color detection in RGB space using a panel of structural experts. In this structure, a set of experts is tasked with evaluating the R, G, B components of a pixel and then constructing a degree of membership in the set of predefined color classes for the given pixel. A master neural network then builds its final decision based on the membership probabilities provided by the expert panel. Qualitative and quantitative evaluations of the results show that the proposed hierarchical structure of neural networks is superior to the conventional neural network classifier in color recognition.

# **CHAPTER 2: INTRODUCTION**

# **2.1 INTRODUCTION:**

Color detection is the system of detecting the call of any shadeation. Simple isn’t it? Well, for human beings that is a really smooth mission however for computers, it isn't always straightforward. Human eyes and brains paintings collectively to translate mild into shadeation. Light receptors which can be found in our eyes transmit the sign to the mind. Our mind then acknowledges the shadeation. Since childhood, we've mapped positive lighting with their shadeation names. We can be the use of the really identical approach to locate shadeation names.

A color detection set of rules identifies pixels in an picture that healthy a detailed color or color range. The color of detected pixels can then be modified to differentiate them from the relaxation of the picture. This instance introduces a workflow to take Simulink version of a color detection set of rules from simulation to deployment onto device: Simulate the color detection version Deploy the color detection version and use the Video Display block to expose simulated statistics at the Android device Modify the color detection version to apply real-time photographs from the Camera Block.

## **2.2 PROJECT IDEA**

In this post, I will show you how to build your own color recognizer using Python. This process is also known as “Color Detection”. We will create a basic application that will help us to detect the colors in an image. The program will also return as the RGB values of the colors, which is really helpful. Many picture designers and internet designers will apprehend how RGB values may be helpful. Building a shade recognizer is an extremely good assignment to get began out with Computer Vision.

## **2.3 PROJECT OVERVIEW**

A color detection algorithm identifies pixels in an image that match a specified color or color range. The color of detected pixels can then be changed to distinguish them from the rest of the image.

This example introduces a workflow to take Simulink model of a color detection algorithm from simulation to deployment onto device:

* Simulate the color detection model
* Deploy the color detection model and use the **picture display** block to show simulated data on the Android device
* Modify the color detection model to use real-time images from the **Camera** Block

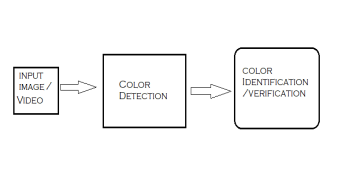


Fig 2.1

## **PURPOSE**

Color detection serves as the basis for further processing such as sorting of products and quality assessment. The proposed system intends to utilize this ability of the color sensor in order to automate further processing tasks such as sorting & arrangement of products and quality assessment

Color detection is the process of detecting the name of any color. Human eyes and brains work together to translate light into color. Light receptors that are present in our eyes transmit the signal to the brain. Our brain then recognizes the color.

## **SCOPE**

The color sensor detects the color of the surface, usually in the RGB scale. Color is the result of interaction between a light source, an object, and an observer. Color sensors have a variety of applications including detection of the environment, choosing the right product and sorting.

Color sensing technology along with the ability to accurately measure the ambient light CCT and intensity enables smartphones to increase its user's experience. The color sensors distinguish between natural and artificial light which is helping smartphone manufacturers to extensively use them in the optical lenses of smartphones.

## **2.4 BACKGROUND**

# The main objective of this application is the methodology for identifying the shades of colors with an exact prediction with their names. A study says, a normal human can able to clearly identify nearly 1 million shades of colors. But in the case of human having “enchroma”, could be able to see only 1% (i.e.10,000 colors) from the normal humans. While painting pictures, a painter needs to identify the color patterns exactly or else the reality of image is not clear.

## **2.5 ADVANTAGE**

* Our project of Color detection System can work in any computer with minimum specification.
* The detection process process takes less than a moment and this is very beneficial for companies.
* The first thing to start with is observation. With the help of Color Detection, it will be easy to identify the color and give its name to the persons using it.
* This project is made to reach each and everyone in the society suffering from color blindness so that everyone can get benefit from this.

# **CHAPTER 3: DATASET**

## **3.1 INTRODUCTION:**

Colors include 3 primary colors; red, green and blue. In computers, we define each color value in the range from 0 to 255. So how many ways can we define a color? The answer is 256 \* 256 \* 256 = 16,581,375. There are about 16.5 million different ways to represent a color. In our dataset, we need to map the values ​​of each color with their respective names. But don't worry, we don't need to map all the values. We will use a dataset containing RGB values ​​with their respective names. The CSV file for our dataset is taken from this link:

<https://github.com/codebrainz/color-names/blob/master/output/colors.csv>

The colors.csv file includes 865 color names along with their RGB and hex values.

## **3.2 DATASET**

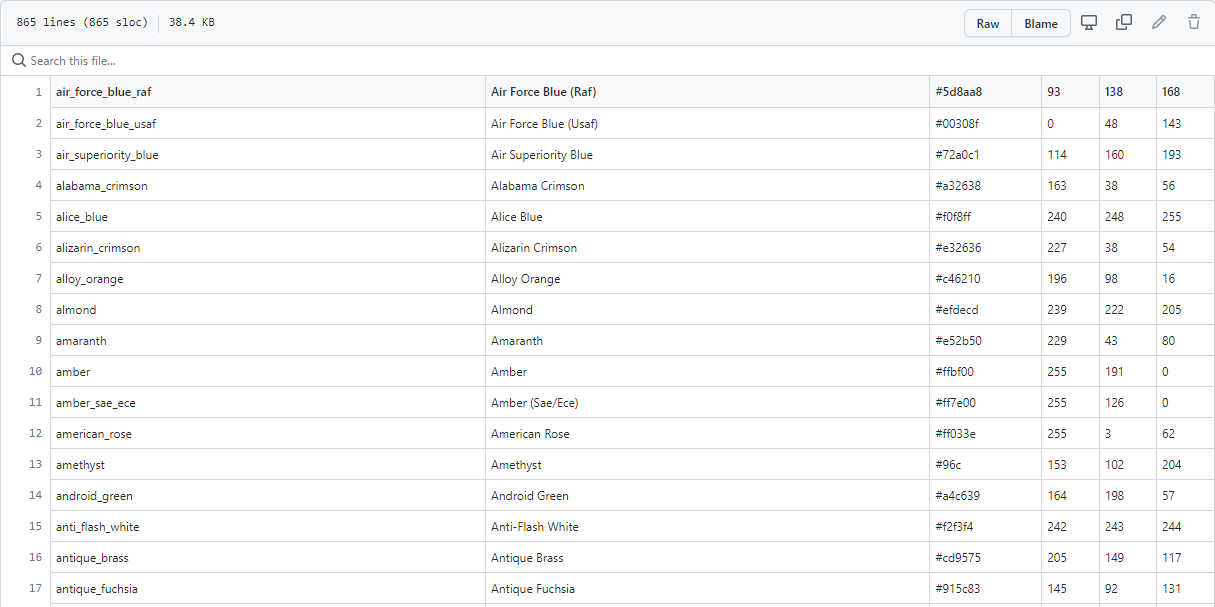


Fig 3.1

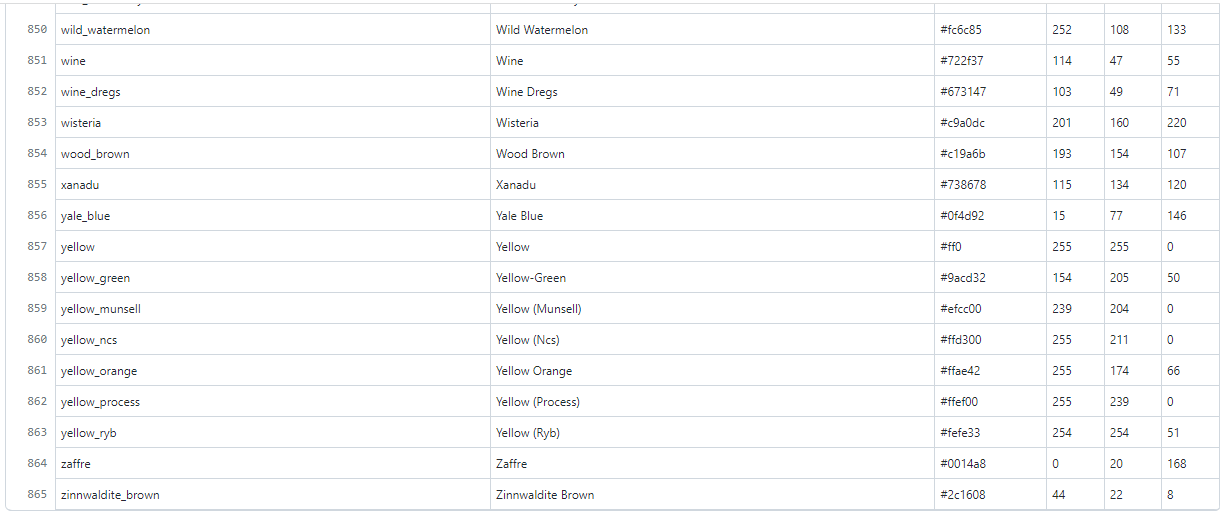


Fig 3.2

# **CHAPTER 4: ANALYSIS**

## **4.1. PROPOSED SYSTEM:**

1. Our project of Color detection System can work in any computer with minimum specification.
2. The detection process process takes less than a moment and this is very beneficial for companies.
3. The first thing to start with is observation. With the help of Color Detection, it will be easy to identify the color and give its name to the persons using it.
4. This project is made to reach each and everyone in the society suffering from color blindness so that everyone can get benefit from this.
5. Color Detection technology is very accurate and no one can doubt it.
6. The project designed by us can be used through basic camera also. No special cameras are required for its basic functioning,
7. But for using it as an assistant in self drive cars the camera should be good and more than one should be used for getting.

# **CHAPTER 5: SYSTEM IMPLIMENTATION**

## **5.1. INTRODUCTION:**

* We have to calculate 1st order spatial moments around x-axis and y-axis and the 0th order central moments of the binary image.
* 0th order central moments of the binary image are equal to the white area of the image in pixels.

## **5.2 CODE:**

import cv2

import numpy as np

import pandas as pd

import argparse

#Creating argument parser to take image path from command line

ap = argparse.ArgumentParser()

ap.add\_argument('-i', '--image', required=True, help="Image Path")

args = vars(ap.parse\_args())

img\_path = args['image']

#Reading the image with opencv

img = cv2.imread(img\_path)

#declaring global variables (are used later on)

clicked = False

r = g = b = xpos = ypos = 0

#Reading csv file

index=["color","color\_name","hex","R","G","B"]

csv = pd.read\_csv('colors.csv', names=index, header=None)

#function to calculate minimum distance from all colors and get the most matching color

def getColorName(R,G,B):

    minimum = 10000

    for i in range(len(csv)):

        d = abs(R- int(csv.loc[i,"R"])) + abs(G- int(csv.loc[i,"G"]))+ abs(B- int(csv.loc[i,"B"]))

        if(d<=minimum):

            minimum = d

            cname = csv.loc[i,"color\_name"]

    return cname

#function to get x,y coordinates of mouse double click

def draw\_function(event, x,y,flags,param):

    if event == cv2.EVENT\_LBUTTONDBLCLK:

        global b,g,r,xpos,ypos, clicked

        clicked = True

        xpos = x

        ypos = y

        b,g,r = img[y,x]

        b = int(b)

        g = int(g)

        r = int(r)

cv2.namedWindow('image')

cv2.setMouseCallback('image',draw\_function)

while(1):

    cv2.imshow("image",img)

    if (clicked):

        cv2.rectangle(img,(20,20), (750,60), (b,g,r), -1)

        text = getColorName(r,g,b) + ' R='+ str(r) +  ' G='+ str(g) +  ' B='+ str(b)

        cv2.putText(img, text,(50,50),2,0.8,(255,255,255),2,cv2.LINE\_AA)

        if(r+g+b>=600):

            cv2.putText(img, text,(50,50),2,0.8,(0,0,0),2,cv2.LINE\_AA)

        clicked=False

    #Break the loop when user hits 'esc' key

    if cv2.waitKey(20) & 0xFF ==27:

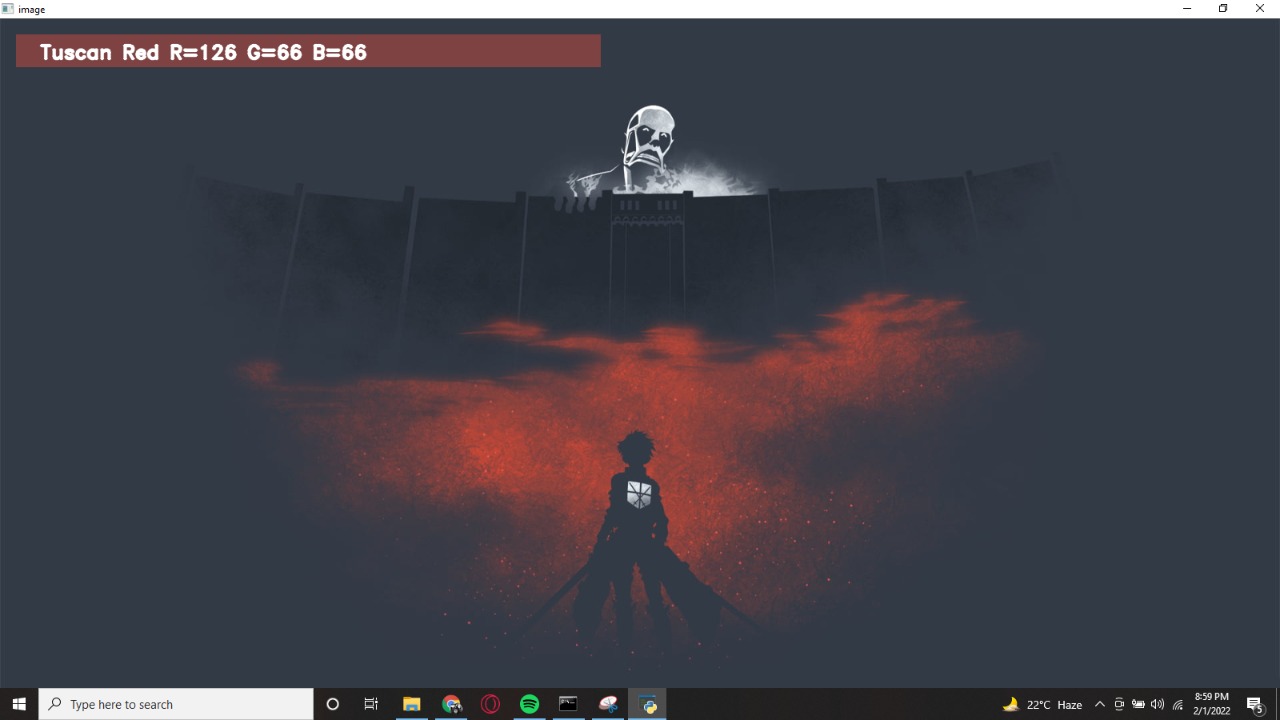
        break

cv2.destroyAllWindows()

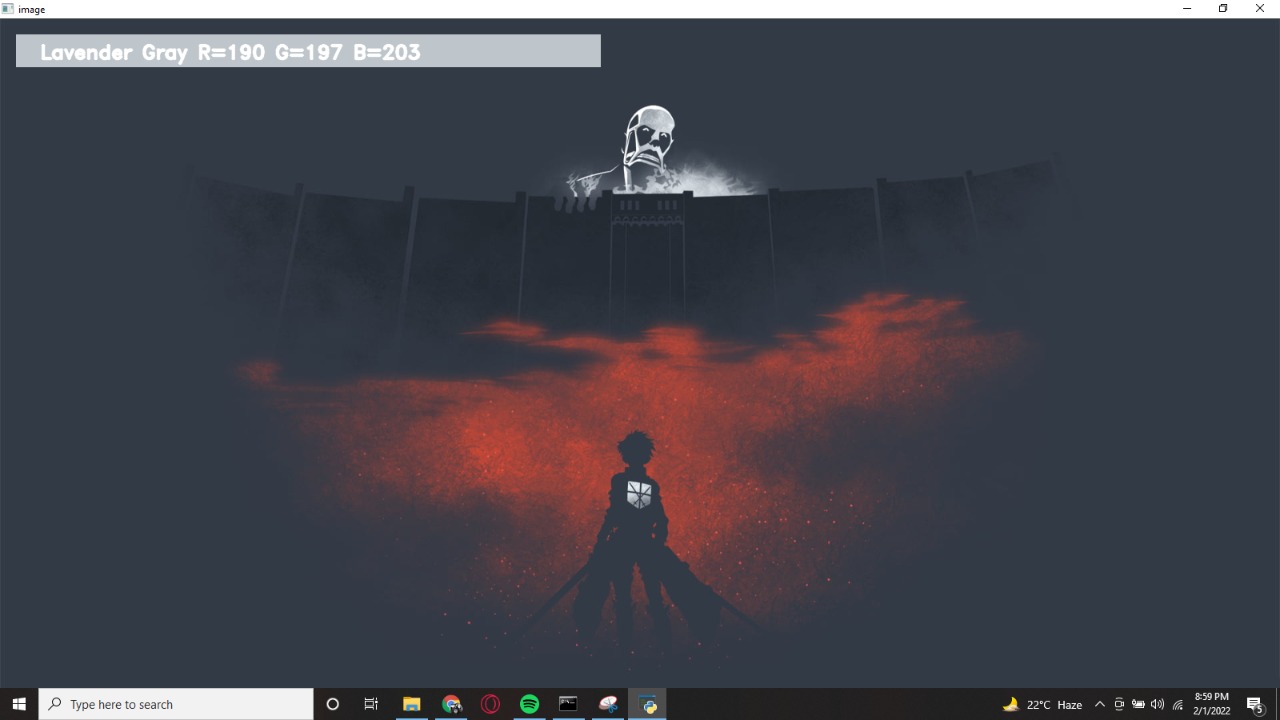
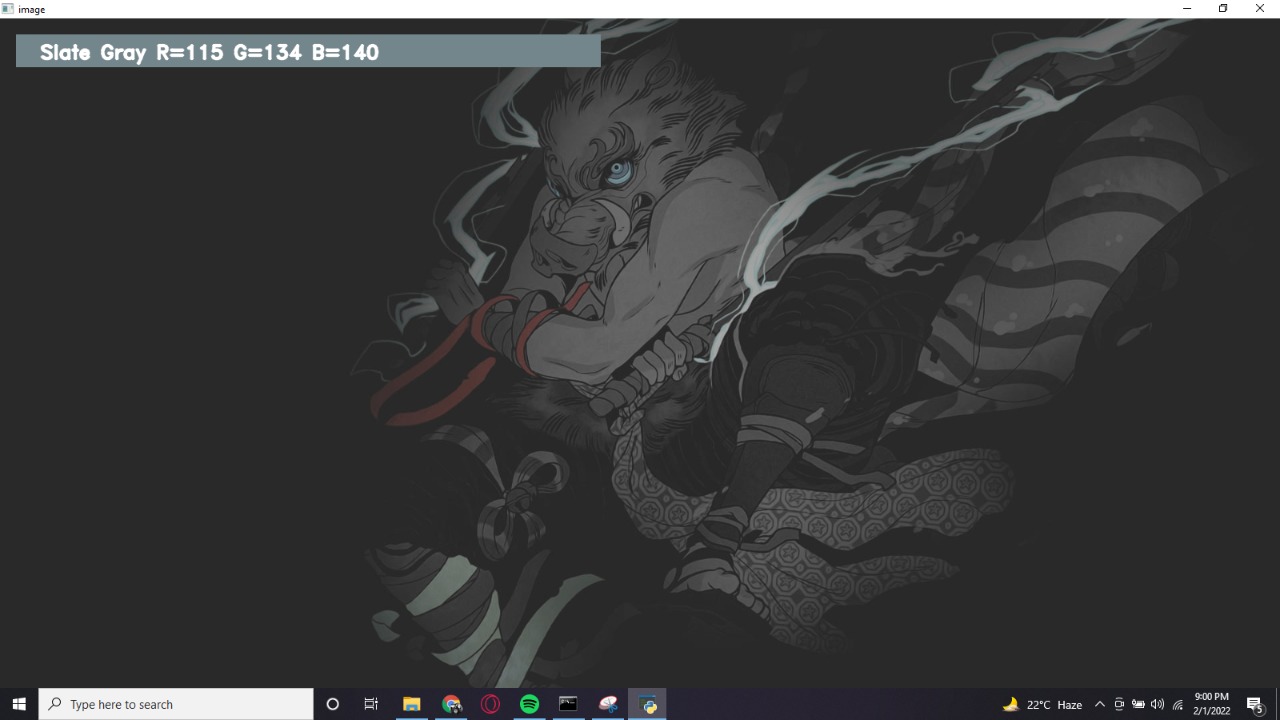
# **CHAPTER 6: USER INTERFACE**

## **6.1. SAMPLE SCREEN SHOTS:**

**Fig. 6.1**



**Fig. 6.2**

****

**Fig 6.4**

**Fig. 6.3**

**7. CONCLUSION:**

In the present project titled as “Color Detection System” using Python by using Jupyter notebook, we implement the extraction of color RGB values and color name along with the pixel.

Color detection can easily be implemented and helps the user to get the colors easily. Colors are as true as the sun or the moon and is always present in the environment, but unlike normal people, people suffering from color blindness can't easily identify the colors. We are seeing an increase in people suffering from this and we know that it can be very frustrating to not able to see the colors what least can be done is to tell them the real colors namely based on clusters, taken at the time of arrest, before the judge includes the possibility of conviction or innocence. Mug shot photos are often not far from the details, even though the detainee has never been charged.

**8. FUTURE WORK**

Color sensing technology has come a long way and has a long way to go. When we see driverless cars running on the road their own by following the traffic rules. Nice weather machines are ready to do it. Tesla is the front runner in this field Technology. However, next generation color detection programs will have more upgrades. Smart app environment where computers and devices are similar to assistant assistants.

To achieve this goal, the computer must be able to define neighboring objects and their basic properties such as size shapes and colors (we can't forget that) in a way fit naturally within the normal human pattern. Surname does not require special interactions and is subject to compliance people's understanding of when to be recognized. This suggests that future intelligent environments should use methods are like humans, and have the same limitations. These goals are now achievable.

# **9. REFERENCE**

* <https://towardsdatascience.com/building-a-color-recognizer-in-python-4783dfc72456>
* <https://towardsdatascience.com/color-identification-in-images-machine-learning-application-b26e770c4c71>
* <https://data-flair.training/blogs/project-in-python-colour-detection/>
* https://ieeexplore.ieee.org/document/5946088