An Effective Solution for Color Blind People
using
Color Detection Model

A

Project Report

Submitted in partial fulfillment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Specialization in

Business Analytics and Optimization

BY

Name	Roll No.	SAPID
ADESH KUMAR GUPTA	R103216006	500053000
SHANKEY GUPTA	R103216089	500054226
TUSHAR SINGH	R103216109	500054191
UTKARSH SANDEEP SINGH	R103216110	500053648

Under the guidance of

Dr. Hitesh Kumar Sharma
AP & PIC

Department of Informatics

School of Computer Science

University of Petroleum & Energy Studies

Bidholi, via Prem Nagar, Dehradun, UK

May 2019

UPES

UNIVERSITY WITH A PURPOSE

CANDIDATES DECLARATION

We hereby certify that the project work entitled **An Effective Solution for Color Blind People using Color Detection Model** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering with Specialization in Business Analytics and Optimization and submitted to the Department of Informatics at School of Computer Science, University of Petroleum And Energy Studies, Dehradun, is an authentic record of our work carried out during a period from **January, 2019 to May, 2019** under the supervision of **Dr Hitesh Kumar Sharma**, **AP & PIC.**

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

(Name of Student(s))

Adesh, Shankey, Tushar, Utkarsh

Roll No.

R103216006, R103216089,

R103216109, R103216110

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Date: 20 May 2019)

(Name of Guide)
Project Guide

Dr. T.P Singh

Head
Department of Informatics
School of Computer Science
University of Petroleum And Energy Studies
Dehradun - 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide Name, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our Head of the Department, Dr. T.P Singh, for his great support in doing our project name at SoCS.

We are also grateful to Dr. Manish Prateek Professor and Dean SoCS, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Name	Shankey Gupta	Tushar Singh	Utkarsh	Adesh
Roll No.	R103216089	R103216109	R103216110	R103216006

Abstract

In this project a model is been created which aims to provide a solution for the people that suffer from a well-known vision deficiency called color blindness. The model aims to help people like them in detecting the colors which is hard for them to recognize. As color blind people are also restricted from various areas which also include some professional fields just because of their lack of distinguishing some different colors. By developing this model, we can make a system to reduce this deficiency by the use of technology, so that it can help them without even getting it cured.

The technique being used in order to develop a setup for the identification of colors is Image Processing. It uses 2 approach first being to detect the color and the second being the edge detection of the object. It then processes the image of the object placed in front of the camera and displays the color along with its corresponding color code depending of the shade of the color. The model after its successful development in future can be later transformed in form of some portable device which can be carried by people suffering from color blindness to use it whenever they are in need.

TABLE OF CONTENTS

Contents

1. Introduction:

Now in order to implement the setup and develop the model which would help as an aid for the color blind people, some techniques and methods needs to be put in use which include libraries like Numpy, OpenCV and a technique called Image Processing.

OpenCV(Open Source Computer Vision) library aims at real time Computer Vision. It is mainly used to do all the operations related to images.

Numpy is Python Package which stands for Numerical Python. This library consists of multidimensional array objects and a collection of routines for processing of array.

Image Processing technique is used to perform some certain operations on an image, in order to get an enhanced image as an output or to extract some useful information from the image. It acts as a type signal processing in which input is an image and output may be an image or charactristics/features associated with that image.

2. Problem Statement:

Nowadays most of the people can be seen as a victim of Color Blindness that is incurable disease because of genetic disorder. It can be cured by some genetic therapy but it is very much costly. The problem of them is that these people are unable to differentiate between shades of color or when two colors are mixed together so it will be very difficult for them to see item's colors clearly. So the problem is how it can be analyzed without curing the disease.

3. Literature Review:

[1] Color can be identified from the sensory optic nerves of the eyes. Color can only be seen or identified when a source of light is applied to an object. Color blindness can be termed as inability of the differentiation between colors. It is incurable disease that can be termed as lifelong disease. Edges can be very helpful in color differentiation boundary.

[2] Color detection model can be used in mixing of colors especially in paints, dyes and color pigments. It can be also very helpful in to differentiating colors that are used in robotics and in other medical fields. It can also be used in Graphic Arts Industry. Other implementations can also be used in agricultural industry like especially detection of quality of soil.

[3] Color Detection can be used in agriculture industry to find the weeds the along with the crops. Via color detection weeds can be identified and destroyed and the crops can be saved. It can be also used in medical industries to detect the disease and other disorders especially in face and other internal diseases like cancers.

[4] The main aim of computer vision is to analyze the behavior of human eye and the reduction of human effort. Through computer vision various task can be done that is done by human eye, whether to detect the object or identify its color. By this method it is very

helpful to detect the symptoms of the disease and the other applications in other industries like agriculture.

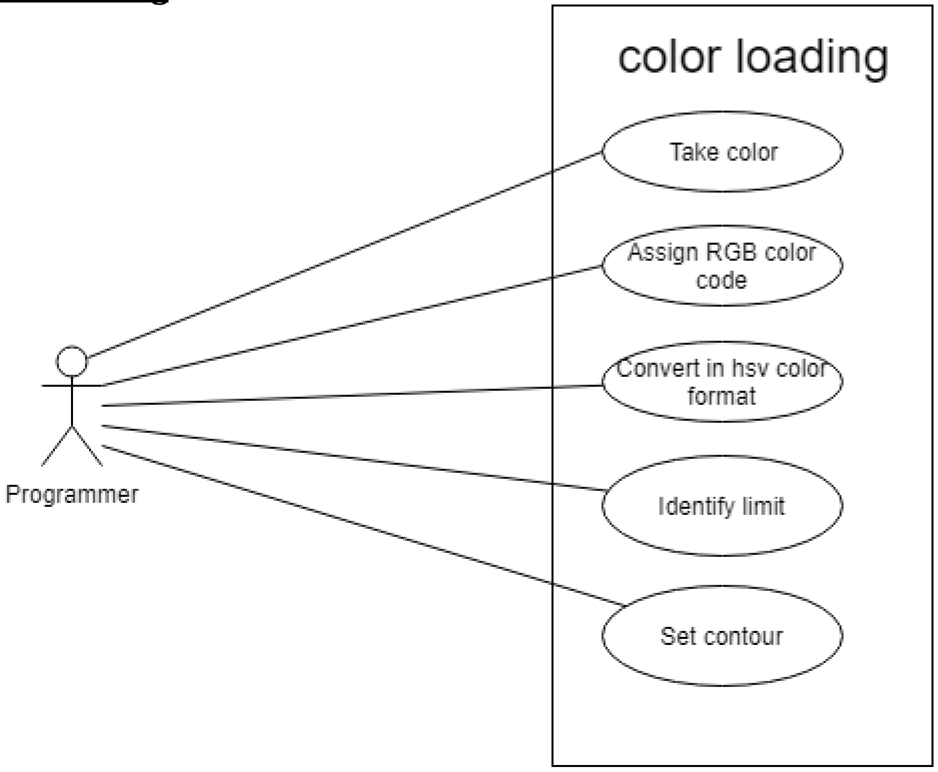
4. Objectives:

a) To identify the color of the given object kept in front of the camera.
b) To represent the color found on the object and generating its respective color code.

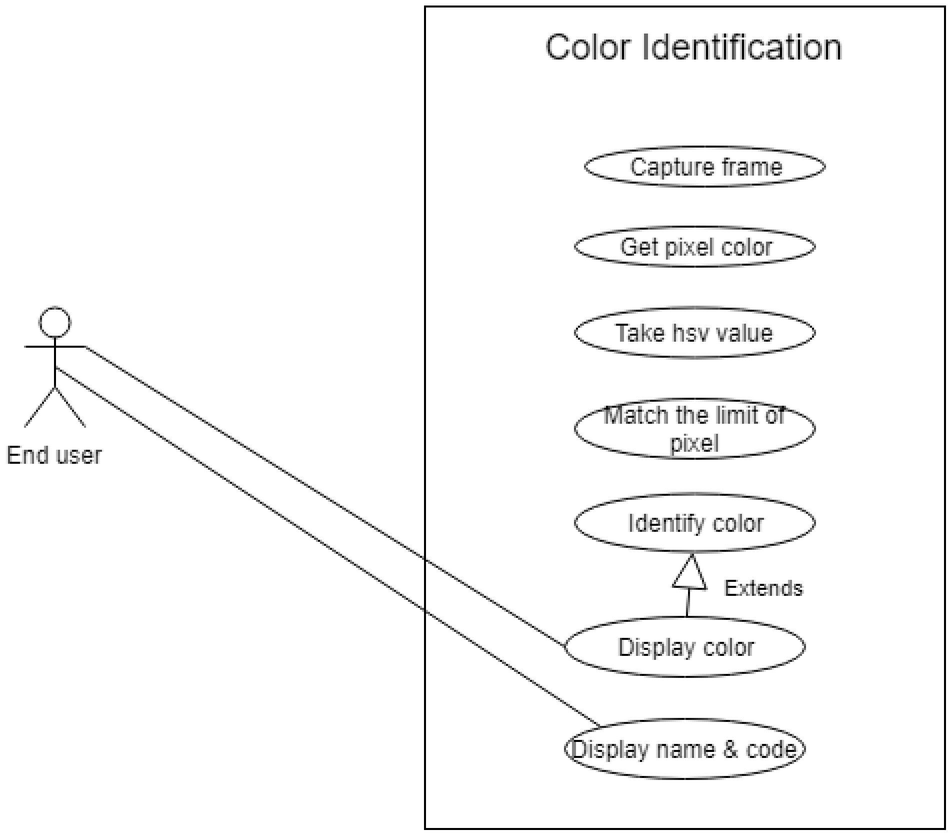
5. Design:

5.1 Use Case Diagram

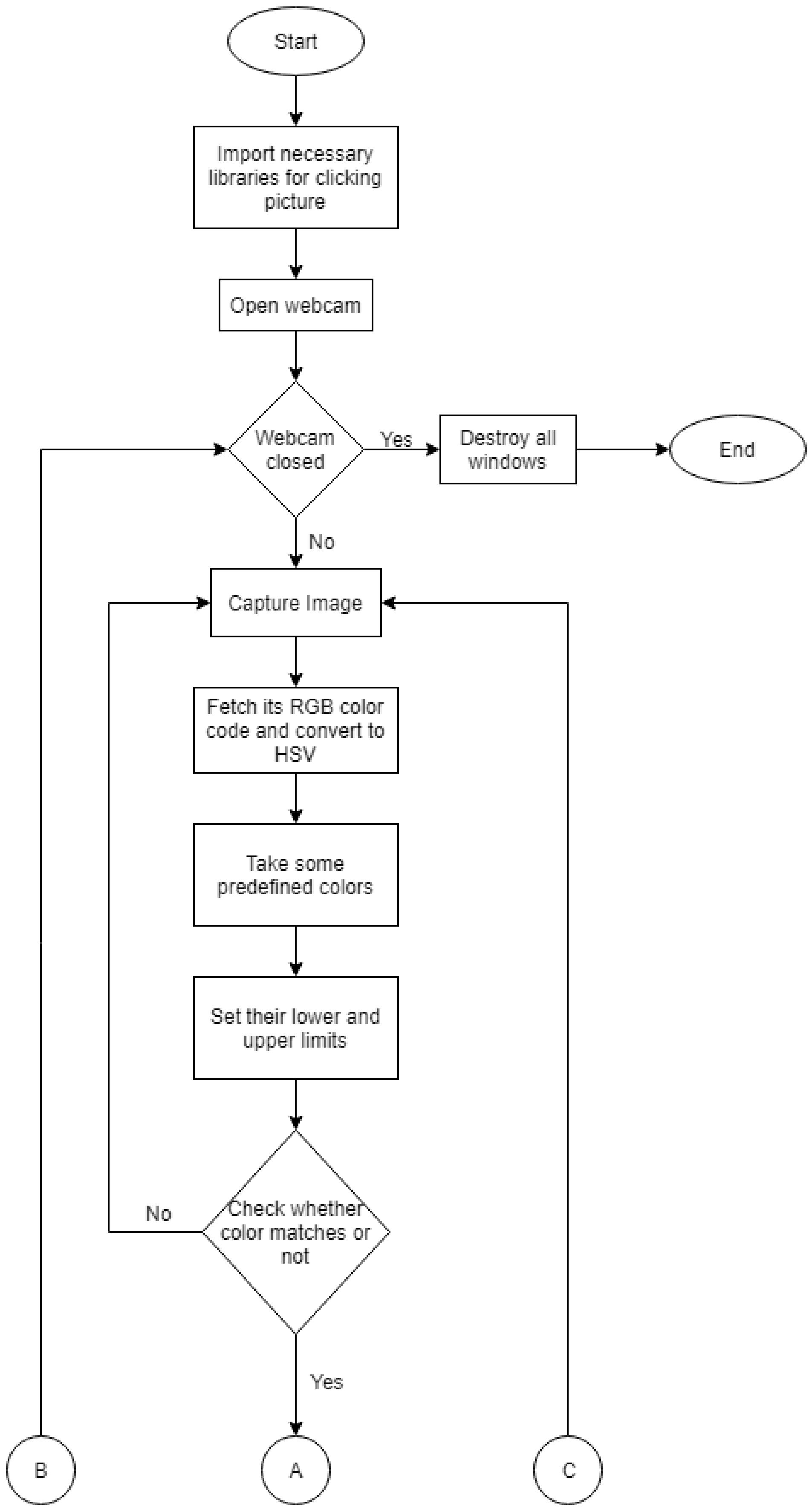
Color Loading

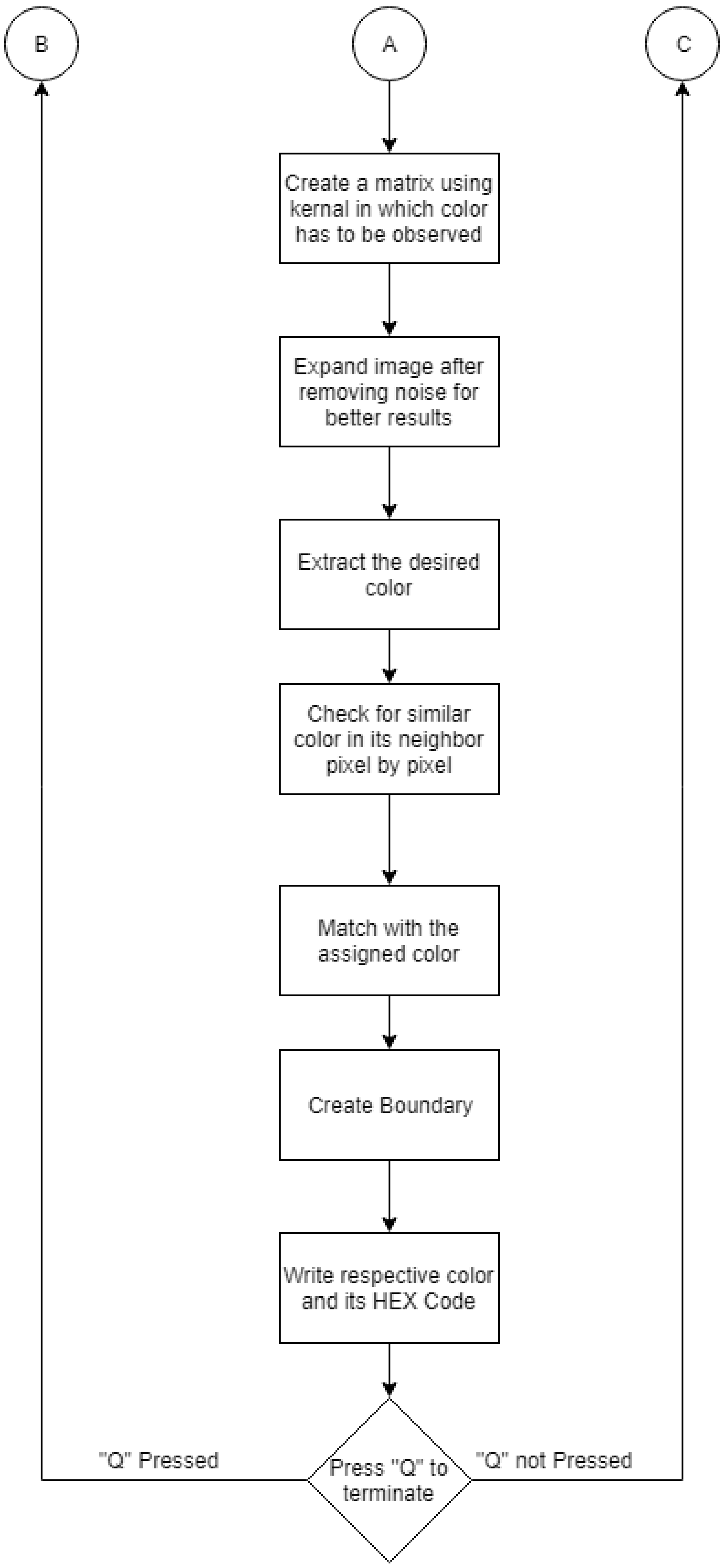


- **<u>Color Identification</u>**



5.2 Flowchart


Start
Import necessary libraries for clicking picture
Open webcam
Webcam closed
Yes
Destroy all windows
End
No
Capture Image
Fetch its RGB color code and convert to HSV
Take some predefined colors
Set their lower and upper limits
Check whether color matches or not
No
Yes
B
A
C


B
A
C
Create a matrix using kernal in which color has to be observed
Expand image after removing noise for better results
Extract the desired color
Check for similar color in its neighbor pixel by pixel
Match with the assigned color
Create Boundary
Write respective color and its HEX Code
Press "Q" to terminate
"Q" Pressed
"Q" not Pressed

6. Methodology:

6.1 Theory:

Color detection model is used to find the respective color, and its' shade. Color detection model will be useful for people having disorder of color blindness, agricultural fields and in medical fields as well.

For the implementation of this technique, we need to have some python libraries:

OpenCV

To use camera and to identify object's color from that camera in our code.

Numpy

To store the range of the color on the screen, i.e. to identify the location of the colors present on the screen.

6.2 Approach:

- Importing Modules
- Capture livestream video through webcam
- Converting image frame from BGR(Blue-Green-Red) to HSV(Hue-Saturation-Value)
- Defining the range of each color in the image.
- Find range of the colors in the frame
- Track the color and draw a rectangle around it
- Display output as a video stream in a window with the colors being tracked displaying their names and color codes as soon as it tracks an object.

6.3 System Requirements: (Software/Hardware)

Hardware:
Ram- 16GB
Processors- Intel Core i3/i5/i7
HDD- 1GB
Software:
Operating System- Windows 10/8.1/8/7/XP |Ubuntu
Programming Language- Python

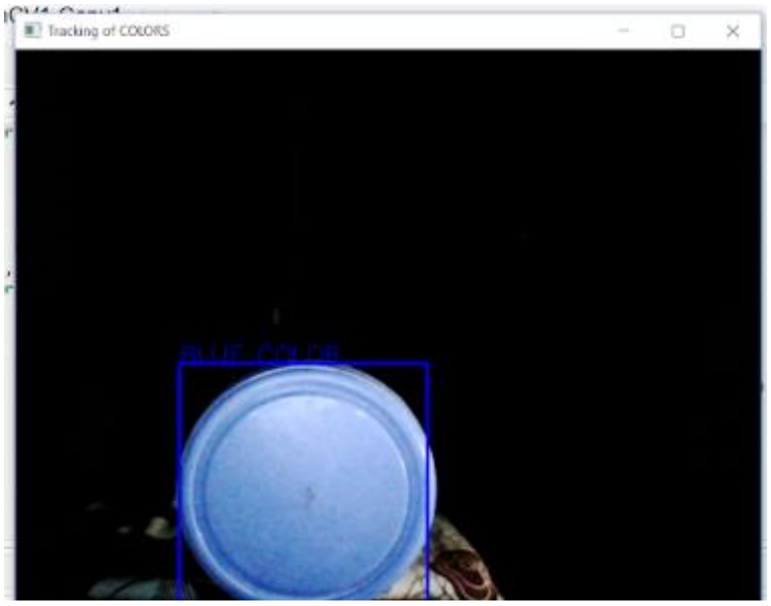
7. Implementation:

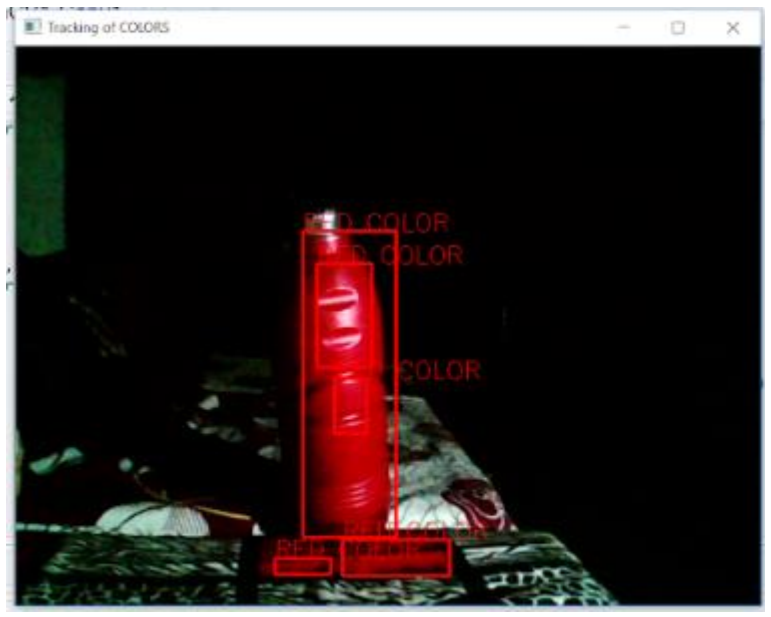
7.1 Algorithm

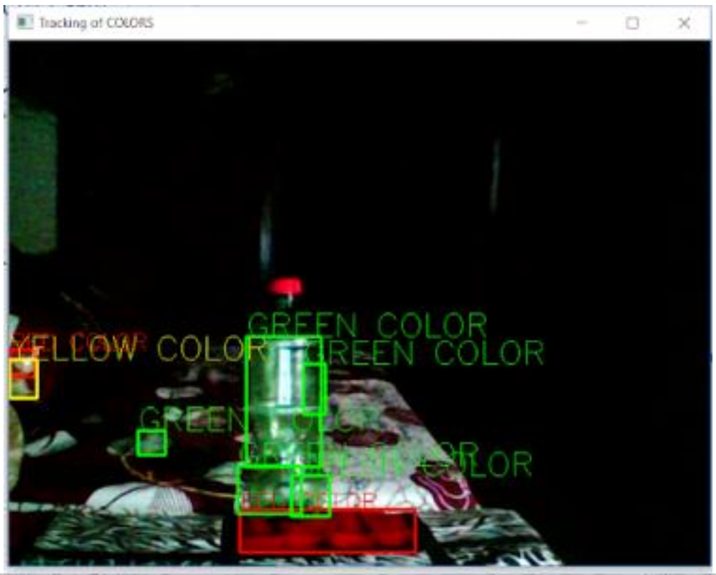
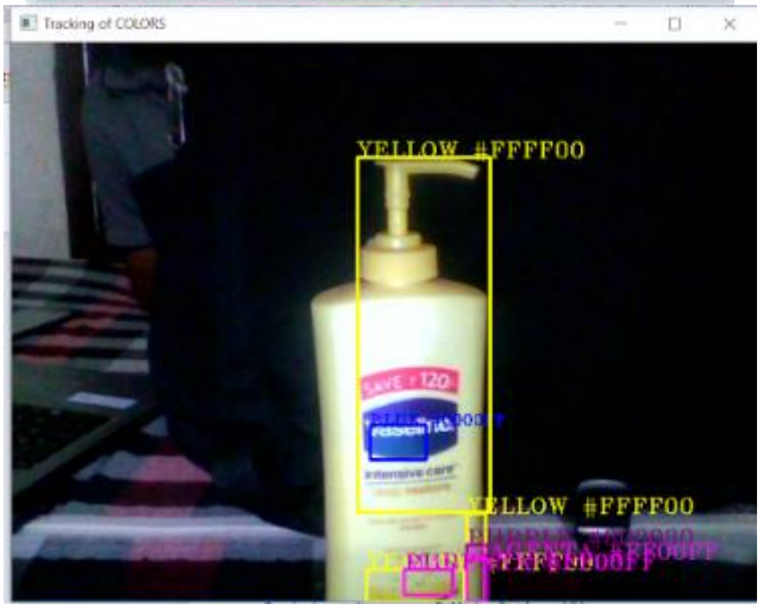
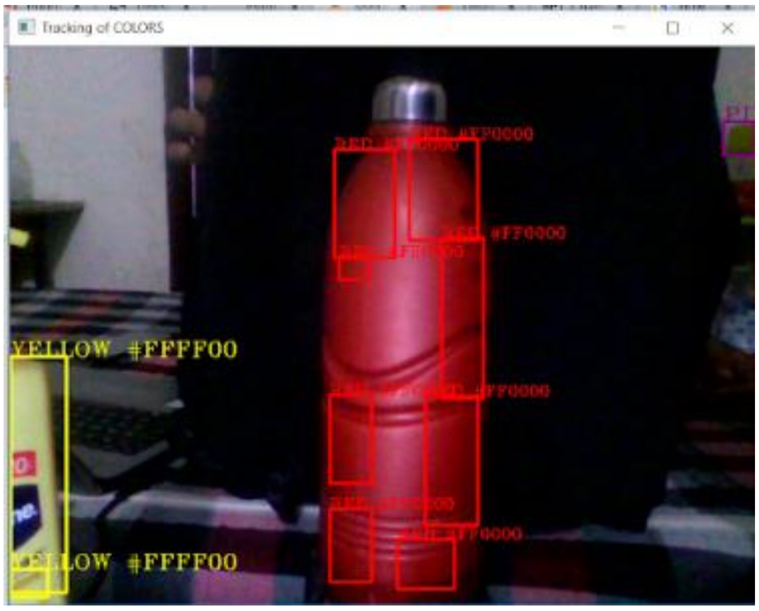
- Import required Modules
- Capture livestream video through webcam
- Converting image frame from BGR(Blue-Green-Red) to HSV(Hue-Saturation-Value)
- Defining the range of each color in the image.
- Find range of the colors in the frame
- Track the color and draw a rectangle around it

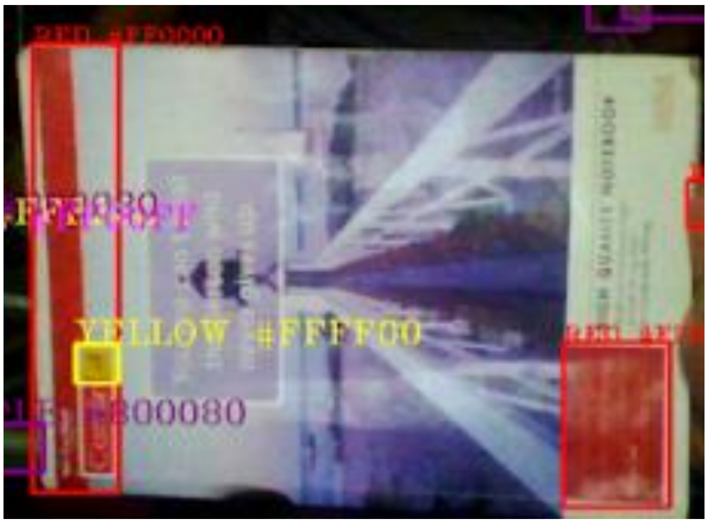
- Display output as a video stream in a window with the colors being tracked displaying their names and color codes as soon as it tracks an object.

7.2 Output Screen



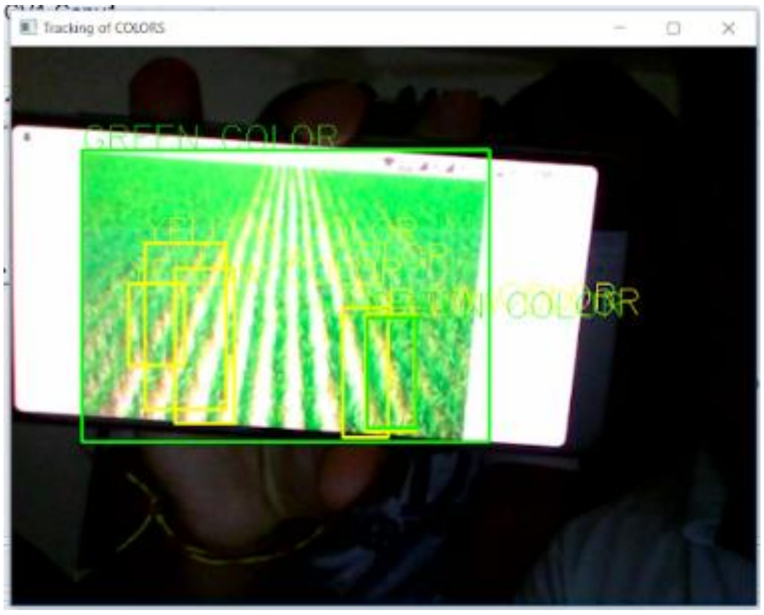



Tracking of COLORS
YELLOW COLOR
GREEN COLOR
GREEN COLOR
GREEN COLOR
GREEN COLOR

Tracking of COLORS
YELLOW #FFFF00
YELLOW #FFFF00

Tracking of COLORS
RED #FF0000
RED #FF0000
YELLOW #FFFF00
RED #FF0000
YELLOW #FFFF00



7.3 Result Analysis

Identification of weeds and crops in an agricultural field.



8. Conclusion and Future Scope:

By using the concept of color detection model, we were able to identify different colors of different objects placed in front of the camera and produce the corresponding accurate color output. The future of this model is that it can be implemented in some IOT based devices which can then can be utilized by color blind people in order to identify and recognize those colors in which they have problems. The model can further act as a helping source for them which will also prevent the high cost for getting the deficiency treated.

9. Schedule: (PERT Chart)

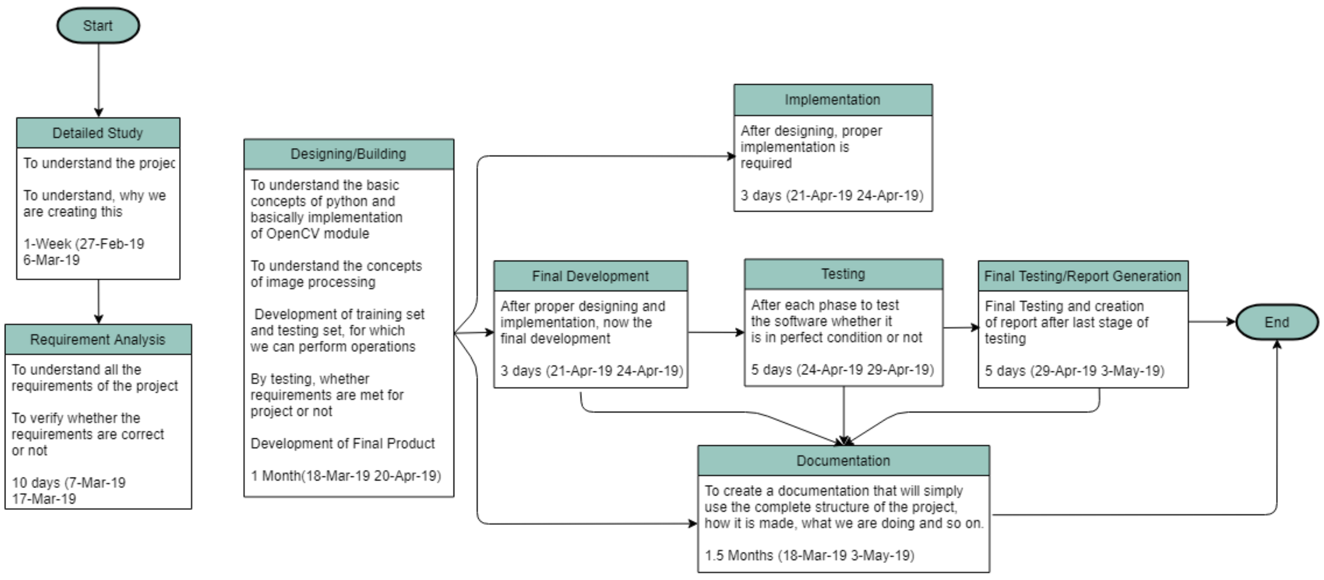

Start
Detailed Study
To understand the projec
To understand, why we are creating this
1-Week (27-Feb-19 6-Mar-19
Requirement Analysis
To understand all the requirements of the project
To verify whether the requirements are correct or not
10 days (7-Mar-19 17-Mar-19
Designing/Building
To understand the basic concepts of python and basically implementation of OpenCV module
To understand the concepts of image processing
Development of training set and testing set, for which we can perform operations
By testing, whether requirements are met for project or not
Development of Final Product
1 Month(18-Mar-19 20-Apr-19)
Implementation
After designing, proper implementation is required
3 days (21-Apr-19 24-Apr-19)
Final Development
After proper designing and implementation, now the final development
3 days (21-Apr-19 24-Apr-19)
Testing
After each phase to test the software whether it is in perfect condition or not
5 days (24-Apr-19 29-Apr-19)
Final Testing/Report Generation
Final Testing and creation of report after last stage of testing
5 days (29-Apr-19 3-May-19)
End
Documentation
To create a documentation that will simply use the complete structure of the project, how it is made, what we are doing and so on.
1.5 Months (18-Mar-19 3-May-19)

Fig 3. Pert Chart

10. References:

Documented Reference:

[1] An image processing technique for color detection and distinguish patterns with similar color: An aid for color blind people
https://www.researchgate.net/publication/282270361

[2] Methods and Means for Color Detection and Modification
https://patents.google.com/patent/US4488245A/en

[3] Method for color detection in video images
https://patents.google.com/patent/US6574363B1/en

[4] Study on Object Detection using Open CV – Python
International Journal of Computer Applications (0975 – 8887) Volume 162 – No 8, March 2017

Appendix Project Code

```
import cv2
import numpy as np
#Open Webcam
cap=cv2.VideoCapture(0)
while(True):
    #Reading Image per Frame
    _, img = cap.read()

    #Converting RGB color to HSV color
    hsv=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)

    #Taking Limits for which color need to be identified

    #Red Color
    red_lower=np.array([0,100,100],np.uint8)
    red_upper=np.array([10,255,255],np.uint8)

    #Blue Color
    blue_lower=np.array([99,115,150],np.uint8)
    blue_upper=np.array([110,255,255],np.uint8)

    #Yellow Color
    yellow_lower=np.array([20,100,100],np.uint8)
    yellow_upper=np.array([40,255,255],np.uint8)

    #Green
    green_lower=np.array([50,100,100],np.uint8)
    green_upper=np.array([70,255,255],np.uint8)

    #Brown
    brown_lower=np.array([0,70,60],np.uint8)
    brown_upper=np.array([10,255,255],np.uint8)

    #Orange
    orange_lower=np.array([4,100,70],np.uint8)
    orange_upper=np.array([48.32,255,255],np.uint8)

    #Purple
    purple_lower=np.array([290,100,50.2],np.uint8)
    purple_upper=np.array([310,255,255],np.uint8)

    #Pink
    pink_lower=np.array([300,80,88],np.uint8)
    pink_upper=np.array([337,255,255],np.uint8)

    #Cyan
    cyan_lower=np.array([170,100,100],np.uint8)
    cyan_upper=np.array([190,255,255],np.uint8)
```

```
#Magenta
magenta_lower=np.array([290,100,100],np.uint8)
magenta_upper=np.array([310,255,255],np.uint8)

#Taking values of each color in its respective name
red=cv2.inRange(hsv, red_lower, red_upper)
blue=cv2.inRange(hsv,blue_lower,blue_upper)
yellow=cv2.inRange(hsv,yellow_lower,yellow_upper)
green=cv2.inRange(hsv,green_lower,green_upper)
brown=cv2.inRange(hsv,brown_lower,brown_upper)
orange=cv2.inRange(hsv,orange_lower,orange_upper)
purple=cv2.inRange(hsv,purple_lower,purple_upper)
pink=cv2.inRange(hsv,pink_lower,pink_upper)
cyan=cv2.inRange(hsv,cyan_lower,cyan_upper)
magenta=cv2.inRange(hsv,magenta_lower,magenta_upper)

#For making a sliding window of 5X5 Matrix
kernal = np.ones((5 ,5), "uint8")

#To expand image from actual size after removal of noise and storing in its respective color name
red=cv2.dilate(red, kernal)
res=cv2.bitwise_and(img, img, mask = red)

blue=cv2.dilate(blue,kernal)
res1=cv2.bitwise_and(img, img, mask = blue)

yellow=cv2.dilate(yellow,kernal)
res2=cv2.bitwise_and(img, img, mask = yellow)

green=cv2.dilate(green,kernal)
res3=cv2.bitwise_and(img, img, mask = green)

brown=cv2.dilate(brown,kernal)
res4=cv2.bitwise_and(img, img, mask = brown)

orange=cv2.dilate(orange,kernal)
res5=cv2.bitwise_and(img,img, mask=orange)

purple=cv2.dilate(purple,kernal)
res6=cv2.bitwise_and(img, img, mask = purple)

pink=cv2.dilate(pink,kernal)
res7=cv2.bitwise_and(img, img, mask = pink)

cyan=cv2.dilate(cyan,kernal)
res8=cv2.bitwise_and(img, img, mask = cyan)

magenta=cv2.dilate(magenta,kernal)
res9=cv2.bitwise_and(img, img, mask = magenta)

#Checking for its neighbor color pixel by pixel value
(_,contours,hierarchy)=cv2.findContours(red,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)

#Creating Frame for Red Color and writing its respective code
```

```
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area>300):
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
            cv2.putText(img,"RED
#FF0000",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 0.7, (0,0,255))

    #Creating Frame for Blue Color and writing its respective code
    (_,contours,hierarchy)=cv2.findContours(blue,cv2.RETR_TREE,cv2.CHAIN_APP
ROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area>300):
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
            cv2.putText(img,"BLUE
#0000FF",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 0.7, (255,0,0))

    #Creating Frame for Yellow Color and writing its respective code
    (_,contours,hierarchy)=cv2.findContours(yellow,cv2.RETR_TREE,cv2.CHAIN_AP
PROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area>300):
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,255),2)
            cv2.putText(img,"YELLOW
#FFFF00",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (0,255,255))

    #Creating Frame for Green Color and writing its respective code
    (_,contours,hierarchy)=cv2.findContours(green,cv2.RETR_TREE,cv2.CHAIN_AP
PROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area>300):
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
            cv2.putText(img,"GREEN
#08000",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (0,255,0))

    #Creating Frame for Brown Color and writing its respective code
   #(_,contours,hierarchy)=cv2.findContours(brown,cv2.RETR_TREE,cv2.CHAIN_AP
PROX_SIMPLE)
   #for pic, contour in enumerate(contours):
    #   area = cv2.contourArea(contour)
     #  if(area>300):
      #     x,y,w,h = cv2.boundingRect(contour)
       #    img = cv2.rectangle(img,(x,y),(x+w,y+h),(42,42,165),2)
        #   cv2.putText(img,"BROWN
#A52A2A",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (42,42,165))

    #Creating Frame for Orange Color and writing its respective code
   #
```

```
(_,contours,hierarchy)=cv2.findContours(orange,cv2.RETR_TREE,cv2.CHAIN_APP
ROX_SIMPLE)
    #for pic, contour in enumerate(contours):
     #   area = cv2.contourArea(contour)
      #  if(area>300):
       #     x,y,w,h = cv2.boundingRect(contour)
        #    img = cv2.rectangle(img,(x,y),(x+w,y+h),(0,165,255),2)
         #   cv2.putText(img,"ORANGE
#FFA500",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (0,165,255))

    #Creating Frame for Purple Color and writing its respective code
    (_,contours,hierarchy)=cv2.findContours(purple,cv2.RETR_TREE,cv2.CHAIN_AP
PROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area>300):
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(128,0,128),2)
            cv2.putText(img,"PURPLE
#800080",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (128,0,128))

   #
(_,contours,hierarchy)=cv2.findContours(pink,cv2.RETR_TREE,cv2.CHAIN_APPRO
X_SIMPLE)
   # for pic, contour in enumerate(contours):
    #    area = cv2.contourArea(contour)
     #   if(area>300):
      #      x,y,w,h = cv2.boundingRect(contour)
       #     img = cv2.rectangle(img,(x,y),(x+w,y+h),(147,20,255),2)
        #    cv2.putText(img,"PINK
#FFC0CB",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (147,20,255))

    #Creating Frame for Cyan Color and writing its respective code
   #
(_,contours,hierarchy)=cv2.findContours(cyan,cv2.RETR_TREE,cv2.CHAIN_APPRO
X_SIMPLE)
    #for pic, contour in enumerate(contours):
     #   area = cv2.contourArea(contour)
      #  if(area>300):
       #     x,y,w,h = cv2.boundingRect(contour)
         #   img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)
         #   cv2.putText(img,"CYAN
#00FFFF",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (255,255,0))

    #Creating Frame for Magenta Color and writing its respective code
    (_,contours,hierarchy)=cv2.findContours(magenta,cv2.RETR_TREE,cv2.CHAIN_
APPROX_SIMPLE)
    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area>300):
            x,y,w,h = cv2.boundingRect(contour)
            img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,255),2)
            cv2.putText(img,"MAGENTA
#FF00FF",(x,y),cv2.FONT_HERSHEY_COMPLEX_SMALL, 1.0, (255,0,255))
```

```
#Displaying Frame of Image
cv2.imshow("Tracking of COLORS",img)

#Checking for termination
if cv2.waitKey(5) & 0xFF == ord('q'):
    cap.release()
    cv2.destroyAllWindows()
    break
```