NEXUS

WEB DEVELOPER INTERN TASK-3

PROBLEM STATEMENT
HELPING AID FOR COLOUR BLINDNESS PEOPLE

Submitted By:

Srinivasan P
Srinivasanpalani2003@gmail.com
9345308520

ABSTRACT

People suffering from color-blindness suffer from the problem of viewing some colors or differentiate between several color shades. The person with colorblindness faces many difficulties such as traffic signal cannot be well understood, they face difficulty to interpret colorful graphs and charts, they also cannot enjoy videos and sports matches. It limits people for viewing true colours with their naked eye. A suffering person faces various challenges in everyday life from identifying the different colour, choosing clothes or using colour dyes for concoction. 1 in 200 women and 1 in 12 men are suffering with tis deficiency. It is a genetic disorder and also it may appear after some years of grown-up. A colour-blind person faces problem in distinguishing the right colour in such case. Similarly, colour-blind struggles to find a software application suited to overcome their colour impairment. So, we propose a vision aid with improved user experience to use voice command. By our system, user can get the colour of an object by requesting the inbuilt assistance. The system processes the video and finds the object and notifies the colour to the user through voice. Our system improves the vision of the colour vision deficiency people to look the world better. Even if it is not curable, at least it extends the hand for the people to enjoy the sights.

KEYWORDS: Color-blindness, Color Vision Deficiency, Genetic disorder, Software application, Voice assistance.

INTRODUCTION

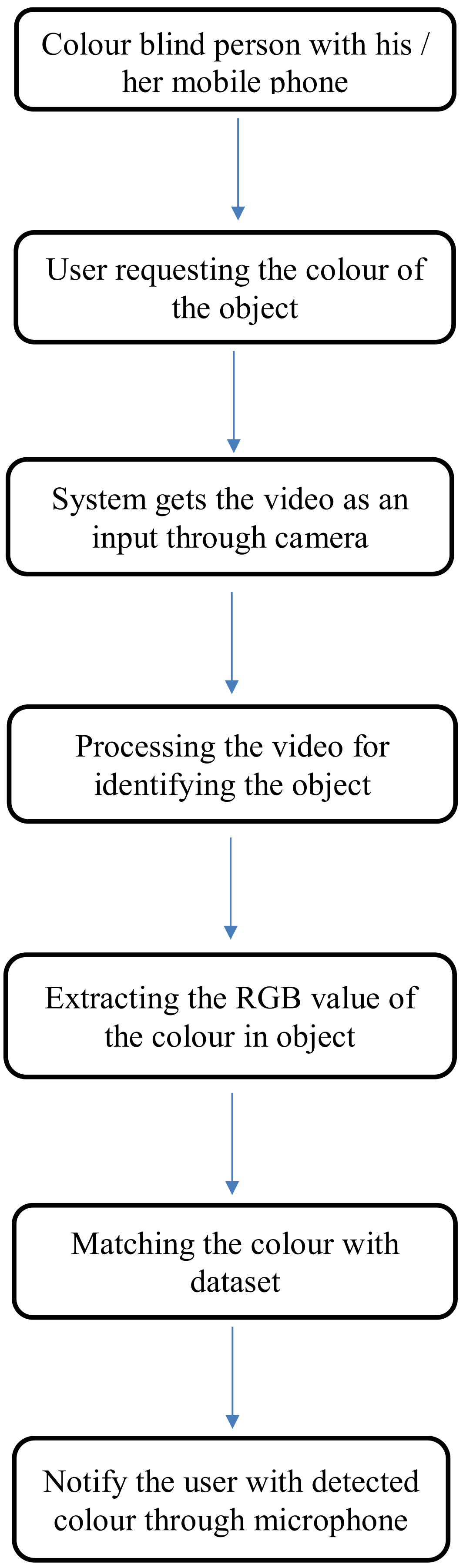
Colorblindness, formally known as color vision deficiency, is referred to as those who have difficulties in discriminating certain color combinations and color differences. There are about 8% of men and 0.8% of women suffering from different types of colorblindness. Existing studies reveal that colors are perceived by humans with their cones absorbing photons and sending electrical signal to the brains. The cones are categorized into Long (L), Middle (M) and Short (S), which absorb long wavelengths, medium wavelengths and short wavelengths, respectively.

Protanopia and deuteranopia have difficulty in discriminating red from green, whereas tritanopia have difficulty in discriminating blue from yellow. Due to the loss of color information, many visual objects (such as images and videos) that have high qualities in the eyes of normal viewers may not be well perceived by colorblind viewers. Several research works have been dedicated to helping colorblind users in better perceiving visual objects. The most straightforward approach is to re-color the objects, i.e., adopt a mapping function to change the colors of the original images, such that the colors that are mixed up by the colorblind users can be discriminated after re-coloring.

Mobile-phones have become invaluable devices frequently used to meet the user’s needs according to their requirements. Currently, Android (Google) and iPhone (Apple) operating systems embed Color Vision Deficiency (CVD) functions for supporting color-blind people. However, usability problems of complexity and flexibility still exist in the User Interfaces (UIs) of many apps.

Even though, many systems are currently evolving, but they are not affordable in low cost. To reduce this inconvenience, our system will act as an extra hand assistance for the people with colour vision deficiency.

ARCHITECTURAL DESIGN


Colour blind person with his / her mobile phone
User requesting the colour of the object
System gets the video as an input through camera
Processing the video for identifying the object
Extracting the RGB value of the colour in object
Matching the colour with dataset
Notify the user with detected colour through microphone

PROPOSED SYSTEM

The proposing system is a mobile application that comprises of the several components. They are,

- Camera accessing,
- Processing the image of object,
- Detecting the colour of an object and
- Notify the colour to user.

Firstly, the application accesses the camera and the user shows or requests the object to know its colour. The system gets the image of an object through the mobile camera and sends it to the next process.

Secondly, the received image has to be processed to know what is the object in the frame. It uses the Yolo (You Only Look Once) algorithm to detect the object. For this process, a dataset from the Kaggle.com is used to train the system for efficiently detecting the object.

And the detected object is checked with the requested object. If they are same, then the colour of the object is detected using the RGB values in the pixels of the object image. The resultant RGB value is compared with the colour dataset (.csv file) which contains the combination of 3000 colour variants with their respective hue values.

Finally, the detected colour of the object is notified to the user through the microphone. To achieve this, the user has to grant permission to access the microphone and the camera.

ALGORITHM

The algorithm to be used in the project is YOLO (You Only Look Once).

YOLO is an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It is used for real time object detection. It is helpful because of its fastness in detecting the object.

It is a deep learning based method to achieve the object detection. YOLO mechanism works by separating the image into S x S grids. Each grid is known as cells. If there is an object in the cell, then it marks it as 1. If not, then 0.

The output of the neural network with the vectors is [P_C, B_X, B_Y, B_W, B_H, C], where,

P_C – Probability of the class
B_X – Boundary values of x-axis
B_Y – Boundary value of y-axis
B_W – Boundary width
B_H – Boundary height
C - Classes

YOLO has number of versions like YOLOV1, YOLOV2, YOLOV3, YOLOV4, YOLOV5, YOLOV6, YOLOV7. The complexity formula of the algorithms is,

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

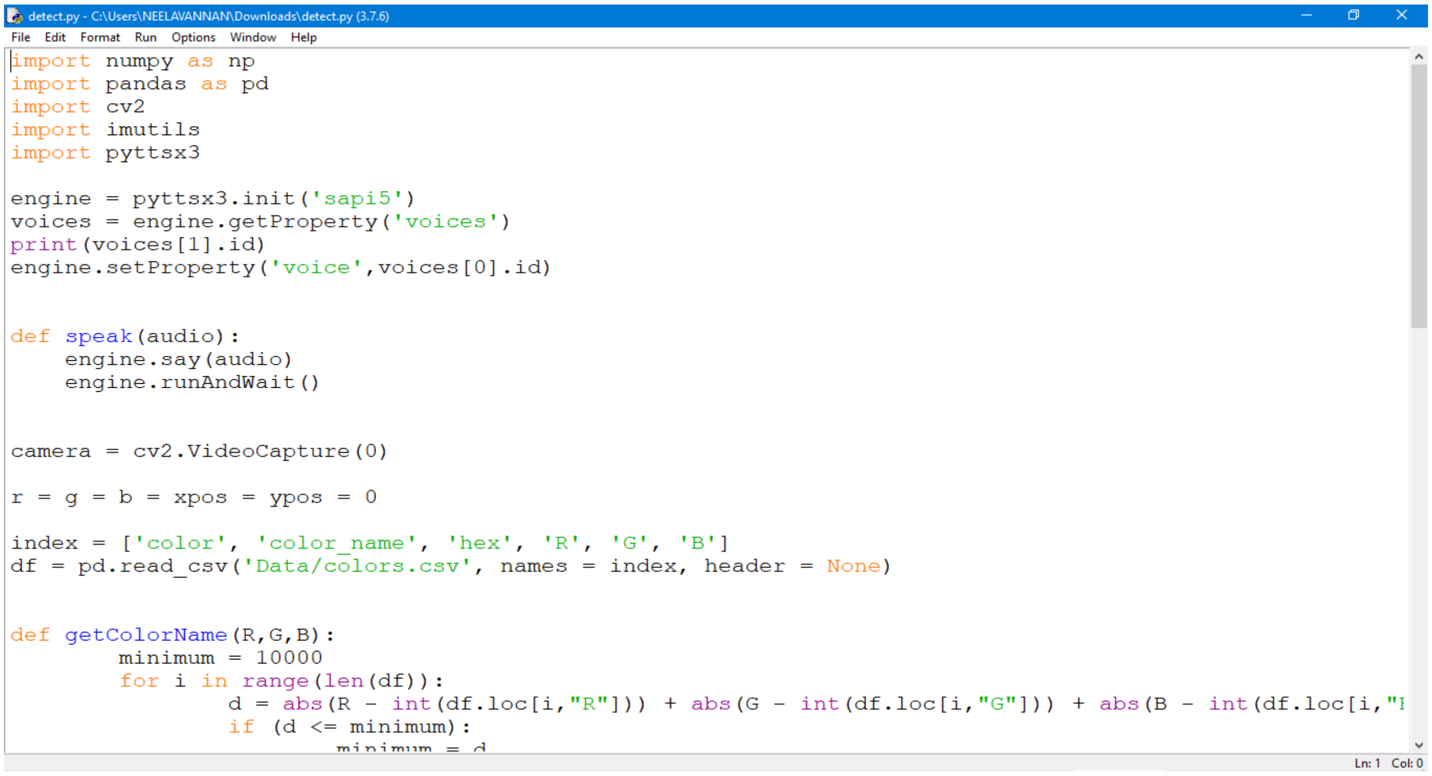
$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

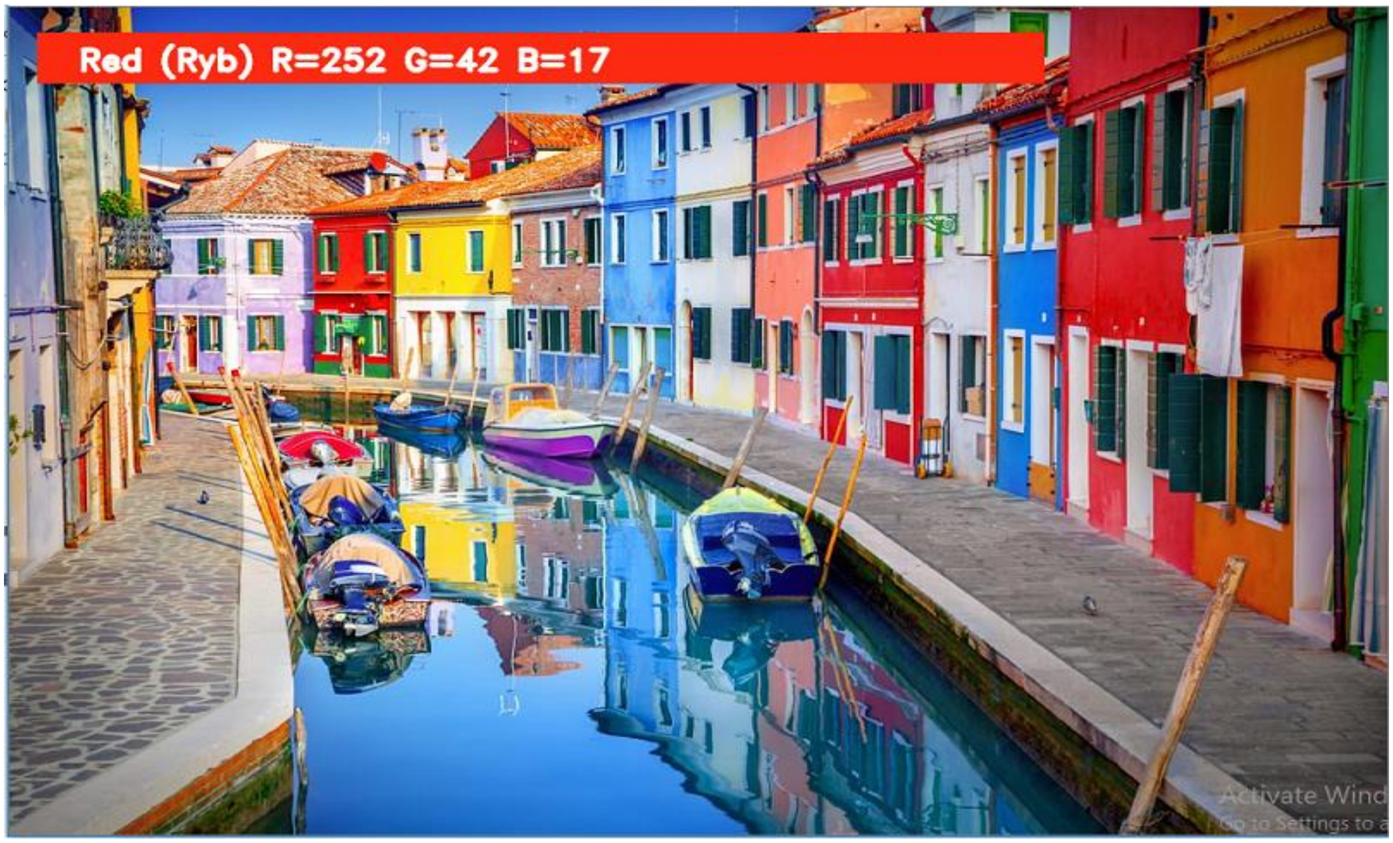
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

CODE IMPLEMENTATION:


detect.py - C:\Users\NEELAVANNAN\Downloads\detect.py (3.7.6)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```python
import numpy as np
import pandas as pd
import cv2
import imutils
import pyttsx3

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
print(voices[1].id)
engine.setProperty('voice',voices[0].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


camera = cv2.VideoCapture(0)

r = g = b = xpos = ypos = 0

index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
df = pd.read_csv('Data/colors.csv', names = index, header = None)


def getColorName(R,G,B):
        minimum = 10000
        for i in range(len(df)):
                d = abs(R - int(df.loc[i,"R"])) + abs(G - int(df.loc[i,"G"])) + abs(B - int(df.loc[i,"
                if (d <= minimum):
                        minimum = d
```

EXPECTED OUTCOMES


Red (Ryb) R=252 G=42 B=17