COLOUR DETECTION

A MINOR PROJECT REPORT

Submitted by

Ambuj Porwal
[RA1911003010236]
Sajal Tyagi
[RA1911003010238]

Under the guidance of

Dr. R Rajkamal

(Associate professor)

Dept. of Computing Technologies

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in
COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M.Nagar, Kattankulathur, Chengalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that 18CSP107L minor project report titled "**COLOUR DETECTION**" is the bonafide work of "**Ambuj Porwal[RA1911003010236],Sajal Tyagi[RA1911003010238]"** who carried out the minor project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. R Rajkamal
GUIDE
Associate
Professor
Dept. of
Computing
Technologies

SIGNATURE

Dr. M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor
Dept. of Computing Technologies

Signature of the Panel Head
Dr. R.Rajkamal
Associate Professor, Dept. of Computing
Technology

ABSTRACT

Color detection is essential for color recognition and is a technique utilized in many image-editing and sketching programs. It is the procedure for identifying any color's name. Well, this is a really simple assignment for people, but it is more complicated for computers. Light is converted into color by the eyes and brains of living things. Our eyes have light receptors that send the signal to the brain. The colour is then recognised by our brain. We must therefore address the issue of how colour blind individuals perceive colours because they are unable to identify the proper hue. So basically, there are three separate components that we need to use in this Python project. A.csv file holding the colours as a dataset, a piece of Python code that will be used to recognise colour, and an image that will be used to test the colour recognition. Thus, the three modules mentioned above will enable us to accomplish our goal of using Python to identify the colours in an image. This study demonstrates how colour detection, which involves identifying colours, is put into practise. Humans can easily identify and select a colour in this situation. By clicking on them, you will instantly learn the name of the hue through the application we are creating. We will therefore have a data file for this that contains the name of the colour and its values. Then, we will measure the distances between each colour and determine which is the shortest. This entire process will be very helpful for those who are colour blind.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS:

ML	Machine Learning
IM	Image processing
OC	OpenCv
GPY	Gaussian Pyramid
LAP	Laplacian Pyramid

CHAPTER 1

INTRODUCTION

1.1 Problem Statement:

How to obtain find the name of the colour for those who are colour blind. If the image is not clean, it should be cleaned as much as possible before the colour is detected.

OBJECTIVES 1.1.1

- Create a CSV file and enter the names of the colours together with their RGB values.
- Take a photo as input and use a CSV file to find the colour that most closely matches it.
- Web-cam mode with that colour is available in live mode.
- Using algorithms to determine the optimum outcome.
- Cleaning up the image using MATLAB image processing.
- Following pixel cleaning, colour can be applied.

Code explanation:

Camera Settings: Runtime actions are carried out via the computer's webcam.

- A VideoCapture object must be made in order to capture the video.
-

Either a device index or a video file name may be used as its argument.

- The device's index is just a number used to specify which camera is being used.
- Since there is typically only one camera connected, we will just cross 0.
- By pressing 1, 2, and so on, you can choose the alternative camera.
- After that, you can take frames one at a time and then publish the capture.

Furthermore, with a few adjustments to the code, which we will cover later, anyone may use this colour detecting technique on any image.

- **Capturing frames**: To guarantee that the camera records photographs in every situation and is open for the length of the programme, an infinite loop is utilised.

- We convert each frame from the default BGR colour space to the HSV colour space after frame-by-frame recording of the live stream.

- OpenCV offers more than 150 different colour space conversion techniques. However, we will just examine BGR to Gray and BGR to HSV, the two most frequently utilised. The method cv2.cvtColor(input image, flag) is used to convert an image's colour, and the flag specifies the conversion's parameters. The cv2.COLOR BGR2HSV setting is used to convert BGR to HSV.

We can use it to extract the colour object now that we know how to convert BGR photos to HSV. HSV colour space makes it simpler to represent colour than RGB.

- We entered the blue hue range as we entered the range. While you can choose whatever hue you choose for the colour range.

- The use of a mask: A mask basically draws a certain portion of the image in accordance with predetermined rules.

- In this case, we design a mask using a blue-colored object.

- After that, I applied bitwise and to the input image and the thresholded image to highlight and save in resolution only the objects that were blue in colour.

- The imshow function is then used to display the frame, resolution, and mask in 3 different windows.

- Frame Display: waitKey must be called on a recurring basis to manage imshow(event)'s loop because it is a HighGui function.

- A "delay" key event is awaited by the waitKey() function (here 5 milliseconds).

- HighGui cannot respond to window events like redraw, resize, input event, etc. if waitKey is not called. So, even with a 1 ms delay, just call it.

Process Synopsis:

1. Take a screenshot of each video frame.
2. Change each frame's colour space from BGR to HSV.
3. Select the blue range's HSV image threshold value.

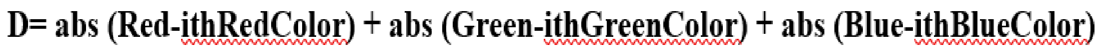
- The software must have a function that returns the colour of the mouse pointer's current location.

- And as we are aware, in order to obtain the colour, we must first obtain the distance and compare it to the dataset.

Distance calculation:

As we've already demonstrated in the screenshot, the R, G, and B values of the colours we've associated to the dataset are available.

The following formulas are used to calculate distance:



$$D= \text{abs (Red-ithRedColor)} + \text{abs (Green-ithGreenColor)} + \text{abs (Blue-ithBlueColor)}$$

where ithRedColor, ithGreenColor, and ithBlueColor are the colour R, G, and B values from the dataset.

Here value from dataset are marked with   red.

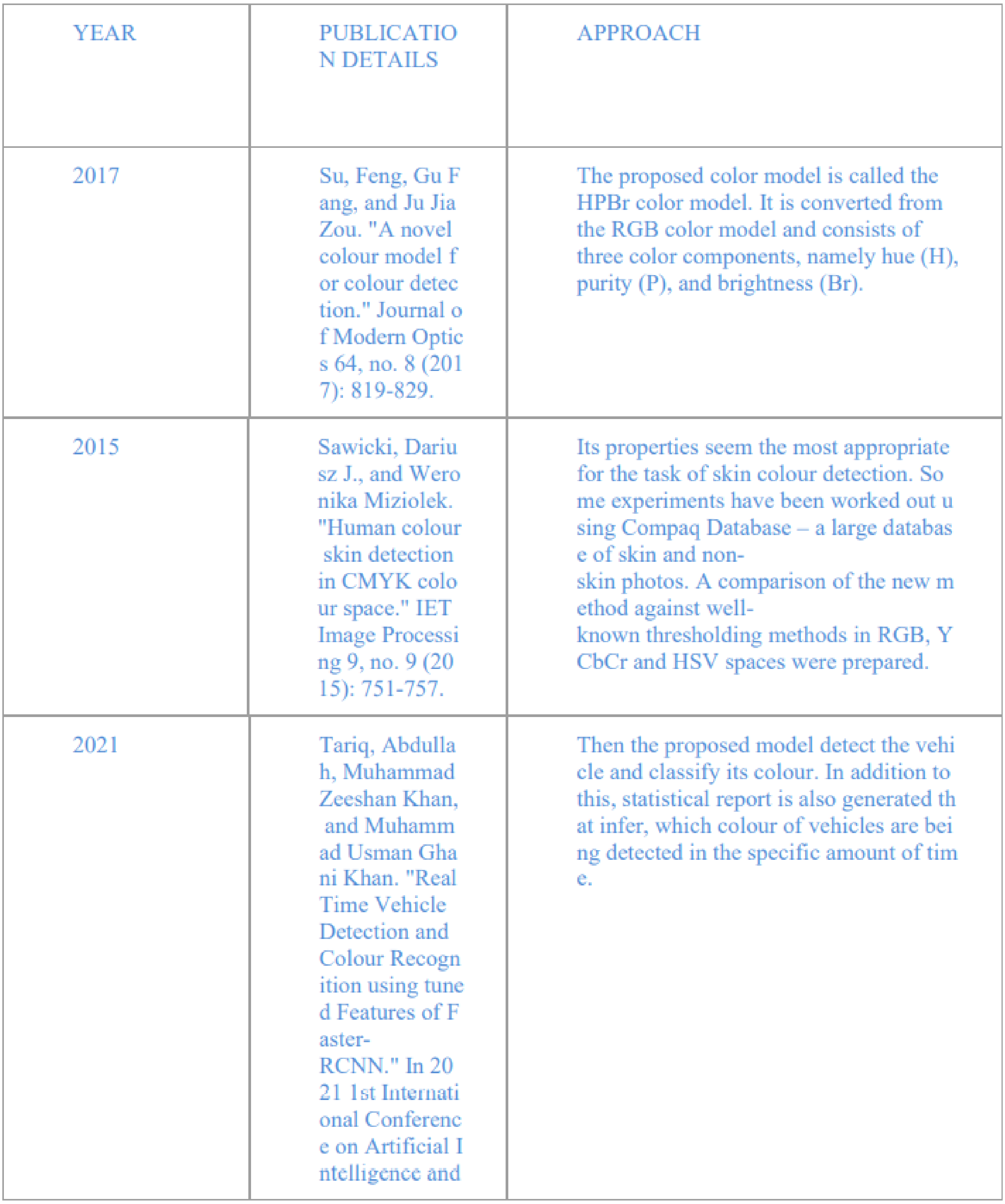
1.1.2 SOFTWARE REQUIREMENT SPECIFICATION:

- A Windows, Mac, Linux OS installed on a computer.
- Anaconda, VS Code installation.
- A latest version python ide.
- A MATLAB software installed.

CHAPTER 2

LITERATURE SURVEY

2.1 Paper Title & Author



YEAR	PUBLICATION DETAILS	APPROACH
2017	Su, Feng, Gu Fang, and Ju Jia Zou. "A novel colour model for colour detection." Journal of Modern Optics 64, no. 8 (2017): 819-829.	The proposed color model is called the HPBr color model. It is converted from the RGB color model and consists of three color components, namely hue (H), purity (P), and brightness (Br).
2015	Sawicki, Dariusz J., and Weronika Miziolek. "Human colour skin detection in CMYK colour space." IET Image Processing 9, no. 9 (2015): 751-757.	Its properties seem the most appropriate for the task of skin colour detection. Some experiments have been worked out using Compaq Database – a large database of skin and non-skin photos. A comparison of the new method against well-known thresholding methods in RGB, YCbCr and HSV spaces were prepared.
2021	Tariq, Abdullah, Muhammad Zeeshan Khan, and Muhammad Usman Ghani Khan. "Real Time Vehicle Detection and Colour Recognition using tuned Features of Faster-RCNN." In 2021 1st International Conference on Artificial Intelligence and	Then the proposed model detect the vehicle and classify its colour. In addition to this, statistical report is also generated that infer, which colour of vehicles are being detected in the specific amount of time.

	Data Analytics (CAIDA), pp. 262-267. IEEE, 2021.	
1994	Matas, Jirí, Radek Marík, and Josef Kittler. "Illumination Invariant Colour Recognition." In BMVC, vol. 94, pp. 469-479. 1994.	A colour based recognition system with three novel features. Firstly, the proposed system can operate in environments with spectrally uneven and changing illumination. Secondly, benefits in terms of speed and quality of output are gained by focusing processing to areas of salient colour. Finally, an automatic model acquisition procedure has been implemented.
2015	B. R. Navada, K. V. Santhosh, S. Prajwal and H. B. Shetty, "An image processing technique for color detection and distinguish patterns with similar color: An aid for color blind people," International Conference on Circuits, Communication, Control and Computing, 2014, pp. 333-336, doi: 10.1109/CIMCA.2014.7057818.	Image Processing Technique is used in proposed work to develop a setup for identifying the colours in an image and indicating the edges in case of different images with similar colours. Image processing involves two parts, first is colour detection and second is edge detection.

Table 2.1

2.2 Comparison of Merits and Demerits:

Merits :

1. We can help the person suffering from Color Blindness without physically operating or doing anything to the person's eyes.
2. Color Detection will help people in sorting of objects based on three color approach which are Red, Blue and Green as we are considering RGB values of color.

Demerits :

1. Color Detection approach is very costly, so due to that reason it cannot be afforded by small scale industries.

2. It does color matching or identification in applications requiring only true and false output.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 UML Diagram

A UML diagram helps us software programmers in understanding and visualizing the design and behavior of an application.

3.1.1 ARCHITECTURAL DIAGRAM

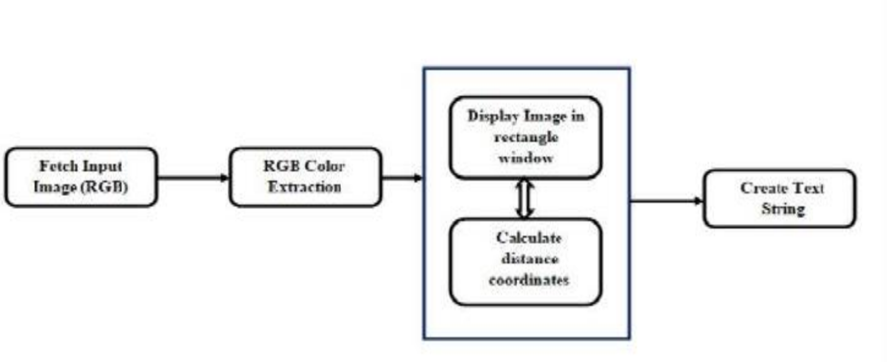

Fetch Input Image (RGB)
RGB Color Extraction
Display Image in rectangle window
Calculate distance coordinates
Create Text String

Fig 3.1

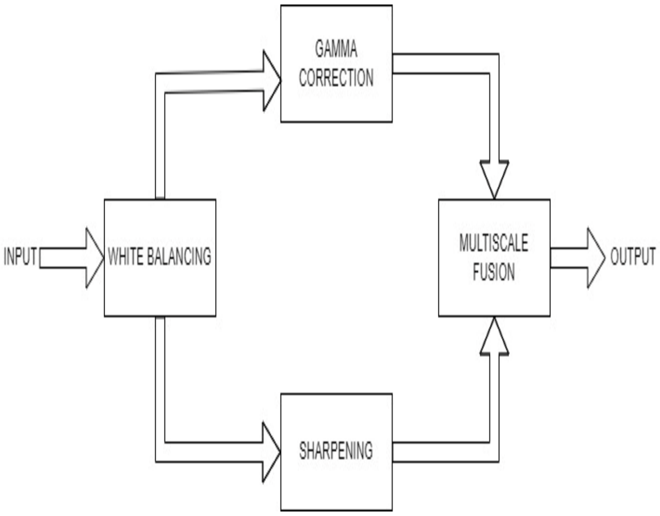


Fig 3.2

3.1.2 BLOCK DIAGRAM

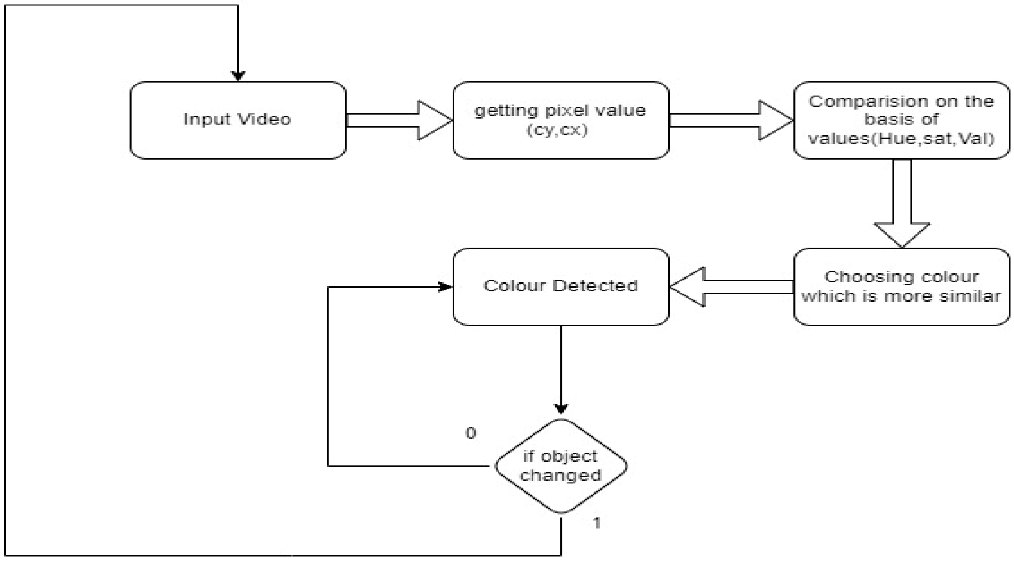

Input Video
getting pixel value (cy,cx)
Comparision on the basis of values(Hue,sat,Val)
Choosing colour which is more similar
Colour Detected
0
if object changed
1

Fig 3.3

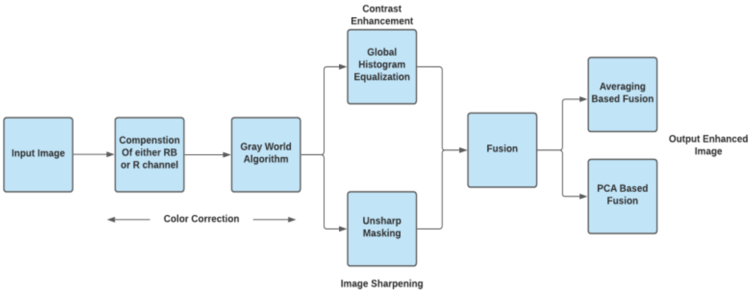


Fig 3.4

CHAPTER 4

METHODOLOGY

4.1 Machine Learning

The machine learning of the project deals with the following points:

- NAÏVE BAYES
- K MEANS

4.1.1 NAÏVE BAYES

The Nave Bayes algorithm is a supervised learning method for classification issues that is based on the Bayes theorem.

1. It is mostly employed in text categorization with a large training set.

2. The Naive Bayes Classifier is one of the most straightforward and efficient Classification algorithms, aiding in the development of quick machine learning models capable of making prompt predictions.

3. It makes predictions based on an object's likelihood because it is a probabilistic classifier.

4. Classifying articles, sentiment analysis, and spam filtration are a few examples of Nave Bayes algorithms that are frequently used.

4.1.2 K MEANS

1. K-Means Clustering is a type of unsupervised learning algorithm that is used to address clustering issues in data science or machine learning.

2. 1. It provides a straightforward technique to categorise the groups in the unlabeled dataset on its own without the requirement for training. It also helps us to cluster the data into distinct groups.

3. 2. This algorithm's primary goal is to reduce the total distances between each data point and its matching clusters.
4. 3. The algorithm separates the unlabeled dataset into k clusters and then repeats the procedure until it is unable to discover the best clusters..

Required tools:

OpenCV:

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

Pandas:

Pandas is Python Package which stands for Python and data analysis. This library helps in data manipulation and analysis.

4.2 IMAGE PROCESSING

When colour detection functions properly, the issue is an unclear input image (reddish,bluish). In order to discover the colour of any point on the image, image processing is employed in this project to make the image more possibly clear. The following methodology is employed for this process:

4.2.1 GAUSSIAN PYRAMID

The Gaussian pyramid is an image-processing method that blurs an image by dividing it into progressively smaller groupings of pixels. It bears the name Johann Carl

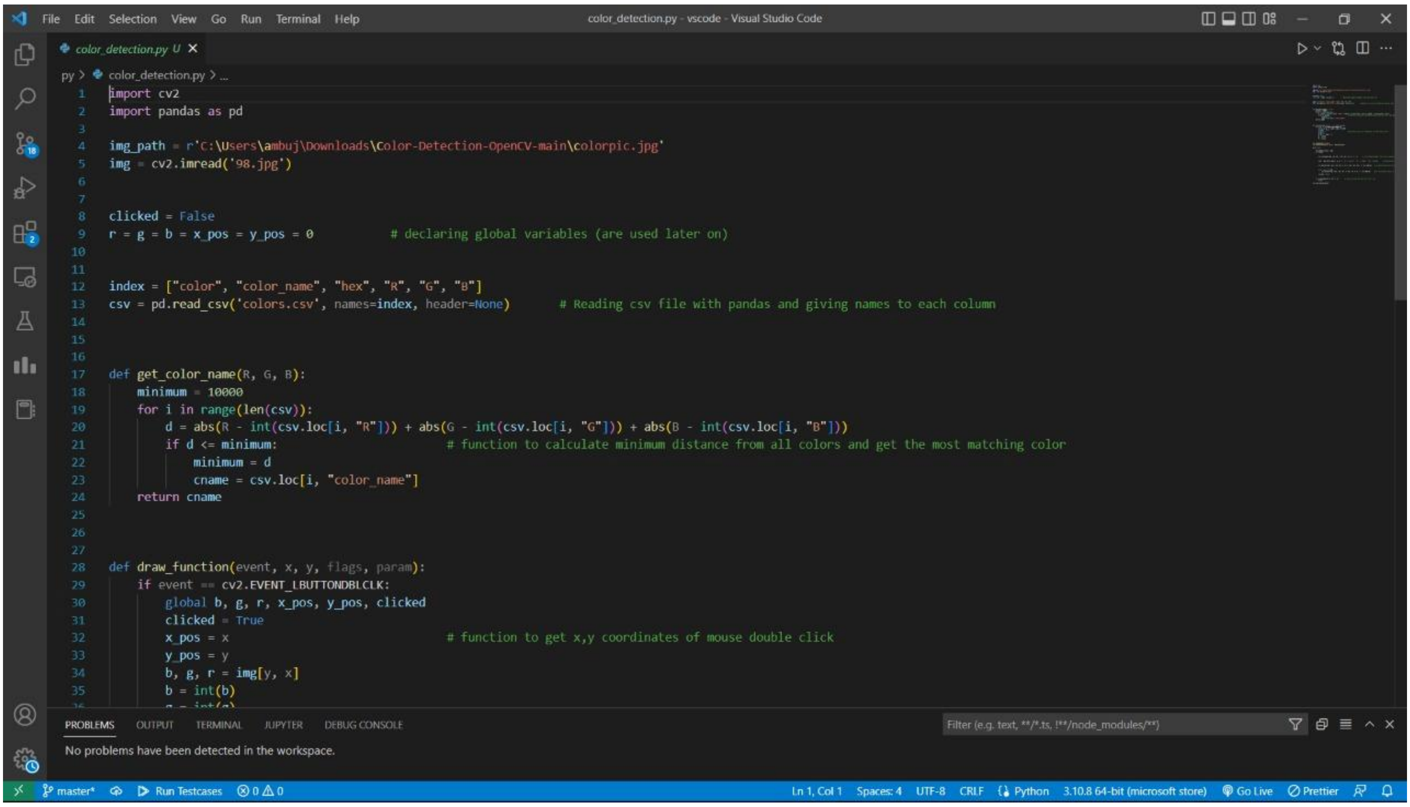
Friederich Gauss, a German mathematician. Artificially intelligent computer vision heavily on this kind of exact mathematical blurring as a pre-processing phase. For instance, this type of blurring makes it easier to distinguish the edges of things in digital photographs, allowing a computer to recognise such items automatically.

4.2.2 LAPLACIAN PYRAMID

The Gaussian pyramid (GP), a multiresolution picture representation created through a recursive reduction (low pass filtering and decimation) of the image data set, is a bandpass image decomposition that was first proposed by Burt and Adelson before multiresolution wavelet analysis was developed..

CHAPTER 5

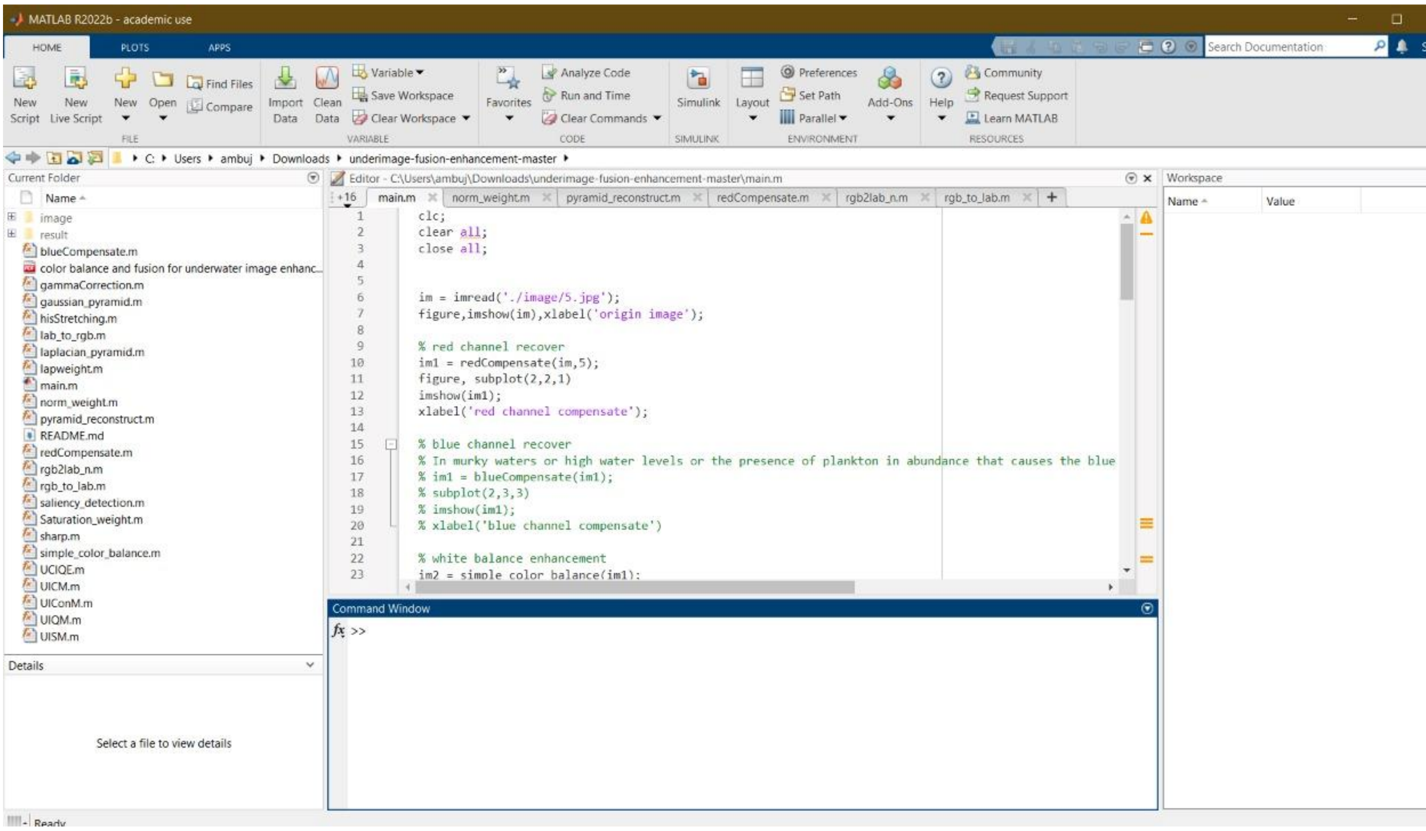
CODING AND TESTING



File Edit Selection View Go Run Terminal Help
color_detection.py - vscode - Visual Studio Code
color_detection.py U
py > color_detection.py > ...
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Filter (e.g. text, **/*.ts, !**/node_modules/**)
master* Run Testcases 0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live Prettier

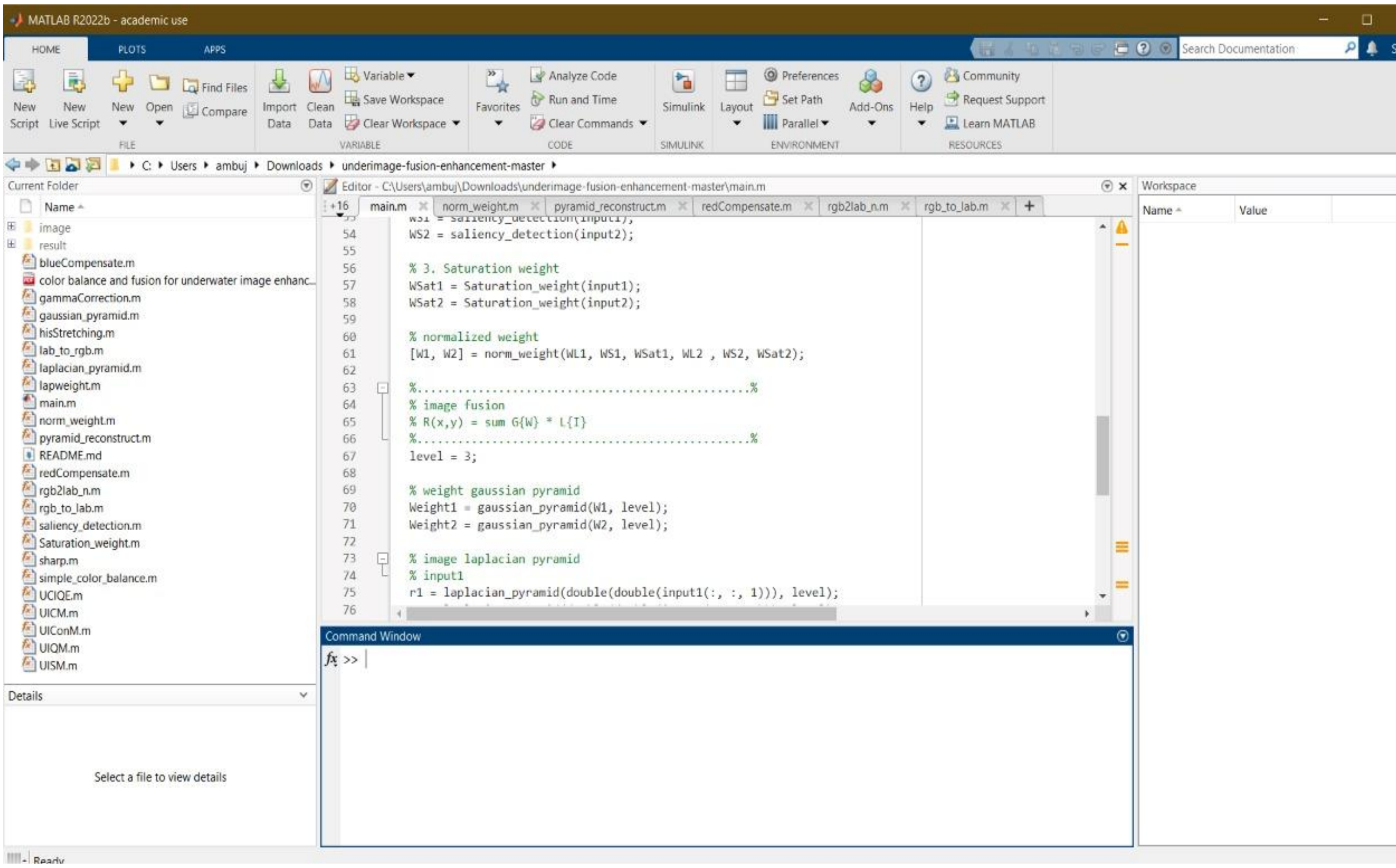
```python
import cv2
import pandas as pd

img_path = r'C:\Users\ambuj\Downloads\Color-Detection-OpenCV-main\colorpic.jpg'
img = cv2.imread('98.jpg')


clicked = False
r = g = b = x_pos = y_pos = 0           # declaring global variables (are used later on)


index = ["color", "color_name", "hex", "R", "G", "B"]
csv = pd.read_csv('colors.csv', names=index, header=None)       # Reading csv file with pandas and giving names to each column



def get_color_name(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:                        # function to calculate minimum distance from all colors and get the most matching color
            minimum = d
            cname = csv.loc[i, "color_name"]
    return cname



def draw_function(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDBLCLK:
        global b, g, r, x_pos, y_pos, clicked
        clicked = True
        x_pos = x                               # function to get x,y coordinates of mouse double click
        y_pos = y
        b, g, r = img[y, x]
        b = int(b)
```

No problems have been detected in the workspace.



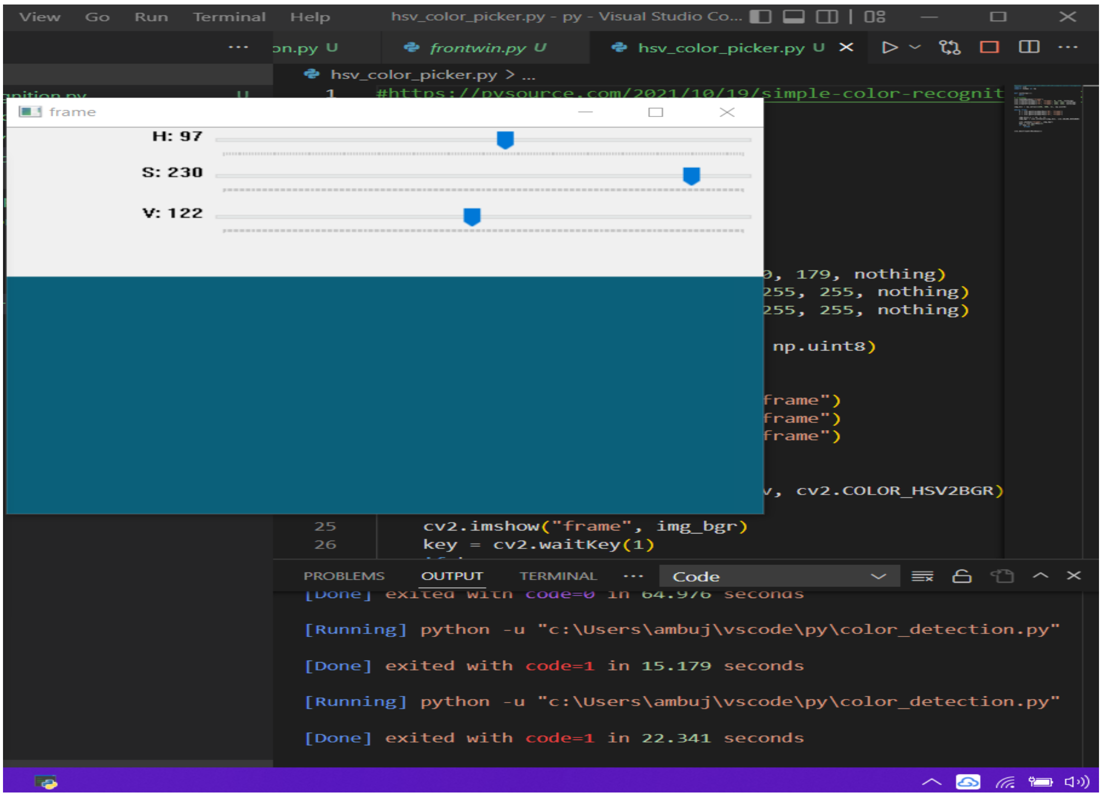
```
clc;
clear all;
close all;


im = imread('./image/5.jpg');
figure,imshow(im),xlabel('origin image');

% red channel recover
im1 = redCompensate(im,5);
figure, subplot(2,2,1)
imshow(im1);
xlabel('red channel compensate');

% blue channel recover
% In murky waters or high water levels or the presence of plankton in abundance that causes the blue
% im1 = blueCompensate(im1);
% subplot(2,3,3)
% imshow(im1);
% xlabel('blue channel compensate')

% white balance enhancement
im2 = simple_color_balance(im1);
```

```
fx >>
```



```
WS1 = saliency_detection(input1);
WS2 = saliency_detection(input2);

% 3. Saturation weight
WSat1 = Saturation_weight(input1);
WSat2 = Saturation_weight(input2);

% normalized weight
[W1, W2] = norm_weight(WL1, WS1, WSat1, WL2 , WS2, WSat2);

%.............................................%
% image fusion
% R(x,y) = sum G{W} * L{I}
%.............................................%
level = 3;

% weight gaussian pyramid
Weight1 = gaussian_pyramid(W1, level);
Weight2 = gaussian_pyramid(W2, level);

% image laplacian pyramid
% input1
r1 = laplacian_pyramid(double(double(input1(:, :, 1))), level);
```

```
fx >>
```

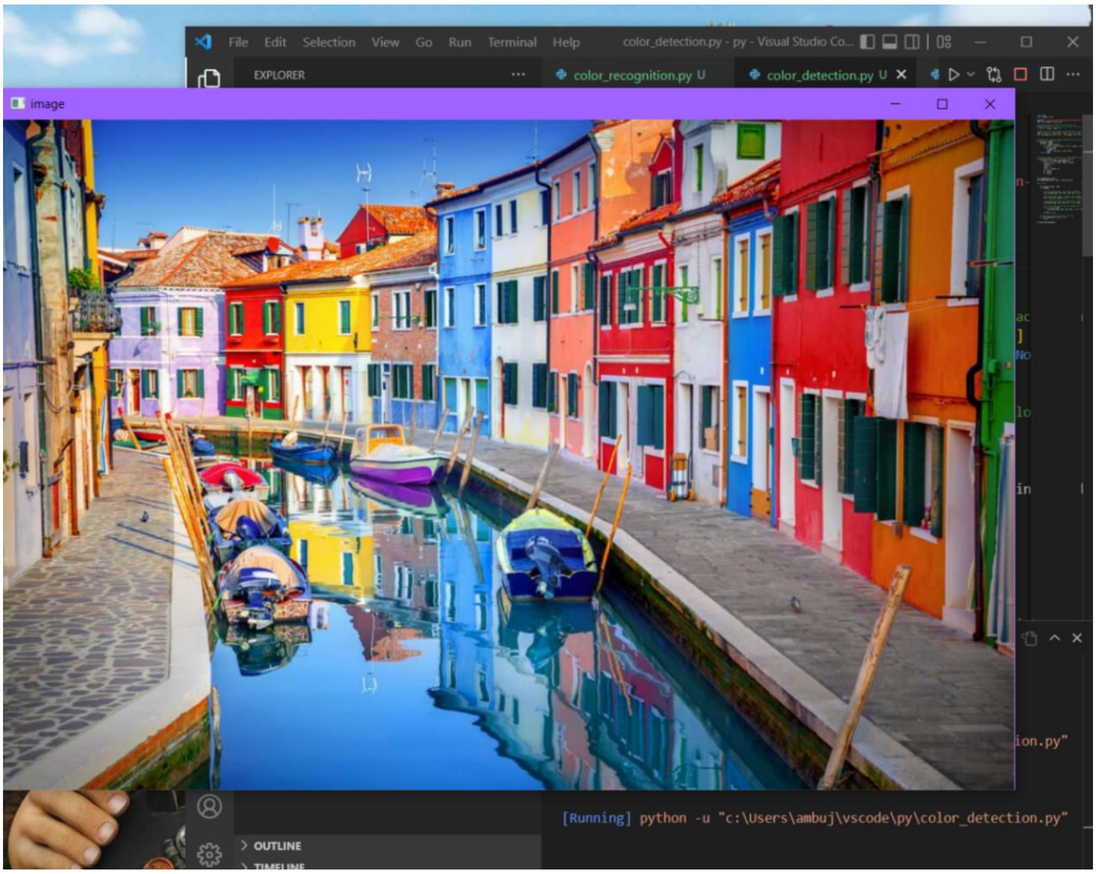


View Go Run Terminal Help
hsv_color_picker.py - py - Visual Studio Co...
on.py U
frontwin.py U
hsv_color_picker.py U
hsv_color_picker.py > ...
frame
H: 97
S: 230
V: 122
PROBLEMS
OUTPUT
TERMINAL
Code

```
#https://pysource.com/2021/10/19/simple-color-recognit
0, 179, nothing)
255, 255, nothing)
255, 255, nothing)
 np.uint8)
frame")
frame")
frame")
v, cv2.COLOR_HSV2BGR)
    cv2.imshow("frame", img_bgr)
    key = cv2.waitKey(1)
```

```
[Done] exited with code=0 in 64.976 seconds

[Running] python -u "c:\Users\ambuj\vscode\py\color_detection.py"

[Done] exited with code=1 in 15.179 seconds

[Running] python -u "c:\Users\ambuj\vscode\py\color_detection.py"

[Done] exited with code=1 in 22.341 seconds
```

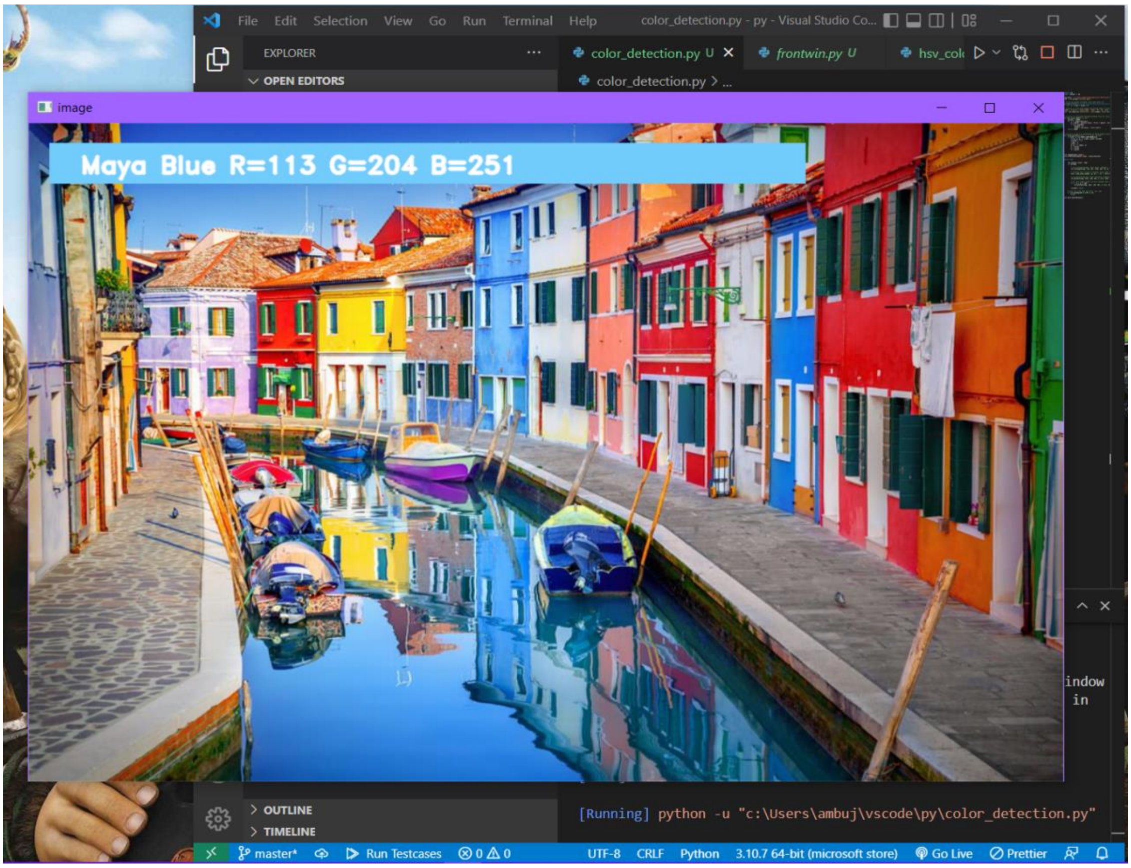
Fig5.1



File Edit Selection View Go Run Terminal Help
color_detection.py - py - Visual Studio Co...
EXPLORER
color_recognition.py U
color_detection.py U
image
OUTLINE
TIMELINE

```
[Running] python -u "c:\Users\ambuj\vscode\py\color_detection.py"
```

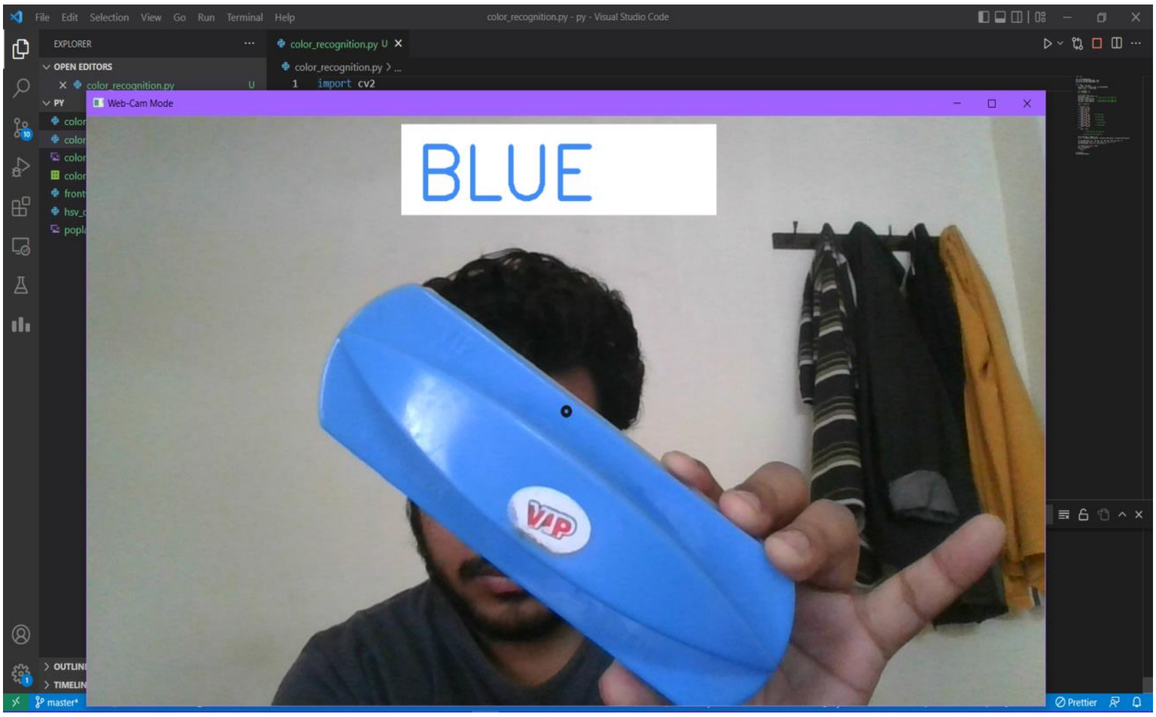
Fig 5.2



Maya Blue R=113 G=204 B=251

```
[Running] python -u "c:\Users\ambuj\vscode\py\color_detection.py"
```

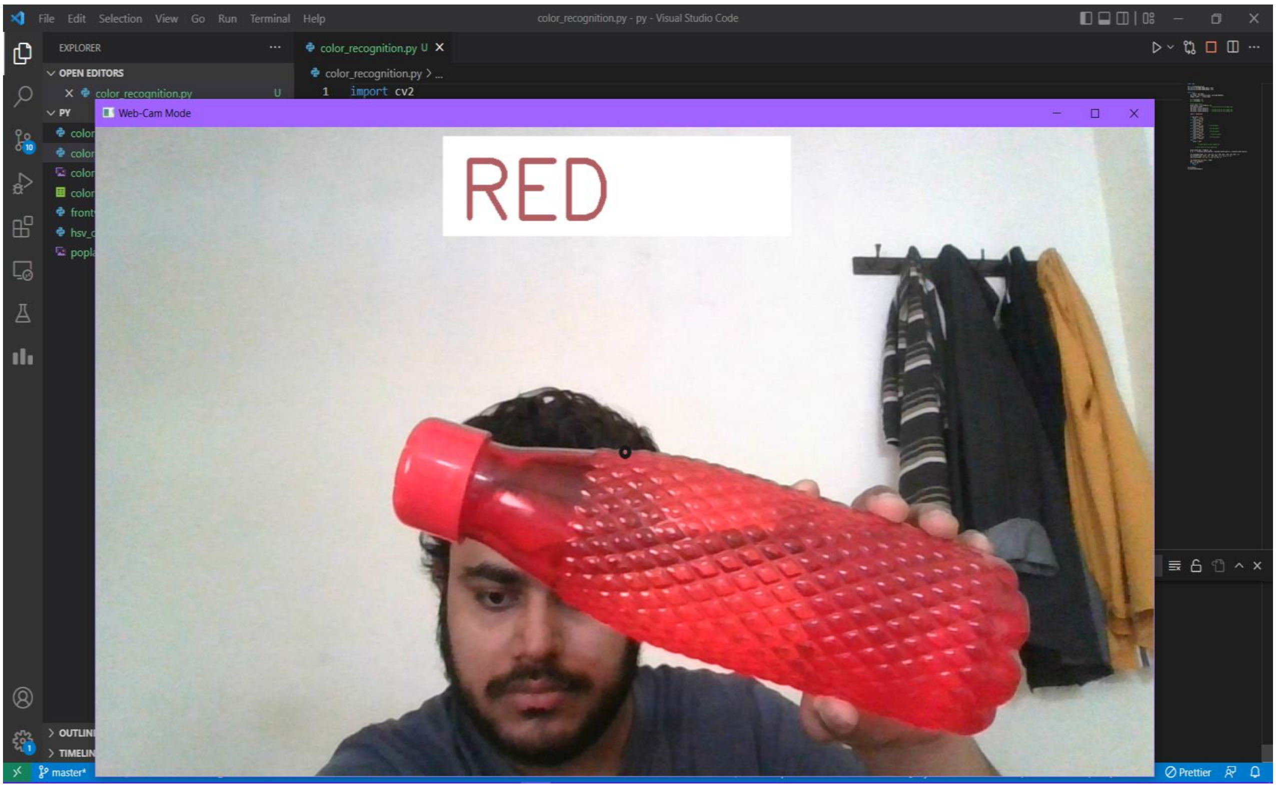
Fig 5.3


File Edit Selection View Go Run Terminal Help
color_recognition.py - py - Visual Studio Code
EXPLORER
OPEN EDITORS
color_recognition.py U
PY
color_recognition.py > ...
Web-Cam Mode
OUTLINE
TIMELINE
master*
Prettier

```
import cv2
```

BLUE

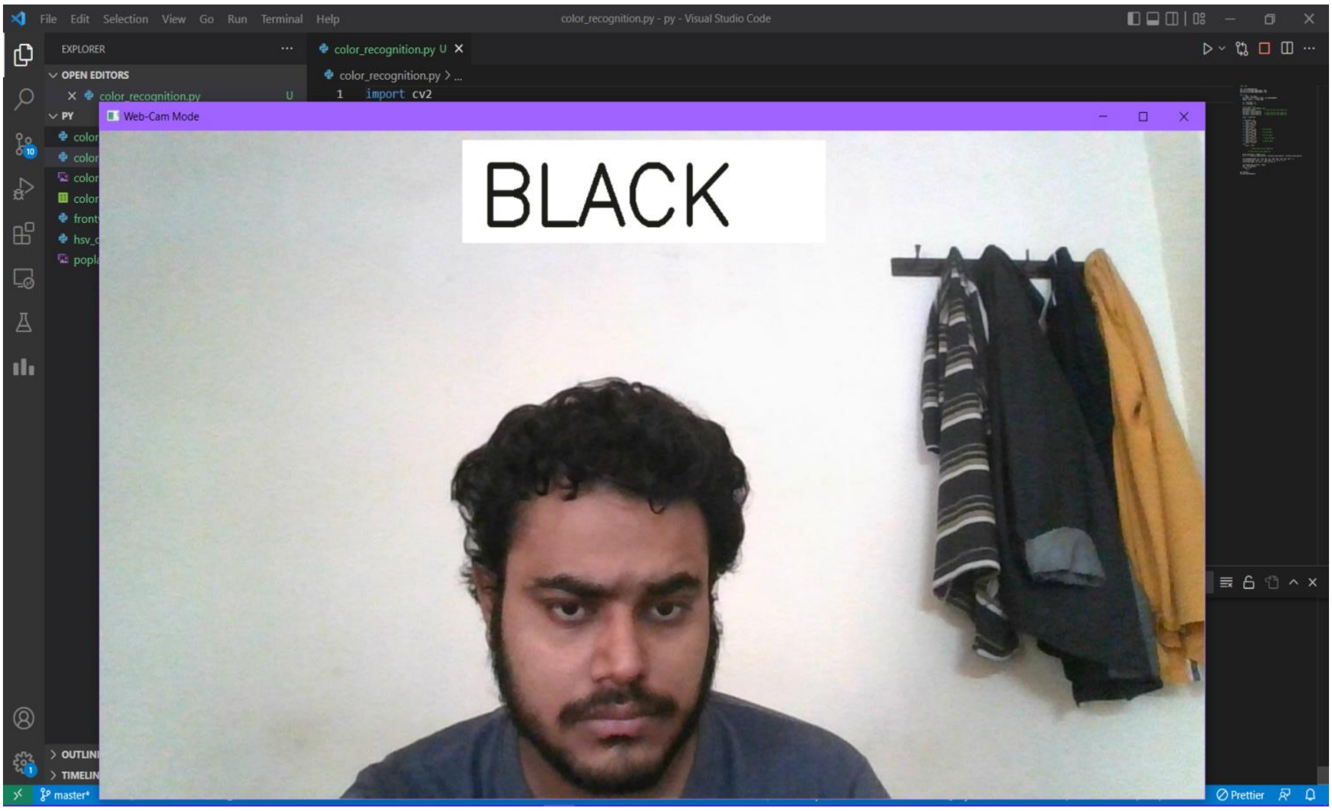
Fig 5.4


File Edit Selection View Go Run Terminal Help
color_recognition.py - py - Visual Studio Code
EXPLORER
OPEN EDITORS
color_recognition.py U
PY
OUTLINE
TIMELINE
master*
Web-Cam Mode
Prettier

```
import cv2
```

RED

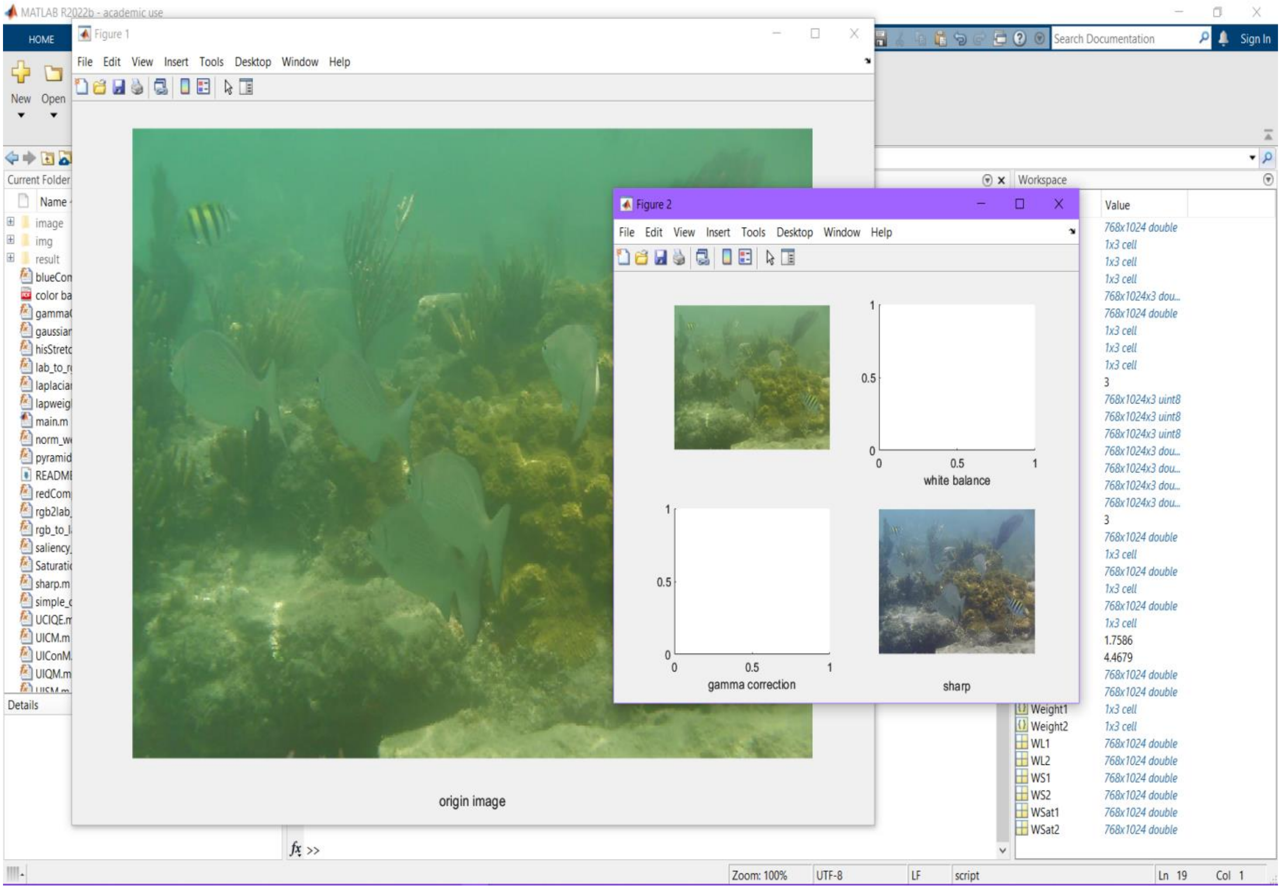
Fig 5.5


File Edit Selection View Go Run Terminal Help
color_recognition.py - py - Visual Studio Code
EXPLORER
OPEN EDITORS
color_recognition.py U
PY
OUTLINE
TIMELINE
master*
Web-Cam Mode
Prettier

```
import cv2
```

BLACK

Fig 5.6



	Value
	768x1024 double
	1x3 cell
	1x3 cell
	1x3 cell
	768x1024x3 dou...
	768x1024 double
	1x3 cell
	1x3 cell
	1x3 cell
	3
	768x1024x3 uint8
	768x1024x3 uint8
	768x1024x3 uint8
	768x1024x3 dou...
	768x1024x3 dou...
	768x1024x3 dou...
	768x1024x3 dou...
	3
	768x1024 double
	1x3 cell
	768x1024 double
	1x3 cell
	768x1024 double
	1x3 cell
	1.7586
	4.4679
	768x1024 double
	768x1024 double
Weight1	1x3 cell
Weight2	1x3 cell
WL1	768x1024 double
WL2	768x1024 double
WS1	768x1024 double
WS2	768x1024 double
WSat1	768x1024 double
WSat2	768x1024 double

Fig 5.7

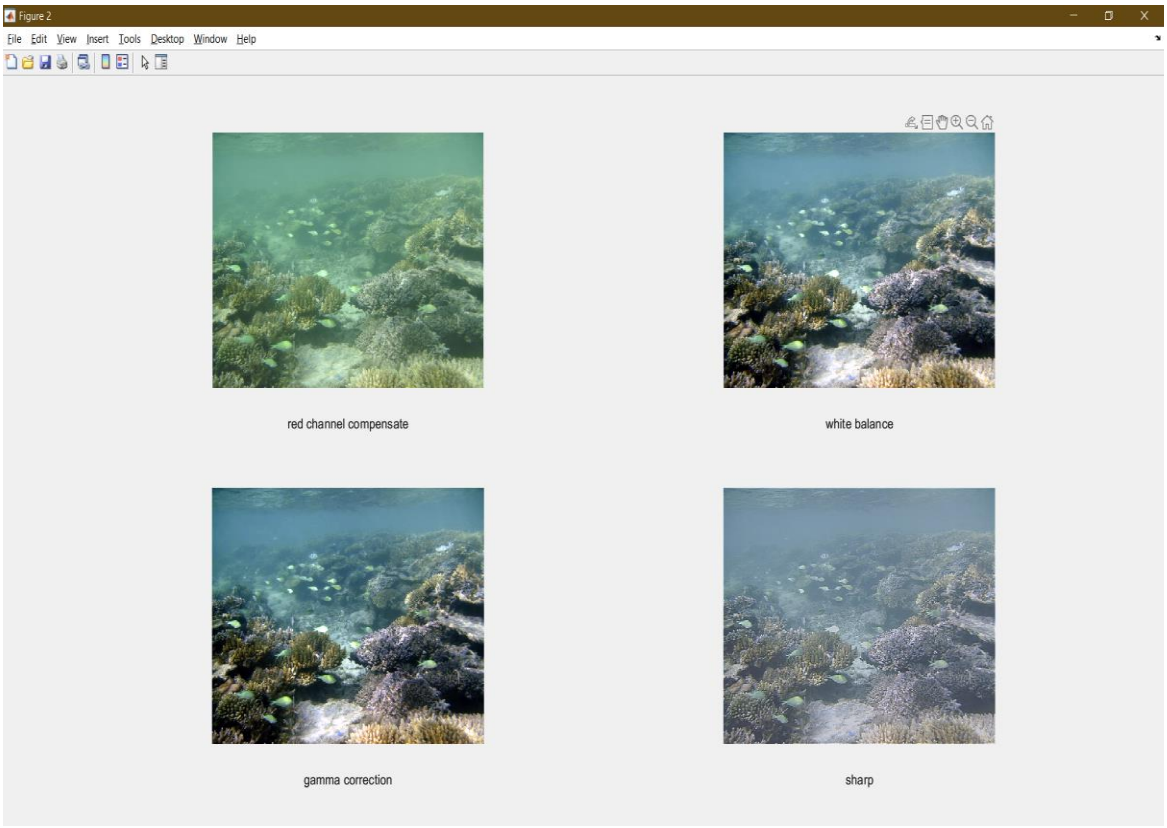


Fig 5.8

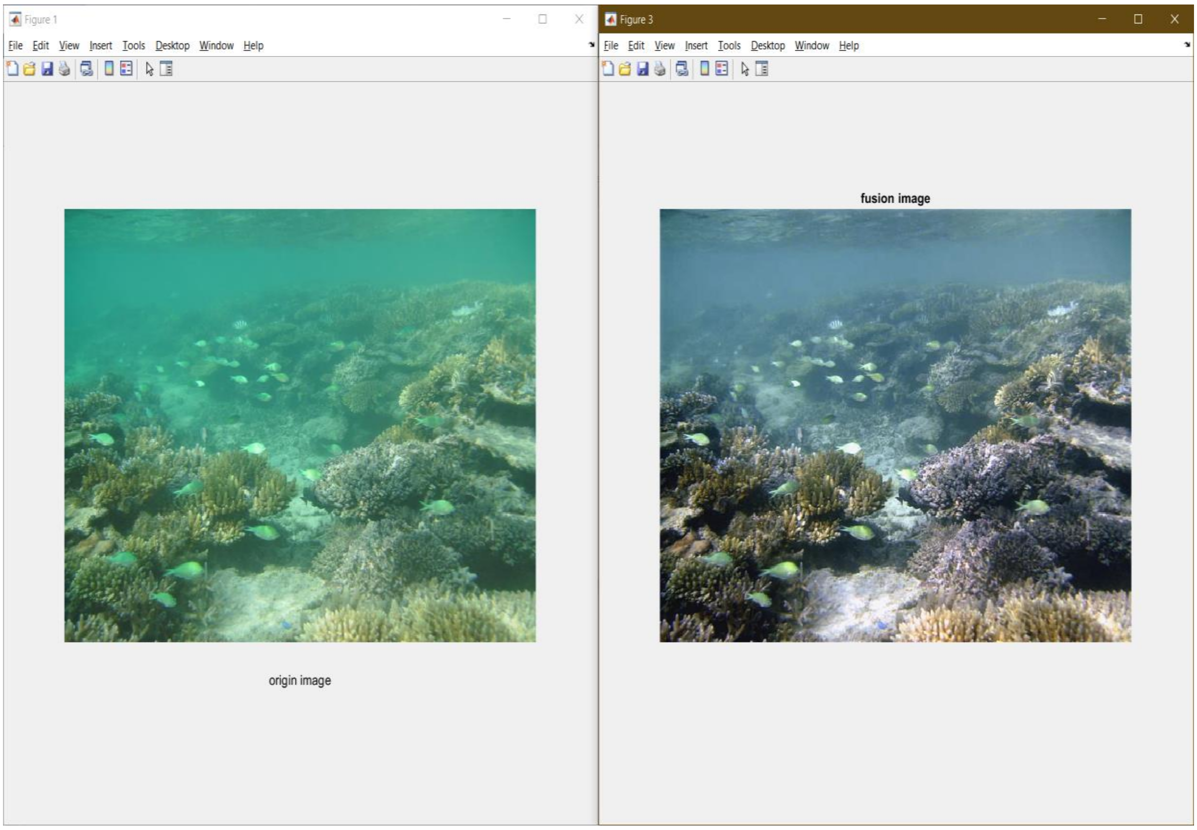


Figure 1
File Edit View Insert Tools Desktop Window Help
origin image
Figure 3
File Edit View Insert Tools Desktop Window Help
fusion image

Fig 5.9

CHAPTER 6

RESULTS AND DISCUSSIONS

• Based on our study, we have discovered that the colour detecting method is one of the most useful for persons who are colour blind.

•Our model's accuracy is around 99.9% thanks to the NAVE BAYES algorithm.

This was created in two versions:

• Webcam colour detection: It will use a webcam to identify photos and output the results.

• Color detection utilising image upload: It will use a device photo and produce results.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

As of now, we have come to the conclusion that colour detection utilising RGB values is a useful method for predicting the colour of a variety of objects. To determine which machine learning algorithm will work finest for us, we shall examine a great deal more of them.

Datasets for distinguishable hues were utilised to make predictions. Out of millions of hues, it comprised 865 or more distinct colours that could be distinguished by the RGB (red, green, and blue) values. Utilizing naive base classification, colour is predicted.

The project's future work seeks to create a colour detecting device that can differentiate between various colours using an AI model for accurate prediction, making life easier for colour blind individuals. This project might also be installed in a car where it would use the front camera to identify traffic light colours in accordance with the feature that is necessary but lacking in the project that serves this purpose.

CHAPTER 8

REFERENCES

1. G.M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 32, NO. 9, SEPTEMBER 2010.
2. Kok -Meng Lee, “Effects of Classification Methods on Color-Based Feature Detection with Food Processing Applications”, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 4, NO. 1,jan 2007.
3. Weiming Hu, Xue Zhou, “Active Contour Based Visual Tracking by Integrating Colors, Shapes and Motions”, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 22, NO. 5, MAY 2013
4. G. L. Foresti, “Visual inspection of sea bottom structures by an autonomous underwater vehicle,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 31, no. 5, pp. 691–705, Oct. 2001.
5. H. Lu, Y. Li, L. Zhang, and S. Serikawa, “Contrast enhancement for images in turbid water,” J. Opt. Soc. Amer. A, Opt. Image Sci., vol. 32, no. 5, pp. 886–893, May 2015.
6. P. Burt and T. Adelson, “The laplacian pyramid as a compact image code,” IEEE Trans. Commun., vol. COM-31, no. 4, pp. 532–540, Apr. 1983.