Colour Detection System

Submitted in partial fulfilment of the requirements
of the degree of
Bachelor of Engineering in Computer

T. E. Computer Engineering

By

GAURAV SUVARNA 20 182112

ROHIT THIRUMUMP 24 182116

SAKSHI TOKEKAR 26 182118

Guide(s):

Mrs. Varsha Shrivastava

Assistant Professor

Department of Computer Engineering

St. Francis Institute of Technology

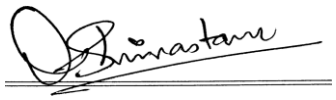
(Engineering College)

University of Mumbai

2020-2021

CERTIFICATE

This is to certify that the project entitled **"Colour Detection System"** is a bonafide work of **"Gaurav Suvarna" (20) , "Rohit Thirumump" (24) and "Sakshi Tokekar" (26)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of T.E. in Computer Engineering



(Mrs. Varsha Shrivastava)
Guide

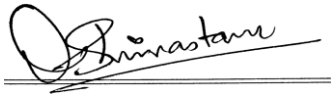
(Dr. Kavita Sonawane)
Head of Department

`

Project Report Approval for T.E.

This project report entitled ***"Colour Detection System"*** by ***Sakshi Tokekar, Rohit Thirumump and Gaurav Suvarna*** is approved for the degree of ***T.E. in Computer Engineering.***

Examiners

1. 
Ms. Varsha Shrivastava

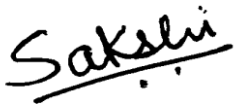
2. Ms. Kriti Srivastava

Date: 28/05/2021

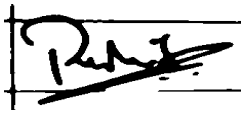
Place: MUMBAI

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Sakshi Tokekar 26



Rohit Thirmump 24

Gaurav Suvarna 20

Date:28/05/2021

Abstract

Colour detection is process of detecting the name of any colour. This process is extremely simple for humans to do as Human eyes and brain work together to translate light into colour. Light receptors that are present in our eyes transmit the signal to the brain. Our brain Recognizes the colour, and then maps them or label them as a particular colour. Since Childhood our brain has mapped certain light with their colour name, this process of mapping the colour is easy for a human to do, But the same couldn't be said to computers. The task which is straightforward for humans, is difficult for a computer. In computer, a colour in a particular pixel of an image is given by its 'RGB' values. The RGB values is basically the ‘Red’, ’Blue’ and ‘Green’ value in a pixel. Any change in the RGB value changes the colour which is present in that pixel. This can be used for mapping more colours by just adjusting the RGB value slightly. This concept is used for creating our project where we used various algorithms to create a model where it trains the model using the RGB values. The user can choose any image that he wants and select any pixel where he wants to get the colour name. Then use the RGB value of that particular pixel and give to the model which is trained and predict what colour it is and give its RGB values.

Keywords – Colour, Colour Detection, Colour Classification, Colour Identification, Classification in Machine Learning, KNN, Logistic Regression, SVM, Decision Tree, Random Forest

Contents

List of Figures

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1	RGB	Red Green Blue
2	LCD	Liquid Crystal Display
3	Knn	K-nearest neighbour
4	FR	Functional Requirement
5	CPU	Central Processing Unit
6	TB	Terra Byte
7	GB	Giga Byte
8	DDR4	Double Data Rate 4

Chapter 1

Introduction

The normal human eyes have three types of sensors and the signal of these three sensors determine the colour response of the observer. The response of this system produces the three-dimensional phenomenon of three-dimensional spaces. When a human sees something, the light enters the eye and hit the light detector on the retina. This behaves similarly to a digital camera that records more reading whenever more lights hit the light detector on the back of the camera. Therefore, in the electronics graphical system RGB colour model is similarly applied to provide such coordinate system of three-dimensional spaces.

In electronics graphic system, RGB colour model is commonly used for input and output devices such as colour TV, digital cameras, computer screen, and LCD. Computer monitors for instance use the RGB colour and the computer images are commonly stored in RGB colour system. RGB colour system is based on three basic colour components of red, green and blue in pixel {r, g, b}.In image colour, for each colour component, there is a range of intensity, from 0 to 255 and when these three combinations of various intensities are superimposed, a colorized hue is obtained.

From this theory, the colour recognition model has been applied widely in industrial sectors, commercial fields as well as in social responsibilities. For instance, it is used as a powerful and reliable parameter in robotics machines, aid for the blind and the colour-blind people, diamond colour sorting, quality control for the manufacture coloured paper, and in characterizing the thermal paints. Rami Al-Hmouz and Subhash Challa in their research in the field of Automatic License Plate Recognition have proposed the application of the information fusion enhanced colour recognition technology in stolen vehicle identification.

1.1 Description

The objective of this project was to develop a system where it will predict the colour of a particular pixel of an image of user’s choice. The user can select an image of his choice and can double click on

any pixel and we can get to know the colour name and the red, blue and green value of that colour. The prediction of colour can be done from various classification model, so the user can choose which classification model he wants the prediction from.

1.2 Problem Formulation

Colour can be detected easily by humans but for computer it is not a straightforward task. Human eyes and brain work together to detect colours in light. In computer a colour has a certain RGB value, and by changing the RGB values we can get different colours.

Humans detect colour since childhood and they mapped particular light with colour name. So, the same strategy can be applied to computers and have a colour a certain RGB value and give it a colour name. So, our system will take these RGB and try and predict the colour name of that particular colour that the user has selected.

1.3 Motivation

For Humans, to detect colour is a wonderful thing, and throughout their life they detect different thing with different colour. But for some, it can't be said the same as they see colour. These people are called colour blindness. There are those who do photo editing, they want get the name of some colour they don't know. So, we created a project based on these factors where a person can upload an image of their choice and can get the name of the colour on a particular pixel.

1.4 Proposed Solution

We use Machine Learning algorithm to predict colour based on their RGB value. We will us different classification algorithm to classify the colour and train the model. In machine learning, classification refers to a predictive modelling problem where a class label is predicted for a given example of input data. And we use OpenCV for Image uploading and image processing. For Window creation we will we tkinter of python.

1.5 Scope of the project

This project has the potential to be used in many different ways. This can be used to check whether the a given fruit is ripe, rotten and raw. It can be also used in photo editing software where a person can check the colour name and use it. This can be used by a colour-blind person to check a colour of image.

Chapter 2

Review of Literature

Papers like ‘Statistical approach for colour image detection’ written by Sidhanta Kumar Kar and Mihir Narayan Mohanty [1] were extremely useful in helping us along the way. The paper gave us an idea about where to begin our work and what path we should use. According to this paper the algorithm used in it was successful in recognizing colour of the object differently from that of the background. An iterative method is used to obtain the binary values of different layers i.e. R, G, B. The codes for acquisition of a colour image of an object and calculating the ROI in the image was written. Finally, the colour of the object was recognized. With this we concluded that we needed to use the RGB values in the image to find the required colour.

The paper ‘An efficient colour detection in RGB space using hierarchical neural network structure’ [2] made us think to go for neural network due to some of the advantages it provides. Eventually thought we decided not to go with neural networks for a few different reasons.
According to the paper colour detection is an intermediate stage in most of colour image processing and widely needed in many applications and the performance of the colour detection directly affect the subsequent processing resulting overall deterioration or further improvements. This is also one of the reasons the images were converted from RGB space to HSV or HSI space.

A hierarchical neural network structure which is able to classify pixel colour accurately. The classical structure consists of one neural network is also good at classifying but as the complexity increases this method will produce better results.

The paper ‘Specific Colour Detection in Images using RGB Modelling in MATLAB’ [3] also helped in understanding on different ways on how we could go about for color detection. In the paper, they went about from taking the image, storing it, then Defining the color you want, then Converting the image into binary form then find the color that is defined and label it using a bounding box. They used binary images as to make a working area for the computer I.e. 1 for white and 0 for black. They removed objects having less than 300 pixels as to increase the performance and accuracy. We got to

learn a different way of doing color detection as this can be used while doing neural networks but we decided to drop it as we are going for supervised learning algorithm.

We decided to go with KNN Classification algorithm based on the paper “Colour based image segmentation using classification of k-nn with contour analysis method” [4]. In this they divided the colours into various sections such as purple, green, greenish yellow, etc. Then they segment the image based on which colour they want and predict it. We found some ideas as how to go our way about using the KNN algorithm but decided not to use the algorithm the same way because of our dataset that we have selected. Because our Dataset has all different and unique values, they cannot be formed into groups and forms too many segments for the computer to handle. Though soon we came to the conclusion that knn is not as effective. We decided to try different algorithms using the same method to get better results.

The paper ‘Object Detection for Color Blind Applying Binary Images’ [5] let us get a better idea of how we could further develop our application in the future. This paper is basically used for Colour-blind people and colour blindness test. Normally algorithms use “background subtraction” to find the output image but the paper considers both the foreground and the background.

When background is subtracted few features of the image are lost but the algorithm is trying to keep intact all these features. Keeping all the features actually helps the output image to be much better for pattern recognition not only that in few cases keeping all features enhances the output image.
As the algorithm works colour pixel difference a colour-blind person will understand the whole image and will detect every object present in the image. A colour-blind person will see the whole image through a device and will understand or detect the objects present , it might be that the eye does not catches a colour and miss the object but the algorithm will help the person to detect that object.

Chapter 3

System Analysis

3.1 Functional Requirements:

FR1: USER INTERFACE: The user interface will be an application. The user will be given option to choose which ever image the user wants.

FR2: PROPER FORECASTING: The system has to properly predict the colour that the user clicks on from the selected image.

FR3: SENTIMENT ANALYSIS SYSTEM:

FR4: DATABASE: Dataset contains large number of colour names and their RGB values for the system to predict the colour accurately.

3.2 Non-Functional Requirements:

Performance Requirements:

Any basic System can run the application as long as it supports images. The system with an average clock speed is enough to run the application smoothly. With the help of modern OS like windows 7 or 10, Mac OS X10.11 or higher. The system should have at least 4 gb ram.

Security Requirements:

The application does not have any security requirements and thus any type of user can use it without any additional privileges

3.3 Specific. Requirements

• Hardware Requirements:

CPU --Intel Core i5 9300H

Hard Disk Space -- 1TB

Memory -- 2666Hz 8gb DDR4 ram

Other Devices – Laptop

• Software Requirements:

Front End --Tkinter

Back End -- Python

Languages -- Python

Operating System -- WINDOWS 10

3.4 Use case diagram

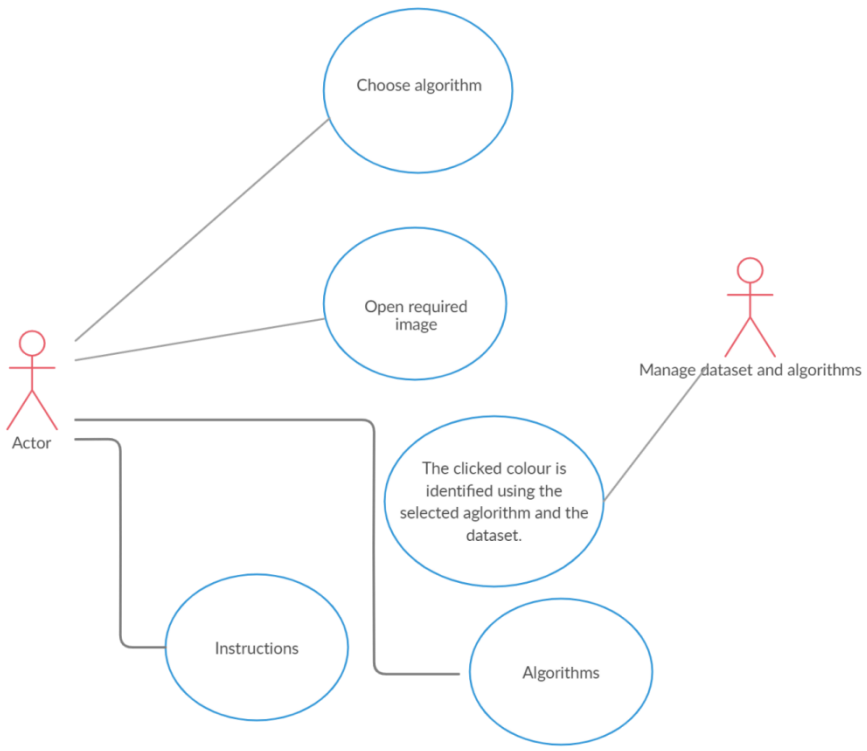


Fig 3.1 Use Case Diagram of Colour Detection System

Chapter 4

Analysis Modelling

4.1 Activity Diagram

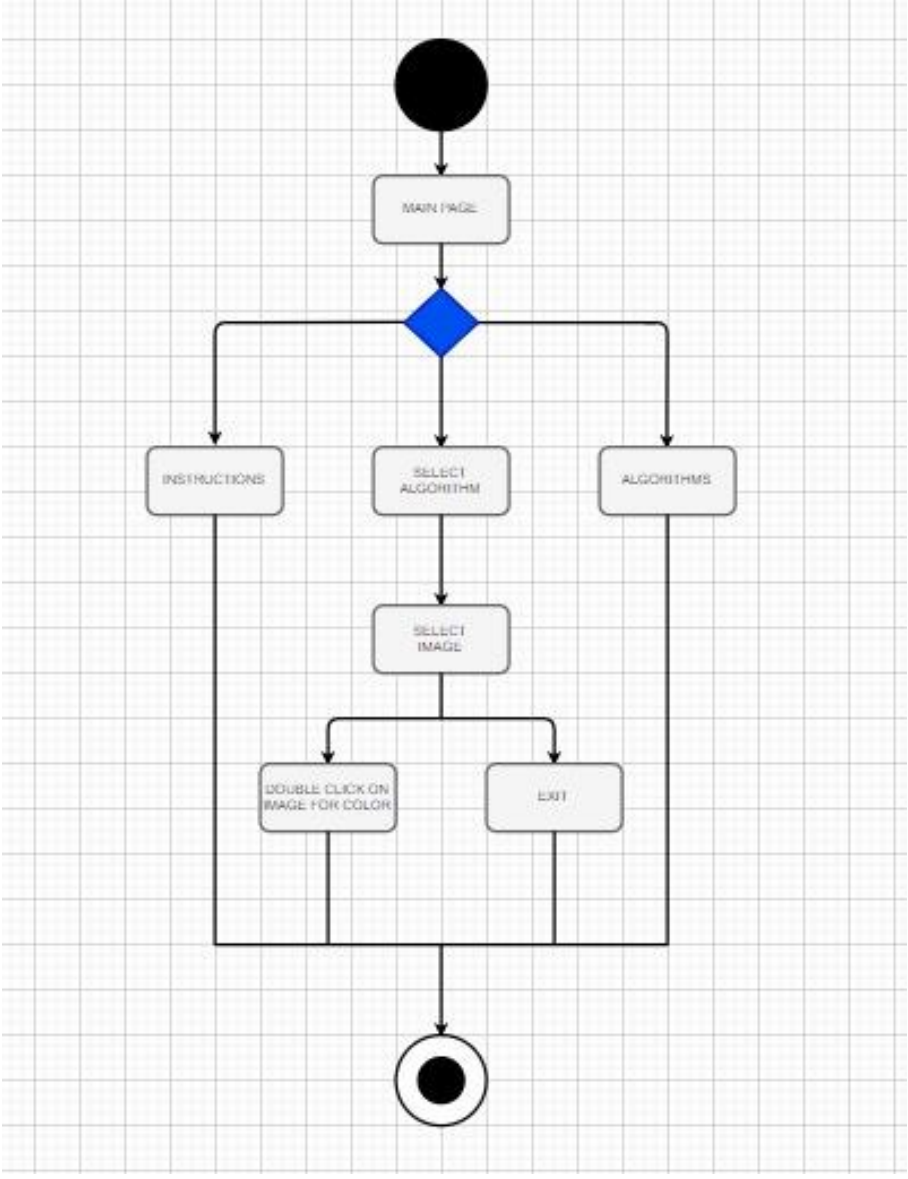


Fig 4.1 Activity Diagram of Colour Detection system

4.2: DATA FLOW DIAGRAM:

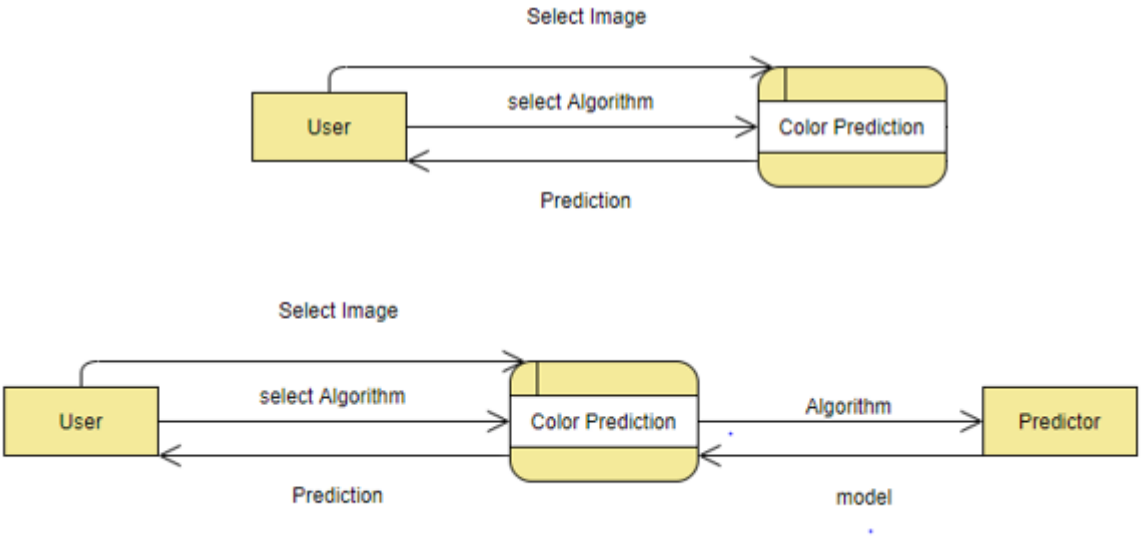


Fig 4.2 Data Flow Diagram of Colour Detection System

4.3: WORK FLOW:

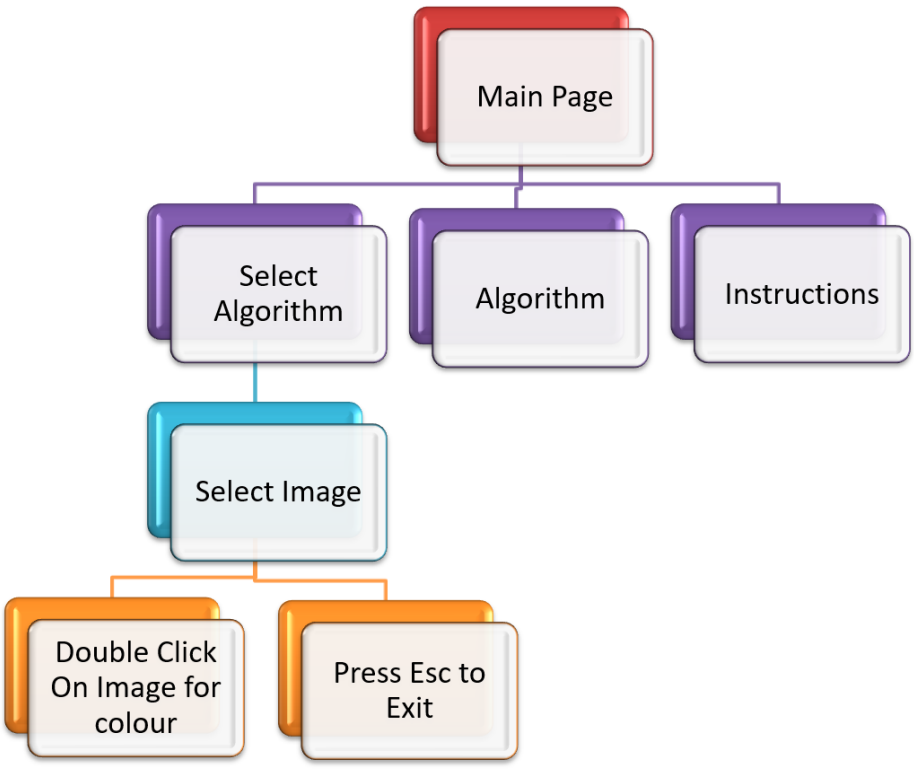


Fig 4.3 Work Flow of Colour Detection System

Chapter 5

Design

5.1 Architecture Design:

FR1: USER INTERFACE: The user interface will be a window. They can upload image of his/her own choice.

FR2: ALGORITHM SELECTION: The system will ask the user to select one algorithm to predict the colour.

FR3: IMAGE SELECTION: The system will ask the user to select image for colour prediction.

FR4: COLOR PREDICTION: The user will double click on any pixel on the image that the user has selected and the colour name and the RGB value will be displayed.

5.2 User Interface Design:

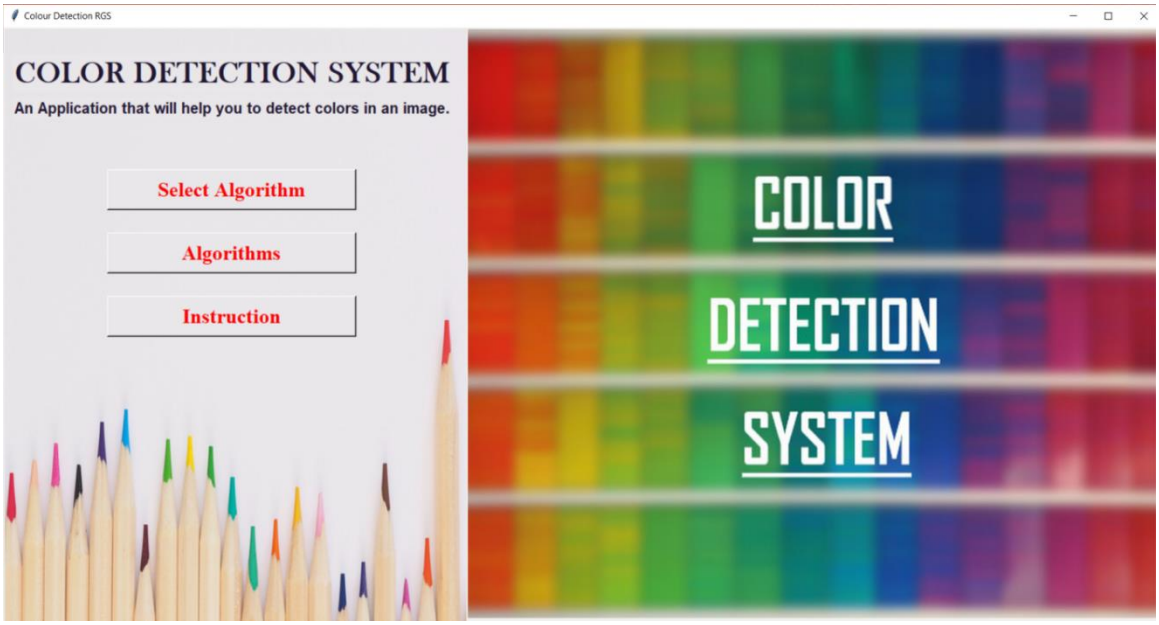


Fig 5.1 Main Page of Colour Detection System

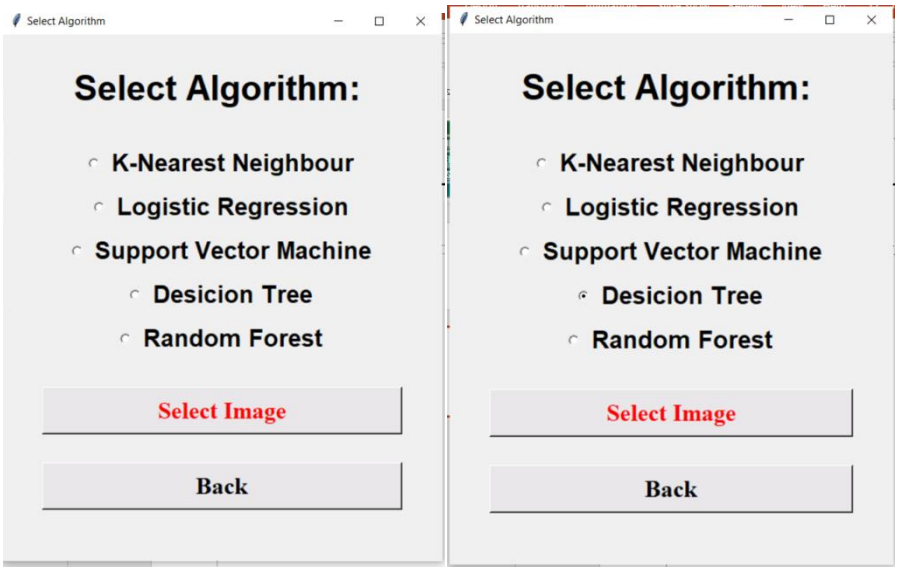


Fig 5.2 Select Algorithm Page

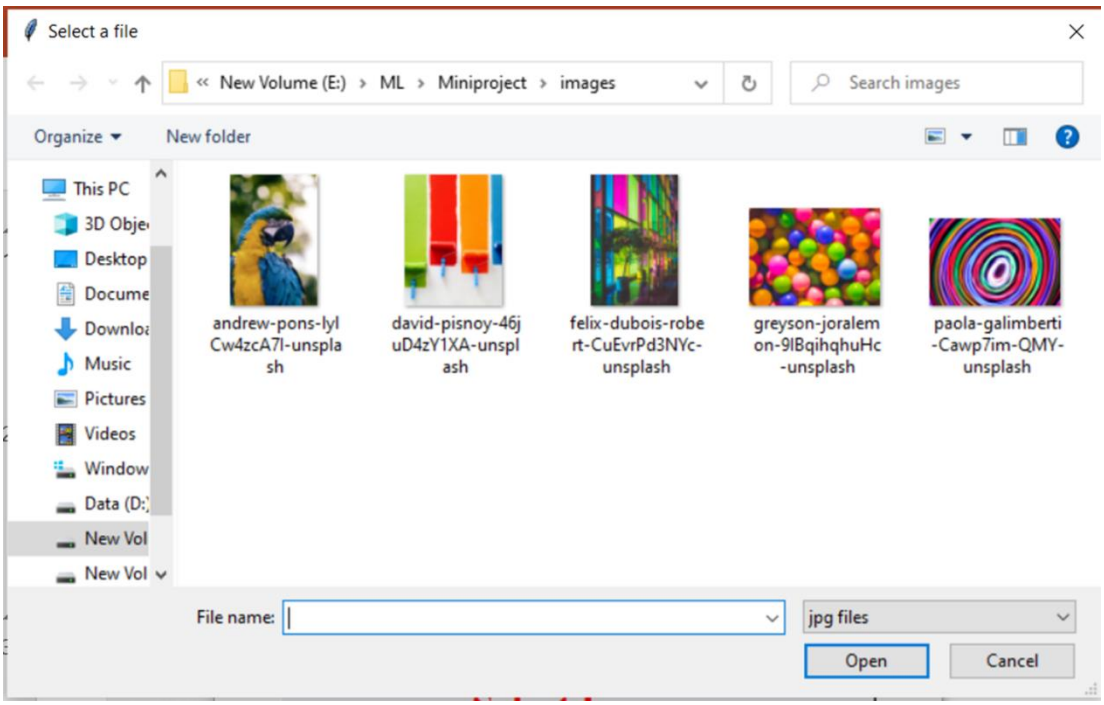


Fig 5.3 Select Image

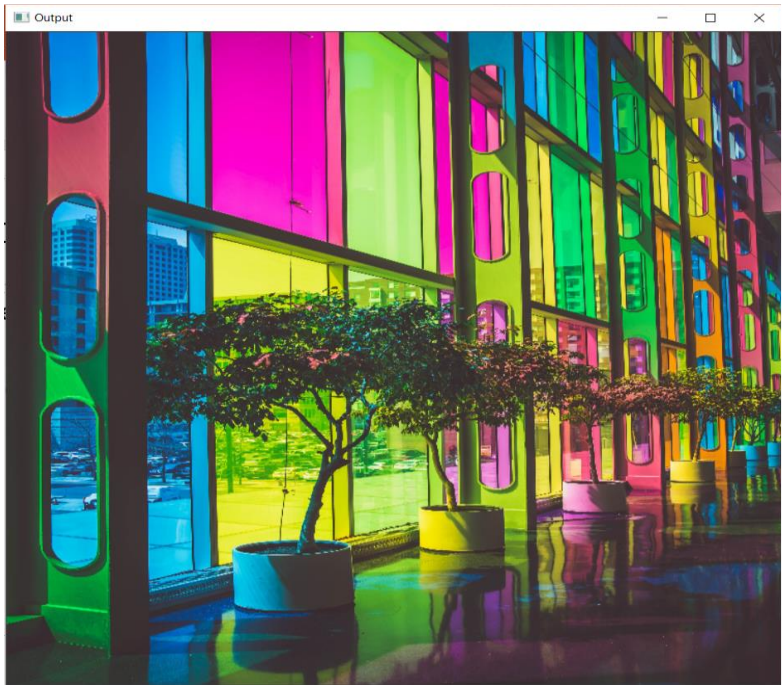


Fig 5.4 Selected Image opened

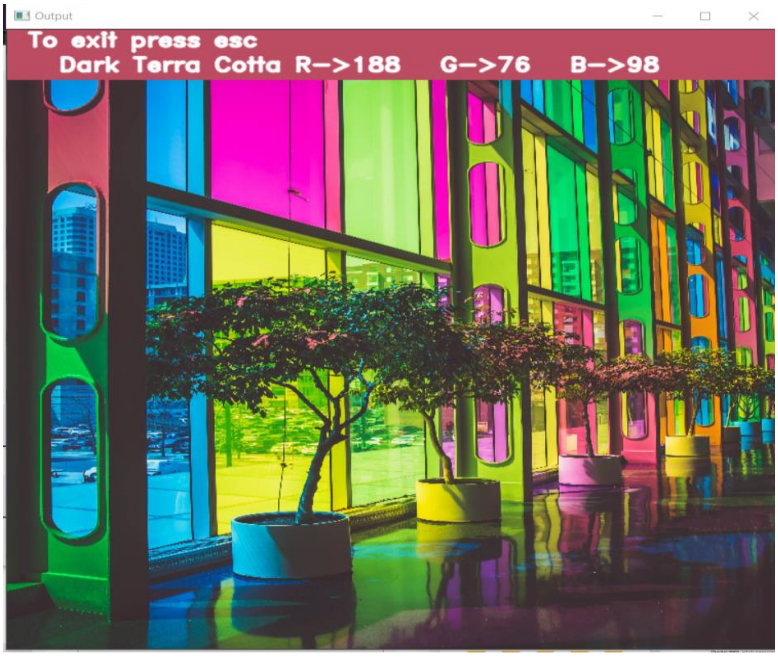


Fig 5.5 Colour Detected on double click

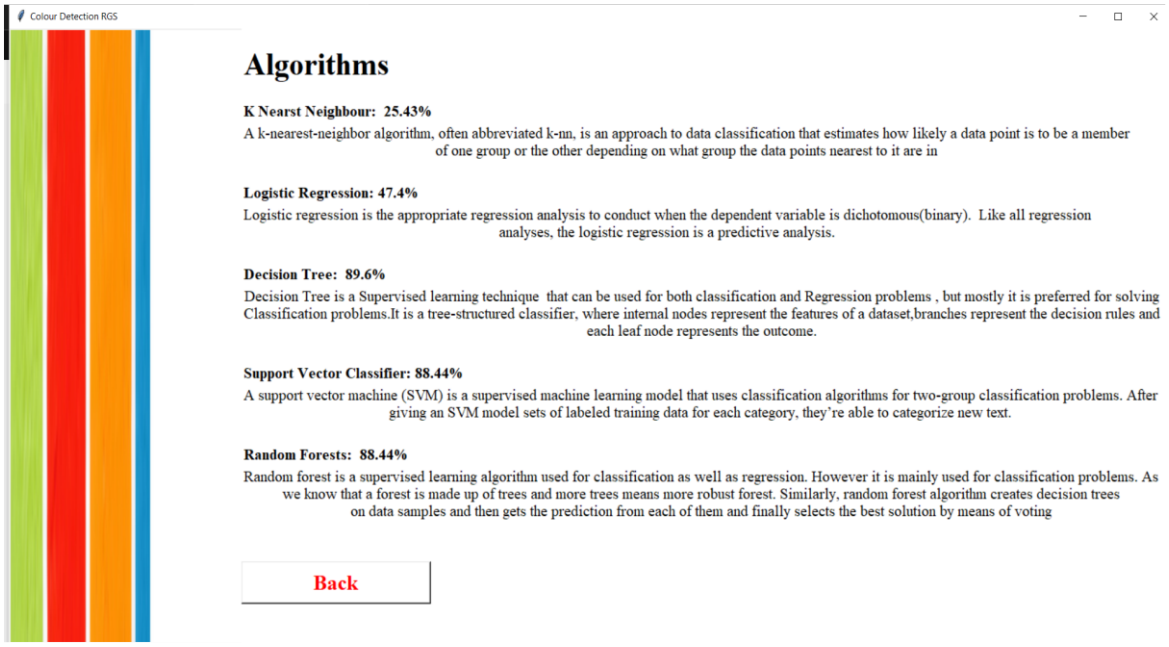


Fig 5.6 Algorithm Information Page

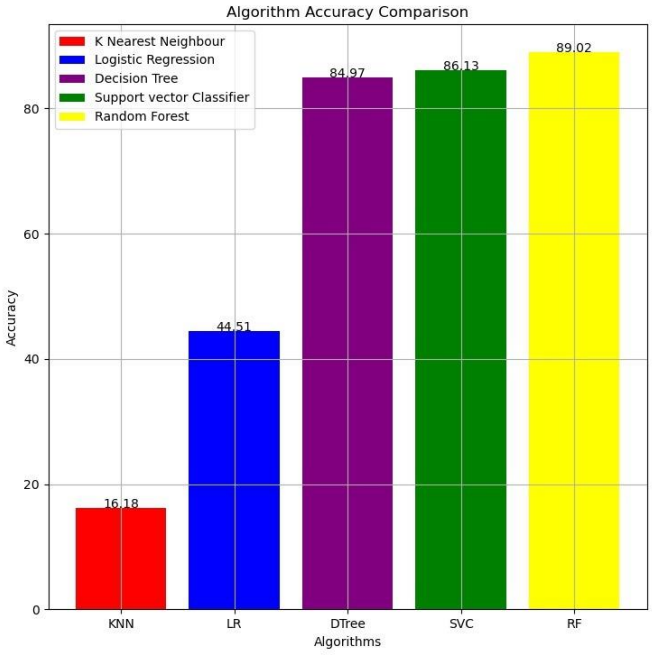


Fig 5.7 Algorithm Accuracy Comparison Page

Chapter 6

Implementation

6.1 Algorithms / Methods Used:

We used different **Classification Algorithm** from scikit learn library to predict the colour which the user wants to predict from his desired image.

- We first trained 5 different **classification** model by feeding them the data set to train with.
- Then we uploaded the image and took the pixel co-ordinate and the RGB value.
- Then we feed the RGB value to the model which was trained and predict the colour name of that pixel.

6.1 Working of the project

1) Taking the Input from the user

We Take the Path of the image from the user and use it to open the image. Then we resize that image to 750x750 pixel image.

```
img = cv2.imread(path)
cv2.imshow('Output',img)
img = cv2.resize(img, (750, 750))
```

Fig 6.1 Input from the user

2) Importing the Libraries

We use various libraries help us create our system:

- pandas: - to read the libraries
- cv2 : - Image processing
- numpy: - numpy array
- scikit learn: - By using scikit learn we can import various classification model

```
import pandas as pd
import numpy as np
import cv2
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

Fig 6.2 Importing the Libraries

3) Reading the Data set

By using pandas library, we can read the .csv file which contains the colour name and its RGB values.

```
index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
dataset = pd.read_csv('colors.csv', names=index, header = None)
dataset = dataset.drop(columns = ['color','hex'])
X = dataset.drop(columns=['color_name'])
y = dataset['color_name']
```

Fig 6.3 Reading the data set

4) Set a mouse callback event on a window

First, we created a window in which the input image will display. Then, we set a callback function which will be called when a mouse event happens.

```
cv2.namedWindow('Output')
cv2.setMouseCallback('Output',DrawFunction)
```

Fig 6.4 Mouse Call-back event setup

With these lines, we named our window as 'Output' and set a callback function which will call the **draw_function()** whenever a mouse event occurs.

5) Create the Draw Function

It will calculate the rgb values of the pixel which we double click. The function parameters have the event name, (x,y) coordinates of the mouse position, etc. In the function, we check if the event is double-clicked then we calculate and set the r, g and b values along with x, y positions of the mouse.

```
def DrawFunction(event,x,y,flag,param):
        if event == cv2.EVENT_LBUTTONDBLCLK:
                global b,g,r,xpos,ypos, clicked,img
                clicked = True
                xpos = x
                ypos = y
                b,g,r = img[y,x]
                b = int(b)
                g = int(g)
                r = int(r)
```

Fig 6.5 Create the Draw Function

6) Create The model and Predicting the colour

From the Prediction Function we let the user choose which algorithm they want and call that particular model and predict the colour by using the. predict function and return the colour name.

```
def Prediction(R,G,B,algo):
        index = ['color', 'color_name', 'hex', 'R', 'G', 'B']
        dataset = pd.read_csv('colors.csv', names=index, header = None)
        dataset = dataset.drop(columns = ['color','hex'])
        X = dataset.drop(columns=['color_name'])
        y = dataset['color_name']
        model = None
        algo_name = None
        if algo == 'KNN':
                model = classify(X,y)
                algo_name = 'K-Nearest Neighbour'
        elif algo == 'LR':
                model =  Logisticregression(X, y)
                algo_name = 'Logistic Regression'
        elif algo == 'SVM':
                model = SVM(X, y)
                algo_name = 'Support Vector Machine'
        elif algo == 'DT':
                model = DesicionTree(X, y)
                algo_name = 'Decision Tree'
        elif algo == 'RF':
                model = RandomForest(X, y)
                algo_name = 'Random Forest'
        y_pred_color = model.predict([[R,G,B]])
        color = y_pred_color[0]
        return color
```

Fig 6.6 Prediction

When the User chooses a particular algorithm then it calls that function and the model is trained and model is returned for prediction.

```
def classify(X, y):
        model = KNeighborsClassifier(n_neighbors=5, metric='euclidean')
        model = model.fit(X,y)
        return model

def Logisticregression(X, y):
        # Training the model
        model = LogisticRegression()
        model = model.fit(X,y)
        return model

def SVM(X, y):
        # Training the model
        model = SVC()
        model = model.fit(X,y)
        return model

def DesicionTree(X, y):
        # Training the model
        model = DecisionTreeClassifier(random_state=0)
        model = model.fit(X,y)
        return model

def RandomForest(X, y):
        model = RandomForestClassifier()
        model = model.fit(X,y)
        return model
```

Fig 6.7 Creating the model

7) Displaying the image on the window

Whenever a double click event occurs, it will update the color name and RGB values on the window.

Using the cv2**.**imshow**()** function, we draw the image on the window. When the user double clicks the window, we draw a rectangle and get the color name to draw text on the window using cv2**.**rectangle and cv2**.**putText**()** functions.

```
while(1):
        cv2.imshow("Output",img)
        if (clicked):
                cv2.rectangle(img,(0,0), (750,60), (b,g,r), -1)
                text = Prediction(r,g,b,al) + ' 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)
                msg = "To exit press esc"
                cv2.putText(img, msg,(20,20),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)
                        cv2.putText(img, msg,(20,20),2,0.8,(0,0,0),2,cv2.LINE_AA)
                clicked=False
        if cv2.waitKey(20) & 0xFF ==27:
                break
cv2.destroyAllWindows()
```

Fig 6.8 Image Display

Chapter 7

Conclusion

• The duration of our working on this project has helped us enhance our knowledge on a variety of different things. From image processing to working with machine learning models, we truly have learnt a lot.

• While building this project, we learnt how and when to use different models, how different algorithms work, how the accuracy and efficiency of the algorithms affects our model, how can we implement image processing in our model are some of things that we learnt.

• As seen before, Random Forest classification has the highest accuracy when it comes to predicting the colour of an image provided to the system by the user

• While the K-Nearest Neighbour algorithm has the worst accuracy in this aspect.

References

[1] S. K. Kar and M. N. Mohanty, "Statistical approach for colour image detection," *2013 International Conference on Computer Communication and Informatics*, 2013, pp. 1-4, doi: 10.1109/ICCCI.2013.6466131.

[2] Goel, Vishesh & Singhal, Sahil & Jain, Tarun & Kole, Silica. (2017). Specific Colour Detection in Images using RGB Modelling in MATLAB. International Journal of Computer Applications. 161. 38-42. 10.5120/ijca2017913254.

[3] H. Altun, R. Sinekli, U. Tekbas, F. Karakaya and M. Peker, "An efficient colour detection in RGB space using hierarchical neural network structure," 2011 International Symposium on Innovations in Intelligent Systems and Applications, 2011, pp. 154-158, doi: 10.1109/INISTA.2011.5946088.

[4] Dr.S.Niraimathi "COLOR BASED IMAGE SEGMENTATION USING CLASSIFICATION OF K-NN WITH CONTOUR ANALYSIS METHOD." (2016).

[5] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell and A. M. Lopez, "Colour attributes for object detection," 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3306-3313, doi: 10.1109/CVPR.2012.6248068.

Acknowledgments

We would like to acknowledge the valuable input given by our guide Mrs. Varsha Shrivastava during the tenure of our project. We are also grateful to have the company of each other in developing this interesting project. Furthermore, we would like to thank Mrs. Safa Hamdare for being our mini-project professor.