Project Report

Colour Detection

Prianandа Azhar
School Of Computer Science
Binus University
Jakarta, Indonesia
prianandа.azhar@binus.ac.id

Jason Surya
School Of Computer Science
Binus University
Jakarta, Indonesia
jason.yauw@binus.ac.id

Mikhael Enrico Setianto
School Of Computer Science
Binus University
Jakarta, Indonesia
mikhael.setianto@binus.ac.id

I. Introduction

Colour detection is the process of detecting the name of any colour. Our eyes and brains work together in order to translate light into colour. Light receptors in our eyes transmit the signal to the brain and then our brain recognizes the colour. Through enough exposure, we are able to distinguish certain lights with their colour names. For example, we are able to tell the difference between black and white, red and yellow ,etc.

However, computers do not have this natural ability. It is not as straightforward for computers to detect colours as it is for human beings. Colours a mixture of three primary colours, Red, Green and Blue. Computers define each primary colour value within a range of 0 to 255 and a specific colour is defined by their combined RGB values.

Because of this, the number of ways to define a colour is 256 to the power of 3 which is also approximately equal to 16.5 million. Most humans cannot even remember 100 colours.

Hence, we need to create a dataset to map combinations of RGB values to their corresponding names and create an algorithm for the computer to be able to predict colour names by using the current combination of RGB values of a pixel and find the nearest match in the dataset.

II. Methodology

For this project, we will use the colour dataset from kaggle [4]. The dataset contains the colour name, hex code, RGB values of 865 colours and is saved in csv format.

In order to solve this problem, we need to use 3 python libraries which are numpy, pandas, and opencv. Numpy is open source and helps with numerical computing[1]. Pandas is used to manipulate and read csv files and opencv is an open source computer vision library that will help us create our computer vision application.

We will also use random test images searched from google that vary in colour and sizes. We will use the resize function from opencv to equally resize all the images.

For colour prediction, we will use the K Nearest Neighbour algorithm or more commonly known as KNN algorithm. In our implementation of KNN algorithm, we find the nearest neighbour(k=1) by calculating the Manhattan distance. The RGB values of a pixel we are trying to find the colour of are passed to the algorithm and then we find the Manhattan distance of the respective RGB values of each colour in the dataset with the respective RGB values of the pixel. The one with the shortest Manhattan distance is the predicted colour name.

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

Fig .1. Formula of Manhattan Distance

III. Results

We want our program to show the user the color name after double clicking, what the program is doing is that it is trying to find the nearest color using those RGB values. As said before, we use the KNN classifier algorithm to find the nearest color that matches those three attributes which are R, G and B. Here are some example of working program of ours to show color names on various images:

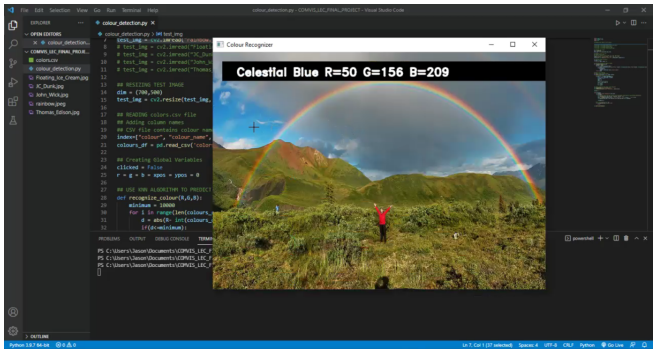

Fig .2. Color classification result

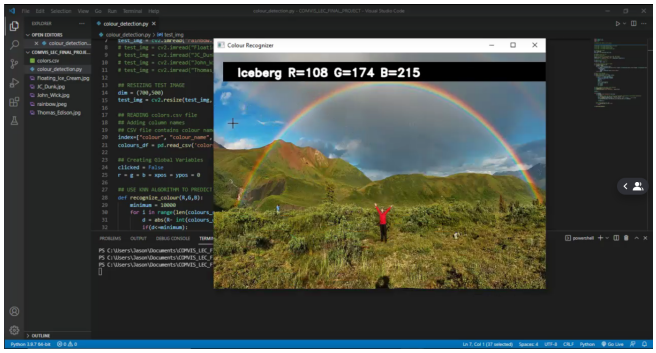

Fig .3. Color classification result

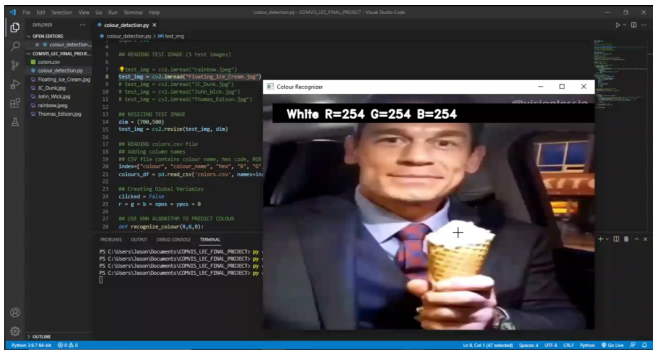

Fig .4. Color classification result

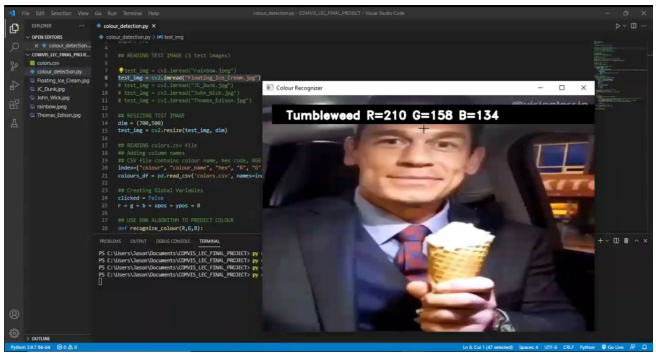

Fig .5. Color classification result

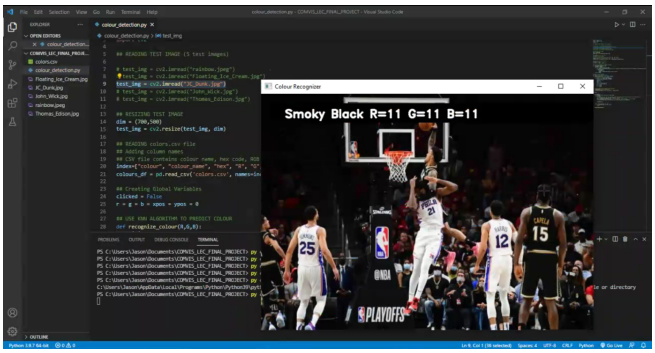

Fig .6. Color classification result

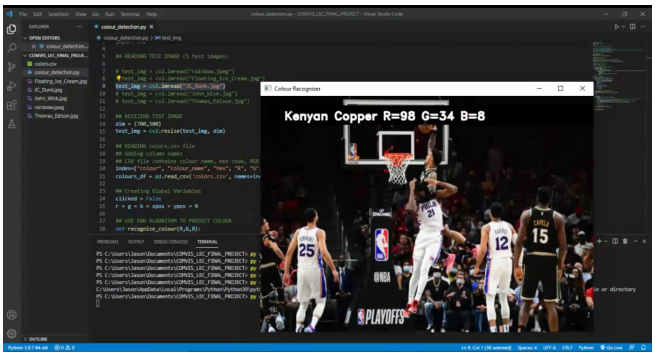

Fig .7. Color classification result

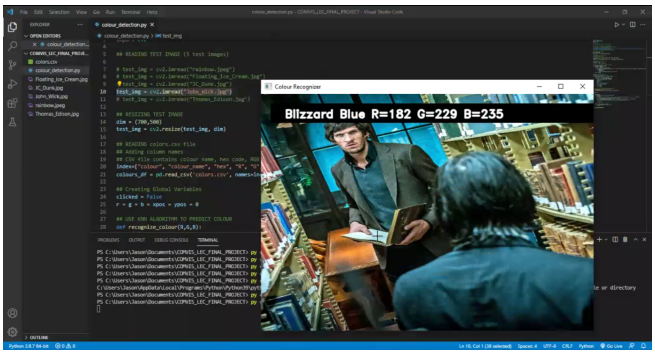

Fig .8. Color classification result

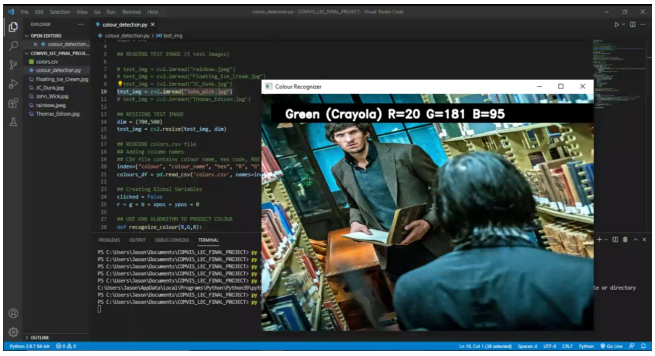

Fig .9. Color classification result

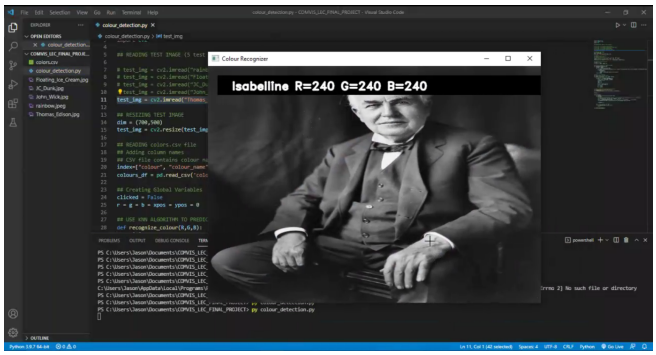

Fig .10. Color classification result

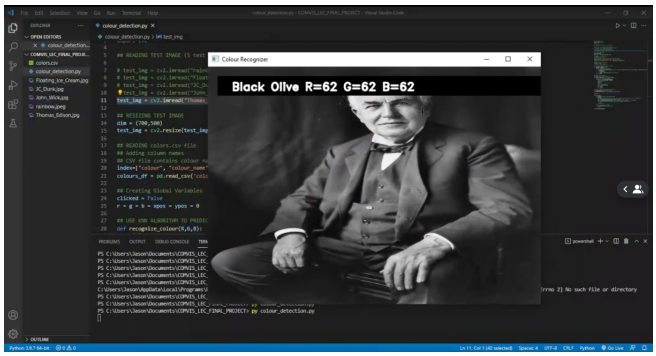

Fig .11. Color classification result

We believe that our program can classify any color name on any images inputted. Since we have several hundreds of color names in the data frame available, we think that it would be more than enough to classify any image provided.

IV. Conclusion

In today's world, computer vision has a wide range of uses. It has aided people in image processing and analysis by allowing them to manipulate images or detect objects in images.

Using opencv to read image data and to draw window output, also KNN algorithm to classify its RGB color values, we manage to predict color names in a cursor click. After conducting some experiments and adjusting the algorithm and the parameters, we can say the results are good enough and accurate as it can classify the color name on the clicked cursor.

References

[1] https://numpy.org/
[2] https://pandas.pydata.org/
[3] https://opencv.org/
[4] https://www.kaggle.com/adityabhndari/color-detection-data-set?select=colors.