

BLG 513E HW3

Colour Classification

Eren Kılıç

504201553

Department of Computer Engineering
Istanbul Technical University

1-) INTRODUCTION

In this assignment, we are asked to implement a program that can classify the colour of given input image. The program contains eight classes of colours which are red, black, orange, yellow, green, blue, white, and violet.

2-) DATASET

In the program, I used two different datasets. The first one whom I found on Kaggle contains raw colours of each classes from different hues.

DATASET 1: TRAINING DATASET

Name:	Color Dataset for Color Recognition
Author:	ADI KURNIAWAN
Link:	https://www.kaggle.com/adikurniawan/color-dataset-for-color-recognition?select=training_dataset#
# of images	25 images for each colour (200 in total since we have 8 classes)

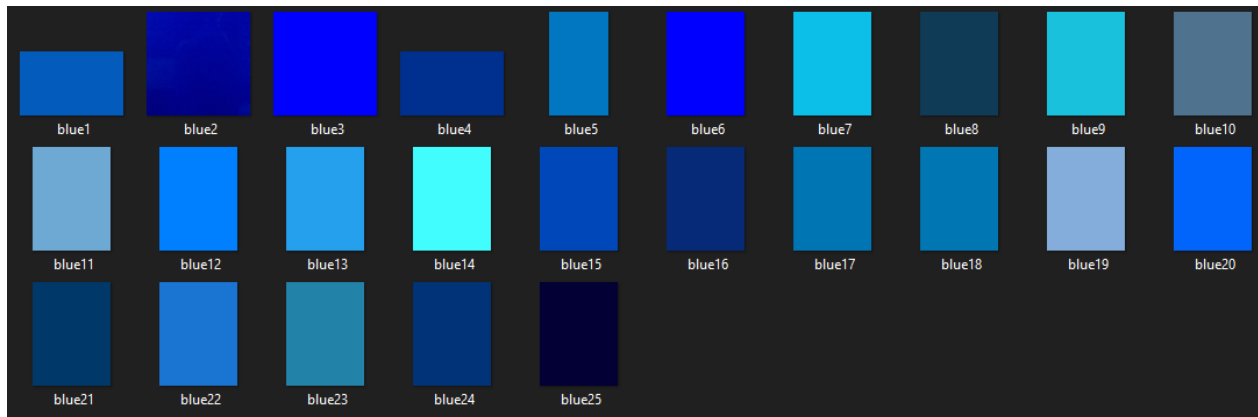


Figure 1: Blue colours of training dataset

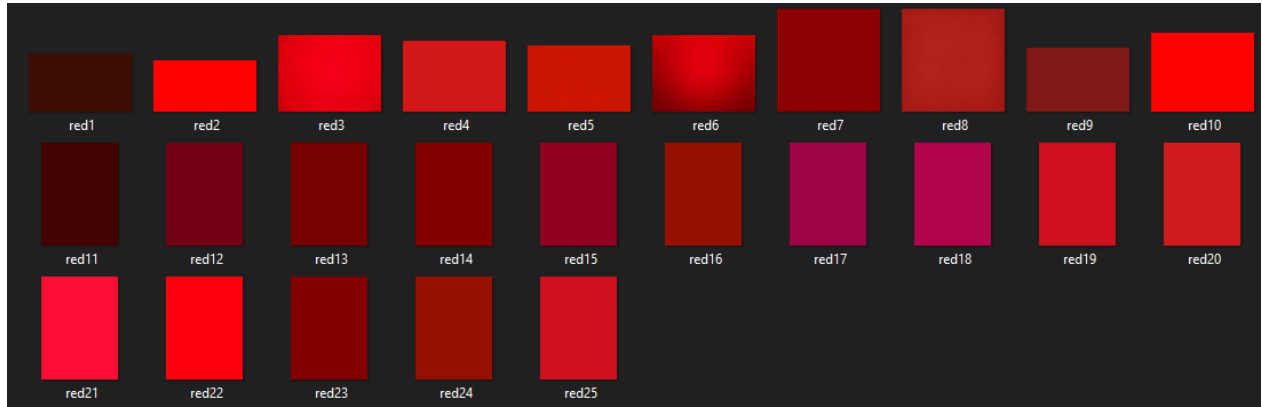


Figure 2: Red colours of training dataset

DATASET 2: TESTING DATASET

The second dataset whom I prepared contains images of daily life objects from different colours. I mostly took images with my camera. However, when I could not find more objects in a class, I requested to my friends to take pictures for me.

Name:	Colourful Daily Life Objects
Author:	Eren Kılıç
Link:	https://drive.google.com/drive/folders/1LP7ysjNEtBILy9NXfTpbEEwhNsH2IKJI?usp=sharing
# of images	10 images for each colour (80 in total since we have 8 classes)

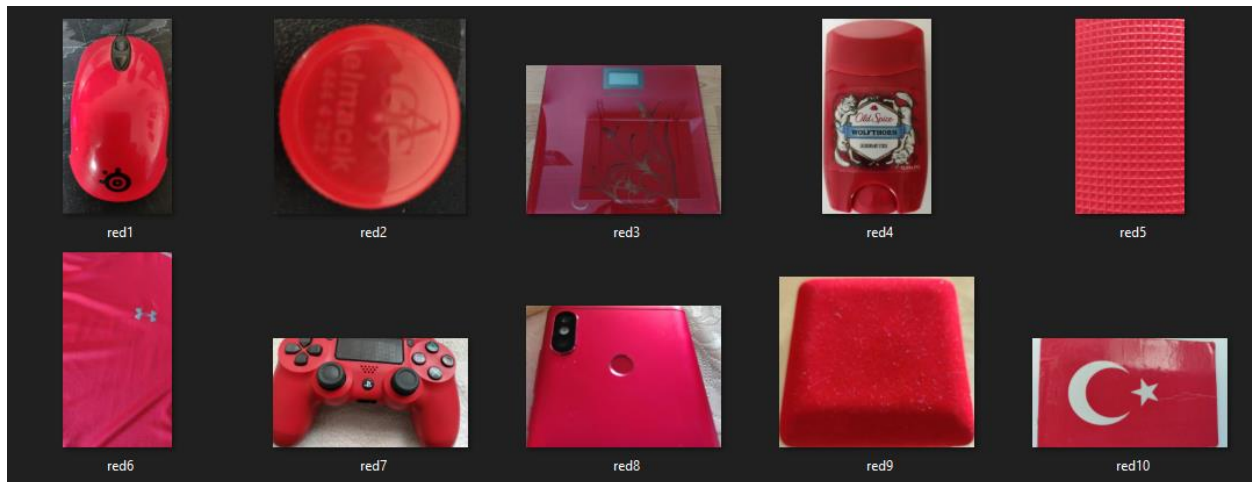


Figure 3: Red colours of test dataset

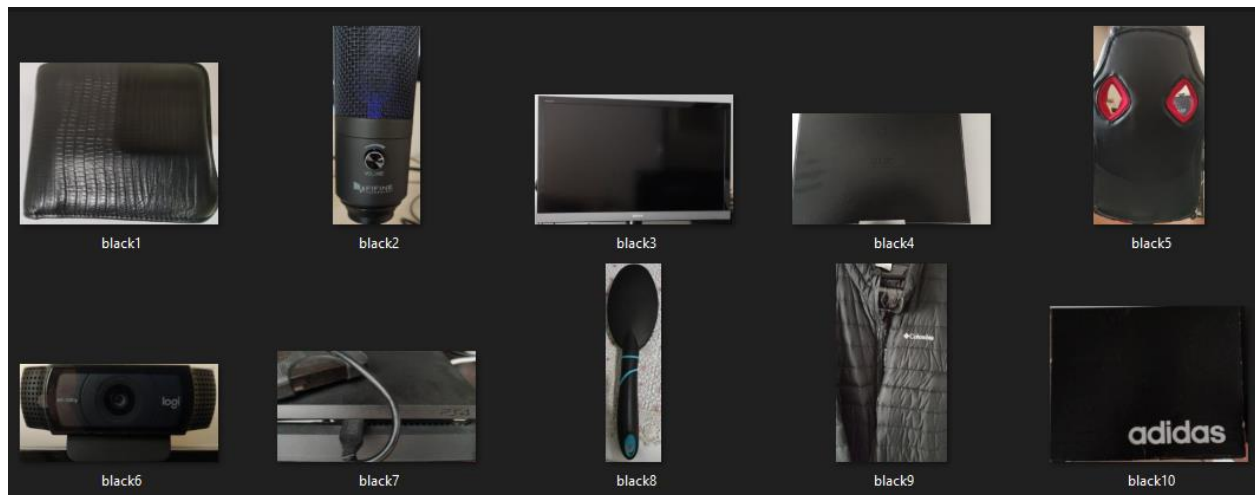


Figure 4: Black colours of test dataset

3-) ALGORITHM

In this assignment, picking the feature extraction method and classification algorithm left to us. I chose colour histogram method for feature extraction and KNN algorithm (K Nearest Neighbour) for classification of corresponding classes.

STEP 1: COLOUR HISTOGRAM FEATURE EXTRACTION

The colour histogram is a representation of an image's colour distribution. A colour histogram represents the number of pixels that have colour in each of the fixed list of colour ranges that cover the colour space of the image, the set of all possible colour, for digital images.

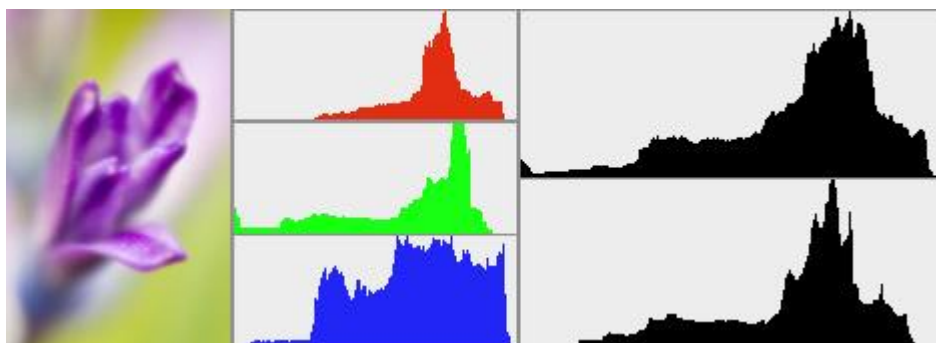


Figure 5: An example of histogram

In this step, I read all the training image inputs in order. Then, using openCV's calcHist function I calculated histogram for each channel (B,G,R). I kept a list for the features and extended it in each iteration. In each iteration, I found the maximum value and appended those values for each channel to my data file called "trained.data". Additionally, the program checks if this .data file exists before and deletes it from the operating system because of previous logs coming from previous run.

In this step parameters of function is checked as well. There can be two types of parameters in the program: training input or test input. If it is test input results for 3 channels are appended to different data file called “tested.data”. Also, we keep labels of test inputs for the future comparison.

STEP 1: K NEAREST NEIGHBOUR CLASSIFICATION

KNN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. (e.g., distance functions).

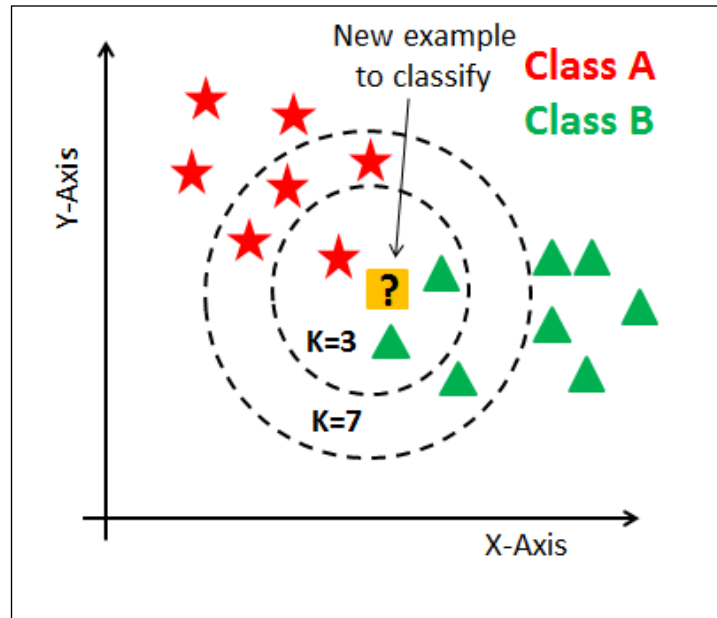


Figure 6: An example of KNN

In my program, I implemented some helper functions to make KNN algorithm worked. A function that calculates Euclidean distance based on the formula is one of them.

$$E(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2}$$

Also, knn function calls calculateEuclideanDistance function to find distance between testImage and trained feature. Then, it sorts the distance in ascending order, and it returns a neighbour list based on the k value. In my program I picked 3 for k value. Finally, responseOfNeighbour function is used to find votes of neighbours.

4-) RESULTS & DISCUSSION

The program outputs the name of the test images and compares it with the prediction result as follow.

```
C:\Users\Terminou\AppData\Local\Programs\Python\
Image name      |      Prediction
-----
red1.jpeg       :      red
red10.jpeg      :      red
red2.jpeg       :      red
red3.jpeg       :      red
red4.jpeg       :      red
red5.jpeg       :      red
red6.jpeg       :      red
red7.jpeg       :      red
red8.jpeg       :      red
red9.jpeg       :      red

black1.jpeg     :      black
black10.jpeg    :      black
black2.jpeg     :      black
black3.jpeg     :      black
black4.jpeg     :      black
black5.jpeg     :      black
black6.jpeg     :      black
black7.jpeg     :      black
black8.jpeg     :      black
black9.jpeg     :      black

orange1.jpeg    :      orange
orange10.jpeg   :      red
orange2.jpeg    :      yellow
orange3.jpeg    :      orange
orange4.jpeg    :      orange
orange5.jpeg    :      red
orange6.jpeg    :      orange
orange7.jpeg    :      orange
orange8.jpeg    :      orange
orange9.jpeg    :      orange

yellow1.jpeg    :      yellow
yellow10.jpeg   :      yellow
```

Figure 7: Image name and prediction output

Then the program outputs the results in a statistical way and shows the accuracy of the correct classifications.

```

Comparing results...

Class | Correct prediction / Total tests | Accuracy
-----|-----|-----
Red   | 10 / 10 | 100.0 %
Black | 10 / 10 | 100.0 %
Orange| 7 / 10  | 70.0 %
Yellow| 8 / 10  | 80.0 %
Green | 8 / 10  | 80.0 %
Blue  | 9 / 10  | 90.0 %
White | 8 / 10  | 80.0 %
Violet| 7 / 10  | 70.0 %
Total | 67 / 80 | 83.75 %

```

Figure 8: Accuracy of each class

The experimental result for the 8 classes is shown in Table 1.

Colour	red	black	orange	yellow	green	blue	white	violet	Average
# of training images	25	25	25	25	25	25	25	25	25
# of test images	10	10	10	10	10	10	10	10	10
Correct classifications	10	10	7	8	8	9	8	7	8.375
Incorrect classifications	0	0	3	2	2	1	2	3	1.625
Accuracy percentage	100	100	70	80	80	90	80	70	83.75

Table 1: Obtained results from the program (result of test images)

The accuracy in general is approximately 84% which is considerably acceptable for classification programs. Because of the camera lens, lighting conditions or lack of training images some colours are classified incorrectly. Yet, the classification results are close. For example, there 3 incorrect detection for class orange. Two of them are classified as red and one of them is classified as yellow. Note that orange is a colour close to both red and yellow. Hence, we can conclude that using colour histogram with KNN algorithm is successful in general.