BLG 513E Image Processing Homework - 3

Kemal Alperen Çetiner

December 24, 2023

Problem

In this problem, we are asked to classify the following colors: Red, Black, Orange, Yellow, Green, Blue, White, and Violet. Color classification refers to the process of categorizing colors so that this information can be utilized to ease other processes such as image retrieval and color constancy estimation.

Figure 1: Colors in the assignment

In this project intended approach is to solve this problem by making use machine learning techniques.

Solution Process

In order to cope with the problem using machine learning approaches, the initial task is to understand the nature of the problem. Color classification problem has the following challenges:

- Challenges due to object
  - Objects may contain several colors
  - Objects may be occluded
- Challenges due to observation device
  - Camera may be biased
  - Cameras may have different color encodings
- Challenges due to environment
  - Illumination may change partially or totally
  - Background colors may overwhelm the object color

After examining the challenges of the problem, it is concluded that color histograms and color moments can represent the color information of the object. To this end, the first task is to collect training data.

Dataset Collection

While collecting the dataset, it has been paid attention that objects include variations of colors given for the objective classification.

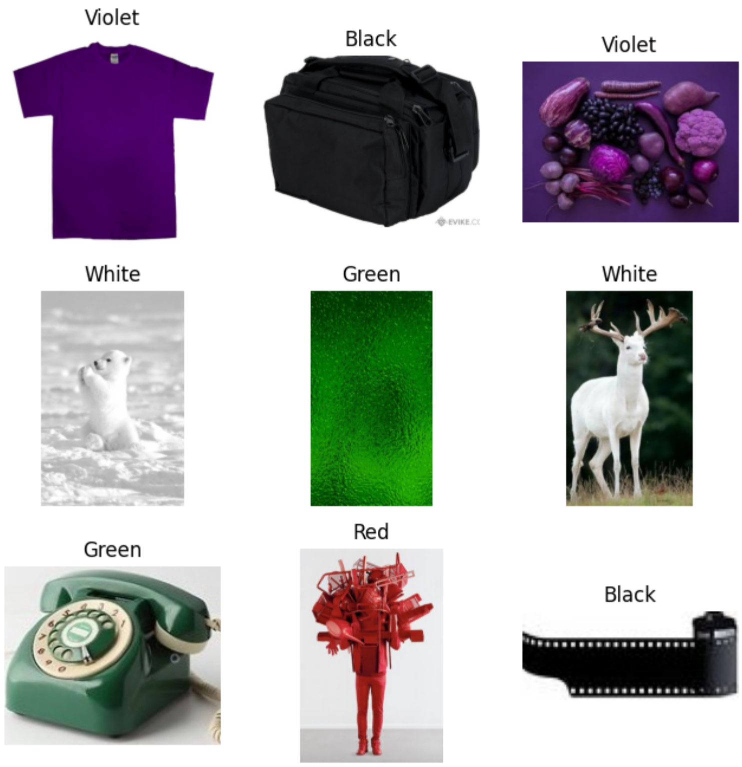

Figure 2: Sample of training dataset

Feature Extraction

Since it is aimed that classical machine learning techniques are to be employed in this project. Features given as inputs to the model are extracted from the image dataset.

Fist chosen feature is color histograms. This is because color histogram is a graphical representation of the distribution of colors in an image and it provides information about the frequency of occurrence of different color values. Color histograms have three hyperparameters namely, quantization size, color space and distance metric. In order to reduce the effects of the illumination changes HS channels of the HSV space are selected. Another consideration is to input the hue information to cosine function since hue is represented by angle while saturation is represented by the distance. Moreover, since we try to categorize colors into only 8 colors (i.e. relatively low number of colors), the number of containers is 8. The final operation for the color histograms is to input them into a softmax function to normalize and compensate for different image sizes.

The other selected feature is color moments specifically the first color moments. RGB color moments are selected in order to provide a rough understanding of the input image color distribution. Moments are divided by 255 to normalize the color moments.

Dataset Creation

After implementing the feature extraction part, we can create the training dataset from collected images. Features are concatenated to create the final feature vector for a given image.

```
...               0           1           2           3           4           5  \
count  255.000000  255.000000  255.000000  255.000000  255.000000  255.000000
mean     0.584627    0.508739    0.518463    0.070588    0.082353    0.101961
std      0.228585    0.211019    0.229275    0.256639    0.275442    0.303192
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.431711    0.368204    0.356760    0.000000    0.000000    0.000000
50%      0.596592    0.512132    0.518394    0.000000    0.000000    0.000000
75%      0.772216    0.637077    0.686297    0.000000    0.000000    0.000000
max      0.998625    0.998625    0.998625    1.000000    1.000000    1.000000

               6           7           8           9          10          11  \
count  255.00000  255.000000  255.000000  255.000000  255.000000  255.000000
mean     0.05098    0.011765    0.023529    0.043137    0.615686    0.482353
std      0.22039    0.108037    0.151876    0.203566    0.487389    0.500671
min      0.00000    0.000000    0.000000    0.000000    0.000000    0.000000
25%      0.00000    0.000000    0.000000    0.000000    0.000000    0.000000
50%      0.00000    0.000000    0.000000    0.000000    1.000000    0.000000
75%      0.00000    0.000000    0.000000    0.000000    1.000000    1.000000
max      1.00000    1.000000    1.000000    1.000000    1.000000    1.000000

               12          13          14          15          16          17  \
count  255.000000  255.000000  255.000000  255.000000  255.000000  255.000000
mean     0.039216    0.019608    0.035294    0.031373    0.043137    0.070588
std      0.194489    0.138921    0.184885    0.174665    0.203566    0.256639
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
...
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      1.000000
```

Figure 3: Dataset Values

Model Selection

While not using CNN's for image processing tasks, SVM's can provide ample opportunities if features can be defined. This is mainly because, SVM's can be adjusted to process errors due to noise and various forms data (e.g. circular, linear, polynomial). Moreover, they provide fast inference rates which can be useful given the color classification task is primarily utilized during image retrieval process. Since there are hyperparameters for SVM's such as C (error bound) and kernel type, we should employ a validation and hyperparameter tuning process to select which SVM to employ in this project.

In order to run the validation process, we have split the training dataset into two sets which are train set and validation set. We train the model in train set and observe the validation accuracy to select the model. Results are as follows:

```
C = 1 kernel = linear Validation Accuracy = 0.6406 Training Accuracy = 0.6549
C = 1 kernel = rbf Validation Accuracy = 0.6250 Training Accuracy = 0.6510
C = 1 kernel = poly Validation Accuracy = 0.6250 Training Accuracy = 0.6627
C = 10 kernel = linear Validation Accuracy = 0.6406 Training Accuracy = 0.7176
C = 10 kernel = rbf Validation Accuracy = 0.6406 Training Accuracy = 0.7451
C = 10 kernel = poly Validation Accuracy = 0.6094 Training Accuracy = 0.7686
C = 50 kernel = linear Validation Accuracy = 0.5312 Training Accuracy = 0.7373
C = 50 kernel = rbf Validation Accuracy = 0.5781 Training Accuracy = 0.7843
C = 50 kernel = poly Validation Accuracy = 0.5469 Training Accuracy = 0.7804
C = 100 kernel = linear Validation Accuracy = 0.5156 Training Accuracy = 0.7373
C = 100 kernel = rbf Validation Accuracy = 0.5781 Training Accuracy = 0.8000
C = 100 kernel = poly Validation Accuracy = 0.5000 Training Accuracy = 0.8275
```

Figure 4: Validation Scores

Following the approach that optimizes both complexity and model accuracy, linear kernel SVM with C=10 is selected as the model. It has %64 accuracy score in validation test and %71 accuracy score in training set. After this point model is trained with the full training dataset without no validation split to obtain the final model to be tested.

Model Testing

In order to test the model properly, a test dataset is formed from unseen images found in the web. Since variety of colors weren't present, I wasn't able to use phone camera for the test set.

Figure 5: Test Results

The test dataset accuracy is found to be %55.

Figure 6: Test Accuracy

Discussion

The final model performed with %55 accuracy in the test dataset. Since it has %64 validation accuracy test score seems reasonable. When examining the test results it has been observed to possess following problems:

- Background color can affect the classification process
- Red-orange-yellow colors seem to have similar features.

In order to cope with given problems, we can propose several solutions for further studies. One of the solution can be having more containers during the color histogram process to boost the separability of red-yellow-orange. Second solution could be to include higher order moments of colors to allow model to learn variations in the color moments further. Another solution can be to have different features such as color correlogram and path color information. Least but not the least solution can be to try more complex models to learn the underlying data better.