Sahil Agarwal, Jesse Fei, Rachel Singleton 1

EEL 5840

Assignment 2

Objective:
Using the dataset collected, decide on an experimental design, extract features of the group's choice, train a probabilistic generative classifier model to classify the images.

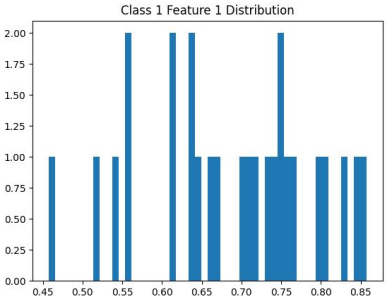
Approach:

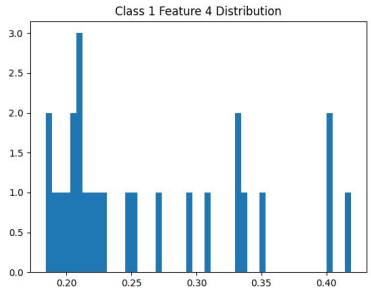
The dataset was split in a manner such that 1/3 of the data would be used for testing and 2/3s of the data would be used for training. This ratio was consistent with information found online regarding general guidelines for dividing a machine learning dataset. For feature extraction, we decided to use features related to the edges of the bricks, as recommended. Modeling wise, we assumed that these features were normally distributed within each class and created a 5-variable Gaussian probabilistic function for each class using the training data. Finally, for testing, we extracted the features from the test data and input them into the generated probabilistic function, returning the class that had the highest probability of having those features for each image's feature vector.

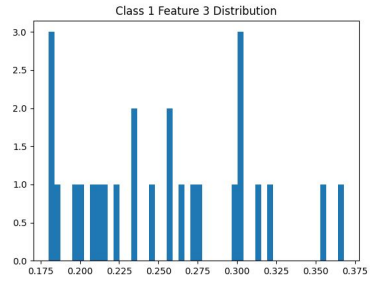
Questions:

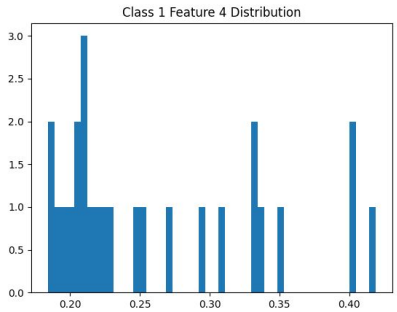
a) What features did you select and implement? Why did you select these features and how do you think they will be informative for this classification problem?

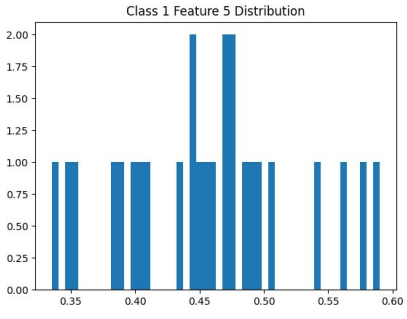
The features we selected were those given by an Edge Histogram descriptor. Specifically, the ratio of edge to non edge areas of the image were extracted with respect to each of the following edge types : 45°, 135°, 0°,90°, and no edge(non edges / total edges). These features were selected because they are common in machine learning solutions related to image recognition. In this context, the orientation of the brick edges is also how we as humans recognize and distinguish brick patterns. Additionally, we believe that these features will be effective in distinguishing "other" brick patterned and "non-brick" patterned images, as they will likely have a completely different distribution of edge ratios in comparison to the brick patterns of class 1-3.









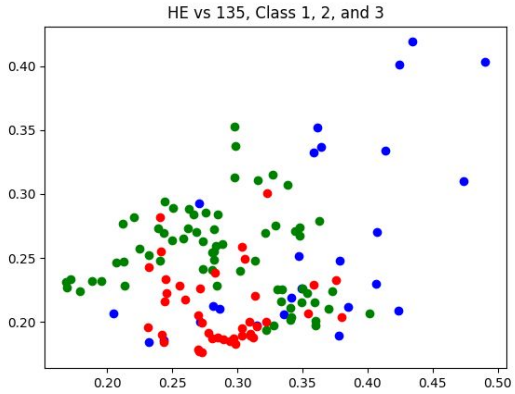


The following images show the distribution of features extracted from class 1. These features were assumed to be normally distributed under the condition that more data points were provided

b) Given your data set and features, were there any outliers in your data set that needed to be removed or handled specially during pre-processing/feature extraction? If so, what was causing these images to be outliers? If not, do you think your data set is a good representation for the problem or are you missing imagery of a certain type in your data set that could impact your overall performance?

As seen in the graphs of our feature sets, no outliers were identified that needed to be removed. Our dataset, however, was extremely limited and infact only contained one example of a class 0 image. For this reason, we believe that our dataset is not a very good representation of the overall problem. Although we were able to generate probabilistic models for each class, the accuracy of these models is questionable at best. This negative impact could be mitigated by training on a larger dataset.

The shortcoming of the dataset can also be seen through the overlap of feature distributions across classes. In the image to the right, features 2 and 4 are compared across classes 1, 2, and 3. Although the class groups can be made out, these groups would likely be more distinct if more data was given. Furthermore, although it can be said that the image below only shows a comparison of two features, rather than all 5 features being considered. The features chosen for the image have means furthest from each other in comparison to all other features. This fact highlights the problem which overlap may be adding to the overall model.



c) How effective is your trained system (preprocessing + feature extraction + classifier)? Do you think you already have a system that would be competitive/effective for the project overall? If so, why? If not, what is needed to improve your approach?

It is hard to determine how successful our system is because the datasets which we trained and tested on were so small. This system will need to be trained over significantly more data before we can make a true assessment as to how well the system works. With that in mind, over our very limited dataset, the model performed well as shown in the confusion matrices below. The preprocessing and feature extraction process of using edge descriptors seemed to provide sufficient data for the model to become relatively accurate when validated against testing data. We were able to achieve an overall accuracy of 78% percent, with specific errors shown in the confusion matrix below. Although this accuracy is measurably better than random assignment, we do not believe that it is enough to be competitive for the project overall. In order to improve, more data must be used to train and test the model. Fine tuning of the threshold parameters used in the edge histogram may also benefit the overall accuracy of the model

```
Confusion Matrix:
[[ 0.  0.  0.  0.  0.]
 [ 0.  8.  1.  2.  1.]
 [ 0.  5. 26.  3.  1.]
 [ 0.  2.  6. 12.  2.]
 [ 0.  1.  1.  0.  5.]]
```

Columns 1 through 5 correspond to the ‘actual’ number of data points in class 0 through 4 respectively. Rows 1 through 5 correspond to the ‘predicted’ classes 0 through 4 respectively, given by the model

Final Observations:
The class which we had the most examples for, class 2, got the most correct in the testing, which is shown in our confusion matrix. This leads us to believe that given more data points for each class our model will improve.