Project 2 CMSC426

Andrew Lee
March 26, 2021

1 Find SIFT Features

I used sift.detectandcompute() on each image and appended the feature array and the feature descriptors array to overall arrays that contain all of the features and all of the feature descriptors. I also saved which features and feature descriptors are related to which image using a hash, and also recorded which class each image belonged to in a list.

2 Clustering

I used KMeans fitted to the array of all the feature descriptors in the training images.

3 Form Histograms

I made a function that takes in a file path, and then uses it as a key in the feature/feature descriptor hash to retrieve the features and feature descriptors. I did this to avoid having to use SIFT again, thinking SIFT was nondeterministic. I then build the histogram by using .predict to find the closest cluster in the KMeans clustering for each feature descriptor, and then incrementing the value at the index of that cluster by 1. Afterwards the histogram is normalized by dividing by the amount of feature descriptors. The function is then used to create a list of histograms for all the images.

4 Preparing for Classification

All of the steps for the Training dataset is repeated, except for KMeans, where we substitute in the training set's KMeans clustering instead.

5 Classification

I used sklearns KNeighborsClassifier fitted to the X values of the array of training histograms and the y value of the list of image classes of the training images created in the SIFTing process to create the classifier. With the classifier, we iterate through the list of testing histograms, using .predict on each one and comparing its result to the actual class of the image. Here are the results of K=1 Nearest Neighbor Classification:(actual on left, predicted on top)

Overall Accuracy: 67%			
Classes	Butterfly	Hat	Airplane
Butterfly	90%	10%	0%
Hat	40%	30%	30%
Airplane	19%	6%	75%

6 Linear Support Vector Machine

I used makepipeline() with LinearSVC() fitted to the list of training histograms and the list of training image classes, and then performing the same prediction process as above. The results:(actual on left, predicted on top)

Overall Accuracy: 77%			
Classes	Butterfly	Hat	Airplane
Butterfly	100%	0%	0%
Hat	30%	60%	10%
Airplane	19%	6%	75%

6.1 Kernel Support Vector Machine

Same process as before, just starting with makepipeline() and SVC(kernel=rbf) fitted to the list of training histograms and the list of training image classes. Results:(actual on left, predicted on top)

Overall Accuracy: 92%			
Classes	Butterfly	Hat	Airplane
Butterfly	90%	10%	0%
Hat	10%	90%	0%
Airplane	0%	6%	94%

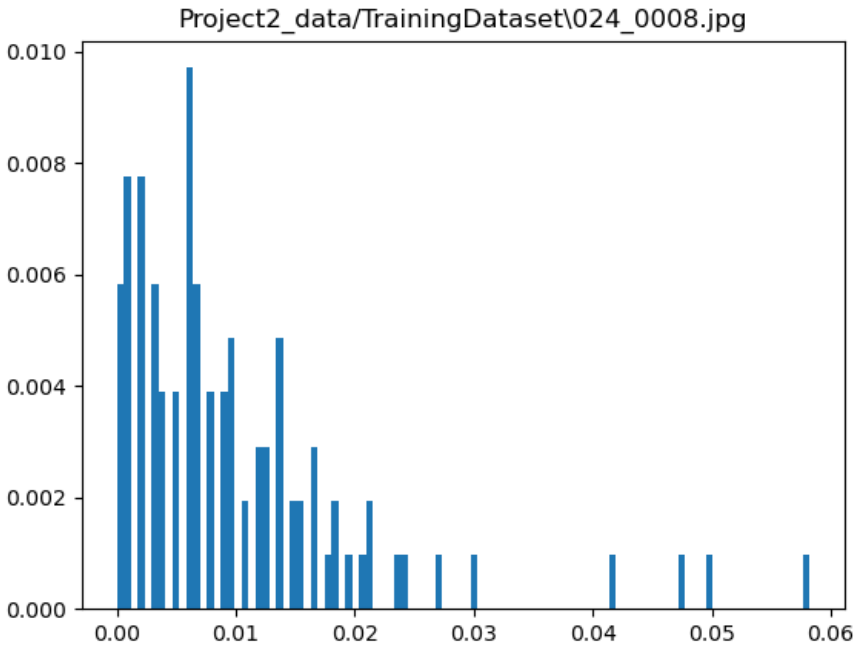

Figure 1: Sample Butterfly Histogram

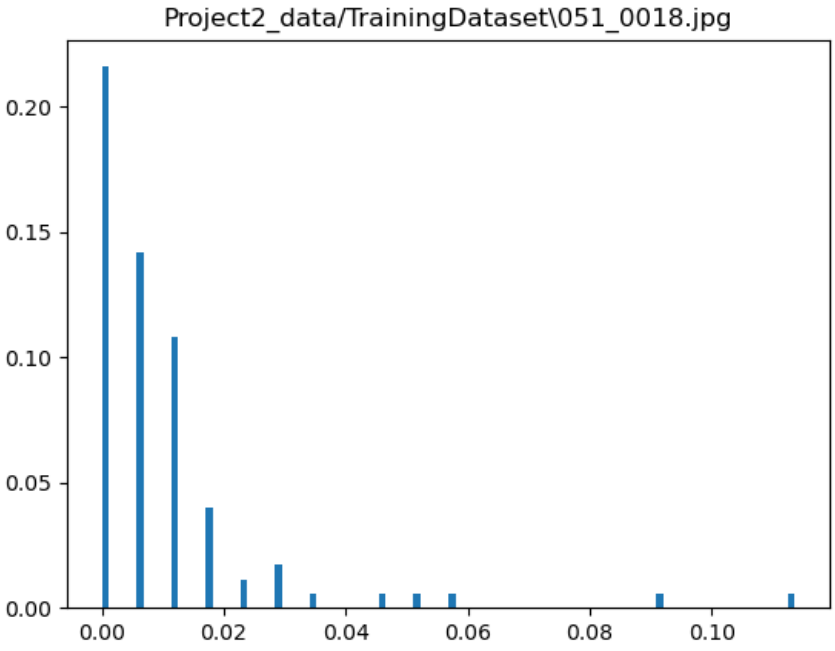

Figure 2: Sample Hat Histogram

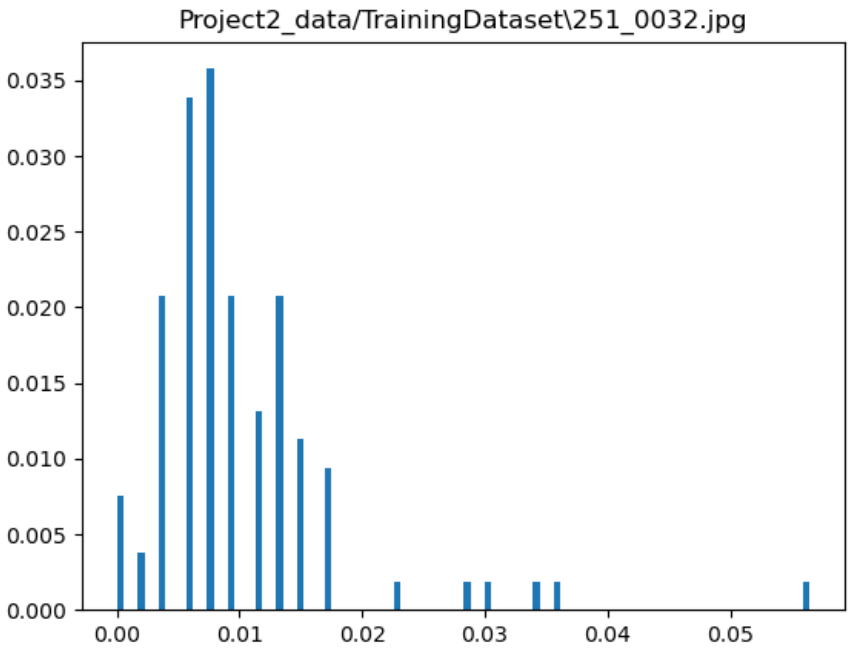

Figure 3: Sample Airplane Histogram