**Assignment 3**

In this Assignment our task was to perform the KNN classification on the given image dataset. The problem contains 30 different classes (types) of images. And our task is to design a knn model which will successfully predict the type/ label of the image. We have done this assignment in Matlab and used some inbuilt matlab functions to complete this assignment.

There are four different files I have used and uploaded in this zip train, test, feature extraction and knn.m. Train is used to train the model and test is used to test (validate) the model created in train. Feature extraction is used to extract the features from the images and then send those to knn for creating the knn model.

Initially I read all the images in train and code for this reading the images is in the train function. Then each images will be send to feature extraction function for extracting the features. I used **detectMinEigenFeatures** function to detect the extraction with sending parameters as image and filter size as **3.** Default is 5 but I gave 3 as a Gaussian feature dimension as Gaussian filter smooths the gradient of the input image. Using this feature detection I was able to detect corners using minimum eigenvalue algorithm and return corner Points object. And this helped to improve the accuracy of my function.

|  |  |  |
| --- | --- | --- |
| Detection Function | Args | Accuracy |
| **detectMinEigenFeatures** | Filter 5 | 32 |
| **detectMinEigenFeatures** | Filter 3 | **36** |
| **Detect Surf** |  | 22 |
| **Detect Surf** | Metric Threshold 700 | 26 |
| **Detect Harris** |  | 20 |

So after using different feature detection techniques I was able to get 36% knn accuracy on the given validation dataset.

To extract the features which are detected by this function I used function name called **extractFeatures** and passed the args called method and method name as SURF. It gives me features and points. Then I took the mean of all the features. And send it to the knn now each image will have only one feature in knn which is given by the mean.

And then I trained the knn with this features and after testing on validation I was able to get 36% accuracy.

