**HOTEL IMAGE CLASSIFICATION**

**Before running the program below requirement should be met:**

1. **Prerequisite Packages Required**:

* Anaconda
* OpenCV(3.1)

Library Used:

***sklearn for hog and svm***

1. **Path and Directory Settings:**

Change the OS directory path for the following method in the code.

* **generateData (Path for train.csv)**

Change the path for the location of train.csv

Current path is **C:/Users/Gaurav/Downloads/**

* **generate\_hog:**

Change the OS path and provide it for the train directory path

Path for train images ***C:/Users/Gaurav/Downloads/train/***

* **predicttestdata: (Test Images Path)**

Current Path Set: ***C:/Users/Gaurav/Downloads/test/***

Set this path to the directory path of test images

* **generatePredictionCSV (CSV generation path)**

Change the path where the CSV will be generated

Current Path for CSV generation: ***C:/Users/Gaurav/Downloads/submission.csv***

Change to the path where the CSV will be generated.

1. **Program Execution**

Post setting the above configuration and setting the paths for train, test directory and for the train.csv.

Run the code using IDE(JetBrains)

***python project.py***

**Important Notes:**

There are four main methods invoked from the main method:

**1. generateData**: This generates the XTrain, YTrain from the train.csv file.It also generates the cross validation set for the classifier

**2. generate\_hog**: It returns the extracted feature vector for the test data. It uses HOG(Histogram of Oriented Gradients for doing feature extraction ) .The images are first resized to a normal format. For now, we identified set of faulty images which are now excluded using the faulty list and these images will be excluded.

**SVM classifier** is used to fit the train data generated from the generate\_hog method and once the model is fitted the prediction can be made from the generated model

We will do the prediction of the cross validated data using ***SVM.predict*** function and then the score is calculated on that basis.

**3. predicttestdata:** This method takes the test images and generates hog and do feature extraction of all the test images, the classifier will predict the labels for each image and will be appended to a list to generate xtest and ytest (prediction generated from the classifier) list

**4. generatePredictionCSV:** This is the last step where the xtest and ytest generated above are written to a csv file.

**Two different versions of code are used to generate the CSV.**

**1)** Initially used the KNN classifier where the predicted model accuracy was varying between ***34%-40%***

**2)** Replaced the KNN classifier with the Linear SVM classifier with different C values which gave an accuracy approximately ***60%-70%.***

The latest version of the code and Kaggle Submission is done using the SVM classifier.

***Note:***

* Out of total sample of ~38k images 35k images are used to train the model and 3k images are used for testing purpose to do the cross validation.
* The faulty images from the train directory are kept in a list and are not iterated while generating the trained data.