CS 165B – Machine Learning, Summer 2020

Final Project

Due *Sunday, September 13, 2020 11:59pm*

1 Visual Bag of Words

In this final project you will combine the knowledge from the lectures + homeworks to create on your own a working, real-life application for image recognition using machine learning methods.
We will be performing classification of Caltech-101 dataset. Description of the dataset is available here and the download files are available here. The task is to use SIFT features (using OpenCV) to extract useful information (features) from every image. Description of SIFT and examples are available here and here. We can later on cluster these features into some groups with a similar meaning (using K-Means) e.g. windows, trees, etc. By doing so we would create a dictionary of visual features (e.g. feature [0, 5.1, 6]=window; feature [2, 1.1, -0.4]=tree). Such action will allow us to describe each image with information on which types of features (windows, trees, etc.) are present in it and how many of them there are). Such a description can be later used as an input to train a classifier to recognize objects on a specific image. In this project you will use an SVM to perform the classification.

2 Details

You will have to fill in the functions supplied to you in the `final.py` file. The main function is written for you already (both `if __name__==”__main__”` and `self.algorithm()`). The rest of the functions is already created with the description of their purpose, inputs and outputs. You can see the order they will be executed in the `self.algorithm()`.

The program should read the data, split it randomly into train/test, extract SIFT features, group them, train SVM on the train set and test its accuracy on the test set. Additionally, you will have to perform the PCA on the test features and plot the test dataset in 2D (2 first Principal Components). Save the plot.

You are allowed to use the existing implementations of the ML algorithms from Scikit-Learn (as included in the header file).

3 Project structure

Shown below is the expected project structure (that will be used to test your code).

```
Project_folder:
      -  final.py
      -  101_ObjectCategories:
            o  accordion
            o  airplanes
            o  …
```

4 Project report

In addition to making the code work, you are expected to create a PDF report where you will submit and discuss your findings (runtimes, accuracies, conclusions and mentioned plots) for following conditions: since the method relies on 2 hyperparameters (k – number of SIFT features/keypoints extracted from every image, and dictionary_size – size of our visual dictionary or simply k in k-means algorithm part), test your method with 3 values of k (2, 10, 20) and 3 sizes of dictionary (10, 50, 200) for a total of 9 experiments.

5 Grading

The code must work on CSIL or you will receive 0% for the implementation.

Grade Breakdown:

- 20% for correct SIFT feature extraction
- 20% for correct dictionary creation
- 20% for correct SVM training/testing
- 20% for correct PCA/plotting
- 20% for the final report

6 Extra credit

After writing the code, creating the report and performing all comparisons and analysis – you can save that version of your code (to be as it was used to create the results present in your report) and start working on possible modifications (without changing the overall approach, so sticking to SIFT, k-means and SVM). 5 students with highest reported accuracy will be awarded additional 10% extra credit. If willing to do that, then report your highest accuracy in the report and submit a separate python script `final_mod.py` with your submission.

7 Submission

Submit your solution (code and the PDF report and possible extra credit code) to Gauchospace (as a single ZIP file).