McGill University
ECSE 415 - Intro to Computer Vision
Assignment #4 - Due 11:59pm - November 24th, 2017

Preface

In this assignment, you will obtain a deeper understanding of using Bag-of-Words for object classification. The following assignment has two parts. In the first part you are required to create two codebooks (one for SIFT features and one for HoG features) to describe a set of training images from different object categories. In the second part you are asked to evaluate the performance of the Bag-of-Words method in classifying different object categories. Please submit your assignment solutions electronically via the myCourse assignment dropbox. The solutions should be in **Jupyter Notebooks** format, see tutorial #2 for a reference. The complete assignment submission should be a zip file containing your Jupyter code (.ipynb) in addition to all output images. Attempt all parts of this assignment. The assignment is out of a total of **45 points**. The student is expected to write his/her own code. Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be marked.

Problem 1 - Bag-Of-Words [30 points]

For this question you will be computing Bag-of-Words features on a portion of the Caltect 101 dataset [1]. The original dataset contains pictures belonging to 101 different categories ranging from animals to cartoons. For this assignment, however, we will choose only five categories. Before implementing the Bag-of-Words features you should first partition the dataset. You should use $\frac{1}{4}$ of each image category as your testing set, $\frac{1}{4}$ as your validation set, and the remaining as your training set. For all hyper-parameter fine-tuning you should calculate accuracies on the validation set and tune your algorithm accordingly. It is not until the very last step, i.e. you have settled on a model, do you use the test set. To compute the Bag-of-Words features you will be following the pipeline described in Lecture 10 slide 12. For the feature extraction stage, you will be testing two different features, namely, SIFT and HoG. You are allowed to use opencv functions to compute the respective features. To learn the visual words, you will be using the Kmeans algorithm. You may use the Kmeans algorithm you implemented in Assignment #3 or you may use opencv's function. The number of clusters (K) is a hyper-parameter and should be adjusted to provide the best on the validation set. The last step of the algorithm requires you to represent the image features via a histogram. You should normalize the final histogram. The result is two different Bag-of-Words codebooks one for SIFT and one for HoG features. For each value of $K = \{25, 50, 100\}$ you should display 5 training images from each of the 5 categories and their respective Bag-of-Words histograms. You must do this for both codebooks. Figure 1 demonstrates a sample output.

Problem 2 - Image Classification [15 points]

To classify a new test image in the Bag-of-Words methodology, we will use the learned codebook to compute a Bag-of-Words histogram representation for the input image. You will then compare

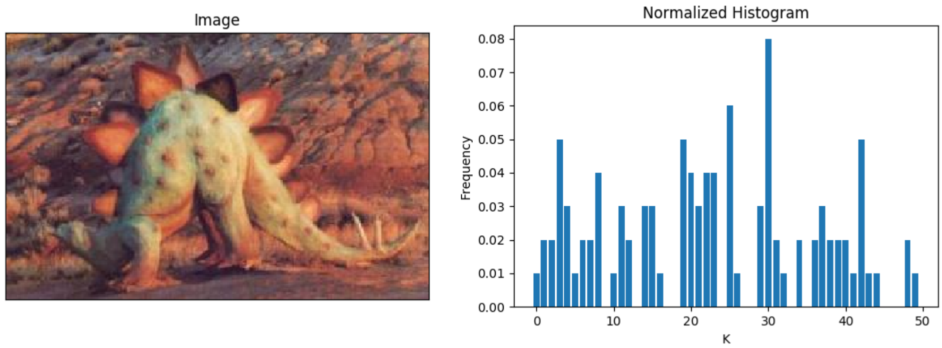

Figure 1 - Bag-of-Words Feature Representation

it to a set of criteria to decide which category it belongs to. For this process of classification we will be utilizing a Support Vector Machine. Similar to what you have seen in tutorial 9, you will be training the SVM using the Bag-of-Words representation and category labels for all training data. You may use *scikit-learn*'s SVM class for training and validation. Since we computed two different codebooks, you will have two SVM models to fine-tune. Once you have settled on the two best models (top accuracies on the validation set), you should now calculate and record the overall accuracy on the testing set. You should display your validation accuracies and the final test accuracy for each codebook. Comment on which codebook performed better and justify your answer.

References

[1] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004