CSE468: Computer Vision
Assignment 1
Due: 9:00am, 14-02-2019

Dr. Nabeel Mohammed

1 Introduction

For this assignment, you will be required to implement and analyze a naive *1-NN* based image classifier's performance when using different features.

2 ***NaiveClassifier***

Create a class $NaiveClassifier$ which accepts the following in its constructor (*__init__*) in the following order

1. Training images as a *(number_of_images, number_of_channels, x_dim, y_dim)* `numpy` array. For now assume every image in the training set should have the same number of channels and the same dimensions.
2. Training labels. This should be a *(number_of_images, 1)* `numpy` array.
3. An object of type *FeatureExtractor*

You should also implement the following methods. The names are self explanatory.

1. *extract_feature_from_single_image*, which should accept an image as parameter. The image is a *(number_of_channels, x_dim, y_dim)* `numpy` array. This method should return the feature extracted by calling the *extract_feature* method of the *FeatureExtractor* object.
2. *extract_feature_from_multiple_images*, which should accept a *(number_of_images, number_of_channels, x_dim, y_dim)* `numpy` array as parameter and return a *(number_of_images, feature_dim)* array.

3. *classify_single_image*, which will accept an image as parameter and return a label value and the corresponding score.

4. *classify_multiple_images*, which will accept multiple images as parameter and return a list of label values and a list of scores.

3 ***ClassificationMetrics*** Class

This class should have the following methods:

1. *__init__*: This method should accept a list of training labels and a list of testing labels as parameter. It should also construct the confusion matrix and cache this information.

2. *get_confusion_matrix_for_heatmap*: Returns a $0 - 1$ normalized confusion matrix.

3. *calculate_accuracy*: Calculate and return the accuracy of the predictions. It should return a value in the range $0 - 1$.

4. *calculate_precision*:Calculate and return the average precision.

5. *calculate_recall*:Calculate and return the average recall.

6. *calculate_f1*:Calculate and return the F1 score.

7. *calculate_roc_values*:This method should accept a list of thresholds as parameter and then calculate and return the *ROC* values in a list of lists. A list per threshold value per class.

8. *calculate_lift_values*: Similar to *calculate_roc_values*, but calculates $lift$ values.

4 ***FeatureExtractor*** class

This is a very simple class which has a single method *extract_feature*, which accepts an image `numpy` array as parameter, extracts the feature from it and returns the image descriptor as a `numpy` array.

For the *FeatureExtractor* class, the *extract_feature* method should simply return a vectored form of the image.

You should also implement the following sub-classes of *FeatureExtractor*

1. *ColourHistogramExtractor*
2. *HoGExtractor*
3. *SIFTBoVWExtractor*: The constructor of this class should accept a *(number_of_images, number_of_channels, x_dim, y_dim)* `numpy` array. You should use this to calculate the visual words while constructing the object.

5 Documentation

You are required to write a report with the following sections.

1. Introduction: Overview of your work and results.
2. Literature Review: Detailed, wherever possible mathematical, description of *1-NN*, *Metrics*, *Features*.
3. Experimental setup: Description of the data set, how it was partitioned, how the comparison of the performance of the different features were done and what metrics were calculated.
4. Resutls and discussion: Presentation of the results in a sensible and readable format. Just presentation is not enough. You will need to analyze the results, especially failure cases on a per-feature basis and provide your justified insights why things work and why things do not work.
5. Conclusion: The conclusion of your report

Please note that to write the report you will be required to perform the classifications using the different features. This will require some driver code. You should be writing the driver code and submiting it as well. Please also note the report must be written in LaTeX. Do not forget to include your names as authors of the report.

6 Submission Instructions

Submission is due by 9am of the 14-02-2019. Your code,documentation (PDF and LaTeXsource) should be zipped in a single zip file and emailed to me at *nabeel.mohammed@northsouth.edu* with the subject line "CSE468

Spring 2019 Assignment 1”. Please mention your group member names and student id numbers in the body of the email.

Also, please note that as discussed in class, you should be writing all the code yourself, without using the help of libraries. The only exception is the use of image loading/writing API and the `numpy` library. You can also use graph plotting libraries only for the purpose of plotting graphs.