SIT789 - Applications of Computer Vision and Speech Processing

Pass Task 4.3: Build your own image recognition system – group task

---

Objectives

The objectives of this lab include:

- Building an image recognition application based on SIFT features and BoW model
- Data collection and processing
- Working in group

Tasks

In this task, you are required to build an image recognition application using SIFT features and BoW model. This is a **group** work and you can make groups yourself. Each group should include 2 – 3 students.

1. Data collection and annotation

You first need to define your objects of interests; you should have at least 4 object classes of interests. Based on your chosen classes, you need to collect a dataset including at least 100 images per object class. Your dataset should cover different variations of the objects, e.g., different resolutions (sizes), viewpoints, lighting conditions, occlusions, etc. Note that you should not choose images with blank background and/or in too small size (e.g., less than 200x200 pixels) as SIFT may not extract any keypoints from those images.

You then divide your dataset into three subsets: training, validation, and test sets with the proportions as ~40% for training, ~30% for validation, and ~30% for testing. For example, if you collect 100 images per object class, then you will have 40 training images, 30 validation images and 30 test images. You may organise your dataset in the same way as the FoodImages dataset used in Task 4.1.

The training set will be used to train classifiers, e.g., SVM, AdaBoost. The validation set will be used to tune hyperparameters that are not trainable, e.g., the parameters C in SVMs, the number of weak classifiers in AdaBoost algorithm. You can find more about hyperparameters at https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning). The test set will be used only for testing your image recognition application.

2. Building image recognition application

2.1. Building BoW model

Like Task 4.1, you will need to use SIFT features and BoW model to build your image recognition application. Recall that in the BoW model, words are determined using the K-means algorithm where each data point is a SIFT descriptor. The number of clusters (i.e., K) must be predefined. There exist several ways to find a good value for K. In this task, you will need to explore two common techniques including elbow method and silhouette analysis. In general, given a range of possible values for K, e.g., [50, 500], those techniques help you identify an optimal value for K in the given range. You can read more about these techniques at https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

Both the elbow method and silhouette analysis have been implemented in sklearn (see https://www.scikit-yb.org/en/latest/api/cluster/index.html). One technique may not work well, e.g., the elbow method may not give you a clear decision on the value for K, and therefore, both techniques should be considered.

To visualise the clustering results, you can use "intercluster distance maps" which represents the cluster centroids in a 2-dimensional space. You can find details on how to generate intercluster distance maps for a clustering algorithm at https://www.scikit-yb.org/en/latest/api/cluster/icdm.html. Intercluster distance maps could be used to assess the quality of your clustering algorithm, i.e., the more separated the centroids are, the better the clustering algorithm is.

Note: the elbow method and silhouette analysis are only applied on the TRAINING set. Suppose that K* is the best number of clusters selected by using the elbow method and silhouette analysis on the TRAINING set, K* then will be used in the K-means algorithm for building a dictionary of words, and for calculating the word histograms of training/validation/test images.

2.1. Building classifiers

Based on the built dictionary, you will define your classifiers. In this task, you are to investigate k-NN, linear SVM, and AdaBoost as classifiers (see Task 4.1). For each classifier, you will need to train the classifier on the TRAINING set and tune its hyperparameters on the VALIDATION set. Specifically,

- For k-NN, you will need to vary num_nearest_neighbours in some range, e.g., [5, 10, 15, 20, 25, 30], and measure the corresponding accuracies on the VALIDATION set to find the best num_nearest_neighbours.
- For SVM, you will need to vary C in some range, e.g., [0.1, 1, 10, 100], and measure the corresponding accuracies on the VALIDATION set to find the best C.
- For AdaBoost, you will need to vary n_estimators in some range, e.g., [50, 100, 150, 200, 250], and measure the corresponding accuracies on the VALIDATION set to find the best n_estimators.

Note: recognition accuracy is the ratio of the number of correct predictions and the total number of test images (see Task 4.1).

2.2. Evaluating classifiers

After training the classifiers with their best hyperparameters selected from Section 2.1, you will need to evaluate the classifiers on the TEST set. In particular, for each classifier, you need to calculate:

- Recognition accuracy, and
- Confusion matrix

You are also required to compare the classifiers based on their recognition accuracy and draw conclusions.
Note: this comparison is also performed on the TEST set only.

3. Documentation

You need to document your application with the following details.

- Provide detailed description of your dataset, e.g., how many classes are used and what are they? Where are the images collected? How many images are there?
- Include some examples of the training, validation, and test images
- Include plots for the elbow method and the silhouette analysis on your clustering results (see examples at https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a).
- Based on the results of the elbow method and silhouette analysis, what is the value chosen for K?
- Report the recognition accuracy of k-NN when varying num_nearest_neighbours on the VALIDATION set
- Report the recognition accuracy of SVM when varying C on the VALIDATION set
- Report the recognition accuracy of AdaBoost when varying n_estimators on the VALIDATION set
- Report the recognition accuracies of all classifiers (with their best hyperparameters) on the TEST set
- Provide the confusion matrices of all classifiers (with their best hyperparameters) on the TEST set
- Declare the contributions of each member in your group

Submission instructions

1. Submit your code including the implementations required in Section 1 and 2 to OnTrack.
2. Submit your document to OnTrack.