CV 2021 HW5 Report

Group 24

0610122 李思賢, 309553049 陳怡安

Introduction

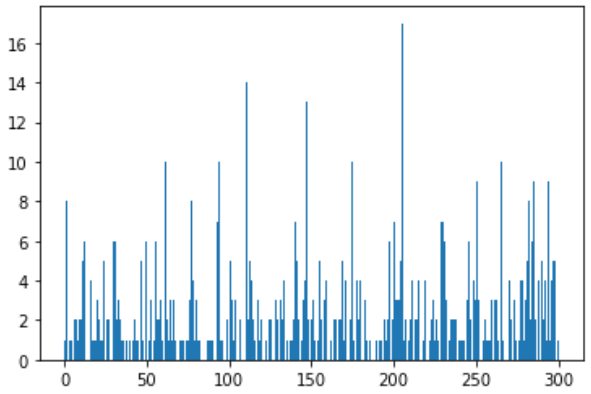
In this assignment, Our goal is to classify data into 15 categories. The data are pictures of different scenes. They are labeled as ‘Coast’, ‘Kitchen’, ‘LivingRoom’ etc.. We use several models, nearest neighbor classifier, linear SVM classifier, and deep learning model, with different means of feature extractions to achieve the task.

Implementation

1. Tiny images representation + nearest neighbor classifier
    a. Tiny images representation
    We resize each image to 16x16, flatten the image into a 256-dimension vector, and then normalize the vector. And the label is represented by the class name of the image. After that, we take the normalized vector and its label as a pair of data.
    b. Nearest neighbor classifier
    For the testing data, we first calculate the distance between each testing data and all training data. Then sort the distance in ascending order, and assign the label with the highest frequency among the first k training data to the testing data.

2. Bag of SIFT representation + nearest neighbor classifier
    a. Bag of SIFT Representation
    First, we resize the images to 256x256, and use Dense SIFT from **cyvlfeat** to obtain the feature descriptions of the keypoints. Dense SIFT is roughly equivalent to SIFT but much faster. We gather the descriptors of all the train data as ‘a bag of SIFT representation’.
    Then we perform k-means clustering on the bag of SIFT to extract the visual words and form the code book. With this we can represent every image by a histogram. For each train data, we use Dense SIFT to extract features and apply **scipy.cluster.vq** to assign them to the closest clusters.
    Finally, we use **np.histogram** to obtain the statistical representations of the assigning results. It gives us outcomes such as the following graph. The x-axis represents each code in the code book, and the y-axis is the

appearance counts of the codes.



b. Nearest Neighbor Classifier
Similar to the last part of the previous task, we compare test data's histogram to that of each train data's. We pick out k test data with the smallest distance, find the class with highest occurrence among them, and label the test data by this.

3. Bag of SIFT representation + linear SVM classifier
   a. Bag of SIFT representation
   This part is basically the same as the part a. in the previous task. We use histograms to represent each data.
   b. We use the svm model from **libsvm**: **svm_train** to train the model and **svm_predict** to get the accuracy and the prediction on the test data. We also perform some parameter tuning.

4. Bonus: deep learning
   We use PyTorch to implement this part. We choose ResNeXt-50(32x4d) as our network architecture, that is we don't use the pre-trained model.
   The size of the images need to be changed to 224x224, and other data augmentations include RandomErasing, RandomHorizontalFlip, and ColorJitter. The loss function is Cross-Entropy Loss and the optimizer is Adam. We also use the learning rate scheduler(ReduceLROnPlateau) to adjust learning rate based on the validation accuracy.

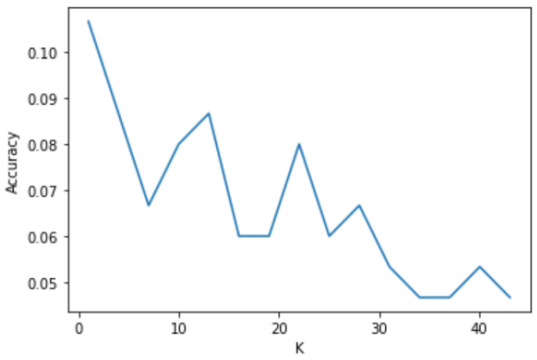
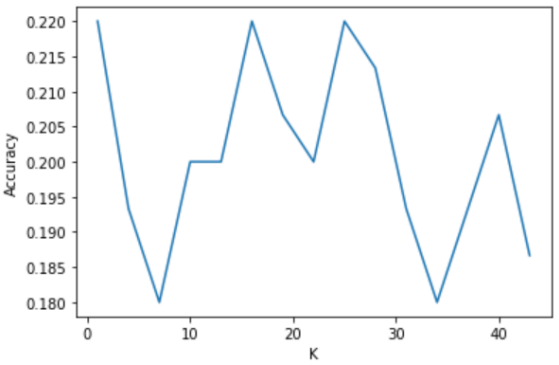
Experimental results

Task	1	2	3	Bonus
Accuracy	0.220	0.573	0.69	0.853

Discussion

Task1:

If normalize the image vectors, the highest accuracy will increase from 0.1 to 0.22.

without normalize	with normalize
	

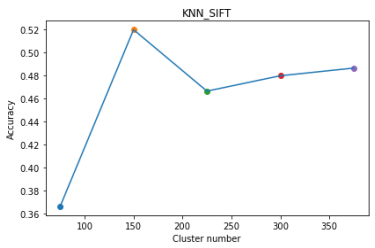
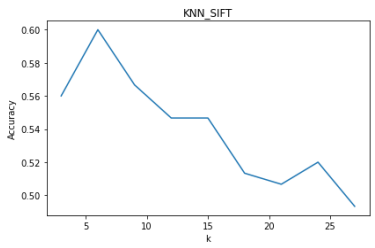
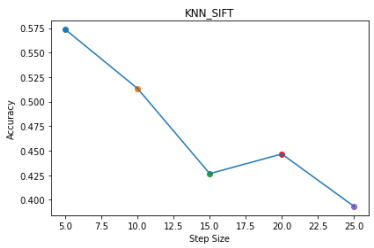
Task2:

We tried different Ks, step sizes and cluster numbers.

The cluster number decides how many clusters k-means will find, which is also the number of code in the code book. Ten times of the class number is a typical choice, and it also gives a good result in our experiment.

Choosing a larger K in KNN can reduce the effect of the noises. However, if K is too large, it will take some irrelevant points into consideration. In this task, it seems that K=6 is a good choice.

The step size is the interval in which Dense SIFT extracts a descriptor. Smaller the step size, longer the runtime. Yet we also get more features and the accuracy tends to be higher.

		
K = 20 step size = 10	step size = 10 cluster number = 150	K = 6 cluster number= 150

Task3:

We tried different step sizes and cluster numbers, cross validation them with C in the range of 2^{-5} to 2^{10}, gamma in the range of 0.001 to 0.02.

It shows that higher cluster number and smaller step size will lead to better results, so we choose cluster number = 300 and step size = 5 on finally training and the accuracy on the test dataset is 64%.

cluster number	300	300	150	150
step size	5	10	5	10
Cross Validation Accuracy	69.33%	64.8%	65.86%	61.33%

Conclusion

We score 22% in Tiny images representation + nearest neighbor classifier, 57.3% in Bag of SIFT representation + nearest neighbor classifier, 64% in Bag of SIFT representation + linear SVM classifier, and 85.3% in ResNeXt-50.
Though there are still some parameters that can be tuned, all the tasks meet the expected accuracy.

Work assignment plan

code: 李思賢 & 陳怡安
report: 李思賢 & 陳怡安