COMP6223 - Scene Classification

Wenhao Wang ww2g18/30624065
Jingxuan Su js5a18/30506107

December 14, 2018

Implementation Decisions

This coursework mainly focuses on image recognition. We achieve this coursework with three simple methods. Specifically, run1 applies the very simple classifier that is k-nearest-neighbours. Run2 develops a set of 15 one-vs-all classifiers based on dense Patch Sampling to classify the test imagine. Finally, run3

Testing the Accuracy

The labeled data provided by training.zip is divided into training (accounting for 90%)and testing sets (accounting for 10%). The training data is used to train the classifier and the test set is used to examine the accuracy.

Classifying Unlabeled Data

All available labeled data from training.zip is used to train the classifier. The fully trained classifier is then used to predict the unlabeled data from testing.zip.

On the Contributions of Team Members

We took steps to ensure all three runs contained significant contributions from each team member. Ultimately, both team members contributed an equal amount of work.

1 K Nearest-Neighbours

K nearest neighbours is a classification algorithm and it is very simple one. The principle is that each image is cropped to a square about the centre and then we resize the new image into 16 by 16 pixels. The image is normalized and the pixel values from each image row are concatenated into a float feature. This is our "tiny-image" feature vector that is one of the simplest possible image representations.

1.1 Testing the accuracy of our classifier

Each row of pixels corresponds to an image's tiny-image feature vector, which is separated in training and test datasets. A function of `FITCKNN` is used, which is training by train test. For every test data, we use the function to retrieve the label of the nearest neighbours found in train set and the class is prediction by the label of neighbours. More specifically, we compare the actual label of image against the prediction label. More specifically, we compare the actual label of image against the prediction label. After all instances of train set have done, we compute the accuracy. The discovery is that 12 neighbours resulted in the best classifier(see the following table), although the accuracy is still very low (approximately 22% for $k = 12$).

Variable K	1	5	10	11	12	13	15
Value(%)	20.22	20.69	21.33	21.67	22.69	21.01	20.69

1.2 Classifier Configuration

Variable	k
Value	12

2 Dense Patch Sampling

In order to implement a linear classifier, we designed a feature extractor. The feature extractor is used to sample and normalise 8x8 patches every 4 pixels in the x and y directions. That means bag-of-visual-words features based on fixed size densely-sampled pixel patches are extracted. These bag-of-visual-words features constitute a feature matrix. The feature matrix by the algorithm K-Means is used to learn a vocabulary. Each vocabulary has a closest centre. When vocabularies of a patch are mapped to the clusters, machines automatically match one of the vocabulary clusters and recognize the image. The specific flow chart is attached on appendix - figure 1.

Noting: Initially, the suggested number of $\sim$ 500 clusters was used, however after testing the classifier with different numbers of clusters, it was found that using $\sim$ 1000 clusters maximized the accuracy.

2.1 Testing the accuracy of our classifier

After the algorithm K-Means, we get the classes. By comparing the classes with the label, we can calculate the accuracy and error of the classifier. When we test the accuracy of our classifier, we choose 90% of training.zip data set as training data and another 10% as testing data.

Noting:

- To achieve more higher accuracy, we compare the test result of 500, 600, 800, 1000 and 1500 clusters. It is appeared that more than 1000 clusters provided no valuable contribution to the accuracy.

Cluster	500	600	800	1000	1500
Value	0.3067	0.3467	0.4067	0.4467	0.3367

2.2 K-Means Configuration

Varbale	Cluster	Patch Size	Grid Size
Value	1000	8	4

3 Convolutional Neural Network

Convolutional neural network is a class of deep neural networks, which most commonly applied to analyzing visual imagery. We use three main types of layers that is Convolutional Layer, Pooling Layer, and Fully-Connected Layer to build CNN architectures. Transfer learning as tool changes specific parameters of layers of CNN realizes the demand of this coursework. Transfer Learning is the reuse of a pre-trained model on a new problem. It is more popular in the field of deep learning because the knowledge of an already trained Machine Learning model is applied to a different but related problem.

3.1 Training and tuning

In this coursework, we compare two approaches to achieve results.

- The first method is that we only use a CNN model named Alexnet [2] to realize image classification. In this coursework, the training.zip data set has 15 scenes. However, we will replace 1000 classifiers of fully-connected layer with 15. To match the CNN architecture, we need to transfer the grey scale image to colourful image.the image size is adjusted to 227 by 227. The accuracy rate is roughly 80%. The final run to produce the file of predictions took approximately 30 minutes.

Network	Alexnet
Value(%)	78.01

- The second method: we use a CNN model to extra the feature of bag- of-visual words. And then do the same steps of run 2 to classify. It is worth noting that we also compare different networks and classifiers to achieve the best-accuracy (see the following table). Finally, we choose the ResNet-50 [1] network and the linear classifier as the best combination.

Network	Alexnet	ResNet-50	ResNet-101	Inception Restnet-v2
Value(%)	75.75	85.43	83.15	84.26

Classifier	Discriminant	SVM	Naivebayes	Linear
Value(%)	83.57	51.26	62.25	85.43

Using the training dataset, the second method achieved a best-accuracy of roughly 85%. Meanwhile, the final run to produce the file of predictions took approximately 20 minutes. Both indexes of second classifier are better than the first one. Thus, we choose the second classifier.

3.2 Classification Configuration

Network	ResNet-50
Classifier	Linear

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Appendix

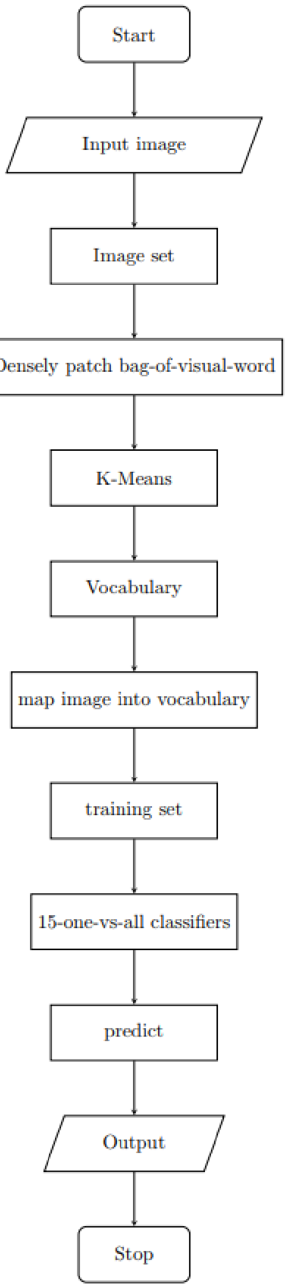

Figure 1: The Flow Chart of Run2

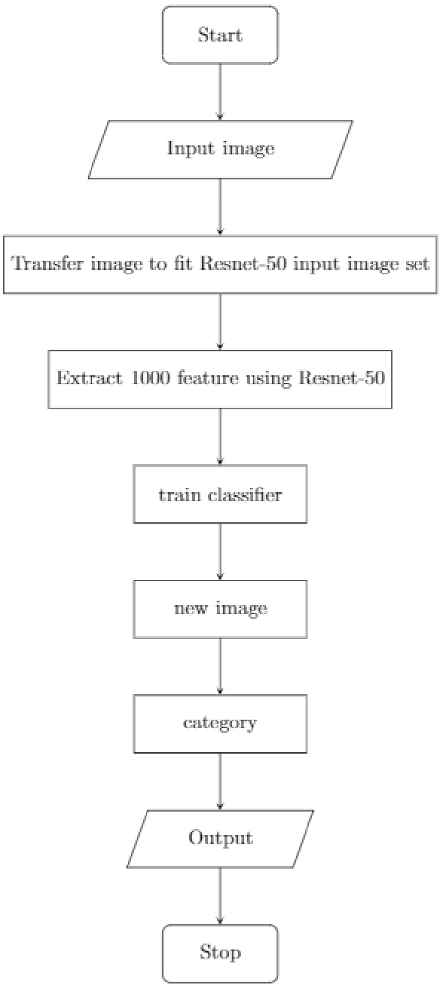

Figure 2: The Flow Chart of Run3