**NAME:LAI WEIZHONG**

**ID:28969367**

**Abstract:**

Each method of classifier includes three steps, feature extraction, classifier training, and prediction of classification.

**Run 1:**

1. **Feature extraction**

In this classifier, first I reshape the picture into a 16\*16 tiny images. Then all the row vectors are concatenated into one vector. Finally, normalizing the vector to make prediction more accuracy(every element minus the mean and be divided by the standard deviation).

* ChangeToOneRow
* Parameter:img
  + the image needed to get the features.
* Return:rlt
  + vector used to keep the features.

1. **KNN Classifier**

the point to be classified looks for the nearest k points, the point will be classified as the class of which the number is the most in the k points.

* **Name:KNN**
* **Parameter:testData,trainingData,k**
  + **testData:** features to be classified.
  + **trainingData:** the matrix made by training set and classification labels.
  + **k:** the value of nearest k.

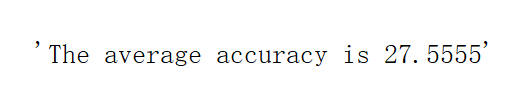
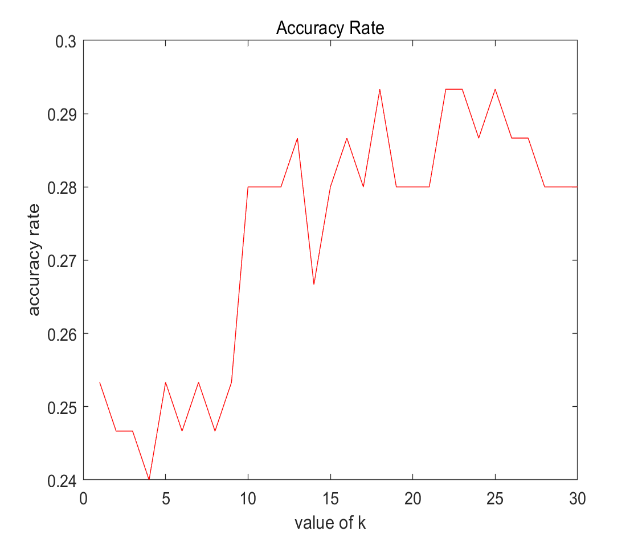
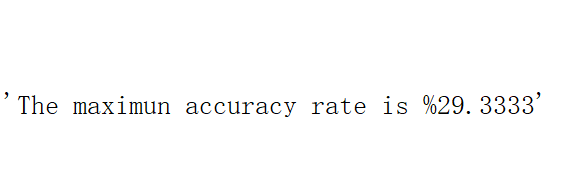
1. **Descripton**:

Construct a matrix: dist. to keep every distance between the test image and training images. That is to say the row of the dist represent the label of classification, and the colum of the dist represent the distance between the test image and the training image.

After that, concatenate every row of the matrix to one row (use the function of ‘ChangeToOneRow’). Then, attached the class label to each element of the first row. Finally, sort the first row in ascending order, and take 1 to k number to choose the maximum number of class, and the class is the prediction of test data.

1. **validation for optimal value k.**

Take a set of training data and test data at a ratio of 9:1 from the training set.

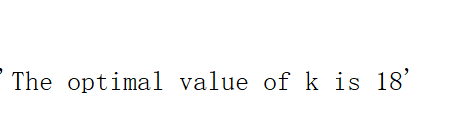


Figure 1.

Then, set the value of k from 1 to 30 to get the classification accuracy rate.

As the picture shown above, the optimal classification value of K is around 18. Figure 2.

**Run 2.**

In this classifier, we selected all patches from the training data’s image and then make these patches into kmeans clustering. So, we can get a new feature vector, and the dimension of this feature vector is k dimension, which represents the classification statistics of all patches. Finally, we use the SVM to classify new features.

Class label

Educlidean distance

Get every central point

Cluster these patches

Get all patches from training

Get features

15 linear classifiers

1. **Feature extraction**

Every 4 pixel of x and y we select a 8\*8 patch, and 32 \* 32 images can be transform to a 7\*7 patches. Repeat the process in the training data to get all the patches of pictures, and then use the method ‘kmeans’ to cluster these patches to 500 clusters.

**Name:GetPatchesOfOneImage**

Paremeters:img,xSam,ySam

Img: the image to be select to be patches.

xSam: the sampling size of x axis. If this value is increased, the size of the patches will be more.

ySam: the sampling size of y axis. If this value is increased, the size of the patches will be more.

**Name: GetFeaturesOfOneImage**

**Parameter**:img,C

**Img**: the resize image

**C**: matrix which consists of the features of each central point. Use the vl\_kmeans to cluster the all the patches.

1. **Description**:

First of all, extract the patches of the image. Then, use Euclidean distance . to calculate the distance between every patch and every central point. Each class finds all the point which have a minimum distance.

1. **Classifier:**

Use linear classifier to classify the pictures. We created 15 linear classifiers for each class. When an image needs to be classified through these linear function, it will be classified by these linear classifier until one has been classified.

1. **Validation:**

Take a set of training data and test data at a ratio of 9:1 from the training set, and calculate the average of accuracy rate which is %24.34553.

**3 run**

1. **SIFT Featrue**

use the toolbox VLFeat to extract the features of images. Vl\_dsift. All the feature points detected are expressed in terms of F，Each feature point is assigned coordinates X,Y, scale s, and direction th. Each column in F represents a feature point, and every descriptoris 128 dimension, owing to that the feature points from the image sperate to d\*d sub district on own center, and every sub district is a square. The gradient direction of the pixel is 8, and each π/2 get a direction. Finally, there will be 128 dimension of every SIFT feature.

1. **Naïve bayes classifier**

Calculate the possibility of every 15 class.

B represents the features, A represents the class. Assume that the features are independent of each other. P(A|B) = P(A1|B)\*P(A2|B)\*…\*P(An|B), because we have only 15 classes, so P(B) = 0.0667,P(A) = P(A1)\*P(A2)\*P(A3)\*…\*P(An). Calculate the possibility of every class, and make the highest one be to predicted label.

1. **Validation**

Take a set of training data and test data at a ratio of 9:1 from the training set, and calculate the average of accuracy rate which is %10.24434

**Conclude**

The purpose of this course work is to implement the process of image classification. By completing the entire course work, improve the practical ability and understand the concept of image classification better.

**Reference:**

Vlfeat.org. (2017). VLFeat - Tutorials > Dense SIFT. [online] Available at: http://www.vlfeat.org/overview/dsift.html [Accessed 15 Dec. 2017].