# HW5 Scene Classification

## Introduction

In this report, we first introduce each part of implementation procedure that classify the scene dataset. Second we show our result accuracy in each method. Finally, we have some discussion and conclusion of this report.
The Scene dataset has 15 class of scene and each class has about 100 images in training set and about 10 images in testing set, and the size of each image are variant and about 200x200 pixels.

## Implement procedure

- Tiny images representation + nearest neighbor classifier :
  First, simply resize each image to a small, fixed resolution (we use 16x16). Now, each image can be seen as a vector of 256 dimensions, and then we normalized each vector (image) into a unit length, zero mean vector. Finally, use the Euclidean distance as the distance metric, and then apply k-nearest neighbors algorithm to classify the test images.

- Bag of SIFT representation + nearest neighbor classifier :
  First, we have to build the vocabulary, so we use SIFT API supported by vlfeat Library to extract the features for each training image. But most features in an single image are redundant, these features are almost the same. So in each image, we only choose 1% of features as our feature finding result.

  Then, applying K-means clustering algorithm to these chosen features, we can get K cluster centers, and these centers are also called as visual words. Now we are ready to represent our training and testing images as histograms of visual words. For each image we will again extract its SIFT features, and for each feature we can find out which center it belongs by nearest neighbors.

  As a result, we can get the histogram for each image, and the bin of the histogram counts how many times a SIFT features was assigned to that cluster centers.

  Finally, take the pairwise distance between each test image bag of words histogram and all of the training images' bag of words histograms. Again, use k-nearest neighbors algorithm can classify the test images.

- Bag of SIFT representation + linear SVM classifier :
  We use the same procedure to get the images' bag of words histograms. But this time, we use 20% of features as our feature finding result.

Then we use libsvm with linear kernel training with the bag of words histograms and the labels of images.

- Deep learning :
  Since the dataset is too small, it is hard to apply deep neural networks to classify without augmentation. Therefore we use random resize crop and normalize to the images to prevent overfitting and enhance the result of the classifier. We use two kind of well known model, one is VGG, another is ResNet. In VGG, since the image is too small, we remove the final pool and some layers to prevent hard to converge and the final fully connected classifier network we reduce the hidden units from 4096 to 1024.
  In ResNet, we only use 4 residual block, each block of units are 64,128,256,512, to the feature extraction network, and two fully connected layer to the classifier network with 1024 units.

## Experiment Result

- Tiny images representation + nearest neighbor classifier :
  k-nearest neighbors k = 1,
  acc: **22.0%**

- Bag of SIFT representation + nearest neighbor classifier :
  vocabulary size = 100,
  k-nearest neighbors k = 7
  acc: **57.3%**

- Bag of SIFT representation + linear SVM classifier :
  vocabulary size = 100,
  k-nearset neighbors k = 7
  acc: **68.0%**

- VGG:
  all image resize to: 128x128
  epochs: 200
  acc: **66%**

- ResNet34:
  all image resize to: 224x224
  epochs: 20
  acc: **94%**

## Discussion

In task1, it is resonable that the accuracy is quite low. Because an image's feature can't be well described by its low resolution image, no matter normalized all images or not.

At the beginning of doing "Bag of SIFT representation", instead of using vlfeat Library, I use only opencv API. The results was not very good. In task2 and task3, I can only reach about 45% of accuracy. But after using SIFT and k-means API supported by vlfeat Library, accuracy improves dramatically, even up to 20% increasement.

I also found that there is a high repeatability about features in a single image. When I randomly choose a little part of features(Ex: 1%,20%...) found by SIFT as our features for k-means' input, the accuracy is almost the same compared with taken all part of features(100%). It probably means that most of the features in an image is almost the same, or has no big difference. So in the end, I respectively choose 1% and 20% as the proportion of features in task 2 and task 3. By using less features from images, k-means has less burden in doing clustering.

## Conclusion

In this homework, we use 4 different methods to do the scene classification. The first task is to represent the image by resizing it, and then use k-nearest neighbors classifier to classify the images. The second and the third task are both represent the images using bag of words, and the details about bag of words are introduced in the 'implement procedure' section. The difference between task 2 and 3 are using different classifiers, task 2 uses k-nearest neighbors classifier, and task 3 uses linear SVM classifier. In addition, we also implement two neural network models to do the classification. The experiment procedures and results are clearly introduced in the above section, and one of our deep learning model got the best results and outperforms the other three methods.

## Work assignment plan between team members

- Main Code吳承翰
- Deep learning 謝秉瑾
- Report & task 1 謝宗祐