Lok Yin LEE
20450512
lyleeaf@connect.ust.hk

Assignment 3
Spatial Pyramid Matching for Scene Classification

COMP5421
2021-04-22

1 Representing the World with Visual Words

In this section, we will extract the feature using visual words from Gaussian filters responses.

1.1 Extracting Filter Responses

Q1.1.1
Gaussian filters are used for reducing the noises and blurring the image.
Laplacian of Gaussian filters can detect all the edges in the image.
Derivative of Gaussian filters can detect the vertical edges and horizontal edges detection.
To extract features with different scales, we need to used different filters to concentrate the corresponding features, which make the system can pick up all kind of features from those filters and can represent the feature better.

Q1.1.2
Using the the function *gaussian_filter* and *gaussian_laplace*, we obtain the filter responses for all 20 filters, which is shown in Figure 1.

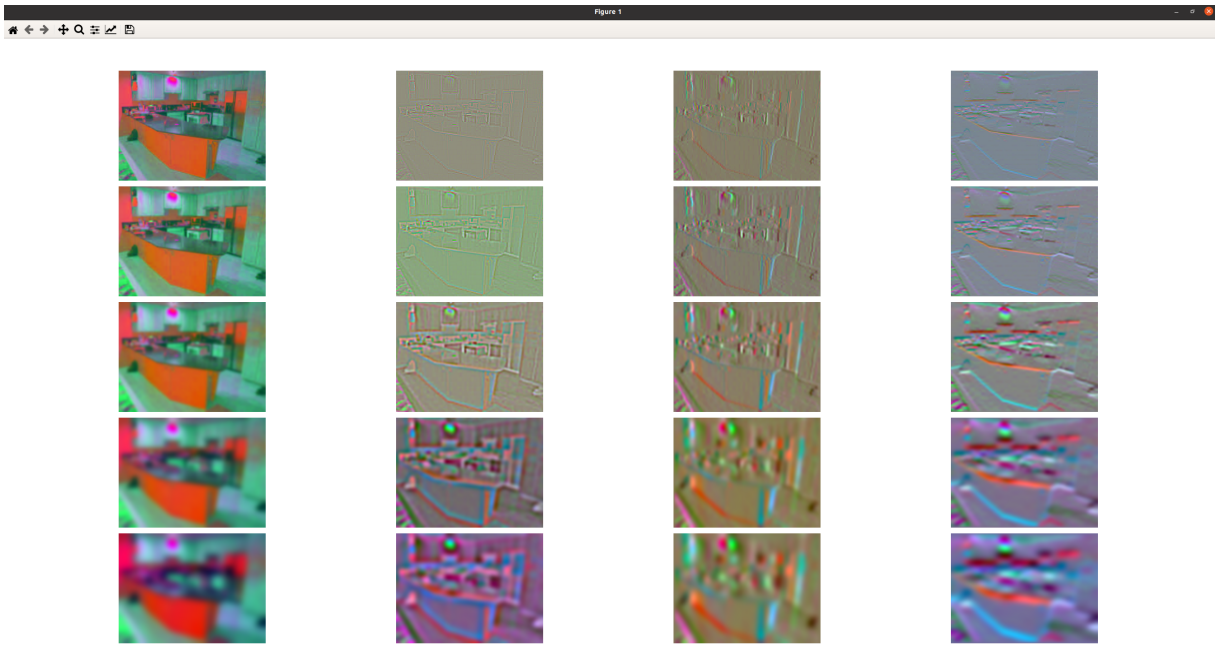

Figure 1: Samples of 20 Filter responses.

1.2 Creating Visual Words

In this section, we save all filter responses to a folder *tmp* and use them for the K means clustering. It will save the cluster formation to *dictionary.npy*.The program utilized the multiprocessing library to accelerate the execution.

1.3 Computing Visual Words

We find the distance between dictionary and the filter responses, then result the wordmap. All wordmap from images is saved in the wordmap folder. Three wordmaps are shown in figure 2.

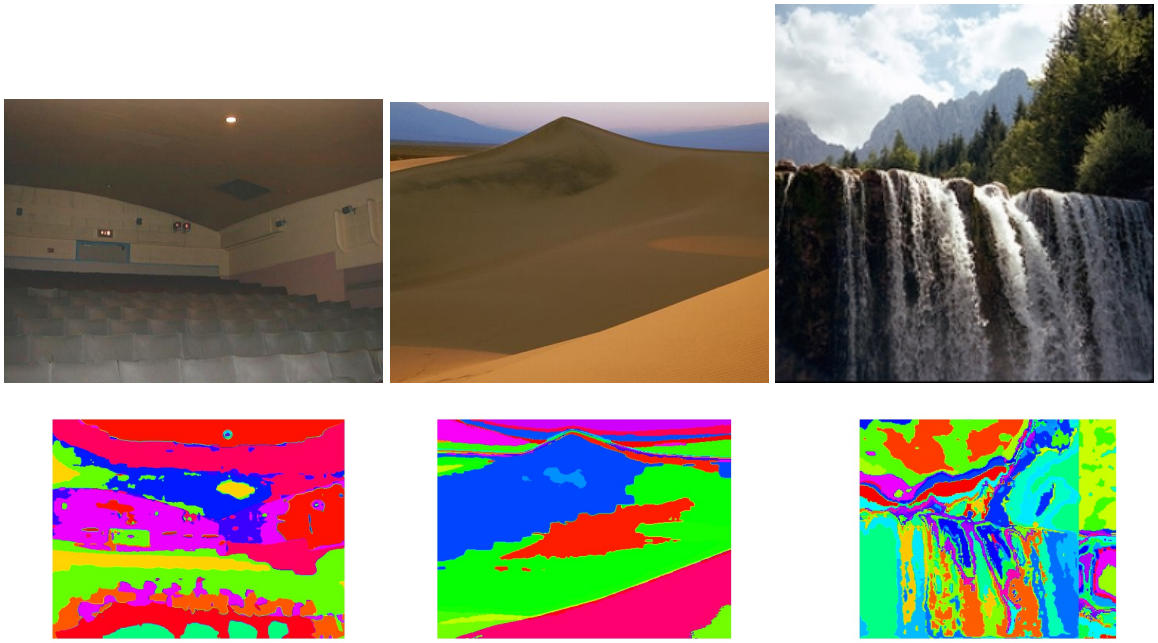

Figure 2: Three wordmaps samples.

The visual words can successfully separate the scene to many different parts. The wordmaps can roughly show the characteristic of the scene by the combination of the scene, for example that the desert mostly contain few category with large cover area and then waterfall contain many small region. This property makes the following data clustering possible.

2 Building a Recognition System

In this section, we create the program that calculate the data distance to match the predict new data.

2.1 Extracting Features

Q2.1 The function `get_feature_from_wordmap` is implemented to get the histogram from the wordmap.

2.2 Multi-resolution: Spatial Pyramid Matching

Q2.2 The function `get_feature_from_wordmap_SPM` is implemented to generate the multi-resolution representation of the given image. All the remain program will use $L = 2$, and the output will be normalized.

2.3 Comparing images

Q2.3 The function `distance_to_set` is implemented to calculate the distance between word histogram and the training samples.

2.4 Building A Model of the Visual World

Q2.4 The function `build_recognition_system` and `get_image_feature` are implemented to pack the word dictionary and training features with label. All the data will be packed to the file `trained_system.npz`.

2.5 Quantitative Evaluation

Q2.5 The function `evaluate_recognition_system` is implemented to perform prediction and evaluation. For each test image, all multi-level resolution representation will be used to calculate out the least distance difference as the predicted label. We obtain the accuracy with 55.625% and the confusion matrix as the following.

```
                Prediction Label
[[11.  0.  0.  0.  3.  3.  3.  0.]
 [ 1. 10.  1.  1.  0.  0.  0.  7.]
 [ 2.  0.  9.  1.  0.  3.  0.  5.]
 [ 0.  1.  0. 12.  0.  0.  1.  6.]
 [ 5.  1.  0.  0. 10.  1.  2.  1.]
 [ 4.  0.  0.  0.  5.  9.  2.  0.]
 [ 1.  0.  0.  2.  0.  3. 14.  0.]
 [ 0.  0.  1.  5.  0.  0.  0. 14.]]
```

True Label

2.6 Find the failed cases

Q2.6 As in the confusion matrix, we see many sample of baseball field was mis-classified as windmill, this may because the feature distribution is similar. Most baseball have tall building, open sky, large field and some people. As the same reason, many image of desert is mis-classified as windmill, since the large open area and sky is the main part of those images. The similar feature combination create similar data point in the data cluster, this make the test image have wrong prediction. Three example is shown in figure 3.

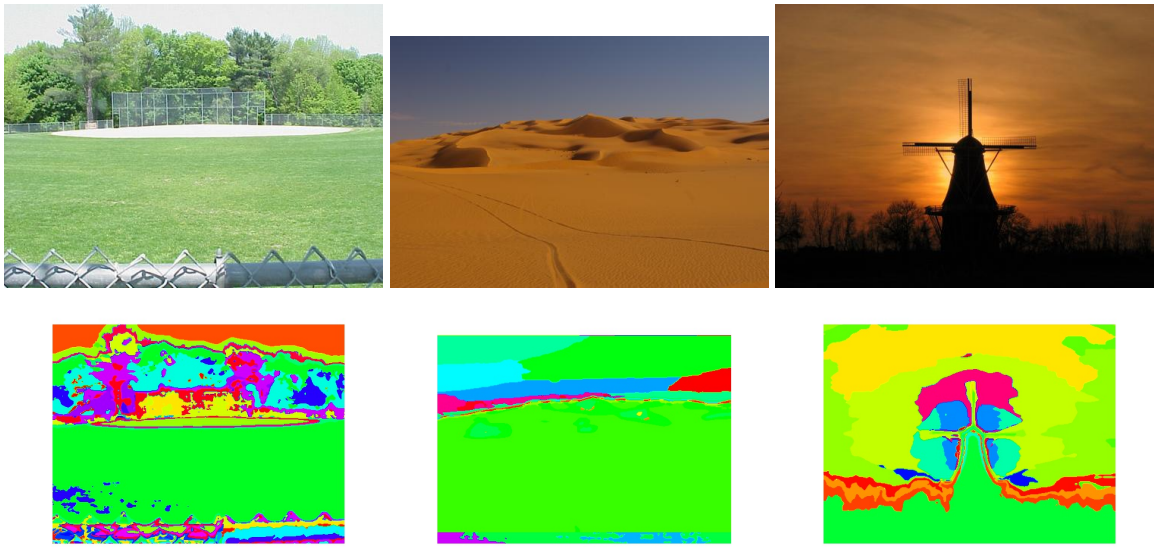

Figure 3: Mis-classification samples.

3 Deep Learning Features - An Alternative to "Bag of Words"

In this section, we will implement the feature extraction part using pre-trained VGG16 CNN.

3.1 Extracting Deep Features

Q3.1 We implemented the basic functions such as `multichannel_conv2d`, `relu`, `max_pool2d` and `linear`.

3.2 Building a Visual Recognition System: Revisited

Q3.2 Finally, we combine those function as the model that able to inference images to lower dimension features as the function `build_recognition_system`. We also create the evaluation function that calculate

the negative Euclidean distance to predict images. It reach accuracy at 93.125% and the confusion matrix is shown in the following.

```
                    Prediction Label
              [[20.  0.  0.  0.  0.  0.  0.  0.]
               [ 1. 17.  1.  0.  0.  0.  1.  0.]
               [ 0.  0. 18.  2.  0.  0.  0.  0.]
True Label     [ 0.  0.  0. 20.  0.  0.  0.  0.]
               [ 0.  0.  0.  0. 19.  1.  0.  0.]
               [ 0.  0.  0.  0.  2. 18.  0.  0.]
               [ 0.  0.  1.  0.  0.  0. 19.  0.]
               [ 0.  0.  1.  1.  0.  0.  0. 18.]]
```

For the result, the CNN feature extraction is better than the classical BoW a lot. This result should be caused by the pre-training VGG network. Since the VGG16 is trained on the ImageNet which is a comprehensive dataset. Since the VGG16 is trained for classification task, the extracted feature is made for distinct the scene. In other way, the VGG network consist more layer and more weights, it can get more important information.