Bachelor Project

Multi-Methods Image Retrieval

Minchen ZHU

Department of Computer and Information Science
Universität Konstanz

Supervisor:
Prof.Dr. Bastian Goldlücke
M.Sc. Antonín Šulc

May 2017

Contents

Chapter 1

Introduction

Image retrieval has been an interesting and important application in computer vision. The features extracted from a digital image have been also developed a lot, from the color-based features such as color moment, texture-based features like contours in image, to content-based features, for example Bags-of-Features and convolutional neural network features.

In this project, 5 methods for extracting features from digital images were implemented: average color values of RGB channels, color moments, LBP (local binary pattern), Bags of Features (SIFT + K-Means) and CNN (convolutional neural network). Their performances were compared, how accurate and how well they could retrieve the images.

In addition, the qualities of the features extracted by those 5 methods were also checked by applying SVM classifier on them. A good method should extract the features from images that are sufficient for the SVM to perform classification on features.

In the experiments, the CNN turned out to be the best method to extract features. It had 100% accuracy in retrieval when the input image is already in the database. When the input image is not in the database, CNN had a chance of 62.18% to return a similar image in the database that is in the same category as the input image. In the test with SVM, the cross validation accuracies of CNN features were also far better than those of the features extracted by the other 4 methods.

This result was summarized based on straight forward implementations of these methods. If some of the methods were very well optimized, they might have an opportunity to get comparable results.

Chapter 2

Methods

2.1 Naive method: average color value of 3 channels

As we know, a (digital) color image is a digital image that includes color information for each pixel. The RGB color space is commonly used in computer displays. A color digital image in RGB color space has three values per pixel and they indicate the intensities of each channel.

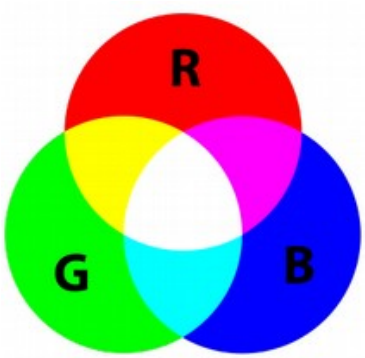

Figure 2.1: The pixel value in RGB color space is the sum of the value in 3 channels

This naive method is to calculate the average value of the 3 channels of one image. Then a 3-element-vector is the feature vector that represents the characteristic of the image:

$$AvgC = \begin{bmatrix} \sum_{j=1}^{N} P_(j,R) * \frac{1}{N} \\ \sum_{j=1}^{N} P_(j,G) * \frac{1}{N} \\ \sum_{j=1}^{N} P_(j,B) * \frac{1}{N} \end{bmatrix} \tag{2.1}$$

where

N: the total number of pixels in the image;

P: the pixel value of certain color channel.

This is the simplest method that extracts key information from a digital image with an obvious disadvantage: the color of the background dominates the feature vector so that several images of different objects might have very similar feature vectors, which significantly reduces the accuracy of image retrieval results. We will see the examples and evaluation in Chapter 4.3.

2.2 Color Moments

Another color based feature that can be extracted from digital images is color moment. Color moments can be computed in any color space and they encode both shape and color information, so they are a good feature to use under changing lighting conditions. The main advantages of color moments features are scaling invariance and rotation invariance.

Most of the color distribution information is contained in the low-order moments:
First moment is the mean values of 3 channels, just the same as the method mentioned above. It indicates the degree of brightness of an image.

$$CM1_i = \frac{1}{N} * \sum_{j=1}^{N} P_(j,i) \tag{2.2}$$

where
N: the total number of pixels in the image;
P: the pixel value of a color channel;
i: a certain color channel.

Second order color moment is the standard deviation. It reflects the color distribution range in an image.

$$CM2_i = \sqrt{\frac{1}{N} * \sum_{j=1}^{N} (P_(j,i) - CM1_i)^2} \tag{2.3}$$

Third order color moment is called skewness. It measures how asymmetric the color distribution is, and thus it gives information about the shape of the color distribution. Skewness can be computed with the following formula:

$$CM3_i = \sqrt[3]{\frac{1}{N} * \sum_{j=1}^{N} (P_(j,i) - CM1_i)^3} \tag{2.4}$$

2.3 LBP (Local Binary Patterns)

LBP (local binary patterns) is an operator that describes the local texture information of an image. It has significant advantages such as rotational invariance and gray scale invariance. The original LBP operator is defined in a 3*3 window and the pixel value of the central pixel in the window is considered as the threshold for the whole window. The values of the 8 neighboring pixels are compared with this threshold. If a neighbor's pixel value is larger than the value of the central pixel, this neighbor's position is marked as "1", else is "0". So in a 3*3 window, the central pixel is described by an 8 bit binary number. This number represents the texture information of this small area.

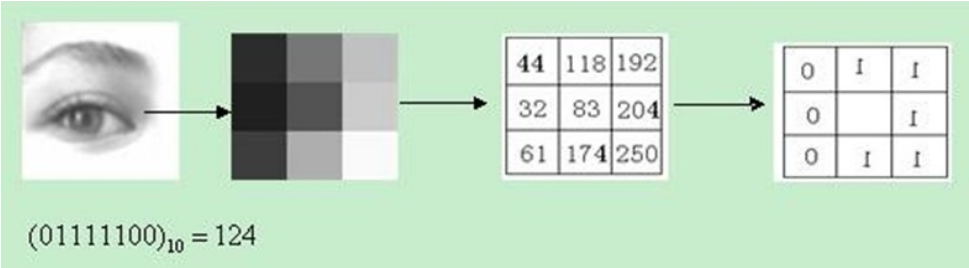

Figure 2.2: Illustration of simple LBP operator[3].

In order to bring the rotational invariance, the circular neighborhood is used instead of a square neighborhood. By rotating the circular neighborhood, different LBP values can be obtained according to its original definition. The minimum is finally the LBP value of this neighborhood, which ensures LBP's advantage of rotational invariance.

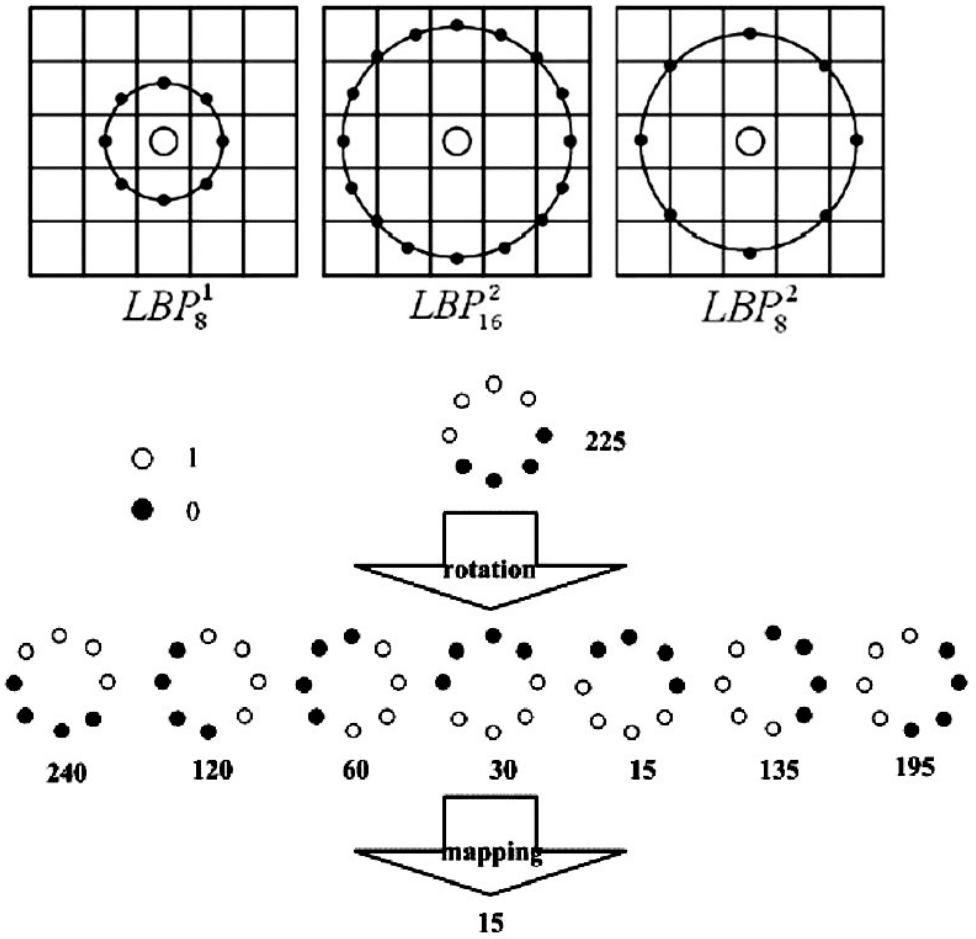

Figure 2.3: The circular neighborhoods of different sizes can adapt to different scale texture features. The lower part shows how the rotational invariance is guaranteed in a LBP_8^2 neighborhood

2.4 BOF (Bags of Features)

The Bags of Features method consists of 2 steps. In the first step, the SIFT features are extracted from the images. The image is scaled from original size to smaller sizes and convolved with Gaussian filter of different σ to build up the image pyramid:

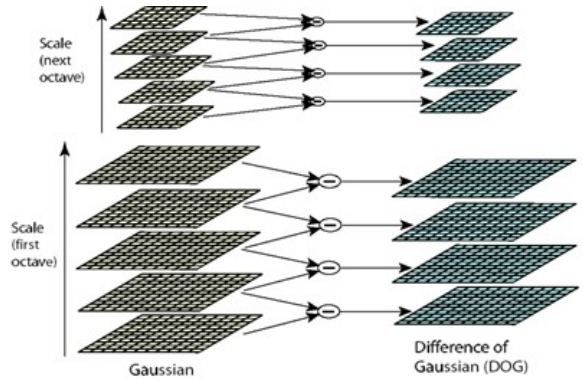

Figure 2.4: Image pyramid constructed for SIFT.

The DoG is the difference between the convolved images of the same scale but with different σ. In order to find the extreme point of the scale space, each sample point should be compared with all its adjacent points to see if it is larger or smaller than the adjacent points of its image domain and scale domain. In the Figure 5, the test point and its 8 adjacent of the same scale and the 9 * 2 points of the adjacent scales (total of 26 points) are compared to ensure that both in the scale space and two-dimensional image space, the extreme point could be detected. If a point is the maximum or minimum among the 26 neighborhoods of the DOG scale space and the upper and lower layers, it is considered to be a feature point of the image at that scale.

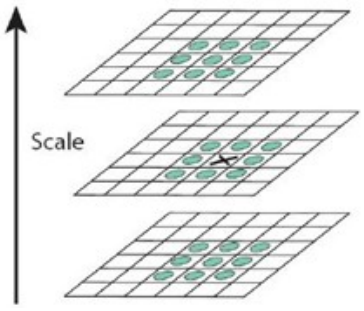

Figure 2.5: Illustration of finding the extreme point.

After some processes, the feature points of lower quality are discarded. Then the magnitude and direction of the remaining feature points are calculated. Along with the spatial location of the feature points, the extraction phase of SIFT features from the image is finished.

The pixels in the neighboring windows around the feature points are sampled and the histogram is used to count the gradient of the neighborhood pixels. The peak of the histogram represents the main direction of the neighborhood gradient at that feature point, which is also marked as the direction of that feature point.

Then the 8 * 8 window is taken, with the feature point as its center.

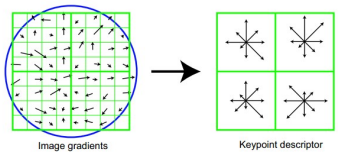

Figure 2.6: Calculate descriptor.

The center of the left part of Figure 6 is the position of the current feature point. Each small cell represents the pixel of the scale space in the neighborhood of it. The gradients and direction of gradient of each pixel are calculated. The direction of the arrow represents the direction of the pixel's gradient, the length of the arrow represents the gradients' magnitude, and then they are weighted with the Gaussian window.

After all feature points are extracted and their descriptors are calculated, a classification process for all the SIFT features is performed via clustering. Simply to say, for each image, a histogram of their SIFT features is generated. This histogram is the feature vector of Bags of Feature.

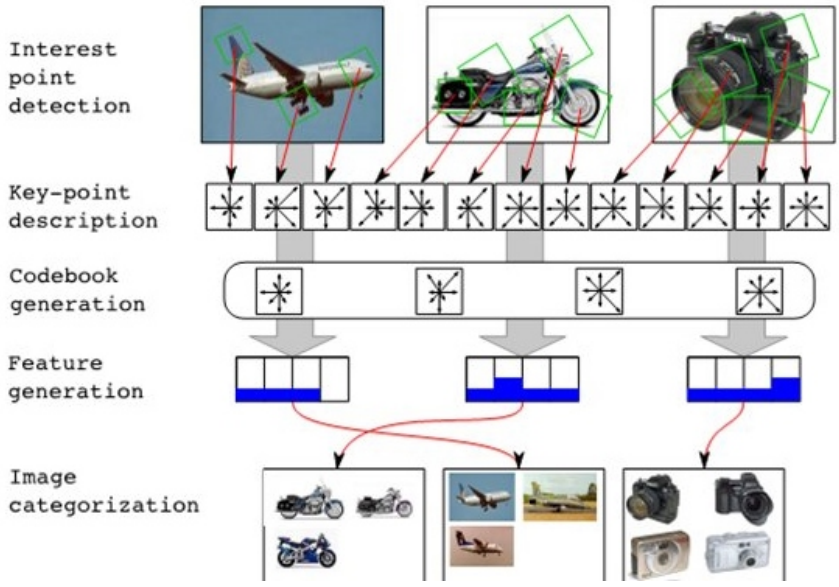

Figure 2.7: Brief process of Bags of Feature.[4]

2.5 CNN (Convolutional Neural Network)

A convolutional neural network is a sequence of connected layers where the layers are usually convolutional layers, rectified linear units layers (ReLU), pooling layers and loss layers.

Convolutional layers apply some convolution operations on the images to extract characteristics like edges, contours, corners, etc. More convolutional layers of a network can extract more complex features from lower-level features. All convolution functions are well designed to extract specific characteristics.

ReLU enhances the decision function and the nonlinear characteristics of the entire neural network, and itself won't change the convolution layers.

Pooling layers divide the image into several rectangular regions, and certain value, for example the maximum or minimum is returned as an output for each sub-region. The pooling operation constantly reduces the size of the data, so the number of parameters and the calculation effort will decline, which to some degree also control the over-fitting. In general, the pooling layers are periodically inserted into the convolution layers of a CNN.

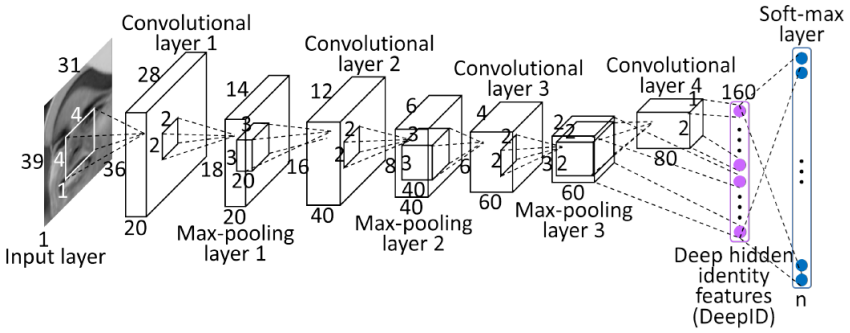

Figure 2.8: Example of CNN: DeepID network.[5]

An intuitive explanation of CNN is, the convolution could reduce the complexity without loss of information of the original images. The results of the convolution in the previous layers, after they are processed by other in-between layers, become the input of the next convolutional layers. The characteristics extracted from later convolutional layers are more specific.

2.6 SVM as classifier

The support vector machine (SVM) tries to find a set of hyper-planes to separate the data points, so that the data points could be classified.

The idea of using SVM is, on the one hand, if the category of the input for image retrieval could be predicted, this input image then doesn't need to be compared with all the images in the database, but only with the images that are in the same category, so the searching time could be reduced. On the other hand, applying a classifier could tell, to some degree, how good a feature is. For example, if there is not much difference between the features of several categories, then we might say this feature is not sufficient to provide enough information of the images.

Chapter 3

Implementation

3.1 Database and pre-processing

The database used in the image retrieval is Caltech-256. It has 256 categories of objects and an additional category for clutter. The total number of images in the original database is 30607 and at least 80 images per category. In practice, one category has at least 85 images.

In order to use SVM to classify the image features, the images were sorted out and reduced to 75 per category. So the training set has a total number of 19275 images. The validation set (test set 1) consists of 1285 images, each category contributed 5 images from the training set. The test set (test set 2) contains also 1285 images, in each category it enrolled 5 images, but these images are not in the training set.

3.2 Flowchart

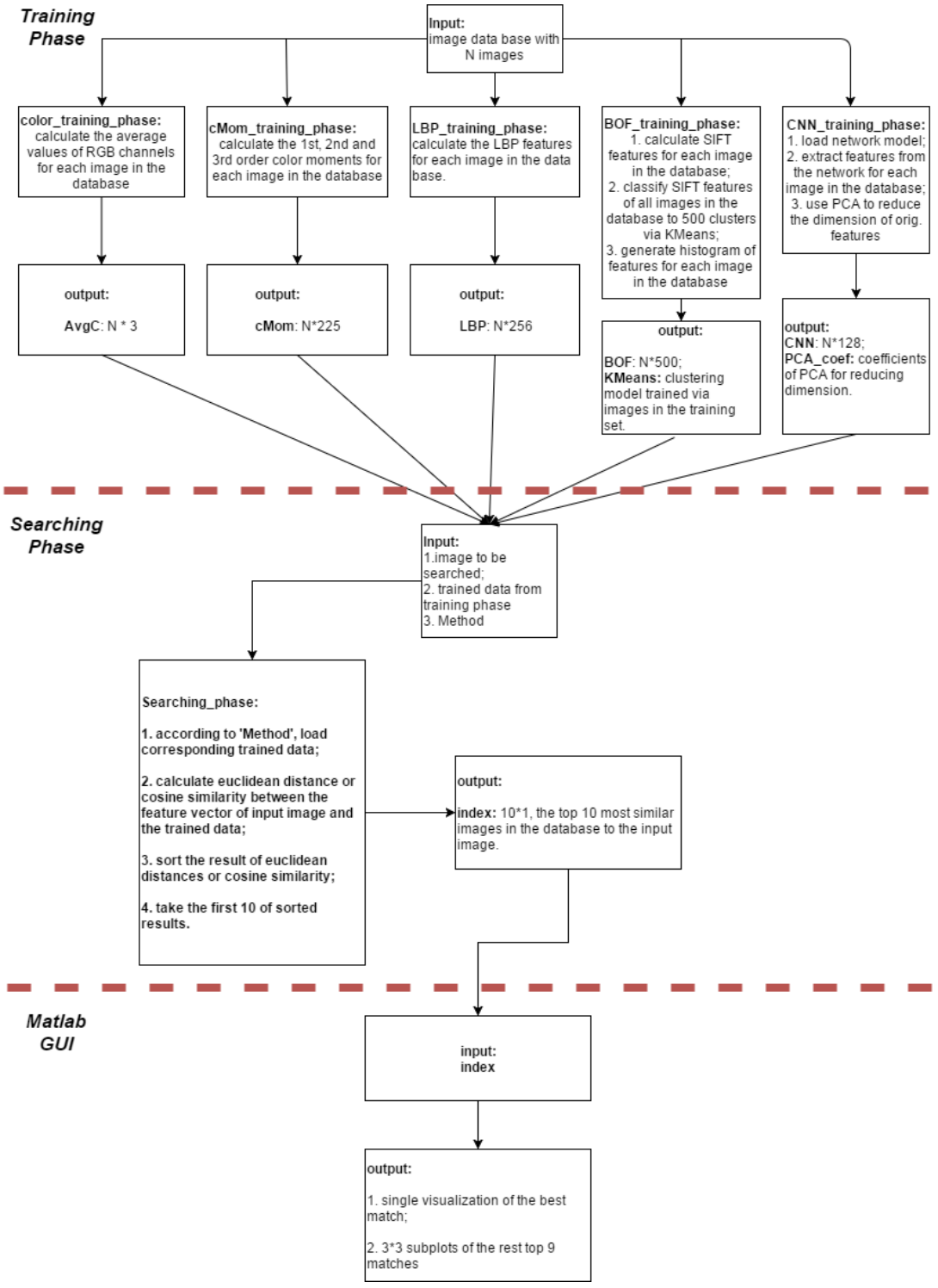

Figure 3.1: Example of CNN: DeepID network.

3.3 Avg. color, color moments, LBP and BOF[1]

These 4 methods were implemented according to the introductions in Chapter 2.

All implementations were not optimized.[1] [2]

3.4 MatConvNet

For CNN feature extraction, MatConvNet[3] toolbox on Matlab was used.

The network model used in this project is imagenet-vgg-f[7, 8]. This network has 8 layers and its structure is shown in the following Figure[4]

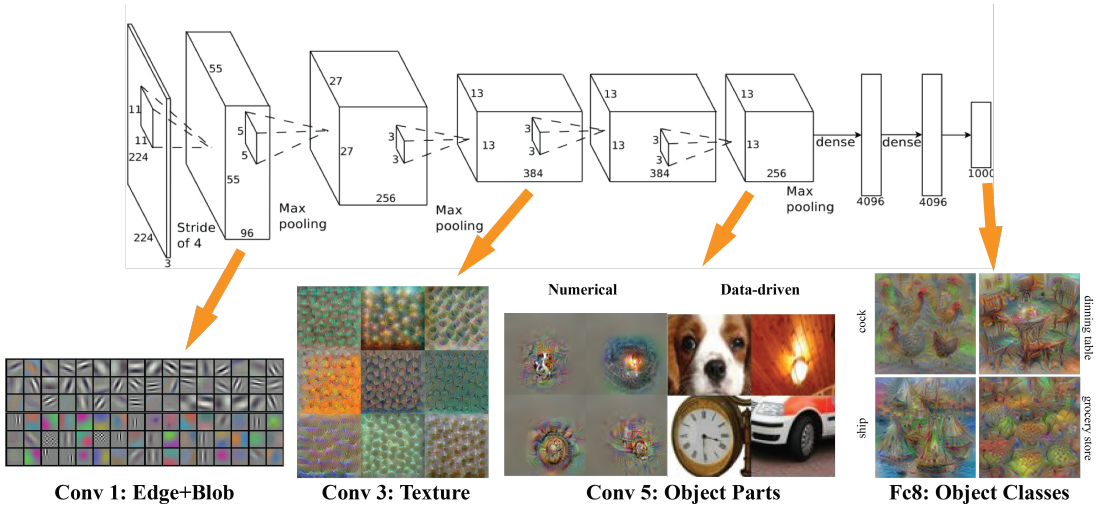

Figure 3.2: Illustration of structure and principle of imagenet-vgg-f model.

The output feature vector from this network is the 4096 dimensional vector[5] just before the classification process (softmax).

[1] The lbp.m is base on the implementation by Marko Heikkilä and Timo Ahonen. In this project, the radius of the circular neighbourhood was not used for feature extraction, so the rotation invariance is not guaranteed. Original matlab code please see <http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab>.

[2] Yantao Zheng and Noemie Phulpin, Scale Invariant Feature Transform Implementation[6]

[3] MatConvNet is a MATLAB toolbox implementing Convolutional Neural Networks (CNNs) for computer vision applications. It is simple, efficient, and can run and learn state-of-the-art CNNs. Several example CNNs are included to classify and encode images. All information about MatConvNet toolbox and the user guide, please check: <http://www.vlfeat.org/matconvnet/>

[4] VGG-F network visualized by mNeuron.

[5] Key code for CNN feature extraction in this project was a slight modification of the source code: <http://www.cc.gatech.edu/ hays/compvision/proj6/>

3.5 LIBSVM[2]

LIBSVM [6] was chosen as the SVM classifier.

3.6 Matlab GUI

An Image Retrieval Program was implemented with the Matlab GUI.

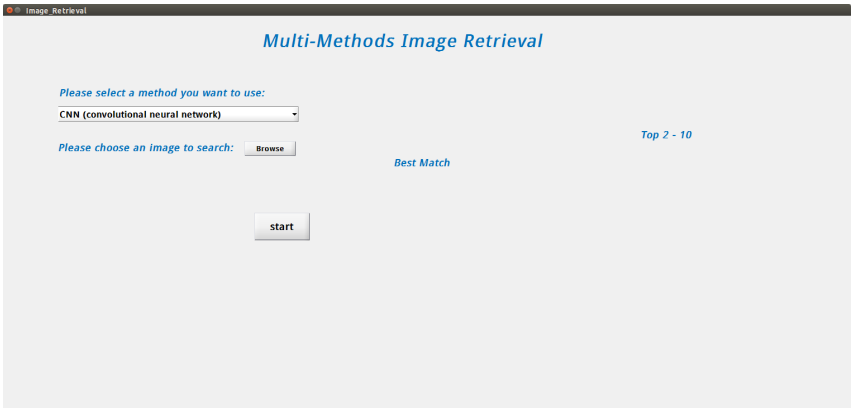

Figure 3.3: Initial Interface of GUI.

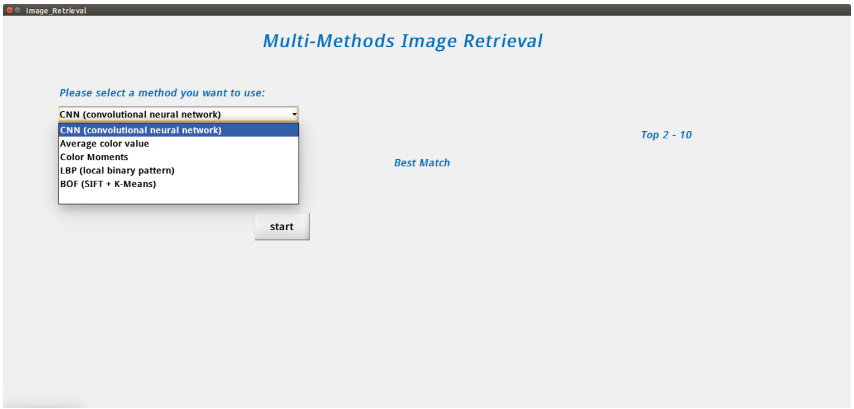

Figure 3.4: The popup menu enables selecting a method for feature extraction.

[6]LIBSVM is a popular open source machine learning libraries, both developed at the National Taiwan University and both written in C++ though with a C API. LIBSVM implements the SMO algorithm for kernelized support vector machines (SVMs), supporting classification and regression. All information about LIBSVM and the user guide, please check: <https://www.csie.ntu.edu.tw/ cjlin/libsvm/>

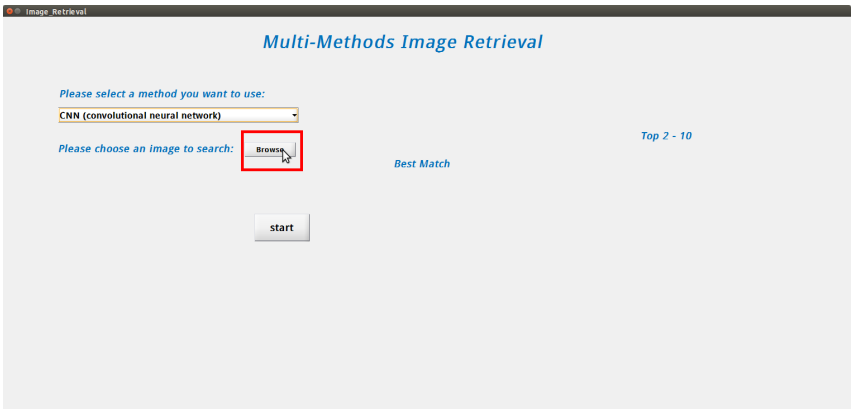

Figure 3.5: Select an image to retrieve by clicking the "Browse" button.

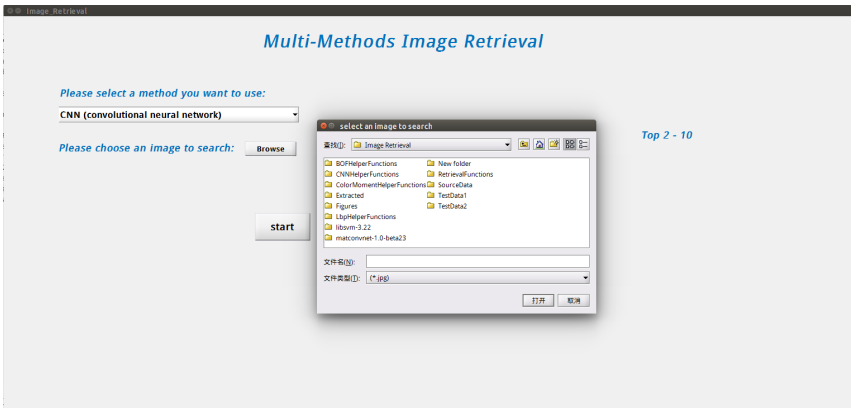

Figure 3.6: Select the input image in file system.

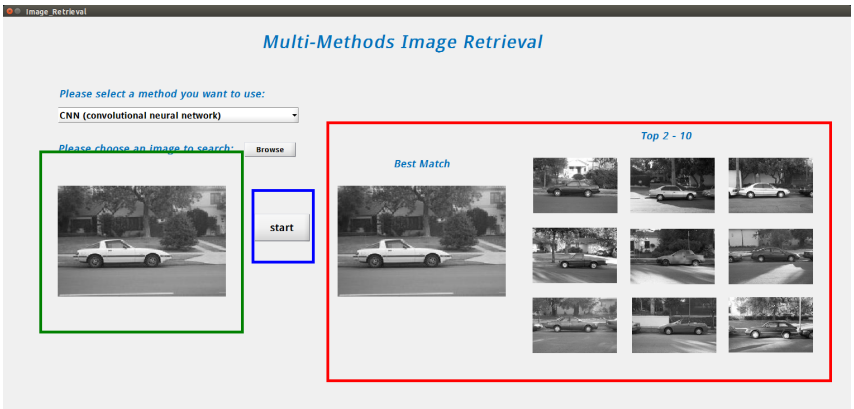

Figure 3.7: In the area surrounded by the green frame, the input image is showed; after clicking the "start" button in the blue frame, the retrieval results are shown in the area in red frame; on the left side is the best match in the database, on the right side is a 3*3 subplots of the Top2 - 9 matches.

Chapter 4

Experiments

4.1 Determine LIBSVM parameters

In LIBSVM the most commonly used kernel is RBF kernel. It has 2 parameters: cost and gamma. The grid-search is an exhaustive method to find the optimal c and γ according to the cross validation accuracy, which indicates how well the SVM can classify the features. Conversely, it would also tell whether a feature contains (good) enough information of original images.

The first part of experiments focused on the performance of SVM while it classifies a different number of categories. The range of c and γ are from 10^{-7} to 10^{7}. The validation set has 5 images per category. The first 50, 100 and 150 images were picked out and the features were extracted by the methods mentioned in Chapter 2. So for each method, 10, 20 and 30 categories were used in this experiment.

The heat maps of the cross validation accuracies are in the Appendix. Here only the maximal cross validation accuracies for each method are shown:

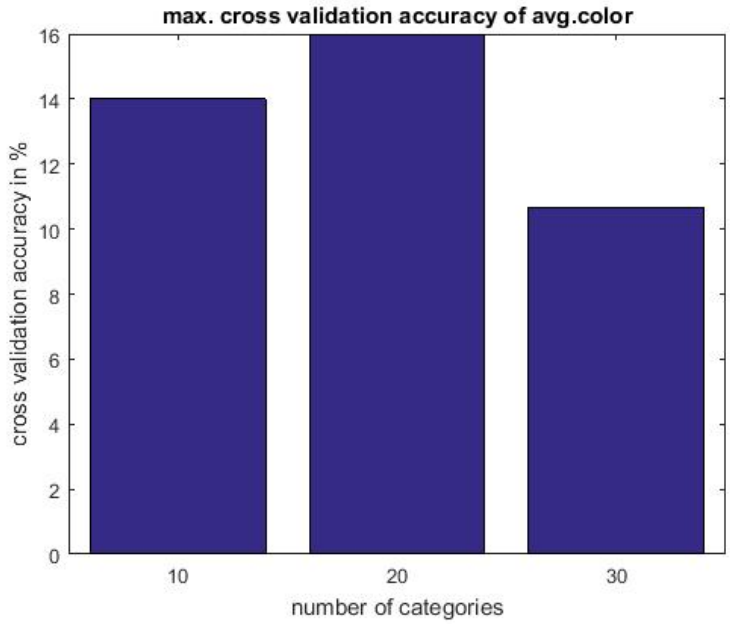

Figure 4.1: maximal SVM cross validation accuracies of avg.color features with 10, 20 and 30 categories.

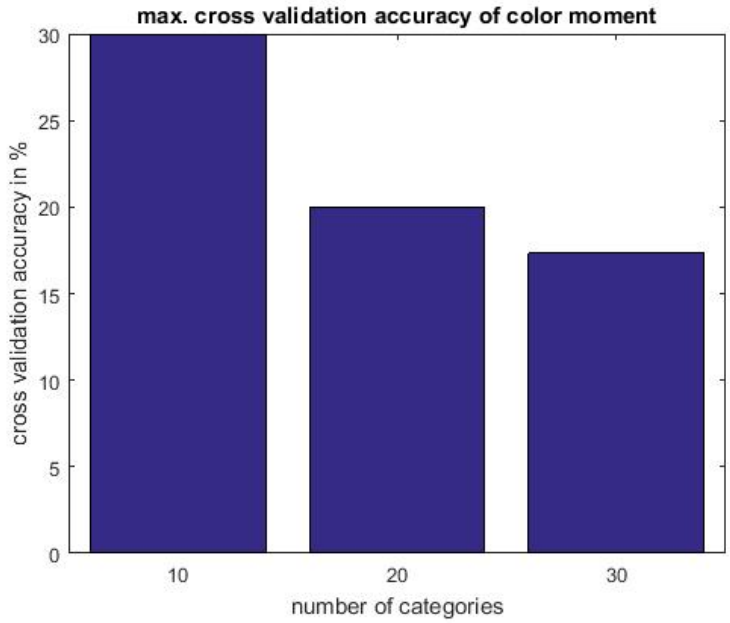

Figure 4.2: maximal SVM cross validation accuracies of color moments features with 10, 20 and 30 categories.

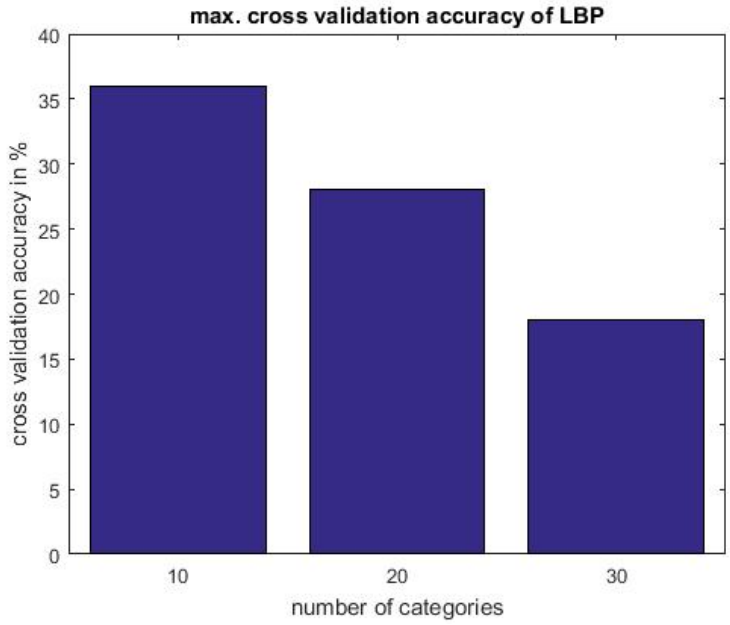

Figure 4.3: maximal SVM cross validation accuracies of LBP features with 10, 20 and 30 categories.

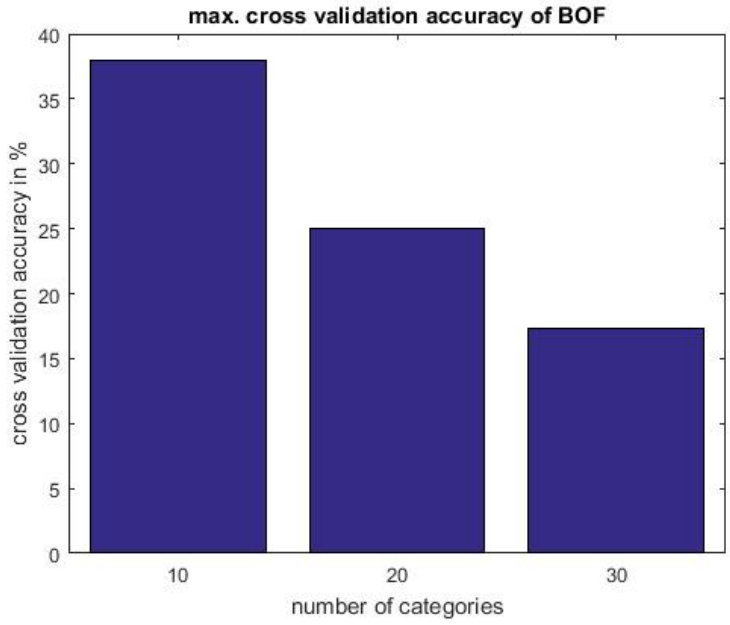

Figure 4.4: maximal SVM cross validation accuracies of BOF features with 10, 20 and 30 categories.

From Fig. 4.1 to Fig. 4.4 we can find a trend that the maximal cross validation accuracy decreases while the number of categories increases. The peak of cross validation accuracy appeared when the trained images had 10 categories in this experiment. As the real training set has 257 categories, we could expect an even worse performance of classification on the real training set by SVM.

In addition, and the maximum has never exceeded 40%. To some degree, it tells that the quality of these 4 methods might be not good enough for SVM to decide hyper-planes to separate all the categories.

Second part of this experiment focused on CNN features and the relation between the number of feature points and the performance of SVM classification.

CNN features had far better cross validation accuracy than the other 4 methods, but one feature vector has 4096 elements. In another word, the CNN features have too large dimensions. In order to reduce the number of dimensions of it, PCA (principle component analysis) was applied on the CNN feature vectors, so it could be reduced to 128 elements per feature vector.

The following two figures showed the heat map of cross validation accuracies of 128-dimensional and 4096- dimensional CNN feature vectors, both with 10 categories.

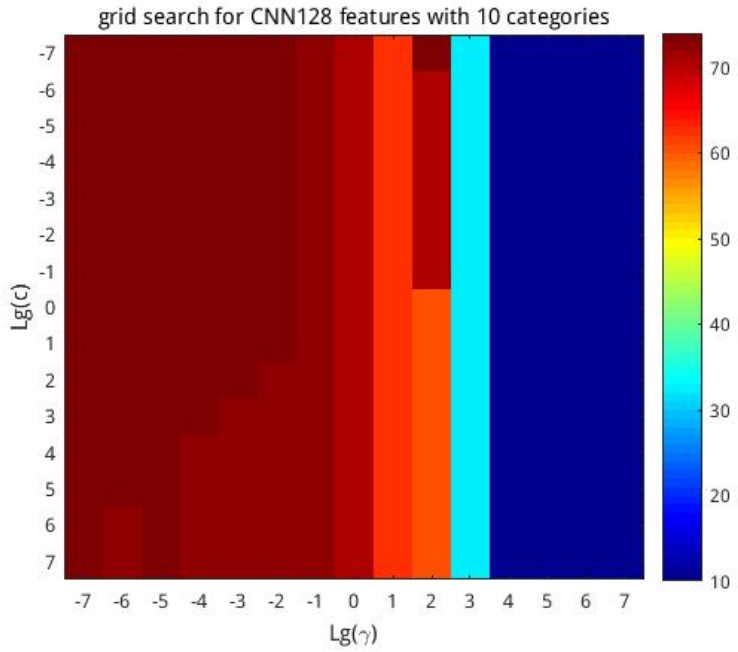

Figure 4.5: Cross validation accuracy heatmap of CNN features of 128 dimensional features

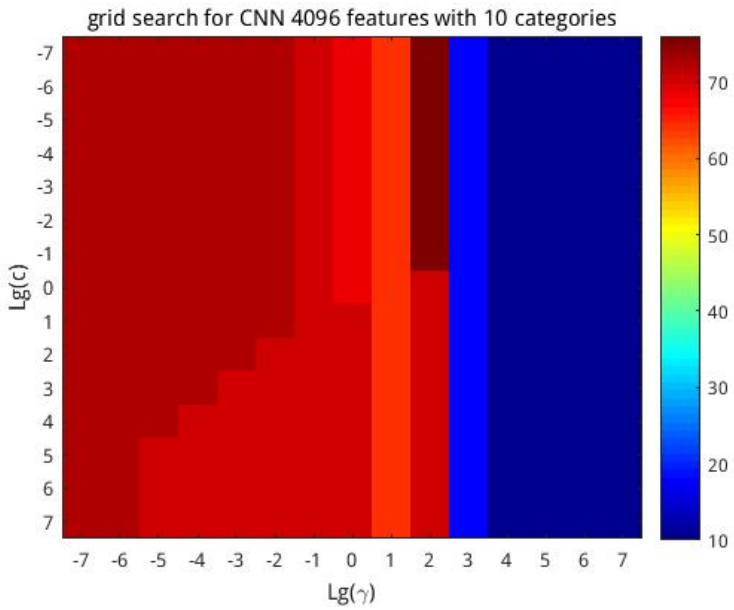

Figure 4.6: Cross validation accuracy heatmap of CNN features of 4096 dimensional features

From the color bars on the right side of both figures we can find that the maximal cross validation accuracies of both cases didn't have much difference. But in the column at $\lg(\gamma) = 3$, the cross validation accuracy of 128 dimensions was higher than that of 4096 dimensions.

This part of experiment showed that the CNN features had better quality than the features of the other 4 methods and applying PCA on CNN feature wouldn't lead to loss of information. Although the main purpose of this experiment was to find the optimal c and g parameter for SVM, but due to the low cross validation accuracies on avg. color, color moment, LBP and BOF features, it might not be a good idea to use SVM to classify the features of all the methods. And in practice, the searching time was generally acceptably short, so the use of SVM is discarded in the program.

4.2 Retrieval accuracies

This experiment had also two parts. The first part was to check the retrieval accuracy of all 5 methods, when the input image is in the training set. The test set was the validation set. The features vectors of validation set of methods avg. color, color moment, LBP and CNN were calculated in the same way as they were in the training set, so the criterion of correctly finding them was

$$EuclideanDistance = 0 \tag{4.1}$$

between the feature vectors of the test set and the training set.

For the feature vectors of BOF, the SIFT features were calculated in the same way, but for the histogram of features, the K-Means model of the training set was used, so the feature vectors of validation set was slightly different as the original ones. The criterion was

$$CosineSimilarity > 0.5 \tag{4.2}$$

Two similar vectors should at least point to the same interval of direction.

The results are summarized in the following table:

Table 4.1: Image Retrieval Accuracy when input in the training set

Method	avg.color	color moments	LBP	BOF	CNN
Accuracy	100%	100%	100%	99.69%	100%

The second part was to check the retrieval accuracy of all 5 methods, when the input image is not in the training set. The criterion is, only when the most similar image in the training set is found, which was in the same category as the input image, counted as "correct". The results are also shown in the Table:

Table 4.2: Image Retrieval Accuracy when input in the training set

Method	avg.color	color moments	LBP	BOF	CNN
Accuracy	2.80%	11.60%	6.54%	7.47%	62.18%

In this experiment we can see that, if the input is in the training set, all methods had very good retrieval result. Once the input is a new comer, only the CNN could give a retrieval accuracy of 62.18%, the other methods were really not good at it.

4.3 Visualization of retrieval's results

The experiment 4.2 only gave a very objective and statistical overview of the performance of those 5 methods. Since the Matlab GUI visualizes the retrieval results, we could have a subjective judgment on the performance of the 5 feature extraction methods.

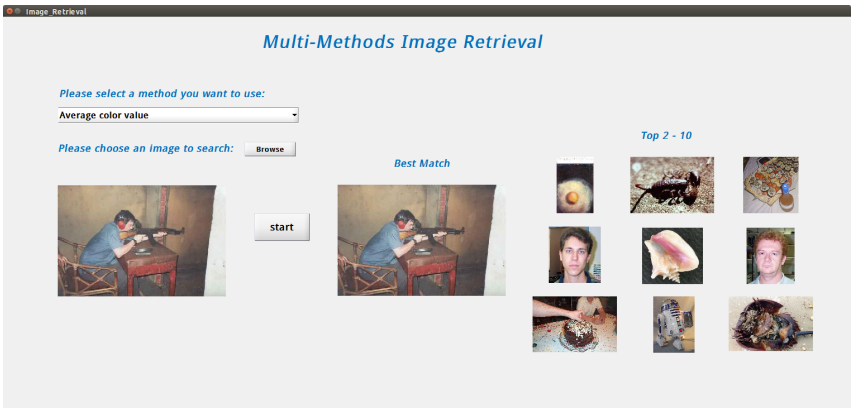

Figure 4.7: Image retrieval by using avg. color as image feature

As promised, an example of image retrieval by using avg. color as image feature is illustrated in Fig. 4.7.

The input image was in the category of "AK47". Although the image retrieval was successful, but if we take a look at the Top 2-10 matches, we can see that all these images had nothing to do with shotguns, but their color of background is very similar to the input image and the color of background dominates the avg. color feature vector, which is just the situation described in Chapter 2.1.

Figure 4.8: Image retrieval by using CNN as image feature

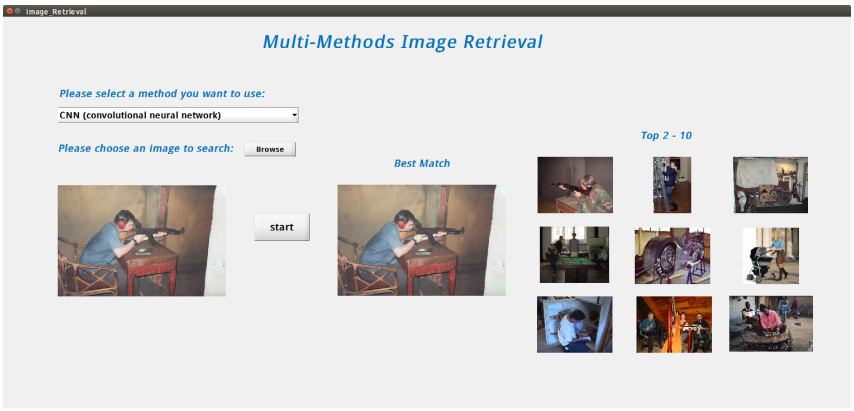

Figure 4.9: Image retrieval by using CNN as image feature

From Fig. 4.8 and Fig. 4.9 we might discover the importance of the image's subject. In Fig. 4.8, the subject is an AK47. It was located in the middle of the image, without any other substances around it. So the top 2-10 matches were very close to the input images, even some of them were not shotguns, but at least they were guns or pistols. By contrast, in Fig. 4.9, the subject was actually a man holding an AK47. The second best match was also brilliant, but the other top matches had no relation with shotguns, but mainly human beings. The CNN actually took the man in the input image as the subject of the image.

So in this experiment, the subject's position and the components of an image turned out to be very important to feature extraction. We have to choose proper images to train the model.

Chapter 5

Conclusion

After all the experiments, the CNN seems to be the best choice to extract image features among all the methods mentioned in this project. It not only has the highest cross validation accuracy, which means it supports the use of SVM, but also 100% accuracy finding an image that is in the database. Even the input image is a new comer, the CNN provides an acceptable result, that it had a chance of 62.18% to give a similar image back, which is in the same category as the input image.

The other methods have the opportunity to be improved. The color moment has no limitation on the color spaces. If the RGB color space could be converted to Lab color space, the features might be better, since the Lab color space has a larger color gamut than RGB color space, thus the values of the pixels might be more precise.

The LBP implementation by Marko Heikkilä and Timo Ahonen is actually powerful. By defining the radius of a circular neighbourhood and the mapping strategy, the LBP feature could be extracted better than that in this project.

For the BOF, the initial cluster centers were generated randomly and the number of clusters was also a default setting. If the initial cluster centers and the number of clusters could be determined by experiments, the performance of BOF features might also have an improvement.

Finally, as we see in Chapter 4.3, the subject's position and the components of an image could have great influence on the feature extraction process. If we want to have better result of image retrieval, selecting good images as the training set, in which the subject dominates the whole image domain, would give a positive contribution to the performance.

Appendix
Heatmaps of cross validation accuracies in Chapter 4.1

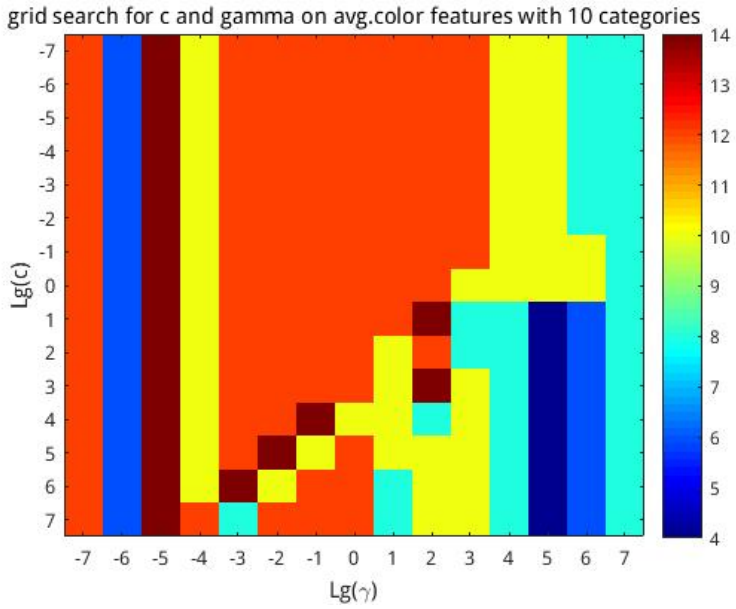

Figure 5.1: CV accuracy of avg.color feature of 10 categories

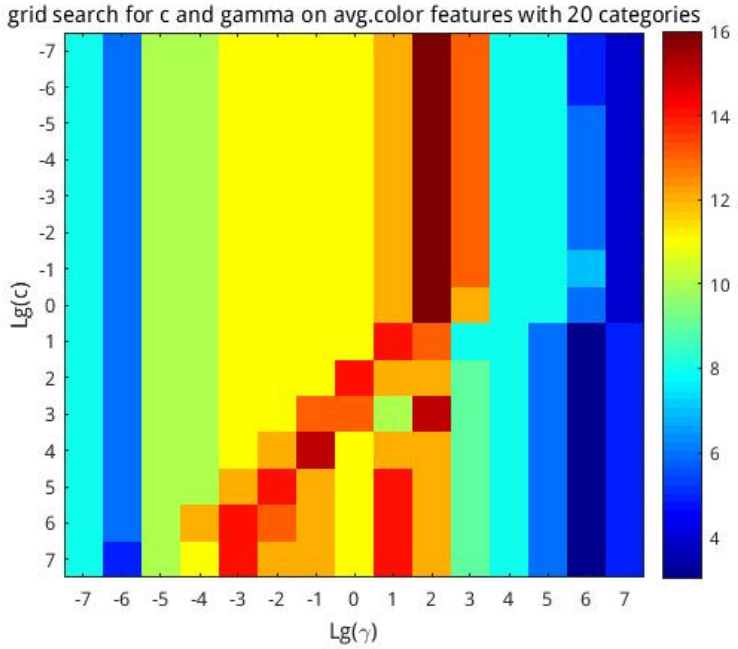

Figure 5.2: CV accuracy of avg.color feature of 20 categories

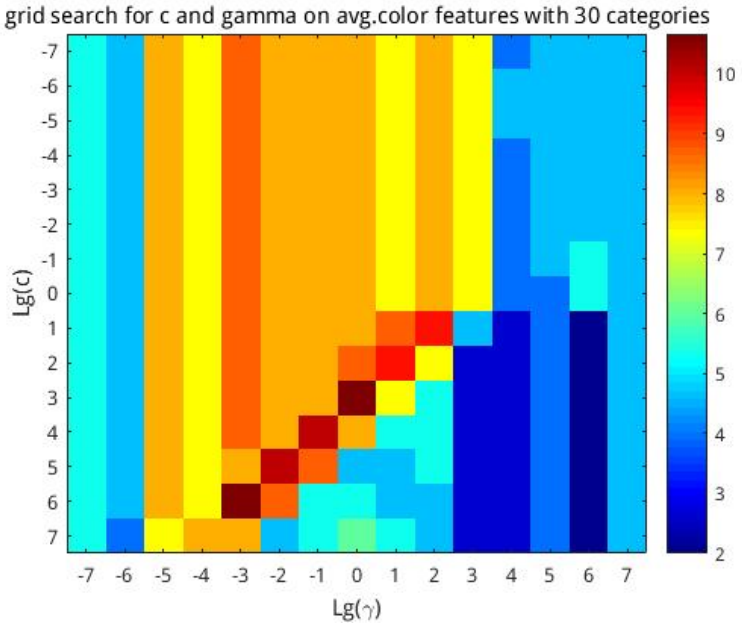

Figure 5.3: CV accuracy of avg.color feature of 30 categories

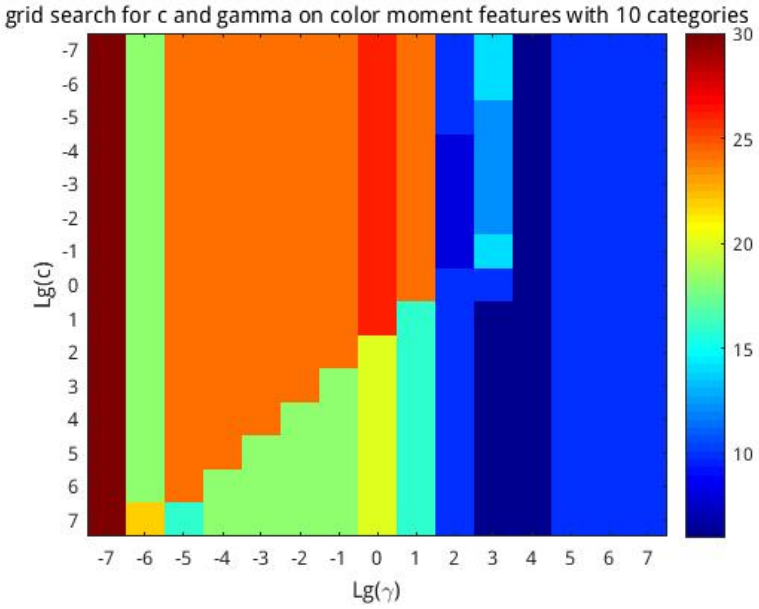

Figure 5.4: CV accuracy of color moments feature of 10 categories

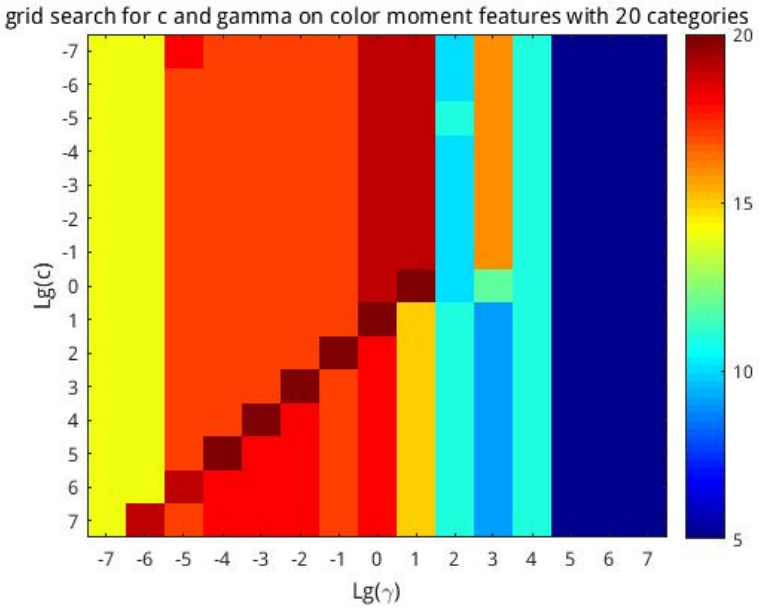

Figure 5.5: CV accuracy of color moments feature of 20 categories

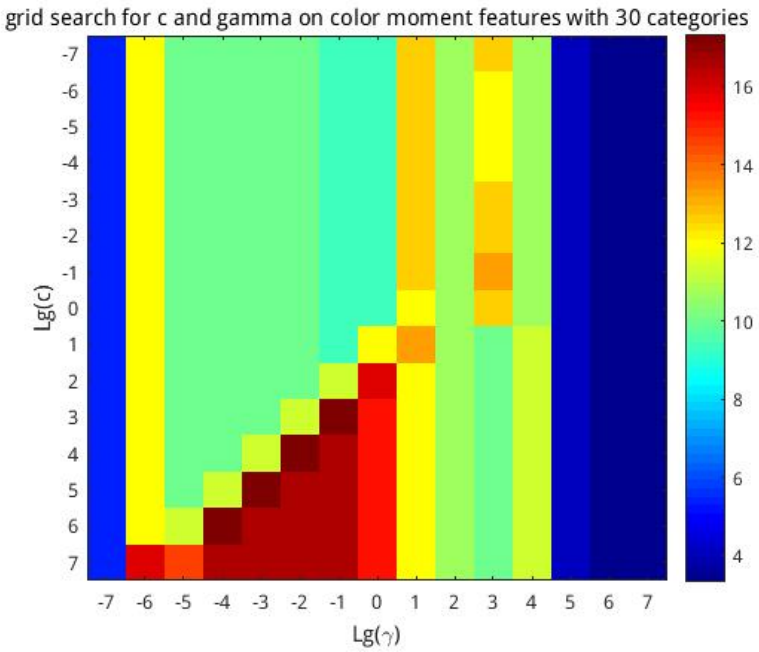

Figure 5.6: CV accuracy of color moments feature of 30 categories

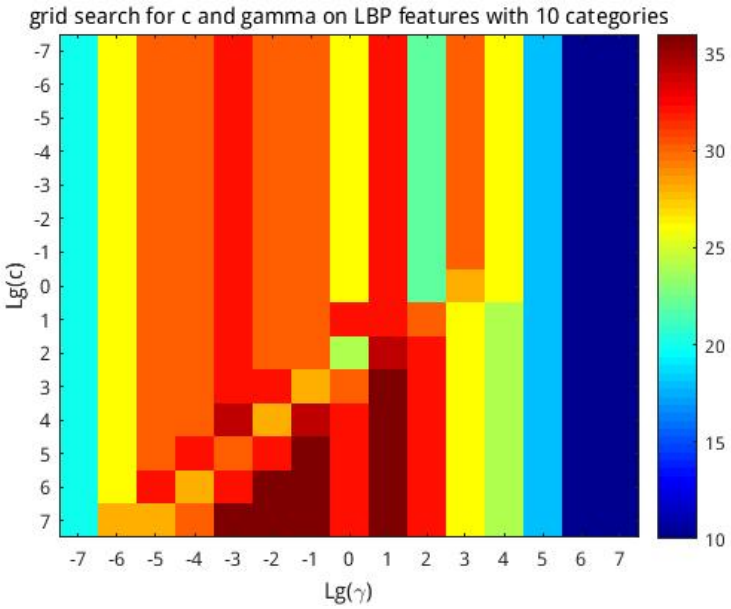

Figure 5.7: CV accuracy of LBP feature of 10 categories

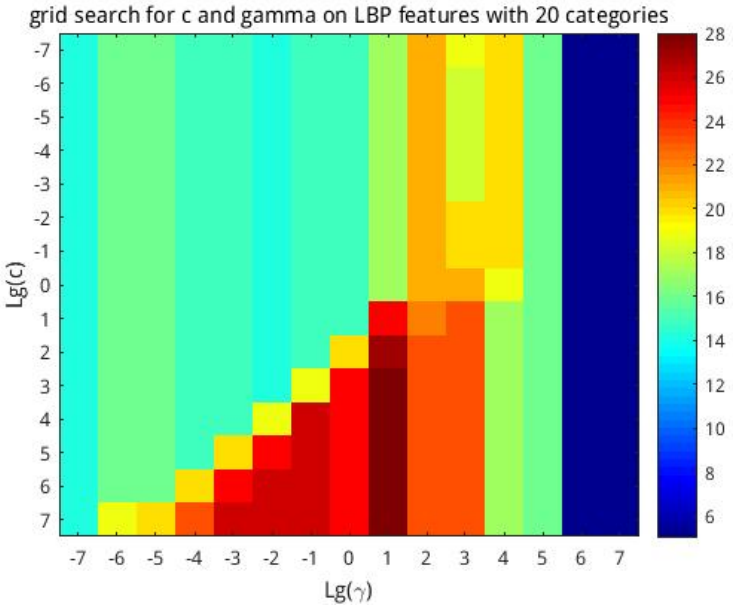

Figure 5.8: CV accuracy of LBP feature of 20 categories

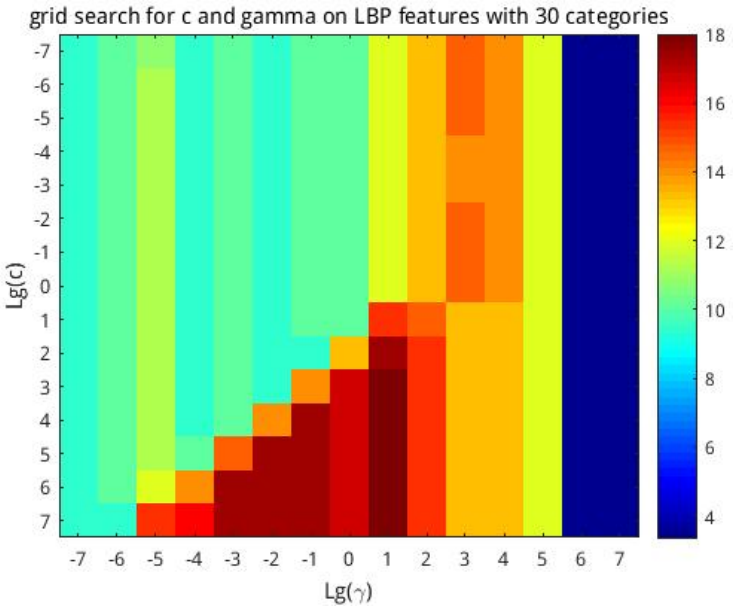

Figure 5.9: CV accuracy of LBP feature of 30 categories

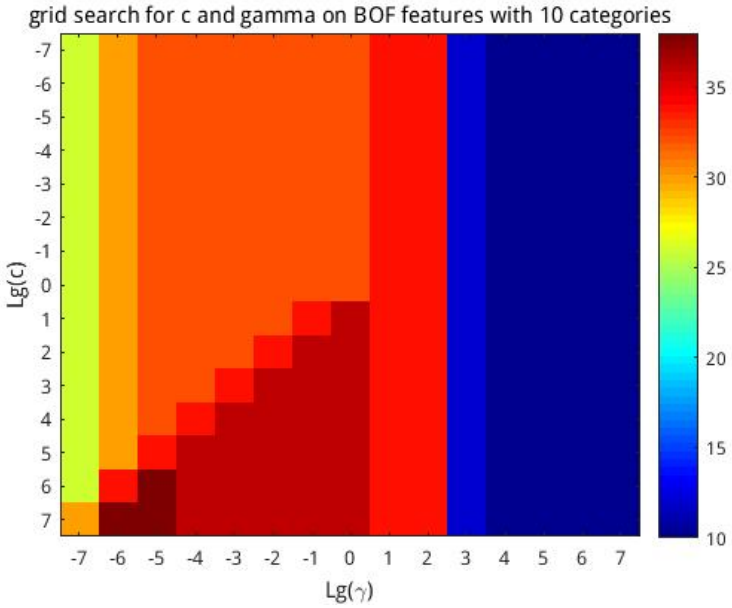

Figure 5.10: CV accuracy of BOF feature of 10 categories

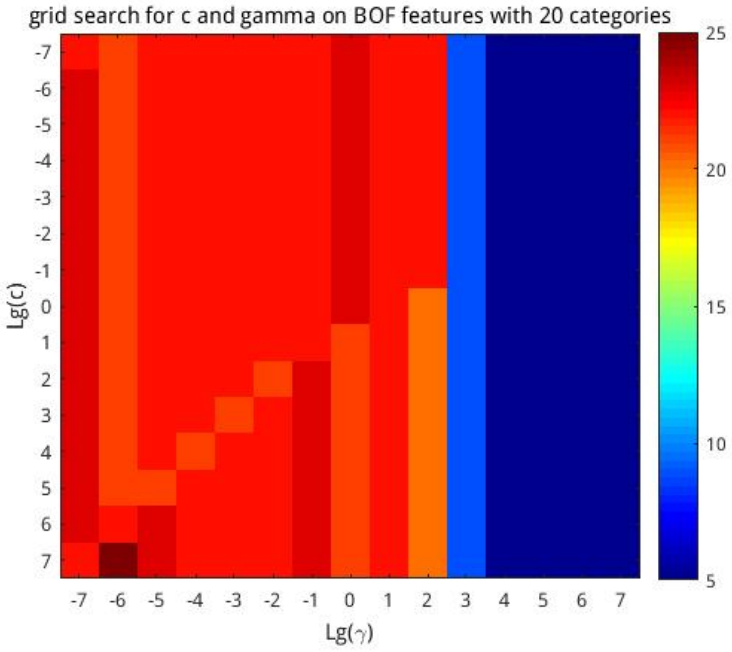

Figure 5.11: CV accuracy of BOF feature of 20 categories

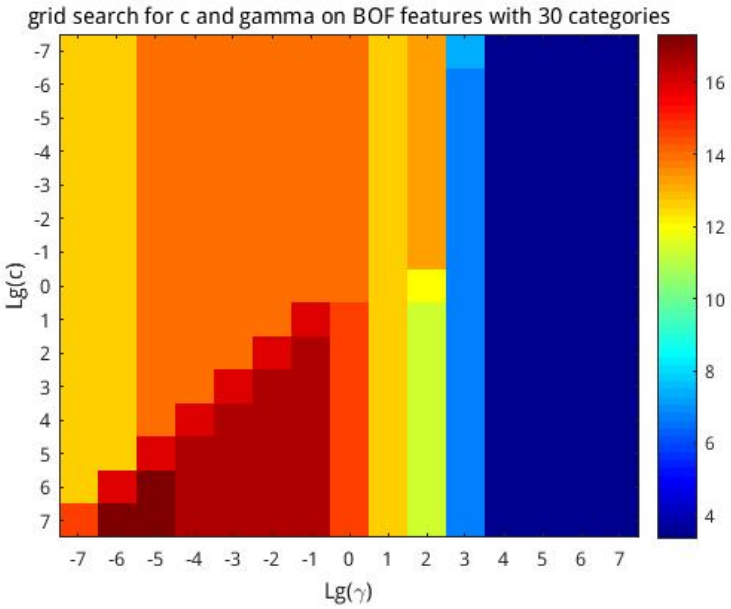

Figure 5.12: CV accuracy of BOF feature of 30 categories

Bibliography

[1] Fei Zuo. *Digital image processing*. Publishing House of Electronics Industry (PHEI), 2014.

[2] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[3] Feifei Huang. Face recognition research based on lbp. Master's thesis, Chongqing University, 2009.

[4] Kinnunen T., Kamarainen JK., Lensu L., and Kälviäinen H. Bag-of-features codebook generation by self-organisation. In *Advances in Self-Organizing Maps. WSOM 2009*, 2009.

[5] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. 2014.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision (2004)*, pages 60–91, 2004.

[7] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. 2015.

[8] Donglai Wei, Bolei Zhou, Antonio Torrabla, and William Freeman. Understanding intra-class knowledge inside cnn. *arXiv:1507.02379*, 2015.