공학석사 학위논문

IMAGE CLASSIFICATION USING SVM CLASSIFIER LEARNED BY ADABOOST METHOD

AdaBoost 방법을 통해 학습된 SVM 분류기를 이용한 영상 분류

2015년 2월

서울대학교 대학원

전기·컴퓨터공학부

이 해 나

IMAGE CLASSIFICATION USING SVM CLASSIFIER LEARNED BY ADABOOST METHOD

지도교수 유 석 인

이 논문을 공학석사학위논문으로 제출함

2014 년 10 월

서울대학교 대학원

전기·컴퓨터공학부

이 해 나

이해나의 석사학위논문을 인준함

2014 년 12 월

위 원 장 : Robert Ian McKay (인)

부위원장 : 유 석 인 (인)

위 원 : Bernhard Egger (인)

Abstract

Image Classification using SVM Classifier Learned by AdaBoost Method

Hae-na Lee
School of Computer Science Engineering
College of Engineering
The Graduate School
Seoul National University

This thesis presents the algorithm that categorizes images by objects contained in the images. The images are encoded with bag-of-features (BoF) model which represents an image as a collection of unordered features extracted from the local patches. To deal with the classification of multiple object categories, the one-versus-all method is applied for the implementation of multi-class classifier. The object classifiers are built as the number of object categories, and each classifier decides whether an image is included in the object category or not. The object classifier has been developed on the AdaBoost method. The object classifier is given by the weighted sum of 200 support vector machine (SVM) component classifiers. Among multiple object classifiers, the classifier with the highest output function value finally determines the category of the object image. The classification efficiency of the presented algorithm has been illustrated on the images from Caltech-101 dataset.

Keywords : Image classification, Object category recognition, Bag-of-features (BoF) model, AdaBoost, Support vector machine (SVM)
Student number : 2013-20866

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recognition is a highly important topic of computer vision field, and it is to analyze and identify all the objects included in an image. Object detection, instance recognition and category recognition are included in the problem of recognition. Some examples of the recognition applications are shown in Figure 1. It presents the results of algorithms on real-time face detection [27], instance recognition [16] and object recognition [20].

Object category recognition, which is also called object classification, is a central recognition problem. It is to classify images of various objects by the

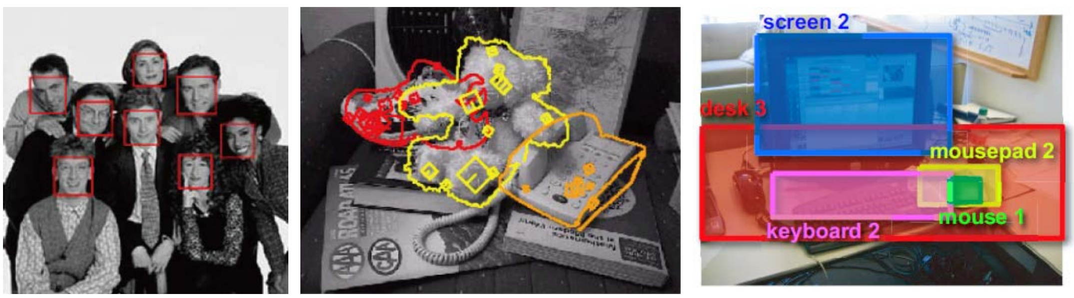

Figure 1. Recognition applications.

object categories. There are broad applications of object category recognition, such as image database annotation, image understanding, and image retrieval.

It is extremely challenging to recognize object categories of images. No algorithm has yet shown the performance level that approximates the level of a two-year-old child [24]. There exist several reasons that make object classification problem difficult. First of all, there are a wide range of object categories. A number of objects constitute the real world. In addition, the variation within a category is another difficulty. The object images that belong to the same category may differ considerably in appearance. They appear in different poses, illuminations and scales. This variation should be captured to correctly identify objects in images.

This thesis deals with the problem of classifying object images. The bag-of-features (BoF) model is applied to represent images. After encoding images with the BoF representation, the classification algorithm determines the category of object images. AdaBoost method is utilized to improve the performance of the classifier. The classifier is constructed with a collection of SVM component classifiers.

The remainder of the thesis is organized as follows. Chapter 2 describes some related work and background information. In Chapter 3, a detailed explanation of the proposed algorithm for image classification is presented. The experiments conducted to evaluate the performance of the proposed algorithm are specified in Chapter 4. Finally, Chapter 5 gives a conclusion of the thesis.

Chapter 2

Related Work

2.1 Image classification approaches

In order to solve the problem of image classification, a number of approaches have been proposed. The part-based model is one of the oldest models. It recognizes an object by detecting constituent parts of the object and evaluating their geometric relationships. Another simple approach is BoF model that represents an image as an unordered collection of local feature descriptors.

Over the years, the BoF model [5, 6] has been one of the most popular and successful approaches in image classification. It is also known as the bag-of-words or bag-of-keypoints model. The main steps of the model are depicted in Figure 2. A general BoF pipeline for image classification is comprised of following stages: feature extraction, codebook construction, feature representation, and classifier design. The BoF model represents an image as a collection of orderless features that are extracted from the local patches, constructs a codebook with visual words which are the representatives of the features, and then computes a histogram representation

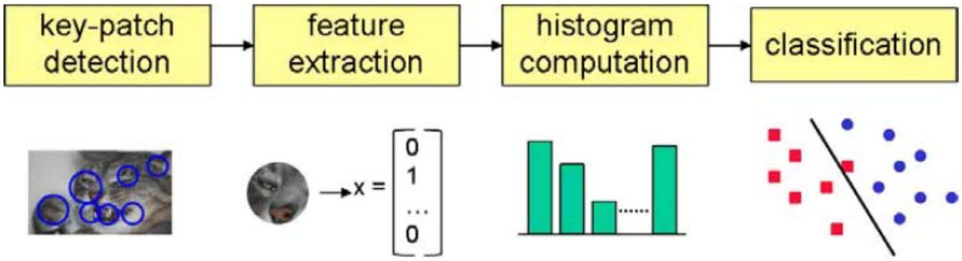

Figure 2. The main steps of the bag-of-features approach.

for the image by quantizing the features into visual words.

Various research works have been performed to improve the performance of the traditional BoF model. It is applied in many image categorization applications, such as scene category recognition [13, 25], object category classification [2, 8] and action recognition [22].

A lot of algorithms using the BoF method have demonstrated outstanding levels of performance. However, the BoF approach disregards the spatial information of local features, which heavily limits the descriptive ability of the image representation. To overcome this limitation, spatial pyramid matching (SPM) scheme [13] which is an efficient extension of the BoF model was proposed. It partitions an image into fine sub-regions and computes histograms of local features that are found within each of the resulting sub-regions. All the histograms are finally concatenated to form a representation of the image.

Figure 3 describes an example of constructing a three-level pyramid. The image contains three feature types that are indicated by circles, diamonds and crosses, respectively. The image is first subdivided into three different levels from level 0 to level 2. Then the features in each spatial region are counted for each level and each feature type.

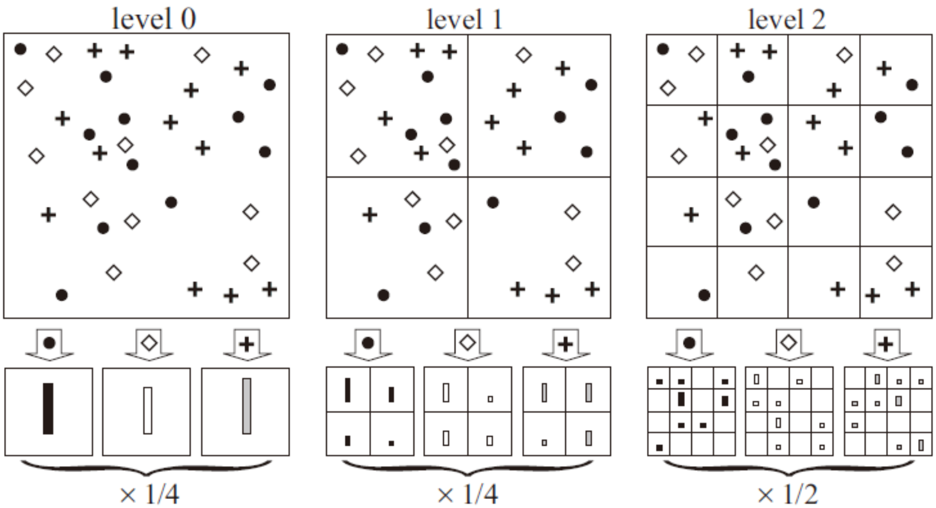

Figure 3. Example of constructing a three-level pyramid in SPM scheme.

In recent years, many different approaches have been suggested to improve method on how local features are encoded to visual words because it is found that the method of encoding local features has a significant impact on the classification performance. The hard-assignment coding used in [13] is the earliest method, and it is modified into other various feature coding methods. The soft-assignment coding [25] assigns a code coefficient for a local feature to each visual word depending on the pairwise distance between the local feature and each of the visual words. To ameliorate the hard-assignment and soft-assignment coding methods, the coding properties such as sparsity, locality and saliency are considered for other encoding methods [12, 28, 29].

The progress in the problem of image classification has also been achieved by developing more powerful classifiers. After encoding local features using the codebook, the resulting BoF representation can be classified by a plethora of classifier models such as SVM [11, 13], nearest neighbor [1], decision tree [2] and boosting [18]. The leading classifier used for image classification is SVM in particular.

Some approaches attempt to improve the performance of classifier by synthesizing classifier models. SVM-KNN [30] proposes a hybrid method of nearest neighbor classifier and SVM. The idea of the classifier is to first find nearest neighbors to a query data and use a local SVM that applies the distance function on the collection of neighbors. The performance of the method is superior to nearest neighbor classifier and SVM.

2.2 Boosting methods

Boosting is an ensemble technique that combines multiple component classifiers, each of which is only moderately accurate, to formulate a strong final classifier showing a better performance than that of any of the component classifiers. The component classifiers are known as weak classifiers because the classification performance of each component classifier is only slightly better than random. However, the boosting method can result in a final classifier with an improved performance. Adaptive Boosting (AdaBoost) [9, 10, 21, 31] is the most widely used boosting algorithm.

AdaBoost is a boosting algorithm that creates each weak classifier using sampled data from the training data. By combining the results of multiple weak classifiers, a strong final classifier is constructed. One key factor that differentiates AdaBoost from other boosting methods is that it adaptively adjusts the weights of training data from which the sampled data are selected. By increasing the weights of the training data that are incorrectly classified, the misclassified training data get to have more chance to be sampled for building weak classifiers. AdaBoost tries to focus on those data that are difficult to be correctly classified and samples more of those data for training the weak classifiers.

The strong final classifier aggregates the classification result of each weak classifier according to its weight, which can be represented as a measure relative to the inverse of the error rate of the weak classifier. The weak classifiers with smaller error rates will have greater weights than those with larger error rates.

The schematic diagram of building a final classifier using AdaBoost method is illustrated in Figure 4. The three weak component classifiers are learned in sequence, and the weights of the training data are adjusted in the course of learning. The weights of the misclassified data are increased, so that they have higher possibilities to be selected as the sampled data used for training subsequent weak classifiers. The set of weak classifiers are linearly combined together to form a final classifier that shows a superior classification performance.

Decision trees or neural networks have been often used for component classifiers of AdaBoost method in many studies [7, 23]. AdaBoostSVM [15] which uses SVMs as component classifiers in AdaBoost method is a variation of the classic AdaBoost. Since strong component classifiers are not appropriate to be used in AdaBoost method, SVM which often shows an excellent classification performance is not used as a component classifier in other AdaBoost methods. However, SVMs with the Gaussian radial basis function (RBF) kernels are properly constructed in AdaBoostSVM algorithm to formulate a final classifier with a high performance rate. For each of the component classifiers, AdaBoostSVM adjusts the Gaussian width value of SVM using RBF kernel. In the training stage, by

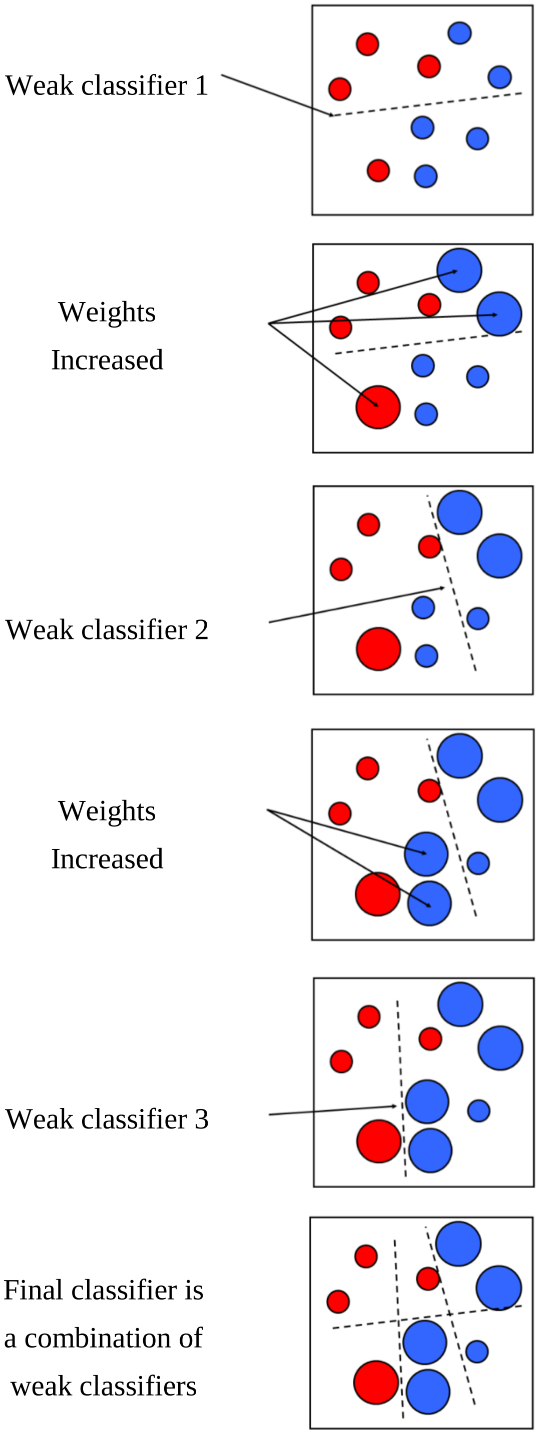

Figure 4. Schematic illustration of AdaBoost.

decreasing the Gaussian width value when the classification performance of the current SVM component classifier is poor, the subsequent SVM component classifier becomes more discriminatory, which leads to an increased learning capability.

The face detection approach with a classifier which is built using the AdaBoost learning [27] was the first to introduce the concept of boosting to the computer vision field. In the approach, a collection of simple classifiers are trained, and then their outputs are combined to formulate an accurate final classifier that can process images and detect faces from the images extremely fast.

Boosting is also utilized as the learning technique for solving object detection and recognition problems [18, 19]. In [18], diverse sets of visual features are used for learning weak hypotheses which are combined later to create a final hypothesis.

2.3 Background

2.3.1 Support vector machine

Support vector machine (SVM) [3, 4] is a supervised learning model for binary classification. In SVM, the optimal hyperplane that separates data points of two classes is the one for which the margin is maximized. Figure 5 shows an illustration of the optimal hyperplane having the maximum margin for the two-class classification problem in a two-dimensional space.

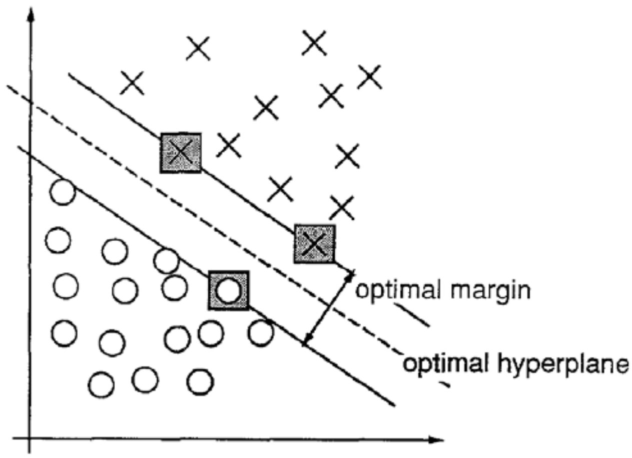

Figure 5. Optimal hyperplane for the two-class classification problem in a two-dimensional space.

The soft-margin SVM [4] is a hyperplane that can separate the training data with a minimal error. Given a set of training data with labels $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, with $y_i \in \{-1, +1\}$, the following optimization problem is solved to find the optimal hyperplane:

$$\arg\min_{\mathbf{w},\xi,b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{subject to } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for all } i = 1, \dots, N$$

The parameter C is a positive constant that controls the tradeoff between constraint violation and regularization.

For the case of linearly non-separable data, the kernel trick can be applied to map the input space to the transformed feature space having a higher dimension. It allows the algorithm to find the optimal hyperplane with the maximum margin in the transformed feature space. There are various kernels including polynomial kernel, Gaussian radial basis function kernel, sigmoid kernel and wave kernel.

Chapter 3

Proposed Algorithm

The proposed algorithm utilizes BoF model that has shown a promising result for classifying categories of object images. It aims to improve the image classification performance by applying AdaBoost method to learn a multi-class classifier.

The algorithm first detects and extracts the Scale Invariant Feature Transform (SIFT) descriptors as features from the images. Then, it establishes a codebook using the SIFT descriptors. The codebook is used for encoding an image as BoF representation which is a histogram of visual words. These representations are entered as input to a classifier model to perform the task of image classification. The classifier is learned by AdaBoost method, and it is built by combining multiple SVM component classifiers.

3.1 SIFT feature extraction

In the first stage of the feature extraction, SIFT descriptors are obtained from images. SIFT descriptor [17] is one of the most popular features that are used in computer vision area to describe local features contained in images. It is suitable to be used for the image classification task because it characterizes image region around a given point, and it is partially invariant to intensity and contrast changes, and small geometric deformations.

To compute the SIFT descriptor, interest point (keypoint) in the image is found by the SIFT detector. Then, gradient magnitude and orientation are calculated for every pixel within a 16×16 pixel region around the interest point. This region is divided into 4×4 grid of cells, and an eight-dimensional histogram of the image orientations is computed for each cell. The 16 histograms for the cells, each of which contains 8 bins are concatenated to make a 128-dimensional vector (4×4×8=128). The creation of SIFT descriptor is depicted in Figure 6. The red circle

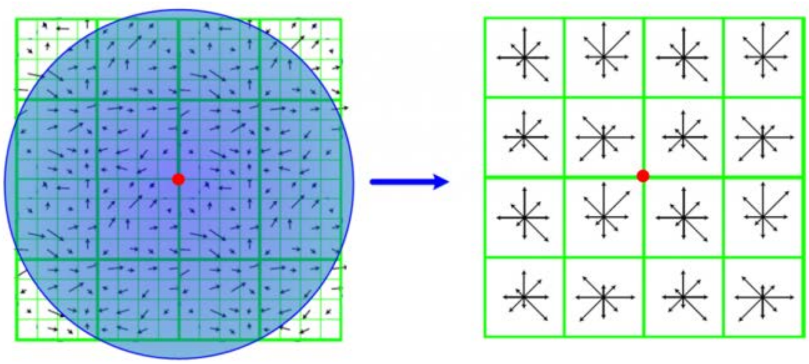

Image gradients Keypoint descriptor

Figure 6. The creation of SIFT descriptor.

indicates the location of the interest point.

In our method, a Pyramid Histogram Of visual Words (PHOW) descriptor [2] is used as a feature for describing image. The PHOW feature is a variant of the SIFT descriptor. It is a densely extracted SIFT descriptor that is applied at multiple scales. In other words, SIFT descriptors are computed on a dense grid of locations at multiple scales. In Figure 7, the left figure shows a dense grid of locations with uniform spacing, and the right figure represents four different scales. The value of the uniform spacing is set to 2, and the values of four scales are assigned as 4, 6, 8 and 10. The efficient code that computes the PHOW features is available in the open source VLFeat library [26].

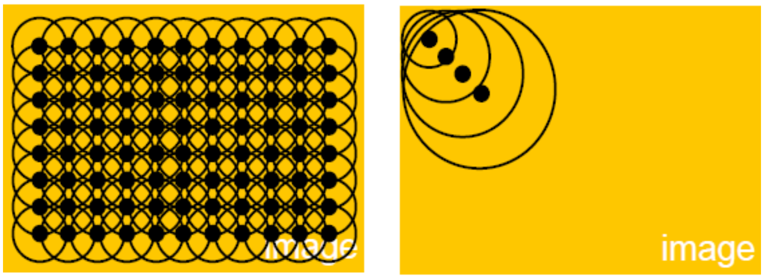

Figure 7. PHOW features.

3.2 Codebook construction

To build a codebook of visual words, K-means clustering algorithm is applied on the set of SIFT descriptors obtained from the feature extraction step. Given the data points, the objective of K-means algorithm is to find K cluster centers, where the sum of distances between data points and the assigned centers is minimized.

Lloyd's algorithm [14], which is the most common K-means method, is used. Lloyd's algorithm alternates the following two steps:

1. Quantization

   Each data point is assigned to one of the K cluster centers, which is the closest to the data point.

2. Center estimation

   Each cluster center is updated to minimize its average distance to the data points assigned to it. This can be done by calculating the mean value of the assigned data points.

These two stages of assigning data points to the cluster centers and optimizing the cluster centers correspond respectively to the E (expectation) and M (maximization) steps of the EM algorithm.

K mean values found by the algorithm are the representatives that can compose the codebook of visual words. The value of K should be set to an optimal value that can decide the size of codebook properly representing the original SIFT descriptors.

3.3 Bag-of-features representation

The SIFT descriptors of images can be represented using the codebook of visual words. For each SIFT descriptor in image, K nearest neighbors of visual words are found from the codebook to represent the descriptor instead. The overall number of times that each visual word is used to represent the SIFT descriptor can be expressed as a histogram of visual words.

The images from different object categories are expected to have different histograms of visual words. This difference of distributions will be learned by classifier to perform the image classification.

3.4 Classifier design

Designing a classifier is the central stage of our algorithm. In order to construct a multi-class classifier that can deal with the problem of classifying images from multiple object categories (classes), the one-versus-all rule with a winner-takes-all strategy is used. It reduces multi-class classification problem to multiple two-class problems. Thus, the binary object classifiers $f_1, f_2, \cdots, f_K$ are learned, where K is the number of object categories. Each classifier f_k can decide whether an image is included in class k or not. An image can be classified by evaluating all object classifiers, and the object classifier with the highest output function value decides the class of image.

AdaBoostSVM [15] is implemented in the one-versus-all manner. Let us refer to the object classifier f_k for class k as a sub-classifier. For each sub-classifier, the adjusted AdaBoostSVM is built, so each sub-classifier is constructed with a set of SVM component classifiers. The classification result of each sub-classifier is the weighted sum of SVM component classifiers. The final multi-class classification is performed by finding a class corresponding to the sub-classifier with the highest output function value.

The kernel trick is applied for SVM component classifiers as in [15]. The Gaussian radial basis function (RBF) kernel is used for the kernel function. The RBF kernel is defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

where σ is the Gaussian width.

Some adjustments have been made to the original AdaBoostSVM algorithm to apply it to multi-class classification. The first adjustment to AdaBoostSVM algorithm is learning a fixed number of T component classifiers to create a sub-classifier. It ensures that each sub-classifier consists of the same number of component classifiers. Since the one-versus-all method is used, balancing the size of each sub-classifier is necessary to acquire the stability of classification system as a whole.

Another adjustment is recreating a component classifier if it has zero error. This is done to correctly follow the inherent characteristic of the AdaBoost algorithm. Ensuring the weakness of component classifiers is essential to achieve an improved performance through boosting method. The component classifier having zero error is a strong classifier, so it is improper to include such a strong classifier as a component classifier.

The adjusted algorithm of AdaBoostSVM combines a collection of T SVM component classifiers to create a sub-classifier f_k which is a strong classifier. After creating all K sub-classifiers $(f_1, f_2, \cdots, f_K)$, the multi-class classification algorithm uses them to perform the task of multi-class classification.

The algorithms of multi-class classification and adjusted AdaBoostSVM can be described as follows.

Algorithm: Multi-class classification

1. Input: a set of training data with labels $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$.
2. For $k = 1$ to K
   (1) Construct a sub-classifier f_k of class k, using adjusted AdaBoostSVM algorithm.
3. Output:

$$F(\mathbf{x}) = \arg\max_k f_k(\mathbf{x})$$

Algorithm: Adjusted AdaBoostSVM

1. Input: a set of training data with labels $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, initial value of the Gaussian width σ_{ini}, minimal value of the Gaussian width σ_{min}, step value of the Gaussian width σ_{step}.
2. Initialize the weights of training data, $w_i = 1/N$ for all $i = 1, \dots, N$.
3. Set the labels of the training data from class k as label 1, and set the labels of the training data from other classes as label -1.
4. For $t = 1$ to T
   (1) Get M weighted samples from the set of training data using w_i.
   (2) Train a SVM component classifier h_t to M training samples.
   (3) Compute the training error ε_t of h_t:

$$\varepsilon_t = \sum_{i=1}^{N} w_i \, \mathbb{I}(y_i \neq h_t(\mathbf{x}_i))$$

   (4) If $\varepsilon_t > 0.5$, decrease σ value by σ_{step} (or set σ value as σ_{min} if the decreased σ value is smaller than σ_{min}) and go to (1).
   If ε_t is 0, go to (1).
   (5) Set the weight α_t of h_t:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$$

   (6) Update w_i:

$$w_i \leftarrow w_i \cdot \exp(\alpha_t \, \mathbb{I}(y_i \neq h_t(\mathbf{x}_i))$$

   (7) Re-normalize w_i:

$$w_i \leftarrow \frac{w_i}{\sum_{i=1}^{N} w_i}$$

5. Output:

$$f_k(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t \, h_t(\mathbf{x})$$

Chapter 4

Experiments

In this chapter, we demonstrate the experiments performed to quantify the performance of the proposed algorithm. The multi-class classification accuracies of the typical SVM classifier and the proposed classifier learned by AdaBoost method are compared to show the improved performance of the proposed classifier.

4.1 Dataset

For the experiments, object images of Caltech-101 dataset are used. The Caltech-101 dataset collected by Fei-Fei *et al.* [8] has been extensively used as one of the benchmark datasets for the object classification task. It consists of 9144 images from 101 object categories (animals, plants, vehicles, etc.) and an additional background category, making the total number of categories 102. The number of images for each category varies from 31 to 800, and the size of each image is about 300 x 200 pixels. The significance of the dataset is that there is a large variance (variability) within a category.

Among 102 categories, ten categories containing the largest number of images were chosen for our experiments. When selecting the categories, we excluded “Background” and “Faces_easy” categories. “Background” category includes diverse background images that are not suitable for the task of classifying object images. There are two categories of face images, “Faces_easy” and “Faces”, and the images from “Faces_easy” category contain larger face segments, which makes them easier to classify compared to the images from “Faces” category. We only selected “Faces” category to perform the experiments using more diverse object images. The selected categories for experiments are as follows.

· Airplanes

· Bonsai

· Car_side

· Chandelier

· Faces

· Hawksbill

· Ketch

· Leopards

· Motorbikes

· Watch

Some example images of ten categories are shown in Figure 8. For each of the categories, 70 images were used for training, and 30 images were used for testing.

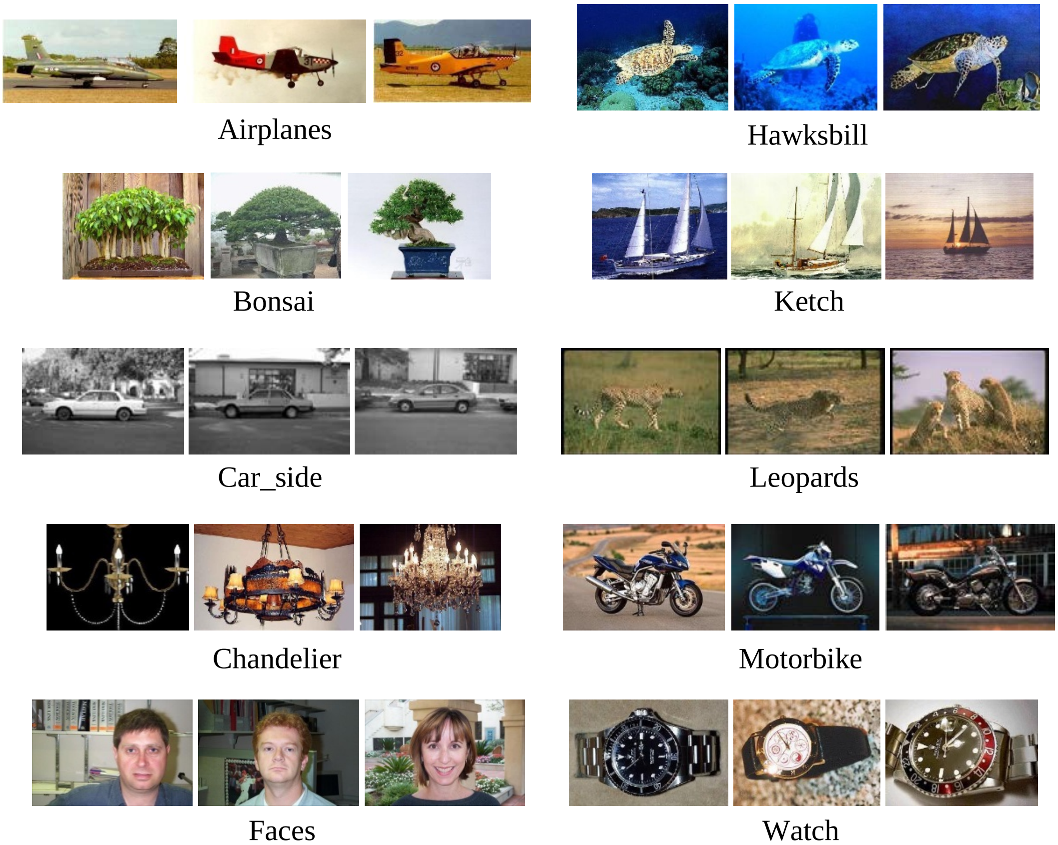

Figure 8. Example images of ten categories from the Caltech dataset.

4.2 Bag-of-features representation

The codebook containing 300 visual words was built for the bag-of-features representation. For each SIFT descriptor, 10 nearest neighbors of visual words are selected from the codebook to represent it.

Figure 9 shows the histogram representations of images from ten different categories. It can be easily seen that the histograms have different distributions of visual words. This difference of distributions will be learned by classifiers.

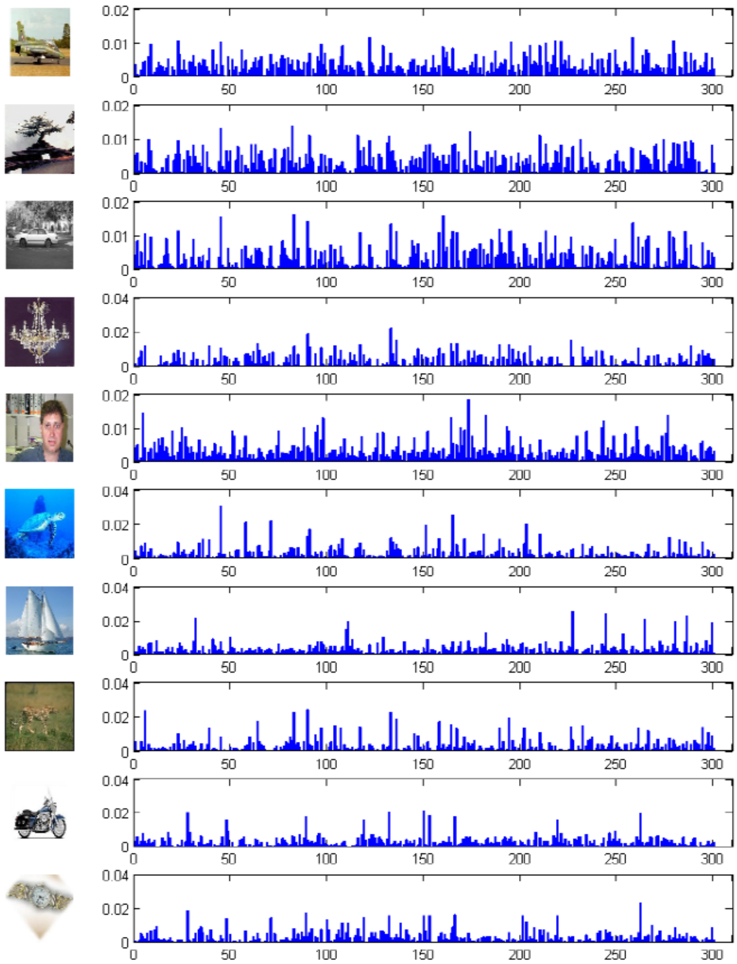

Figure 9. Histogram representations of images from ten categories.

4.3 Classifiers

In order to compare the classification performance of the proposed algorithm, we construct classifiers using either with AdaBoost or without AdaBoost. Also, two different kernel functions are applied for SVM. They are RBF kernel and linear kernel. The linear kernel is dot product, which is defined as

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{x}'$$

In total, four different classification algorithms are constructed for comparison. We will refer to the algorithms in abbreviations as shown in Table 1 for convenience.

Algorithm	Abbreviation
SVM using RBF kernel	$\mathrm{SVM}_{\mathrm{RBF}}$
AdaBoost with SVM component classifiers using RBF kernel	$\mathrm{AB}_{\mathrm{RBF}}$
SVM using linear kernel	$\mathrm{SVM}_{\mathrm{LIN}}$
AdaBoost with SVM component classifiers using linear kernel	$\mathrm{AB}_{\mathrm{LIN}}$

Table 1. Abbreviations of the classification algorithms.

When applying AdaBoost method, the number of component classifiers per each category is set to 200 to construct a sub-classifier. Out of 700 histogram representations for the training images, 14 histogram representations are sampled and used for building each of component classifiers.

4.4 Classification results

All experiments were repeated thirty times for each of the classification algorithms and parameter settings. The classification accuracy is reported as the mean and standard deviation of the results from the individual runs.

Table 2 contains the classification results of the four classification algorithms. The column indicates the value of parameter C for the soft margin SVM. The smaller the value of C becomes, the more regularized the algorithm becomes. The classification results with AB_{RBF} had shown the best performance among all algorithms.

Accuracy (%)

C	SVM_{RBF}	AB_{RBF}	SVM_{LIN}	AB_{LIN}
1	69.2±1.0	**80.5±1.7**	76.9±1.3	74.8±2.1
5	76.1±1.0	79.9±1.4	76.9±1.3	74.9±2.0
10	77.3±0.9	78.6±1.8	76.9±1.3	74.9±1.8

Table 2. Classification results of the classification algorithms.

For the SVMs using the RBF kernel, AdaBoost method had boosted the classification accuracies. However, SVMs using the linear kernel had shown no improvement. The parameter C acted differently depending on the classification algorithms. For the case of AB_{RBF}, smaller C value seemed to perform better. This was due to the fact that the SVM components classifiers tend to have accuracies around 50% when a smaller value is set as the value of C, which leads to boosting effect. However, the value of C had almost no impact on the classification algorithms using the linear kernel.

Table 3 represents the confusion matrix of the experiment using AB_{RBF} which results in a high accuracy rate of 84.0%. The numbers on the main diagonal represents the numbers of images that were correctly classified. For the case of "Airplanes" category, out of 30 test images, 24 images were correctly classified. Some misclassification had been made, e.g. three images were incorrectly classified as "Bonsai" category.

Accuracy: 84.0%

True Category	**Classified Category**									
	Airplanes	Bonsai	Car_side	Chandelier	Faces	Hawksbill	Ketch	Leopards	Motorbikes	Watch
Airplanes	**24**	3		2					1	
Bonsai		**24**	1	1			1	1	1	1
Car_side			**30**							
Chandelier	2	2		**19**	1	3				3
Faces				1	**29**					
Hawksbill		1		1		**24**		2		2
Ketch		3	1				**25**		1	
Leopards		1				1	1	**27**		
Motorbikes	1			5		1			**23**	
Watch		1		2						**27**

Table 3. Confusion matrix of a classification result with AB_{RBF}

Generally, the classification accuracies for “Car_side” category were superior, while those for “Chandelier” category were inferior. In Figure 10 and Figure 11, the histogram representations of images from “Car_side” and “Chandelier” categories are depicted respectively. It can be found that the histogram representations of “Car_side” category have similar distributions, which is a better

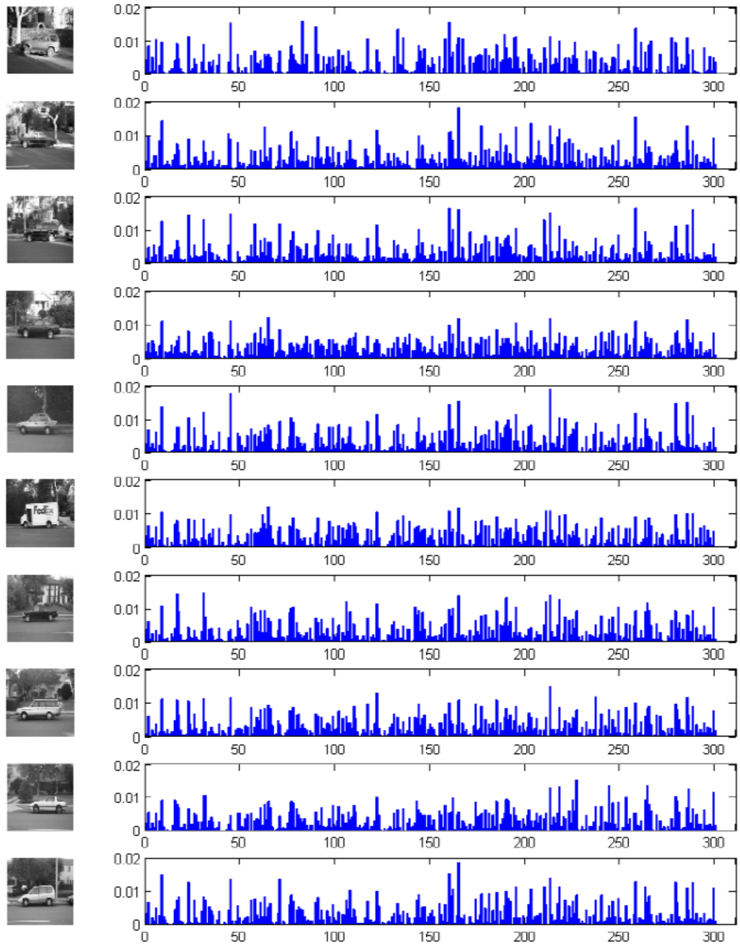

Figure 10. Histogram representations of images from “Car_side” category

condition to build a superior sub-classifier. However, the histogram representations of “Chandelier” category show disparate distributions. To improve the algorithm, the encoding method should be modified, so that images from same category share similar distributions.

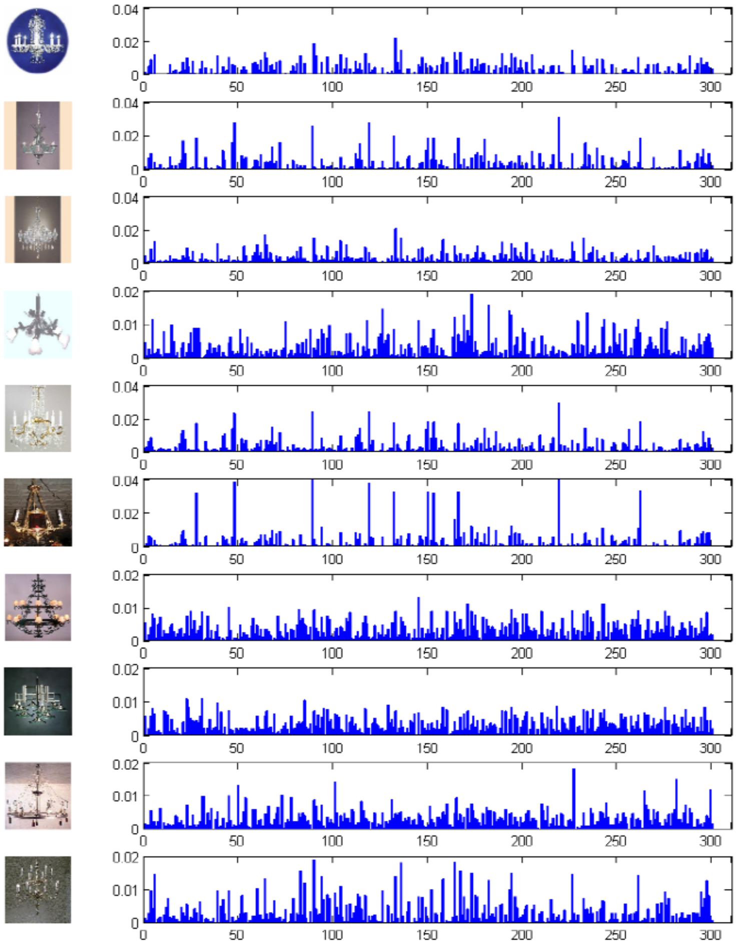

Figure 11. Histogram representations of images from “Chandelier” category

Chapter 5

Conclusion

In this thesis, we deal with the problem of object image classification. The BoF model is used for representing images as histograms of visual words. The main contribution of the thesis is the development of multi-class classifier using SVM and AdaBoost method, which shows a superior classification performance over the traditional SVM classifier. It showed promising results, so that it can be extended to general object classification.

For the future work, the proposed algorithm can be implemented with other feature coding methods to improve the encoding ability. The algorithm can be also extended to cover more diverse categories of object images.

Bibliography

[1] Boiman, O., Shechtman, E., & Irani, M. (2008, June). In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.

[2] Bosch, A.; Zisserman, A.; Munoz, X., "Image Classification using Random Forests and Ferns," *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on,* vol., no., pp.1,8, 14-21 Oct. 2007

[3] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). ACM.

[4] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20.3 (1995): 273-297.

[5] Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004, May). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision,* ECCV (Vol. 1, No. 1-22, pp. 1-2).

[6] Csurka, G., Dance, C. R., Perronnin, F., & Willamowski, J. (2006). Generic visual categorization using weak geometry. In *Toward Category-Level Object Recognition* (pp. 207-224). Springer Berlin Heidelberg.

[7] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning, 40*(2), 139-157.

[8] Fei-Fei, L.; Fergus, R.; Perona, P., "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested

on 101 Object Categories," *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, vol., no., pp.178,178, 27-02 June 2004

[9] Freund, Y., & Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer System Sciences*. 55, 1 (August 1997), 119-139.

[10] Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence*, 14(771-780), 1612.

[11] Grauman, K., & Darrell, T. (2005, October). The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Vol. 2, pp. 1458-1465). IEEE.

[12] Huang, Y., Huang, K., Yu, Y., & Tan, T. (2011, June). Salient coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 1753-1760). IEEE.

[13] Lazebnik, S.; Schmid, C.; Ponce, J., "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol.2, no., pp.2169,2178, 2006

[14] Lloyd, S. (1982). Least squares quantization in PCM. *Information Theory*, IEEE Transactions on, 28(2), 129-137.

[15] Li, X., Wang, L., & Sung, E. (2008). AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21(5), 785-795.

[16] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). IEEE.

[17] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

[18] Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In *Computer Vision-ECCV 2004* (pp. 71-84). Springer Berlin Heidelberg.

[19] Opelt, A., Pinz, A., Fussenegger, M., & Auer, P. (2006). Generic object recognition with boosting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3), 416-431.

[20] Russell, B., Torralba, A., Liu, C., Fergus, R., & Freeman, W. T. (2007). Object recognition by scene alignment. In *Advances in Neural Information Processing Systems* (pp. 1241-1248).

[21] Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3), 297-336.

[22] Schuldt, C., Laptev, I., & Caputo, B. (2004, August). Recognizing human actions: a local SVM approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (Vol. 3, pp. 32-36). IEEE.

[23] Schwenk, H., & Bengio, Y. (2000). Boosting neural networks. *Neural Computation*, 12(8), 1869-1887.

[24] Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer.

[25] van Gemert, J. C., Geusebroek, J. M., Veenman, C. J., & Smeulders, A. W. (2008). Kernel codebooks for scene categorization. In *Computer Vision–ECCV 2008* (pp. 696-709). Springer Berlin Heidelberg.

[26] Vedaldi, A. and Fulkerson, B. 2010. Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia* (MM '10). ACM, New York, NY, USA, 1469-1472.

[27] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), 137-154.

[28] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., & Gong, Y. (2010, June). Locality-constrained linear coding for image classification. In *Computer*

Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 3360-3367). IEEE.

[29] Yang, J., Yu, K., Gong, Y., & Huang, T. (2009, June). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 1794-1801). IEEE.

[30] Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (Vol. 2, pp. 2126-2136). IEEE.

[31] Zhu, J., Zou, H., Rosset, S., & Hastie, T. (2009). Multi-class adaboost. *Statistics and Its Interface*.

국문초록

본 논문에서는 영상에 포함된 물체로 영상을 분류하는 알고리즘을 제안한다. 영상들은 지역 패치에서 추출된 특징들의 집합으로 영상을 나타내는 bag-of-features 모델로 표현된다. 다수의 물체 카테고리 분류를 다루기 위하여, 다중 클래스 분류기의 구현에 one-versus-all 방법이 적용되었다. 물체 분류기는 물체 카테고리 수만큼 만들어지고, 각각의 물체 분류기는 영상이 해당 물체 카테고리에 포함되는지 여부를 결정한다. 물체 분류기는 아다부스트 방법으로 개발되었다. 물체 분류기는 200개의 서포트 벡터 머신 구성 분류기들의 가중치 합으로 구성된다. 여러 물체 분류기들 가운데, 가장 높은 출력 함수 값을 가지는 분류기가 최종적으로 물체 영상의 카테고리를 결정한다. 제안된 알고리즘의 분류 능률은 Caltech-101 데이터 세트의 영상들로 보여진다.

주요어 : 영상 분류, 물체 범주 인식, Bag-of-features (BoF) 모델, 아다부스트, 서포트 벡터 머신

학 번 : 2013-20866