Journal of Computers Vol. 29 No. 5, 2018, pp. 244-249
doi:10.3966/199115992018102905019

Multiclass Object Classification Using Covariance Descriptors with Kernel SVM

Chun-Yi Tsai*, Wen-Hsiu Chung

Department of Computer Science and Information Engineering, National Taitung University, Taitung 950, Taiwan
cytsai@nttu.edu.tw, u10011235@ms100.nttu.edu.tw

Received 7 January 2018; Revised 7 July 2018; Accepted 7 August 2018

Abstract. Feature descriptor is a crucial part for image object detection and classification in computer vision. This study adopts the covariance descriptor which is a ROI(region of interest) based feature reserving the integrity of regions with rotation and scale invariant, combining with kernel SVM for multiclass object classification. The experimental results show that the combination of covariance descriptor and kernel SVM is very feasible and practical to be applied on dataset images in which the foreground objects are the major portions and the backgrounds are relatively simple.

Keywords: covariance feature, kernel method, object classification, SVM

1 Introduction

Covariance descriptor is a region of image features which can be efficiently computed by integral images [3]. It was well-applied by Tuzel [1, 4] for human detection and object classification. The covariance descriptor is a region-based feature vector with a finite dimension consisting of several image features to represent an image object. It can be seen as a collection of features which effectively represent distinctiveness of a region of interest due to its scale and rotation invariant. On the other hand, the well-known kernel methods applied on SVM is widely-used in object classification. Kernel SVM maps data into a feature space with high dimension, and uses a function in RKHS (Reproducing Kernel Hilbert Space) to be as kernel function to facilitate the computation of inner products in the original space, and thus separates data in feature space by finding the hyperplanes and support vectors with the largest margins. In this paper, we adopt kernel SVM [6-7] and take covariance descriptors extracted from images to be as training and testing samples for multiclass object training and classification to study the practicality of the combination of covariance descriptor and kernel SVM. Although in recent days, many novel classification approaches, such as deep learning algorithms are proposed, our method is also efficient, especially in case of the image objects with relatively simple backgrounds.

2 Covariance Descriptors

The covariance descriptors was adopted in Tuzel's work [1, 4] to apply on object matching and textures classification. A covariance descriptor is a region descriptor represented by the covariance matrix of image features. Since a covariance matrix is also a symmetric positive definite matrix, the distance between any two covariance matrices can be evaluated with eigenvalues in previous analysis [5] for similarity measurement. Tuzel [2] further extended covariance descriptors to be classifiable using LogitBoost on a tangent space of Riemannian manifold. Harandi [10] applied kernel methods on Riemannian manifold, and suggested kernel k-means classification on high dimensional space to promote accuracy.

* Corresponding Author

2.1 Covariance Matrix

In this study, we use the covariance descriptor which is well applied by Tuzel [1-2, 4] to be as the feature extraction method. The essence of covariance descriptor in his works is elaborated as follows. A feature image $F(x, y, i)$ is an extracted image with dimension $W \times H \times d$ from an original image. $F(x, y, i)$ represents the i^{th} feature value inside the feature vector at the pixel (x, y). The covariance descriptor is also a covariance matrix denoted by C_R to represent a region of image features in F. Let $\{Z_k(i)\}_{k=1...S, i=1,2,...d} = F(x, y, i)$ denote the i^{th} feature value inside the feature vector, and let $\mu(i)$ be the mean of $Z_k(i)$. Then, by definition of covariance matrix, we have

$$C_R(i,j) = \frac{1}{S-1}\sum_{k=1}^{S}(Z_k(i)-\mu(j))(Z_k(j)-\mu(j)).$$

Replace $\mu(i)$ and $\mu(j)$ by $\frac{1}{S}\sum_{k=1}^{S}Z_k(i)$ and $\frac{1}{S}\sum_{k=1}^{S}Z_k(j)$ respectively, it gives that

$$C_R(i,j) = \frac{1}{S-1}[\sum_{k=1}^{S}Z_k(i)Z_k(j) - \frac{1}{s}\sum_{k=1}^{S}Z_k(i)\sum_{k=1}^{S}Z_k(j)].$$

2.2 Computation Using Integral Image

The efficient integral image technique used in [3], can be applied on feature image calculation to facilitate the covariance descriptor of a region. Assume that $P(x', y', i), P: W \times H \times d \to R,$ is a tensor function of the integral image of the feature image F associated with feature i, and $R(1,,1: x', y')$ represents a region of rectangle containing S pixels.

Then $P(x', y', i) = \sum_{x \le x', y \le y'} F(x, y, i) = \sum_{k=1}^{S} Z_k(i).$ Similarly, let $Q: W \times H \times d \times d \to R,$ be another integral image that $Q(x', y', i, j) = \sum_{x \le x', y \le y'} F(x, y, i)F(x, y, j) = \sum_{k=1}^{S}[Z_k(i)Z_k(j)]$.

Define $\boldsymbol{P}_{x,y}$ and $\boldsymbol{Q}_{x,y}$ be the following matrices.

$$\boldsymbol{P}_{x,y} = [P(x, y, 1), P(x, y, 2), ..., P(x, y, d)]^T$$

$$\boldsymbol{Q}_{x,y} = \begin{bmatrix} Q(x,y,1,1) & Q(x,y,1,2) & \ldots\ldots & Q(x,y,1,d) \\ Q(x,y,2,1) & Q(x,y,2,2) & \ldots\ldots & Q(x,y,2,d) \\ & \vdots & & \\ Q(x,y,d,1) & Q(x,y,d,2) & \ldots\ldots & Q(x,y,d,d) \end{bmatrix}$$

Then the covariance matrix C_R can be represented by $\boldsymbol{P}_{x,y}$ and $\boldsymbol{Q}_{x,y}$ such that

$$\begin{aligned} C_R(i,j) &= \frac{1}{S}[\sum_{k=1}^{S}Z_k(i)Z_k(j) - \frac{1}{S}\sum_{k=1}^{S}Z_k(i)\sum_{k=1}^{S}Z_k(j)] \\ &= \frac{1}{S}[Q(x,y,i,j) - \frac{1}{S}P(x,y,i)P(x,y,j)] \\ &= \frac{1}{S}[\boldsymbol{Q}_{x,y}(i,j) - \frac{1}{S}\boldsymbol{P}_{x,y}\boldsymbol{P}_{x,y}{}^T P(i,j)]. \end{aligned}$$

Thus, the covariance matrix C_R in the region $R(1,1: x', y')$ is

$$\boldsymbol{C}_{\boldsymbol{R}(1,1;x,y)} = \frac{1}{S}[\boldsymbol{Q}_{x,y} - \frac{1}{S}\boldsymbol{P}_{x,y}\boldsymbol{P}_{x,y}{}^T].$$

Therefore, the covariance matrix in the region $\boldsymbol{C}_{\boldsymbol{R}(x',y':x'',y'')}$ as illustrated in Fig. 1, becomes

$$\boldsymbol{C}_{\boldsymbol{R}(x',y':x'',y'')} = \frac{1}{S-1}[\sum_{k=1}^{S} Z_k(i)Z_k(j) - \frac{1}{S}\sum_{k=1}^{S} Z_k(i)\sum_{k=1}^{S} Z_k(j)] = \frac{1}{S-1}[\boldsymbol{Q}_{x'',y''} + \boldsymbol{Q}_{x'\text{-}1,y'\text{-}1} - \boldsymbol{Q}_{x'',y'\text{-}1} - \boldsymbol{Q}_{x'\text{-}1,y''} - \frac{1}{S}$$
$$(\boldsymbol{P}_{x'',y''} + \boldsymbol{P}_{x'\text{-}1,y'\text{-}1} - \boldsymbol{P}_{x'\text{-}1,y''} - \boldsymbol{P}_{x'',y'\text{-}1})(\boldsymbol{P}_{x'',y''} + \boldsymbol{P}_{x'\text{-}1,y'\text{-}1} - \boldsymbol{P}_{x'\text{-}1,y''} - \boldsymbol{P}_{x'',y'\text{-}1})^T].$$

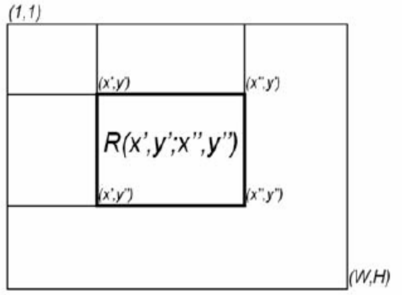

Fig. 1. Integral image

3 Kernel Method

3.1 Feature Space and Kernel Function

The aim of kernel method [6] is to promote the accuracy of classification when samples are not distinguishable enough in the original space. By applying a nonlinear feature mapping to map samples to a higher dimensional space, i.e., the feature space, hyperplanes can be found in feature space to partition these mapping samples. The inner product of any pair of samples in the feature space can be computed by the kernel function taking their inverse mappings in the original space as inputs. Thus, the kernel function is used to fast inner product computation to find distance between them. Many widely-used kernel functions, such as linear kernel, polynomial kernel, and RBF (Radial Basis Function) kernel, can be applied with classification algorithms or classifiers. In this study, we adopt Lin's libsvm [8] to apply his kernel SVM [6-7] using the above three kernel functions combined with covariance descriptors to perform multiclass object classification.

3.2 The RBF Kernel

The RBF kernel is a well-known kernel function and widely applied in kernel-based classification or clustering applications due to its high distinctiveness. However, parameter settings in RBF are crucial effects to the accuracy of classification. The RBF kernel has two important properties. First, the length of each sample in the feature space is a unity. Second, function values of the RBF kernel are positive. Each sample is mapped to a hypersphere by RBF kernel, and the value of the RBF kernel for this sample is the cosine value of this sample. Since the length of each sample is unity, it facilitates the similarity determination between any pair of samples by the angle. Because the function values of the RBF are always positive, it gives that two samples are the most dissimilar if their included angle reaches 90 degree. This property can be applied in the k-fold cross validation [9] to find the feasible parameters C and γ for RBF kernel.

4 Experiment Results

4.1 Object Detection

Since the covariance matrix is symmetric, for any $d \times d$ covariance matrix, we can represent it using a vector with dimension $d(d+1)/2$. In this experiment, the feature vector of a pixel we adopt consists of seven entries, the R, G, and B values, the first partial derivatives I_x and I_y, and the second partial derivatives I_{xx} and I_{yy}. In the application of object detection, we can firstly compute the covariance descriptor of the object image. Then, for the target image, a ROI(Region of Interest) image which is a scanning window with identical size of the object image scans over the target image. A ROI image can be efficiently computed for its covariance descriptor by integral image. Thus, the distance between the covariance descriptors of the object image and the ROI image can be calculated with the eigenvalues, λ_i's. Taking two covariance descriptors C_1, C_2, denoted by $C_1 = [x_1, x_2, x_3, \ldots, x_{d(d+1)/2}]$, and $C_2 = [y_1, y_2, y_3, \ldots, y_{d(d+1)/2}]$ for example, the distance between them is defined by $\rho(C_1, C_2) = \sqrt{\sum_{i=1}^{d} \ln^2 \lambda_i(C_1, C_2)}$ [5]. If some of distances are smaller than an empirical threshold, then it can be determined that the corresponding ROIs match the object image, and hence, it achieves the objective of detection as shown in Fig.2.

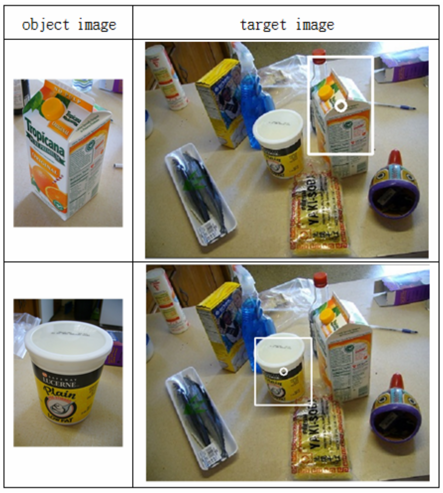

Fig. 2. Objects detection

4.2 Multiclass Object Classification

4.2.1 ETH-80 Dataset

In our experiment, ETH-80 [11] is the first dataset tested for multiclass object classification. The ETH-80 dataset contains 8 classes of images, and each class consists of 10 groups in which there are 41 images presented by varieties of viewing angles or colors as shown in Fig. 3. Firstly, each image is computed for its corresponding covariance descriptor. The training data consists of the first half of images for each participated group, and the second half is for testing data. Then we use libsvm to apply kernel SVM with three kernel functions, linear kernel, polynomial kernel, and RBF kernel, taking all covariance descriptors generated by training data as inputs to train classifiers. In the training part of RBF kernel, the training data and testing data needed to be normalized to get better precision, and the k-fold cross validation is used to compute the optimal parameters c and g for RBF kernel training. The results of classification accuracy show that RBF performs best as listed in Table 1.

Fig. 3. ETH-80 dataset

Table 1.

Nb. of classes	Linear	Polynomial	RBF
3	97.67%	96.67%	98.50%
4	87.63%	89.25%	94.13%
5	80.80%	84.90%	91.50%
6	77.33%	79.83%	85.33%
7	80.64%	82.29%	85.64%
8	82.44%	84.75%	87.00%

4.2.2 COIL-100 Dataset

In addition, we also test COIL-100 dataset [12] for experiment as shown in Fig. 4. The COIL-100 dataset contains 100 classes, and each class owns 72 images. To evaluate the impacts on accuracy causes by the partition of training and testing data, we adopt two different approaches. The first test is that for any class, the training image and the test image are selected by interleaving. The second test is selecting the first half for training data and the rest is for testing data.

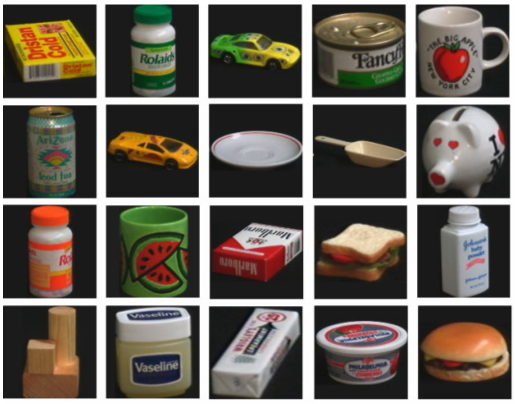

Fig. 4. COIL-100 dataset

The results are shown in Table 2 and Table 3, respectively. It can be seen that the accuracy is possible to be affected by the ways of data selections. Besides, the RBF kernel might not be always the best but indeed a feasible choice for kernel SVM because the dataset itself could be also a crucial effect to the classification result.

Table 2.

Nb. of classes	Linear	Polynomial	RBF
30	100%	99.90%	100%
50	100%	99.89%	99.94%
70	100%	99.84%	99.76%
80	100%	99.86%	99.76%
90	99.94%	99.85%	99.81%
100	99.94%	99.86%	99.78%

Table 3.

Nb. of classes	Linear	Polynomial	RBF
30	95.37%	94.72%	96.11%
50	93.28%	92.17%	93.56%
70	91.94%	90.63%	91.15%
80	92.08%	90.17%	90.45%
90	92.56%	90.56%	90.86%
100	92.61%	90.42%	89.81%

Recently, deep learning methods [13], such as convolutional neural networks [14], are widely developed and applied on machine learning. From this perspective, we also adopt Caffenet [15-16] to train and test both ETH-80 and COIL-100. The results show that the mean top-1 accuracies for 8 classes of ETH-80 and 100 classes of COIL-100 are 97.6% and 98.33% respectively, indicating that the deep convolutional neural network is indeed a novel and competitive approach.

5 Conclusion

This paper conducts the practicality of multiclass object classification using covariance descriptors with kernel SVM. The experimental results show that although the accuracy of classification can be varied when applying different kernel functions or different approaches for training and testing samples partition, it is indeed that for images in which the foreground objects are the major portions and the backgrounds are relatively simple, then the combination of covariance descriptor and kernel SVM is a feasible solution to such datasets for multiclass object classification. Due to the high computation efficiency of covariance descriptor and kernel SVM, the combination of them is also very practical to be applied on large scale datasets.

References

[1] O. Tuzel, F. Prikli, P. Meer, Region covariance: a fast descriptor for detection and classification, in: Proc. European Conference on Computer Vision (ECCV), 2006.

[2] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on riemannian manifold, in: Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008.

[3] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57(2)(2004) 137-154.

[4] F. Porikli, O. Tuzel, P. Meer, Covariance tracking using model update based on lie algebra, in: Proc. IEEE Computer Society Conference on Computer Vision and Patter Recognition(CVPR), 2006.

[5] W. Forstner, B. Moonen, A metric for covariance matrices, in: E. W. Grafarend, F. W. Krumm, V. S. Schwarze (Eds.), Geodesy-The Challenge of the 3rd Millennium, Spring, Berlin, Germany, 2003, pp. 299-309.

[6] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, 2004.

[7] C. Cortes, V. Vapnik, Support-vector networks. Machine Learning 20(1995) 273-297.

[8] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, [Technical report] Taipei, Taiwan: National Taiwan University, 2003.

[9] C.-H. Li, H.-H. Ho, C.-T. Lin, B.-C. Kuo, J.-S. Taur, An automatic method for selecting the parameter of the normalized kernel function to support vector machines, in: Proc. International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2010.

[10] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, M. Harandi,: Kernel methods on the Riemannian manifold of symmetric positive definite matrices, in: Proc. Computer Vision and Patter Recognition (CVPR), 2013.

[11] ETH-80 dataset. <http://www.mis.informatik.tu-darmstadt.de/Research/Projects/categorization/eth80-db.html>, 2014.

[12] COIL-100 dataset. <http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>, 2014.

[13] Y. LeCun, Y. Bengio, G Hinton, Deep learning, Nature 521(2015) 436-444.

[14] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, in: Proc. Advances in Neural Information Processing Systems (NIPS), 2012.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proc. the 22nd ACM International Conference on Multimedia, 2014.

[16] Caffe. <http://caffe.berkeleyvision.org/>, 2018.