A Multi-class Object Classifier Using Boosted Gaussian Mixture Model

Wono Lee and Minho Lee

School of Electronics Engineering, Kyungpook National University,
1370 Sankyuk-Dong, Puk-gu, Taegu 702-701, Korea
wolee@ee.knu.ac.kr, mholee@knu.ac.kr

Abstract. We propose a new object classification model, which is applied to a computer-vision-based traffic surveillance system. The main issue in this paper is to recognize various objects on a road such as vehicles, pedestrians and unknown backgrounds. In order to achieve robust classification performance against translation and scale variation of the objects, we propose new C1-like features which modify the conventional C1 features in the Hierarchical MAX model to get the computational efficiency. Also, we develop a new adaptively boosted Gaussian mixture model to build a classifier for multi-class objects recognition in real road environments. Experimental results show the excellence of the proposed model for multi-class object recognition and can be successfully used for constructing a traffic surveillance system.

Keywords: Object classification, Gaussian mixture model, Adaptive boosting, Traffic surveillance system.

1 Introduction

Recently, traffic surveillance technologies have been hot research topics for developing intelligent transportation systems. Among various technologies, camera-based systems such as CCTV monitor are most popular. In the field of researches for traffic surveillance systems with intelligent computer vision technologies, main issues are object detection, recognition and tracking. In this paper, we focus on object recognition problems in a traffic surveillance system.

In previous work, model-based object classification using wire-frame models for video surveillance is proposed [1]. However this approach has disadvantages that wire-frame models must be designed by an external modeling tool and the computational requirement grows linearly with the number of object models. In other study, a patch-based algorithm using a hierarchical feed-forward architecture shows high performance for object recognition [2, 3]. But the algorithm needs much computational load, which makes a difficulty to apply for real time systems. Also, in our previous work, the system using the biologically motivated feature extraction method and support vector machine (SVM) classifiers shows a good performance [4]. A 2-class SVM shows good performance on classification problems but it is not easy to apply for multi-class problems [5]. Depending on the number of class (n), many SVMs are required to classify all of the classes (n*(n+1)/2). In order to solve multi-class problems, we develop a Gaussian

K.W. Wong et al. (Eds.): ICONIP 2010, Part I, LNCS 6443, pp. 430–437, 2010.

mixture model (GMM) based classifier. A GMM is a probabilistic model for density estimation, and can be used to do both clustering and categorization [6]. After training a GMM, we can obtain a probability of an object class, and it is used to compare a feature characteristic of an input object with the Gaussian components in the GMM. Then, we can recognize an object if the maximum similarity of a GMM output is over a threshold for a specific trained object class. However, the conventional GMM based classifier just learns an object class with each GMM, it is difficult to generalize for different object classes and inefficient to build the GMs without considering the characteristics of the other object class. In order to overcome those limitations, we adopt the Adaptive boosting (Adaboost) training for constructing the GMMs [7, 8]. Each Gaussian component of GMMs is considered as a weak classifier that has low accuracy. The Adaboost combines the weak classifiers to builds a strong classifier to efficiently recognize the multi-class objects in real traffic environments.

This paper organized as follows; in section 2, we describe the feature extraction method and the proposed boosted GMM classifier. In section 3, experimental results will be shown. At last, summary and conclusion are discussed in section 4.

2 The Proposed Model

The proposed model consists of two main stages. The first is the feature extraction stage from object images. The feature extraction model extracts global and local features from the orientation MAX pooling which is based on the C1 features of the Hierarchical MAX (HMAX) model [9]. The HMAX model proposed by Riesenhuber and Poggio is based on the biological object perception mechanism of the visual cortex of a brain. The C1 feature which is one of the layers in the conventional HMAX model has robust characteristics to scale variation and translation. The second stage is object classification using the extracted features. The classification procedure is performed by probabilities calculation using a GMM. The Expectation-Maximization (EM) algorithm is well known method to update the parameters of a GMM [10, 11]. In this study, we use the greedy learning algorithm of GMM and partial EM searches to create a GMM [12]. After building the GMM for each object class, we can recognize multi-class objects using the probabilities of GMMs. In order to achieve a higher accuracy of classification, the Adaboost algorithm is applied to the GMMs.

2.1 Features Extraction

In the proposed model, we use the C1-like features based on the C1 features. However, we use edge orientation information based on Sobel operator instead of using Garbor filters, which can reduce the computation time. The orientation maps relevant to the S1 units are obtained by calculating of edge orientation as shown in Eq. (1).

$$\begin{gathered} G_x = HO_n * I, \quad G_y = VO_n * I \\ G = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(G_y / G_x\right) \end{gathered} \tag{1}$$

where G is the magnitude of gradient, θ is the direction of gradient, HO is the horizontal operator and VO is the vertical operator, I is intensity of an object image and n is the size of Sobel operator. While the S1 units have 8 bands, the orientation maps consist of 6 bands, thus 6 sizes of Sobel operators are used (band 1: 3x3, band 2: 5x5, band 3: 7x7, band 4: 9x9, band 5: 11x11, band 6: 13x13). For each band, by taking the maximum value over a 50% overlapped window with cells of different sizes for 2 adjacent bands (band 1 and 2: 8x8, band 3 and 4: 10x10, band 5 and 6: 12x12), the C1-like features consisting of 3 bands are obtained.

Using the similar way of our previous work [4], the proposed model extracts global and local features. Global features are obtained by the modified GIST algorithm [4], however, global features from 3 bands are not combined into 1 feature vector. They are separately used. Each band has 4 directions. Since each band means different scale of an object, this method is more robust to scale variation. Therefore, each global feature is 64 dimensions equally. Local feature is extracted by the same way of our previous work [4], but the parameters are modified to reduce the dimension of local feature. It is reduced from about 320 dimensions to about 70 dimensions because appropriate dimension is depending on training samples.

2.2 Gaussian Mixture Model

A GMM is a popular method for density estimation and clustering. It is defined as a weighted combination of Gaussian distributions. In d-dimensional space, probabilistic density of a Gaussian distribution is defined by Eq. (2).

$$\begin{aligned}&\phi(x;\theta)=\frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}}\exp\left(\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)\\&\mu=E[x],\;\;\Sigma=E[(x-\mu)^T(x-\mu)]\end{aligned}\tag{2}$$

where θ is a model with mean μ and covariance matrix Σ. A GMM with k Gaussian distributions is defined as:

$$f_k(x)=\sum_{i=1}^{k}\pi_i\phi(x;\theta_j)\;\;\mathit{where}\;\;\sum_{i=1}^{k}\pi_i=1\tag{3}$$

where π_i are the mixing weights and have non-negative values.

As a learning algorithm of a GMM, the EM algorithm is generally used. The EM algorithm is known to converge to a locally optimal solution. However, it does not guarantee a globally optimal solution. Therefore we use the greedy learning algorithm with the partial EM searches for an efficient learning [12]. The procedure of the greedy learning algorithm is described as follows:

For the training data $X=\{x_1,x_2,...,x_n\}$, the log-likelihood is defined as:

$$L(X,f_k)=\sum_{i=1}^{n}\log f_k(x_i)\tag{4}$$

1. Compute an optimal one-component GMM f_1 which maximize Eq. (4).
2. Find the optimal new component $\theta^*=\{\mu^*,\Sigma^*\}$ and α^* using Eq. (5)

$$\{\theta^*,\alpha^*\}=\arg\max_{\theta,\alpha}\sum_{i=1}^{n}\log[(1-\alpha)f_k(x_i)+\alpha\phi(x_i;\theta)] \tag{5}$$

3. Set $f_{k+1}(x)=(1-\alpha^*)f_k(x)+\alpha^*\phi(x;\theta^*)$

4. Update $\{\theta^*,\alpha^*\}$ using the partial EM searches in Eq. (6).

$$p(k+1\,|\,x_i)=\frac{\alpha\phi(x_i;\mu_{k+1},\Sigma_{k+1})}{(1-\alpha)f_k(x_i)+\alpha\phi(x_i;\mu_{k+1},\Sigma_{k+1})},\ \mu_{k+1}=\frac{\sum_{i\in A_j}p(k+1\,|\,x_i)x_i}{\sum_{i\in A_j}p(k+1\,|\,x_i)}$$
$$\Sigma_{k+1}=\frac{\sum_{i\in A_j}p(k+1\,|\,x_i)(x_i-\mu_{k+1})(x_i-\mu_{k+1})^T}{\sum_{i\in A_j}p(k+1\,|\,x_i)},\ \alpha=\frac{\sum_{i\in A_j}p(k+1\,|\,x_i)x_i}{n} \tag{6}$$

where A_j is a subset of X corresponding to a new component.

5. $k\leftarrow k+1$, according to the stopping criterion, stop or repeat from step 2.

2.3 Adaptive Boosting for Gaussian Mixture Models

After training of a GMM for an object, GMM can estimate the probabilities for the inputs. The higher probability means more probable to be involved with the learned class. Since the learning of each GMM is performed for each different class, the GMM by the specific class learning may have poor classification accuracy when the objects in different classes are very similar to each other. In order to solve this problem, we applied the Adaboost algorithm to efficiently collect each GMM for constructing a strong classifier [8]. As a weak learning algorithm for the Adaboost algorithm, we build a simple classifier $h_i(x)$ using a component of GMMs

$$h_i(x)=\begin{cases}1 & if\ \phi(x;\theta_i)\ge h_{th}\\ 0 & otherwise\end{cases} \tag{7}$$

where h_{th} is a threshold. The final classification result $H(x)$ is obtained by the weighted sum of weak classifier's results as shown in Eq. (8).

$$H(x)=\sum_{i=1}^{k}w_i h_i(x)\quad where\quad \sum_{i=1}^{k}w_i=1 \tag{8}$$

And the weights w_i is calculated by Eq. (9)

$$w_i=\log\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right) \tag{9}$$

where ε_i is the error rate of the weak classifier. After learning procedure, the weights are normalized to meet the condition of Eq. (8). The error rate is obtained by both the positive class trained by the GMM and the negative classes that are not trained by the GMM, thus weak classifiers which can mistake over classes will have lower weights.

2.4 Object Classification for a Traffic Surveillance System

The proposed model is developed for traffic surveillance systems. We build the 3-class object classifier recognizing vehicles, pedestrians and unknown backgrounds. The procedure of classification is shown in Fig. 1. Since the backgrounds do not have specific shapes or structures, we construct clusters of boosted GMM classifiers for 2 classes such as vehicles and pedestrians. Each cluster has 4 boosted GMM classifiers of 3 global feature bands and 1 local feature. Outputs of boosted GMM classifiers, $H_n(x)$, for each object class are summed and the classification result is obtained by the decision module as shown in Eq. (10).

$$\text{Result} = \begin{cases} \text{Vehicle} & \text{if } H\_V > H\_P \text{ and } H\_V > R_{th} \\ \text{Pedestrian} & \text{if } H\_V < H\_P \text{ and } H\_P > R_{th} \\ \text{Background} & \text{otherwise} \end{cases} \tag{10}$$

where R_{th} is a threshold for the decision module. *H_V* and *H_P* are the results from the clusters of boosted GMM classifiers for vehicles and pedestrians, respectively.

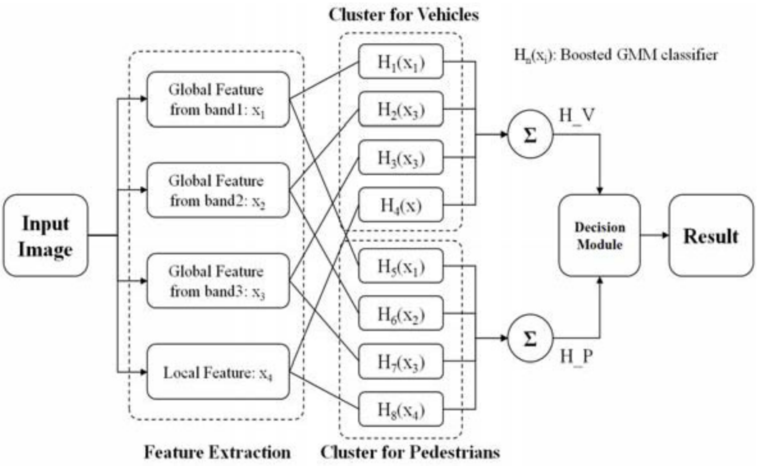

Fig. 1. The procedure of 3-class object classification

3 Experimental Results

In order to evaluate the proposed multi-class object classifier, we use 4 datasets. The dataset from 1 to 3 are shown in table 1 [13-16]. Background images of dataset 1 and 2 are collected in the part of background images of Caltech vehicle database. Dataset 3 is segmented object images from the ABR database that contains a real traffic environment [16]. Fig. 2 shows some of examples of 3 datasets. The train sets for dataset

1 and 2 consist of 900 images including 200 images for each class and 300 images for codebook generation. And the train set for dataset 3 consists of 1000 images including 300 images for vehicle class, 200 images for pedestrian class, 200 images for background and 300 images for codebook generation. For the experiments, we use the same number of training images for each class.

Table 1. The sources of the datasets for experiments

No.	Vehicles	Pedestrians	Background
Dataset 1	CBCL vehicle	CBCL pedestrian	Caltech DB
Dataset 2	Caltech vehicle	Daimler pedestrian	Caltech DB
Dataset 3	ABR DB	ABR DB	ABR DB

Table 2 shows the comparison of experimental results of the proposed model and our previous model [4] for the datasets. The proposed model has similar accuracy for vehicles and pedestrians, but higher performance for background images on average. Moreover, computational speed is much faster than the previous model [4], in which the proposed model takes 64msec on average, while the previous model takes 137 msec for the 78x78 size object images. Fig. 3 shows the averaged receiver operating characteristic (ROC) curve of the proposed model.

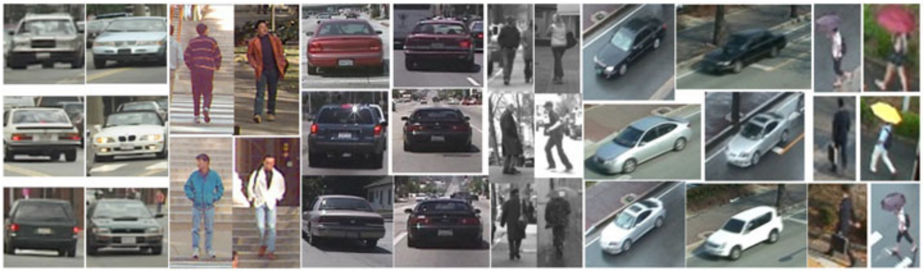
(a) (b) (c) (d) (e) (f)

Fig. 2. Examples of datasets for experiments. (a) CBCL vehicles, (b) CBCL pedestrians, (c) Caltech vehicles, (d) Daimler pedestrians, (e) ABR DB vehicles, (f) ABR DB pedestrians.

Table 2. The comparative experimental results of the proposed model and our previous model

		Vehicles	Pedestrians	Background
Dataset 1	Proposed Model	98.5%	99.5%	88%
	Previous Model	98.5%	99%	75.5%
Dataset 2	Proposed Model	97.5%	99%	74.5%
	Previous Model	100%	94.5	79.5%
Dataset 3	Proposed Model	96%	98.5%	72%
	Previous Model	99%	97%	62.5%

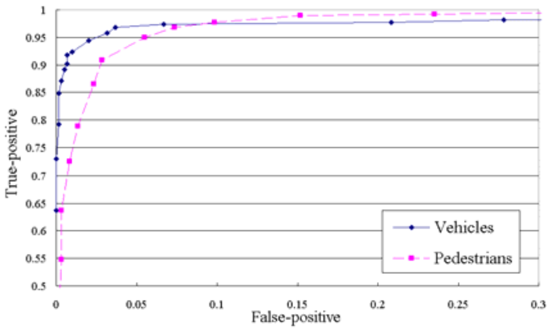

Fig. 3. Averaged ROC curve of the proposed model

Also, in order to compare the proposed model with the previous approach for multi-class object recognition, we conduct additional experiments. We compare the proposed model with the object recognition model by Zisserman et al [17]. For experiments, the dataset 4 consisting of 800 object images for each class (100 for generating the codebook, 350 for training and 350 for test) is used [14]. The proposed model shows higher performance than the Zisserman's model as shown in table 3.

Table 3. The comparative experimental results of the proposed model and the object recognition model by Zisserman et al.

	Motorbikes	Airplanes	Cars (Rear)
Proposed Model	97.7%	94.6%	96%
Zisserman's Model (unscaled)	93.3%	93%	90.3%

4 Conclusion

We proposed a new multi-class object classifier for traffic surveillance systems using boosted GMMs. Owing to the biologically motivated and efficiently modified feature extraction method, the proposed model has not only robustness of translation and scale variation of objects but also reliable computational speed. In our experiments, the classification algorithm based on GMMs with the Adaboost algorithm shows higher performance on multi-class recognition problems. Using the boosted GMM classifier, it is easy to implement a multi-class classifier.

As further works, we need to develop more efficient object detection and tracking algorithm to complete the traffic surveillance system. Also, we plan to develop an incremental object classifier which can learn objects incrementally.

Acknowledgments. This work was supported by the Industrial Strategic Technology Development Program funded by the Ministry of Knowledge Economy (MKE, Korea) (70%) and was also supported by the Converging Research Center Program through the Converging Research Headquarter for Human, Cognition and Environment funded by the Ministry of Education, Science and Technology (2010K001130) (30%).

References

1. Wijnhoven, R., De With, P.H.N.: 3D Wire-frame Object-modeling Experiments for Video Surveillance. In: Proc. of 27th Symp. Inform. Theory in the Benelux, pp. 101–108 (2006)
2. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust Object Recognition with Cortex-like Mechanisms. Trans. on Pattern Analysis and Machine Intelligence (PAMI) 29(3), 411–426 (2007)
3. Wijnhoven, R., De With, P.H.N.: Patch-based Experiments with Object Classification in Video Surveillance. In: Blanc-Talon, J., et al. (eds.) ACIVS 2007. LNCS, vol. 4678, pp. 285–296. Springer, Heidelberg (2007)
4. Woo, J.-W., Lim, Y.-C., Lee, M.: Obstacle Categorization Based on Hybridizing Global and Local Features. In: Chan, J.H. (ed.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 1–10. Springer, Heidelberg (2009)
5. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. J. Data Min. Knowl. Discov. 2, 121–167 (1998)
6. Scherrer, B.: Gaussian Mixture Model Classifiers (2007)
7. Schapire, R.E.: The Strength of Weak Learnability. Machine Learning 5, 197–227 (1990)
8. Freund, Y., Schapire, R.E.: A Short Introduction to Boosting. The Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
9. Riesenhuber, M., Poggio, T.: Hierarchical Models of Object Recognition in Cortex. J. Neurosci. 2, 1019–1025 (1999)
10. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. of the Royal Statistical Society 39(1), 1–38 (1977)
11. Blimes, J. A.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report ICSI-TR-97-021 (1997)
12. Verbeek, J., Vlassis, N., Krose, B.: Efficient Greedy Learning of Gaussian Mixture Models. Neural Computation 15, 469–485 (2003)
13. Center for Biological & Computational Learning, MIT, http://cbcl.mit.edu
14. Computational Vision Lab, Caltech, http://www.vision.caltech.edu
15. Daimler Pedestrian Detection Benchmark Data Set, http://www.gavrila.net
16. Artificial Brain Research Lab, Kyungpook National University, http://abr.knu.ac.kr/ABR_surveillance_DB.html
17. Fergus, R., Perona, P., Zisserman, A.: Object Class Recognition by Unsupervised Scale-Invariant Learning. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 264–271 (2003)