Real-time Object Classification in Video Surveillance Based on Appearance Learning

Lun Zhang, Stan Z. Li, Xiaotong Yuan and Shiming Xiang
Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation,Chinese Academy of Science
100080,Beijing,P.R.China
{lzhang,szli,xtyuan,smxiang}@nlpr.ia.ac.cn

Abstract

Classifying moving objects to semantically meaningful categories is important for automatic visual surveillance. However, this is a challenging problem due to the factors related to the limited object size, large intra-class variations of objects in a same class owing to different viewing angles and lighting, and real-time performance requirement in real-world applications. This paper describes an appearance-based method to achieve real-time and robust objects classification in diverse camera viewing angles. A new descriptor, i.e., the Multi-block Local Binary Pattern (MB-LBP), is proposed to capture the large-scale structures in object appearances. Based on MB-LBP features, an adaBoost algorithm is introduced to select a subset of discriminative features as well as construct the strong two-class classifier. To deal with the non-metric feature value of MB-LBP features, a multi-branch regression tree is developed as the weak classifiers of the boosting. Finally, the Error Correcting Output Code (ECOC) is introduced to achieve robust multi-class classification performance. Experimental results show that our approach can achieve real-time and robust object classification in diverse scenes.

1. Introduction

With the rapid development of video capture technology, video is becoming a cheap yet important media for information record. Understanding video objects is attracting extensive interest due to its greatly enhanced automation in public security surveillance. One important task in video surveillance is to classify moving objects into semantically meaningful categories. Typical applications include constructing intelligent parking systems for different vehicles and systems of object retrieval from videos, and so on. However, this recognition task is difficult, due to the following three aspects. First, the objects have diverse visual appearances and they may vary significantly due to different viewing angles and lighting. This may result in large intra-class variations. Second, the size of the moving objects may change with the distance to the camera. Third, the performance of video object recognition should be real-time so that the system has time to respond to the ongoing events in time. Thus, constructing such a real-time robust object recognition system is desired in real-world applications.

In recent years, much attention has been attracted on classifying object after motion segmentation. Moving objects can be separated from a static background reasonably by background subtraction, so the problem of clutter can be minimized. Most previous approaches in this area [13, 4, 10, 14] often use shape and motion information, such as area size, compactness, bounding box, speed, etc. However, object shapes in video may change drastically under different camera view angles. In addition, the detected shapes may be noised by shadow or other factors. Actually, shape-based approaches often require that the scene and camera view for test are very similar to those for training. Such assumptions are inadequate in real applications. Another important feature is based on object motion. They can be used to recognize humans and vehicles [14]. However, it is difficult to use motion to classify vehicles into more categories, such as car, truck, van, etc.

This paper focuses on applying appearance method to achieve real-time and robust objects classification in diverse camera viewing angles. Specifically, our goal is to classify the objects in the video into car, van, truck, person, bike and group of people by extracting distinct visual features and constructing appearance-based classifier.

Recently, there has been a great progress on appearance-based methods for object recognition [18, 19] from still images. One reason the recognition methods for still images have not been widely used in the past years in video is the small size of the objects or low resolution in video surveillance. Typically, for example, when monitoring the car-park

1-4244-1180-7/07/$25.00 ©2007 IEEE

or airport, the objects of interest are usually less than 80 pixels in height. In video analysis, edge-based rich representation with SIFT features [15] is used to recognize different types of vehicles in visual surveillance for an un-calibrated camera. However, this approach is designed for a single static scene, even restricts all the vehicles in a same pose.

In this paper, we propose a new feature for object representation and call it multi-block local binary pattern (MB-LBP). MB-LBP is extended from the original LBP feature [16], which has been proven to be a powerful appearance descriptor with computational simplicity. Besides, this feature is also successfully applied in many low resolution image analysis tasks [9]. However, it is limited to calculate the information in a small region and has no ability to capture large-scale structures of objects. To remedy this limit, MB-LBP is proposed to calculate the LBP values on large image windows (patches). The original LBP has also been extended in several ways. The most similar one to MB-LBP is the multi-scale LBP (MS-LBP) [17]. The distinct characteristic from MS-LBP is that MB-LBP is developed on image patches divided into sub-blocks (rectangles) with different sizes, while MS-LBP is still constructed on single pixel. This treatment provides a mechanism for us to capture appearance structures with various scales and aspect ratios. Intrinsically, MB-LBP is to measure the intensity differences between sub-blocks in image patches. Calculation on blocks is robust to noises, lighting changing. At the same time, MB-LBP can be computed very efficiently by using integral images [23].

Similar to the original LBP [16], the proposed MB-LBP feature is just a binary string sequence. For this non-metric feature value, multi-branch tree is designed as weak classifiers and Gentleboost [6] is used to select the features and construct the binary classifier. Finally, the Error Correcting Output Codes (ECOC) method [5] is used to reduce the multi-class problem to multiple binary classification problems. The good error correction property in ECOC [5] guarantees that even if some of the individual hypotheses were wrong, the example may still be right classified in some right classifier. In this way, we solve the multi-class problem as well as enhance the classification performance with MB-LBP features. Experimental results illustrate the validity of our method.

1.1. Related Work

There are two main classes of approaches used for localization and categorization of candidate objects. One approach is to directly detect object in single frames without prior segmentation. These methods often focus on detecting a specific object type in surveillance, such as pedestrians [24] or vehicles [11]. The other approach is to perform object classification on detected moving objects or tracked object sequences. Moving objects can be separated from a static background reasonably by background subtraction. Our system exemplifies the latter type.

One important step in all object classification methods is to extract suitable features for image data representation. Common features for classifying objects after motion-based detection include size, compactness, aspect ratio and simple descriptors of shape or motion. The main limitation of the systems based such features [13, 4, 10, 14] is that they often just work well in the restricted settings. Furthermore, these systems can just classify the objects into a few class, mostly only distinguish pedestrians from vehicles.

Some recent approaches have tried to address the issue of view independence. Bose and Grimson [2] describe a scene-invariant classification system that use the learning of scene context information for a new viewpoint. Brown [3] presented an object classification system for distinguishing humans from vehicles for an arbitrary scene. The limitation of above methods is that they just recognize humans from vehicles.

Many other features are also used. Researchers have investigated 3D based methods for classifying different types of vehicles [21]. These methods need camera calibration to reduce the parameters to be estimated and can hardly achieve real time performance due to high computation complexity. In [15], edge-based rich representation with SIFT features is used to classify all kinds of vehicles for an uncalibrated camera. However, this approache just design for a single static scene even restrict all the vehicles in a same pose. In [22], Tsuchiya and Fujiyoshi evaluate the relative importance of features such as shape, texture and motion by applying adaboost algorithm. The problems associated with objects view, scale and real time are not mentioned.

The original LBP, introduced by Ojala [16], is a powerful texture descriptor with computational simplicity. Later the operator was extended to consider different neighborhood sizes [17]. The LBP descriptor has been successfully used in many areas, such as face detection [7] and recognition [9], etc. Error correcting output code (ECOC) [5] is used as a general framework for handling multi-class problems by reducing the multi-class problem to a set of binary problems. ECOC classifier design concept has been used in many applications, such as text classification [8] and face verification [12].

1.2. Outline of Our method

Our method performs object classification on detected moving objects. A simple background subtraction based on on-line Gaussian Mixture Model (GMM) [20] is used to detect the moving objects. Different from using the detected binary shape information, we calculate their bounding boxes and select to calculate the corresponding image patches, i.e., the detected foreground patches. Each such

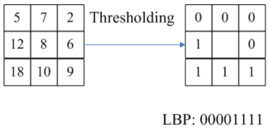


Figure 1. The basic LBP operator. This operator compares each neighborhood pixel value with the center pixel value.

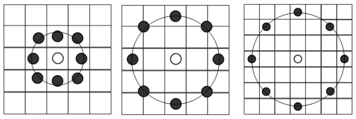


Figure 2. Three examples of the multi-scale LBP. (a) The circular (8,1) neighborhood. (b) The circular (8,2) neighborhood. (c) The circular (8,3) neighborhood. The pixel values are bi-linearly interpolated when the sampling point is not located at the point with integer coordinate.

patches is normalized to a unified scale (20×20 pixels) and converted to a gray-scale patch. In this way, we construct the training set. MB-LBP feature is then used to represent the objects' appearance features. By applying the AdaBoost learning algorithm [23], the most efficient MB-LBP features are finally selected, and a decision function is learned from training data. During training, the ECOC-based approach is used to divide the multi-class problem into several two-class classification problems via a predefined two-class task code matrix.

In the test phase, the foreground patches including the moving objects are first detected and normalized to the same unified scale as that for training. Then only the selected features via the AdaBoost learning method are calculated and supplied to the decision function to obtain a class score. Finally, we apply a simple voting method to the tracked sequence to get a final class score. The performance is real-time and can be able to recognize simultaneously multiple objects in the scene.

2. Multi-block Local Binary Pattern Representation

The original LBP operator is defined for each pixel, by thresholding the pixel values of its 3×3 neighborhood with the center pixel value and considering the results as a sequence of eight binary numbers. Fig. 1 shows the LBP operator. Such binary patterns can describe local structure of image, such as edges, lines, spots, flat areas and corners [16].

The most prominent limit of the LBP operator is its small spatial support area, hence the bit-wise comparison therein made between two single pixel values is much affected by noise. Moreover, features calculated in a local 3×3 neighborhood have no ability to capture the large scale structures which may be the dominant component for the visual appearances. Later, this operator was extended to different neighborhood sizes [17]. Fig. 2 shows a basic extension, i.e., multi-scale LBP (MS-LBP). Specifically, the size of the neighborhood can be changed to different scales and the number of the neighbors can be more than eight. When calculating the binary values in MS-LBP, however, only the selected pixel values are considered.

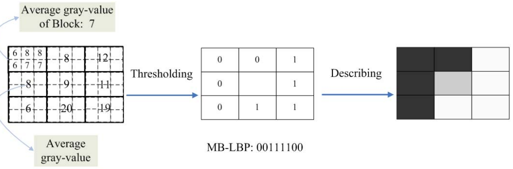


Figure 3. Multi-block LBP operator for image representation. As shown in the figure, this operator encodes intensities of the rectangular regions by local binary pattern. Compared with the original LBP operator calculated in a local 3×3 neighborhood, MB-LBP can capture image structure at large scales and aspect ratios.

Here we introduce a novel extension. The basic idea is to divide the image patch into sub-blocks (rectangles). The comparison operator between single pixels in original LBP is replaced by sub-blocks. We call this new feature Multi-block Local Binary Pattern (MB-LBP) feature. To encode the rectangles, the MB-LBP operator is defined by comparing the central rectangle's average intensity g_c with those neighboring rectangles $\{g_0, ..., g_7\}$, see Fig. 3. In this way, it can output a binary sequence. An output value of the MB-LBP operator can be obtained as follows:

$$MB - LBP = \sum\nolimits_{i=0}^{7} s(g_i - g_c)2^i \qquad (1)$$

where g_c is the average intensity of the center rectangle, g_i $i = 0, \cdots, 7$, are those of the eight neighboring rectangles,

$$s(x) = \begin{cases} 1, & if \quad x > 0 \\ 0, & if \quad x < 0 \end{cases}$$

Totally, we can get 256 ($= 2^8$) kinds of binary patterns. Fig. 4 shows some demos of MB-LBP patterns. In Fig. 4, each patches is further divided into 3×3 sub-blocks. The center sub-block is shown with light color. If the average gray value of the sub-block is greater than that of the center sub-block, it is shown with white color; otherwise, it is shown with dark color. We can see that such patterns can capture large scale structures.

Generally, the histograms of the Local Binary Patterns in local region are used as descriptors. For computational simplicity, in this paper we directly use the output value of MB-LBP operator at each pixel as image feature. For an image with 20×20, totally we can get 2049 local patches with

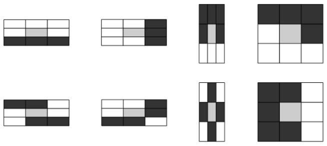


Figure 4. Some demos of the MB-LBP patterns.

different sizes and aspect ratios at different pixel locations. In this way, we get a MB-LBP feature vector with length of 2049. The features will be selected by an adaboost learning algorithm.

However, a problem is that the value of MB-LBP feature is nonmetric. The output of MB-LBP operator is just a symbol of a binary string. Each MB-LBP feature has totally 256 kinds of discrete values, since the number of all possible output value of MB-LBP operator is 256. In the next section, we design multi-branch tree as a weak classifier for each MB-LBP feature.

3. Feature Selection and Binary Classifier Learning

The feature set of MB-LBP feature is large and contains much redundant information. AdaBoost algorithm is used to select significant features and construct a binary classifier. Here, AdaBoost is adopted to solve the following three fundamental problems in one boosting procedure: (1) learning effective features from the large feature set, (2) constructing weak classifiers, each of which is based on one of the selected features, (3) boosting the weak classifiers into a stronger classifier.

3.1. Gentle AdaBoost for Binary Classification

We choose to use the gentle adaboost [6, 1] for the reason that it is simple to be implemented and numerically robust. Given a set of training examples as $\{(x_1, y_1), ..., (x_N, y_N)\}$, where $y_i \in \{+1, -1\}$ is the class label of the example $x_i \in R^n$. Boosting learning provides a sequential procedure to fit additive models of the form $F(x) = \sum_{m=1}^{M} f_m(x)$. Here $f_m(x)$ are often called weak learners, and $F(x)$ is called a strong learner. Gentle adaboost uses adaptive Newton steps for minimizing the cost function: $J = E[e^{-yF(x)}]$, which corresponds to minimizing a weighted squared error at each step.

In each step, the weak classifier $f_m(x)$ is chosen to minimize the weighted squared error:

$$J_{wse} = \sum_{i=1}^{N} w_i (y_i - f_m(x_i))^2 \tag{2}$$

3.2. Weak Classifiers

It is common to define the weak learners $f_m(x)$ to be the optimal threshold classification function [23], which is often called a stump. However, as indicated in Section 2, the value of MB-LBP feature is non-metric. Hence it is impossible to use threshold-based function as weak learner.

```
1. Start with weight $w_i = \frac{1}{N}, i = 1, 2, ..., N, F(x) = 0$
2. Repeat for t = 1, ,M
    (a) Fit the regression function by weighted least squares
        fitting of $Y$ to $X$.
    (b) Update $F(x) \leftarrow F(x) + f_m(x)$
    (c) Update $w_i \leftarrow w_i e^{-y_i f_m(x_i)}$ and normalization
3. Output the classifier $F(x) = sign[\sum_{m=1}^{M} f_m(x)]$
```

Table 1. *Algorithm of Gentle AdaBoost*

In this paper, for each MB-LBP feature, we adopt multi-branch tree as weak classifiers. Each branch of such weak classifier corresponds to a certain discrete value of MB-LBP feature. So the number of branches is equal to the number of all possible feature values. Since each MB-LBP feature has totally 256 possible feature values, the corresponding weak classifier has 256 branches. The weak classifier can be defined as:

$$f_m(x) = \begin{cases} a_0, & x^k = 0 \\ & ... \\ a_j, & x^k = j \\ & ... \\ a_{J-1}, & x^k = J-1 \end{cases} \tag{3}$$

Where x^k denotes the k-th element of the feature vector x, and a_j, J is the total number of branches, $j = 0, \cdots, J-1$, are regression parameters to be learned. These weak learners are often called decision or regression trees. We can find the best tree-based weak classifier (the parameter k, a_j with minimized weighted squared error as Equ.(2)) just as we would learn a node in a regression tree.The minimization of Equ.(2))gives the following parameters:

$$a_j = \frac{\sum_i w_i y_i \delta(x_i^k = j)}{\sum_i w_i \delta(x_i^k = j)} \tag{4}$$

As each weak learner depends on a single feature, one feature is selected at each step. In the test phase, given a MB-LBP feature, we can get the corresponding regression value fast by such multi-branch tree. This function is similar to the lookup table (LUT) weak classifier for Haar-like features [25], the difference is that the LUT classifier gives a partition of real-value domain.

4. Learning ECOC-Based Classifier for Robust Multi-class Object Classification

ECOC-based classifier is introduced to deal with the multi-class classification problem. Error correcting out-

Person	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Group of People	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Bike	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1
Car	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1
Van	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1
Truck	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
Bug	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Table 2. ECOC code matrix for object classification.

put code (ECOC) [5] is a general framework for handling multi-class problems by reducing the multi-class problem to a set of binary problems. Owing to the large intra-class variation, the binary boosting classifier with MB-LBP features has still relatively high error rate. The error correction mechanism in ECOC guarantees that even if some of the individual hypotheses were wrong, the example may still be classified in some right classifier. In this way, we can obtain more robustness multi-class classification performance.

4.1. Background Of Error Correcting Output Code

ECOC is used as a general framework for handling multi-class problems [5]. This approach reduces the multi-class problem to a set of two-class problems by a "coding matrix" M, where $M \in \{0,1\}^{k \times n}$, k is the number of classes (rows), and n is the length of the "code word" (columns).

Each class is assigned a "code word", defined by the corresponding row of "coding matrix" M. Each column of M induces a two-class learning problem. For the training task assigned by the j-th column of M, a sample belonging to the i-th class can be labeled as positive or negative one, depending on the value of M_{ij}. Here M_{ij} is the element of the i-th row and the j-th column of matrix M. In this way, each column can be viewed as a classification hypotheses. Totally, we can get n hypotheses $(h_1, ..., h_j, ..., h_n)$ from the coding matrix M. Finally, we can get n classifiers via adaboosting learning from the MB-LBP features.

In the testing phase, given a sample x, an n-bit binary vector $\mathbf{v}$ is produced by evaluating $h_1(x), ..., h_n(x)$. Then the label of x is predicted as that class whose code word is closet to $\mathbf{v}$ according to the Hamming distance:

$$H(x) = \arg\min_i \{\sum\nolimits_{j=1}^{n} |M_{ij} - v_j|\} \tag{5}$$

The error-correction property of ECOC method is that even if some of the individual hypotheses h_j were wrong, the example x can still be correctly classified in some class. This property can be explained as follows. If "coding matrix" is designed to ensure that the distance between every pair of code words is greater than d, then the closed Hamming balls with radius $\left[\frac{d-1}{2}\right]$ around each code word are disjoint. As long as the Hamming distance between the $\mathbf{v}$ of a sample x and the i-th row of M is less than $\left[\frac{d-1}{2}\right]$, then the i-th row is the unique closest one to $\mathbf{v}$. This indicates that ECOC approach has the ability to error-correction to some degree.

4.2. Object Classification via ECOC-based Method

Our goal is to classify the moving objects into six classes, including person, bike, group of people, car, van and truck. Since many factors (e.g. abrupt changes in lighting, shaking trees and fluttering flags) may affect the results of background subtraction, large numbers of detected regions may correspond to spurious objects, for example, the meaningless clutters. Simple treatment as setting minimum duration of a track can partially filter the spurious objects. To further reject the remaining spurious objects, we integrate rejecting spurious objects to our foreground classifier by treating them as a type of object class, which is called "bug". And therefore our foreground classifier based on ECOC actually faces a seven-class classification problem.

We construct a complete code matrix with $n = 31$ via the exhaustive code designing method [5], as shown in Table. 2. According to information theory, this code has error correcting ability for any base classifier. Experimental results (In section 5.2.3) show the multi-class classifier based on this coding matrix can effectively improve the robustness, compared with the simple One-Vs-Others classifier due to its inherent error-correction ability.

5. Experiment

5.1. Data Set

We collected videos from some outdoor overlooking cameras which locate in different places. Multi-view objects can be obtained by changing camera view angles.

5.1.1 Foreground Samples Collection

We firstly collect foreground samples to train the multi-class classifier. We implemented real-time background subtraction and tracking as [20, 26], so that moving objects can be reasonably separated from background. For each moving object, the detected window was fixed to a square according to its mass center. Foreground image samples are

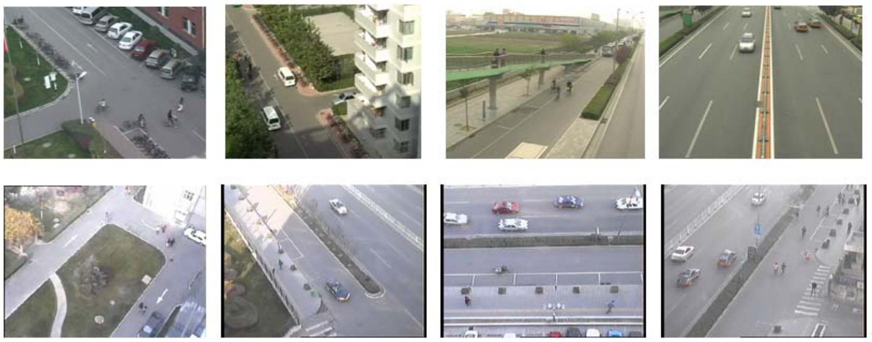


Figure 5. The diverse scenes used for testing. The objects' appearances in these scenes vary significantly mostly due to camera view.

Figure 6. Some examples of training samples. The training set contains van, car, truck, bike, person, group of people and spurious object, which are collected in diverse camera viewing angles.

then obtained by normalizing such window to 20×20. We collected samples per 10 frames in order to reduce the correlation between objects. As discussed in Section 4.2, we first filter spurious objects by setting minimum duration of a track. The remaining objects are manually labeled to person, bike, group of people, car, van, truck and bug. Our collected sample set consists of 55,458 cars, 7,032 vans, 5,324 trucks, 8,070 persons, 14,076 bikes, 16,116 groups of people and 7,108 bug samples. Some of them are shown in Fig. 6.

5.1.2 Collection of Testing Data

To test the performance of the whole system, we collect 432 tracked object sequences from 8 different scenes shown in Fig.5. The objects in these test sequences are all not included in the training set.

5.2. Experimental Result

5.2.1 Performance of whole Method

We trained our object classifier by the collected training samples and apply it to the test set. As discussed in Section 1.2, we first classify the detected foreground image patches in each frame via the learned decision function; then combine the individual class labels to produce a sequence of class labels. Table 3 shows the classification results. This results suggest that our approach achieve considerable performance in diverse scenes. Furthermore, the processing time of our classification method for a 320x240 image resolution is less than 0.1s/frame on a P4 3.0GHz PC.

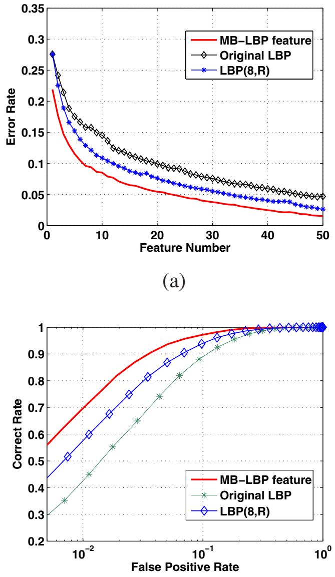
	Tracks	Correct Tracks	Correct Rate
Cars	208	179	86.1%
Vans	39	33	84.6%
Trucks	19	14	73.6%
Persons	71	63	88.72%
Bikes	55	45	82%
People Groups	40	33	82.5%

Table 3. Experimental result on test set.

5.2.2 Evaluation the distinctive of MB-LBP features

We evaluated the performance of MB-LBP features by comparing with original LBP features and extended LBP features. According to Ojala's work [17], with a circular neighborhood P at radius R, LBP can be represent as $LBP_{P,R}$. Thus, the original LBP can be represent as $LBP_{8,1}$, and the MS-LBP can be constructed with any radius and number of pixels in the neighborhood.

Because our multi-class classifier is composed of multiple binary classifiers, the comparison can be illustrated by a binary classification problem which distinguish humans



(a)

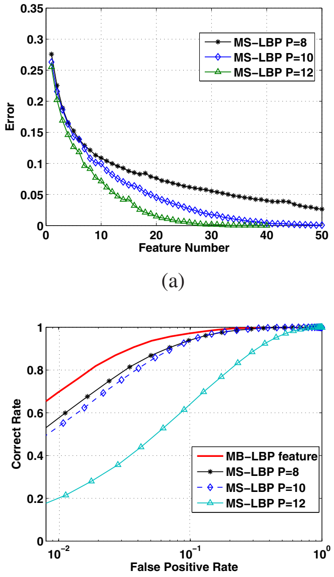
(b)

Figure 7. Comparative results with MB-LBP features, original LBP features and multi-scale LBP features. (a) The curves show the error rate as a function of the selected features in training process. (b) The ROC curves show the classification performance of the three classifiers on the test set.

(including person, group of people) from vehicles (including bike, car, van and truck) and bug samples. The training set contains 19,131 positive samples (humans) and 37,461 negative samples (others), and the test set contains 19,131 positive samples and 37,461 negative samples.

We first compare MB-LBP with the original LBP and MS-LBP $LBP_{8,R}$. Here R can be equal to all possible values. That is, R belongs to $\{1, 2, ..., 9\}$ in our experiments since the image patches are all normalized to 20×20 pixels. Based on Adaboost learning framework, three boosting classifiers are trained with 50 original LBP features, $LBP_{8,R}$ and MB-LBP features, respectively. Then they are evaluated on the test set. Fig. 7(a) shows the curves of the error rate (average of false alarm rate and false rejection rate), with the increment of the number of the selected features. We can see that the curve corresponding to MB-LBP features has the lowest error rate. This indicates that the weak classifiers based on MB-LBP features are more discriminative. The ROC curves of the three classifiers on the test set can be found in Fig. 7(b). It is shown that in the given false alarm rate at 0.01, classifier based on MB-LBP features shows 12% higher correct rate than $LBP_{8,R}$ and 28% higher than original LBP feature.

In our algorithm, the weak classifier corresponding to each LBP feature is constructed by multi-branch tree. Each branch of such weak classifier corresponds to a certain discrete value of LBP. So the total number of branches is 2^P.



(a)

(b)

Figure 8. Comparative results of MS-LBP features with different P. Although larger P can get lower error rate on the training data, the ROC performance on the test set is not high. (a) The curves show the error rate as a function of the selected features in training process. (b) The ROC curves show the classification performance of the three classifiers on the test set.

This number will significantly increase with the increase of P. When P is 10, the total number of branches is 1046. Actually, our experiments with larger P show low performance. On the one hand, the over-fitting problem occurs (See Fig.8). On the other hand, a large number of branches also increases the computation and memory burden.

5.2.3 Evaluation the advantage of Error correction property of ECOC-based classifier

There are many different approaches for reducing multi-class problem to multiple two-class classification problems. The most straightforward way is One-Vs-All. This method considers the comparison between each class from all the others. An example is classified in the class whose corresponding classifier has the highest output. This classifier decision function is defined as:

$$f(x) = \arg \max_{j \in \{1,...,K\}} f_j(x)$$

This method directly uses the output of single classifiers. We implemented a One-Vs-All classifier and compared its performance on the test set with ECOC-based classifier.

In this experiment, we randomly divide the collected foreground samples (described in section 5.1.1) to two equally parts, one for training the other for testing. Table 4 shows the correct rate of classification comparison between One-Vs-All and ECOC. It is shown that the error rates decreased after combining all the binary classifiers via error correcting code. It is illustrated that the advantage of the error-correction property of ECOC method improves the classification performance.

	Samples	One-vs-all	ECOC-based
Cars	27729	86.2%	92.6%
Vans	3516	67.6%	76.0%
Trucks	2662	66.1%	71.1%
Persons	4035	81.2%	85.2%
Bikes	7038	72.6%	78.8%
People Groups	8058	71.6%	75.8%
Bug	3554	52%	60.4%

Table 4. Classification correct rates comparison between ECOC and One-Vs-All.

6. Conclusions

In this paper, we have described a moving object classification algorithm based on appearance learning. Multi-block local binary pattern (MB-LBP) feature is proposed as foreground image descriptor. The basic idea of MB-LBP is to encode the neighboring rectangular regions by LBP operator. Compared with the original LBP, MB-LBP can capture image structures with different scales and aspect ratios. Experimental results show that MB-LBP feature is more distinctive than original LBP.

Aiming at dealing with the non-metric feature value of MB-LBP features, multi-branch regression tree is developed to construct the weak classifiers when using AdaBoost algorithm to select the discriminant features and construct the strong two-class classifier. Finally, to solve the multiclass problem, the ECOC-based method is introduced to reduce the multi-class problem to a group of two-class classification problems. Recognition experiments indicate that our method is validate.

7. Acknowledgements

This work was supported by the following funds: Chinese National Natural Science Foundation Project #60518002, Chinese National 863 Program Projects #2006AA01Z192 and #2006AA01Z193, Chinese National Science and Technology Supporting Platform Project #2006BAK08B06, and Chinese Academy of Sciences 100 people project, and AuthenMetric Co.Ltd.

References

[1] A.Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

[2] B.Bose and E. Grimson. Improving object classification in far-field video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[3] L. M. Brown. View independent vehicle/ person classification. In *the ACM 2nd international workshop on Video Surveillance and Sensor Networks*, New York,USA, October 2004.

[4] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[5] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Artificial Intelligence Research*, 1995.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 2000.

[7] B. Froba and A. Ernst. Face detection with the modified census transform. In *AFGR*, 2004.

[8] R. Ghani. Using error-correcting codes for text classification. In *ICML*, 2000.

[9] A. Hadid, M. Pietikainen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. In *CVPR*, 2004.

[10] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *European Conf. on Computer Vision*, 2002.

[11] Z. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *ICCV*, 2003.

[12] J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas. Face verification via error correcting output codes. *Image and Vision Computing*, 2003.

[13] A. Lipton, H. Fujiyoshi, and R. Patil. Moving target classification and tracking from real-time video. In *IEEE Workshop on Applications of Computer Vision*, 1998.

[14] A. J. Lipton. Local application of optic flow to analyze rigid versus nonrigid motion. In *Computer Vision Workshop Frame-Rate Vision*, 1999.

[15] X. Ma and W. E. L. Grimson. Edge-based rich representation for vehicle classification. In *IEEE Conference of Computer Vision*, 2005.

[16] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 1996.

[17] T. Ojala, M. Pietikainen, and M. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification width local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[18] R.Fergus, P.Perona, and A.Zisserman. Object class recogniton by unsupervised scale-invariant learning. In *CVPR*, 2003.

[19] S.Belongie, J.Malik, and J.Puzicha. Shape matching and object recognition. *IEEE Trans. PAMI*, 2002.

[20] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, 1999.

[21] T. N. Tan, G. D. Sullivan, and K. D. Baker. Model-based localization and recognition of road vehicles. *IJCV*, 1998.

[22] M. Tsuchiya and H.Fujiyoshi. Evaluating feature importance for object classification in visual surveillance. In *18th International Conference on Patten Recognition*, 2006.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[24] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *International Conference on Computer Vision*, 2003.

[25] B. Wu, H.Ai, and C.Huang. Fast rotation invariant multiview face detection based on real adaboost. In *AFG'04*, 2004.

[26] T. Yang, Q. Pan, J. Li, and S. Li. Real-time multiple objects tracking with occlusion handling in dynamic scenes. In *IEEE Computer Vision and Pattern Recognition*, 2005.