Published in IET Computer Vision
Received on 28th April 2008
Revised on 8th September 2008
doi: 10.1049/iet-cvi:20080026

Special Issue - selected papers from DICTA 2007

ISSN 1751-9632

Performance evaluation of local features in human classification and detection

S. Paisitkriangkrai[1,2] *C. Shen*[1,3] *J. Zhang*[1,2]

[1]*National ICT Australia (NICTA), Neville Roach Laboratory, Kensington, NSW 2052, Australia*
[2]*School of Computer Science and Engineering, University of New South Wales, Kensington, Sydney, NSW 2033, Australia*
[3]*RSISE, Australian National University, Canberra, ACT 0200, Australia*
E-mail: paul.pais@nicta.com.au

Abstract: Detecting pedestrians accurately is the first fundamental step for many computer vision applications such as video surveillance, smart vehicles, intersection traffic analysis and so on. The authors present an experimental study on pedestrian detection using state-of-the-art local feature extraction and support vector machine (SVM) classifiers. The performance of pedestrian detection using region covariance, histogram of oriented gradients (HOG) and local receptive fields (LRF) feature descriptors is experimentally evaluated. The experiments are performed on the DaimlerChrysler benchmarking data set, the MIT CBCL data set and 'Intitut National de Recherche en Informatique et Automatique (INRIA) data set. All can be publicly accessed. The experimental results show that region covariance features with radial basis function kernel SVM and HOG features with quadratic kernel SVM outperform the combination of LRF features with quadratic kernel SVM. Furthermore, the results reveal that both covariance and HOG features perform very well in the context of pedestrian detection.

1 Introduction

Detecting pedestrians has attracted a lot of research interests in recent years, because of its key role in several important applications in computer vision, for example, smart vehicles, surveillance systems with intelligent query capabilities, intersection traffic analysis. In particular, there has been a growing effort in the development of intelligent video surveillance systems. Public places like airports, train stations and parking areas have a large number of security cameras recording at all times. Because of the vast amount of video data being processed, it is very difficult to detect and respond to an abnormal event in real-time. An example of such abnormal events is unusual human activity in a scene. An automated method for finding humans in a scene serves as the first important preprocessing step in understanding human activity. Human detection, however, is considered among the hardest examples of object detection problems. The challenges include a wide range of poses that humans adopt, large variations in clothing, as well as cluttered backgrounds and environmental conditions. All these issues have made this problem very challenging from a machine vision perspective.

Pattern classification approaches have been shown to achieve successful results in many areas of object detections. These approaches can be decomposed into two key components: feature extraction and classifier construction. In feature extraction, dominant features are extracted from a large number of training samples. These features are then used to train a classifier. During testing, the trained classifier scanned the entire input image to look for particular object patterns. This general approach has shown to work very well in the detection of many different objects, for example, face [1] and car number plate [2], etc.

The performance of several pedestrian detection approaches has been evaluated in [3]. Multiple feature-classifier combinations have been examined with respect to their receiver-operating characteristic (ROC) performance and efficiency. Different features including principal component analysis (PCA) coefficients, local receptive fields (LRF) feature [4] and Haar wavelets [5] are used to train neural networks, support vector machines (SVM) [6, 7] and k-NN classifiers. The authors conclude that a combination of SVM with LRF features performs best. Their results show that local feature-based detectors significantly outperform those using global features [3]. This is because of the large variability of pedestrian shapes. Global features like PCA are more powerful modelling objects with stable structures such as frontal faces, rigid car images taken from a fixed view angle.

Although [3] provides some insights on pedestrian detection, it has not compared the state-of-the-art techniques because of the fast progress in this topic. Recently, histogram of oriented gradients (HOG) [8] and region covariance features [9] are preferred for pedestrian detection. It has been shown that they outperform those previous approaches. HOG is a gray-level image feature formed by a set of normalised gradient histograms; while region covariance is an appearance-based feature, which combines pixel coordinates, intensity, gradients, etc., into a covariance matrix. Hence, the type of features employed for detection ranges from purely silhouette-based (e.g. HOG) to appearance based (e.g. region covariance feature) features. To our knowledge, these approaches have not yet been compared. It remains unclear whether silhouette or appearance-based features are better for pedestrian detection. This paper tries to answer this question. The main purpose of the paper therefore is a systematic comparison of some novel techniques for pedestrian detection.

In this paper, we perform an experimental study on the state-of-the-art pedestrian detection techniques: LRF, HOG and region covariance along with various combinations of SVM. The reasons we select these three features along with SVM classifiers are mainly:

- These three local features seem to be the best candidates for this task.

- SVM is one of the advanced classifiers. It is easy to train and, unlike neural networks, the global optimum is guaranteed. Thus the variance caused by suboptimal training is avoided for fair comparison.

The paper is organised as follows. Section 2 reviews various existing techniques for pedestrian detection. Sections 3 and 4 describe methods used for feature extraction and a brief introduction to two of the well-known classifiers. The experimental setup and experimental results are presented in Section 5. The paper concludes in Section 6.

2 Related work

Many pedestrian classification approaches have been proposed in the literature. These algorithms can be roughly classified into two main categories: (1) approaches which require preprocessing techniques like background subtraction or image segmentation (e.g. [10] segments an image into so-called super pixels and then detects the human body and estimates its pose) and (2) approaches which detects pedestrian directly without using pre-processing techniques [4, 5, 8, 9].

Background subtraction and image-segmentation techniques can be applied to segment foreground objects from the background. The foreground objects can then be classified into different categories like human, vehicle and animal, based on their shape, colour, texture, etc. Some of the main drawbacks of these techniques are that they usually assume that the camera is static, background is fixed and the differences are caused only by foreground objects. In addition, the performance of the system is often affected by outdoor light changes.

The second approach is to detect humans based on features extracted from the image. Features can be distinguished into global features, local features and key-points depending on how the features are measured. The difference between global and local features is that global features operate on the entire image of data sets, whereas local features operate on the subset regions of the images. One of the well-known global feature extraction method is PCA. The drawback of global features is that the approach fails to extract meaningful features if there is a large variation in object's appearance, pose and illumination conditions. On the other hand, local features are much less-sensitive to these problems since the features are extracted from the subset regions of the image. Some examples of the commonly used local features are wavelet coefficient [1], gradient orientation [8], region covariance [9], etc. Local feature approaches can be further divided into whole body detection and body parts detection [11, 12]. In part-based approach, individual results are combined by a second classifier to form whole body detection. The advantage of using part-based approach is that it can deal with variation in human appearance owing to body articulation. However, this approach adds more complexity to the pedestrian detection problem. As pointed out in [3], the classification performance reported in different literature is quite different. This is because of data sets' composition with respect to negative samples. Data sets with negative samples containing large uniform image regions typically lead to much better classification performance.

3 Feature extraction

Feature extraction is the first fundamental step in most object detection and pattern recognition algorithms. The performance of most computer vision algorithms often relies on the extracted features. The ideal feature would be the one that can differentiate objects in the same category from objects in different categories. Commonly used low-level features in computer vision are colour, texture and shape. In this paper, we evaluate three local features, namely LRF, HOG and region covariance. LRF features are extracted using multilayer perceptrons by means of their hidden layer. The features are tuned to the data during training. HOG uses histogram to describe oriented gradient information. Region covariance computes covariance from several low-level image features such as image intensities and gradients.

3.1 Local receptive fields

The features are generated from a parallel hierarchical artificial neural network similar to the one shown in Fig. 1. Neural network is a mathematical model based on biological neural networks. It is an adaptive system that changes its structure based on the information that flows through the network during training process. One of the well-known feedforward neural network system is multilayer perceptron. Multilayer perceptrons consist of three or more layers of neurons with nonlinear activation function. The three layers consist of input layer (input images), hidden layer and output layer. Each layer extracts informative features from the output of the preceding stage to form a more compact representation of the extracted features. A neuron of a higher layer does not receive input from all neurons of the underlying layer but only from a limited region of it, which is call local receptive fields (LRF). The hidden layer is divided into a number of branches with all neurons within one branch sharing the same set of weights. Each branch captures the spatial information by encoding local image features.

In [3], the authors further investigate the concept of LRF. In their experiments, they have shown that receptive fields of size 5 × 5 pixels, shifted at a step size of two pixels over the input image of size 18 × 36 are optimal. In order to further improve the performance of LRF, the authors combine SVM with the output of the hidden layer of a neural network/LRF. In other words, multilayer perceptrons extract compact and meaningful features from input images. The extracted features can then be used to train SVM classifiers.

3.2 Histograms of oriented gradients

Since the advent of scale invariant feature transformation (SIFT) [13], which uses normalised local spatial histograms as a descriptor, many research groups have been studying the use of orientation histograms in other areas. Dalal and Triggs [8] show one of the successful examples. [8] propose histogram of oriented gradients in the context of human detection. Their method uses a dense grid of histogram of oriented gradients, computed over blocks of various sizes. Each block consists of a number of cells. These blocks can overlap with each other. For each pixel, $\boldsymbol{I}(x, y)$, the gradient magnitude, $m(x, y)$ and orientation, $\theta(x, y)$ is computed from

$$dx = \boldsymbol{I}(x+1, y) - \boldsymbol{I}(x-1, y) \tag{1}$$

$$dy = \boldsymbol{I}(x, y+1) - \boldsymbol{I}(x, y-1) \tag{2}$$

$$m(x, y) = \sqrt{dx^2 + dy^2} \tag{3}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{dy}{dx}\right) \tag{4}$$

A local one-dimensional orientation histogram of gradients is formed from the gradient orientations of sample points within a region. Each histogram divides the gradient angle range into a predefined number of bins. The gradient magnitudes vote into the orientation histogram. In [8], each detection window is divided into cells of size 8 × 8 pixels and a group of 2 × 2 cells is integrated into a block. Block can overlap with each other. The orientation histogram of each cell contains nine bins covering an orientation range of $0^\circ - 180^\circ$ (unsigned gradients–a gradient vector and its negative vote into the same bin). Each block contains a concatenated vector of all its cells. In other words, each block is represented by a 36-D feature vector (9 bins/cell × 4 cells/block) (Fig. 2).

Each of the HOG descriptor blocks is then normalised based on the energy of the histogram contained within it. Normalisation introduces better invariance to illumination, shadowing and edge contrast. In order to reduce the effect of nonlinear illumination changes owing to camera saturation or environmental illumination changes that affect three-dimensional(3D) surfaces, ℓ_2-norm is applied followed by clipping (limiting the maximum values of the gradient magnitudes to 0.2) and renormalising. The value of 0.2 is determined experimentally using images containing

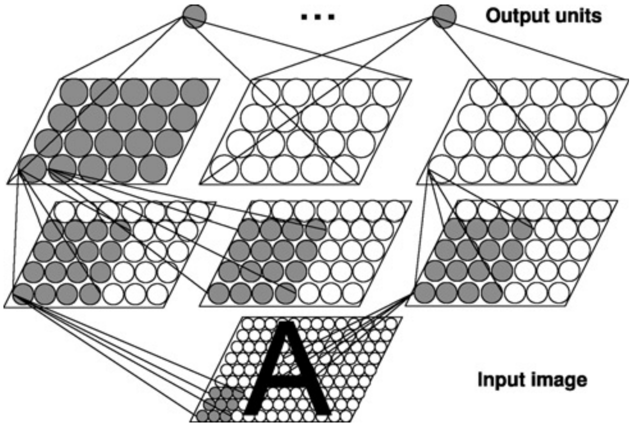

Figure 1 *Multilayer perceptrons with local receptive fields*

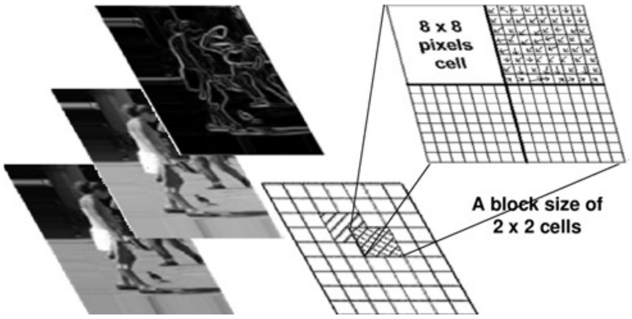

Figure 2 *HOG features*

Each block consists of a grid of spatial cells. For each cell, the weighted vote of image gradients in orientation histograms is computed

doi: 10.1049/iet-cvi:20080026

different illuminations for the same 3D objects [13]. The final step is to combine these normalised block descriptors to form a feature vector. The feature vector can then be used to train SVM classifiers.

3.3 Region covariance

Tuzel *et al.* [9, 14] have proposed region covariance in the context of object detection. Instead of using joint histograms of the image statistics (b^d dimensions where d is the number of image statistics and b the number of histogram bins used for each image statistics), covariance is computed from several image statistics inside a region of interest (dimensions). This results in a much smaller dimensionality. Similar to HOG, the image is divided into small overlapped regions. For each region, the correlation coefficient is calculated. The correlation coefficient of two random variables X and Y is given by

$$\rho_{X,Y} = \frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)\mathrm{var}(Y)} = \frac{\mathrm{cov}(X, Y)}{\sigma_x^2 \sigma_y^2} \tag{5}$$

$$\begin{aligned}\mathrm{cov}(X, Y) &= \boldsymbol{E}\big[(X - \mu_X)(Y - \mu_Y)\big] \\ &= \frac{1}{n-1}\sum_k (X_k - \mu_X)(Y_k - \mu_Y)\end{aligned} \tag{6}$$

where $\mathrm{cov}(\cdot, \cdot)$ is the covariance of two random variables, μ the sample mean and σ^2 the sample variance. Correlation coefficient is commonly used to describe the information we gain about one random variable by observing another random variable.

A positive correlation coefficient, $\rho_{X,Y} > 0$, suggests that when X is high relative to its expected value, Y also tends to be high and vice versa. A negative correlation coefficient, $\theta_{X,Y} < 0$, suggests that a high value of X is likely to be accompanied by a low value of Y and vice versa. A linear relationship between X and Y produces the extreme values, $\theta_{X,Y} = \{+1, -1\}$. In other words, correlation coefficient is bounded by -1 and 1.

Image statistics used in this experiment are similar to the one used in [9]. The eight-dimensional feature image used are pixel location x, pixel location y, first-order partial derivative of the intensity in horizontal direction $|\boldsymbol{I}_x|$, first-order partial derivative of the intensity in vertical direction $|\boldsymbol{I}_y|$, the magnitude $\sqrt{\boldsymbol{I}_x^2 + \boldsymbol{I}_y^2}$, edge orientation $\tan^{-1}(|\boldsymbol{I}_y|/|\boldsymbol{I}_x|)$, second-order partial derivative of the intensity in horizontal direction $|\boldsymbol{I}_{xx}|$, second-order partial derivative of the intensity in vertical direction $|\boldsymbol{I}_{yy}|$. The covariance descriptor of a region is an 8×8 matrix. Because of the symmetry, only the upper triangular part is stacked as a vector and used as covariance descriptors. The descriptors encode information of the correlations of the defined features inside the region. Note that this treatment is different from [9, 14], where the covariance matrix is directly used as the feature and the distance between features is calculated in the Riemannian manifold covariance matrices are symmetric and positive semi-definite, hence they reside in the Riemannian manifold. However, eigen-decomposition is involved in calculating the distance in the Riemannian manifold. We instead vectorise the symmetric matrix and measure the distance in the Euclidean space, which is faster.

In order to improve the covariance matrices' calculation time, technique which employs integral image [1] can be applied [14]. By expanding the mean from the previous equation, covariance equation can be written as

$$\mathrm{cov}(X, Y) = \frac{1}{n-1}\left[\sum_k X_k Y_k - \frac{1}{n}\sum_k X_k \sum_k Y_k\right] \tag{7}$$

Hence, to find the fast covariance in a given rectangular region, the sum of each feature dimension, for example, $\sum_k X_k$, $\sum_k Y_k$ and the sum of the multiplication of any two feature dimensions, for example, $\sum_k X_k Y_k$ can be computed using integral image.

The final step is to concatenate these covariance descriptors from all regions into a combined feature vector, which can then be used to train SVM classifiers.

4 Classifiers

There exist several classification techniques that can be applied to object detection problem. Some of the commonly applied classification techniques are SVM [6, 7] and AdaBoost [1, 15]. owing to space constraints, we limit our explanation of SVM and AdaBoost classifiers algorithm to an overview.

4.1 Support vector machines

Large margin classifiers have demonstrated their advantages in many vision tasks. SVM is one of the popular large margin classifiers [6, 7] which has a very promising generalisation capability. Linear SVM is best understood and simplest to apply. However, linear separability is a rather strict condition. Kernels are combined into margins for relaxing this restriction. SVM is extended to deal with linearly non-separable problems by mapping the training data from the input space into a high-dimensional, possibly infinite-dimensional, feature space. Using the kernel trick, the mapping function is not necessarily known explicitly. Like other kernel methods, SVM constructs a symmetric and positive definite kernel matrix (Gram matrix) which represents the similarities between all training datum points. Given N training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, the kernel matrix is written: $\boldsymbol{K}_{ij} \equiv \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)\rangle$, $i, j = 1, \ldots, N$. When $\boldsymbol{K}_{ij}$ is large, the labels of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, are expected to be the same. Here, $\boldsymbol{y}_i, \boldsymbol{y}_j \in \{+1, -1\}$. The decision rule is given by $\mathbf{sign}(f(\boldsymbol{x}))$ with

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_S} \hat{\beta}_i K(\hat{\boldsymbol{x}}_i, \boldsymbol{x}) + b \tag{8}$$

where $\hat{\boldsymbol{x}}_i$, $i = 1, \ldots, N_S$, are support vectors, N_S is the number of support vectors, $\hat{\beta}_i$ the weight associated with $\hat{\boldsymbol{x}}_i$ and b is the bias. The training process of SVM then determines the parameters $\{\hat{\boldsymbol{x}}_i, \hat{\beta}_i, b, N_S\}$ by solving the optimisation problem

$$\begin{aligned} &\underset{\xi, \boldsymbol{w}, b}{\text{minimise}} && \frac{1}{2}\|\boldsymbol{w}\|_r^r + C\sum_{i=1}^{\text{N}} \xi_i \\ &\text{subject to} && y_i(\boldsymbol{w}^\top \Phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \quad \forall i \\ & && \xi_i \geq 0, \forall i \end{aligned} \tag{9}$$

where $\boldsymbol{\xi} = \{\xi_i\}_{i=1}^N$ is the slack variable set and the regularisation parameter C determines the trade-off between SVM's generalisation capability and training error. $r = 1$, 2 corresponds to 1-norm and 2-norm SVM, respectively. The solution takes the form $\boldsymbol{w} = \sum_{i=1}^N y_i \alpha_i \Phi(\boldsymbol{x}_i)$. Here, $\alpha_i \geq 0$ and most of them are 0, yielding sparseness. The optimisation (9) can be efficiently solved by linear or quadratic programming in its dual. Refer to [7] for details.

In this experimental work, SVM classifiers with three different kernel functions, linear, quadratic and RBF kernels, are compared with the features calculated from previous section.

4.2 AdaBoost

AdaBoost (Adaptive Boosting) is the first practical and efficient algorithm for ensemble learning [15]. The training procedure of AdaBoost is a greedy algorithm, which constructs an additive combination of weak classifiers such that the exponential loss is minimised:

$$L(y, f(\boldsymbol{x})) = e^{-yf(\boldsymbol{x})} \tag{10}$$

Here $\boldsymbol{x}$ is the labelled training example and y is its label; $f(\boldsymbol{x})$ is the final decision function which outputs the decided class label. AdaBoost combines iteratively a number of weak classifiers to form a strong classifier. Weak classifier is defined as a classifier with accuracy on the training set greater than average. The final strong classifier $H(\cdot)$ can be defined as

$$H(\boldsymbol{x}) = \mathbf{sign}\left(\sum_{i=1}^{N_S} \alpha_i h_i(\boldsymbol{x})\right) \tag{11}$$

where α_i a weight coefficient; $h_i(\cdot)$ a weak learner and N_S the number of weak classifiers. At each new round, AdaBoost selects a new hypothesis $h(\cdot)$ that best classifies training samples with minimal classification error. Each training sample receives a weight that determines its probability of being selected for a training set. If a training sample is correctly classified, then its probability of being used again in a subsequent component classifier is reduced. Conversely, if the pattern is misclassified, then its probability of being used again is increased. In this way, the algorithm focuses more on the misclassified samples after each round of boosting.

5 Experiments

The experimental section is organised as follows. First, the data sets used in this experiment, including how the performance is analysed, are described. Preliminary experiments and the parameters used to achieve optimal results are then discussed. Finally, experimental results and analysis of different techniques are compared. In all the experiments, associated parameters are optimised via cross-validation.

5.1 Experiments on DaimlerChrysler data set

This data set consists of three training sets and two test sets. Each training set contains 4800 pedestrian examples and 5000 non-pedestrian examples (Table 1). The pedestrian examples were obtained from manual labelling and extracting pedestrians in video images at various time and locations with no particular constraints on pedestrian pose or clothing, except that pedestrians are standing in an upright position. Pedestrian images are mirrored and the pedestrian bounding boxes are shifted randomly by a few pixels in horizontal and vertical directions. A border of two pixels is added to the sample in order to preserve contour information. All samples are scaled to size 18 × 36 pixels.

Table 1 Benchmark data set of [3]

No.	Data splits	Pedestrians/ split	Non-pedestrians/ split
Train	3	4800	5000
Test	2	4800	5000

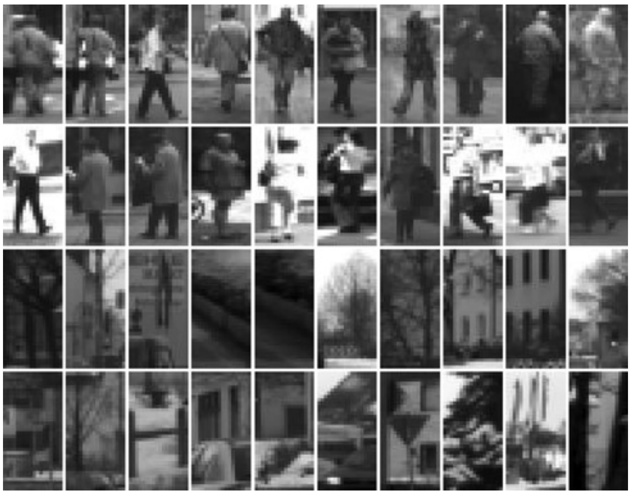

Figure 3 *Pedestrian and non-pedestrian samples from the benchmark data set*

Some examples of pedestrian and non-pedestrian samples are shown in Fig. 3. Performance on the test sets is analysed similar to the techniques described in [3]. For each experiment, three different classifiers are generated. Testing all three classifiers on two test sets yields six different ROC curves. A 95% confidence interval of the true mean detection rate is given by the t-distribution.

5.1.1 Parameter optimisation: For the HOG features, the configurations reported in [8] are tested on the benchmark data sets. However, our preliminary results show a poor performance. This is because of the fact that the resolution of benchmark data sets used (18×36 pixels) is much smaller than the resolution of the original data sets (64×128 pixels). In order to achieve a better result, HOG descriptors are experimented with various spatial/orientation binning and descriptor blocks (cell size ranging from three to eight pixels and block size of 2×2 to 4×4 cells).

Fig. 4*a* shows our experimental results for various descriptor blocks trained using linear SVM. The number of orientation bins is set to nine and the gradient vector is set to unsigned (unsigned gradients are when a gradient vector and its negative vote into the same bin). The following conclusions may be drawn from the figure:

- At data sets' resolution of 18×36 pixels, 2×2 cell blocks of 3×3 pixel cells with a descriptor stride of two to three pixels performs best.

- Increasing the number of cells in a block beyond 3×3 cells decreases the performance proportionally. The explanation for this might be that by increasing the number of cells, we are decreasing the feature length of HOG descriptors to be trained by SVM and, therefore, decreases the overall performance.

- Increasing the number of pixels in a cell (increasing cell width) decreases the performance. The reason may be owing to the fact that by increasing cell width, the HOG descriptors fail to capture the informative spatial information.

- The size of the descriptor strides should be similar to the number of pixels in a cell for optimal performance.

- The HOG feature length per training sample in this experiment is between 2000 and 4000. It seems that there exists a correlation between feature length and the overall performance, i.e. the longer the feature length, the better the performance.

Fig. 4*b* shows the results for different orientation binning and gradient signs. The classifiers are trained using linear SVM. The following observations can be made. Increasing the number of orientation bins increases the detection rate up to about 18 bins (signed gradients). For small resolution human data sets, the gradient sign becomes relevant. The performance of signed gradients significantly outperforms the performance of unsigned gradients. This is in contrast to large resolution human data sets as reported in [8]. From the results shown in Fig. 4*a* and 4*b*, we have decided to use a cell size of 3×3 pixels with a block size of 2×2 cells, descriptor stride of two pixels and 18 orientation bins of signed gradients (total feature length of 8064) to train SVM classifiers.

For region covariance features with nonlinear SVM, our preliminary experiments show a region of size 7×7 pixels, shifted at a step size of two pixels over the entire input image of size 18×36 to be optimal for our benchmark data sets. Increasing the region width and step size decreases the performance slightly. The reason is that increasing the region width and step size decreases the feature length of covariance descriptors to be trained by SVM. However,

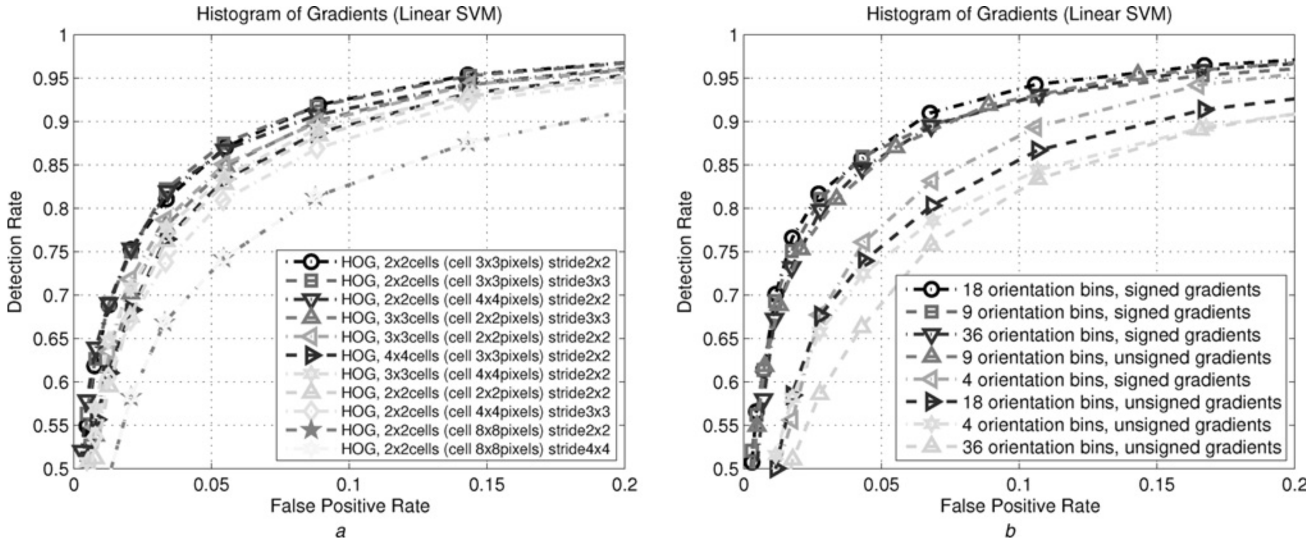

Figure 4 *Performance on histogram of oriented gradient (HOG) features*

a Performance of different descriptor blocks
b Performance of different orientation binning and gradient signs

doi: 10.1049/iet-cvi:20080026

training a linear SVM with a region of size 7 × 7 pixels gives a very poor performance (all positive samples are misclassified). We suspect that the region size is too small. As a result, calculated covariance features of positive and negative samples cannot be separated by linear hyperplane. The feature length of covariance descriptors per training sample is between 1000 and 2000 features. The length is proportional to the number of image statistics used and the total number of regions used for calculating covariance. Preliminary experimental results for region covariance are shown in Fig. 5. For performance comparison, we train both HOG and region covariance features with linear, quadratic and Gaussian kernel SVM using SVMLight [16]. The results show that setting parameter γ in Gaussian RBF kernel to 0.01 gives the optimal performance. Results of different kernel functions are shown in the next section.

5.1.2 Results and analysis based on the data set:

This section provides experimental results and analysis of the techniques described in previous section. We compare our results with LRF features as experimented in [3]. Fig. 6*a* shows detection results of HOG features trained with different SVM classifiers. From the figure, it clearly indicates that a combination of HOG features with quadratic SVM performs best. Obviously non-linear SVM outperforms linear SVM. It is also interesting to note that linear SVM trained using HOG features performs better than non-linear SVM trained using LRF features. This means that HOG features are much better at describing spatial information in the context of human detection than LRF features. Fig. 6*b* shows detection results of covariance features trained with different SVM classifiers. When trained with RBF SVM, a region of size 7 × 7 pixels turns out to perform best compared to other region sizes. From the figure, region covariance features perform better than LRF features when trained with the same SVM kernel (quadratic SVM).

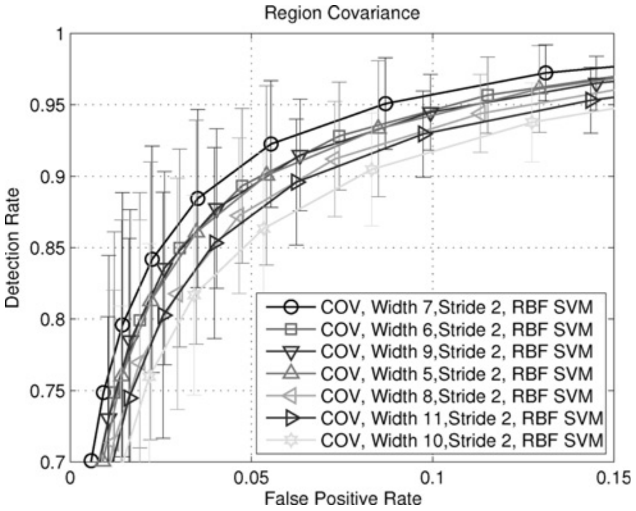

Figure 5 *Performance of different parameters on region covariance features*

A comparison of the best performing results for different feature types are shown in Fig. 7*a*. The following observations can be made. Out of the three features, both HOG and covariance features perform much better than LRF. HOG is slightly better than covariance features. From the figure, we can see that gradient information is very helpful in human detection problems. In all experiments, nonlinear SVM (quadratic or Gaussian RBF SVM) improves classification performance significantly over the linear one. However, this comes at the cost of a much higher computation time (approximately 50 times slower in building SVM model).

5.2 Experiments on the MIT CBCL data set

The MIT CBCL Pedestrian data set (http://cbcl.mit.edu/software-datasets/PedestrianData.html) consists of 924

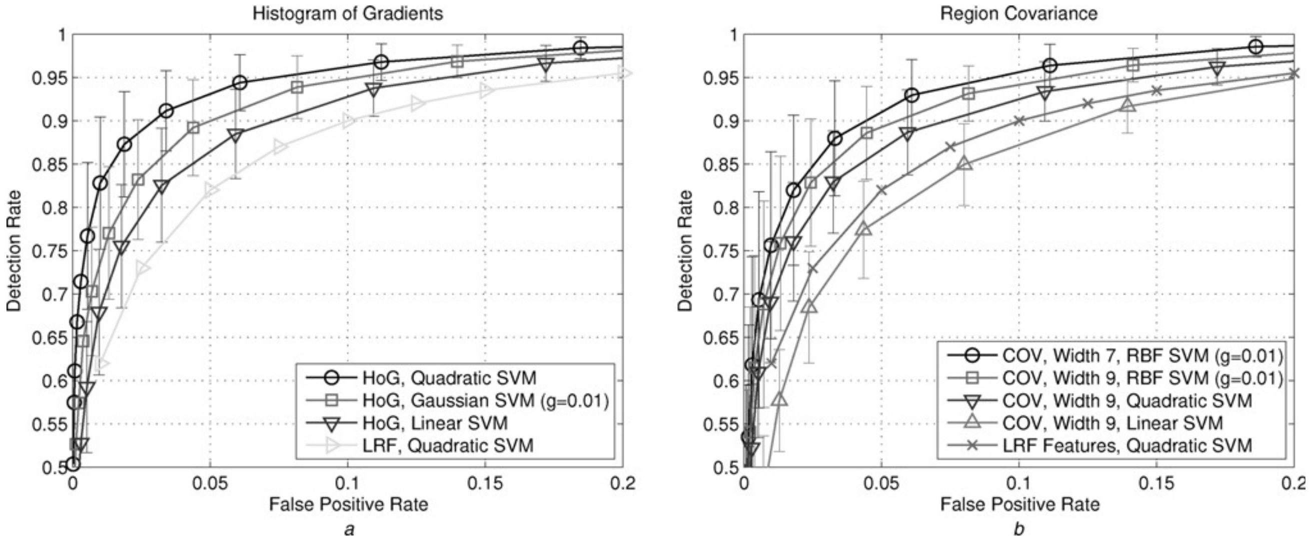

Figure 6 *Performance of different SVM classifiers*

a On histogram of oriented gradients features
b Region covariance features

doi: 10.1049/iet-cvi:20080026

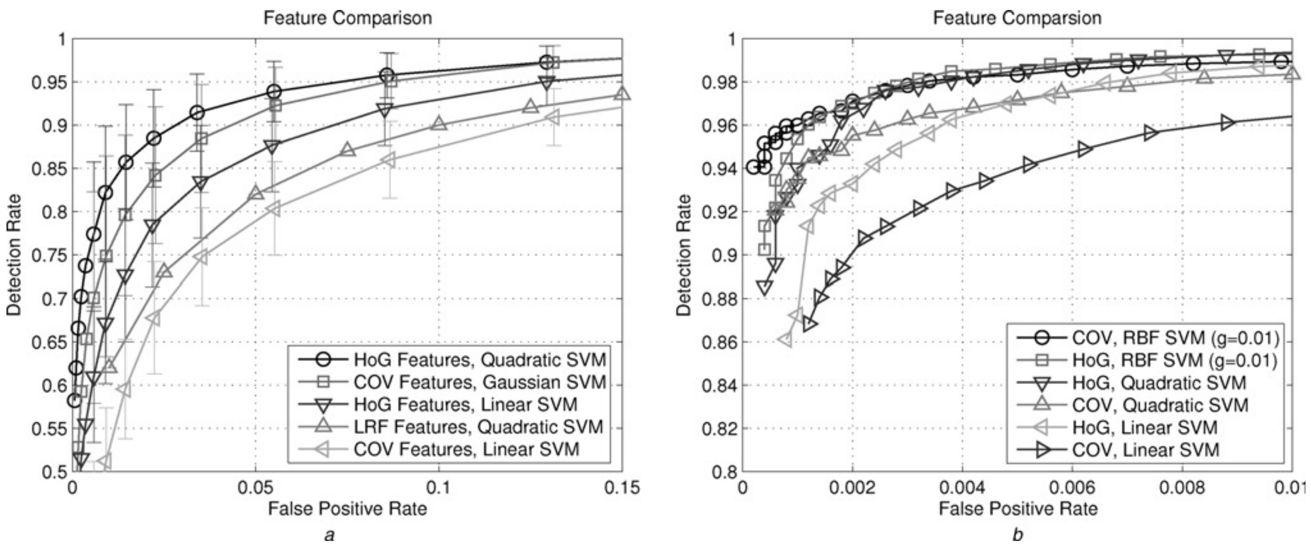

Figure 7 *A performance comparison of the best classifiers for different feature types*

a On the data set of [3]
b On MIT CBCL data set

non-mirrored pedestrian samples. Each sample has a resolution of 64 × 128 pixels. The database contains a combination of frontal and rear view human. We applied the same techniques as described in [3] by dividing the pedestrian samples into five sets (Table 2). Each set consists of 184 pedestrian samples. Each sample is mirrored and shifted randomly by a few pixels in horizontal and vertical directions before being cropped and resized to a resolution of 18 × 36. Each sample contains approximately two to three pixels of margin around the person on all four sides.

For MIT CBCL Pedestrian database, the parameters used are the same as the ones used previously in the data set of [3].

5.2.1 Results and analysis based on MIT CBCL data set:

Fig. 7*b* shows a comparison of experimental results on different feature types using the MIT CBCL pedestrian data set. Both HOG and covariance features perform extremely well on the MIT CBCL data set. This is not too surprising knowing that the MIT CBCL data set contains only a frontal view and rear view of human. Less variation in human poses makes the classification problem much easier for SVM classifiers. As a result, there is a noticeable improvement in the experimental results compared with Fig. 7*a*.

Table 2 MIT CBCL pedestrian data set

No.	Data splits	Pedestrians/ split	Non-pedestrians/ split
Train	3	1840	5000
Test	2	1840	5000

The non-pedestrian examples are randomly sampled from [3]

It is also interesting to note that the performance of covariance features (with Gaussian RBF SVM) is very similar to HOG features trained using Gaussian RBF and quadratic SVM. It even outperforms HOG features at a low false-positive rate. Also nonlinear SVMs are always better than the linear SVMs.

5.3 Experiments on INRIA pedestrian data set

The data set consists of one training set and one test set. The training set contains 1208 pedestrian samples (2416 mirrored samples) and 1200 non-pedestrian images. The pedestrian samples were obtained from manually labelling images taken from a digital camera at various time of the day and at various locations. The pedestrian samples are mostly in standing position. A border of 16 pixels is added to the sample in order to preserve contour information. Experimental results in [8] show that this border provides a significant amount of context that helps improve the detection performance. All samples are scaled to size 64 × 128 pixels.

Unlike the previous experiment where pedestrian samples and non-pedestrian samples are provided, we employ a technique called bootstrapping to incrementally construct a new non-pedestrian training set. We begin by randomly selecting a set of 10 000 non-pedestrian patches from the 1200 non-pedestrian images as an initial non-pedestrian training set. A preliminary classifier is trained and used to classify patches of non-pedestrian samples from the 1200 non-pedestrian images. False-positives are collected and added to the initial negative training set. A new classifier is then trained on the new negative training set. The process can be repeated until there is no significant improvement in the performance of the classifiers.

doi: 10.1049/iet-cvi:20080026

The test set contains 1176 pedestrian samples (mirrored) extracted from 288 images and 453 non-pedestrian images. We evaluate the performance of our classifiers on the given test set using classification approach. Pedestrian samples are cropped from the pedestrian test images while non-pedestrian samples are obtained uniformly from a set of negative images. For quantitative analysis, we plot miss rate against false-positive rate on log–log scale.

5.3.1 Parameter optimisation:

For the HOG features, we use the best configurations reported in [8]. In brief, we used 2 × 2 cell blocks of 8 × 8 pixel cells for spatial binning. For orientation binning, we used nine orientation bins spaced over 0°–180°, that is the sign of the gradients is ignored. For more details, we refer the reader to [8].

Fig. 8 plots the miss rate at 10^{-4} false-positives per window for different region size and step size of covariance features trained using linear SVM. From the figure, decreasing the step size often improves the performance of the classifiers. The reason is that by decreasing the step size, we increase the overlapping covariance regions. In other words, we increase the length of final covariance descriptor vector to be trained by SVM and, as a result, this improves the performance of the classifiers. The smallest step size experimented in this paper is four pixels. Note that decreasing the step size beyond this is possible but it reduces the number of hard non-pedestrian samples that can be fitted into memory during re-training (bootstrapping). Small number of hard negative samples results in little improvement in the performance gain in the next round of re-training. As a result, we have decided to use a region of size 16 × 16 pixels, shifted at a step size of six pixels over the entire human samples of size 64 × 128 pixels. Note that we use a step size of six pixels instead of four pixels for faster computation time.

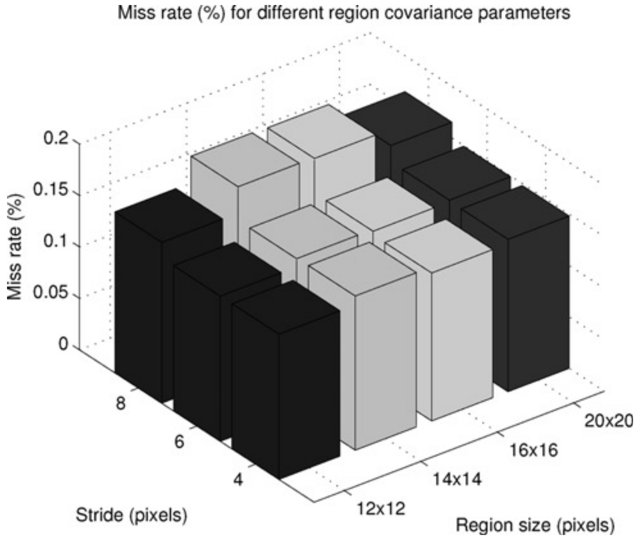

Figure 8 *The miss rate at 10^{-4} false-positives per window for different region size and step size of covariance features*

In our experiments, 10 000 new non-pedestrian patches are added to the training sets in each iteration of bootstrapping. We applied bootstrapping technique twice. Additional iteration of bootstrapping makes very little difference to the performance of the classifier. Note that we used new non-pedestrian patches generated by the linear SVM classifier to train the non-linear SVM classifier. This is because non-linear SVM classifier generates too few false-positives in each iteration of bootstrappings.

5.3.2 Results and analysis based on INRIA data set:

In Fig. 9, we plot the detection performance curve on a log-log scale. The y-axis corresponds to the miss rate and the x-axis corresponds to false-positives per window. Lower values are better. Fig. 9 shows the detection results of HOG

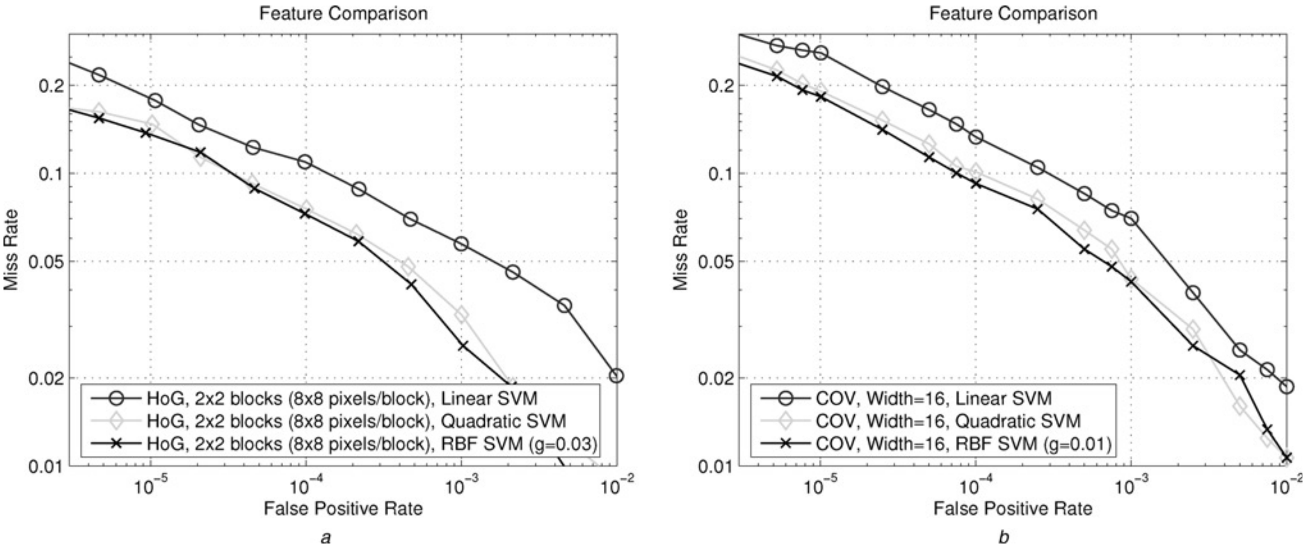

Figure 9 *Performance of different SVM classifiers*

a On histogram of oriented gradients features
b On region covariance features

doi: 10.1049/iet-cvi:20080026

and covariance features trained with different SVM classifiers. Similar to the previous experimental results, nonlinear SVM outperforms linear SVM in both HOG and covariance features.

In Fig. 10, we compare the results of HOG features with region covariance features. From the figure, both features perform similarly in the context of human detection. However, HOG features slightly outperform covariance features. Tuzel *et al.* [9] conclude that the covariance descriptor outperforms the HOG descriptor (using variable-size covariance blocks with logitBoost classification). Instead of minimising exponential loss as in AdaBoost, LogitBoost minimises the logistic loss function. We suspect the difference would be in the covariance block size and the classifier used in their experiments. Fig. 11 shows some of the detection results on INRIA test images using HOG features and covariance features. Note that no post-processing has been applied to the detection results.

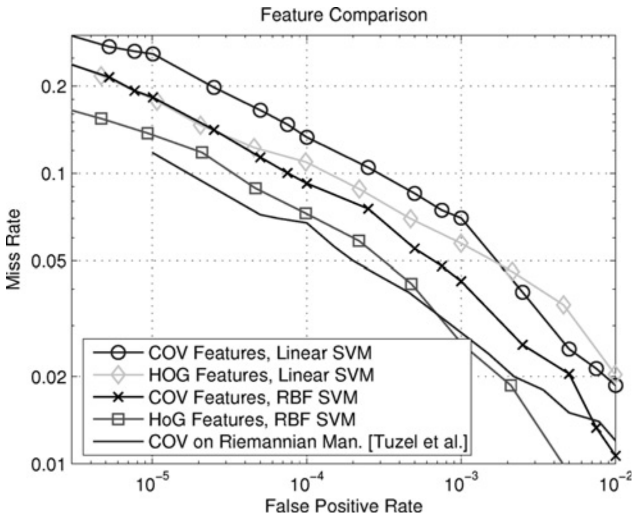

Figure 10 *A performance comparison of the best classifiers for different feature types on the INRIA data set [8]*

5.4 Discussion

Although, covariance features trained using RBF kernel SVM outperform the combination of LRF features with quadratic kernel SVM, the covariance detector has a number of disadvantages. First, the block size used in covariance detector is fixed (i.e. in our experiment, the block size of 7 × 7 pixels performs best on pedestrian data sets of [3] whereas the block of 16 × 16 pixels performs best on INRIA data sets [8]). Having a square fixed size block means we are unable to capture some of the human body parts which have a rectangular shape, for example, human limbs, human torso, etc. It is possible to combine covariance descriptors at multiple block sizes to improve the overall performance. However, combining features from multiple block sizes greatly increases the covariance descriptor size. As a result, computation cost during training and classification also significantly increases.

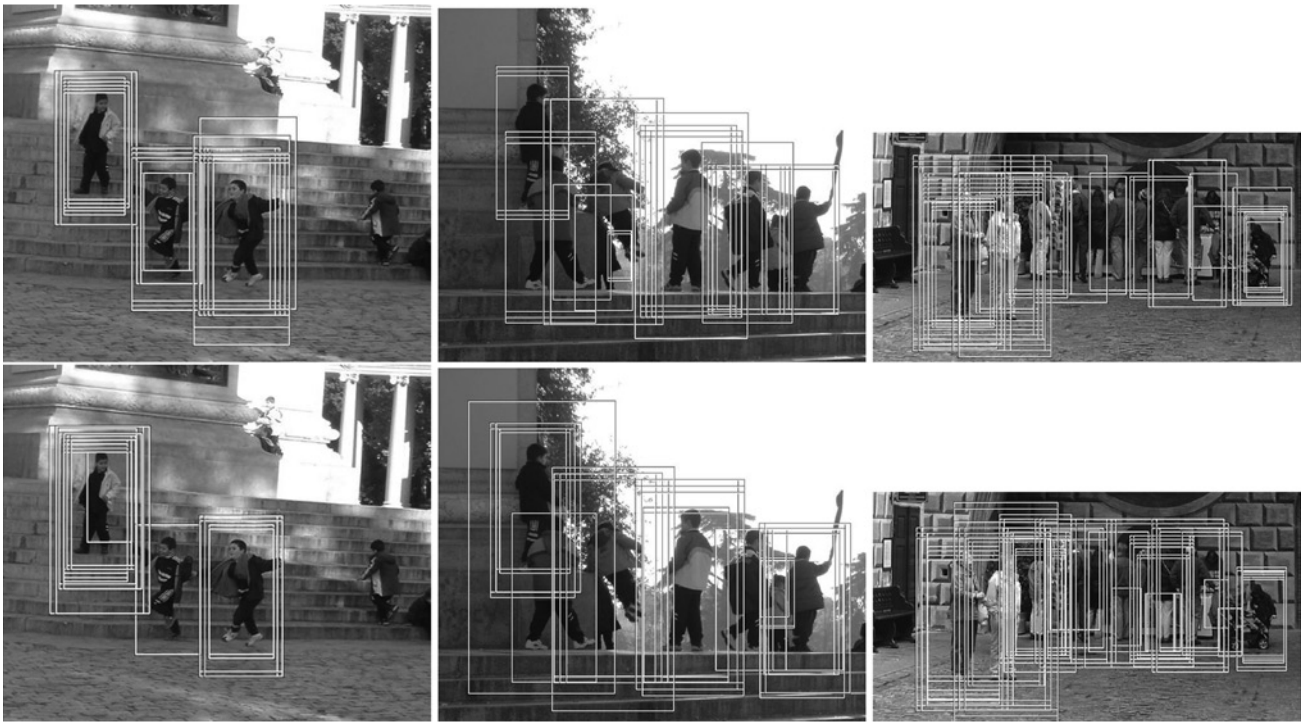

Figure 11 *Detection results on INRIA test images*

The top row shows the detection results of linear SVM using covariance features
The bottom row shows the detection results of linear SVM using HOG features
Note that no postprocessing has been applied to the detection results (scale factor of 0.8 and window step-size of four pixels)
Again we see that covariance and HOG features perform very similarly

doi: 10.1049/iet-cvi:20080026

The second disadvantage is the use of nonlinear SVM as a classifier. Training nonlinear SVM with covariance features has several drawbacks. The first drawback comes from the high-dimensionality of covariance features. Owing to the limited amount of system memory, we are unable to fit all training samples in memory during SVM training. In other words, the large size of the feature vectors limits the number of bootstrapped non-pedestrian samples that can be used. As a result, the detection performance often degrades if the system is trained on a computer with small amount of memory. The second drawback of nonlinear SVM is the parameter tuning process. SVM has a number of parameters that need to be manually optimised for the specific classification task using cross-validation technique. Finding the optimised value for each parameter combination is rather tedious and time-consuming. The third drawback is the high computation time of nonlinear SVM. Although, nonlinear SVM performs significantly better than the linear SVM, it comes at the cost of a much higher computation time during training and evaluation.

As an ongoing work, we are conducting experiments on a new covariance detector that will avoid the above problems by employing AdaBoost feature selection [15] and a cascade of classifiers [1].

6 Conclusion

This paper presented an in-depth experimental study on pedestrian detection using three of the state-of-the-art local features extraction techniques. Our experimental results show that region covariance (correlation coefficient between image statistics) and normalised HOG features in dense overlapping grids significantly outperform the adaptive approach like LRF features. In [3] the authors show that LRF is the best among the features they have compared. Also we show that the covariance features' performance is very similar to HOG's, on all the data sets we have used.

7 Acknowledgments

The authors thank Dr. Fatih Porikli for helpful advice. NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative, in part, through the Australian Research Council.

8 References

[1] VIOLA P., JONES M.J.: 'Robust real-time face detection', *Int. J. Comp. Vis.*, 2004, **57**, (2), pp. 137–154

[2] AMIT Y., GEMAN D., FAN X.: 'A coarse-to-fine strategy for multiclass shape detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004, **26**, (12), pp. 1606–1621

[3] MUNDER S., GAVRILA D.M.: 'An experimental study on pedestrian classification', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (11), pp. 1863–1868

[4] WÖHLER C., ANLAUF J.: 'An adaptable time-delay neural-network algorithm for image sequence analysis', *IEEE Trans. Neural Netw.*, 1999, **10**, (6), pp. 1531–1536

[5] PAPAGEORGIOU C., POGGIO T.: 'A trainable system for object detection', *Int. J. Comp. Vis.*, 2000, **38**, (1), pp. 15–33

[6] VAPNIK V.: The nature of statistical learning theory. 'Statistics for engineering and information science' (Springer-Verlag, Berlin, 2000)

[7] SHAWE-TAYLOR J., CRISTIANINI N.: 'Support Vector Machines and other kernel-based learning methods' (Cambridge University Press, 2000)

[8] DALAL N., TRIGGS B.: 'Histograms of oriented gradients for human detection'. Proc. IEEE Conf. Comp. Vis. Patt. Recogn., San Diego, CA, 2005, vol. 1, pp. 886–893

[9] TUZEL O., PORIKLI F., MEER P.: 'Human detection via classification on Riemannian manifolds'. Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Minneapolis, MN, 2007, pp. 1–8

[10] MORI G., REN X., EFROS A., MALIK J.: 'Recovering human body configurations: combining segmentation and recognition'. Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Washington, DC, 2004, vol. 2, pp. 326–333

[11] WU Y., YU T.: 'A field model for human detection and tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (5), pp. 753–765

[12] MIKOLAJCZYK K., SCHMID C., ZISSERMAN A.: 'Human detection based on a probabilistic assembly of robust part detectors'. Proc. Eur. Conf. Comp. Vis., Prague, Czech Republic, May 2004, vol. 1, pp. 69–81

[13] LOWE D.G.: 'Distinctive image features from scale-invariant keypoints', *Int. J. Comp. Vis.*, 2004, **60**, (2), pp. 91–110

[14] TUZEL O., PORIKLI F., MEER P.: 'Region covariance: a fast descriptor for detection and classification'. Proc. Eur. Conf. Comp. Vis., vol. 2, Graz, Austria, May 2006, pp. 589–600

[15] SCHAPIRE R.E.: 'Theoretical views of boosting and applications'. Proc. Int. Conf. Algorithmic Learn. Theory, London, UK, 1999, pp. 13–25

[16] JOACHIMS T.: Making large-Scale SVM learning practical. 'Advances in Kernel methods – support vector learning' (MIT Press, 1999)

doi: 10.1049/iet-cvi:20080026