ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Image and Vision Computing xxx (2007) xxx–xxx

www.elsevier.com/locate/imavis

Object detection using image reconstruction with PCA

Luis Malagón-Borja [a], Olac Fuentes [b,*]

[a] Computer Science Department, I.N.A.O.E., Tonantzintla, Puebla 72840, Mexico
[b] Computer Science Department, University of Texas at El Paso, El Paso, TX 79968, USA

Received 25 January 2006; received in revised form 8 November 2006; accepted 5 March 2007

Abstract

In this paper, we present an object detection system and its application to pedestrian detection in still images, without assuming any a priori knowledge about the image. The system works as follows: in a first stage a classifier examines each location in the image at different scales. Then in a second stage the system tries to eliminate false detections based on heuristics. The classifier is based on the idea that Principal Component Analysis (PCA) can compress optimally only the kind of images that were used to compute the principal components (PCs), and that any other kind of images will not be compressed well using a few components. Thus the classifier performs separately the PCA from the positive examples and from the negative examples; when it needs to classify a new pattern it projects it into both sets of PCs and compares the reconstructions, assigning the example to the class with the smallest reconstruction error. The system is able to detect frontal and rear views of pedestrians, and usually can also detect side views of pedestrians despite not being trained for this task. Comparisons with other pedestrian detection systems show that our system has better performance in positive detection and in false detection rate. Additionally, we show that the performance of the system can be further improved by combining the classifier based on PCA reconstruction with a conventional classifier using a Support Vector Machine.

Keywords: Object detection; Pedestrian detection; Principal Component Analysis; Support Vector Machines

1. Introduction

The object detection problem can be seen as a classification problem, where we need to distinguish between the object of interest and any other object. In this paper, we focus on a single case of the object detection problem, detecting pedestrians in images.

Pedestrian detection is more difficult than detecting many other objects due to the fact that people can show widely varying appearances when the limbs are in different positions. In addition, people can dress in clothes with many different colors and types. For the characteristics of the pedestrian class we need a robust method that can learn the high variability in the class.

Many object detection systems that have been developed focus on face detection. An early very successful system was presented by Rowley et al. [20], which consists of an ensemble of neural networks and a module to reduce false detections. Similar example-based face detection systems have been developed by Sung and Poggio [22], Osuna et al. [17], and Yang et al. [28].

Most pedestrian detection systems use motion information, stereo vision, a static camera or focus on tracking; important works include [5,8,10,29]. Papageorgiou has reported a system to detect pedestrians in images, without restrictions in the image, and without using any additional information [15,16,9]. It uses the wavelet template to represent the image and a Support Vector Machine (SVM) to classify. The system has been improved in [12,4], detecting pedestrians through the detection of four components of the human body: the head, legs, left arm and right arm. Viola, Jones and Snow developed a system to detect pedestrians from image sequences. This system uses a large set of

* Corresponding author.
E-mail addresses: jmb@inaoep.mx (L. Malagón-Borja), ofuentes@utep.edu (O. Fuentes).

0262-8856/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.imavis.2007.03.004

Please cite this article in press as: L. Malagón-Borja, O. Fuentes, Object detection using image reconstruction with PCA, Image Vis. Comput. (2007), doi:10.1016/j.imavis.2007.03.004

simple filters as features, and then applies the Adaboost algorithm to generate a cascade of classifiers [26].

We present an object detection system to detect pedestrians in gray level images, without assuming any a priori knowledge about the image. The system works as follows: in a first stage a classifier based on Principal Component Analysis (PCA) examines and classifies each location in the image at different scales. Then, in a second stage, the system tries to eliminate false detections based on two heuristics.

The system uses PCA as a classification tool; the main idea is that PCA can compress optimally only the kind of images that were used to do the PCA, and that any other kind of image will not be compressed well in a few attributes, so we do PCA separately for positive and negative examples; when a new pattern needs to be classified we compare the reconstruction made by both sets of principal components (PCs). In order to improve the performance of the classifier we have used the edge image as additional information for it. Additionally, we show that the performance of the system can be further improved by combining the classifier based on PCA reconstruction with a conventional classifier using a Support Vector Machine.

The organization of the reminder of this paper is as follows: Section 2 presents a detailed description of the system. In Section 3 the performance of our system, and a comparison with similar systems are presented. Section 4 reports conclusions and possible directions for future work.

2. The detection system

2.1. Overview of system architecture

The system works scanning the whole image by means of a detection window of size 105×45 pixels; the window is shifting with two pixel jumps to accelerate the process without losing much information from one window to another. We need a classifier that decides for each window if it contains a pedestrian or not. The construction of the classifier is the most complicated stage, we have created a classifier based on image reconstruction with PCA, this classifier uses the edge image in addition to the gray level image.

The scanning of the whole image is part of an iterative process where the image is resized several times to achieve multi-scale detection. For our experiments, the image has been scaled from 0.26 up to 1.35 times its original size, with increases of approximately 17% in every cycle, thus the image is processed at the following 12 different scales: 0.26, 0.3, 0.35, 0.4, 0.47, 0.55, 0.64, 0.74, 0.86, 1, 1.17 and 1.35, this implies that pedestrians of sizes between 78×33 and 404×173 pixels will be detected by the system.

When the system has finished examining the image in all scales, a second process eliminates some detections that are believed to be false detections. The form in which this process works is eliminating the detections that do not repeat at least twice times and eliminating the detections that overlap.

Fig. 1 shows the complete process to detect pedestrians in an image, starting with the gray level image and finishing with the image with the detected pedestrians.

2.2. Stage 1: a classifier based on image reconstruction with PCA

In this stage, we present a classifier that decides if an image of size 105×45 belongs or does not belong to the pedestrian class. This classifier is based on doing image reconstruction using PCA and comparing the reconstructed with the original images. First, the reasons to work with both the gray level image and the edge image are explained, later we explain how the reconstruction of an image is performed using PCA, and finally, we present the way in which a classifier can use these reconstructions to decide if an image belongs or does not belong to the pedestrian class.

2.2.1. Edge images

Because pedestrians appear in many colors and different textures, it is not advisable to use characteristics based on color or texture to do pedestrian detection. For this reason, we have chosen to use the edge image with the idea of obtaining the typical silhouette of a pedestrian and to eliminate useless information for the classifier.

The edge images were computed using x and y Sobel filters, this edge image serves as complementary information to the gray level image and it allows the classifiers to obtain more data to decide if an image is a pedestrian or not.

In Fig. 2 we can see examples of the corresponding edge images of some pedestrian gray level images. In these images we can observe that although the gray level images are very different in color and background, the edge images present fewer changes from one image to another. This is the reason why the edge images are very important to aid in the classifier's task.

2.2.2. Image reconstruction with PCA

Principal Component Analysis is a popular technique for data compression and has been successfully used as an initial step in many computer vision tasks, including face recognition [2,23] and object recognition [14]. The formulation of standard PCA is as follows. Consider a set of m images, each of size $r \times c$. Each image I_i is represented by a column vector v_i of length rc. The mean object of the set is defined by

$$\mu = \frac{1}{m}\sum_{i=1}^{m} v_i$$

C, the covariance matrix, is given by

$$C = \sum_{i=1}^{m}(v_i - \mu)(v_i - \mu)^{\mathrm{T}}$$

Please cite this article in press as: L. Malagón-Borja, O. Fuentes, Object detection using image reconstruction with PCA, Image Vis. Comput. (2007), doi:10.1016/j.imavis.2007.03.004

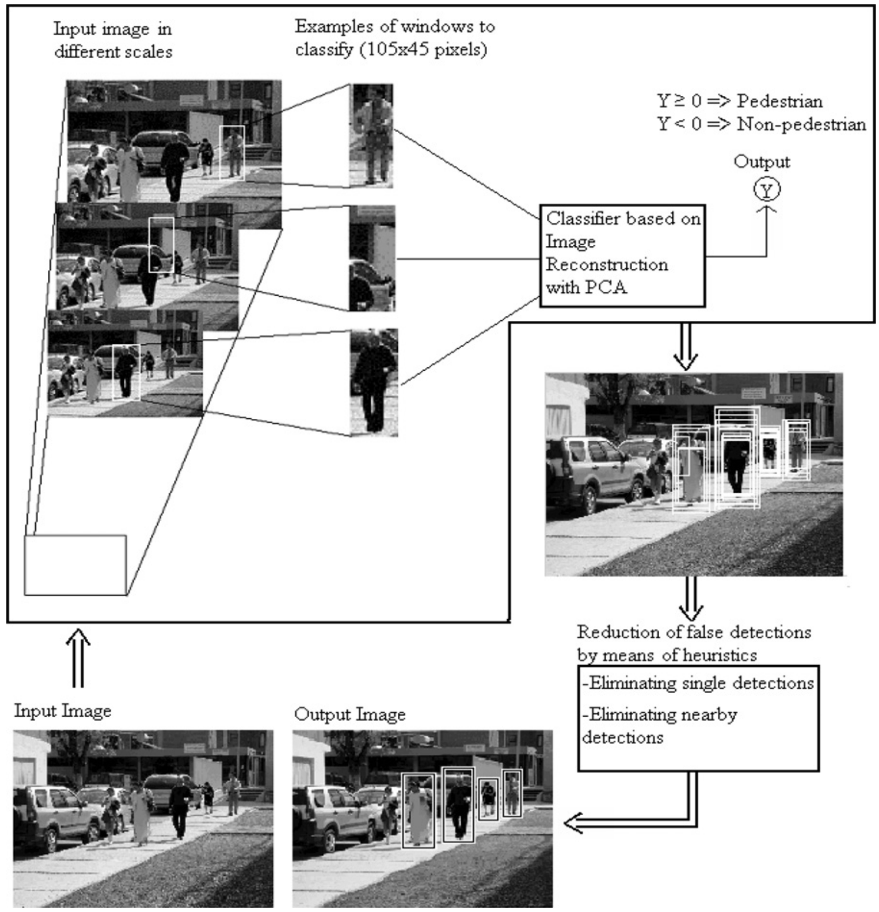

Fig. 1. Architecture of the system for pedestrian detection in images.

Fig. 2. Edge images. The edge images eliminate information about color and texture, therefore they present less variation among pedestrians.

The principal components are then the eigenvectors of C. These eigenvectors can be computed in several ways. Perhaps the easiest one is to solve the generalized eigenvector problem using the QZ algorithm or its variants [13]. It is also common to formulate the problem as that of finding the basis vectors that minimize the reconstruction error and then solve it using standard least-squares techniques (see [3]). In our system we compute these eigenvectors using the implementation provided by Matlab, which is based on the QZ algorithm [13].

If we sort the eigenvectors by decreasing order of their corresponding eigenvalues, a projection onto the space defined by the first k eigenvectors ($1 \leqslant k \leqslant rc$) is optimal with respect to information loss. Let P be the matrix whose columns are the first k eigenvectors of C. The projection of an image u into this eigenspace is given by

$$p = P(u - \mu)$$

When we speak of reconstructing an image with PCA, what we understand is to project the image into the PCs, and from this projection, try to recover the original image. Thus the reconstructed image u' is

$$u' = P^{\mathrm{T}}p + \mu = P^{\mathrm{T}}P(u - \mu) + \mu$$

Let P be the matrix whose columns are the first k eigenvectors of C. The projection of an image into this eigenspace is given by

$$d = |u - u'| = \sqrt{\sum (u_i - u'_i)^2}$$

In general, the more PCs we use to obtain the projection, the less information loss we will have, thus the reconstruction of the image will be more accurate. Also, the more similar u is to the images used to generate P, the better the reconstruction will be for a fixed number of eigenvectors.

2.2.3. Classification using reconstruction

By definition, PCA looks for the set of PCs that best describe the distribution of the data that are being analyzed. Therefore, these PCs are going to preserve better the information of the images from which PCA was performed, or of those that are similar. Thus, if we have a set of PCs that were obtained from a set of pedestrian images only, these must reconstruct better the images of other pedestrians than any other type of images, and viceversa, if we have a set of PCs obtained from images of anything except pedestrians, the reconstruction of the pedestrian images will not be as good. We can observe this fact in Fig. 3, both for gray level images and for edge images.

From this fact we can create a classifier based on image reconstruction with PCA, which decides if an image belongs or does not belong to the pedestrian class. The algorithm to do this classification is the following:

Before doing any classification:

1. Perform PCA on the set of pedestrian gray level images to obtain the projection matrix P_{gp} and the mean μ_{gp}.
2. Perform PCA on the set of pedestrian edge images to obtain the projection matrix P_{ep} and the mean μ_{ep}.
3. Perform PCA on the set of non-pedestrian gray level images to obtain the projection matrix P_{gn} and the mean μ_{gn}.
4. Perform PCA on the set of non-pedestrian edge images to obtain the projection matrix P_{en} and the mean μ_{en}.

When we want to classify a new gray level image g:

1. Obtain the edge image e from g.
2. Do four reconstructions:
   (a) $r_{\mathrm{gp}} = P_{\mathrm{gp}}^{\mathrm{T}}P_{\mathrm{gp}}(g - \mu_{\mathrm{gp}}) + \mu_{\mathrm{gp}}$
   (b) $r_{\mathrm{ep}} = P_{\mathrm{ep}}^{\mathrm{T}}P_{\mathrm{ep}}(e - \mu_{\mathrm{ep}}) + \mu_{\mathrm{ep}}$
   (c) $r_{\mathrm{gn}} = P_{\mathrm{gn}}^{\mathrm{T}}P_{\mathrm{gn}}(g - \mu_{\mathrm{gn}}) + \mu_{\mathrm{gn}}$
   (d) $r_{\mathrm{en}} = P_{\mathrm{en}}^{\mathrm{T}}P_{\mathrm{en}}(g - \mu_{\mathrm{en}}) + \mu_{\mathrm{en}}$
3. Obtain reconstruction errors:
   (a) $d_{\mathrm{gp}} = |r_{\mathrm{gp}} - g|$
   (b) $d_{\mathrm{ep}} = |r_{\mathrm{ep}} - e|$
   (c) $d_{\mathrm{gn}} = |r_{\mathrm{gn}} - g|$
   (d) $d_{\mathrm{en}} = |r_{\mathrm{en}} - e|$
4. Let total error be given by $d_{\mathrm{t}} = d_{\mathrm{gn}} + d_{\mathrm{en}} - d_{\mathrm{gp}} - d_{\mathrm{ep}}$
5. Classify the image according to the following criterion

$$\mathrm{class}(g) = \begin{cases} \text{Pedestrian}, & d_{\mathrm{t}} \geqslant 0 \\ \text{Non-pedestrian}, & d_{\mathrm{t}} < 0 \end{cases}$$

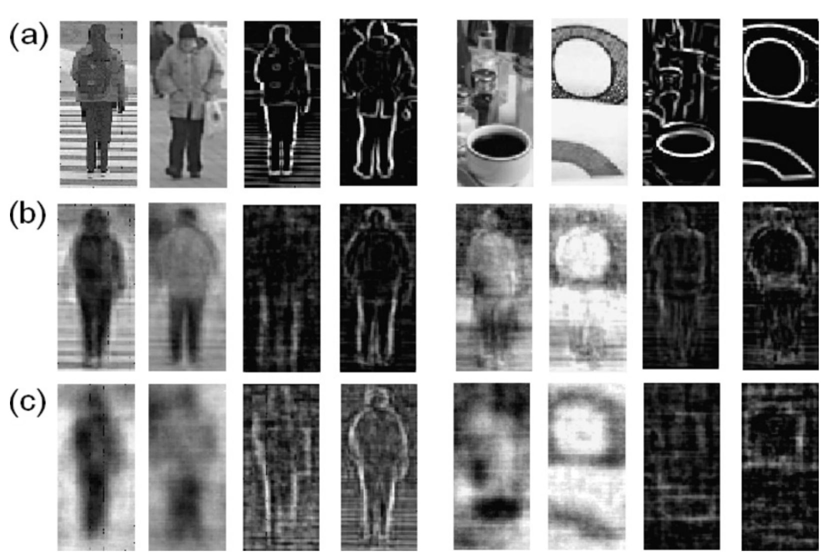

Fig. 3. Image Reconstruction with different sets of PCs. Row (a) shows the original images; row (b) shows the images reconstructed using 100 PCs obtained from pedestrian images, and row (c) shows the images reconstructed using 100 PCs obtained from non-pedestrian images. We can see that, for both the gray level images and the edge images, the pedestrian images are better reconstructed with the PCs obtained from pedestrian images (row b) than with the PCs obtained from non-pedestrian images (row c). This does not happen with the non-pedestrian images, which are better reconstructed with the PCs obtained from non-pedestrian images (row c).

Please cite this article in press as: L. Malagón-Borja, O. Fuentes, Object detection using image reconstruction with PCA, Image Vis. Comput. (2007), doi:10.1016/j.imavis.2007.03.004

2.2.4. Adding a Support Vector Machine classifier

The main feature of the first stage in the algorithm is its simplicity. The preprocessing phase requires to find the edges in every image in the training set and then to compute four sets of principal components, while the classification phase just requires to project the subimage to the four eigenspaces and back to the original spaces and apply a threshold. We will show that this simple method yields accurate results in the pedestrian detection task.

If higher training times are acceptable, we could use a potentially more accurate learning algorithm. We experimented using a Support Vector Machine as base classifier, using the gray levels of the original as well as the edge images as attributes. We will show in the experimental results section that, although the SVM-based classifier does not perform as well as our classifier based on reconstruction errors, a weighted combination of the outputs of both classifiers yields better results than either of them individually, albeit at a significantly higher computational cost.

SVM is a learning algorithm developed by Vapnik [24], that is based on the method of structural risk minimization, which minimizes a bound on the generalization error. The main idea is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. Instead of constructing this hyperplane in the original input space, SVM uses a nonlinear kernel to project the original variables into a higher-dimensional feature space, which yields a nonlinear decision surface in input space. This is a very powerful feature, because it allows SVM to overcome the limitations of linear boundaries. For more information about this algorithm we refer the reader to [21,24].

Given that Support Vector Machines have proven to perform well over high dimensionality data, they have been successfully used in many vision-related applications, such as face detection [17], 3D object recognition [19], and tracking [1].

In our work, the optimization algorithm used for training the support vector classifier is an implementation of Platt's sequential minimal optimization algorithm [18]. The kernel function used for mapping the input space was a polynomial of exponent one. We used the implementation of SVM included in the WEKA environment [27].

The optimal separating hyperplane found by an SVM algorithm for a particular training set is given by the vector w and the scalar b. Thus a test example x is classified as positive iff $w \cdot x - b \geqslant 0$. Let w_p and b_p be the parameters of the optimal separating hyperplane obtained using the training set of pedestrian and non-pedestrian images. Then the classification rule that combines our original classifier and the SVM is:

$$\text{class}(g) = \begin{cases} \text{Pedestrian}, & \alpha_1 d_t + \alpha_d (w_p \cdot u - b_p) \geqslant 0 \\ \text{Non-pedestrian}, & \text{otherwise} \end{cases}$$

where α_1 and α_2 are positive weights that control the relative influence of both classifiers. Experimentally, we found that classification accuracy was not very sensitive to the choice of weights as long as α_1 was greater than α_2, since the first classifier is more accurate than the second. For the experiments we used $\alpha_1 = 3$ and $\alpha_2 = 1$.

2.3. Stage 2: reduction of false detections by means of heuristics

The output after classifying all the windows of the image in multiple scales still contains a significant number of false detections, in this stage we present two heuristics that allow to reduce the number of false detections by means of two processes, namely, eliminating single detections and eliminating nearby detections.

2.3.1. Eliminating single detections

As we can see in Fig. 4a, most of the pedestrians are detected at multiple nearby positions and scales, while false detections usually appear at a single position. This observation allows us to eliminate some false detections, eliminating detections that appear only once.

Each detection found can be grouped with those detections whose centroid is inside the same neighborhood,

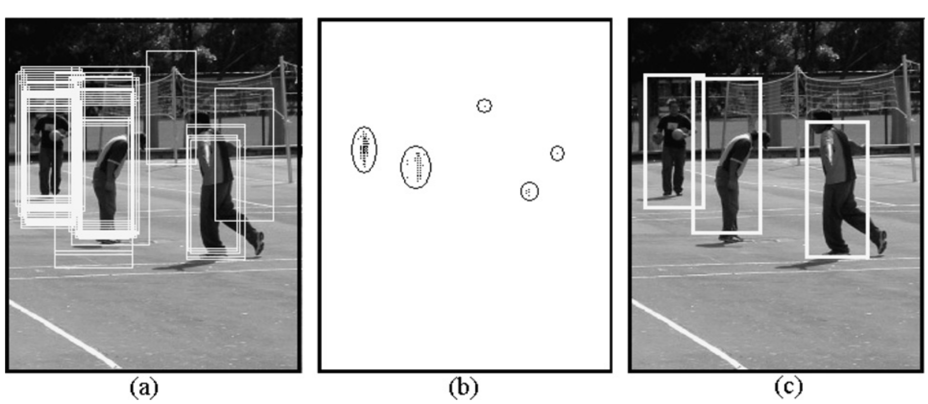

Fig. 4. Process to eliminate single false detections. In figure (a) we can see the original detections found by the classifier. In figure (b) each detection is grouped with those detections whose centroid is in the same neighborhood. Finally in figure (c) we have the grouped detections composed by two or more original detections.

obtaining a new set of detections which we will call grouped detections, composed by one or more of the original detections. Once we have the set of grouped detections we will ignore the original detections, and we will eliminate the grouped detections composed by only one original detection. We can see this process in Fig. 4.

2.3.2. Eliminating clustered detections

If a window is identified correctly as a pedestrian, then it is very likely that there are no pedestrians either above or below it, and if there are pedestrians beside it, they cannot be too overlapped. This heuristic allows us to eliminate nearby detections. With this purpose, we define a region around a detection which we are going to use to eliminate any detection whose centroid is inside this region. The size of the region was defined empirically as 1.4 times the detection height upwards and downwards from the centroid and between 0.5 and 0.75 times the detection width towards each side of the centroid.

We know that when we have multiple detections we must choose only one to keep, but how do we make this decision? A reasonable way to choose is to maintain the grouped detection composed by the most original detections; nevertheless, we observe that usually the biggest detections were the correct ones, due to the fact that arms, legs and head are often confused with pedestrians, so when we need to decide among a set of detections that are in the same region, we must consider the number of times that they have been detected originally as well as the size of the detected regions.

To achieve this, the detections that compose a grouped detection are weighted by their height, then the grouped detection with the greatest *Preference*, according to the following formula is chosen.

$$\text{Preference} = \text{Detections} \cdot \text{Weight}(\text{height})$$

where *Detections* is the number of original detections that compose the grouped detection that we are evaluating and *Weight* is a function that determines the value that each detection has, according to the *height* of the grouped detection, and is given by the formula:

$$\text{Weight}(\text{height}) = (\text{height} - 50)^2$$

There are very few cases where this heuristic does not work, and thus it allows to eliminate many false detections when the classifier confuses the arms, the legs, or some other object with a pedestrian. Fig. 5 shows an example where several false detections are eliminated applying this heuristic to the output of the classifier.

3. Experimental results

As we explained in the previous section, we need a set of pedestrian images and a set of non-pedestrian images to obtain the four sets of PCs from which we are going to perform the four reconstructions, and to train the SVM classifier.

The pedestrian images were obtained from the MIT pedestrian database, which contains pedestrians in frontal or rear views under different scene conditions. We converted these color images to gray level and we cropped part of the background to reduce the variation that exists among pedestrian images, finally obtaining a set with 500 gray level images.

For the negative images we obtained two different sets, the first set of negative images had 2315 images that were obtained randomly from a set of 90 images of scenery that did not contain any pedestrian and was used in the classifier based on image reconstruction with PCA. The second set had 2248 images obtained in a bootstrap manner [22] and it was used to train the SVM classifier.

For the classifier based on image reconstruction we used 200 PCs in each set, that contain between 75% and 85% of the variance, to do the reconstructions. It was observed that this number of PCs allowed a good classification.

The SVM classifier used as kernel a Radial Basis Function (RBF) with $\sigma = 0.00015$ and $C = 10{,}000$. The SVM was trained with the projection of the 2748 pedestrian and non-pedestrian images onto the first 200 PCs of six different PCs sets, each one obtained from a different set of images; these sets are the following:

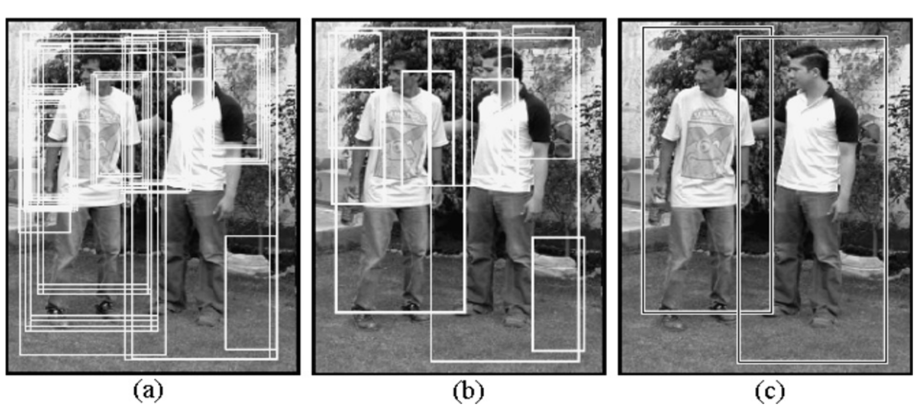

Fig. 5. Process to eliminate nearby false detections. Figure (a) shows the detections found by the classifier. Figure (b) shows the grouped detections. In figure (c) the grouped detections with the greatest *Preference* have been preserved and the nearby grouped detections have been eliminated.

Please cite this article in press as: L. Malagón-Borja, O. Fuentes, Object detection using image reconstruction with PCA, Image Vis. Comput. (2007), doi:10.1016/j.imavis.2007.03.004

ARTICLE IN PRESS

1. The 500 pedestrian gray level images.
2. The 2248 non-pedestrian gray level images.
3. The 2748 pedestrian and non-pedestrian gray level images.
4. The 500 pedestrian edge images.
5. The 2248 non-pedestrian edge images.
6. The 2748 pedestrian and non-pedestrian edge images.

So each image of size 105×45 pixels was described by 1200 attributes (200 for each projection).

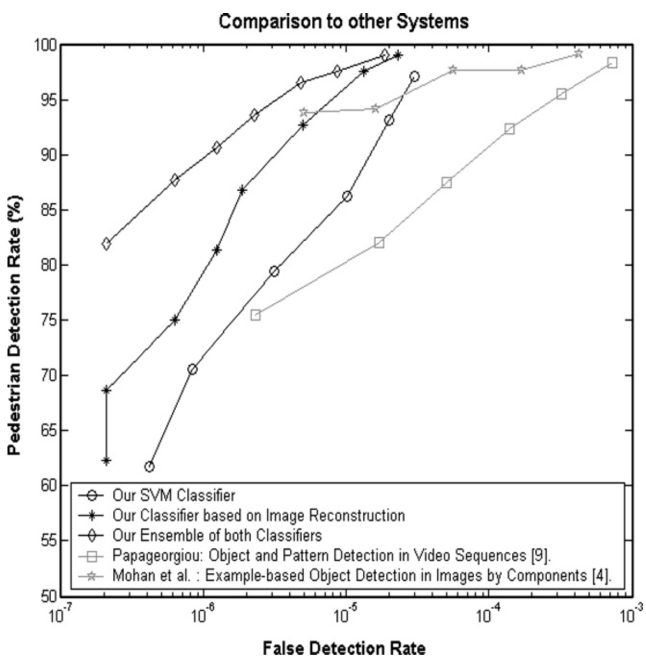

Fig. 6. ROC curves comparing the performance of our classifiers versus the best reported in the literature. The detection rate is plotted against the false detection rate measure on logarithmic scale.

The system was tested with a database containing 204 pedestrian images in frontal or rear view to determine the pedestrian detection rate; these images were not used before. The false detection rate was obtained by running the system over a database with 17 images that did not contain any pedestrian; by running the system over these 17 images 4,850,103 windows were classified.

In general, the performance of any object detection system shows a tradeoff between the positive detection rate and the false detection rate. We ran the system over the test images at several different thresholds. The results were plotted as a Receiver Operating Characteristic (ROC) curve, given in Fig. 6. We can see that the best individual classifier is the classifier based on image reconstruction, this is better than the best reported in the literature of pedestrian detection in systems than do not assume any a priori scene structure or use any motion information. The SVM shows the second best performance among individual classifiers, and the best overall performance is obtained by the ensemble that outputs the weighted combination of the classifier based on reconstruction error and the SVM. The curve indicates that the system can achieve a detection rate of 99.02% with one false positive every 53,890 windows examined, or if we want a more conservative system, it can achieve a detection rate of 90.69% with one false detection every 808,351 windows examined. Fig. 7 shows the result of applying the system to sample images in cluttered scenes under different conditions.

4. Conclusions and future work

In this paper, we have presented an object detection system for static images, without assuming any a priori knowledge, applied to the specific problem of locating pedestrians in cluttered gray level images.

Fig. 7. These images demonstrate the capability of the system for detecting people in still images with cluttered backgrounds.

Please cite this article in press as: L. Malagón-Borja, O. Fuentes, Object detection using image reconstruction with PCA, Image Vis. Comput. (2007), doi:10.1016/j.imavis.2007.03.004

Our system is able to detect frontal and rear views of pedestrians, and usually it can also detect side views of pedestrians despite not being trained for this task.

The success of PCA for pedestrian detection comes from its capability to capture most of the information about the objects of interest by using both intensity and edge images. This allows to distinguish between a pedestrian image and any other image in the huge universe of non-pedestrian images. The success of the ensemble of classifiers is due to the low correlation in errors between both classifiers and the capability of each classifier to learn the pedestrian class accurately.

An interesting possibility for future work is to use alternate projection spaces to derive the attributes used by the SVM. In particular, Fisher's linear discriminant (FLD), which has been used successfully in the face recognition domain [2], could be used to derive those features. This has the potential to provide improved results because, in contrast to principal components, FLD uses class information to find a projection that separates examples of different classes.

The current system does not work as well for side views of pedestrians as for pedestrians in frontal or rear views. To solve this, we can add side views of pedestrians to the training set, or we can create an additional part of the system that could be specialized for these views.

Another way to improve the system's performance is to obtain more positive and negative examples for training. We only use 500 positive examples and 2315 negative examples, while other works in object detection use around 2000 positive examples and 10,000 negative examples.

The framework described here is applicable to other domains besides pedestrian detection; it can be generalized to the detection of several different types of objects, such as faces, vehicles, and others. A promising direction for future work is to apply the method presented in this paper in a component-based approach. This approach has shown better performance in pedestrian detection (see [12,4]) than a similar full-body pedestrian detector [15,16,9]. Also, we intend to investigate if the classification approach based on reconstruction error with PCA can be applied to other problems in computer vision and other areas.

References

[1] Shai Avidan, Subset selection for efficient SVM tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[2] Peter N. Belhumeur, Joao P. Hespanha, David Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection with a view-based representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[3] Ronan Fablet, Michael J. Black, Automatic detection and tracking of human motion with a view-based representation, in: Proceedings of European Conference on Computer Vision, 2002, pp. 476–491.

[4] Ismail Haritaoglu, David Harwood, Larry S. Davis, W4: Who, when, where, what: a real time system for detecting and tracking people, in: Procedings of Third Face and Gesture Recognition Conference, 1998, pp. 222–227.

[5] Sumer Jabri, Zoran Duric, Harry Wechsler, Azriel Rosenfeld, Detection and location of people in video images using adaptive fusion of color and edge information, in: Procedings of International Conference on Pattern Recognition, vol. 4, 2000, pp. 4627–4631.

[6] Fernando De la Torre, Michael Black, Robust principal component analysis for computer vision, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. 362–369.

[7] Anuj Mohan, Object detection in images by components, A.I. Memo 1664, Center for Biological and Computational Learning, MIT, Cambridge, MA, 1999.

[8] Anuj Mohan, Constantine Papageorgiou, Tomaso Poggio, Example-based object detection in images by components, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (4) (2001) 349–361.

[9] C.B. Moler, G.W. Stewart, An algorithm for generalized matrix eigenvalue problems, SIAM Journal on Numerical Analysis 10 (2) (1973) 241–256.

[10] Shree K. Nayar, Sameer A. Nene, Hiroshi Murase, Real-time 100 object recognition system, in: IEEE International Conference on Robotics and Automation (ICRA), vol. 3, Minneapolis, MN, 1996, pp. 2321–2325.

[11] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, Tomaso Poggio, Pedestrian detection using wavelet templates, in: Procedings of IEEE Conference Computer Vision and Pattern Recognition, 1997, pp. 193–199.

[12] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, Tomaso Poggio, A trainable system for people detection, in: Procedings of Image Understanding Workshop, 1997, pp. 207–214.

[13] Edgar Osuna, Robert Freund, Federico Girosi, Training support vector machines: an application to face detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 130–136.

[14] Constantine Papageorgiou, Object and pattern detection in video sequences, Master's thesis, M.I.T., Cambridge, MA, 1997.

[15] John C. Platt, Sequential minimal optimization: a fast algorithm for training support vector machines, in: Bernhard Scholkopf, Christopher J.C. Burges, Alex J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, USA, 1999, pp. 185–208.

[16] Massimiliano Pontil, Alessandro Verri, Support vector machines for 3D object recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (6) (1998) 637–646.

[17] Henry A. Rowley, Shumeet Baluja, Takeo Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.

[18] Bernhard Scholkopf, Alexander J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2001.

[19] Kah-Kay Sung, Tomaso Poggio, Example-based learning for view-based human face detection, IEEE Transactions Pattern Analysis and Machine Intelligence 20 (1) (1998) 39–51.

[20] Matthew Turk, Alex Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.

[21] Vladimir N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, USA, 1995.

[22] Paul Viola, Michael Jones, Daniel Snow, Detecting pedestrians using patterns of motion and appearance, International Journal of Computer Vision 63 (2) (2005) 153–161.

[23] Ian Witten, Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, second ed., Morgan Kaufmann, 2005.

[24] Ming-Hsuan Yang, Dan Roth, Narendra Ahuja, A SNoW-based face detector, in: Advances in Neural Information Processing Systems, vol. 12, 2000, pp. 855–861.

[25] Liang Zhao, Charles E. Thorpe, Stereo and neural network-based pedestrian detection, IEEE Transactions on Intelligent Transportation Systems 1 (3) (2000) 148–154.

Please cite this article in press as: L. Malagón-Borja, O. Fuentes, Object detection using image reconstruction with PCA, Image Vis. Comput. (2007), doi:10.1016/j.imavis.2007.03.004