PERGAMON

Pattern Recognition 36 (2003) 1997–2006

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Full-body person recognition system

Chikahito Nakajima[a], Massimiliano Pontil[b,*], Bernd Heisele[c], Tomaso Poggio[c]

[a] *Central Research Institute of Electric Power Industry, 2-11-1, Iwado Kita, Komae, Tokyo 201-8511, Japan*
[b] *Department of Computer Science, University College London, Gower Street, London WC1E 6BT, England, UK*
[c] *Center for Biological and Computational Learning, M.I.T., 45 Carleton St., 02142, Cambridge, MA, USA*

Received 15 January 2003; accepted 15 January 2003

Abstract

We describe a system that learns from examples to recognize persons in images taken indoors. Images of full-body persons are represented by color-based and shape-based features. Recognition is carried out through combinations of Support Vector Machine (SVM) classifiers. Different types of multi-class strategies based on SVMs are explored and compared to k-Nearest Neighbors classifiers. The experimental results show high recognition rates and indicate the strength of SVM-based classifiers to improve both generalization and run-time performance. The system works in real-time.

Keywords: Multi-class classification; Person recognition; Pattern classification; Support vector machines; Surveillance systems; Object recognition

1. Introduction

Digital video cameras connected to computers have come into wide use recently. Still, visual surveillance is mainly performed by humans. In the future, automatic visual surveillance systems could play an important role in supporting and eventually replacing human observers. To become practical these systems have to be able to perform the following basic tasks: (a) Detect and track people and (b) person recognition. In this paper we mainly focus on the identification task. However, since person identification often implies a prior detection step we also discuss approaches to detecting people.

There is a vast number of computer vision techniques that can be used in automatic visual surveillance systems: Face detection [1–4] and face recognition [5–7] have been thoroughly studied over the past 10 years in computer vision. While recent face detection systems [1,3] are able to deal with large pose variations, face recognition systems are restricted to identifying persons in frontal and near-frontal views only. More recently, learning-based techniques [8] and template matching [9] have been applied to detecting people in still images. As shown in Refs. [10,11] the periodicity of gait allows to detect walking people in image sequences. Gait has also been used to for person recognition in image sequences [12]. The results in Ref. [12] have been reported on a small set of five subjects. It is not clear whether larger numbers of people can be distinguished based on gait only.

Although the above mentioned techniques show good results in constrained application scenarios, the general task of people detection and identification still presents a number of challenges: The invariance against pose changes, the invariance against changes in illumination, and the selection of image features that allow to reliably identify people.

This paper addresses surveillance scenarios where the pose of people is unconstrained which makes it difficult to apply common face recognition algorithms. Such scenarios typically occur in so called crowd surveillance applications.

* Corresponding author.
E-mail addresses: nakajima@criepi.denken.or.jp (C. Nakajima), pontil@dii.unisi.it (M. Pontil).

0031-3203/03/$30.00
doi:10.1016/S0031-3203(03)00061-X

In addition the image resolution is too low for face recognition systems to be applied.

In our experimental setup the goal was to recognize members of our Lab while they were using an espresso coffee machine located in the Lab's main office [13]. The camera was located in front of the coffee machine at a distance of about 4.5 m; background and lighting were almost invariant. Recognition was based on the assumption that the same person was going to have the same general appearance (clothes) during the day.

Recognition is carried out through combinations of Support Vector Machines (SVMs). SVMs [14,15] have been already successfully applied to various two-class problems, such as pedestrian and face detection [16,17,1]. Recently several methods have been proposed in order to expand the application field of SVMs to multi-class problems [18–21]. In this paper, we use and compare these methods to recognize persons and their poses. The experiments show high recognition rates indicating the relevance of our system for the development of more sophisticated indoor surveillance applications.

The paper is organized as follows: Section 2 presents an outline of the person recognition system. In Section 3 we briefly describe SVMs and multi-class SVMs. Section 4 presents the experimental results for recognizing persons and pose estimation. Section 5 discusses multi-class SVMs and the extension methods of the system. We conclude in Section 6.

2. System outline

The system consists of two modules: Image pre-processing and person/pose recognition. Fig. 1 shows the system overview. Each image from the camera is forwarded to the pre-processing module where we extract the person from the background and calculate shape and color features. Based on these features, the persons identity and pose are determined in the recognition module.

2.1. Pre-processing

The pre-processing module consists of two parts: detection of moving persons and extraction of image features.

2.1.1. Person detection

The system uses two steps to detect a moving person in an image sequence. In the first step the system subtracts the current background image (see below) from the latest k images,[1] and stores one of these k images, if any, whose corresponding subtracted image has energy larger than a threshold. However, the result of background subtraction may include a lot of noise and sometimes the stored image does not contain a person. The second step helps discarding

[1] In our experiments we chose $k = 3$.

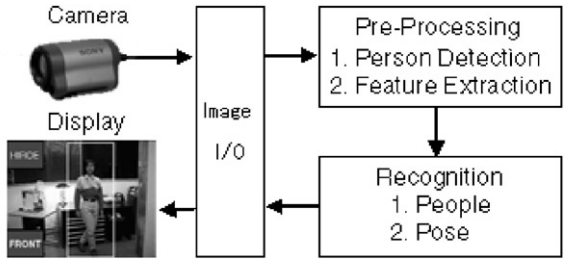

Fig. 1. Outline of the system.

Fig. 2. An example of moving person detection.

those images not containing a person. To this purpose, the system extracts the silhouette of a possible person by using edge detection. Assuming that the person is slightly moving between two frames, the system performs edge detection on the image obtained by subtracting two consecutive images in the sequence. If the number of edge pixels is larger than a threshold, one of the k images is eventually stored. Finally, if no person image is detected, the background is updated by computing the average of the k latest images. Fig. 2(a) shows an image from the sequence and Fig. 2(b) shows the combined result of the two steps.

2.1.2. Feature extraction

Once the person has been detected and extracted from the background, we calculate different types of image features:

(1) RGB color histogram
We calculate one dimensional color histogram with 32 bins for each color channel. The total number of extracted features is 96 (32×3) for a single image.

(2) Normalized color histograms
We calculate two dimensional normalized color histograms; $r = R/(R+G+B)$, $g = G/(R+G+B)$. Again, we chose 32 bins for each color channel. Overall, the system extracts 1024 (32×32) features from a single image.

(3) RGB color histogram+shape histogram
We calculate simple shape features of people by counting pixels along rows and columns of the extracted body images. We chose a resolution of 10 bins for column histograms and 30 bins for row histograms. The total

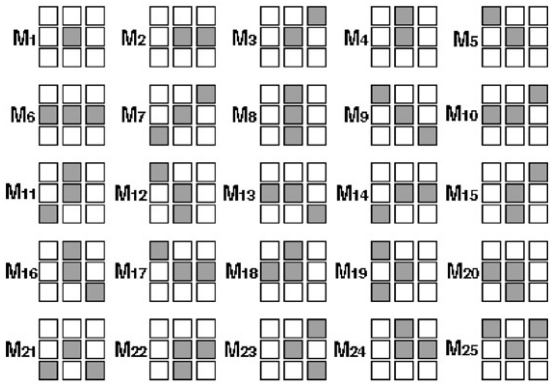

Fig. 3. Shape patterns.

number of extracted features is 136, 32×3 for the RGB histograms and $10 + 30$ for the shape histograms.

(4) Local shape features
Local features of an image are obtained by convolving the local shape patterns shown in Fig. 3. These patterns were introduced in Ref. [22] for position invariant person detection. Let M^i, $i = 1, \ldots, 25$, be the patterns in Fig. 3 and V_k the 3×3 patch at pixel k in an image. We consider two different types of convolution operations. The first is the linear convolution given by $\sum_k V_k \cdot M^i$, where the sum is on the image pixels. The second is a non-linear convolution given by $F_i = \sum_k C_{(k,i)}$, where

$$C_{(k,i)} = \begin{cases} V_k \cdot M^i & \text{if } V_k \cdot M^i = \max_j (V_k \cdot M^j) \\ 0 & \text{otherwise.} \end{cases}$$

The system uses the simple convolution from the pattern 1 to 5 and the non-linear convolution from the pattern 6 to 25. The non-linear convolution mainly extracts edges and has been inspired by recent work in the field of brain models [23]. The shape features are extracted for each of the following color channels separately: $R + G - B$, $R - G$ and $R + G$. This color model has been suggested by physiological studies [24]. The system extracts 75 (25×3) features from the three color channels.

2.2. Recognition

In order to develop a recognition system we first collect a data set of N images of people and manually label it according to the identity and pose of the person. The set of input–output examples is

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N),$$

where the input $\mathbf{x}_i$ denotes the feature vector extracted from image i and the output y_i is a class label. If the task is person recognition, the class label y_i encodes the identity of the person, in the case of pose estimation the class label encodes the pose (right, left, front or back) of the person. We train different SVM classifiers on the labeled data to perform the multi-class classification task for person identification and pose estimation.

SVMs are a technique to train classifiers, regressors and probability densities that is well-founded in statistical learning theory [15]. One of the main attractions of using SVMs is that they are capable of learning in *sparse*, *high dimensional spaces* with very few training examples. SVMs accomplish this by minimizing a bound on the empirical error and the complexity of the classifier, at the same time. This controlling of both the training error *and* the classifier's complexity has allowed SVMs to be successfully applied to very high dimensional learning tasks such as face detection [25], 3-D object recognition [18], stop word detection in speech signals [26], and text categorization [27]. We will make use of this property of being able to apply SVMs to very high dimensional classification problems.

3. Support vector machines

In this section we briefly overview the main concepts of SVMs [15] for pattern classification. More detailed accounts are Refs. [15,28,19].

3.1. Binary classification

SVMs perform pattern recognition for two-class problems by determining the separating hyperplane[2] with maximum distance to the closest points of the training set. These points are called *support vectors*. If the data is not linearly separable in the input space, a non-linear transformation $\Phi(\cdot)$ can be applied which maps the data points $\mathbf{x} \in \mathbb{R}^n$ into a high (possibly infinite) dimensional space H which is called feature space. The data in the feature space is then separated by the optimal hyperplane as described above.

The mapping $\Phi(\cdot)$ is represented in the SVM classifier by a kernel function $K(\cdot, \cdot)$ which defines an inner product in H, i.e. $K(\mathbf{x}, \mathbf{t}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{t})$. The decision function of the SVM has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i c_i K(\mathbf{x}_i, \mathbf{x}), \tag{1}$$

where ℓ is the number of data points, and $c_i \in \{-1, 1\}$ is the class label of training point $\mathbf{x}_i$. Coefficients α_i in Eq. (1) can be found by solving a quadratic programming problem with linear constrains. The support vectors are the nearest points to the separating boundary and are the only ones for which α_i in Eq. (1) can be nonzero.

An important family of admissible kernel functions are the Gaussian kernel:

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|/2\sigma^2),$$

[2] SVMs theory also includes the case of non-separable data, see Ref. [15].

with σ the variance of the Gaussian, and the polynomial kernels:

$$K(\mathbf{x},\mathbf{y}) = (1+\mathbf{x}\cdot\mathbf{y})^d,$$

with d the degree of the polynomial. For other important examples of kernel functions used in practice see Refs. [28,15].

3.2. Multi-class classification

Like many discriminative classifiers, SVMs are designed to solve binary classification problems. However, many real-world classification problems involve more than two classes. Attempts to solve q-class problems with SVMs have involved training q SVMs, each of which separates a single class from all remaining classes [14,29], or training q^2 machines, each of which separates a pair of classes [30,31,18]. The first type of classifiers are usually called *one-vs-all*,[3] while classifiers of the second type are called *pairwise* classifiers.

In this paper we used three types of multi-class classification schemes which are based on the combination of binary SVMs: the first scheme combines one-vs-all classifiers and the other two schemes combine pairwise classifiers. Let us see how each of these systems is computed:

- *One versus all*
  In the one-vs-all type scheme [14], there is one SVM classifier associated to each class. For each class $j \in \{1,\dots,q\}$ the corresponding classifier is trained to separate the examples in this class (positive labeled) from the remaining ones (negative labeled). A new input vector is classified in one class for which associated classifier has the highest score among all classifiers.
- *Bottom-UP decision tree*
  We begin to form a binary tree with $q-1$ nodes. This tree has no more than $q-1$ layers and exactly q leaves. Each node in the tree has two inputs and one output. We assign a different class to each leaf.
  Classification of an input point starts at the bottom layer. For each node in this layer, the pairwise SVMs classifier is computed and the result of the classification (the winning class) assigned to the output node. The same procedure is repeated for the next layer until the top node is reached. At the top node only two classes remain and the final classification is determined by the SVM corresponding to these two classes. We call this classification scheme bottom-up decision tree as the classification is carried out form the bottom of the tree to the top.
  As an example, consider the tree of Fig. 4(a). This tree has two layers. There are two nodes in the bottom layer and one node in the first layer. In the bottom nodes, the A/B and C/D SVMs classifiers are evaluated to classify an input vector. In the top node, the SVM trained to distinguish between the two winning classes is evaluated. For example, if A and D win in the bottom layer, the A/D SVM is evaluated in the top node.
  This scheme is an extension of the so called "tennis tournaments" discussed in Ref. [18]. There, the number of classes was restricted to be a power of two and only fully symmetric trees were considered, i.e. the depth of the tree was chosen to be $\log_2 q$.
- *Top-down decision graph*
  This scheme was recently proposed in Ref. [19]. The system architecture is based on the so called direct acyclic graph. This graph has the shape of a triangle, with $q-1$ layers. The jth layer has j nodes, each with two edges. For each layer except the last one, the ith node is connected to the ith and $(i+1)$th nodes of the next layer. The first layer contains only one node, which is the root node of the graph. The number of nodes is equal to $q(q-1)/2$ and each node is associated to a pairwise SVM classifier as following: (i) An ordered pair of classes at the root node is selected. (ii) If (a,b) is the ordered pair of classes assigned to the ith node of the jth layer, the ith and $(i+1)$th nodes in the $(j+1)$th layer will have pairwise classifier $(a,\cdot)$ and $(\cdot,b)$, respectively.
  Considering a four-class example, each node in the decision graph in Fig. 4(b) represents a pairwise SVM. Classification of an input vector starts from the root node of the graph and follows the decision path along the graph. For example, if the A/D SVM in the root node of the graph in Fig. 4(b) classifies the input as belonging to class A, the node is exited via the left edge.

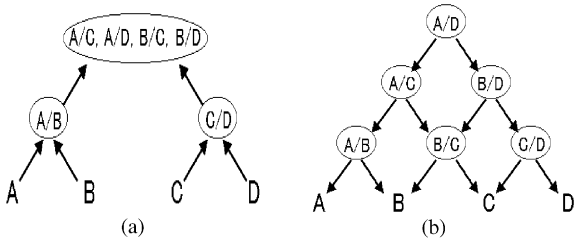

Fig. 4. (a) Bottom-up decision tree; (b) Top-down decision graph.

Notice that at run-time all three strategies require the evaluation of about the same number of classifiers (q for the pairwise schemes and $q-1$ for the one-vs-all scheme). However, we expect the computation time to be much shorter when pairwise classifiers are used, since in this case fewer support vectors are expected. Furthermore, the training time is significantly faster in the pairwise case as each classifier is trained only on a small subset of the data. There is no theoretical analysis of the schemes with respect to classification performance. Based on the previous observations and experiments, the pairwise scheme seems to be more user-friendly than the one-vs-all.

[3] This notation is an abbreviation for "one versus all the remaining".

(a) (b)

Fig. 5. (a) Examples of the four people in the frontal pose; (b) Examples of the four poses for one person.

4. Experiments

In this section we report on two different sets of experiments. In our experimental setup a color camera recorded Lab members in our main office while they were using a coffee machine. The camera was located in front of the coffee machine at a distance of about 15 feet. Images were recorded at a fixed focus, background and lighting were almost invariant.

In the first experiment, we evaluated the use of different sets of image features and different types of classifiers (multi-class SVMs and k-NNs). The task in the first experiment was to distinguish four different persons and to recognize their poses using recordings of one day. In the second experiment, we chose the best features as determined in the first experiment and increased the number of persons to be recognized to eight and the time span of our recordings to 16 days.

4.1. Person recognition and pose estimation

In this experiment the system was trained to recognize four persons and to estimate their poses (front, back, left and right). Training and test images were recorded during one day. Example images of the four persons are shown in Fig. 5(a); example images of a person in four poses are shown in Fig. 5(b). We used 640 images to train the system, 40 for each person at each pose. First, we trained a multi-class classifier to identify a person. The training set contained 160 images per person, 40 per pose. Then, multi-class pose classifiers were trained for each person separately. To summarize, five multi-class classifiers were trained, one for recognizing persons and four for pose estimation. The system first recognized the person and then selected the proper multi-class classifier to determine the pose of the person.

Fig. 6 shows an example of the output of the system. The upper left corner shows the name of the recognized person, the lower left corner shows the estimated pose. The white boxes in the center of the images are the results of the detection module. Table 1 shows the people identification rates for different types of features and different types of classifiers including three versions of multi-class SVMs and a k-NN classifier.[4] Table 2 shows the pose estimation rates. These rates were computed on a test set of 418 images containing approximatively the same number of people/pose images. For both tasks, person identification and pose estimation, the best results were obtained with normalized color features (1024 dimension). The three types of SVM classifiers showed similar recognition rates, which were slightly better than the recognition rates achieved by k-NN classifiers. Notice that recognition rates for poses are lower than that for identification. People can be easily distinguished based on their clothes. Pose estimation is a more difficult because of the similarity between right/left poses and front/back poses. We expected global shape features based on row and column histograms to be helpful for pose estimation. However, the performance decreased when adding row and column histograms to the input features. This is because of arm movements of people and varying distances between people and camera that lead to significant changes in the shape and size of the body silhouettes. On the other hand, local shape features performed well for both: person recognition and pose estimation.

4.2. Increasing the data set

In the second experiment we repeated the previous experiment on a data set containing images of eight persons recorded over several days. About 3500 images were recorded during 16 days. From the 3500 images we selected 1127 images belonging to the eight most frequent users of the coffee machine. Some example images[5] of these eight persons are shown in Fig. 7. The images were represented by their normalized color histograms. We chose these features because they showed the best performance in the first experiment.

[4] In case of a tie, the system chose the class whose nearest neighbors has minimum average distance from the input.

[5] The complete dataset can be downloaded from the CBCL Homepage at http://www.ai.mit.edu/projects/cbcl.

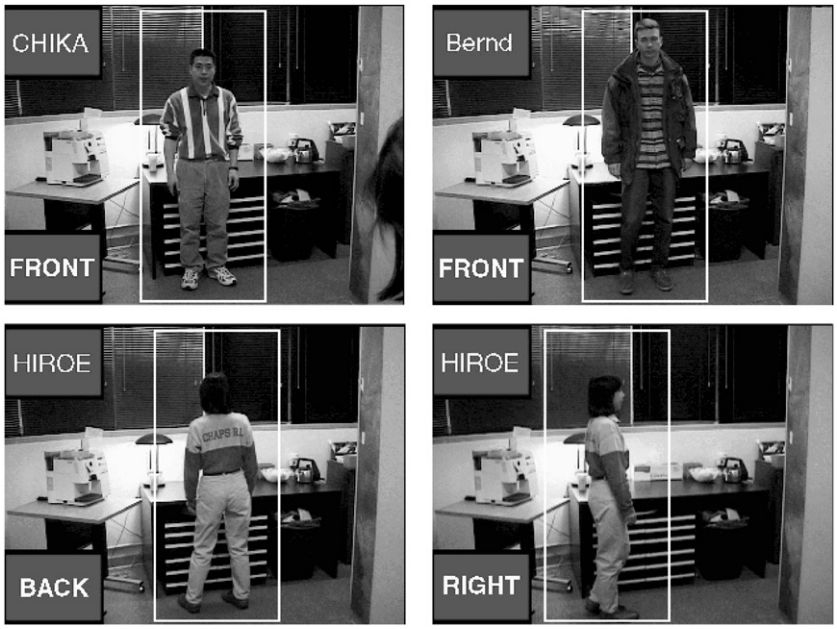

Fig. 6. Examples of people identification and pose estimation results.

Table 1
Person recognition and pose estimation rates from the test set of the four persons

Features	SVMs			k-Nearest Neighbor		
	Top-down	Bottom-up	One-vs-all	$k=1$	$k=3$	$k=5$
RGB (96)	99.5	99.2	99.5	99.0	98.7	98.5
Norm. RGB (1024)	100	100	100	100	100	100
RGB+Shape (136)	91.4	91.6	96.2	94.7	94.4	94.1
Local Shape (75)	99.5	99.5	97.5	88.3	85.0	84.8

Table 2
Pose estimation rates from the test set of the four persons

Features	SVMs			k-Nearest Neighbor		
	Top-down	Bottom-up	One-vs-all	$k=1$	$k=3$	$k=5$
RGB (96)	74.9	75.9	83.8	70.1	70.6	72.3
Norm. RGB (1024)	86.5	86.3	87.8	85.5	85.8	86.0
RGB+ Shape (136)	68.0	68.2	70.1	67.8	66.8	65.7
Local Shape (75)	84.5	84.3	84.0	82.0	82.7	82.0

We performed five different sets of experiments where the system was trained to recognize the eight persons. In the first four experiments we used about 90%, 80%, 50%, and 20% of the image database for training. The remaining images were used for testing. In the fifth experiment the training set consisted of all images recorded during the first 15 days, the test set included all images recorded during the last day.

Recognition rates are shown in Table 3. The system performed well when the training set contained images from all 16 days (first four experiments). The recognition rate

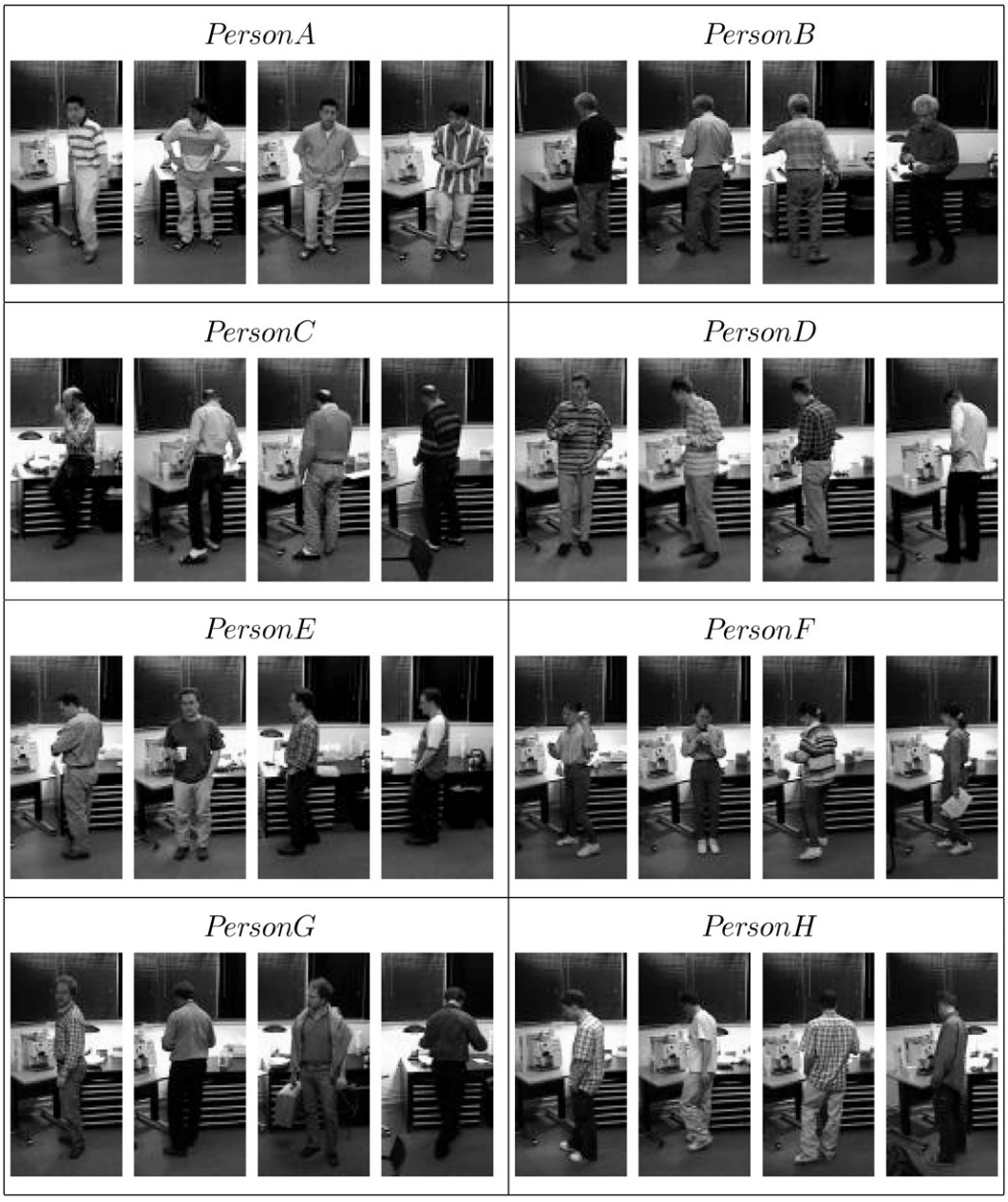

Fig. 7. Image examples of the eight people recorded during different days used in the experiments in Section 4.

decreased to about 50% when the system was tested on images recorded during a new day (last experiment). This is because people wore different clothes every day, so that the system was not able to recognize them based on the color of their clothes only. Notice that this rate is still significantly better than chance (12.5%). Overall k-NN was slightly better than linear SVMs. Preliminary tests with SVMs with second degree polynomial showed a significantly better performance than k-NN; the best recognition rate is clearly achieved with second degree polynomial SVMs (see Table 3). However, due to the off of the computational complexity, we are using the linear kernel for real-time recognition.

5. Discussion

In this section we discuss a few issues arising in designing multi-class SVMs and outline future extensions of the proposed system.

Table 3
People recognition rates for eight people. The using feature is the normalized color features: 1024 dimension

	(test : training)	1:9 (113:1014)	1:5 (188:939)	1:1 (564:563)	5:1 (939:188)	New Day (122:1005)
	Top-down	92.3	91.2	90.5	73.3	45.9
	Bottom-up	90.6	91.7	90.6	66.1	45.9
	One-vs-all	87.2	90.6	85.9	84.6	49.2
SVM	One-vs-all (Polynomial)	98.3	96.4	94.7	88.1	52.9
	$k = 1$	92.9	92.0	92.7	85.1	53.3
kNN	$k = 3$	92.9	92.0	92.2	81.3	50.0
	$k = 5$	94.7	91.0	90.1	76.0	50.8

5.1. More about multi-class

There are two observations about designing multi-class classifiers which we want to discuss.

First, note that both in the one-vs-all and in the pairwise approach, each classifier is trained separately from the others, i.e. classifiers are computed by solving separate optimization problems. Earlier work [32] has attempted to formulate the one-vs-all SVM scheme as a single optimization problem. However, their approach appears to be both slower and less effective than the standard one-vs-all method used here. Very recent work has extended the concept of margin to the case of multi-class classifiers and a new algorithm for globally training the one-vs-all SVM scheme was proposed [33]. In future work we will explore the advantage of this approach on our database. At the same time it would be interesting to explore algorithms as the one in Ref. [33] for training the pairwise SVM classifiers as well.

The second observation is that classification schemes based on training one-vs-all and pairwise classifiers are two extreme approaches: the first uses all the data, the second the smallest portion of the data. It would be interesting to study intermediate classification strategies in the style of error-correcting codes (ECC) [34,35]. In this case, each classifier is trained to separate a subset of classes from another disjoint subset of classes (the union of these two subsets does not need to cover all the classes). The classification works as follows: If a subset is selected by a classifier all classes within the subset get a vote. In the end the class with the most votes is the winner. [6]

ECC seems to be effective when the classes are based on some common features each of which is "active" on a different subset of the classes. In our case, each machine could be based on different color/shape features. See also Ref. [36] for related considerations.

[6] In the case of a tie, the winner class is picked randomly among the tied class.

Finally, it will be also interesting in the future to investigate other simple classification tools such as Fisher discriminant analysis and their variations (see, e.g., [37]), which could be potentially useful for this task. However, it is clear that in order to obtain major improvements one needs to change the system design which is what we briefly discuss next.

5.2. System extensions

The proposed system showed high performance rates for person recognition when both the training and test images were recorded during the same days. When the test set contained images of a day not represented in the training set the performance of the system drops down to about 53%. This is the main limitation of the current system. Since clothes of people usually change every day we have to add more invariant features to extend the capabilities of the system. These features could be extracted from the face regions in the images. Unfortunately the image resolution in our current system is too low for face recognition. One possibility would be to add a second camera which takes a higher resolution image of the persons face. Then we could perform recognition of a person by combining body and the face classification. Future research will investigate these problems.

6. Conclusion

We have presented a system that recognizes full-body persons in a constrained environment. Person recognition was performed by multi-class SVMs that were trained on color images of people. The images were represented by different sets of color and shape-based features. The proposed system works in real-time using linear SVMs, achieving high recognition rate on normalized color histograms of peoples' clothes. When non-linear SVMs are used the system performs significantly improves over k-NN classifiers. These results indicate the relevance of the

presented method for automatic daily surveillance in indoor environments.

References

[1] B. Heisele, T. Poggio, M. Pontil, Face detection in still gray images, MIT AI Memo 1687, 2000.

[2] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Machine Intelligence 20 (1) (1998) 23–38.

[3] H. Schneiderman, T. Kanade, A statistical method for 3d object detection applied to faces and cars, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 746–751.

[4] K. Sung, T. Poggio, Example-based learning for view-based human face detection, MIT AI Memo 1521, 1994.

[5] R. Brunelli, T. Poggio, Face recognition: Features versus templates, IEEE Trans. Pattern Anal. Machine Intelligence 15 (10) (1993) 1042–1052.

[6] A. Pentland, T. Choudhury, Face recognition for smart environments, Computer 33 (2) (2000) 50–55.

[7] L. Wiskott, J.M. Fellous, N. Kruger, C. von der Malsburg, Face recognition by elastic bunch graph matching, IEEE Trans. Pattern Anal. Machine Intelligence 19 (7) (1997) 775–779.

[8] C. Papageorgiou, T. Poggio, A trainable object detection system: car detection in static images, MIT AI Memo 1673, 1999.

[9] D. Gavrila, Pedestrian detection from a moving vehicle, Computer Vision: 3rd European Conference on Computer Vision, 2000, pp. 37–49.

[10] R. Cutler, L. Davis, Robust periodic motion and motion symmetry detection, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2000, pp. 615–622.

[11] B. Heisele, C. Woehler, Motion-based recognition of pedestrians, Proceedings of International Conference on Pattern Recognition and Image Processing, 1998, pp. 1325–1330.

[12] M.S. Nixon, J.N. Carter, J.M. Nash, P.S. Huang, D. Cunado, S.V. Stevenag, Automatic gait recognition, IEE Colloq. Motion Anal. Track. 3 (1999) 1–6.

[13] C. Nakajima, M. Pontil, T. Poggio, People recognition and pose estimation in image sequences, Proceedings of International Joint Conference on Neural Networks, Vol. 4, 2000, pp. 189–194.

[14] C. Cortes, V. Vapnik, Support vector networks, Machine Learn. 20 (1995) 273–297.

[15] V. Vapnik, Statistical Learning Theory, Wiley & sons, Inc., New York, 1998.

[16] C. Papageorgiou, T. Poggio, Trainable pedestrian detection, Proceedings of IEEE International Conference on Image Processing, Vol. 4, 1999, pp. 35–39.

[17] A. Mohan, Object detection in images by components, MIT AI Memo 1664, 1999.

[18] M. Pontil, A. Verri, Support vector machines for 3-d object recognition, IEEE Transactions on Pattern Analysis Machine Intelligence 20 (6) (1998) 637–646.

[19] J. Platt, N. Cristianini, J. Shawe-Taylor, Large margin dags for multiclass classification, Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2000, pp. 547–553.

[20] O. Chapelle, P. Haffner, V. Vapnik, Support vector machines for histogram-based image classification, IEEE Transactions on Neural Networks 10 (5) (1999) 1055–1064.

[21] A. Elisseeff, Y. Guermeur, H. Paugam-Moisy, Margin error and generalization capabilities of multi-class discriminant systems, Technical Report NC2-TR-1999-051, NeuroCOLT2 Technical Report, 1999.

[22] T. Kurita, K. Hotta, T. Mishima, Scale and rotation invariant recognition method using higher-order local autocorrelation features of log-polar image, Proceedings of IEEE Asian Conference on Computer Vision, Vol. 2, 1998, pp. 89–96.

[23] M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex, Nature Neurosci. 2 (1999) 1019–1025.

[24] K. Uchikawa, Mechanism of color perception, Asakura syoten, 1998.

[25] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, Proceedings of IEEE Workshop on Neural Networks and Signal Processing, 1997, pp. 276–285.

[26] P. Niyogi, C. Burges, P. Ramesh, Distinctive feature detection using support vector machines, Proceedings of International Conference on Acoustic, Speech, and Signal Processing, 1999, pp. 425–428.

[27] T. Joachims, Text categorization with support vector machines, Technical Report LS-8 Report 23, University of Dortmund, 1997.

[28] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Comput. Math. 13 (2000) pp. 1–50.

[29] B. Schölkopf, C. Burges, V. Vapnik, Extracting support data for a given task, Proceedings of the First International Conference on Knowledge Discovery and Data Mining, 1995, pp. 252–257.

[30] P.R. Clarkson, P.J. Moreno, On the use of support vector machines for phonetic classification, Proceedings of IEEE International Conference on Speech and Signal Processing, Vol. 2, 1999, pp. 585–588.

[31] Jerome H. Friedman, Another approach to polychotomous classification, Technical report, Department of Statistics, Stanford University, 1996.

[32] J. Weston, C. Watkins, Multi-class support vector machines, Technical Report CSD-TR-98-04, Royal Holloway, University of London, 1998.

[33] Y. Guermeur, A. Elisseeff, H. Paugam-Mousy, A new multi-class svm based on a uniform convergence result, Proceedings of International Joint Conference on Neural Networks, Vol. 4, 2000, pp. 183–188.

[34] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artificial Intelligence Res. 2 (1995) 263–286.

[35] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, J. Machine Learn. Res. 1 (2000) 113–141.

[36] Y. Yao, G. Marcialis, M. Pontil, P. Frasconi, F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, Pattern Recognition 36 (2) (2003) 397–406.

[37] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, Springer Series in Statistics, Springer, Berlin, 2001.

About the Author—CHIKAHITO NAKAJIMA is a researcher at the Central Research Institute of Electric Power Industry (CRIEPI), Tokyo, Japan. He owns a Master degree in Electrical Engineering from the Tokyo Denki University and joined CRIEPI in 1989. He was a visiting scientist at the Center for Biological and Computational Learning at MIT in 1999 and 2000. His research interest involves visual surveillance and human-computer interaction.

About the Author—MASSIMILIANO PONTIL received his Bachelor and Ph.D. in Physics from the University of Genova in 1994 and 1999. He is currently a Lecturer in the Department of Computer Science and Honorary Lecturer in the Gatsby Unit at the University College London. His research interests involve learning theory and its relation to pattern recognition, statistics, and approximation theory. Previously, he has been a visiting student at MIT in 1998, a Post-doctoral Fellow in 1999 and 2000 at the Center for Biological and Computational Learning at MIT, a Research Fellow at City University of Hong Kong in 2001 and 2002, and a Research Associate in the Department of Information Engineering of the University of Siena in 2001 and 2002. He has also spent some time as a visiting researcher at the RIKEN Brain Science Institute, Tokyo, and AT& T Labs, USA. He has published more than 50 papers in international journals and conferences on different aspects of learning theory, machine learning, and computer vision.

About the Author—BERND HEISELE received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Stuttgart, Stuttgart, Germany, in 1993 and 1999, respectively. In 1999 he was awarded a postdoctoral fellowship by the DFG in Germany. From 1999 to 2001 he worked as a postdoctoral researcher at the Center for Biological and Computational Learning, Massachusetts Institute of Technology, Cambridge. He subsequently joined Honda and is currently heading the Honda Research Laboratory in Cambridge where he is conducting research in computer vision. His research interests are learning-based object detection/recognition and motion analysis in image sequences.

About the Author—TOMASO POGGIO Ph.D. is an Uncas and Helen Whitaker Professor of Vision Sciences and Biophysics, in the Department of Brain and Cognitive Sciences; Co-Director, Center for Biological and Computational Learning; Member for the last 20 years of the Artificial Intelligence Laboratory at MIT; and, since 2000, member of the faculty of the McGovern Institute for Brain Research. Earlier Prof. Poggio had worked on the visual system of the fly with W. Reichardt in Tuebingen at the Max Planck Institut fuer Biologische Kybernetik and with D. Marr on computational analysis of human and machine vision. He was responsible for the Vision Machine project at the AI Lab. Serving on the editorial boards of a number of leading interdisciplinary journals, Professor Poggio is a Founding Fellow of the American Association for Artificial Intelligence, an Honorary Associate of the Neuroscience Research Program at Rockefeller University and a member of several scientific and engineering associations including IEEE, AAAS; he is an elected member of the American Academy of Arts and Sciences. Professor Poggio received his doctorate in theoretical physics from the University of Genoa in 1970, had a tenured research position at the Max Planck Institute from 1971 to 1981 when he became Professor at MIT. He is the author of hundreds of papers in areas ranging from biophysics to information processing in man and machine, artificial intelligence, machine vision and learning. A former Corporate Fellow of Thinking Machines Corporation, he was and is still peripherally involved in several companies in the areas of bioinformatics, computer graphics, computer vision, computer networks, and financial engineering.