Appears in Proceedings of Intelligent Vehicles, Stuttgart, Germany, October 1998

A Trainable Pedestrian Detection System

Constantine Papageorgiou Theodoros Evgeniou Tomaso Poggio

Center for Biological and Computational Learning and Artificial Intelligence Laboratory
MIT
Cambridge, MA 02139

Abstract

In the near future, we can expect on-board automotive vision systems that inform or alert the driver about pedestrians, track surrounding vehicles, and read street signs. Object detection is fundamental to the success of this type of next-generation vision system. In this paper, we present a trainable *object detection system that automatically learns to detect objects of a certain class in unconstrained scenes. We apply our system to the task of pedestrian detection. Unlike previous approaches to pedestrian detection that rely heavily on hand-crafted models and motion information, our system learns the pedestrian model from examples and uses no motion cues. The system can easily be extended to include motion information. We review our previous system, describe a new system that exhibits significantly better performance, provide a comparison between using different combinations of feature sets with classifiers of varying complexity, and describe improvements that increase the system's processing speed by two orders of magnitude.*

1 Introduction

The possible applications of object detection research to practical problems are significant. The impact of this technology in automotive systems could be especially great. We have developed a general, trainable object detection system that can successfully find pedestrians in complex images, without assuming any *a priori* scene structure [16] [15] [17]. The technique is founded on a representation that encodes local intensity differences. Our system learns from examples, meaning that the object model is derived from a set of training images of pedestrians. To enable the architecture to detect a different class of objects, we can change the training set of object examples; this general architecture has successfully been applied to both pedestrian and face detection in cluttered scenes.

Most previous systems designed to detect pedestrians in video sequences have relied heavily on hand-crafted models and/or motion information. Hand-crafted models limit system portability and the assumption that all pedestrians are moving is clearly restrictive for a general system. Discussions of these systems can be found in [9] [19] [12] [11] [18] [1] [6] [21] [14] [7] [8].

Our system is based on a novel object representation that uses projections of the object images onto a dense Haar wavelet basis that efficiently encodes structural features at different scales. Each wavelet coefficient corresponds to a single feature, or basis function. The set of features we use contains two scales of basis functions with 1326 components. These features are used to train a Support Vector Machine (SVM) classifier [20], which provides bounds on the generalization error. Using this representation, we can reliably detect pedestrians in static images with no motion information; extensions for processing video sequences can use motion as an additional cue.

In this paper, we build on our previous work and introduce a new system that exhibits significantly better performance, provide a comparison of the performance obtained by using different combinations of feature sets with classifiers of varying complexity, and describe improvements that can increase the processing speed of our system by two orders of magnitude.

This paper is organized as follows. Section 2 describes our base pedestrian detection system and the SVM technique. Section 3 presents several optimizations to the system that are designed to reduce processing time so that the system may eventually be near real-time. In Section 4, we show the results of our system. Section 5 summarizes our work and presents areas of future research.

2 System Architecture

This section presents an overview of our pedestrian detection system; for more in-depth discussions of our system, see [16] [15] [17].

2.1 Wavelet Features

One of the key issues in developing an object detection system is what object representation to use. The ultimate goal is a representation that yields high inter-class variability and low intra-class variability. Figure 1 shows several

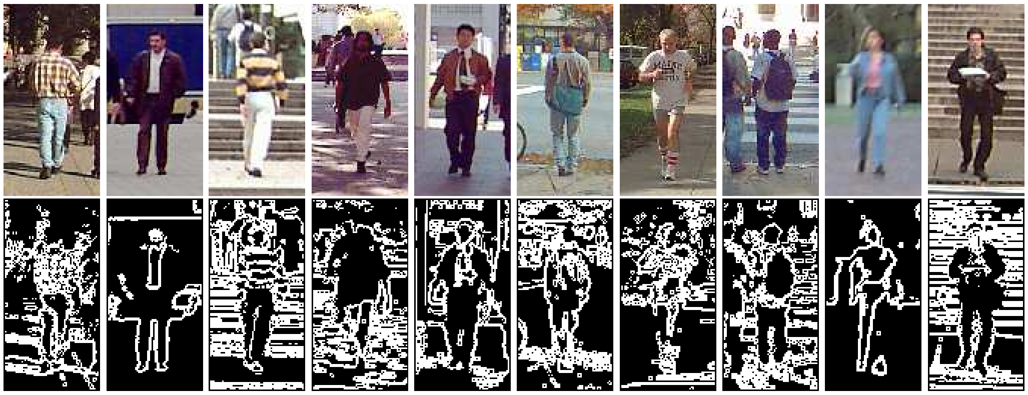

Figure 1. The top row shows examples of images of pedestrians in the training database. The examples vary greatly in color, texture, and background. The bottom row shows the corresponding edge maps generated by a sobel filter; fine-scale edge information does not characterize the pedestrian class well.

example images of pedestrians from our training set. From these images, it is clear that a pixel-based representation would likely fail on account of the high degree of variability in the color of the pedestrian patterns. A traditional fine-scale edge-based representation is also not adequate; Figure 1 clearly shows how the results of this type of processing yields edge maps with little consistency between the patterns and a lot of spurious information. Region-based approaches that use color, for instance, would have the same problems as pixel-based systems due to the lack of consistent color information.

The representation we use overcomes these problems by looking at intensity differences in small local regions; essentially, finding multi-scale edges. This is implemented in a computationally efficient framework called Haar wavelets [13]. The Haar wavelet transform is run over an image and results in a set of coefficients at several scales that indicate the response of the wavelets over the entire image. Different wavelets respond to vertical, horizontal, and diagonally oriented intensity differences, so what the transform yields is three sets of coefficients, one for each of the wavelet orientations.

Since we are using color images, we run the Haar transform over each color channel separately. Then, for a given spatial location and orientation, we choose as the wavelet coefficient the one from the three color channels that gives the maximal response. In this way, we expect that strong visual intensity differences will be accurately reflected in the representation.

2.2 Feature Selection

Our training database is a set of 1848 frontal and rear color views of pedestrians (924 plus mirror images) that have been scaled to 128 $\times$ 64 pixels and aligned such that the pedestrian is in the center of the image. Using the Haar wavelet representation, we look at both coarse-scale (32 $\times$ 32 pixels) and fine-scale (16 $\times$ 16 pixels) features. At these scales of wavelets, there are 1326 total features for a 128 $\times$ 64 pattern.

Many of these coefficients will be irrelevant for the task of pedestrian detection, for instance, the coefficients at the corner areas that do not touch the pedestrian's body. We can obtain a more compact representation by choosing a small subset of the coefficients that are consistently strong or weak across the ensemble of training patterns. Average responses for each coefficient for the three orientations and the two scales are shown in Figure 2 where a dark coefficient indicates consistently strong response or the presence of an edge and a light coefficient indicates consistent weak response, or a uniform region.

Once this statistical analysis is done, we manually choose 29 of these coefficients that will be the features of the pedestrian class, so each pedestrian image is represented by a 29-dimensional feature vector.

2.3 Support Vector Machine Classification

Using our 1848 pedestrian patterns and a set of 7189 negative patterns gathered from images of outdoor scenes not containing pedestrians, we can train a classifier to differentiate between pedestrian and non-pedestrian patterns. To this end, we use a Support Vector Machine classifier.

Support Vector Machines is a training technique which, instead of minimizing the training error of a classifier, minimizes an upper bound on its generalization error. This technique has recently received a great deal attention and has been applied to areas such as handwritten character recognition [3], 3D object recognition [2], text categorization [10], and object detection [16] [15] [17]. The appealing characteristics of SVMs are a) by choosing different kernel functions, we can implement various classifiers like polynomial classifiers, multilayer perceptrons, and radial basis functions, b) the only tunable parameter is a penalty term for misclassifications, and c) as mentioned before, the algorithm finds the separating decision surface that should

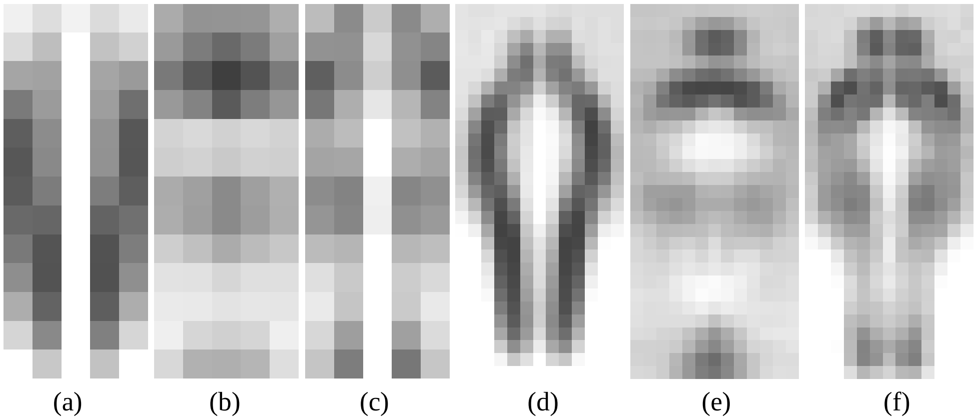

Figure 2. Ensemble average values of the wavelet coefficients coded using gray level. Coefficients whose values are above the average are darker, those below the average are lighter. (a)-(c) vertical, horizontal, and diagonal coefficients at scale 32×32 of images of pedestrians, (d)-(f) vertical, horizontal, and diagonal coefficients at scale 16×16 of images of pedestrians.

provide the best out-of-sample performance. The SVM decision surface is obtained by solving a quadratic programming problem; for more details on the algorithm, see [20] [5].

2.4 Detecting Pedestrians

To detect pedestrians in a new image, we shift the 128×64 detection window over all locations in the image. This will only detect pedestrians at a single scale, however. To achieve multi-scale detection, we incrementally resize the image and run the detection window over each of these resized images.

3 Optimizations

The original system we have described for pedestrian detection in color images initially processed at a rate of 1 frame per 2 minutes; this was clearly inadequate for any real-time automotive applications. We have implemented several optimizations that have yielded two orders of magnitude worth of speedups.

3.1 Reduced Set Vectors

One of the shortcomings of SVM classification is that the computation involved in classifying a pattern can be immense, when compared to other classification approaches with similar performance. For each pattern that the system classifies, the following equation that encodes the decision surface must be evaluated for our case where we use a polynomial classifier of degree two:

$$f(\mathbf{x}) = \theta\left(\sum_{i=1}^{N_s} \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i + 1)^2 + b\right) \qquad (1)$$

where $\mathbf{x}$ is the pattern to classify, i indexes into the N_s *support vectors* $\mathbf{x}_i$, that is, data points that are part of the solution, y_i indicates the class ($\{+1, -1\}$) of example i, and α_i is a Lagrange parameter.

For our 29-dimensional feature vectors, each classification entails $N_s \times 29$ multiplications (excluding the Lagrange and class multiplications). For our color system, we obtain 331 support vectors meaning that a single classification needs over 9500 multiplications.

To overcome this computational hurdle, we use a reduced set method [4] to obtain an equivalent decision surface in terms of a small number of synthetic vectors. This method yields a new decision surface that is equivalent to the original one but uses just 29 vectors, thus, 841 multiplications per classification, a significant reduction in computational overhead from the original version. We take advantage of an exact solution by changing the kernel from $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2 + 1$ to the homogeneous polynomial of degree two, $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2$. The homogeneous kernel results in a kernel with less representative power than the kernel with the $+1$ but our results indicate that it is a viable alternative.

3.2 Grey Level Processing

Our use of color images is predicated on the fact that the three different color channels (RGB) contain a significant amount of information that gets washed out in grey level images of the same scene. This use of color information results in significant computational cost; the resizing and Haar transform operations are performed on each color channel separately. In order in improve processing speed, we modify the system to process intensity images. By comparing the performance to the color systems, we can analyze exactly how much degradation in performance this results in.

For the grey-level version, we use the same 29 wavelets

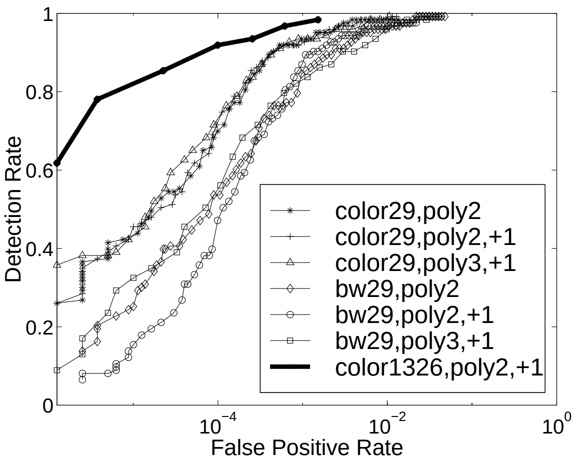

Figure 3. ROC curves for different detection systems. The detection rate is plotted against the false positive rate, measured on a logarithmic scale. The false detection rate is defined as the number of false detections per inspected window.

that are used in the color version. What this means is that we are doing feature selection for our grey-level system using color images. In the future, the grey-level system's features should be derived from the grey-level training set.

4 Experimental Results

To gauge the performance of a detection system, it is necessary to analyze the full ROC curve which gives an indication of the tradeoff between accuracy and the number of false positives. We emphasize that our ROC curves are computed over an *out-of-sample* test set gathered around MIT and over the Internet. Figure 3 compares the ROC curves of several different incarnations of our system. They are as follows:

- color processing with 29 features using a homogeneous polynomial of degree two (to take advantage of the reduced set method)
- color processing with 29 features using a polynomial of degree two
- color processing with 29 features using a polynomial of degree three
- grey-level processing with 29 features using a homogeneous polynomial of degree two (to take advantage of the reduced set method)
- grey-level processing with 29 features using a polynomial of degree two
- grey-level processing with 29 features using a polynomial of degree three
- color processing with all 1326 features using a polynomial of degree two

From the ROC curve, it is clear that most of the impact on performance comes from what features are used; the complexity of the classifier is secondary. As expected, using color features results in a more powerful system. The curve of the system with *no feature selection* is clearly superior to all the others. This indicates that for the best accuracy, using all the features is optimal. When classifying using this full set of features, we pay for the accuracy through a slower system. It may be possible to achieve the same performance as the 1326 feature system with fewer features; this is an open question, however. Examples of processed images are shown in Figure 4; these images were not part of the training set.

We can also extend the system to allow it to detect frontal, rear, and side views of pedestrians. This more complete system is trained on a set of 3600 positive and 12437 negative examples, using all 1326 features. Figure 5 shows the results of processing a video sequence from downtown Ulm, Germany without using any motion or tracking information; adding this information to the system would improve results. From the sequence, we can see that our system generalizes extremely well; this test sequence was gathered with a different camera, in a different location, and in different lighting conditions than our training data.

5 Conclusion

This paper presents a description of our framework for object detection, as applied to the task of pedestrian detection. Our system is based on obtaining a model of pedestrians using a small set of local wavelet features derived from an ensemble of training examples. We highlight the differences in using color and grey-level features and compare the use of polynomial classifiers of different complexities. We also present a new super-accurate detection system in which there is no feature selection step. From these experiments, we conclude that the features themselves contribute most to performance; classifier complexity is secondary.

Pedestrian detection has many possible applications in the areas of automotive assistance systems, image and video database indexing, and surveillance. It is our belief that trainable techniques like those presented here will become increasingly important in developing such systems. For practical applications, processing speed becomes critical; in this paper, we have described some system optimizations – grey-level processing and a reduced set method – that have been implemented to reduce processing time by two orders of magnitude.

6 Acknowledgments

Support for this work has been provided by DARPA, ONR, Daimler-Benz, Siemens, and Kodak. Portions of

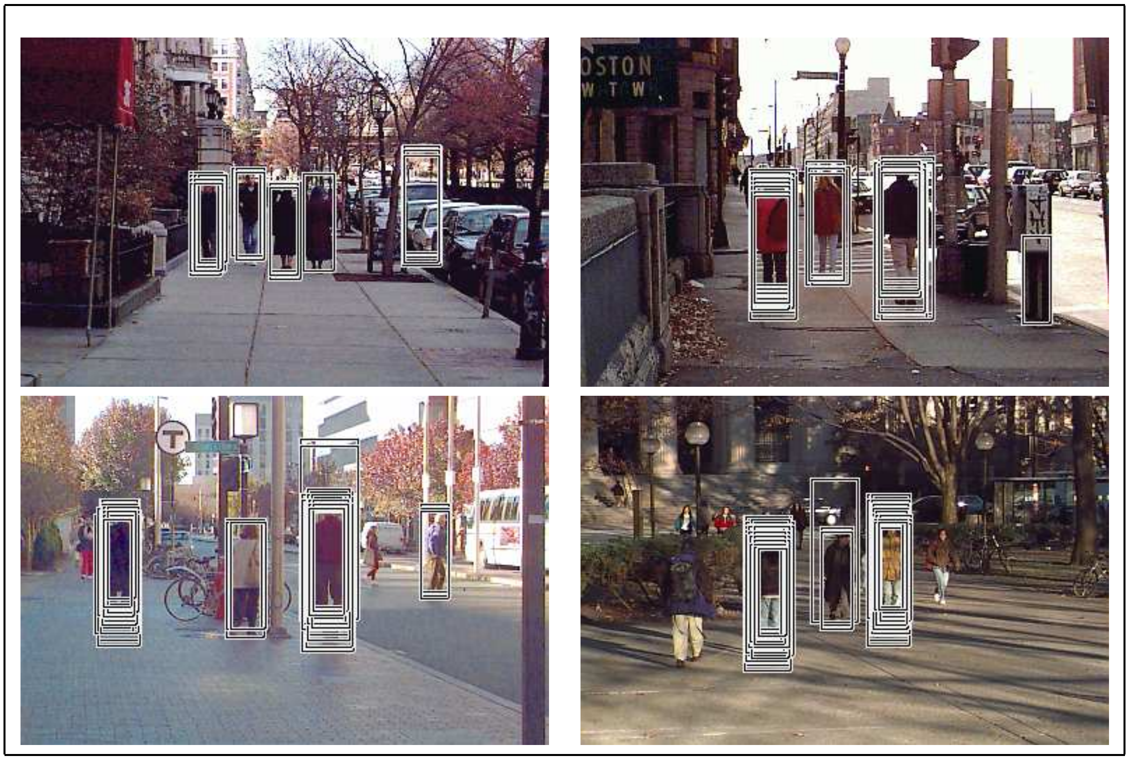

Figure 4. Results from our pedestrian detection system. Typically, missed pedestrians are due to occlusion or lack of contrast with the background. False positives can be eliminated with further training.

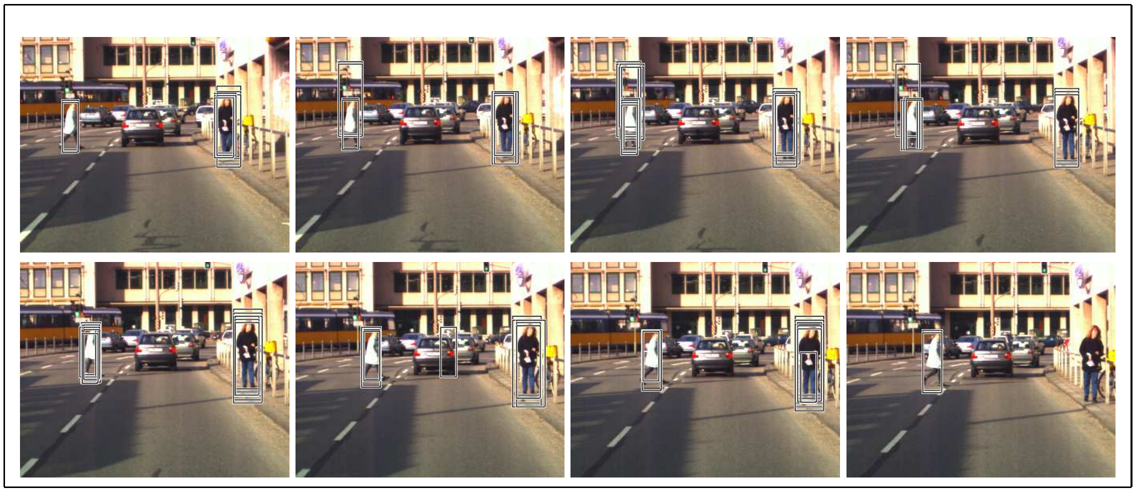

Figure 5. Processing the "Downtown Ulm" sequence with our frontal, rear, and side view detection system. The system uses no motion or tracking; adding this information to the system would improve results.

this work were done at Daimler-Benz, Ulm, Germany. The "Downtown Ulm" sequence was provided by Daimler-Benz.

References

[1] A. Baumberg and D. Hogg. An efficient method for contour tracking using active shape models. In *Proceedings of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 194–199, 1994.

[2] V. Blanz, B. Scholkopf, H. Buelthoff, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition using realistic 3d models. In C. von der Malsburg, W. von Seelen, J. Vorbruggen, and B. Sendhoff, editors, *Artificial Neural Networks – ICANN'96*, pages 251–256, 1996.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifier. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–52. ACM, 1992.

[4] C. Burges. Simplified Support Vector decision rules. In *Proceedings of 13th International Conference on Machine Learning*, 1996.

[5] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. In U. Fayyad, editor, *Proceedings of Data Mining and Knowledge Discovery*, pages 1–43, 1998.

[6] H.-J. Chen and Y. Shirai. Detecting multiple image motions by exploiting temporal coherence of apparent motion. *Computer Vision and Pattern Recognition*, pages 899–902, 1994.

[7] B. Heisele, U. Kressel, and W. Ritter. Tracking Non-rigid, Moving Objects Based on Color Cluster Flow. In *Computer Vision and Pattern Recognition*, 1997.

[8] B. Heisele and C. Wohler. Motion-Based Recognition of Pedestrians. In *Proceedings of International Conference on Pattern Recognition*, 1998. (in press).

[9] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[10] T. Joachims. Text Categorization with Support Vector Machines. Technical Report LS-8 Report 23, University of Dortmund, November 1997.

[11] M. Leung and Y.-H. Yang. Human body motion segmentation in a complex scene. *Pattern Recognition*, 20(1):55–64, 1987.

[12] M. Leung and Y.-H. Yang. A region based approach for human body analysis. *Pattern Recognition*, 20(3):321–39, 1987.

[13] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–93, July 1989.

[14] S. McKenna and S. Gong. Non-intrusive person authentication for access control by visual tracking and face recognition. In J. Bigun, G. Chollet, and G. Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, pages 177–183. IAPR, Springer, 1997.

[15] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition*, pages 193–99, 1997.

[16] C. Papageorgiou. Object and Pattern Detection in Video Sequences. Master's thesis, MIT, 1997.

[17] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of International Conference on Computer Vision*, 1998.

[18] K. Rohr. Incremental recognition of pedestrians from image sequences. *Computer Vision and Pattern Recognition*, pages 8–13, 1993.

[19] T. Tsukiyama and Y. Shirai. Detection of the movements of persons from a sparse sequence of tv images. *Pattern Recognition*, 18(3/4):207–13, 1985.

[20] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[21] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. Technical Report 353, MIT Media Laboratory, 1995.