Trainable Pedestrian Detection

Constantine Papageorgiou Tomaso Poggio

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

Abstract

Robust, fast object detection systems are critical to the success of next-generation automotive vision systems. An important criteria is that the detection system be easily configurable to a new domain or environment. In this paper, we present work on a general object detection system that can be trained to detect different types of objects; we will focus on the task of pedestrian detection. This paradigm of learning from examples allows us to avoid the need for a hand-crafted solution. Unlike many pedestrian detection systems, the core technique does not rely on motion information and makes no assumptions on the scene structure or the number of objects present. We discuss an extension to the system that takes advantage of dynamical information when processing video sequences to enhance accuracy. We also describe a real, real-time version of the system that has been integrated into a DaimlerChrysler test vehicle.

1 Introduction

The robust detection of pedestrians is a potentially important application for next-generation automotive vision systems. This paper describes a general example-based approach to object detection that has been primarily applied to detecting people. Since it is trainable, applying the system to a new domain – faces or cars, for example – simply involves plugging in a new set of training data, making the system easily portable.

One of the keys to our system's performance is the representation we use, an overcomplete dictionary of Haar wavelets. This allows the system to identify the important characteristics of the people class while ignoring noise present in the pixel-level representations. Using a large set of positive and negative examples, we train a support vector machine classifier to differentiate between people and non-people. In Section 3, we describe some comparisons across different feature classes and different feature set sizes, showing how these affect the detection performance.

To detect people in images, our core static detection system implements a brute force search. An important aspect of our system is that, unlike previous work in people detection ([16] [6] [12] [18] [4]), it does not rely on motion or tracking and makes no assumptions about the scene structure, number of people in the scene, or camera movement. Directly applying this static approach to video sequences ignores significant dynamical information. In these cases, the system can be enhanced to use this dynamical information when it is available; such an enhancement is presented in Section 5.

This paper also describes a real-time version of our pedestrian detection system as a module in the Daimler-Chrysler Urban Traffic Assistant. This integrated system achieves processing rates of 10Hz.

2 System architecture

In this section we describe our static detection system. The key components are the overcomplete Haar dictionary and the support vector machine classifier.

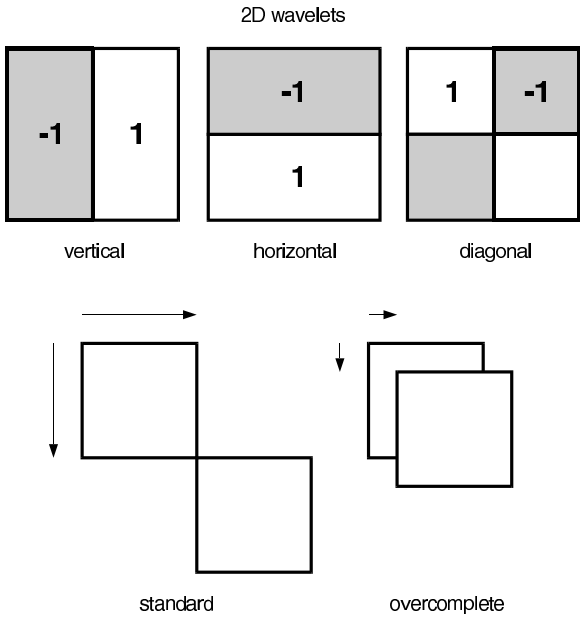
2.1 Representation

To train our system, we have gathered a set of 1,800 example color images of people that have been aligned and scaled to the dimensions 128×64. The images were taken in Cambridge and Boston during different seasons with Kodak DC50 and Sony DKC-ID1 digital cameras and a Sony DCR-VX1000 digital video camera. The images show people in many different poses (frontal, rear, side walking, side standing), under different lighting conditions, and with varied backgrounds. Some example images from our database are shown in Figure 1.

Perhaps the most important issue in the development of an object detection system is the representation of the object class. Images of people show a great deal of variability in the color, texture, and pose as well as the lack of a constant background. If we tried to learn the structure of the people class using a pixel-based representation, it is not likely to succeed due to the lack of consistency in the patterns at this level of detail. A similar argument could also be used in considering a traditional edge-based approach.

To circumvent these difficulties and provide a representation that achieves high inter-class variability with low intra-class variability, we use an overcomplete dictionary of Haar wavelets. This dictionary contains a large set of features that respond to local intensity differences at several orientations.

For a given pattern, the wavelet transform computes the responses of the wavelet filters over the image. Each of the three oriented wavelets – vertical, horizontal, and diagonal – are computed at several different scales allowing the system to represent coarse scale features all the way down



$\times$ 32 and 16 $\times$ 16. In the traditional wavelet transform, the wavelets do not overlap; they are shifted by the size of the support of the wavelet in x and y. To achieve better spatial resolution and a richer set of features, our transform shifts by $\frac{1}{4}$ of the size of the support of each wavelet, yielding an overcomplete dictionary of wavelet features. This results in a 1,326 dimensional feature vector for each pattern, which is used as training data for our classification engine. Figure 2 shows the three orientations of the Haar wavelets and the quadruple density shift. More details on wavelets and our version of the wavelet transform can be found in [7] [14] [10] [8].

There is certain *a priori* knowledge embedded in our choice of the wavelets. First, we use the absolute values of the magnitudes of the wavelets; this tells the system that a dark body on a light background and a light body on a dark background have the same information content. Second, we compute the wavelet transform for a given pattern in each of the three color channels and then, for a wavelet of a specific location and orientation, we use the one that is largest in magnitude. This allows the system to use the most visually significant features.

2.2 Support vector machine classification

Support vector machines (SVM) is a principled technique to train classifiers that is well-founded in statistical learning theory; for details, see [17] [2]. Unlike traditional training algorithms like back propagation which only minimizes training set error, one of the main attractions of using SVMs is that they minimize a bound on the empirical error and the complexity of the classifier, at the same time. In this way, they are capable of learning in *sparse, high-dimensional spaces* with relatively few training examples.

This controlling of both the training set error *and* the classifier's complexity has allowed support vector machines to be successfully applied to very high dimensional learning tasks; [5] presents results on SVMs applied to a 10,000 dimensional text categorization problem and [9] show a 283 dimensional face detection system.

Using the SVM formulation, the classification step for a pattern $\mathbf{x}$ using a polynomial of degree two is as follows:

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

where N_s is the number of support vectors – training data points that define the decision boundary – and α_i are Lagrange parameters.

2.3 Detecting people in new images

To detect pedestrians in a new image, we shift the 128 $\times$ 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. Figure 3 shows an example of a sequence processed with our pedestrian detection system. It is important to reiterate that no motion or tracking is used and the system is classifying on the order of 25,000 patterns per frame. This brute force search over the image is quite time consuming; several methods can be used to reduce the computation (see Section 6).

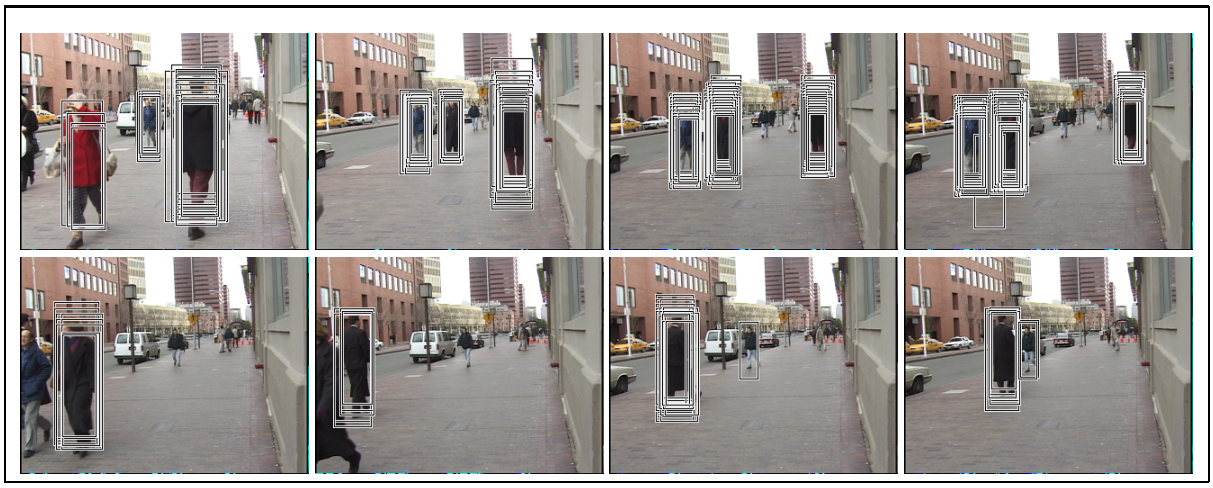


Figure 3. Processing a sequence with our frontal, rear, and side view pedestrian detection system. The system uses no motion or tracking; adding in this information would improve results.

3 Feature comparison

In Section 2.1, we discussed the characteristics of the class of features the system extracts. Here, we provide empirical results that show that using all the features leads to a higher-performing system than if the dimensionality of the representation is reduced using feature selection.

To determine the performance of a detection system, it is necessary to analyze a full ROC curve that shows the tradeoff between accuracy and the rate of false positives. For the comparison, we train the system over a database of 1,848 positive patterns (924 and their mirror images) and 11,361 negative patterns. We emphasize that our ROC curves are computed over an *out-of-sample* test set gathered outdoors and over the Internet. Figure 4 compares the ROC curves of several different versions of our system. They are as follows:

- color images with all 1,326 wavelet features
- gray-level images with all 1,326 wavelet features
- color images with 29 manually chosen wavelet features
- gray-level images with 29 manually chosen wavelet features
- gray-level images with 1,769 overlapping averages

Each of these systems uses a quadratic decision surface. The 29 feature versions were developed to reduce the dimensionality of the inner product in Equation 1 and thus lead to a faster system (see Section 6). The 29 "important" features were manually chosen by looking at the average magnitudes of the 1,326 wavelets over the positive training data. The version using 1,769 overlapping 8×8 averages is used to show that for the domain of people detection, pixel-like features do not effectively encode the information necessary to differentiate people from non-people – the local intensity difference features are much better. We use 8×8 averages that overlap by 75% (in the same manner as the overcomplete wavelets) instead of the 8,192 pixels so that we can have a fair comparison against the 1,326 feature versions (so they have approximately the same dimensions).

As expected, using color features results in a more powerful system than the gray-level versions. The performance of the system with *no feature selection* is clearly superior to all the others. Going to 1,326 gray-level features results in a drop in performance. Using 29 features decreases the performance even more, but, depending on the target application, could still offer acceptable performance. The version using overlapping 8×8 averages performs relatively poorly which seems to indicate that for people detection local intensity is not as important as the local intensity differences, as we expected.

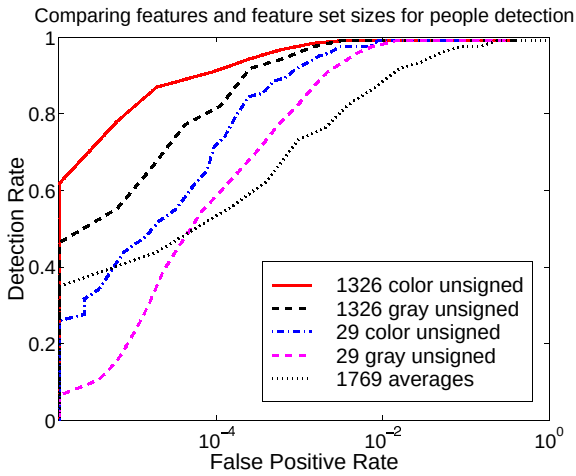
These results indicate that for the best accuracy, using all the color features is optimal. When classifying using this full set of features, we pay for the accuracy through a slower system.

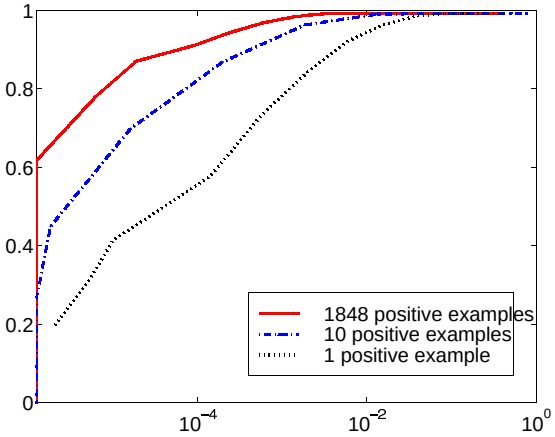
4 Training with few positive examples

One of the main attractions of the SVM framework is that it controls both the training error and the complexity of the decision classifier at the same time. This can be contrasted with other training techniques like back propagation that only minimize training error; since there is no controlling of the classifier complexity, this type of system will tend to overfit the data and provide poor generalization.

In practical terms, this means that SVMs can find good solutions to classification problems in very high dimensions; in addition to being a theoretically sound property, this capability has been demonstrated empirically in the literature in face detection [9], text categorization [5], and people detection [11]. All of these systems and other object detection systems ([15],[13]) use a large set of positive examples in addition to a large set of negative examples.

Typically, we have cheap access to an unlimited number of negative examples, while obtaining positive examples is





$\times 8$ **averages performs relatively poorly; for people detection, local intensity is not as important as the local intensity differences.**

relatively expensive. In our domain of people detection, we invested significant effort in gathering a large number of positive examples. What if our detection problem was such that we only had information about a small number of elements of the positive class? Figure 5 quantifies the performance of our 1,326 wavelet feature color people detection system when trained on 1, 10, and the full set of 1,848 (924 plus mirror images) people, each using the same set of 11,361 negative training points. The size 1 and 10 training set experiments were each run 10 times; the figure reports the average performance.

Even with a single example of our positive class, the SVM finds a decision surface that performs quite well. With 10 positive examples, the performance approaches that of the system trained with the full data set of 1,848 positive examples.

5 Integration Through Time

Processing video sequences with our technique ignores critical dynamical information since each frame is processed statically. A complete detection system for video sequences could include dynamical models of the object and tracking modules. Here, we demonstrate the impact and importance of including dynamical information by presenting a simple heuristic that serves as a zeroth order approximation to a Kalman-like model – we call this *integration through time*.

Our heuristic smooths the information in an image sequence over time by taking advantage of the fundamental *a*

reduced set vectors: From Equation 1, we can see that the computation time is also dependent on the number of support vectors, N_s; in our system, this is typically on the order of 1,000. We use results from [1] 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.

gray-level images: We use color images so that the system will be able to take advantage of the most visually significant information in the three color channels (RGB) that gets washed out in gray-level images of the same scene. Each of the image processing steps – resizing and Haar transform – are performed on each color channel separately. In order to improve system speed, our real-time version processes intensity images.

Figure 4 explicitly quantifies the reductions in performance that we must accept when going from color to gray-level and from 1,326 features to 29 features.

6.2 Integration with the DaimlerChrysler Urban Traffic Assistant

The DaimlerChrysler Urban Traffic Assistant (UTA) is a real-time vision system for obstacle detection, recognition, and tracking [3]. The system uses stereo vision to detect and segment obstacles and provides an estimate of the distance to each obstacle. We can use this information as a *focus of attention* mechanism for our people detection system. Using the knowledge of the location and approximate size of the obstacle allows us to target the people detection system to process relatively small regions for just a few sizes of people, instead of the entire image for all scales of people.

The combined system runs at more than 10 Hz. The portion of the total system time that is spent in our pedestrian detection module is 15 ms per obstacle. Within the module, the smallest amount of time is spent in the SVM classification step. This indicates that we may be able to use a much larger set of features on the order of 100 or 200 without adversely impacting the speed.

7 Conclusion

Object detection is a key technology for many future applications, among them automotive assistance systems. We have described a general *trainable* system for object detection in images that, when applied to pedestrian detection, achieves a very high detection accuracy with a low false positive rate. Our system differs significantly from previous work in that it is a pure pattern classification-based approach; it does not use motion, tracking, or any information about the scene or the camera. In this respect, it can be viewed as a high-power detection system that can be combined with other modules, that do tracking for instance, to offer even better performance. We have presented a simple heuristic scheme inspired by a Kalman filter to reduce the rate of false positives by taking advantage of dynamical information in video sequences as well as a real-time implementation of our system as part of a driver assistance system.

References

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

[2] 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.

[3] U. Franke, D. Gavrila, S. Goerzig, F. Lindner, F. Paetzold, and C. Woehler. Autonomous driving goes downtown. *IEEE Intelligent Systems*, pages 32–40, November/December 1998.

[4] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who? when? where? what? a real time system for detecting and tracking people. In *Face and Gesture Recognition*, pages 222–227, 1998.

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

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

[7] 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.

[8] 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.

[9] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Computer Vision and Pattern Recognition*, pages 130–36, 1997.

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

[11] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *Intelligent Vehicles*, pages 241–246, October 1998.

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

[13] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[14] E. Stollnitz, T. DeRose, and D. Salesin. Wavelets for computer graphics: A primer. Technical Report 94-09-11, Department of Computer Science and Engineering, University of Washington, September 1994.

[15] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, January 1998.

[16] 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.

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

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