# Pedestrian Detection with Convolutional Neural Networks

Máté Szarvas, Akira Yoshizawa, Munetaka Yamamoto and Jun Ogata
DENSO IT LABORATORY, INC.
Research and Development Group
Shibuya-Prestage 6th Floor, 3-12-22 Shibuya Shibuya-ku Tokyo, 150-0002 Japan
Tel. +81-3-6419-2321, Fax. +81-3-6419-2329
Email: {mate, ayoshizawa, muyamamoto, jogata}@d-itlab.co.jp

***Abstract*— This paper presents a novel pedestrian detection method based on the use of a convolutional neural network (CNN) classifier. Our method achieves high accuracy by automatically optimizing the feature representation to the detection task and regularizing the neural network.**

**We evaluate the proposed method on a difficult database containing pedestrians in a city environment with no restrictions on pose, action, background and lighting conditions. The false positive rate (FPR) of the proposed CNN classifier is less than 1/5-th of the FPR of a support vector machine (SVM) classifier using Haar-wavelet features when the detection rate is 90%. The accuracy of the SVM classifier using the features learnt by the CNN is equivalent to the accuracy of the CNN, confirming the importance of automatically optimized features. The computational demand of the CNN classifier is, however, more than an order of magnitude lower than that of the SVM, irrespective of the type of features used.**

## I. Introduction

More than 3000 pedestrians are killed each year in traffic accidents in Japan. Looking at the reason of these accidents, it is almost always the lack of attention on the side of the driver. Similar statistics have been reported in other countries as well. There has been a great deal of interest in recent years in the development of pedestrian detection systems that could help reduce the number and impact of these accidents. Most of the proposed systems use a camera as the sensor, because cameras can provide the high resolution needed for accurate classification and position measurement. Cameras can also be shared with other safety support subsystems in the car, such as a lane-keep assist system, improving the price-benefit ratio of the camera. In the following subsection we give a brief overview of the pedestrian detection systems described in the literature and point out issues in the design of their classifier component that motivate the method proposed in this paper.

### A. Related Work

*1) Pedestrian detection overview:* The structure of the pedestrian detection systems described in the literature can roughly be divided into region of interest (ROI) detection, feature extraction, candidate classification and tracking.

Region of interest detection is most frequently based on stereo vision [1][2] or hot-spot detection in far infrared images [3].

The feature extraction module is different in almost all systems, as illustrated in Table I. The common point is that the features are always manually designed, for example wavelet coefficients or edge strength coefficients.

The most frequently used classifier is the support vector machine (SVM) due to its high generalization ability [2] [3] [4] [5], but there are also systems using neural networks [6], time delay neural networks [7], boosted combination of linear classifiers [8][9] and template matching [10].

TABLE I
Features used in existing pedestrian detection systems.

| Feature | Classifier | Reference |
|---|---|---|
| Gray-scale value | SVM | [3] |
| Haar-wavelet coefficients | SVM | [4] |
| Four directional features | SVM | [2] |
| Vertical and horizontal edge intensities | SVM | [5] |
| Edge intensities | Chamfer | [11] |
| Gradient image | Neural Network | [6] |
| Rectangular filter | Boosted cascade | [8] |

Tracking is usually based on Kalman-filter [2], mean-shift tracker [3] or alpha-beta tracker [11].

We believe that the key-component of the detection system is the classifier. Even though the ROID and tracker also plays an important role in improving computational efficiency and accuracy, they are more or less independent from the classifier. Therefore in this paper we focus on the classifier.

*2) Feature optimization:* Traditionally, feature extraction and classification have been treated independently; the role of the feature extractor being considered to reduce the dimensionality of the input vector before classification. Recent research in the object detection literature indicates, however, that the features play a critical role for high accuracy classification [12][13]. [13] has shown that automatically derived features significantly improve the accuracy of a face detection system modeled after the mammalian vision system. The cascade-based detection architecture of Viola and Jones [8], as well as the detection system of Schneiderman [14] also rely on automatically selected features for efficient, high accuracy detection. Although these systems consider the problem of automatically choosing features that are best suited for the detection task, feature optimization is limited to feature selection from a predetermined discrete set, such as Haar-wavelets in [14] or the rectangle filters in [8].

*3) Convolutional neural networks:* The convolutional neural network (CNN) architecture proposed by LeCun [15], on the other hand, treats the feature exctractor and the classification component identically. The feature extraction filters are implemented as a hidden layer with shared weights that are optimized together with the weights of the classification component so that the total classification error is minimized. Since the optimization is based on a gradient-based

minimization procedure, the resulting feature extraction filters can have arbitrary continuous coefficients, not like the binary rectangular filters used in [8][14]. CNNs have been reported to achieve high accuracy at a low computational cost in several image recognition problems, including character recognition [15][16], hand-tracking [17], face detection [18][19], face recognition [20], facial expression recognition [21] and generic object classification [22]. Although the CNN has several favorable characteristics, it has been employed by still relatively few research groups. We conjecture that, besides the lack of a CNN implementation in most widely used neural network simulators, this is due to the small number of studies systematically comparing CNN-s with other classifiers.

It is, therefore, one purpose of this paper to provide a well controlled comparison of CNN with SVM, the most popular classifier in recent pedestrian detection systems. We demonstrate that CNN-s provide superior detection accuracy to SVM-s using classic image features. Although CNN-s have been previously applied with success to frontal face detection, this is the first demonstration of applicability to object detection that requires contour information in an unconstrained natural environment. ([22] demonstrated applicability in a difficult artificially controlled environment.)

We also propose a method to improve the generalization ability of the CNN by applying the large-margin idea of SVM-s to CNN training. Then we evaluate an original combination of CNN and SVM, training the SVM with the features extracted by the CNN. Although the accuracy of this "hybrid" SVM is higher than that of the baseline CNN, it is only marginally higher than that of the regularized CNN. Comparing the computational demand of the two methods, we demonstrate that the CNN requires about 40 times less computation than the SVM while providing a superior accuracy.

### B. Paper Outline

In the following section we give a brief overview of our complete detection system. In Section III we describe the classifiers evaluated in Section IV. In Section V we present the single-frame accuracy of the current system. Finally, in Section VI we conclude the paper with a discussion of results.

## II. System Description

The block diagram of our pedestrian detection system is depicted in Fig. 1. The input image from the camera is searched with a detection window of $30 \times 60$ pixels at several resolutions, similarly to the system of [4]. The step size is 3 pixels in both directions and we use the rescaling factor of 0.9 for building the image pyramid. The $30 \times 60$ candidate images are input to the CNN-classifier without any preprocessing, such as illumination compensation. The CNN assigns a classification score to each candidate, and the candidates with their score higher than a threshold are saved to the *raw-detection list*. This *raw-detection list* is sorted by the classification scores. Since the CNN classifier is robust to small shifts and size changes, usually each pedestrian will generate several results to the *raw-detection list*. In order to remove multiple results for the same pedestrian, we perform a multiple detection merging operation before outputting the final detection result. The multiple detection merging operation is using the following algorithm:

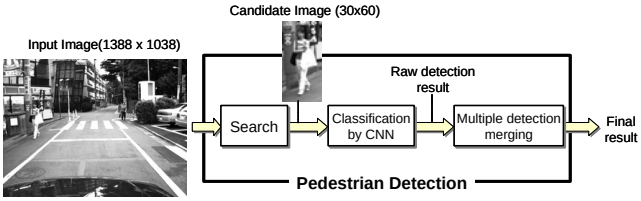


Fig. 1. The block-diagram of our pedestrian detection system.

1) move the first element of the *raw detection list* to the end of the *final detection list*
2) delete all elements of the *raw detection list* that overlap more than a predetermined threshold with the last element of the *final detection list*
3) go to 1. if the *raw detection list* is not empty

After this algorithm terminates, *final detection list* holds the final results of the detection system. We note, that this merging algorithm permits the detection of overlapping pedestrians, as long as the overlap is smaller than a threshold. A more complex merging algorithm could make use of the fact that correct pedestrians are usually generating more raw detections than the false positives, and a raw detection would be accepted only if there are several others in the same region. It is also desirable to utilize time sequence information by tracking detected pedestrians and accept a detection only if it persists. At the present stage of the development we were focusing on the classifier component as it is the key to high detection accuracy. We are planning to improve the post processing module in the future.

## III. Classifiers

In this section we give a brief overview of the different classifiers evaluated in Section IV.

### A. Convolutional Neural Network (CNN)

Convolutional Neural Networks [15] are a special variant of multilayer perceptrons (MLP) in which the first layers are configured to act as a hierarchical feature extractor. The difference to the usual fully connected MLP is that each processing node in the feature extracting layers (also called "feature maps") is connected to a different subrectangle of the preceding layer and processing nodes in each feature map share the same weight vector. This connection structure essentially configures the feature map to perform a trainable convolutional filtering on the output of the preceding layer. Feature maps are usually followed by a sub-sampling layer that reduces the dimensionality and also improves robustness to small distortions. Higher level feature maps may take their input from several lower level maps, implementing a hierarchical feature model resembling to that in the mammalian vision system [13]. The last layers of the CNN are fully connected, implementing a general purpose classifier over the features extracted by the earlier layers.

The structure and intermediate processing results of the CNN used in our experiments are shown in Fig. 2. The size of the input layer is $30 \times 60$ pixels. The height of the pedestrian in the input image is 40 pixels during training. The relatively large margin of 10 pixels at the top and bottom are needed in order to compensate for boundary effects during the two levels of convolutional filtering. There are

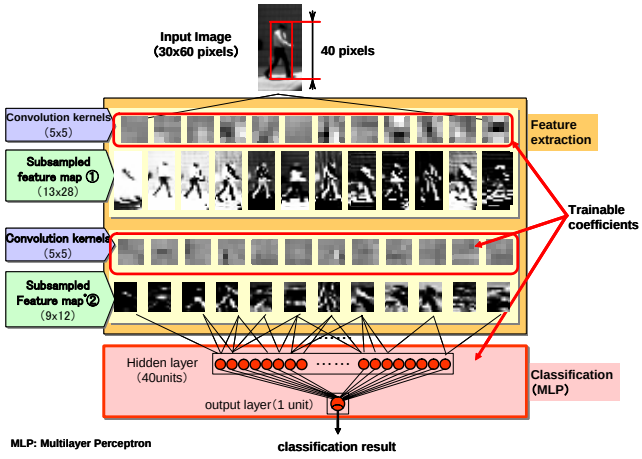


Fig. 2. The structure and intermediate processing results of the convolutional neural network.

12 feature maps in both the first and second level. Convolution and subsampling is implemented in a single step for computational efficiency, as described in [16]. Each second-level feature map is connected to exactly one first-level feature map (that is no hierarchical feature combination is performed). The size of the hidden layer, fully connected to all level 2 feature maps, is 40 units. We trained the CNN with the cross-entropy error function (see, for example, Chapter 6. in [23]) using the stochastic-gradient based training algorithm with a fixed learning rate.

### B. Support Vector Machine (SVM)

Neural networks trained with the usual mean-squared-error (or cross-entropy) minimization target function are prone to overfitting to the training data and may perform poorly on independent data. The support vector machine (SVM) classifier is optimizing an error function that minimizes the misclassifications on the training set and the L2 norm $||\underline{w}||$ of the weight vector $\underline{w}$ at the same time [24]. Geometrically, $||\underline{w}||$ is inversely proportional to the $\Delta$ separating margin between the training points and decision plane: $\Delta = \frac{1}{||\underline{w}||}$, therefore keeping $||\underline{w}||$ small results in a large margin classifier with provably high generalization ability. In practice, SVM-s map the original input vectors to a high dimensional feature space through a non-linear mapping in order to ensure separability. In the experiments of Section IV we used a Gaussian kernel with an optimized variance. The simulations were run using the *Torch* library [25].

### C. Large Margin CNN

The idea of keeping the $\Delta$ separating margin large or, equivalently, keeping $||\underline{w}||$ small can also be applied to neural networks. The usual implementation incorporates $||\underline{w}||$ in the error function as a regularizer, leading to the well known weight decay algorithm [26]. We found it numerically more stable to apply a weight norm bound regularizer. More specifically, we rescale the weight vector of each neuron whenever its norm exceeds a limit: $\underline{w} = w_{lim} \frac{\underline{w}}{||\underline{w}||}$ if $||\underline{w}|| > w_{lim}$. In our implementation the value of the weight norm limit, $w_{lim}$, is layer dependent. Setting $w_{lim} = \infty$ has the same effect as using no regularizer at all.

TABLE II
THE SIZE OF DIFFERENT DATA-SETS.

| Data set | Positive samples | Negative samples |
|---|---|---|
| Training data | 3,699 | 30,000 |
| Test data | 1,655 | 30,000 |

## IV. EXPERIMENTAL EVALUATION OF THE CLASSIFIER COMPONENT

In this section we present the experimental results of evaluating the classification component of our detection system. We evaluate the importance of large-margin training for the CNN and also present a comparison with the SVM classifier using different feature representations.

### A. Evaluation Data

The evaluation experiments in this section were performed using a set of $30 \times 60$ pixel images containing either a pedestrian ("positive samples") or a background image ("negative samples"). Both the positive and negative samples were collected using a color camera mounted on the roof of a car. The car was driving around in central Tokyo recording over 20 hours of image data. The data is equally distributed between early-morning, noon and evening (late afternoon) time zones in order to include a variety of illumination conditions. Although the parameters of the camera were set to match the illumination conditions, the high dynamic range and quick change of real-life scenes caused many of the images to be over- or under-exposed. Even though the raw data was recorded using color information, all experiments in the section were performed after converting the data to grayscale.

The positive samples were extracted by manually marking the bounding box of pedestrians in the images. We were considering only pedestrians not occluded by environmental objects, but occlusion by a bag carried by the pedestrian was permitted. There was no other restriction on the visiblity, action, orientation and pose of the pedestrian. As a result, the variability of the data-set is very high and there are many examples judged "hard to recognize" by the human observer.

The negative samples were automatically generated using the bootstrap method of Sung and Poggio [27]. 200 input images were used for generating the training negative samples and a different set of 200 images were used for generating the testing negative samples. This method of negative data generation ensured that a relatively small set of samples contained enough difficult examples and experiments could be performed efficiently.

There is no overlap between the recording time range of the training and testing data so that the same pedestrian does not appear in both of them. The size of the different data sets is displayed in Table II. Examples of positive and negative images are shown in Fig. 3.

### B. Baseline CNN

First, we trained a CNN with the standard stochastic backpropagation algorithm without any regularization. The ROC-curve of this baseline CNN classifier is shown as the lowermost curve in Fig. 4


Fig. 3. Examples of pedestrian and non-pedestrian images used for training and testing the classifiers.

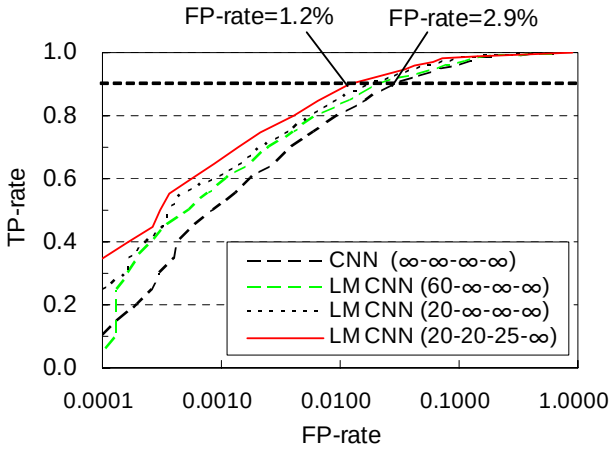


Fig. 4. The accuracy of the CNN classifier with different regularization settings. The regularization settings for each network are indicated in parentheses. The first number indicates the weight norm limit for the output layer, the second for the hidden layer, the third one for the level 2 feature maps and the last one for the level 1 feature maps. A setting of $\infty$ means that the given layer is not regularized.

(indicated as $\infty$–$\infty$–$\infty$–$\infty$). For this configuration the false positive rate (FP-rate) is 2.9% when the true positive rate (TP-rate) is 90%.

### C. Large margin CNN

Next, we evaluated the effect of regularization, applying progressively stronger weight-norm constraints during training. The ROC curves of 4 selected classifiers are shown in Fig. 4. The regularization setting for each curve is indicated in the legend. A smaller number indicates a stronger constraint. For example $\infty$–$\infty$–$\infty$–$\infty$ means no regularization and 20–$\infty$–$\infty$–$\infty$ means that the weight norm of the output node was constrained to 20. The accuracy of the network is gradually getting better as stronger weight-norm constraints are applied. The best accuracy was achieved when all the layers, including the feature maps, were regularized. The FP-rate of the best configuration decreased to 1.2% from the baseline of 2.9% when the TP-rate was 90%, indicating the importance of regularization.

### D. Comparison with SVM

Finally, we compared the CNN classifier to the more popular SVM in a controlled setting. The training and testing data was as described in Section IV-A. The Gaussian kernel SVM-s were trained using the Torch libraray [25]. The variance of the Gaussian kernel and the trade off parameter between the training error and margin was optimized separately for each experiment. The number of resulting support vectors was in the range of 5,000 and did not vary significantly among different feature parameter settings.

*1) Accuracy:* First, we evaluated the accuracy of the SVM using linear Haar wavelet features, absolute Haar wavelet features [4] and four directional features [2]. As we can see from the ROC curves displayed in Fig. 5, both the absolute Haar wavelets and the four directional features (that also take the absolute value of the filter outputs) give much better accuracy than the linear wavelet coefficients. There is no large difference between four directional features and absolute Haar wavelets. There is, however, a large difference between the accuracy of these SVM classifiers and that of the CNN. For example, the false positive rate of the SVM is 6.9%, while that of the CNN is only 1.2% when the detection rate is 90%.

We believe that this large difference in accuracy is caused by the difference of the features. In order to confirm this hypothesis, we also trained an SVM using the features computed by the CNN. As we can see in Fig. 5 the SVM using the CNN features has the highest accuracy, confirming the hypothesis that the high accuracy of the CNN classifier was the result of the features optimized to the task.

*2) Computation time:* The time needed by the CNN and by the SVM classifier to process one candidate image, including feature computation, is displayed in Fig. 6. We can see that the SVM is using almost 40 times more time than the CNN. This large difference is caused by the difference in the size of the models. The CNN has only 40 fully connected hidden nodes, but the SVM has more than 5000 support vectors. Since the computation for one hidden node in the CNN and one support vector in the SVM is roughly the same, the ratio of computation times is determined by the ratio of support vectors and hidden nodes. (In fact, this would imply a computation time ratio of 125, but the computation in the CNN is dominated by the feature extraction, since we redundantly recompute all features for each window in the present implementation.)

## V. Evaluation of Detection in Still Images

The experiments in the previous section were evaluating only the performance of the classifier component. In this section we present the evaluation results of the complete system, including search and multiple detection merging.

### A. Evaluation Data

The complete system's evaluation has been conducted using 50 $1388 \times 1038$ pixel images selected uniformly from the data-set described in Section IV.

### B. Evaluation Method

The images have been processed with the large margin CNN-based detection system and the detected pedestrians were saved in a

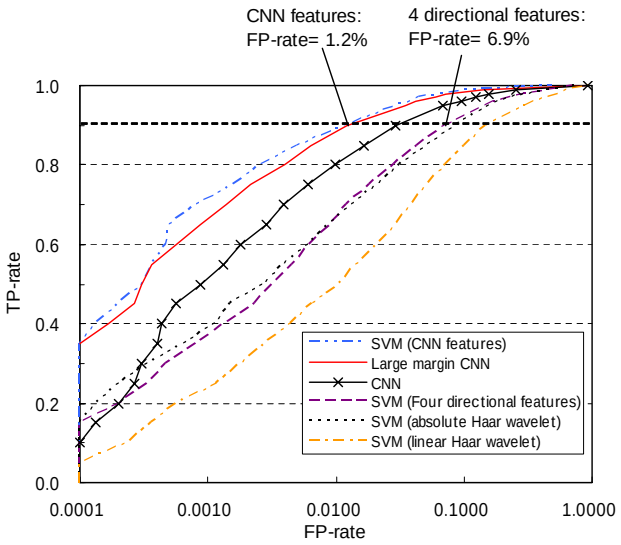


Fig. 5. The accuracy of SVM classifiers using different feature types.

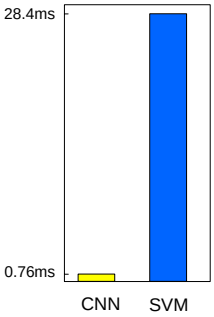


Fig. 6. The computation time used for processing one candidate by the CNN and the SVM.

file. Then these detections have been automatically compared to the manual references. This automated procedure ensures repeatability of the results as well as permits the evaluation to be conducted efficiently.

A detection result was considered correct if the height and position was within a margin of tolerance for one of the references. The tolerance for the height and vertical position of the detection frame was $\pm 30\%$ and the tolerance for the horizontal position of the detection frame was $\pm 40\%$.

### C. Evaluation Results

The evaluation has been conducted using two different settings for the search procedure. In the baseline setting we conducted the search over the full image ("full search"). In the other setting the search on the vertical axis was limited to a 200 pixel wide band ("limited search"), based on considerations of possible pedestrian heights and camera geometry. The ROC-curves for the two cases are displayed in Fig. 7. The two curves have been generated using the same models and the same detection thresholds.

The introduction of the search region constraint reduced the number of false positives, since each of the 5 evaluated points shifted to the left. However, the introduction of the search region constraint not only decreased the number of false positives but also increased the number of detections, even though exactly the same

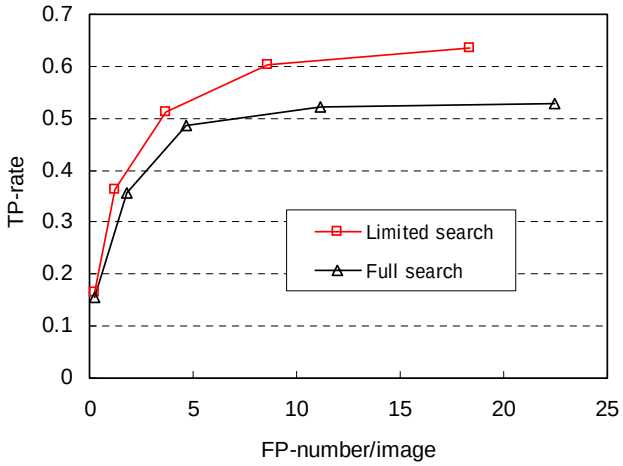


Fig. 7. The ROC curve of the pedestrian detection system with and without search region limitation.


Fig. 8. Example detection results. The blue arrow (1) indicates a false positive completely overlapping a pedestrian. The green arrow (2) indicates pedestrians detected in a crowd waiting at a crossing. (Green box: correct detection, red box: false positive.)

thresholds have been used for both curves. Analyzing the reason of this phenomenon, we found that there are many false positives that overlap a pedestrian, but the size of the detection frame is outside the tolerance range. Due to the multiple detection merging algorithm, such false positives may prevent the detection of a pedestrian. If the classifier output is higher for the false positive than for the pedestrian that it overlaps, the pedestrian cannot be detected even if the threshold is decreased to 0. Two such examples are illustrated in Figures 8 and 9. Eliminating this type of false positives by the search range constraint makes it possible to detect the pedestrian that they overlap. The reason why the classifier is not robust against this type of error is that the negative training data has been generated using images that contained no pedestrians at all. The suppression of correct detections by overlapping false positives also explains why the ROC-curves saturate at a detection rate less than 1.0.

## VI. Conclusion

In this paper we proposed and evaluated the use of the convolutional neural network (CNN) as the classifier in pedestrian detection systems. The most important conclusion from the experimental results is that good features automatically optimized to the detection task

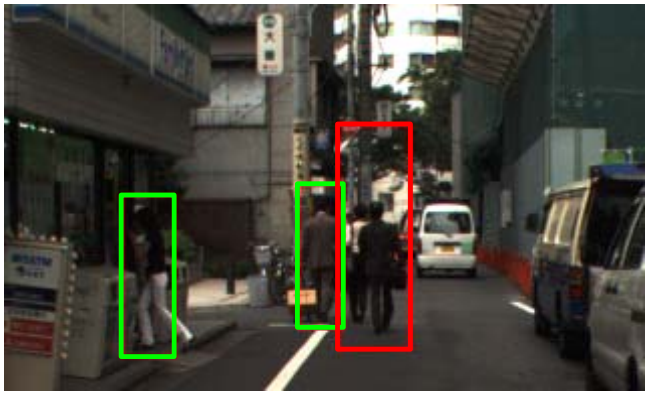

Fig. 9. Example detection results under low-contrast conditions. (Green box: correct detection, red box: false positive.)

are inevitable for high accuracy detection. The false positive (FP) rate of the CNN was less than 1/5-th of the FP-rate of the standard Haar wavelet+SVM combination. However, the accuracy of the SVM classifier surpassed the accuracy of the basic CNN when it could use the same features, due to the use of large margin training inherent to the SVM. When the CNN was regularized during training in order to obtain a large-margin decision bounday, its false positive (FP) rate dropped to half of the basic FP-rate, virtually becoming equivalent to the CNN-features+SVM combination. Besides its ability to learn the optimal features, another big advantage of the CNN is its very low computational complexity. Even though our implementation is far from optimal, the time to process one pedestrian candidate with the CNN was less than 3% of the time needed by the SVM.

Because the convolutional neural network provides both higher accuracy and needs significantly less computation, we believe that it is a better suited classifier for pedestrian detection than the popular "hand-designed feature"+SVM combination.

## ACKNOWLEDGMENT

The authors would like to thank the support of the Image Processing Group in conducting the experiments. We also thank Patrice Simard for his advice in training the CNN.

## REFERENCES

[1] D. M. Gavrila and S. Munder, "Vision-based pedestrian protection: The PROTECTOR system," in *Proc. IEEE Intelligent Vehicle Symposium, IV 2004*, Parma, Italy, June 2004.

[2] M. Soga, T. Kato, M. Ohta, and Y. Ninomiya, "Pedestrian detection using stereo vision and tracking," in *Proc. World Congress on ITS*, Nagoya, Japan, Oct. 2004.

[3] F. Xu and K. Fujimura, "Pedestrian detection and tracking with night vision," in *Proc. IEEE Intelligent Vehicle Symposium, IV 2002*, Versailles, France, June 2002.

[4] C. Papageorgiou, T. Evgeniou, and T. Poggio, "A trainable pedestrian detection system," in *Proc. Intelligent Vehicle Symposium IV'98*, Stuttgart, Germany, Oct. 1998.

[5] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe, "3d vision sensing for improved pedestrian safety," in *Proc. IEEE Intelligent Vehicle Symposium, IV 2004*, Parma, Italy, June 2004.

[6] L. Zhao and C. Thorpe, "Stereo and neural network-based pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148–154, Sept. 2000.

[7] C. Woehler, J. K. Anlauf, and U. Franke, "A time delay neural network algorithm for real-time pedestrian recognition," in *Proc. IEEE International Conference on Intelligent Vehicles IV 1998*, Stuttgart, Germany, Oct. 1998, pp. 247–252.

[8] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. IEEE International Conference on Computer Vision, ICCV 2003*, Nice, France, Oct. 2003.

[9] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: Single-frame classification and system level performance," in *Proc. IEEE Intelligent Vehicle Symposium, IV 2004*, Parma, Italy, June 2004.

[10] A. Broggi, A. Fascioli, M. Carletti, T. Graf, and M. Meinecke, "A multi-resolution approach for infrared vision-based pedestrian detection," in *Proc. IEEE Intelligent Vehicle Symposium, IV 2004*, Parma, Italy, June 2004.

[11] D. M. Gavrila and J. Giebel, "Shape-based pedestrian detection and tracking," in *Proc. IEEE Intelligent Vehicle Symposium, IV 2002*, Versailles, France, June 2002.

[12] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: the importance of good features," in *Proc. International Conference on Computer Vision ICCV 2003*, Nice, France, Oct. 2003.

[13] J. Louie, "A biological model of object recognition with feature learning," Master's thesis, Massachusetts Institute of Technology, Cambridge, 2003.

[14] H. Schneiderman, "Learning a restricted bayesian network for object detection," in *Proc. Computer Vision and Pattern Recognition CVPR'04*, Washington, DC, USA, June 2004.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[16] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. International Conference on Document Analysis and Recognition, ICDAR'03*, vol. 2, Edinburgh, Scotland, Aug. 2003.

[17] S. Nowlan and J. Platt, "A convolutional neural network hand tracker," in *Proc. International Conference on Computer Vision ICCV 2003*. San Mateo, CA: Morgan Kaufmann, 1995, pp. 901–908.

[18] R. Vaillant, C. Monrocq, and Y. LeCun, "Original approach for the localization of objects in images," *IEE Proc. on Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 245–250, Aug. 1994.

[19] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *PAMI*, vol. 26, no. 11, pp. 1408–1423, Nov. 2004.

[20] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[21] B. Fasel, "Multiscale Facial Expression Recognition using Convolutional Neural Networks," in *Proc. International Conference on Pattern Recognition, ICVGIP 2002*, Ahmedabad, India, Dec. 2002.

[22] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition," in *Proc. Computer Vision and Pattern Recognition CVPR'04*, vol. 2, Washington, DC, USA, June 2004, pp. 97–104.

[23] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995.

[24] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 2000.

[25] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: a modular machine learning software library," IDIAP, Tech. Rep. IDIAP-RR 02-46, 2002.

[26] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. Eighth Annual Conference of the Cognitive Science Society*, Hillsdale, Erlbaum, 1986, pp. 1–12.

[27] K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *PAMI*, vol. 20, no. 1, pp. 39–51, Jan. 1998.