1-1-2009

A car detection system based on hierarchical visual features

Fok Hing Chi Tivive
University of Wollongong, tivive@uow.edu.au

Abdesselam Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

A car detection system based on hierarchical visual features

Abstract

In this paper, we address the problem of detecting and localizing cars in still images. The proposed car detection system is based on a hierarchical feature detector in which the processing units are shunting inhibitory neurons. To reduce the training time and complexity of the network, the shunting inhibitory neurons in the first layer are implemented as directional nonlinear filters, whereas the neurons in the second layer have trainable parameters. A multi-resolution processing scheme is implemented so as to detect cars of different sizes, and to reduce the number of false positives during the detection stage, an adaptive thresholding strategy is developed. Tested on the UIUC car database, the proposed method achieves better classification results than some of the existing car detection approaches.

Keywords

car, detection, system, based, hierarchical, visual, features

Disciplines

Physical Sciences and Mathematics

Publication Details

F. Tivive & A. Bouzerdoum, "A car detection system based on hierarchical visual features," in IEEE Symposium on Computational intelligence for Multimedia Signal and Vision Processing, 2009, pp. 35-40.

A Car Detection System based on Hierarchical Visual Features

Fok Hing Chi Tivive and Abdesselam Bouzerdoum

Abstract—**In this paper, we address the problem of detecting and localizing cars in still images. The proposed car detection system is based on a hierarchical feature detector in which the processing units are shunting inhibitory neurons. To reduce the training time and complexity of the network, the shunting inhibitory neurons in the first layer are implemented as directional nonlinear filters, whereas the neurons in the second layer have trainable parameters. A multi-resolution processing scheme is implemented so as to detect cars of different sizes, and to reduce the number of false positives during the detection stage, an adaptive thresholding strategy is developed. Tested on the UIUC car database, the proposed method achieves better classification results than some of the existing car detection approaches.**

I. Introduction

DETECTING cars in digital images is a very challenging problem in computer vision and pattern analysis research as the same object can appear vastly different under changes in lighting conditions, viewpoint, and imaging equipment. Car detection is motivated by its potential use in industrial, civilian, and military applications; for example, it can be employed to develop intelligent traffic control and traffic guidance systems that can automate the management of traffic flow. Another potential application is to collect traffic-related data, which can be used for road planning, estimation of air pollution and marketing purposes.

In recent years, many approaches have been proposed for car detection. Agarwal *et al.* [1] developed a car detection method in which a snow classifier was trained on sparse, part-based representations, which were obtained by applying the Förstner interest operator. Fang and Qui [2] used a support vector machine (SVM) for detecting cars, and employed the maximal mutual information transform to reduce the input dimensions. Zhu *et al.* [3] proposed a two-stage car detection method. In the first stage, they used an edge area and corner area templates to reduce the number of non-car windows, and in the second stage they applied Gabor filters to extract global structure and local texture cues for further processing. Mutch and Lowe [4] proposed an object recognition model which was used for car detection. Their model had two pairs of alternate S- and C-layers (by analogy to simple and complex cells in the visual cortex). The S-layer contained Gabor filters, which were used for template matching, and the C-layer had a max filter to perform pooling operations. The last layer consisted of an SVM to classify objects in different classes. Among the aforementioned approaches, the feature extraction and the classification stages were designed separately; the drawback is that the extracted features may contain redundant information that make the training of the classifier much harder, and therefore decreasing the generalization ability of the entire system.

In this paper, we develop a car detection system that can detect and localize cars in still images. The proposed system is based on a network architecture with shunting inhibitory neurons that are trained to extract hierarchical visual features. These features are processed by a linear classifier to classify the input image as a car or a non-car object. To detect cars of different sizes, a multi-resolution processing scheme is developed. Furthermore, an adaptive thresholding method is implemented to reduce the number of false positives in the detection stage.

The organization of the paper is as follows. Section II describes the network architecture which serves as a hierarchical feature detector, followed by its training method. Section III presents the multi-resolution processing scheme and the proposed adaptive thresholding method for detecting cars of different sizes. The experimental results based on the UIUC car database is given in Section IV. Finally, Section V gives a conclusion.

II. The Proposed Car Detection Method

In this section, we describe the network architecture and its training algorithm used to develop the car detection system.

A. Network Architecture

Recently, we have developed a neural network architecture, called the SICoNNet (acronym as Shunting Inhibitory Convolutional Neural Network) [5], which was applied to several visual pattern recognition problems, such as digit recognition [6], texture classification [7], [8], gender recognition [9], [10], eye detection [11], and face detection [12], [13]. In comparison to the multilayer perceptrons, the SICoNNet has shunting inhibitory neurons as processing units which are trained to extract visual features. Herein, to reduce the complexity and training time of the proposed network, the neurons in the first layer are implemented as nonlinear bandpass filters and those in the second layer are trained by a gradient-based training method.

The network architecture has an input layer and two processing layers that contain planes of neurons, known as *feature maps*. The input layer however has a single plane of nodes which is used to receive input image of arbitrary size. The feature map, on the other hand, contains shunting inhibitory neurons, where each of them is locally connected to a small region of the input plane, i.e. its local receptive field. The set of weights connecting the neuron to its receptive field is shared among all the neurons in the feature map. Hence,

Fok Hing Chi Tivive and Abdesselam Bouzerdoum are with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Northfields Ave, Wollongong, NSW, 2522, Australia (email: {tivive, a.bouzerdoum}@uow.edu.au).

978-1-4244-2771-0/09/$25.00 ©2009 IEEE

all neurons in a feature map have the same set of weights to extract the same type of features at different parts of the input plane. To compress the extracted visual features, a down-sampling operation is performed on the feature map by reducing its spatial resolution. In [14], the feature extraction layer (i.e., the convolutional layer) was followed by a sub-sampling layer. Each neuron of the sub-sampling layer had a 2×2 non-overlapping receptive field to compute the average of its four inputs. The averaged input was then multiplied by a trainable weight, added to a bias term, and then passed to a hyperbolic tangent function to generate the output signal. In the proposed network architecture, however, the feature map of the first layer is decomposed into four sub-sampled feature maps, as shown in Fig. 1. The sub-sampled feature map in Fig. 1(b) is obtained by taking outputs at odd rows and columns, and those feature maps in Figs 1(c) and (d) take outputs at odd rows and even columns and even rows and odd columns, respectively. The fourth feature map (Fig. 1(e)) is generated by discarding outputs from odd row and odd columns.

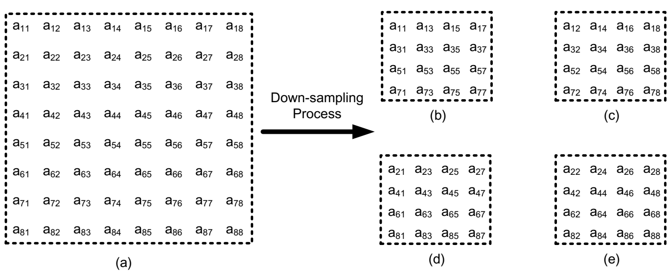

Fig. 1. In the first processing layer, the feature map is decomposed into four planes of outputs.

The rationale for using shunting inhibitory neurons as processing units is that this type of neurons is based on a nonlinear shunting inhibition mechanism, which plays an important role in neuronal information processing in the visual system [15], [16]. Furthermore, in [17]–[19], empirical results have shown that a single shunting inhibitory neuron is more powerful than a sigmoid-type neuron. In our previous work [13], we implemented classifiers based on shunting inhibitory neurons and perceptrons for classifying face images. Experimental results shown that the classifier using shunting inhibitory neurons as processing elements achieved better classification performance than that based on perceptrons. Therefore, the feature map of the proposed network is made up of an array of shunting inhibitory neurons, whose response is given by

$$z = \frac{g\left(\sum_{j=1} v_j I_j + b\right)}{a + f\left(\sum_{j=1} c_j I_j + d\right)}, \tag{1}$$

where z is the activity of the neuron, I_j denotes the inputs within the receptive field, a is the passive decay rate, W_j and C_j are the trainable weights, b and d are bias terms, and f and g are the activation functions. The activation function has to be differentiable for backpropagation training, and the most common activation functions used in neural networks are logistic, hyperbolic tangent and the Gaussian functions.

As mentioned earlier, the processing units in the first layer are designed as nonlinear bandpass filters. Therefore, each feature map computes a response map given by

$$Z_{L1} = \frac{D_\theta * I}{G * I}, \tag{2}$$

where the bias terms, a, b and d in (1) are set to zero, the activation functions f and g are set to a linear function. The set of weights $[c]$'s is defined as a Gaussian kernel, G, i.e.,

$$G = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}), \tag{3}$$

and $[w]$'s are the coefficients of the directional first derivative Gaussian, D_θ, computed as:

$$D_\theta = cos(\theta) G'_x + sin(\theta) G'_y, \tag{4}$$

with

$$G'_x = \frac{-x}{2\pi\sigma^4} \exp(-\frac{x^2 + y^2}{2\sigma^2}), \tag{5}$$

and

$$G'_y = \frac{-y}{2\pi\sigma^4} \exp(-\frac{x^2 + y^2}{2\sigma^2}). \tag{6}$$

For car detection, the first layer has five feature maps with the following filters: four directional Gaussian derivative masks oriented at 0^0, 45^0, 90^0, and 135^0 angles and a radial derivative mask. To design these filters, the Gaussian mask has a size of 7×7 with a standard deviation equal to 0.8. Figure 2 shows some examples of the response maps produced at the first layer of the network. Inspired by the "centre-on surround-off" and "centre-off surround-on" centre-surround cells in the lateral geniculate nucleus (LGN) of the thalamus of the brain, we segregate the output of each feature map of the first layer into two separate channels: "on" and "off" channels before processing by the neurons of the second layer, i.e.,

$$z^+_{L2} = \frac{g\left(\sum_{j=1} v_j max(z_{L1,j}, 0) + b\right)}{a + f\left(\sum_{j=1} c_j max(z_{L1,j}, 0) + d\right)}, \tag{7}$$

$$z^-_{L2} = \frac{g\left(\sum_{j=1} v_i min(z_{L1,j}, 0) + b\right)}{a + f\left(\sum_{j=1} c_j min(z_{L1,j}, 0) + d\right)}. \tag{8}$$

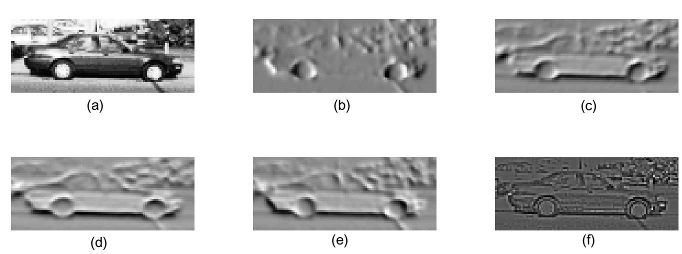

Fig. 2. The output images produced from the designed filters: (a) original image, (b) at 0^0 orientation, (c) 45^0 orientation, (d) 90^0 orientation, (e) 135^0 orientation, and (f) based on the radial derivative.

In the second layer of the network architecture, the weights of the shunting inhibitory neurons are adapted by a supervised training method and the receptive field size is 5×5. Based on some preliminary experiments, the activation functions g and f are chosen as the hyperbolic tangent and exponential functions, respectively. Before the classification stage, a local averaging operation is performed on all the feature maps of the second layer, where each non-overlapping block of responses of size $(2 \times 2 \times 4)$ from the feature map are averaged into an output signal. These outputs signals are arranged into a feature vector, as shown in Fig. 3b. The final layer is a classification layer that has a single neuron to produce an output response, which is given by

$$y = \sum_{j=1} w_j z_j + b. \tag{9}$$

The response is then compared to a threshold, and inputs with output responses that are above the threshold are classified as car objects. A schematic diagram of the network architecture with 5 and 10 feature maps in the first and second layers, respectively, is illustrated in Fig. 4.

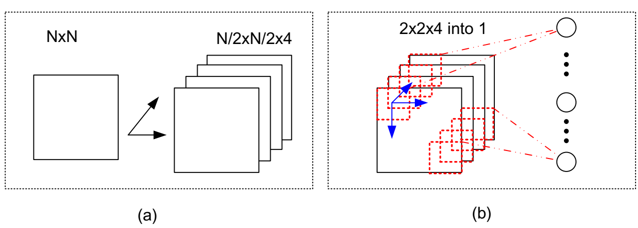

Fig. 3. The down-sampling operation performs at (a) the first processing layer and (b) the second processing layer.

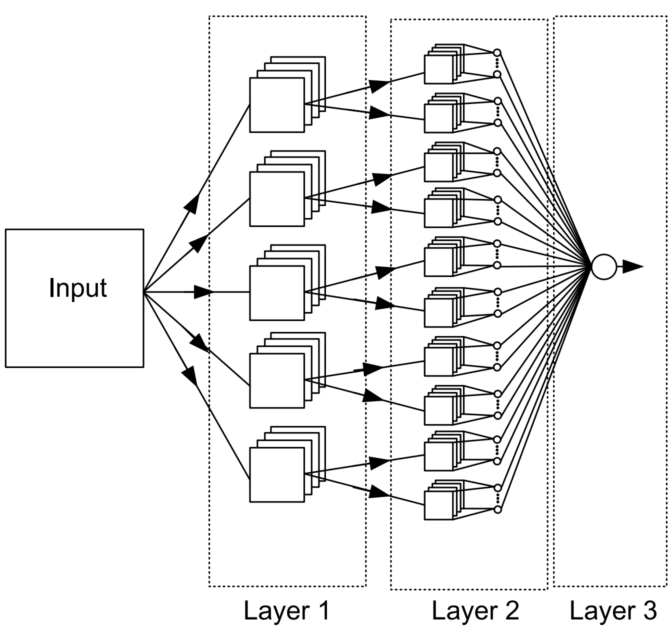

Fig. 4. A schematic diagram of the proposed network architecture.

B. Training

To train the shunting inhibitory neurons in the second layer and the neuron in the classification layer, we have developed a training algorithm that combines a gradient-based method with Least squares. The training phase consists of the following steps:

Step–1: Perform forward computation, i.e. compute the feature map outputs in response to the training patterns.

Step–2: Using the responses generated from the feature maps and the given desired outputs, the weights of the neuron in the classification layer are computed using the least squares method. This is achieved by solving the optimization problem defined below:

$$\text{minimize} \;\; J(\mathbf{w}) = ||X\mathbf{w} - \mathbf{r}||^2 \tag{10}$$

where X is a matrix whose rows contain the responses of the feature detectors to the given input patterns, $\mathbf{r}$ is the vector of desired outputs (1 for car and -1 for non-car), and $\mathbf{w}$ is the weight vector of the linear neuron, which includes a bias term. A general solution of (10) is given by

$$\mathbf{w} = (X^T X)^{-1} X^T \mathbf{r}. \tag{11}$$

Step–3: The error between the actual outputs produced from the linear classifier and the desired outputs is computed and backpropagated so as to calculate the local gradient $g(k)$ of each trainable weight in the second layer. Based on the computed local gradient $g(k)$, each weight of the shunting neuron is updated by the following expression:

$$w(k+1) = w(k) + \Delta w(k) + \mu(k)\Delta w(k-1). \tag{12}$$

The weight update $\Delta w(k)$ is computed based on the sign of the local gradients during two successive iterations, similar to the Rprop technique [20], i.e.,

$$\Delta w(k) = -\text{sign}(g(k))\gamma(k), \tag{13}$$

where

$$\gamma(k) = \begin{cases} \min(1.2\gamma(k-1), 10), & \text{if } g(k)g(k-1) > 0 \\ \max(0.5\gamma(k-1), 10^{-10}), & \text{if } g(k)g(k-1) < 0 \\ \gamma(k-1), & \text{if } g(k)g(k-1) = 0 \end{cases}. \tag{14}$$

When the current local gradient has a change in sign with respect to the previous local gradient of the same weight, the step size $\gamma(k)$ is reduced by 0.5. On the other hand, the step size is increased by a factor of 1.2 when there is no change in sign between the current and previous local gradients. However, when the product of the local gradients is zero, the weight is not updated. To prevent the training method from diverging, the step size is bounded between 10^{-10} and 10. The second term of (12) is the momentum term $\mu(k)$, which is computed as the magnitude of the Quickprop-step [21] , i.e,

$$\mu(k) = \left| \frac{g(k)}{g(k-1) - g(k)} \right|. \tag{15}$$

A complete derivation of this training method together with the derivation of the backpropagation algorithm is reported in [5].

Step–4: Repeat Step–1 to 3 until the number of training epochs has reached the desired value.

III. Car Detection Procedure

To detect and localize cars of different sizes, we have developed a multi-resolution processing scheme in which the proposed car detector is applied to each down-scaled image. An image pyramid is firstly built by down-sampling the input image at a scale factor of 1.1 so as to detect cars of bigger sizes. At each level of the image pyramid, the scaled image is fed to the car detection system, which produces a response map of size $1/2$ the input size. The response map is then compared to a threshold, T, and those responses that exceed the threshold are considered as car objects. The value of this threshold can be chosen as the value that gives the minimum error on a test set. However, this threshold may not work well for all real images due to variations between image equipment and the quality of the image. Therefore, we have implemented an adaptive strategy to compute the threshold for each input image. Assuming that the top level of the image pyramid is the smallest scale, which may contain a car. Then the threshold T is computed as the average responses generated at that level that are above a predefined cutoff value V_0. For the next scale (second level of the image pyramid), the threshold is computed as the average of the responses that are greater than V_0 at the current and higher levels of the pyramid. Mathematically, the proposed adaptive thresholding method can be expressed as follows:

At Level L_0, i.e. the top level of the image pyramid, the threshold is

$$T_0 = \frac{1}{P_0}\sum_{i=1}^{P_0} y_{0,i}^{+}, \tag{16}$$

where $y_{0,i}^{+}$'s denotes the responses at level L_0 that are above V_0, and P_0 is the number of those responses. At the next level, L_1, the threshold is given by

$$T_1 = \frac{1}{P_0+P_1}\left(\sum_{i=1}^{P_0} y_{0,i}^{+} + \sum_{i=1}^{P_1} y_{1,i}^{+}\right) = \frac{K_0 T_0 + \sum_{i=1}^{P_1} y_{1,i}^{+}}{K_1}, \tag{17}$$

where $K_0 = P_0$, $K_1 = K_0 + P_1$, and $y_{1,i}^{+}$'s are the responses at Level L_1 that exceed V_0. This can be generalized, at the L_N^{th} level, the threshold is given by

$$T_N = \frac{K_{N-1} T_{N-1} + \sum_{i=1}^{P_N} y_{N,i}^{+}}{K_{N-1} + P_N}. \tag{18}$$

From (18), it is clear that only the threshold and the total number of responses that exceed V_0 at the previous level are needed to compute the threshold of the current level of the image pyramid. During the detection phase, there will always be overlapping detections with the true car position as the system possesses a certain degree of translation and distortion invariance. The positive overlapping detections are merged into a single representative car candidate by employing a grouping technique similar to that proposed by Garcia and Delakis [22]. The positive detections are clustered according to their proximity in image and scale spaces. For each cluster, the center of the representative car candidate is taken as the centroid of the cluster and its confidence score is computed as the averaged responses of all detections within the cluster.

IV. Evaluation

This section presents an experimental evaluation of the proposed approach developed in the previous section. The approach is tested on the UIUC car database [1], which has a training set and two test sets. The training set consists of 550 segmented car images and 500 segmented non-car images of size 40×100 pixels. The first test set (Test set I) comprises 170 images containing 200 cars that are nearly the same size as those in the training set. The second test set (Test set II) has 108 images with 139 cars of different sizes. To adapt the weights of the feature detector and linear classifier, the training images are resized and clipped to the size of 20×48 pixels, and the desired outputs for set to 1 (for a car) and -1 (for a non-car). To increase the number of training images, the bootstrapping training method [23] was employed, where a larger training set of 5,000 images with equal number of car and non-car images was created. The training method explained in the previous section was then used to train the system several times; the best performing trained system on a validation set was chosen for further testing.

The classification performance of the proposed car detector is determined based on the same evaluation criteria and performance measures proposed by Agarwal *et al.* [1]. These performance measures are recall (RC), precision (P), and F-measure (Fm), which are given by

$$\text{Recall} = \frac{\text{TP}}{\text{nP}} \tag{19}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20}$$

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{21}$$

where TP is the number of true positives, FP is the number of false positives, and nP is the total number of positives in the test set. To verify the efficacy of the adaptive thresholding method, the cutoff value, V_0 was set to -0.2, and a fixed threshold of -0.2 was also used. The car detector was tested on the Test set I and the above performance measures were recorded, as shown in Table I.

TABLE I
PERFORMANCE OF THE CAR DETECTOR EVALUATED ON THE TEST SET 1, SETTING V_0 AND THE FIXED THRESHOLD TO -0.2.

	Test-Set I					
Threshold	TP	FP	RC	P	Fm	FP rate
Adaptive	200	19	100.0%	91.3%	95.5%	0.0029%
Fixed	199	81	99.5 %	71.1%	82.9%	0.0124%

TABLE II
PERFORMANCE OF THE PROPOSED METHOD EVALUATED ON THE UIUC CAR DATABASE.

	Test-Set I						Test-Set II					
V_0	TP	FP	Recall	Precision	F-measure	FP rate	TP	FP	Recall	Precision	F-measure	FP rate
-0.50	200	82	100.0%	70.9%	82.9%	0.0125%	137	86	98.6%	61.4%	75.7%	0.0090%
-0.40	200	54	100.0%	78.7%	88.1%	0.0082%	137	50	98.6%	73.3%	84.0%	0.0052%
-0.30	200	34	100.0%	85.5%	92.2%	0.0052%	138	29	99.3%	82.6%	90.2%	0.0030%
-0.20	200	19	100.0%	91.3%	95.5%	0.0029%	137	16	98.6%	89.5%	93.8%	0.0017%
-0.10	200	7	100.0%	96.6%	98.3%	0.0011%	137	8	98.6%	94.5%	96.5%	0.0008%
0	200	1	100.0%	99.5%	99.8%	0.0002%	137	6	98.6%	95.8%	97.2%	0.0006%
0.10	196	0	98.0%	100%	99.0%	0%	137	1	98.6%	99.3%	98.9%	0.0001%
0.20	189	0	94.5%	100%	97.2%	0%	131	1	94.2%	99.2%	96.7%	0.0001%
0.30	186	0	93.0%	100%	96.4%	0%	131	1	94.2%	99.2%	96.7%	0.0001%
0.40	182	0	91.0%	100%	95.3%	0%	126	1	90.6%	99.2%	94.7%	0.0001%
0.50	176	0	88.0%	100%	93.6%	0%	121	1	87.0%	99.2%	92.7%	0.0001%

Compared with a fixed threshold, the proposed adaptive thresholding method achieves lower false detection rate and false dismissal rate. With a fixed threshold, more background windows are detected as positive detections, and some of them may have higher network responses than the true car object. Hence, the true positive detection is discarded by the grouping method in the post processing stage. The sensitivity of the proposed car detection system is represented in terms of the receiver operating characteristic (ROC) and recall-precision curves. This is achieved by changing the cutoff value V_0 so as to increase the number of detections in the image pyramid, where different values ranging -0.5 to 0.5 were employed. The recall-precision and ROC curves were plotted, as shown in Figs 5a and b.

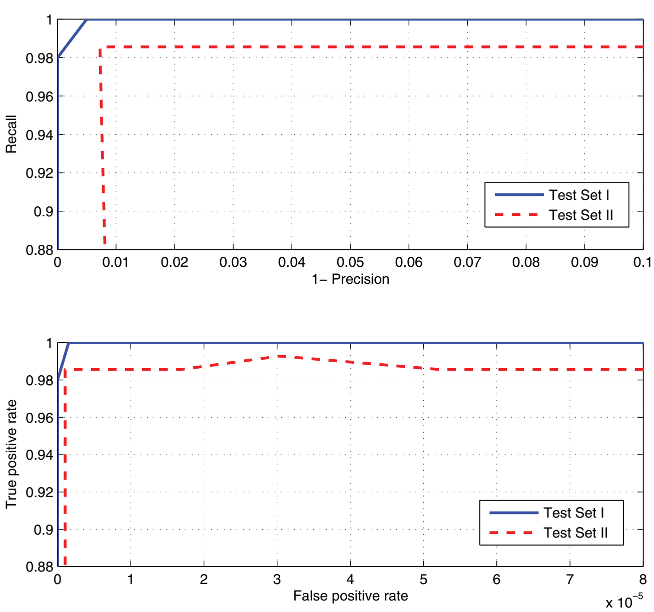

Fig. 5. The recall-precision and roc curves showing the performance of the proposed method, tested on the test set I and II.

These curves illustrate the trade-off between the recall rate and the error rate. Tables II lists the number of correct detections, number of false detections, recall rate, precision rate, F-measure and the false positive rate with respect to the changes of the cutoff value V_0. As V_0 increases, the number of false detections decreases; hence increasing the precision rate. When the precision rate is approximately equal to the recall rate, i.e., at the point of the highest F-measure, the proposed method achieves a recall rate of 100% on Test set I and 98.6% on Test set II. Herein, both test sets are assumed to have cars of different sizes, and therefore the multi-resolution processing scheme is applied to generate an image pyramid. Table III lists the performance of the proposed method along with those of existing approaches mentioned in Section I. Compared with the performances of other approaches [1]–[4], [24], our system achieves better recall rate. Furthermore, our system employs a linear classifier, whereas systems proposed in [2], [3] use more powerful classifiers, such as SVMs. The experimental results show that the proposed hierarchical feature detector is capable of extracting features that can be linearly separated. Figure 6 shows some examples of the detected image produced by the proposed method.

TABLE III
RECALL RATES BASED ON THE UIUC CAR DATABASE. THE RECALL RATE IS CHOSEN AT THE POINT OF THE HIGHEST F-MEASURE.

Approach	Test set I	Test Set II
Proposed method	100%	98.6%
Agarwal *et al.* [1]	76.5%	39.6%
Zhu *et al.* [3]	81.5%	–
Fang and Qiu [2]	87.0%	–
Fritz *et al.* [24]	–	87.8%
Mutch and Lowe [4]	99.94%	90.6%

V. CONCLUSION

In this paper, a new car detection method has been proposed. To reduce the complexity and the training time of the proposed architecture, the processing units of the first layer are designed as nonlinear bandpass filters, whereas the second layer comprises shunting inhibitory neurons with trainable parameters. These shunting neurons are used as feature detectors, whose outputs are processed by a simple linear classifier. A multi-resolution processing scheme has been developed to localize cars of different sizes in the

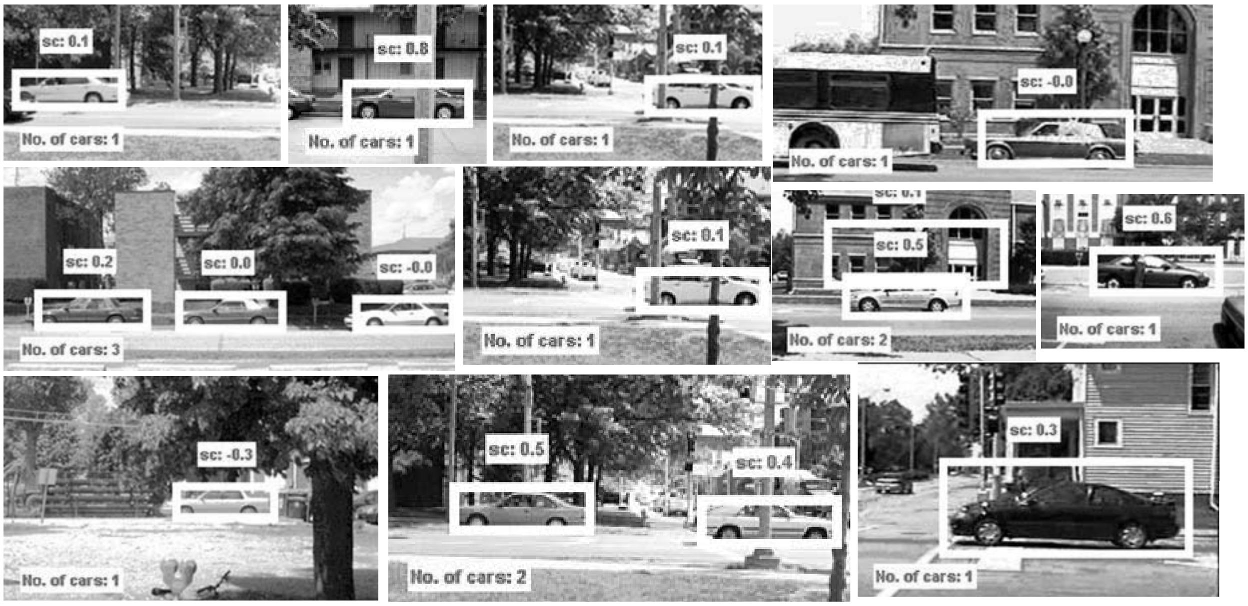

Fig. 6. Examples of the detected car images from the UIUC car database.

image. Moreover, to reduce the number of false positives during the detection stage, an adaptive thresholding technique has been implemented to process each level of the image pyramid. Compared with existing techniques, the new method outperforms existing techniques in terms of recall rate and precision.

References

[1] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Patterns Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.

[2] J. Fang and G. Qiu, "Car/non-car classification in an informative sample subspace," in *Proc. of the 18th International Conference on Pattern Recognition*, vol. 2, 2006, pp. 962–965.

[3] Z. Zhu, Y. Zhao, and H. Lu, "sequential architecture for efficient car detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[4] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 11–18.

[5] F. H. C. Tivive and A. Bouzerdoum, "Efficient training algorithms for a class of shunting inhibitory convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 541– 556, 2005.

[6] ——, "Application of siconnets to handwritten digit recognition," *International Journal of Computational Intelligence and Applications*, vol. 6, no. 1, pp. 45–59, 2006.

[7] ——, "Texture classification using convolutional neural networks," 2006, pp. 1–4.

[8] ——, "A nonlinear feature extractor for texture aegmentation," in *IEEE International Conference on Image Processing*, vol. 2, 2007, pp. II–37–II–40.

[9] ——, "A gender recognition system using shunting inhibitory convolutional neural networks," in *Proc. International Joint Conference on Neural Networks*, 2006.

[10] ——, "A shunting inhibitory convolutional neural network for gender classification," in *Proc. of the Eighteenth International Conference on Pattern Recognition (ICPR'06)*, 2006, pp. 421–424.

[11] ——, "A fast neural-based eye detection system," in *Proc. of the International Symposium on Intelligent Signal Processing and Communication Systems*, 2005, pp. 641–644.

[12] ——, "Rotation invariant face detection using convolutional neural networks," in *Neural Information Processing*. Springer Berlin / Heidelberg, 2006, vol. 4233, pp. 260–269.

[13] ——, "A hierarchical learning network for face detection with-in plane rotation," *Neurcomputing*, vol. 71, no. 16-18, pp. 3253–3263, 2008.

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

[15] J. S. Anderson, M. Carandini, and D. Ferster, "Orientation tuning of input conductance, excitation, and inhibition in cat primary visual cortex," *Journal of Neurophysiology*, vol. 84, pp. 909–926, 2000.

[16] Y. Frégnac, C. Monier, F. Chavane, P. Baudot, and L. Graham, "Shunting inhibition, a silent step in visual computation," *Journal of Physiology*, vol. 97, no. 4-6, pp. 441–451, 2003.

[17] G. Arulampalam and A. Bouzerdoumn, "Training shunting inhibitory artificial neural networks as classifiers," *Neural Network World*, vol. 10, no. 3, pp. 333–350, 2000.

[18] A. Bouzerdoum, "A new class of high-order neural networks with nonlinear decision boundaries," in *Proc. of the Sixth International Conference on Neural Information Processing*, vol. 3, Perth, 1999, pp. 1004–1009.

[19] ——, "Classification and function approximation using feed-forward shunting inhibitory artificial neural networks," in *Proc. of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 6, 2000, pp. 613–618.

[20] M. Riedmiller, "Advanced supervised learning in multilayer perceptrons – from backpropagation to adaptive learning algorithms," *Neural Networks*, no. 5, pp. 265–278, 1994.

[21] S. E. Fahlman, "An empirical study of learning speed in backpropagation networks," Computer Science, Carnegie Mellon University, Tech. Rep. CMU-CS-88-162, 1988.

[22] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408–1423, 2004.

[23] K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 20, no. 1, pp. 31–59, 1998.

[24] M. Fritz, B. Leibe, B. Caputo, and B. Schiele, "Integrating representative and discriminative models for object category detection," Tech. Rep., 2005.