Adaptive hierarchical architecture for visual recognition

Fok H. C. Tivive,[1] Abdesselam Bouzerdoum,[1] Son Lam Phung,[1,*] and Khan M. Iftekharuddin[2]

[1]School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

[2]Department of Electrical and Computer Engineering, The University of Memphis, Memphis, Tennessee 38152, USA

*Corresponding author: phung@uow.edu.au

Received 2 October 2009; revised 18 December 2009; accepted 26 December 2009; posted 5 January 2010 (Doc. ID 118048); published 28 January 2010

We propose a new hierarchical architecture for visual pattern classification. The new architecture consists of a set of fixed, directional filters and a set of adaptive filters arranged in a cascade structure. The fixed filters are used to extract primitive features such as orientations and edges that are present in a wide range of objects, whereas the adaptive filters can be trained to find complex features that are specific to a given object. Both types of filter are based on the biological mechanism of shunting inhibition. The proposed architecture is applied to two problems: pedestrian detection and car detection. Evaluation results on benchmark data sets demonstrate that the proposed architecture outperforms several existing ones. © 2010 Optical Society of America

OCIS codes: 100.3008, 100.5010, 100.2000.

1. Introduction

Given the superiority of biological systems in performing cognitive tasks, learning from nature has become a major theme in computer vision and pattern recognition research. In fact, many computational models for visual pattern recognition are motivated by our understanding of the visual systems in insects and mammals. Fukushima [1] developed a hierarchical neural network called *neocognitron* that is inspired by the discovery of simple and complex cells in a cat's visual cortex [2]. LeCun and colleagues [3] proposed convolutional neural networks for 2-D pattern recognition, which are based on the concept of local receptive fields in biology. Local receptive fields are also adopted in the pyramidal neural architecture proposed in [4]. Riesenhuber and Poggio [5] created the HMAX model for object recognition consisting of units similar to the view-tuned cells found in macaque inferotemporal cortex.

It has been shown that neurons exist in the human temporal lobe that respond best to faces, houses, or specific objects in the environment [6]. However, the striate cortex consists of many neurons that are specialized in sensing simple stimuli such as corners, bars of a particular orientation or length, or bars that move in a particular direction [7]. We propose a hierarchical architecture for classification of visual patterns. The proposed architecture consists of a set of fixed, directional filters and a set of adaptive filters arranged in a cascade structure. The fixed filters are used to extract primitive features such as line orientations and edges that are present in a wide range of objects. The adaptive filters, on the other hand, are trained to find complex features that are specific to a given object. Both types of filter are based on the biological mechanism of shunting inhibition [8].

In Section 2 we present the proposed architecture, its major stages, and a training method for solving a given visual recognition task. In Sections 3 and 4 we

0003-6935/10/1000B1-08$15.00/0

describe applications of the proposed architecture in detecting pedestrians and cars from images, respectively, including the performance of the proposed architecture in comparison with those of existing detectors on benchmark data sets. We provide our concluding remarks in Section 5.

2. Proposed Architecture

The proposed hierarchical structure consists of three processing stages as shown in Fig. 1. The first and second stages comprise nonlinear filters that are used to extract hierarchical visual features, whereas the third stage is used for classification. For a given input image, the first stage calculates elementary features at several orientations using fixed (nontrainable) filters. In contrast, the filters in the second stage have adaptive kernels that are optimized by training to extract more specific, salient features for a given problem. These features are then processed by a classifier in the third stage to detect or recognize a visual object of interest. Here we describe in detail each stage and then present a learning algorithm to train the adaptive filters and the classifier.

A. Stage 1: Directional Filters

The first stage is designed to extract features at different orientations and consists of a set of nonlinear filters that are based on a biological mechanism known as shunting inhibition. This mechanism, found in the cortical cells of the human visual system [8], has been adopted to improve image contrast [9]. Here we apply the shunting inhibition mechanism to design directional nonlinear filters whose output response is given by

$$\mathbf{Z}_{1,i} = \frac{\mathbf{D}_i * \mathbf{I}}{\mathbf{G} * \mathbf{I}}, \tag{1}$$

where $\mathbf{I}$ is a 2-D input pattern, $\mathbf{Z}_{1,i}$ is the output of the ith filter, $\mathbf{D}_i$ and $\mathbf{G}$ are the filter coefficients, and $*$ denotes 2-D convolution. Subscripts 1 and 2 in $\mathbf{Z}_{1,i}$ and $\mathbf{Z}_{2,i}$ indicate the output of the first and second processing steps in the proposed architecture, respectively.

To reduce noise in the input image, kernel $\mathbf{G}$ is chosen as an isotropic Gaussian, which is a low-pass filter:

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

The kernel $\mathbf{D}_i$ is formulated as a directional derivative Gaussian to detect image features at a particular orientation. For a given angle θ_i, the kernel is defined as

$$\mathbf{D}_i(x,y) = \cos(\theta_i)\mathbf{G}'_x(x,y) + \sin(\theta_i)\mathbf{G}'_y(x,y), \tag{3}$$

where

$$\mathbf{G}'_x(x,y) = \partial\mathbf{G}(x,y)/\partial x = \frac{-x}{2\pi\sigma^4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \tag{4}$$

$$\mathbf{G}'_y(x,y) = \partial\mathbf{G}(x,y)/\partial y = \frac{-y}{2\pi\sigma^4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{5}$$

The number of filters, N_1, for Stage 1 is chosen according to the complexity of the given problem. Each filter is associated with an angle θ_i, where

$$\theta_i = \frac{(i-1)\pi}{N_1}$$

for $i = 1, 2, \ldots, N_1$. Robust image classification requires visual features that are tolerant to small translations or geometric distortions in the input image. To achieve this, we perform a subsampling operation on the filter outputs to reduce their spatial resolution by half. This operation, illustrated in Fig. 2(a), decomposes each filter output $\mathbf{Z}_{1,i}$ into four smaller maps:

$$\mathbf{Z}_{1,i} \rightarrow \{\mathbf{Z}_{2,4i-3}, \mathbf{Z}_{2,4i-2}, \mathbf{Z}_{2,4i-1}, \mathbf{Z}_{2,4i}\},$$
$$i = 1, 2, \ldots, N_1. \tag{6}$$

The first map $\mathbf{Z}_{2,4i-3}$ is formed from the odd rows and odd columns in $\mathbf{Z}_{1,i}$; the second map, $\mathbf{Z}_{2,4i-2}$, is formed from the odd rows and even columns, and so on. Note that, in some applications where the input image size is small, the subsampling operation can be skipped and we simply have $\mathbf{Z}_{2,i} = \mathbf{Z}_{1,i}$.

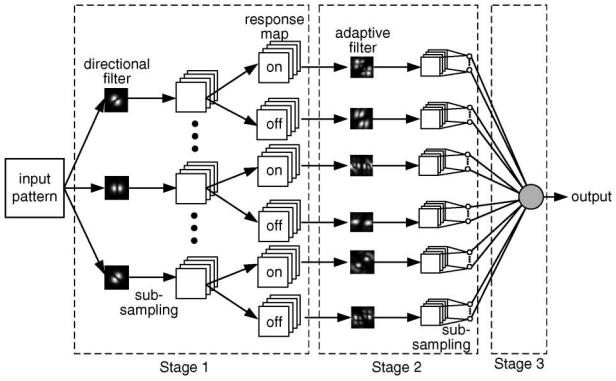


Fig. 1. Overview of the proposed hierarchical structure.

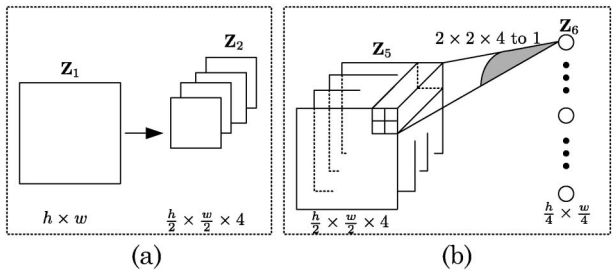


Fig. 2. Subsampling operations performed in (a) Stage 1 and (b) Stage 2.

The next processing step is motivated by the center-surround receptive fields that are found in the lateral geniculate nucleus (LGN) of the thalamus in the brain. There are two major configurations: on-center and off-center. Accordingly, we separate each subsampled map $\mathbf{Z}_{2,i}$, where $i = 1, 2, \ldots, 4N_1$, into an on-response map and an off-response map using zero as the threshold:

$$\mathbf{Z}_{2,i} \rightarrow \begin{cases} \text{on-response map}: \mathbf{Z}_{3,2i-1} = \max(\mathbf{Z}_{2,i}, 0) \\ \text{off-response map}: \mathbf{Z}_{3,2i} = -\min(\mathbf{Z}_{2,i}, 0) \end{cases}. \tag{7}$$

Essentially, for the on-response map, all the negative entries are set to 0, whereas for the off-response map, positive entries are set to 0 and the entire map is then negated. At the end of Stage 1, the features in each map are contrast-normalized, using the following transformation:

$$\mathbf{Z}_{4,i} = \frac{\mathbf{Z}_{3,i}}{\mathbf{Z}_{3,i} + \mu_i}. \tag{8}$$

where μ_i is the mean value of map $\mu_i = E(\mathbf{Z}_{3,i})$.

B. Stage 2: Adaptive Filters

Whereas Stage 1 is designed to extract fixed, elementary features, Stage 2 aims to detect more specific features that will simplify the classification task. The output maps produced by each directional filter in Stage 1 are processed by exactly two filters in Stage 2: one filter for the on-response and the other filter for the off-response. Hence, the number of filters, N_2, in Stage 2 is twice the number of filters in Stage 1: $N_2 = 2N_1$.

The filters in Stage 2 are also based on the shunting inhibition mechanism. Consider an input map $\mathbf{Z}_{4,i}$ to Stage 2. Suppose that $\mathbf{P}_k$ and $\mathbf{Q}_k$ are two adaptive kernels for the filter that corresponds to this input map. The filter output is calculated as

$$\mathbf{Z}_{5,i} = \frac{g(\mathbf{P}_k * \mathbf{Z}_{4,i} + b_k) + c_k}{a_k + f(\mathbf{Q}_k * \mathbf{Z}_{4,i} + d_k)}, \tag{9}$$

where a_k, b_k, c_k, and d_k are adjustable bias terms, and f and g are two activation functions. To avoid dividing by zero, the bias term a_k is constrained as follows:

$$a_k \geq \varepsilon - \inf(f), \tag{10}$$

where $\inf(f)$ denotes the lower bound of activation function f and ε is a small positive constant. To form a feature vector, a subsampling operation is performed across each set of four output maps. From four output maps, each nonoverlapping block of size $(2 \times 2\,\text{pixels}) \times (4\,\text{maps})$ is averaged into a single output signal, as shown in Fig. 2(b):

$$\{\mathbf{Z}_{5,4i-3}, \mathbf{Z}_{5,4i-2}, \mathbf{Z}_{5,4i-1}, \mathbf{Z}_{5,4i}\} \rightarrow \mathbf{Z}_{6,i}. \tag{11}$$

C. Stage 3: Classification

The features produced by Stage 2 are sent to the classification stage. Stage 3 can consist of any classifier; however, we use a simple linear classifier whose output y is given as

$$y = \sum_{i=1}^{N_3} w_i \mathbf{Z}_{6,i} + b, \tag{12}$$

where the w_i are adjustable weights, b is an adjustable bias term, the $\mathbf{Z}_{6,i}$ are input features to Stage 3, and N_3 is the number of features. Output y indicates the class or the label of input pattern $\mathbf{I}$. The parameters of this classifier will be determined through a supervised learning process.

D. Training Method

To train the adaptive filters in Stage 2 and the classifier in Stage 3, we propose a fast algorithm that combines two gradient-based methods and the least-squares method. Consider a training set of K input patterns $\{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_K\}$ and K corresponding desired outputs $\{d_1, d_2, \ldots, d_K\}$. The training steps can be described as follows.

Step 1: Initialize trainable parameters of nonlinear filters in Stage 2 with random values from a uniform distribution in the range $[-1, 1]$.

Step 2: Perform forward computation to find the outputs of each stage in response to the input patterns.

Step 3: Apply the least-squares method to determine the weights and bias of the linear classifier. Let $\mathbf{Z}_6$ be the inputs to the linear classifier for the given training set, where an extra column of 1's is added to $\mathbf{Z}_6$. Let $\mathbf{w}$ be a vector of all free parameters of the classifier $\mathbf{w} = [w_1, w_2, \ldots, w_{N_3}, b]^T$. Let $\mathbf{d}$ be the vector of the desired outputs $\mathbf{d} = [d_1, d_2, \ldots, d_K]^T$. The parameters of the linear classifier are found by solving the following optimization problem:

$$\text{minimize}\, E(\mathbf{w}) = ||\mathbf{Z}_6\mathbf{w} - \mathbf{d}||^2. \tag{13}$$

A general solution to Eq. (13) is given by

$$\mathbf{w} = (\mathbf{Z}_6^T\mathbf{Z}_6)^{-1}\mathbf{Z}_6\mathbf{d}. \tag{14}$$

Step 4: Compute the error given in Eq. (13) between the actual outputs of the linear classifier and the desired outputs. Apply backpropagation to compute the error gradient $g(\mathbf{v})$ for all the trainable parameters $\mathbf{v}$ in Stage 2. At training epoch t, update each trainable parameter v of the nonlinear filters as follows:

$$v(t+1) = v(t) + \Delta v(t) + \mu(t)\Delta v(t-1). \quad (15)$$

The weight update $\Delta v(t)$ is computed based on the sign of the error gradient, similar to the Rprop method [10]:

$$\Delta v(t) = -\text{sign}[g(t,v)]\gamma(t), \quad (16)$$

where

$$\gamma(t) = \begin{cases} \max(0.5\gamma(t-1), 10^{-10}), & \text{if } g(t,v)g(t-1,v) < 0 \\ \min(1.2\gamma(t-1), 10), & \text{if } g(t,v)g(t-1,v) > 0 \\ \gamma(t-1), & \text{if } g(t,v)g(t-1,v) = 0 \end{cases}. \quad (17)$$

For a given trainable coefficient v, if the gradient $g(t,v)$ changes sign, the step size $\gamma(t)$ is reduced by half. If the gradient keeps the same sign, the step size $\gamma(t)$ is increased by a factor of 1.2. To prevent divergence, the step size is bounded between $[10^{-10}, 10]$. The last term in Eq. (15) is the momentum term $\mu(t)$, which is computed using the Quick-prop method [11]:

$$\mu(t) = \left| \frac{g(t,v)}{g(t-1,v) - g(t,v)} \right|. \quad (18)$$

3. Pedestrian Detection

Here we present experimental results and performance analysis of the proposed architecture for pedestrian detection task. Pedestrian detection aims to determine the presence and the location of people or pedestrians in images and video. It is a vision task that has important applications in video surveillance [12], road safety, autonomous driving [13], and many other areas [14]. Pedestrian detection is a difficult task because pedestrian patterns can change drastically, for example, by some change in clothing or the walking, standing, or running pose of the person. Furthermore, many practical applications of pedestrian detection are in outdoor environments where the lighting conditions vary greatly.

To support studies in pedestrian detection, the Daimler–Chrysler research center has released a benchmark database [15]. Examples of pedestrian and nonpedestrian images from this database are shown in Fig. 3. The database contains three training sets (labeled 1, 2, 3) and two test sets (labeled $T1$ and $T2$). Each set has 4800 segmented pedestrian images and 5000 nonpedestrian images; the image size is 36×18 pixels.

A. Design of the Pedestrian Classifier

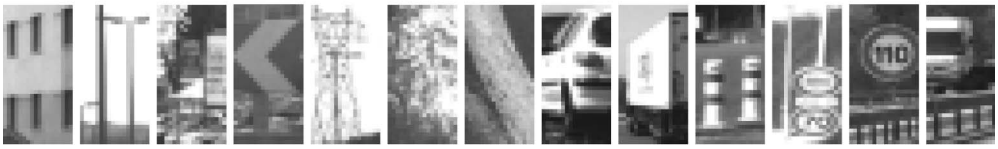
A classifier based on the proposed architecture is designed to differentiate pedestrian from nonpedestrian patterns. Stage 1 has nine directional filters with a standard deviation of $\sigma = 1.2$ and a kernel size of 7×7 pixels. Stage 2 has 18 filters with a kernel size of 5×5 pixels. The activation functions f and g for Stage 2 are chosen as hyperbolic tangent and exponential functions, respectively. In this application, because the input image size is quite small (36×18 pixels), subsampling is performed only in Stage 2. The proposed network is trained on a combination of two training sets: $\{1,2\}$, $\{1,3\}$, or $\{2,3\}$. After training, the network is evaluated on test sets $T1$ and $T2$.

In a previous study on automatic gender recognition from a single facial image [16], we found that classification accuracy is improved by presenting both the input pattern and its mirror image to the classifier and using the average response to form a classification decision. Here we evaluate both classification approaches: (i) adaptive hierarchical architecture with mirror image (or AHA with mirror) and (ii) adaptive hierarchical architecture without mirror image (or AHA without mirror).

B. Performance Evaluation and Comparison

Munder and Gavrila [15] analyzed several classifiers for pedestrian detection using the Daimler–Chrysler database. The classifiers include neural networks [17], support vector machines (SVMs), and an adaptive boosting (AdaBoost) classifier. They also evaluated different feature extraction methods, including principal component analysis (PCA), local receptive fields (LRFs), and Haar wavelets. To compare

(a) Pedestrian examples



(b) Non-pedestrian examples

Fig. 3. Image patterns from the Daimler–Chrysler pedestrian detection database.

with their results, we adopt the same evaluation procedure.

Three classifiers are generated; each classifier is trained with two training sets and the remaining training set is used for validation. The three classifiers are then evaluated on the two test sets $T1$ and $T2$ to obtain six receiver operating characteristic (ROC) curves. The six ROC curves are averaged to produce a final curve that is used for classifier comparison.

Figure 4 shows the ROC curves for the proposed approaches (AHA with mirror, AHA without mirror) and three best classifiers in the study by Munder and Gavrila [15]. In Fig. 4 the false detection rate (FDR) is the percentage of nonpedestrian patterns that are misclassified, whereas the correct detection rate (CDR) is the percentage of pedestrian patterns that is correctly classified. The figure shows that, at the same FDR, the AHA classifiers achieve higher CDRs in comparison with the other three classifiers.

The classification rates for AHA with and without a mirror are 91.9% and 90.8%, respectively. In comparison, the classification rates of the SVM classifiers using PCA, Haar, and LRF features are 84.2%, 86.2%, and 89.8%, respectively. LRF features, which are based on adaptive filters, outperform linear features such as PCA and Haar wavelets. However, the SVM classifier with the LRF features does not perform as well as the linear classifier with the features that are extracted by our cascaded structure of nonlinear filters.

4. Car Detection

Car detection has many applications in traffic safety, law enforcement, and industry. For example, it can be employed to collect traffic data for road planning, traffic management, marketing, or estimation of air pollution. In recent years, car detection has attracted significant research interest. Agarwal *et al.*[18] used a SnoW (Sparse Network of Winnows) classifier and sparse, part-based features obtained with the Förstner interest operator. Fang and Qiu used a SVM to detect cars and employed the maximal mutual information transform to reduce the input dimensions [19]. Zhu *et al.* proposed a two-stage method for car detection [20]; the first stage uses edge-area and corner-area templates to reduce the number of noncar windows, and the second stage uses Gabor filters to extract global structure and local texture features. In these car detection approaches, feature extraction and classification stages are designed separately. A drawback is that the extracted features can contain redundant information, which leads to difficulty in training and decreases the generalization ability of the entire system.

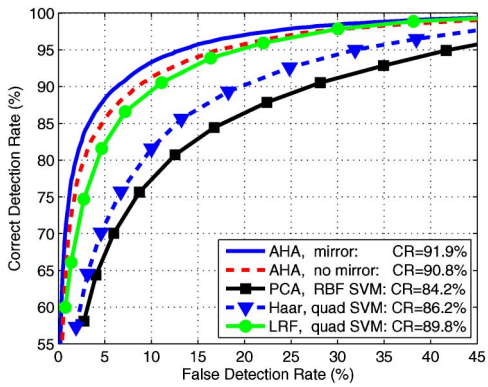


Fig. 4. (Color online) Performance comparison of different classifiers on the Daimler–Chrysler pedestrian detection database. CR denotes classification rate.

A. Car Detection Data Set and Performance Measures

A standard benchmark for car detection is the University of Illinois at Urbana–Champaign (UIUC) car database [18]. This database has a training set and two test sets. The training set consists of 550 segmented car images and 500 segmented noncar images; each image is 40×100 pixels in size. Test set 1 has 170 images containing 200 cars that have nearly the same size as those in the training set. Test set 2 has 108 images with 139 cars of different sizes. The proposed car detector is evaluated on the UIUC database, using the same criteria as in [18]. Given a test set with nP positive patterns, after the detector is applied on the test set, we record TP as the number of true positives and FP as the number of false positives. Three performance measures recall (RC), precision (P), and F-measure (Fm) are then defined as

$$\text{recall} = \frac{TP}{nP}, \tag{19}$$

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

$$F - \text{measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{21}$$

A good classifier should have a high recall rate and a high precision rate. The F-measure takes into account both criteria: the higher the F-measure, the better the classifier.

B. Designing the Car Detector

We train a car versus noncar classifier that accepts an input pattern of 20×48 pixels and produces an output of 1 for a car pattern and -1 for a noncar pattern. Stage 1 has five directional filters and a kernel size of 7×7 pixels. Stage 2 has ten adaptive nonlinear filters and a kernel size of 5×5 pixels. Subsampling steps are applied in both Stages 1 and 2. The activation functions f and g are hyperbolic

Table 1. Comparison of Thresholding Approaches on UIUC Test Set 1

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 2. Performance of the Proposed Car Detector on the UIUC Database for Different Initial Cutoff Values V_0

	Test Set 1						Test Set 2					
V_0	*TP*	*FP*	*R* (%)	*P* (%)	*Fm* (%)	*FP* Rate (%)	*TP*	*FP*	*R* (%)	*P* (%)	*Fm* (%)	*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

tangent and exponential functions, respectively. To increase the number of training images, a bootstrapping method is employed [21].

To detect cars at any position and in any size, we use a multiresolution processing scheme. A given input image $\mathbf{I}$ of arbitrary size is iteratively downsampled with a scale factor of $s = 1.1$ to form an image pyramid of $\{\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2, \ldots\}$. Each downsampled image $\mathbf{I}_k$ is scanned by the car versus noncar classifier and a response map $\mathbf{y}_k$ is produced. The response $\mathbf{y}_k(m,n)$ indicates how similar a fixed window centered at pixel location (m,n) is to a car pattern. A response that exceeds a threshold is considered a car pattern. The threshold is typically chosen to minimize the error rate on a test set of fixed-size car and noncar patterns. However, this approach does not work well on real images because variations in imaging equipment or image quality affect the choice of an optimum threshold. Therefore, we propose a strategy to determine the thresholds adaptively for each input image.

In our approach, a cutoff value V_0 is defined initially. At the top level of the image pyramid ($k = 0$), let S_0 be the set of all responses in $\mathbf{y}_0$ that exceed V_0:

$$S_0 = \{\mathbf{y}_0(m,n) | \mathbf{y}_0(m,n) > V_0\}. \tag{22}$$

The threshold T_0 is defined as the average of all responses in S_0:

$$T_0 = \frac{1}{|S_0|} \sum_{\forall s_i \in S_0} s_i. \tag{23}$$

For other levels of the image pyramid ($k > 0$), let S_k be the set of all responses in $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_k$ that exceed V_0:

$$S_k = S_{k-1} \cup \{\mathbf{y}_k(m,n) | \mathbf{y}_k(m,n) > V_0\}. \tag{24}$$

The threshold T_k is defined as the average of all responses in S_k:

$$T_k = \frac{1}{|S_k|} \sum_{\forall s_j \in S_k} s_j. \tag{25}$$

Because the classifier possesses some degree of invariance to image translation and distortion, there will be overlapping detections with the true car locations. In our system, overlapping detections are merged

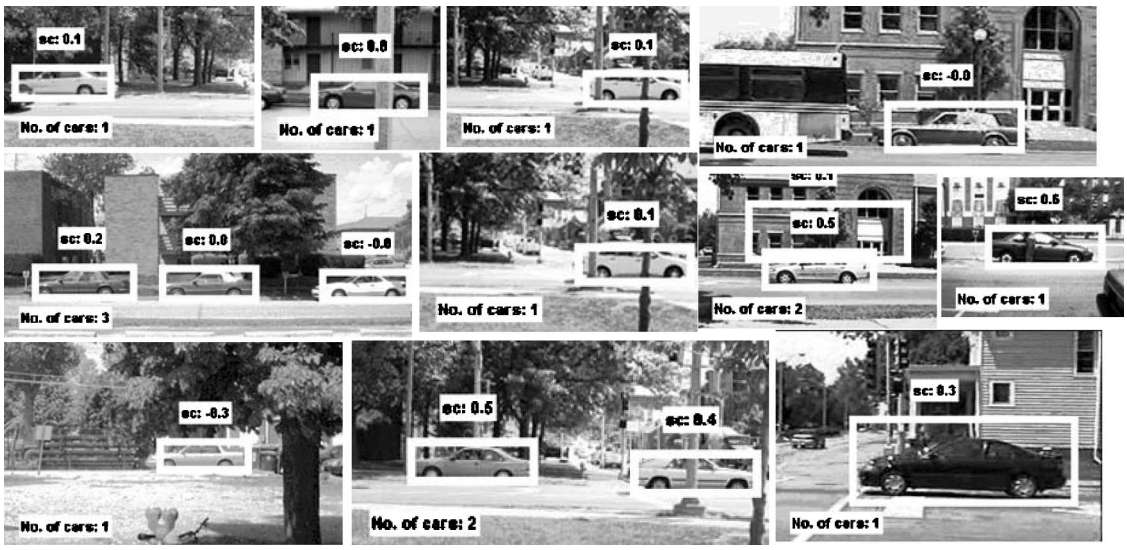


Fig. 5. Car detector outputs for some images in the UIUC database. The detection score (sc) is also shown.

Table 3. Comparison of Recall Rates of Different Car Detectors in the UIUC Database[a]

Car Detector	Year	Test Set 1 (%)	Test Set 2 (%)
Proposed AHA	2009	100	98.6
Zhu *et al.*[20]	2007	81.5	—
Mutch and Lowe [23]	2006	99.9	90.6
Fang and Qiu [19]	2006	87.0	—
Fritz *et al.*[24]	2005	—	87.8
Agarwal *et al.* [18]	2004	76.5	39.6

[a]For each detector, the recall rate is recorded at the point at which the F-measure is the highest.

using a grouping technique similar to the one proposed in [22]. The detections are clustered according to their proximity in the image space and the scale spaces. For each cluster, the center of the representative car candidate is taken as the centroid of the cluster. The confidence score of the candidate is the average response of all detections in the cluster.

C. Fixed Threshold Versus Adaptive Thresholds

We compared two thresholding approaches using: (i) a fixed threshold for all downsampled images and (ii) an adaptive threshold for each downsampled image. The fixed threshold was set to −0.2; the initial cutoff value V_0 for the adaptive threshold approach was also set to −0.2. The performance measures on the UIUC test set 1 are listed in Table 1. Use of an adaptive threshold achieves a higher precision, recall, and F-measure, in comparison with use of a fixed threshold. We observed that, with a fixed threshold, more background windows are detected as cars, and some of them have higher responses than the true car object. As a result, in postprocessing the true detection is sometimes incorrectly discarded by the grouping method.

D. Car Detector Performance

We evaluated the proposed car detector when the initial cutoff value V_0 varies in the range $[-0.5, 0.5]$. The performance measures for test sets 1 and 2 are listed in Table 2. As V_0 increases, the recall rate decreases and the precision rate increases. The F-measure increases when the recall rate is close to the precision rate. At the point at which the F-measure reaches the maximum value, the proposed detector has a recall rate of 100% with test set 1 and 98.6% with test set 2. Figure 5 shows the outputs of the proposed car detector for images in the UIUC data set.

E. Comparison with Other Car Detectors

The proposed system was compared with other car detectors in the same UIUC database. The same comparison criterion as in the existing articles was used: the recall rate at which the F-measure is at maximum. Note that the recall rate is essentially the correct detection rate; the F-measure indicates the trade-off between correct detection rate and false detection rate. The results in Table 3 indicate that, for both test sets, the proposed system achieves higher recall rates in comparison with the other five car detectors [18–20,23,24]. Note that the car detectors in [19,20] use SVMs, whereas our car detector uses a linear classifier, which means that the proposed structure of nonlinear filters is capable of extracting discriminative features that can be processed by a simple classifier.

5. Conclusion

We have presented a new architecture for visual pattern classification that is based on a combination of fixed and trainable nonlinear filters. The fixed filters, inspired by lateral geniculate and simple cortical cells, are used to extract primitive features that are common to most visual recognition tasks. The trainable filters, inspired by the more sophisticated neurons in the visual cortex, are tuned to extract features specific to a type of visual object. Evaluation results on benchmark tests in two vision tasks, detecting pedestrians and cars, show clearly that the proposed architecture outperforms existing detectors in terms of accuracy. Separating fixed and trainable filters enables us to integrate some prior knowledge about salient features into the architecture and accelerate training significantly. Among future directions, we plan to integrate feature selection into the second stage and use a more powerful classifier in the third stage.

References

1. K. Fukushima, "Neocognitron: a hierarchical neural network capable of visual pattern recognition," Neural Networks **1**, 119–130 (1988).
2. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," J. Physiol. (London) **160**, 106–154 (1962).
3. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Comput. **1**, 541–551 (1989).
4. S. L. Phung and A. Bouzerdoum, "A pyramidal neural network for visual pattern recognition," IEEE Trans. Neural Netw. **18**, 329–343 (2007).
5. M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," Nat. Neurosci. **2**, 1019–1025 (1999).
6. G. Kreiman, C. Koch, and I. Fried, "Category-specific visual responses of single neurons in the human medial temporal lobe," Nat. Neurosci. **3**, 946–953 (2000).
7. E. B. Goldstein, *Sensation and Perception* (Wadsworth, 2007).
8. L. J. Borg-Graham, C. Monier, and Y. Fregnac, "Visual input evokes transient and strong shunting inhibition in visual cortical neurons," Nature **393**, 369–373 (1998).
9. T. Hammadou and A. Bouzerdoum, "Novel image enhancement technique using shunting inhibitory cellular neural networks," IEEE Trans. Consumer Electron. **47**, 934–940 (2001).
10. M. Riedmiller, "Advanced supervised learning in multilayer perceptrons—from backpropagation to adaptive learning algorithms," Comput. Standards Interfaces **16**, 265–275 (1994).
11. S. E. Fahlman, "An empirical study of learning speed in backpropagation networks," Computer Science Technical Report CMU-CS-88-162 (Carnegie Mellon University, 1988).
12. R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," Proc. IEEE **89**, 1456–1477 (2001).

13. C. Papageorgiou and T. Poggio, "Trainable pedestrian detection," in *Proceedings of the International Conference on Image Processing* (IEEE, 1999), Vol. 4, pp. 35–39.
14. I. Haritaoglu and M. Flickner, "Attentive billboards," in *Proceedings of the International Conference on Image Analysis and Processing* (Italian Group of Researchers in Pattern Recognition, 2001), pp. 162–167.
15. S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1863–1868 (2006).
16. F. H. C. Tivive and A. Bouzerdoum, "A shunting inhibitory convolutional neural network for gender classification," in *Proceedings of the International Conference on Pattern Recognition* (International Association of Pattern Recognition, 2006), pp. 421–424.
17. C. Wohler and J. K. Anlauf, "An adaptable time-delay neural-network algorithm for image sequence analysis," IEEE Trans. Neural Netw. **10**, 1531–1536 (1999).
18. S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," IEEE Trans. Pattern Anal. Mach. Intell. **26**, 1475–1490 (2004).
19. J. Fang and G. Qiu, "Car/non-car classification in an informative sample subspace," in *Proceedings of the International Conference on Pattern Recognition* (International Association of Pattern Recognition, 2006), vol. 2, pp. 962–965.
20. Z. Zhu, Y. Zhao, and H. Lu, "Sequential architecture for efficient car detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2007), pp. 1–8.
21. K-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell. **20**, 39–51 (1998).
22. C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," IEEE Trans. Pattern Anal. Mach. Intell. **26**, 1408–1423 (2004).
23. J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2006), Vol. 1, pp. 11–18.
24. M. Fritz, B. Leibe, B. Caputo, and B. Schiele, "Integrating representative and discriminative models for object category detection," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, 2005), Vol. 2, pp. 1363–1370.