Simulation

Special Issue Article

Simulation: Transactions of the Society for Modeling and Simulation International
2017, Vol. 93(9) 759–769

DOI: 10.1177/0037549717709932
journals.sagepub.com/home/sim
SAGE SCS

Object recognition and detection with deep learning for autonomous driving applications

Ayşegül Uçar[1], Yakup Demir[2] and Cüneyt Güzeliş[3]

Abstract

Autonomous driving requires reliable and accurate detection and recognition of surrounding objects in real drivable environments. Although different object detection algorithms have been proposed, not all are robust enough to detect and recognize occluded or truncated objects. In this paper, we propose a novel hybrid Local Multiple system (LM-CNN-SVM) based on Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) due to their powerful feature extraction capability and robust classification property, respectively. In the proposed system, we divide first the whole image into local regions and employ multiple CNNs to learn local object features. Secondly, we select discriminative features by using Principal Component Analysis. We then import into multiple SVMs applying both empirical and structural risk minimization instead of using a direct CNN to increase the generalization ability of the classifier system. Finally, we fuse SVM outputs. In addition, we use the pre-trained AlexNet and a new CNN architecture. We carry out object recognition and pedestrian detection experiments on the Caltech-101 and Caltech Pedestrian datasets. Comparisons to the best state-of-the-art methods show that the proposed system achieved better results.

Keywords

Convolutional Neural Networks, Support Vector Machines, Object recognition pedestrian detection

1. Introduction

For decades, object recognition and detection have been important problems in real-life applications of autonomous vehicles. There are many applications for these utilities, including lane departure warning systems and lane-keeping assist systems that detect white lines on roads, the detection of obstacles in front of the vehicle using stereo images, a pedestrian detection warning system on images obtained from infrared cameras,[1–4] and the detection of vehicles on the road environment using both laser range sensors and camera systems.[5] In autonomous vehicle applications, it is very important to recognize, track, and detect dynamic and statics objects such as cars, trucks, animals, motorbikes, traffic signs, buildings, and pedestrians. The detection and recognition of an object or pedestrian present growing and challenging problems in the field of computer vision. The challenges come from the numerous factors affecting the performance of classifying the target objects, such as variations in light conditions, large variability in deformation, partial occlusion, the presence of shadows, and surrounding background clutters.

To solve this issue, some works have focused on developing new feature extraction algorithms, such as Scale Invariant Feature Transform (SIFT),[6,7] Histogram of Gradient (HOG),[8] Binary Robust Independent Elementary Feature (BRIEF),[9] the Speeded up Robust Feature (SURF),[10] and Fast Retina Keypoint (FREAK).[11] Some works have proposed more powerful classification algorithms, such as Support Vector Machines (SVMs),[12] Spherical/Elliptical classifiers,[13] Extreme Learning Machines (ELMs),[14,15] Adaboost, Decision Forests, and Naïve Bayes,[16,17] and the cascade boosted structure[18] to recognize and detect the objects, while others have applied accurate object classification using both good feature descriptors and good classifiers, such as Bag-of-Words (BoW) methods.[19] In BoW, the features are extracted using methods such as SIFT and SURF, and then the

[1]Fırat University, Department of Mechatronic Engineering, Elazığ, Turkey
[2]Fırat University, Department of Electrical Electronics Engineering, Elazığ, Turkey
[3]Yaşar University, Department of Electrical Electronics Engineering, İzmir, Turkey

Corresponding author:
Ayşegül Uçar, Department of Mechatronic Engineering, Fırat University, 4.kat, Elazığ, WV 23119, Turkey.
Email: agulucar@firat.edu.tr

discriminative data obtained from some interval process steps is classified using a classifier.

In recent years, deep learning methods have emerged as powerful machine learning methods for object recognition and detection.[20–25] Deep learning methods are different from traditional approaches in that they automatically and quickly learn the features directly from the raw pixels of the input images without using approaches such as SIFT, HOG, and SURF.[20] In deep learning methods, local receptive fields grow in a layer-by-layer manner. The low-level layers extract fine features, such as lines, borders, and corners, while high-level layers exhibit higher features, such as object portions, like pedestrian parts, or the whole object, like cars and traffic signs. In other words, they allow for representing an object at different granularities end-to-end. Their successes are presented on the challenging ImageNet classification task across thousands of classes[23,24] by using a kind of deep neural network called a Convolutional Neural Networks (CNN). It has been shown that CNNs outperform the recognition performance of classifiers that use conventional feature extraction methods.[25–29] However, the global features extracted using CNNs are significantly affected by illumination, noise, or occlusion when a CNN is applied to the complete image. In this paper, we therefore propose a new system using both the CNN and the SVM to solve these challenges. Recently, it was seen that some works included different combinations of both the CNN and the SVM.[30–32] For example, one study carried out robust face detection using local CNNs and SVMs based on kernel combination.[30] Scene recognition was realized by collecting the features relating to different layers of CNNs and importing them to the input of a SVM.[31] Another study[32] proposed a novel single CNN–SVM classifier for recognizing handwritten digits. Different from these works, however, in this paper, we propose to determine local robust features by multiple CNNs defined for local regions of objects and then using the SVM to recognize and detect all objects. Thus, the proposed hybrid Local Multiple CNN-SVM (LM-CNN-SVM) system can extract more robust and efficient features than those of a single CNN. In addition, we deploy two CNN architectures. One of them is a pre-trained AlexNet architecture[22] with eight layers, excluding an input layer, and the other is a CNN architecture with nine layers, excluding an input layer, similar to AlexNet. AlexNet[22] has an architecture with five convolutional layers and three fully connected layers. We prefer this network, since it was applied successfully on an ImageNet dataset for object recognition tasks.[23] We perform detailed experiments to evaluate the proposed CNNs. This paper is an extended version of our previous conference paper.[33]

The rest of this paper is organized as follows. In Section 2, the principles of the SVM, the CNN, and the hybrid LM-CNN–SVM system are described. Section 3 illustrates and discusses the experimental results. In Section 4, the paper is concluded.

2. The new hybrid Local Multiple Convolutional Neural Network–Support Vector Machine system

This section presents a brief review of the CNN and SVM classifiers used to generate the new hybrid the LM-CNN-SVM classifiers.

2.1. Convolutional Neural Networks

The basic structure of CNNs is inspired by the animal visual cortex organization. Hubel and Wiesel presented a work about ''local receptive fields'' in 1968.[34] They showed that the animal visual cortex has complex cell arrangements in small sub-regions of the visual field. The first ideas relating to CNNs were originally given by Fukushima with hierarchically organized image transformations to simulate the human visual system by using the concepts in Hubel and Wiesel[34] in 1980.[35] Unlike conventional CNNs, however, his work does not contain a shared weight. In the early 1990s, CNNs started to appear in the literature with the disadvantage of big computational load.[36,37] Nowadays, CNNs have gained popularity as both a powerful feature extractor and a classifier with the advent of powerful Graphics Processing Units (GPUs). In a short time, CNNs have been successfully applied to many computer vision fields, such as autonomous vehicles, speech recognition, and medical imaging tasks.[38–41]

CNNs consist of multiple layers similar to feed-forward neural networks. The outputs and inputs of the layers are given as a set of image arrays. CNNs can be constructed by different combinations of the convolutional layers, pooling layers, and fully connected layers with point-wise nonlinear activation functions. A typical CNN architecture is shown in Figure 1.

The layer definitions of CNNs are briefly described as follows.

Input layer: images are directly imported to the input of the network.

Convolutional layer: this layer performs the main workhorse operation of the CNN.[20,21] In CNNs, the convolution is used instead of the matrix multiplication in conventional feed-forward neural networks so as to decrease the number of weights and hereby the network complexity. The input image is convoluted by the kernels or learnable filters. The convolution provides a feature map in the output image. The obtained feature maps are imported to the input of the later convolutional layer.

Pooling layer: this layer is used to reduce the feature dimension. Thus, the resolution of the feature maps is reduced and spatial invariance is achieved. The input images are partitioned into a set of non-overlapping rectangles. Each region is down-sampled by a nonlinear down-sampling operation, such as maximum or average. The

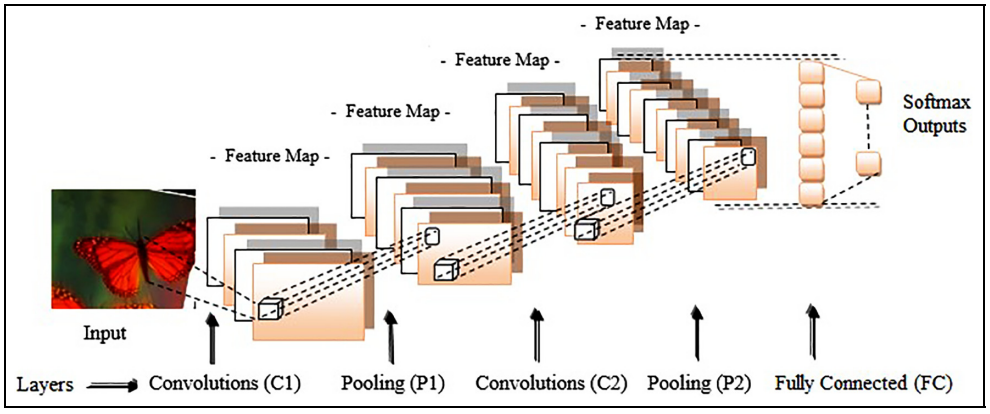

Figure 1. A Convolutional Neural Network architecture.

layer achieves faster convergence, better generalization, small invariance to translation, and distortion. The layers are usually located between successive convolutional layers to reduce the spatial size.

Rectified Linear Units (ReLU) layer: this layer includes units employing the rectifier to achieve scale invariance. The activation function of this layer is mathematically described as $f(x) = \max(0,x)$ for an input x. Thanks to this function, the gradient does not saturate in the positive region and a small and non-zero gradient is obtained, increasing the accuracy of the CNN. Moreover, the computation is realized by a simple threshold and in faster way than the equivalents with a sigmoid function and a tanh function. However, the gradient of the activation function is not definite at the origin. Instead, it is used in practice as a smoothed function $f(x) = \ln(1 + e^x)$ so that its derivative is equal to a logistic function.

Fully Connected layer: this layer is connected after several convolutional, max pooling, and ReLU layers. This layer is similar to the layers in conventional feed-forward neural networks. Its neurons are fully connected to all activations in the former layer. The layer is considered a final feature selecting layer. The outputs are calculated by means of matrix multiplication and bias addition. Moreover, as with the conventional feed-forward neural networks, the weights of these layers are estimated by minimizing only the training error.

Loss layer: a loss function is applied to measure the discrepancy between the prediction of the CNN and the real target at the final layer of a CNN. There are several loss functions. Euclidean loss can be used in a real-value regression problem. Softmax loss is used to assign the label with a single class of K mutually exclusive classes. Cross-entropy loss is commonly deployed to calculate K independent probability outputs in the range between 0 and 1 at the classification problems.

CNNs are trained using Stochastic Gradient Descent.[20] Firstly, input data are propagated in the feed-forward direction through different layers. Secondly, the output values are calculated after the digital filters extract salient features at each layer. Thirdly, the error between the actual output and network output are computed and the error is minimized by being back propagated. The weights of the CNN are then further adjusted to fine-tune them. So, the end-to-end learning process succeeds in CNNs that find a direct mapping from the raw input image data to the target class without prior knowledge and human interference.

2.2. Support Vector Machines

Given L samples of training data $(x^1, y^1), \ldots, (x^L, y^L), x \in \Re^n, y \in \{-1, 1\}$, the SVM constructs an optimal separating surface as $y^i[w^T\phi(x^i) + b] \geqslant 1$, where (ã) is a linear/nonlinear mapping function, $w \in \Re^n$ represents the normal vector, and $b \in \Re$ determines a measure for the offset of the separating hyperplane. An optimal separating surface is obtained by maximizing the minimum distance of the points closest to the hyperplanes of two classes, $\frac{2}{\|w\|}$, called a margin. In addition, the SVM formulation allows for misclassified intra-margin training examples, $y^i[w^T\phi(x^i) + b] \geqslant 1 - \xi_i$. The SVM formulation with an inequality constraint aims to minimize both the training error and the generalization error, as follows:

$$\min_{w,b,\xi} L_{\text{primal}}(w, \xi_i) = C\sum_{i=1}^{L}\xi_i + \frac{1}{2}\|w\|^2 \qquad (1)$$

$$\text{subject to } y^i[w^T\varphi(x^i) + b] \geqslant 1 - \xi_i \, \xi_i \geqslant 0 \qquad (1a)$$

where the first term, the sum of absolute errors ξ_i, indicates the measures of the distances between the intra-margin misclassified training samples and the separating hyper plane. The second term defines the margin, and C is a parameter that controls the penalty incurred by each misclassified point in the training set. Generally, larger C values cause a SVM structure with a smaller margin and better training accuracy. On the other hand, relatively smaller C

values generate a larger margin and better generalization accuracy.

The Lagrange multiplier method is firstly applied to the primal problem in (1), as follows[12,13]:

$$\hat{L}_{\text{primal}}(w,b,\xi,\lambda,\beta) = C\sum_{i=1}^{L}\xi_i + \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L}\lambda_i\{y^i[w^T\varphi(x^i)+b]-1+\xi_i\} - \sum_{i=1}^{L}\beta_i\xi_i \quad (2)$$

where $\lambda_i \geqslant 0$ and $\beta_i \geqslant 0$ are Lagrange multipliers.

The unconstraint problem in (2) is solved by minimizing it with respect to *w, b*, and ξ_i and by maximizing it with respect to λ_i andβ_i, as follows:

$$\frac{\partial\hat{L}_{\text{primal}}}{\partial w} = 0 \rightarrow \text{w} = \sum_{i=1}^{L}\lambda_{\text{i}}y^i\varphi(x^i) \quad (3)$$

$$\frac{\partial\hat{L}_{\text{primal}}}{\partial b} = 0 \rightarrow \sum_{i=1}^{L}\lambda_{\text{i}}y^i = 0 \quad (4)$$

$$\frac{\partial\hat{L}_{\text{primal}}}{\partial\xi_i} = 0 \rightarrow \lambda_{\text{i}} + \beta_i = C \quad (5)$$

and applying Karush–Kuhn–Tucker optimality conditions:

$$\lambda_i\{y^i[w^T\varphi(x^i)+b]-1+\xi_i\} = 0 \quad (6)$$

$$\beta_i\xi_i = 0 \rightarrow 0 \leqslant \lambda_{\text{i}} \leqslant C \text{for} \lambda_i \geqslant 0,\ \beta_i \geqslant 0,\ \xi_i \geqslant 0,\ \text{for } i = 1,...,L \quad (7)$$

By stating the use of λ_i, all primal variables in (3), and by introducing kernel function $K(x^i,x^j) = \varphi(x^i)^T\varphi(x^j)$ (assumed to be symmetric and positive definite), the following quadratic optimization problem, called the dual form of (2), is obtained as follows:

$$\max_{\lambda} L_{dual}(\lambda) = -\frac{1}{2}\sum_{i,j=1}^{L}\lambda_i\lambda_j y^i y^j K(x^i,x^j) + \sum_{i=1}^{L}\lambda_i \quad (8)$$

$$\text{subject to} \sum_{i=1}^{L} y^i\lambda_i = 0,\ 0 \leqslant \lambda_i \leqslant C,\ i = 1,...,L \quad (8a)$$

By solving λ_i of the dual problem (8) with quadratic programming, the separating hyper plane of the SVM is constructed as follows:

$$\ell(x) = \text{sign}\Big(\sum_{s\text{upport vectors}} y^i\lambda_i K(x^i,x^j) + b\Big) \quad (9)$$

where the support vectors are the data points x^i corresponding to $\lambda_i > 0$ and $K(x^i,x^j)$ is a kernel function. It is defined as $K(x^i,x^j) = (x^i)^T(x^j)$ for the linear basis and $K(x^i,x^j) = \exp\left(-\|x^i - x^j\|^2/2\sigma^2\right)$ for the radial basis, where σ is the spread parameter of the kernel.

2.3. Proposed Local Multiple CNN-SVM system

In this paper, we propose to use a hybrid LM-CNN system instead of a single CNN system for learning the salient features relating to the whole object. We first divide the whole image into local regions and apply LM-CNNs for extracting discriminative features of local regions. The CNN training is based on only training error minimization, similar to that of feed-forward neural networks. Feed-forward neural networks have a lower generalization performance than that of the SVM because the SVM minimizes both structural risk and empirical risk.[13] Hence, we replace the last output layer of a local CNN with a SVM classifier to compensate for the CNN's limitation. The fully connected layer of the CNN is expressed as the linear combination of the previous hidden layer outputs expressed with weights and a bias term. Moreover, the outputs of the layer are given as the inputs to the last layer of the CNN. The CNN provides the class probabilities for each input image by using the softmax activation function. On the other hand, our work uses the outputs of a fully connected layer of a CNN as the SVM inputs. We increase the generalization performance of the SVM in terms of the obtained features from the CNN. Thus, we get rid of the limitations of the CNN and SVM classifiers. The proposed approach is similar to that of feature fusion,[40] but the feature extraction process of the proposed method is different from traditional learning methods.

Figure 2 illustrates the block schema of the proposed CNN detection and recognition algorithm. Figure 3 shows the structure of the hybrid LM-CNN-SVM system. The proposed algorithm is applied in eight steps as follows.

(1) All images are divided into local regions.
(2) Each patch is wrapped to a fixed size, namely $64 \times 64 \times 3$, and converted to a gray image.
(3) Each patch is imported into the pre-trained AlexNet, shown in Table 1, and the proposed CNN, shown in Table 2.
(4) All networks are trained by stochastic gradient descent.
(5) The salient features obtained from the final fully connected layer of the networks are saved.
(6) Principal Component Analysis (PCA) is deployed to reduce the dimension of the saved features.
(7) SVM classifiers are applied to the reduced features. The one-against-rest method is used for solving multi-class classification problems.
(8) Decision fusion is used to combine the outputs of multi-class SVM classifiers.

Note that the pooling layer is used to select the features after convolutions at the CNN, while PCA is used for reducing them before importing the large features obtained from the CNN in this algorithm into the inputs of the

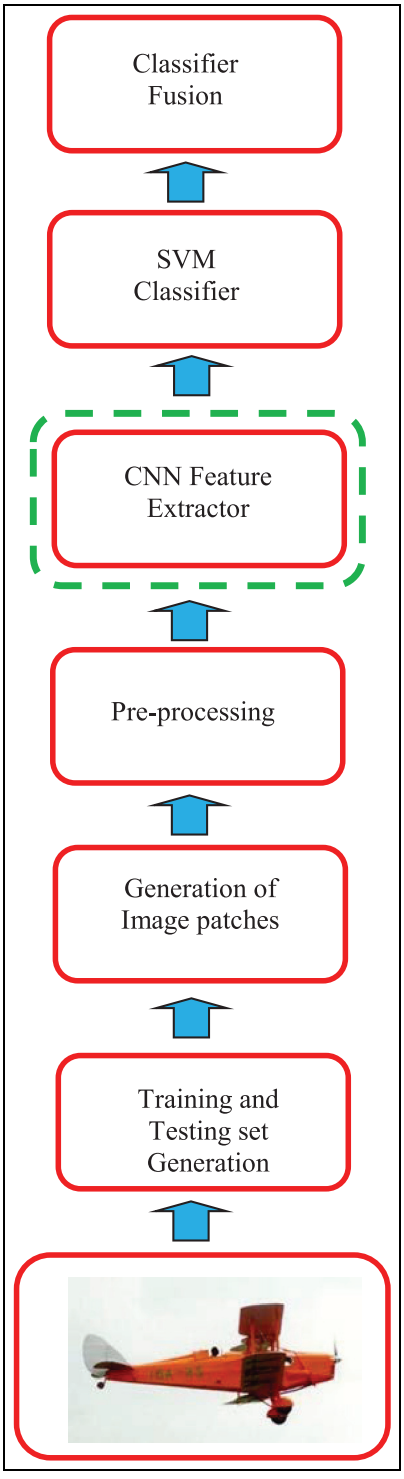

Figure 2. The block schema of the proposed algorithm. SVM: Support Vector Machine; CNN: Convolutional Neural Network.

SVM. Thus, the obtained features are de-correlated and classification is achieved quickly with less computation.[42–44] In this paper, the feature dimension is fixed to 800 to achieve both fast and accurate solutions. In addition, we employed two different network architectures in the proposed algorithm. In the first architecture, we constructed a pre-trained AlexNet.[22] AlexNet has five convolutional layers, some of which are put through by the max-pooling layers, and three fully connected layers, as shown in Table 1. When we used the pre-trained AlexNet network, we resized each image or each image patch (227 × 227) to match the dimensions of the input of the AlexNet. In the second architecture, we constructed a nine-layer CNN, given the architecture in Table 2. In this network architecture, two more convolution layers were added to AlexNet and a fully connected layer was reduced. Thus, a deeper model was designed, which leads to better performance.[44]

The proposed system can also be used for object detection. In object detection, each image is divided into patches, which are then warped to a fixed input size. A pre-trained AlexNet and the proposed network are used to extract the features relating to each patch. A two-class

Table 1. The full convolutional architecture of the AlexNet.

Type	Filter-stride	Feature maps
Input	227 × 227	1
Convolution 1	11 × 11-4	96
Max Pooling 1	3 × 3-2	96
Convolution 2	5 × 5-1	256
Max Pooling 2	3 × 3-2	256
Convolution 3	3 × 3-1	384
Convolution 4	3 × 3-1	384
Convolution 5	3 × 3-1	256
Max Pooling 5	3 × 3-2	256
Fully connected 6	6 × 6-1	4096
Fully connected 7	1 × 1	4096
Fully connected 8	1 × 1	1000

Table 2. The full convolutional architecture of the proposed Convolutional Neural Network.

Type	Filter-stride	Feature maps
Input	64 × 64	1
Convolution 1	3 × 3-1	64
Convolution 2	3 × 3-1	64
Average Pooling 2	2 × 2-2	64
Convolution 3	3 × 3-1	96
Max Pooling 3	2 × 2-2	96
Convolution 4	5 × 5-1	256
Max Pooling 4	3 × 3-2	256
Convolution 5	3 × 3-1	384
Convolution 6	3 × 3-1	384
Convolution 7	3 × 3-1	256
Max Pooling 7	3 × 3-2	256
Fully connected 8	6 × 6-1	4096
Fully connected 9	1 × 1	1 × 1 × Class number

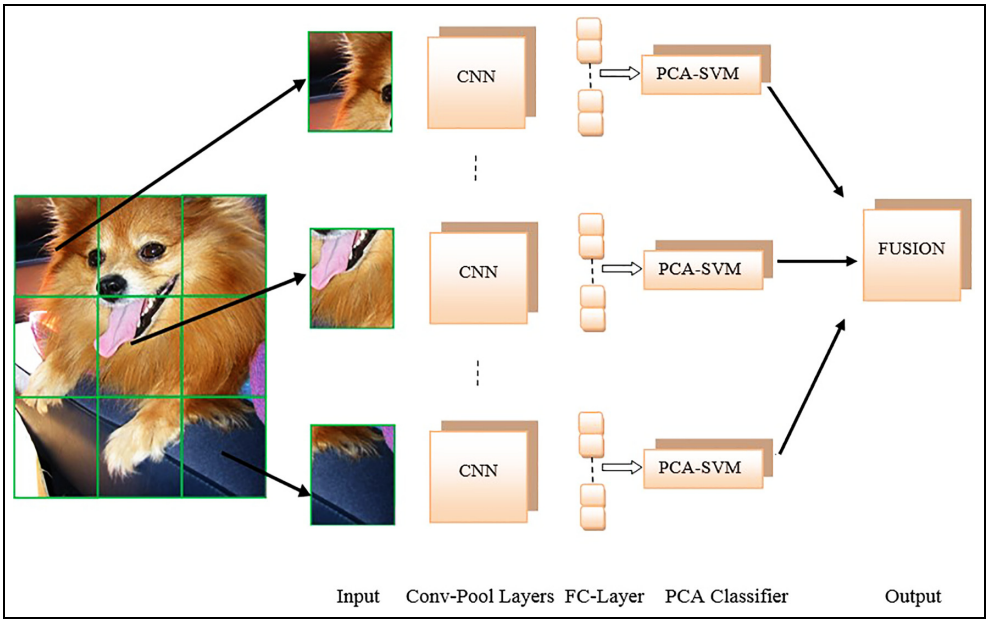

Figure 3. Structure of the proposed hybrid Local Multiple Convolutional Neural Network-Support Vector Machine (LM-CNN-SVM) system. PCA: Principal Component Analysis.

linear SVM classifier is then trained by using these features for object detection.

3. Experiments

To evaluate the performance of the proposed LM-CNN-SVM system, we conducted experiments on two well-known datasets – Caltech-101[45,46] and Caltech pedestrian[47] in this section. We carry out experiments by using MatConvNet and Vlfeat toolboxes in MATLAB.[48,49] MatConvNet is a computer vision toolbox that allows for the use of a GPU, thanks to CUDA support. In our experiments, we use the NVIDIA GTX960.

3.1. Experimental results on object recognition

We used the Caltech-101 database for object recognition. The database consists of 101 object categories, except for a background category of 9144 images with large inter-class variability.[46] Each class has a different number of images between 31 and 800. The database includes both rigid objects, such as airplanes, cars, bikes, chairs, cars, and cameras, and non-rigid objects, such as sheep, lions, and cows. We constructed training and testing sets with respect to an experimental setup procedure used in previous research[45,46] for a fair comparison. We conducted the training set by randomly selecting 15 or 30 images per category, and the testing was set to no more than 50 images per category, since the dimension of some categories is very small. We normalized each image by subtracting from the per-pixel mean across all images. We divided each of the images into nine patches and then resized them to $64 \times 64 \times 3$. We converted all images to gray scale. We located all the images as an array into a matrix. We applied a stochastic gradient descent with a minimum batch size of 30 and with the learning rates for weights and biases as 0.001 and 0.02, respectively. After we extracted the features using LM-CNNs, we applied PCA to the outputs of the final fully connected layer. We fed the reduced features as the input to the linear SVM classifier. In order to determine regularization parameters, we employed five-fold cross-validation in the range from 2^{-10} to 2^{10}. We fused the outputs of the SVM classifiers relating to each path. For this aim, we applied a decision fusion rule by using a weighted majority voting rule.

All experiments were repeated 10 times with different randomly selected training and testing images, and the averages of the per-class recognition rates were recorded for each run. In the figures and tables, CNN-SVM-1 and CNN-SVM-2 show the obtained system using AlexNet and the proposed CNN architecture, respectively. Figure 4 shows the classification accuracy of the proposed single CNN-SVM-1 and 2 systems and LM-CNN-SVM-1 and 2 systems on the Caltech-101 database. As can be seen in Figure 4, for 15 images per class, the averages of the per-class recognition rates of the proposed single CNN-SVM-1 and 2 systems and LM-CNN-SVM-1 and 2 systems are 84.80, 84.93, 87.43, and 89.80, respectively, while they are 86.80, 88.80, 91.13, and 92.80, respectively, for 30 images per class. The comparisons between the single CNN-SVM systems and the LM-CNN-SVM systems show that the LM-CNN-SVM system is always better than the single

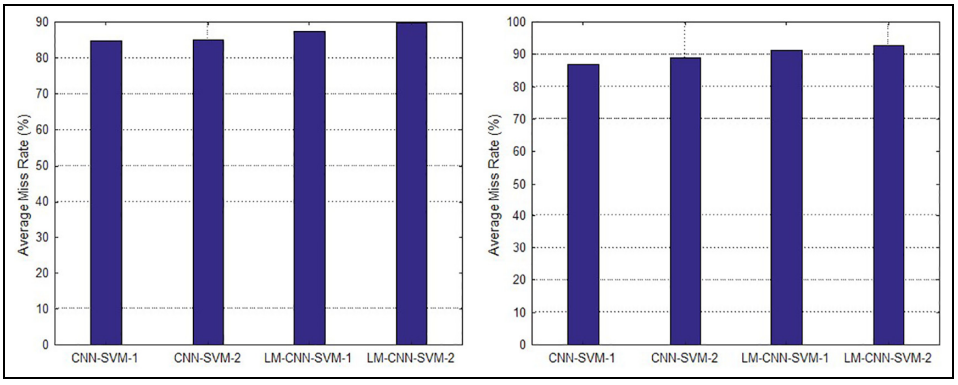

Figure 4. Comparison of the proposed Local Multiple Convolutional Neural Network-Support Vector Machine (LM-CNN-SVM) and single CNN-SVM for 15/class (a) and 30/class (b) from the Caltech-101 benchmark dataset.

Table 3. Comparison between the proposed Local Multiple Convolutional Neural Network-Support Vector Machine (LM-CNN-SVM) model and the other state-of-the-art models on the Caltech-101 benchmark dataset.

Method	Accuracy (%) 15/class	Accuracy (%) 30/class
M-HMP[50]	-	81.40 ± 0.33
ScSPM[51]	73.20	84.30
LCNM[52]	83.80 ± 0.50	86.50 ± 0.5
SPP-Net[53]	-	91.44 ± 0.70
CNN S TUNE-CLS[54]	-	88.35 ± 0.56
Proposed LM-CNN-SVM	89.80 ± 0.50	92.80 ± 0.5

M-HMP: Multipath Hierarchical Matching Pursuit model; ScSPM: linear Spatial Pyramid Matching model; LCNM: Large Convolutional Network Model; SPP-Net: CNN model applying a Spatial Pyramid Pooling process layer; CNN S TUNE-CLS: slow CNN model.

CNN-SVM system. The features extracted from CNNs make the SVM classifier more accurate, since each local region exhibits properties with a similar texture structure. In addition, we evaluated current state-of-the-art methods, including deep learning and Spatial Pyramid Matching (SPM), known as an extended version of the BoW model. The methods are briefly defined as follows:

M-HMP[50]: Multipath Hierarchical Matching Pursuit model that extracts expressive features from images through multi-pathways, similar to deep learning.

ScSPM[51]: the linear SPM model that uses a linear kernel on Spatial Pyramid Pooling of SIFT sparse codes.

LCNM[52]: Large Convolutional Network Model using a multi-layered deconvolutional network approach to visualize the feature activation.

SPP-Net[53]: CNN model applying a Spatial Pyramid Pooling process layer to remove the fixed-size constraint of the CNN.

CNN S TUNE-CLS[54]: slow CNN model including the filters with small strides using a combination of deep representations and a linear SVM.

Detailed comparison results between the proposed LM-CNN-SVM systems with a number of state-of-the-art methods are given in averages of the per-class accuracies are shown in Table 3. For 15 images per class, the ScSPM and LCNM performances are 73.20% and 83.80 ± 0.50%, respectively. For 30 images per class, the performances of M-HMP, ScSPM, LCNM, SPP-Net, and CNN S TUNE-CLS are 81.40 ± 0.33%, 84.30%, 86.50 ± 0.5%, 91.44 ± 0.70%, and 88.35 ± 0.56%, respectively. Our best results were 89.80 ± 0.50 % for 15 images per class and 92.80 ± 0.43% for 30 images per class. It can be seen that our local multiple hybrid system is superior to the competitors at object recognition. Moreover, an increasing number of training images also improves the performance. This shows that affluent features are important for recognition with large classes and large images.

3.2. Experimental results on pedestrian detection

In the object detection application, we used the Caltech pedestrian benchmark dataset.[47] The Caltech dataset was captured over 11 sessions of 640 × 480 30 Hz video taken from a vehicle driving through regular traffic in an urban environment. We used the first five subsets for training and the others for testing. We used pedestrian images that

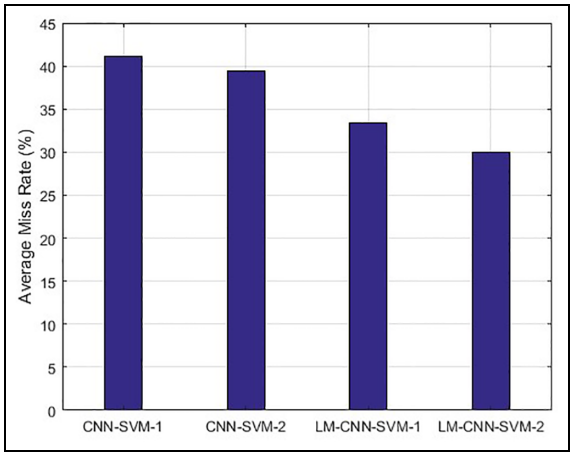

Figure 5. Comparison of single CNN-SVM and the proposed Local Multiple Convolutional Neural Network-Support Vector Machine (LM-CNN-SVM) on the Caltech benchmark pedestrian dataset.

Table 4. Comparison between the proposed Local Multiple Convolutional Neural Network-Support Vector Machine (LM-CNN-SVM) model and other state-of-the-art models on the Caltech pedestrian dataset.

Method	Average miss rate (%)
ConvNet[56]	77.20
HIKSVM[57]	73.39
HOG-SVM[6]	66
JointDeep[58]	39.3
SDN[59]	37.8
MT-DPM + Context[60]	37.64
DNNSliding[61]	32.4
DeepCascade[62]	31.11
Katamari[63]	22.0
Proposed LM-CNN-SVM-2	30.0

HIKSVM: Histogram Intersection Kernel Support Vector Machine; HOG-SVM: Histogram of Gradient Support Vector Machine; SDN: CNN model including multiple switchable layers; DNNSliding: CNN model applying a sliding window to the image.

were larger than 50 pixels in height and have at least 65% of their body parts visible. We generated the labels and evaluation code in Dollar et al.[47] In this application, we divided the images into 16 patch images for correct pedestrian detection due to a small dimension of pedestrians. We removed the regions that did not include pedestrians from the resulting image. As in Sun,[55] we measured the average miss rate to summarize the detector performance, as previous methods did. Figure 5 shows the average miss rates of the proposed algorithms on the Caltech pedestrian dataset. As can be seen from Figure 5, the average miss rates of the proposed single CNN-SVM-1 and 2 systems and LM-CNN-SVM-1 and 2 systems are 41.12%, 39.44%, 33.43%, and 30.00%, respectively. LM-CNN-SVM-2 outperformed LM-CNN-SVM-1 and the other single CNNs.

We also compared the proposed method with the best state-of-the-art models, including deep learning and HOG. The methods are given as follows.

ConvNet[56]: CNN model generated by the combination of the unsupervised and supervised methods.

HIKSVM[57]: a fast model using the Histogram Intersection Kernel (HIK) and SVM classifiers.

HOGSVM[6]: a fundamental model using both the HOG and the SVM for object detection.

JointDeep[58]: a unified deep model realized in combination with fast feature cascades, all-feature extraction, occlusion, deformation handling, and classification.

SDN[59]: CNN model including multiple switchable layers built with restricted Boltzmann machines.

MT-DPM + Context[60]: Context Multi-Task Deformable Part Model applying multi-resolution-aware transformations and the HOG.

DNNSliding[61]: CNN model applying a sliding window to the image.

DeepCascade[62]: very fast cascade classifiers built with deep networks.

Katamari[63]: extended integral channel features method.

It can be seen from the results given in Table 4 that the average miss rates of ConvNet, HIKSVM, HOG-SVM, JointDeep, SDN, MT-DPM + Context, DNNSliding, DeepCascade, and Katamari are 77.20%, 73.39%, 66%, 39.3%, 37.8%, 37.64%, 32.4%, 26.1%, and 22.0%, respectively, while the average miss rate of LM-CNN-SVM-2 is 30%. Through these analyses, it is obvious that the proposed LM-CNN-SVM-2 with an average miss rate of 30% is better than most methods based on deep learning and the HOG. There is one exception.[59] It is known from Benenson et al.[63] that the Katamari method gives very high false positive values even though it provides a low average miss rate.[62, 63] The false alarms may cause road accidents in autonomous vehicle applications. On the other hand, the cascade deep classifiers are faster than our proposed method because it does not use additional classifiers like the SVM.

The experiments demonstrate that the proposed deep model outperforms the state-of-the-art algorithms, except for that proposed by Benenson et al.[63] The proposed LM-CNN-SVM system achieves good results under partial occlusion, shadow, and background clutters for pedestrian detection.

4. Conclusions

In this work, we proposed a hybrid system using both the CNN and the SVM for object recognition and pedestrian detection. In a real environment, the appearance of objects

varies due to the variations in light conditions, partial occlusion, the presence of shadows, and surrounding background clutters. In autonomous vehicle applications, this may cause wrong object recognition and object detection, which can lead to dangerous events. We presented a LM-CNN-SVM system to handle these challenges in this paper. In our system, we used a pre-trained AlexNet architecture and a new CNN architecture including nine layers. We divided whole images into patches and employed the CNN to extract their discriminative features. Then we applied PCA to the features obtained from the CNN in order to decorrelate and reduce them. Finally, we imported them to the input of the SVM classifiers to increase the generalization ability of the system and, finally, effectively fused the images by using a majority voting rule.

The effectiveness of the proposed object recognition method was shown by means of the average of the per-class accuracies on the Caltech-101 dataset. The obtained results demonstrated that the proposed method performed significantly well in object recognition. It was observed that the LM-CNN-SVM system achieved the highest values of accuracy, 89.80 ± 0.50 and 92.80 ± 0.5 for 15 and 30 images per class, respectively. Meanwhile, a comparative study was conducted on the methods of deep learning and the HOG. The results showed that the LM-CNN-SVM system significantly surpassed the state-of-the-art methods in excellence in terms of higher recognition performance.

The LM-CNN-SVM system was also applied for object detection. The performance was evaluated by means of the average miss rate on the Caltech pedestrian dataset. The obtained results demonstrated that the proposed method that the proposed method reached to a low miss rate at object recognition. The proposed method will continue to improve object recognition and detection in terms of both accuracy and speed in using for real-time applications.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

5. References

1. Mobileye Pedestrian Collision Warning System, http://www.mobileye.com/en-uk/mobileye-features/pedestrian-collision-warning/. (accessed 1 January 2016).
2. Coelingh E, Eidehall A and Bengtsson M. Collision warning with full auto brake and pedestrian detection - a practical example of automatic emergency braking. In: *Proceedings of the 13th international IEEE conference on intelligent transportation systems (ITSC)*, Funchal, Portugal, 19–22 September 2010. Piscataway, NJ: IEEE.
3. BMV Driving Assistance Package. http://www.bmw.com/ (accessed 1 January 2016).
4. VW Emergency Assistance System. http://safecarnews.com/ (accessed 1 January 2016).
5. Toyota. http://www.toyota-global.com/innovation/safety_technology/toyota-safety-sense/.
6. Ke Y and Sukthankar R. Pca-sift: a more distinctive representation for local image descriptors. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* Washington, DC, USA, 27 June–2 July 2004, pp.II-506–II-513, vol. 2. Piscataway, NJ: IEEE.
7. Lowe D. Object recognition from local scale-invariant features. In: *Proceedings of the seventh IEEE international conference on computer vision*, Washington, DC, USA, 20–25 September 1999, vol. 2, pp.1150–1157. Piscataway, NJ: IEEE.
8. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, San Diego, CA, USA, 20–25 June 2005, pp.886–893. Piscataway, NJ: IEEE.
9. Calonder M, Lepetit V, Strecha C, et al. Brief: binary robust independent elementary features. In: *The European conference on computer vision (ECCV)* (ed K Daniilidis, P Maragos and N Paragios), Crete, Greece, 5–11 September 2010, LNCS 6314, pp.778–792. Berlin: Springer.
10. Bay H, Tuytelaars T and Van Gool L. SURF: speeded up robust features. In: *The European conference on computer vision (ECCV)* (ed A Leonardis, H Bischof and A Pinz), Graz, Austria, 7–13 May 2006, LNCS 3951, pp.404–417. Berlin: Springer.
11. Leutenegger S, Chli M and Siegwart R. Brisk: Binary robust invariant scalable keypoints. In: *Proceedings of IEEE international conference on computer vision (ICCV)*, Barcelona, Spain, 6–13 November 2011, pp.2548–2555. Piscataway, NJ: IEEE.
12. Cortes C and Vapnik VN. Support vector networks. *Mach Learn* 1995; 20: 273–297.
13. Uçar A, Demir Y and Güzeliş C. A penalty function method for designing efficient robust classifiers with input–space optimal separating surfaces. *Turk J Elec Eng Comp Sci* 2014; 22: 1664–1685.
14. Uçar A, Demir Y and Güzeliş C. A new facial expression recognition based on curvelet transform and online sequential extreme learning machine initialized with spherical clustering. *Neural Comput Appl* 2016; 27: 131–142.
15. Huang GB, Zhu QY and Siew CK. Extreme learning machine: theory and applications. *Neurocomputing* 2006; 70: 489–501.
16. Ho TK. The random subspace method for constructing decision forests. *IEEE Trans Pattern Anal Mach Intell* 1998; 20: 832–844.
17. Webb GI, Boughton J and Wang Z. Not so naive Bayes: aggregating one-dependence estimators. *Mach Learn* 2005; 58: 5–24.
18. Viola P and Jones M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Kauai, HI, USA, 8–14 December 2001, pp.511–518. Piscataway, NJ: IEEE.
19. Lin W-C, Tsai C-F, Chen Z-Y, et al. Keypoint selection for efficient bag-of-words feature generation and effective image classification. *Inform Sci* 2016; 329: 33–51.

20. Deng L and Yu D. Deep learning: methods and applications. *Foundat Trend Signal Proc* 2014; 7: 3–4.
21. Christian S, Toshev A and Erhan D. Deep neural networks for object detection. In: *Proceedings of advances in neural information processing systems (NIPS)*, Lake Tahoe, Nevada, 5–10 December 2013, pp.2553–2561. Red Hook, NY: Curran Associates Inc.
22. Krizhevsky A, Sutskever I and Hinton GE. Imagenet classification with deep convolutional neural networks. In: *Proceedings of advances in neural information processing systems*, (NIPS), Lake Tahoe, Nevada, 2012, pp.1097–1105. Red Hook, NY: Curran Associates Inc.
23. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vision* 2015; 115: 211–252. Berlin: Springer.
24. Deng J, Berg A, Satheesh S, et al. ILSVRC-2012, http://www.image-net.org/challenges/LSVRC/2012/ (accessed 1 January 2016).
25. Yan K, Wang , Liang D, et al. CNN vs. SIFT for image retrieval: Alternative or complementary? In: *Proceedings of the ACM on multimedia conference (MM)*, Amsterdam, The Netherlands, 15–19 October 2016, pp.407–411. New York: ACM.
26. Tian Y, Luo P, Wang X, et al. Deep learning strong parts for pedestrian detection. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Washington, DC, USA, 07–13 December 2015, pp.1904–1912. Piscataway, NJ: IEEE.
27. Ouyang W, Wang X, Zeng X, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015, pp.2403–2412. Piscataway, NJ: IEEE.
28. Zhang S, Benenson R and Schiele B. Filtered channel features for pedestrian detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, USA, 7–12 June 2017, pp.1751–1760. Piscataway, NJ: IEEE.
29. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: *Proceedings of advances in neural information processing systems (NIPS)*, Montréal, Canada, 2015, pp.91–99. Cambridge, MA: MIT Press.
30. Tao Q-Q, Zhan S, Li X-H, et al. Robust face detection using local CNN and SVM based on kernel combination. *Neurocomputing* 2016; 211: 98–105.
31. Tanga P, Wanga H and Kwong S. G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing* 2017; 225: 188–197.
32. Niu X-X and Suen CY. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recogn* 2012; 45:1318–1325.
33. Uçar A, Demir Y and Güzeliş C. (2016). Moving towards in object recognition with deep learning for autonomous driving applications. In: *Proceedings of IEEE international conference on innovations in intelligent systems and applications (INISTA)*, Sinaia, Romania, 2–5 August 2016, pp.1–5. Piscataway, NJ: IEEE.
34. Hubel DH and Wiesel TN. Receptive fields and functional architecture of monkey striate cortex. *J Physiol* 1968; 195: 215–243.
35. Fukushima K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern* 1980; 36: 193–202.
36. LeCun Y, Boser B, Denker JS, et al. Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1989; 1: 541–551.
37. LeCun BB, Denker JS, Henderson D, et al. Handwritten digit recognition with a back-propagation network. In: *Proceedings of advances in neural information processing systems (NIPS)*, Denver, Colorado, USA, 26–29 November 1990, pp.396–404. San Francisco, CA: Morgan Kaufmann Publishers Inc.
38. Kim H, Hong S, Son S, et al. High speed road boundary detection on the images for autonomous vehicle with multilayer CNN. In: *Proceedings of IEEE conference on circuits and systems (ISCAS)*, Bangkok, Thailand, 25–28 May 2003, vol. 5, pp.V-769–V-772. Piscataway, NJ: IEEE.
39. Shen W, Zhou M, Yang F, et al. Multi-scale convolutional neural networks for lung nodule classification. In: *International conference on information processing in medical imaging (IPMI) Isle of Skye* (ed S Ourselin, DC Alexander, C-F Westin, et al.), Scotland, 28 June–3 July 2015, LNCS 9123, pp.588–599. Berlin: Springer.
40. Chen D and Mak BKW. Multitask learning of deep neural networks for low resource speech recognition. *IEEE/ACM Trans Audio Speech Lang Process* 2015; 23: 1172–1183.
41. Narhe MC and Nagmode MS. Vehicle classification using SIFT. *Int J Eng Res Technol* 2014; 3: 1735–1738.
42. Cimpoi M, Maji S, Kokkinos I, et al. Deep filter banks for texture recognition, description, and segmentation. *Int J Comput Vis* 2016; 118: 65–94.
43. Kataoka H, Iwata K and Satoh Y. Feature evaluation of deep convolutional neural networks for object recognition and detection. arXiv preprint arXiv:1509.07627, 2015.
44. Harsha SS and Anne KR. Gaussian mixture model and deep neural network based vehicle detection and classification. *Int J Adv Comput Sci Appl* 2016; 7: 17–25.
45. Fei-Fei L, Fergus R and Perona P. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Comput Vis Image Underst* 2007; 106: 59–70.
46. Fei-Fei L, Fergus R and Perona P. One-Shot learning of object categories. *IEEE Trans Pattern Anal Mach Intell* 2006; 28: 594–611.
47. Dollar P, Wojek C, Schiele B, et al. Pedestrian detection: an evaluation of the state of the art. *IEEE Trans Pattern Anal Mach Intell* 2012; 34: 743–761.
48. Vedaldi A and Lenc K. MatConvNet – convolutional neural networks for MATLAB. In: *Proceedings of ACM international conference on multimedia (ACMMM)*, Brisbane, Queensland, Australia, 26–30 October 2015, pp.1–4. New York: ACM.
49. Vedaldi A and Fulkerson B. VLFeat: an open and portable library of computer vision algorithms, http://www.vlfeat.org/ (accessed 1 January 2016).

50. Bo L, Re X and Fox D. Multipath sparse coding using hierarchical matching pursuit. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Portland, Oregon, USA, 23–28 June 2013, pp.660–667. Piscataway, NJ: IEEE.
51. Jianchao Y, Kai Y, Yihong G, et al. Linear spatial pyramid matching using sparse coding for image classification. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Miami, FL, USA, 20–25 June 2009, pp.1794–1801. Piscataway, NJ: IEEE.
52. Zeiler MD and Fergus R. Visualizing and understanding convolutional networks. In: *European conference on computer vision (ECCV)* (ed D Fleet, et al.), Zurich, Switzerland, 1–5 September 2014, Part I, LNCS 8689, pp.818–833. Berlin: Springer.
53. He KK, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *European conference on computer vision (ECCV)* (ed D Fleet, T Pajdla, B Schiele, et al.), Zürich, Switzerland, 1–5 September 2014, LNCS 8691, pp.346–361. Berlin: Springer.
54. Chatfield K, Simonyan K, Vedaldi A, et al. Return of the devil in details: delving deep into convolutional nets. In: *Proceedings of the British machine vision conference (BMVC)*, Nottingham, UK, 1–5 September 2014, pp.3626–3633. BMVA Press.
55. Sun S. Local within-class accuracies for weighting individual outputs in multiple classifier systems, *Pattern Recogn Lett* 2010; 31: 119–124.
56. Sermanet P, Kavukcuoglu K, Chintala S, et al. Pedestrian detection with unsupervised multi-stage feature learning. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Portland, OR, USA, 23–28 June 2013, pp.3626–3633. Piscataway, NJ: IEEE.
57. Maji S, Berg A and Malik J. Classification using intersection kernel support vector machines is efficient. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Anchorage, Alaska, USA, 23–28 June 2008, pp.1–8. Piscataway, NJ: IEEE.
58. Ouyang W and Wang X. Joint deep learning for pedestrian detection. In: *Proceedings of IEEE international conference on computer vision (ICCV)*, Sydney, NSW, Australia, 1–8 December 2013. Piscataway, NJ: IEEE.
59. Luo P, Zeng X, Wang X, et al. Switchable deep network for pedestrian detection. In: *Proceedings of the British machine vision conference (CVPR)*, Columbus, Ohio, 23–28 June 2014 pp.899–906. Piscataway, NJ: IEEE.
60. Yan J, Zhang X, Lei Z, et al. Robust multi-resolution pedestrian detection in traffic scenes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Portland, Oregon, USA, 23–28 June 2013, pp.3033–3040. Piscataway, NJ: IEEE.
61. Hosang M, Omran R, Benenson, et al. Taking a deeper look at pedestrians. In: *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015 pp.4073–4082. Piscataway, NJ: IEEE.
62. Angelova A, Krizhevsky A, Vanhoucke V, et al. Real-time pedestrian detection with deep network cascades. In: *Proceedings of the British machine vision conference (BMVC)*, Swansea, UK, 7–10 September 2015, pp.1–12. Swansea: BMVA Press.
63. Benenson R, Omran M, Hosang J, et al. Ten years of pedestrian detection, what have we learned? In: Agapito L, Bronstein M and Rother C. (eds) *Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science*, Berlin: Springer, 2014, vol. 8926, pp.613–627.

Author biographies

Ayşegül Uçar received a BS degree, MS degree, and PhD degree from the Electrical and Electronics Engineering Department at the University of Firat of Turkey in 1998, 2000, and 2006, respectively. In 2013, she was a visiting professor at Louisiana State University in the USA. She has been an associate professor in the Department of Mechatronics Engineering since 2009. Her research interests include modeling and control, artificial intelligence systems, robotics vision, pattern recognition, and signal processing.

Yakup Demir received a BS degree, MS degree, and PhD degree from the Electrical and Electronics Engineering Department at the University of Firat of Turkey in 1986, 1990, and 1992, respectively. He is currently a professor in the Electrical and Electronics Engineering Department at the University of Firat of Turkey. His research interests include circuits and systems.

Cüneyt Güzeliş received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from İstanbul Technical University, İstanbul, Turkey, in 1981, 1984, and 1988, respectively. He was with İstanbul Technical University from 1982 to 2000 where he became a full professor. He worked between 1989 and 1991 in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California, as a visiting researcher and lecturer. He was with the Department of Electrical and Electronics Engineering from 2000 to 2011 at Dokuz Eylül University, İzmir, Turkey. He was with İzmir University of Economics, Faculty of Engineering and Computer Sciences, Department of Electrical and Electronics Engineering from 2011 to 2015. He is currently working in Yaşar University, Faculty of Engineering, Department of Electrical and Electronics Engineering. His research interests include artificial neural networks, biomedical signal and image processing, nonlinear circuits-systems, and control, and educational systems.