Improving the Generalization Capacity of Cascade Classifiers

Oswaldo Ludwig, Urbano Nunes, *Senior Member, IEEE*, Bernardete Ribeiro, *Member, IEEE*, and Cristiano Premebida

***Abstract*—The cascade classifier is a usual approach in object detection based on vision, since it successively rejects negative occurrences, e.g., background images, in a cascade structure, keeping the processing time suitable for on-the-fly applications. On the other hand, similar to other classifier ensembles, cascade classifiers are likely to have high Vapnik-Chervonenkis (VC) dimension, which may lead to overfitting the training data. Therefore, this work aims at improving the generalization capacity of the cascade classifier by controlling its complexity, which depends on the model of their classifier stages, the number of stages, and the feature space dimension of each stage, which can be controlled by integrating the parameter setting of the feature extractor (in our case an image descriptor) into the maximum-margin framework of support vector machine training, as will be shown in this paper. Moreover, to set the number of cascade stages, bounds on the false positive rate (FP) and on the true positive rate (TP) of cascade classifiers are derived based on a VC-style analysis. These bounds are applied to compose an enveloping receiver operating curve (EROC), i.e., a new curve in the TP–FP space in which each point is an ordered pair of upper bound on the FP and lower bound on the TP. The optimal number of cascade stages is forecasted by comparing EROCs of cascades with different numbers of stages.**

***Index Terms*—Cascade classifier, maximal margin principle, pattern recognition, pedestrian detection, statistical learning.**

I. Introduction

CASCADES of linear support vector machines (SVM) are able to perform nonlinear classification under low computational requirements, since this approach avoids the high computational cost associated to the nonlinear kernels. Moreover, the cascade classifier successively rejects negative occurrences in a cascade structure, being for instance suitable in vision-based object detection, in which the most usual approach is to scan the image frame by using a sliding window, generating thousands of negative occurrences for each positive image.

Object detection involves several tasks, such as image enhancement, segmentation, scanning, and non-maximal suppression algorithms. This paper focuses on the learning framework, presenting the current stage of our research work on on-the-fly pedestrian detection by using cascade classifiers. Along the last two years, we have been increasing the performance of our methods [1] through theoretical and methodological contributions, which are listed below:

1) a new maximum-margin (MM) training method that integrates the parameter setting of the feature extractor (in our case, an image descriptor) into the MM framework of SVM training, by using a regularization technique that minimizes the number of features (Section III-C and D);
2) a modified genetic algorithm (GA), adapted to deal with chromosomes with different numbers of genes, whose population is composed by different species that cannot reproduce together, implying in competitions within and between species (Section III-E);
3) a new training method for cascades of linear classifiers that trains all the stages at once, in order to avoid the threshold adjustment after training, which usually decreases the overall performance (Section IV);
4) derivation of a growth function for cascade classifiers (Theorem 1 of Section V);
5) derivation of bounds on the false positive rate (FP) and true positive rate (TP), and their application on cascade classifiers through the growth function mentioned in the previous item (Theorem 2 and 3 of Section V);
6) the enveloping receiver operating curve (EROC), a new kind of ROC whose points are ordered pairs of upper bound on the FP and lower bound on the TP, useful in choosing the optimal number of cascade stages (Section V).

As a case study, the proposed training method is applied in pedestrian detection, [2]–[4], aiming at improving the detection rate in outdoor environment.

The paper is organized as follows. Section II presents the state of the art; Section III introduces a new GA-based MM training that automatically controls the number of features, which is extended to deal with the whole cascade classifier at once in Section IV. The adopted classifier model is analysed in Section V, yielding the EROC, a project tool in choosing the optimal number of stages. Experiments on two-well known benchmark data sets are presented in Section VI. Finally, Section VII presents the conclusions.

Manuscript received December 1, 2011; revised September 6, 2012; accepted December 18, 2012. Date of publication February 11, 2013; date of current version November 18, 2013. This work was supported by the Portuguese Foundation for Science and Technology (FCT) and COMPETE program (co-founder by FEDER) under Grant PTDC/EEA-AUT/113818/2009. Oswaldo Ludwig was supported by FCT under grant SFRH/BD/44163/2008. This paper was recommended by Editor E. C. C. Tsang.

O. Ludwig, U. Nunes, and C. Premebida are with the ISR-Institute of Systems and Robotics, University of Coimbra Polo II, 3030-290 Coimbra, Portugal (e-mail: oludwig@isr.uc.pt; urbano@isr.uc.pt; cpremebida@isr.uc.pt).

B. Ribeiro is with the Informatics Engineering Department, Faculty of Science and Technology, University of Coimbra, 3030-790 Coimbra, Portugal (e-mail: bribeiro@dei.uc.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2240678

II. State of the Art

Generalization is one of the most widely studied problems in machine learning. A mathematical formalism for the generalization problem was proposed in [5]; however, the study of distribution-free learnability, more specifically, non-parametric methods for pattern recognition, started earlier with the pioneer work [6], which studied the uniform convergence of relative frequencies of events to their probabilities as function of the hypothesis space complexity and the cardinality of the training data set. Then, the work [7] introduced the VC dimension, a simple combinatorial parameter of the class of concepts to be learned that measures the capacity of the set of hypothesis, in order to deal with infinite hypothesis spaces. Such parameter enables to place bounds on the expected classification risk, even in case of infinite hypothesis spaces [8]. During almost three decades statistical learning theory, was a purely theoretical analysis of the problem of function estimation from a given collection of data [9]. However, in 1992, statistical learning theory became not only a tool for the theoretical analysis but also a tool for creating practical algorithms for non-parametric pattern recognition. Namely, the MM principle, which underlies the SVM, was introduced in [10], where it was proposed a generic training algorithm that maximizes the margin between the training patterns and the decision boundary. The MM framework has been applied in training other classifier models beyond the SVM, such as neural networks [11]. In [12], the generalization problem is formulated in the framework of the bias-variance dilemma, which decomposes the expected classification error into a bias term and a variance term. Assuming an infinite supply of independent training data sets, the bias term measures how closely the learning-algorithm average guess matches the target (averaged over all training data sets). The variance term measures how much the learning-algorithm guess bounces around for the different training data sets, that is, it measures how consistent the classifier decisions are. Such analysis shows that a classifier space with a high capacity, i.e., high VC dimension, is likely to have low bias, but large variance. On the other hand, a classifier space with a low capacity usually have a low variance but a large bias. The works [13] and [14] extended the analysis introduced in [12] in such a way to deal with the usual zero-one loss functions. The introduction of the bias-variance dilemma supported the idea that by composing an ensemble of multiple classifiers, it is possible to reduce the variance term without affecting the bias term. Some works, such as [15] and [16], have provided empirical evidence that an ensemble of classifiers is often better than single classifiers. One of the most known classifier ensemble is the AdaBoost [17], a method derived from the multiplicative weight-update technique proposed in [18]. In [18] and [17], a theoretical analysis of the proposed algorithms is provided; however, a more comprehensive analysis on classifier ensembles is presented in [19], which introduces bounds on the expected risk for bagging classifiers.

The present work also deals with classifier ensembles; however, it focuses on the cascades of linear SVMs, because such classifier ensemble is particularly important for machine vision applications, such as object detection, since it is possible to combine classifiers in a cascade structure, focusing the attention on promising regions of the image [20], saving processing time.

Apart of the usual approaches on cascade classifiers, we highlight the seminal work of Viola and Jones [20], where a boosted cascade classifier scheme is proposed. This approach can be viewed as an object specific focus-of-attention mechanism that discards regions which are unlikely to contain the objects of interest. A cascade classifier scheme based on a specific formulation of the expectation-maximization algorithm was proposed in [21], in order to allow the unsupervised estimation of both the class-conditional density functions and the prior joint probabilities of classes. The proposed technique also allows to include in the estimation process additional prior information.

Regarding image descriptors, gradient-based descriptors have been showing high gains over intensity-based descriptors, whose features are not invariant to light intensity shifts. The most usual gradient-based descriptor was introduced by Dalal and Triggs [4], who developed the Histograms of Oriented Gradients (HOG) inspired on the discriminatory property of local position-dependent gradient orientation histograms used in the SIFT descriptor [22]. Most of modern object detectors make use of some form of HOG descriptor [23], which became less time consuming after the use of integral histograms [24]. Practical experience has shown that the HOG is specially useful for robust detection of humans in natural environments, which is the case study of this work. Despite those developments, the automatic setting of image descriptors is still an open field that might bring a strong improvement in the efficiency of detection systems (see, for instance, Fig. 4 of [25]). In this sense, this work aims at exploiting the MM training, in such a way to integrate the descriptor's parameter setting into the classifier's learning framework, by studying a regularization metric for the HOG descriptors and an optimization method able to deal with both models at once, i.e., able to optimize the discrete parameters of the descriptor and the continuous parameters of the classifier.

III. Controlling the Feature Space Dimension

This section presents the model of a single cascade stage, composed by image descriptor and classifier, specifically HOG and linear SVM. Moreover, it is also presented a GA-based MM training method able to jointly optimize the parameters of both HOG and SVM. The resulting model is so named MMHOG-SVM. This training method will be extended, in the next section, to deal with the whole cascade at once. The idea of the present section is to introduce a new approach in controlling the classifier complexity, in Vapnik sense, by automatically controlling the feature space dimension. To do so, a regularization technique that minimizes the number of features is introduced, in such a way to integrate the parameter setting of the image descriptor into the MM framework of SVM training. Therefore, the algorithm optimizes a tradeoff between the number of features and a SVM-like margin constraint.

A. Image Descriptor

This work applies HOG descriptors composed by histogram channels calculated over rectangular cells, i.e., R-HOG, by the computation of unsigned gradient. The cells overlap half of

their area, meaning that each cell contributes more than once to the final feature vector. In order to account for changes in illumination and contrast, the gradient strengths are locally normalized, i.e., normalized over each cell, through $L2$-norm. The Matlab source code of the HOG descriptor applied in this work was made available for download at Matlabcentral.[1] In this paper, there are three HOG parameters to be tuned; the first one is the number of histogram bins, n_b, and the last two parameters, n_{ph} and n_{pv}, are related to the cells arrangement, i.e., these parameters define a grid of $n_{ph} \times n_{pv}$ cells.

B. Training Linear SVMs by GA

To better understand our GA-based training method (Section III-E), it is convenient to take into account the usual modeling of the soft margin linear SVM optimization problem by GA, as follows:

$$\min_{\mathbf{w},\xi_i} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right) \tag{1}$$

subject to

$$\forall i | y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{2}$$

$$\forall i | \xi_i \geq 0 \tag{3}$$

where n is the cardinality of the training data set, $\mathbf{w}$ and b compose the separating hyperplane, C is a constant, y_i is the target class of the ith training example, and ξ_i are slack variables, which measure the degree of misclassification of the vector $\mathbf{x}_i$. The optimization is a tradeoff between a large margin $(\min \|\mathbf{w}\|^2)$, and a small error penalty $(\min C\sum_{i=1}^{n} \xi_i)$.

The constrained optimization problem (1)–(3) can be replaced by the equivalent unconstrained optimization problem (4) [26], that has the discontinuous objective function Φ, which disables gradient-based optimization methods; therefore, a real-coded GA is applied to solve (4), using Φ as fitness function [27]

$$\min_{\mathbf{w},b} \Phi \tag{4}$$

where

$$\Phi = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H(y_i\hat{y}_i) \tag{5}$$

and $H(t) = \max(0, 1-t)$ is the Hinge loss.

Note that the last term of (5) penalizes models whose estimated outputs do not fit the constraint $y_i\hat{y}_i \geq 1$, in such a way as to save a *minimal margin*, while the minimization of the first term of (5) aims at the enlargement of such *minimal margin*.

Standard SVM training algorithms have $O(m^3)$ time and $O(m^2)$ space complexities, where m is the number of training examples. Therefore, despite the good suitability for small training data sets, standard SVM training algorithms are often highly time consuming for large training data sets. On the other hand, SVM training based on GA has time and space complexities $O(m)$, which means that it can be the best approach for training data sets with large cardinality, which is usual in the case of object detection.

[1] http://www.mathworks.com/matlabcentral/fileexchange/28689-hog-descriptor-for-matlab

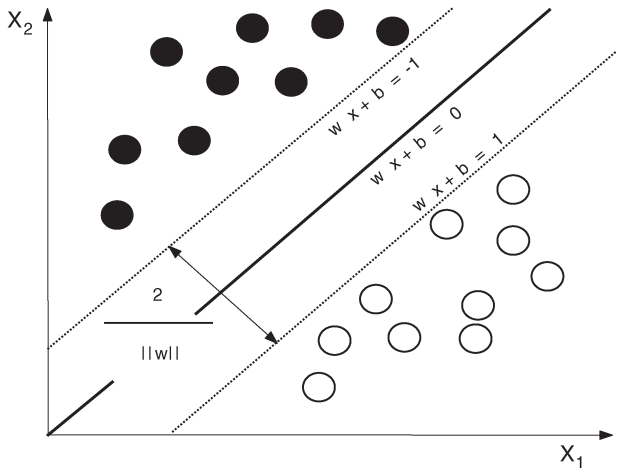

Fig. 1. SVM separating surface and corresponding margins.

C. Regularization Method for the MMHOG-SVM

To include the HOG setting into the MM training, a regularization method is required, similarly to the first term of (5), in such a way to avoid overfitting. Therefore, this work proposes a regularization method for HOG that takes into account the classifier complexity, in Vapnik sense. The VC dimension of a linear classifier, such as the one used in this work, is $h = N_p + 1$, where N_p is the number of features. Taking into account that the number of features depends on the HOG parameters, i.e., $N_p = n_b \times n_{ph} \times n_{pv}$, this work proposes the minimization of N_p, in order to regularize the ensemble $descriptor + classifier$, by decreasing its complexity.

Moreover, the number of features, N_p, has a strong influence on the SVM margin. Let K denotes the field of real numbers and $K^{1\times N_p}$ a vector space, containing all vectors with N_p elements with entries in K, of which $K^{1\times(N_p-1)}$ is a subspace. Let us consider two sets of SVM models $S_1 = \{f(\mathbf{w}, b), \mathbf{w} \in K^{1\times N_p}\}$ and $S_2 = \{f(\mathbf{w}^*, b), \mathbf{w}^* \in K^{1\times(N_p-1)}\}$, where $\mathbf{w} = [w_1, \ldots, w_{N_p-1}, w_{N_p}]$ and $\mathbf{w}^* = [w_1, \ldots, w_{N_p-1}]$, i.e., $\mathbf{w}^*$ is a subvector of $\mathbf{w}$, therefore

$$\|\mathbf{w}^*\| \leq \|\mathbf{w}\|. \tag{6}$$

Taking into account that the SVM margin, m, illustrated in Fig. 1 as the distance between the pointed lines, is given by

$$m = \frac{2}{\|\mathbf{w}\|}. \tag{7}$$

it is possible to state that the margin of $f(\mathbf{w}^*, b)$ is larger or equal to the margin of $f(\mathbf{w}, b)$. Note that this property is true for any natural value of N_p. Therefore, we can state that the larger the number of removed features, the larger the SVM margin; however, the harder is the task of minimizing the second term of (1), i.e., the punishing term. Therefore, our joint MM training has to take into account a punishing parameter, C, similar to the usual SVM training, in order to control the tradeoff between the margin width and the margin constraint.

D. Including HOG Parameter Setting Into the SVM Training

Once it was defined a regularization method for HOG, it is possible to model the optimization problem of the joint MM training, inspired in (4), as follows:

$$\min_{\mathbf{w},b,n_b,n_{ph},n_{pv}} \Omega \tag{8}$$

where

$$\Omega = N_p + \|\mathbf{w}\| + \frac{C}{n}\sum_{i=1}^{n} H(y_i\hat{y}_i). \tag{9}$$

$N_p = n_b \times n_{ph} \times n_{pv}$, $\hat{y}_i = \mathbf{w}\cdot\mathbf{x}_i + b$, and $\mathbf{x}_i$ is a feature vector, obtained directly from the ith training image, Im_i, by using the HOG algorithm with the parameters n_b, n_{ph}, n_{pv}, here defined by the function $\mathbf{x}_i = HOG(Im_i, n_b, n_{ph}, n_{pv})$.

E. Training MMHOG-SVM by GA

This work introduces a modified GA, adapted to deal with chromosomes with different numbers of genes, i.e., different numbers of classifier parameters, according to the HOG setting. Therefore, the population is composed of different species, which cannot reproduce together. The computational resources are bounded by a fixed population, which implies competitions within species and between species. The species whose individuals reach the largest fitness values has larger probability of reproduction, dominating the population in the long term.

Regarding the chromosome structure, the first three genes represent the HOG parameters, n_b, n_{ph}, and n_{pv}. The following $n_b \times n_{ph} \times n_{pv}$ genes represent the elements of the classifier vector $\mathbf{w}$, and the last gene represents the classifier bias b. There are different combinations of n_b, n_{ph}, and n_{pv} that yields the same number of features; therefore, each specie contains individuals with different HOG settings, which can reproduce together.

The algorithm starts by randomly generating the initial population of N_{pop} individuals in a uniform distribution among the species. During the loop over generations, the fitness value of each individual is evaluated on the training data set, which is composed by images with their respective labels. To do so, the HOG descriptor is applied with its parameters instantiated according to the first three chromosome genes. Then, the individuals are ranked according to their fitness values, and the crossover operator is applied in a special way; each new individual is generated by randomly selecting the parents by their ranks, according to the random variable $p \in [1, N_{pop}]$ proposed in our previous work [28]

$$p = (N_{pop} - 1)\frac{e^{a\vartheta} - 1}{e^a - 1} + 1 \tag{10}$$

where $\vartheta \in [0, 1]$ is a random variable with uniform distribution and $a > 0$ sets the selective pressure, more specifically, the larger a, the larger the probability of low values of p, which are related to high-ranked individuals.

If the parents belong to different species, the selection process is repeated until the algorithm finds both parents from the same specie, in such a way to enable the crossover. The first three genes are not combined; these genes are copied from one of the parents that is randomly selected, in order to generate a new individual of the same specie of the parents. Algorithm 1 details the proposed optimization process.

Algorithm 1 Joint GA training of the MMHOG-SVM

Input: Im, y: cubic matrix containing n training images and their respective labels;
C: regularization hyperparameter;
a: selective pressure;
max_{gener}: maximum number of generations;
N_{pop}: population size

Output: n_b, n_{ph}, n_{pv}, $\mathbf{w}$, and b: HOG and SVM parameters

1: randomly generate a set of N_{pop} chromosomes, $\{Cr\} = \{[n_b, n_{ph}, n_{pv}, w_1, w_2, \ldots, w_{(n_b\times n_{ph}\times n_{pv})}, b]\}$, uniformly distributed among species, for the initial population;
2: **for** $generation = 1$: max_{gener} **do**
3: **for** $ind = 1$: N_{pop}; **do**
4: obtain the HOG and SVM parameters of individual ind from Cr^{ind};
5: **for** $i = 1$: n **do**
6: calculate $\mathbf{x}_i$ by applying HOG, with the parameters of chromosome Cr^{ind}, on the image Im_i;
7: calculate $\hat{y}_i = \mathbf{w}\cdot\mathbf{x}_i + b$, using the $\mathbf{w}$ and b of the individual ind;
8: **end for**
9: calculate the fitness, Ω, of individual ind, according to (9), using y and the set of estimated outputs $\{\hat{y}_i\}$, calculated in steps 5–8;
10: $\Omega_{ind} \leftarrow \Omega$: storing the fitness of individual ind;
11: **end for**
12: rank the individuals according to their fitness Ω_{ind};
13: store the genes of the best individual in Cr^{best};
14: **for** $ind = 1$: N_{pop}; **do**
15: **while** parent chromosomes have different dimension **do**
16: $\vartheta_j \leftarrow$ random number $\in [0, 1]$ with uniform distribution, $j = (1, 2)$;
17: $parent_j \leftarrow round((N_{pop}-1)(e^{a\vartheta_j}-1/e^a-1)+1)$, $j=(1,2)$: randomly selecting the indexes of parents by using the asymmetric distribution proposed in [28];
18: compare dimensions of parent chromosomes;
19: **end while**
20: $\eta \leftarrow$ random number $\in \{1,2\}$ with uniform distribution;
21: $Cr^{son}_{(ind,1:3)} \leftarrow Cr_{(parent_\eta,1:3)}$: coping HOG parameters, i.e., the three first genes, from one of the parents, who is randomly selected;
22: **for** $j = 4$: chromosome dimension **do**
23: $\eta \leftarrow$ random number $\in [0,1]$ with uniform distribution;
24: $Cr^{son}_{(ind,j)} \leftarrow \eta Cr_{(parent_1,j)} + (1-\eta)Cr_{(parent_2,j)}$: calculating the jth gene by means of weighted average, in order to compose the chromosome of the individual ind of the new generation;
25: **end for**
26: **end for**
27: **end for**
28: obtain the HOG and SVM parameters from Cr^{best};

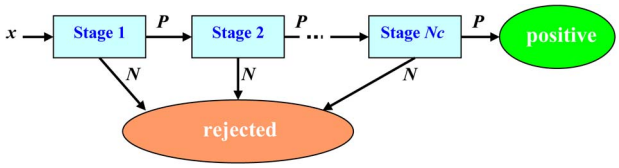

Fig. 2. Cascade classifier scheme.

IV. Cascading MMHOG-SVMs

Cascading linear SVMs is an usual approach in performing nonlinear classification[2] under the MM criterion. Moreover, given that object detection by sliding window usually generates highly unbalanced data sets, the sequential negative rejection, usual in the case of cascade classifiers, can speed up the detection process.

The cascade classifier can be seen as a degenerate decision tree, i.e., a positive result from the first classifier triggers the evaluation of a second classifier. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the pattern, according to the scheme shown in Fig. 2. The cascade classifier applied in the present work is summarized in Algorithm 2.

Algorithm 2 Computing the estimated output of the cascade

Input: Im, $n_{(b,j)}$, $n_{(ph,j)}$, $n_{(pv,j)}$, $\mathbf{w}_j$, and b_j, with $j=1, \ldots, N_s$: image matrix and MMHOG-SVM parameters of all N_s stages;

Output: $\hat{y}$: estimated output;

1: $\mathbf{x} \leftarrow HOG(Im, n_{(b,1)}, n_{(ph,1)}, n_{(pv,1)})$;
2: $\hat{y} \leftarrow \mathbf{w}_1\mathbf{x} + b_1$;
3: $j \leftarrow 1$;
4: **while** $\hat{y} > 0$ **and** $j < N_s$; **do**
5: $\quad j \leftarrow j + 1$;
6: $\quad \mathbf{x} \leftarrow HOG(Im, n_{(b,j)}, n_{(ph,j)}, n_{(pv,j)})$;
7: $\quad \hat{y} \leftarrow \mathbf{w}_j\mathbf{x} + b_j$;
8: **end while**

The usual training process [20] constructs the cascade iteratively, i.e., for each stage of the cascade, a new training data set is generated by aggregating the original set of pedestrian examples with false positives of the currently trained cascade, collected out of a set of randomly extracted non pedestrian examples. Then, the classifier obtained from this new training data set is appended to the cascade.

All the stages, excepting the last one, have to achieve high TP, since an example rejected by a stage cannot be recovered by the next stages. Therefore, the usual training methods for cascade of SVM adjust the classifiers thresholds after the training of each stage, during the iterative cascade training, in order to achieve high TP. Fig. 3 illustrates an example of decision surface generated by a cascade of two linear SVMs trained by usual methods, i.e., each stage trained independently with

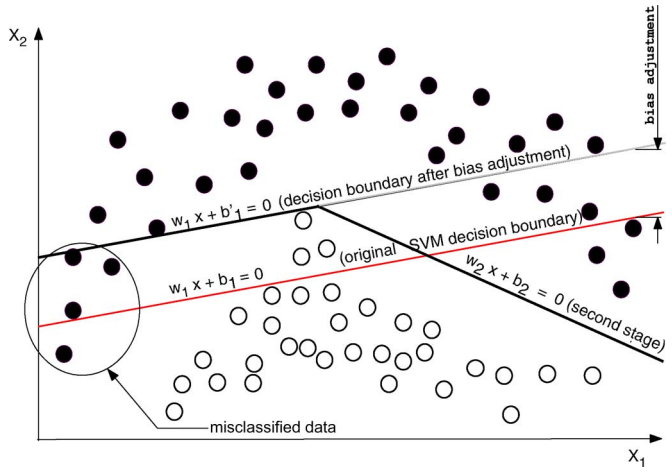

Fig. 3. Example of nonlinear separating surface (solid black lines) of a cascade of linear classifiers trained by usual methods (filled circles represent negative examples).

threshold adjustment. To ease the understanding, it is assumed that both stages are working in the same feature space. Despite the nonlinear decision boundary (solid black lines), some training data are misclassified. The inclusion of more stages could solve the misclassification on training data; however, the bias adjustment (equivalent to a threshold adjustment) affects the original SVM training, which aims at improving the generalization capability by placing the separating surface far from the training data. This problem can be observed in Fig. 3, in which some training examples are quite close to the decision surface of the first stage. Note that this issue could not be overcome by appending new stages.

To overcome the requirement of threshold adjustment after training, we propose a novel joint-training method for cascade of linear SVMs that train all the stages, including the image descriptors, at once, using all of training data. The proposed method applies the GA described in Algorithm 1 on an extended version of the objective function (9), as follows:

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_{N_s},b_1,\ldots,b_{N_s},N_{(p,1)},\ldots,N_{(p,N_s)}} \Omega \tag{11}$$

where

$$\Omega = \sum_{j=1}^{N_s} N_{(p,j)} + \sum_{j=1}^{N_s} \|\mathbf{w}_j\| + \frac{C}{n}\sum_{i=1}^{n} H(y_i\hat{y}_i). \tag{12}$$

$N_{(p,j)} = n_{(b,j)} \times n_{(ph,j)} \times n_{(pv,j)}$ is the number of HOG features used by the stage j, with $j = 1 \ldots N_s$, $\hat{y}_i$ is the estimated output, computed according to Algorithm 2, $\mathbf{w}_j$ and b_j are the weight vector and bias of the SVM used in stage j, and n is the number of training data.

The first two terms of (12) aim at improving the margin,[3] i.e., the dashed lines in Fig. 4, while the last term penalizes models whose estimated outputs, $\hat{y}_i$, do not fit the constraint $y_i\hat{y}_i \geq 1$, i.e., the training data within the dashed lines or in the "wrong" side of the separating surface (black solid lines).

[2]The reader can easily verify that a cascade composed by two linear classifiers can solve the XOR problem.

[3]For details on the relationship between the first term of (12) and the classification margin, see Section III-C.

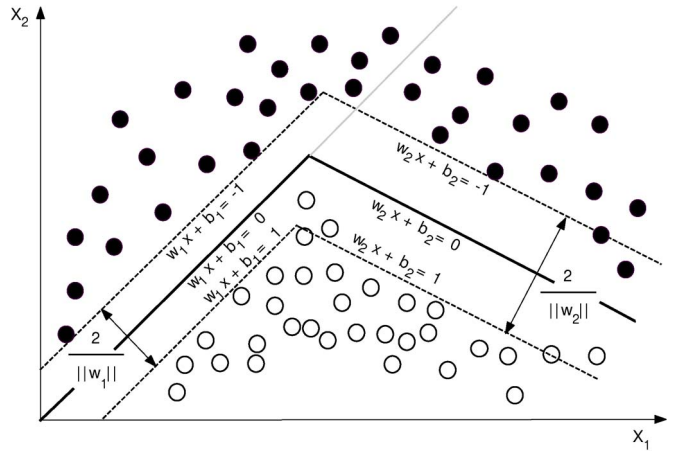

Fig. 4. Example of nonlinear separating surface (black solid lines) and margins (dashed lines) of a cascade of MMHOG-SVM trained by the proposed method.

The proposed joint-optimization method makes possible a better control on the decision boundary, while avoiding threshold adjusting, as can be seen in the example of Fig. 4, which illustrates the same training data of Fig. 3 without misclassifications. However, the larger the capacity of a classifier, the larger the risk of overfitting the data. This fact justifies the inclusion of two regularization methods in our SVM-like training framework, i.e., the first two terms of (12), which, together with the last term of (12), i.e., the punishing term, place the separating surface far from the training data, as can be seen in Fig. 4. Moreover, the proposed method is based on the usual cascade model, i.e., a sequential negative rejection that speeds up the detection process, particularly when compared with SVM with nonlinear kernels.

A. Advantages of Cascading Linear SVMs Over Nonlinear Kernel Functions

The use of SVM with nonlinear kernels [29] enables nonlinear classification by mapping input vectors to a high-dimensional feature space, where a linear decision surface is constructed; however, in [30] the authors show how SVM with polynomial and Gaussian radial basis function (RBF) kernels can have very large, and even infinite, VC dimension, which can affect the generalization performance. On the other hand, a cascade of linear SVMs is able to perform nonlinear classification without using nonlinear kernels, offering the possibility of a better control of the classifier space complexity, as will be shown in the next section. Moreover, SVM with nonlinear kernels is less common in the case of sliding-window detection, since the use of nonlinear kernels turns the SVM decision function, $c(\mathbf{x})$, time expensive, particularly when the number of support vectors, N_{sv}, is high, as follows:

$$c(\mathbf{x}) = \mathtt{sgn}\left(\sum_{i=1}^{N_{sv}} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{13}$$

where α_i and b are SVM parameters, $(\mathbf{x}_i, y_i)$ is the ith support vector data pair, $\mathrm{sgn}(\cdot)$ is 1 if the argument is greater than zero, and -1 if it is less than zero, and $K(\cdot,\cdot)$ is a nonlinear kernel function. The iterative computation (13) becomes more simple if a linear kernel, $(\mathbf{x}_i^T \mathbf{x}_j)$, is used

$$c(\mathbf{x}) = \mathtt{sgn}\left(\sum_{i=1}^{N_{sv}} y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b\right) = \mathtt{sgn}(\mathbf{a} \cdot \mathbf{x} + b) \tag{14}$$

where $\mathbf{a} = \sum_{i=1}^{N_{sv}} y_i \alpha_i \mathbf{x}_i^T$ is calculated once. The right-hand side of (14) does not involve an iterative computation and thus decreases the processing time by, at least, N_{sv} times. Therefore, since this work aims at on-the-fly object detection, cascading linear SVMs is the best option.

V. VC-Style Analysis on a Cascade of Linear Classifiers

Similar to other classifier ensembles, cascade classifiers are likely to have high overfitting the training data. To deal with this problem, it was introduced, in Section III, a method that automatically controls the number of features in each cascade stage; however, since the VC dimension is also function of the number of stages, this section presents a VC-style analysis on cascade of linear classifiers, aiming at forecasting the optimal number of stages. We start deriving, for the first time, a growth function[4] for cascade classifiers. This growth function is applied in the derivation, also for the first time, of VC-style bounds on the FP and TP, which are applied in calculating the EROC, i.e., a new kind of ROC whose points are ordered pairs of upper bound on the FP and lower bound on the TP. The EROC is applied in our experiments as a project tool in choosing the optimal number of stages. We also included an Appendix that briefly introduces some principles of statistical learning theory to support our theorems.

Theorem 1: Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ be a training data set, whose target values are $Y = \{y_1, \ldots, y_l\}$, both generated according to some unknown (but fixed) probability distribution function $F(\mathbf{x}, y)$. Then, the growth function of a cascade classifier with N_s stages of VC dimensions $h_k (k = 1, \ldots, N_s)$ is given by

$$G_{casc} = \log N_{casc} \tag{15}$$

where

$$N_{casc} = \sum_{i_1=0}^{\min(h_1,l)} \left(C_l^{i_1} \sum_{i_2=0}^{\min(h_2,(l-i_1))} \left(C_{(l-i_1)}^{i_2} \times \sum_{i_3=0}^{\min(h_3,(l-i_1-i_2))} \left(C_{(l-i_1-i_2)}^{i_3} \cdots \right. \right. \right.$$
$$\left. \left. \left. \cdots \sum_{i_{N_s}=0}^{\min\left(h_{N_s},\left(l-\sum_{j=1}^{(N_s-1)} i_j\right)\right)} C_{\left(l-\sum_{j=1}^{(N_s-1)} i_j\right)}^{i_{N_s}} \right) \right) \cdots \right). \tag{16}$$

[4]The definition of growth function is in the Appendix.

Proof: According to (31), the largest number of different separations that the first stage can perform is $\sum_{i_1=0}^{\min(h_1,l)} C_l^{i_1}$, where $C_l^{i_1}$ is the number of i_1-combinations from a given set with l elements, while the summation computes the combinations for different numbers of examples per class, up to the classifier capacity, i.e., its VC dimension (or the number of examples, if $h_1 \geq l$). Therefore, one can assume i_1 as the number of positive examples and $l - i_1$ as the number of negative examples, or vice versa.

By definition, see Fig. 2, the second stage acts only on the set of positive examples that pass through the first stage, whose cardinality can be assumed as i_1 or $l - i_1$. As we are interested in the largest value of N_{casc}, and $l - i_1 \geq i_1$, we assume the largest cardinality, i.e., $l - i_1$. Therefore, for each separation performed by the first stage, the second stage can perform $\sum_{i_2=0}^{\min(h_2,l-i_1)} C_{l-i_1}^{i_2}$ different separations on the set of positive examples, i.e., the largest number of different separations that both stages can perform is given by $\sum_{i_1=0}^{\min(h_1,l)}(C_l^{i_1} \sum_{i_2=0}^{\min(h_2,(l-i_1))} C_{(l-i_1)}^{i_2})$. The same reasoning can be applied for the other stages, yielding (16). ■

The following theorems present, for the first time, VC-style bounds on FP and TP.

Theorem 2: Let l be the cardinality of the training data set, n_n be the number of negative examples, and $\{f(\mathbf{x}, \alpha) : X \to R, \forall \alpha \in \Lambda\}$ be a set of functions related to the cascade classifier model, from which $f(\mathbf{x}, \alpha_l)$ is an element whose parameters are represented by the vector α_l. Then, with probability at least $1 - \eta, 0 < \eta < 0.5$, the risk for the indicator function $Q(\mathbf{x}, y, \alpha_l) = L(y, f(\mathbf{x}, \alpha_l))$ that minimizes the functional $FP_{emp} = (1/n_n) \sum_{i=1}^{n_n} Q(\mathbf{x}_i, y_i, \alpha_l)$ satisfies the inequality

$$|FP(\alpha_l) - FP_{emp}(\alpha_l)| \leq \frac{1}{n_n} + \sqrt{\frac{G_{casc}(2l) - \log \frac{\eta}{4}}{n_n}} \quad (17)$$

where $G_{casc}(2l)$ is the growth function of the set of functions for a training data set with cardinality $2l$ (see Theorem 1), $FP(\alpha_l) = \int Q(\mathbf{x}, y, \alpha_l) dF(\mathbf{x}, y)$, and $Q(\mathbf{x}, y, \alpha_l) = 1$ if x is a false positive, otherwise $Q(\mathbf{x}, y, \alpha_l) = 0$.

Proof: Considering any fixed function $Q(\mathbf{x}, y, \alpha^*)$, i.e., $\alpha = \alpha^*$, and taking into account that **the functional FP is based only on the negative examples**, from (37), for any fixed sample size $2l$, with $2n_n$ negative examples, and any two randomly chosen half-samples with the same number of negative examples, the inequality

$$P\left\{\frac{1}{n_n}\left|\sum_{i=1}^{n_n} Q(\mathbf{x}_i, y_i, \alpha^*) - \sum_{i=n_n+1}^{2n_n} Q(\mathbf{x}_i, y_i, \alpha^*)\right| > \epsilon\right\} \leq 2e^{-\epsilon^2 n_n} \quad (18)$$

holds true. Therefore, doing $\epsilon = \epsilon^* - (1/n_n)$, the inequality

$$P\left\{\frac{1}{n_n}\left|\sum_{i=1}^{n_n} Q(\mathbf{x}_i, y_i, \alpha^*) - \sum_{i=n_n+1}^{2n_n} Q(\mathbf{x}_i, y_i, \alpha^*)\right| > \epsilon^* - \frac{1}{n_n}\right\} \leq 2e^{-\left(\epsilon^* - \frac{1}{n_n}\right)^2 n_n} \quad (19)$$

holds true. The above inequality takes into account a fixed indicator function that performs a specific dichotomy on the set of vectors; however, the learning machine has to choose one among a finite set of different dichotomies that the set of indicator functions $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$ can produce on the set of vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_{2l}\}$. This fact characterizes disjunctive events, whose probability is not greater than the sum of their probabilities. Taking into account that the number of distinguishable events, i.e., the number of different dichotomies given by (30), depends on the entire sample size, which in this case is $2l$, for the set of indicator functions $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$, we have

$$\begin{aligned} &P\left\{\frac{1}{n_n}\sup_{\alpha\in\Lambda}\left|\sum_{i=1}^{n_n} Q(\mathbf{x}_i, y_i, \alpha) - \sum_{i=n_n+1}^{2n_n} Q(\mathbf{x}_i, y_i, \alpha)\right| > \epsilon^* - \frac{1}{n_n}\right\} \\ &\leq \sum_{\alpha^*\in\Lambda^*} P\left\{\frac{1}{n_n}\left|\sum_{i=1}^{n_n} Q(\mathbf{x}, y, \alpha^*) - \sum_{i=n_n+1}^{2n_n} Q(\mathbf{x}_i, y_i, \alpha^*)\right| > \epsilon^* - \frac{1}{n_n}\right\} \\ &\leq 2N_{casc}(2l) e^{-\left(\epsilon^* - \frac{1}{n_n}\right)^2 n_n} \\ &\leq 2e^{\left(G_{casc}(2l) - \left(\epsilon^* - \frac{1}{n_n}\right)^2 n_n\right)} \end{aligned} \quad (20)$$

where $\Lambda^* = \Lambda^*(x_1 \ldots, x_{2l})$ is the finite set of distinguishable functions $Q(\mathbf{x}_i, y_i, \alpha^*)$, whose cardinality is $N_{casc}(2l)$. The last inequality of (20) came from (28), which can be rewritten as $N_{casc}(2l) = e^{G_{casc}(2l)}$.

The inequality (20) is related to the absolute difference between two realizations of FP, evaluated on different data sets of the same cardinality. However, we aim at bounding the probability of deviation of the FP evaluated on the training data set from its expected value. Combining (20) with (36), we obtain

$$P\left\{\sup_{\alpha\in\Lambda}\left|E\left(Q(\mathbf{x}, y, \alpha)\right) - \frac{1}{n_n}\sum_{i=1}^{n_n} Q(\mathbf{x}_i, y_i, \alpha)\right| > \epsilon^*\right\} \leq 4e^{\left(G_{casc}(2l) - \left(\epsilon^* - \frac{1}{n_n}\right)^2 n_n\right)} \quad (21)$$

where $E(Q(\mathbf{x}, y, \alpha)) = \int Q(\mathbf{x}, y, \alpha) dF(\mathbf{x}, y)$. Doing

$$\eta := 4e^{\left(G_{casc}(2l) - \left(\epsilon^* - \frac{1}{n_n}\right)^2 n_n\right)} \quad (22)$$

and solving (22) with respect to ϵ^*, yields

$$\epsilon^* = \frac{1}{n_n} + \sqrt{\frac{G_{casc}(2l) - \log \frac{\eta}{4}}{n_n}}. \quad (23)$$

Substituting (22) into (21), yields

$$P\left\{\sup_{\alpha\in\Lambda}\left|E\left(Q(\mathbf{x}, y, \alpha)\right) - \frac{1}{n_n}\sum_{i=1}^{n_n} Q(\mathbf{x}_i, y_i, \alpha)\right| > \epsilon^*\right\} \leq \eta. \quad (24)$$

Rewriting (24) as

$$P\left\{\sup_{\alpha\in\Lambda}\left|E\left(Q(\mathbf{x},y,\alpha)\right)-\frac{1}{n_n}\sum_{i=1}^{n_n}Q(\mathbf{x}_i,y_i,\alpha)\right|\leq\epsilon^*\right\}>1-\eta \tag{25}$$

and by substituting (23) into (25) completes the proof. ■

Theorem 3: With probability at least $1-\eta$, $0<\eta<0.5$, the risk for the function $Q(\mathbf{x},y,\alpha_l)$ which maximizes the functional $TP_{emp}=(1/n_p)\sum_{i=1}^{n_p}Q(\mathbf{x}_i,y_i,\alpha_l)$ satisfies the inequality

$$|TP(\alpha_l)-TP_{emp}(\alpha_l)|\leq\frac{1}{n_p}+\sqrt{\frac{G_{casc}(2l)-\log\frac{\eta}{4}}{n_p}} \tag{26}$$

where $TP(\alpha_l)=\int Q(\mathbf{x},y,\alpha_l)dF(\mathbf{x},y)$ and $Q(\mathbf{x},y,\alpha_l)=1$ if $\mathbf{x}$ is a true positive, otherwise $Q(\mathbf{x},y,\alpha_l)=0$.

Proof: The proof starts by determining the upper bound on the false negative rate (FN) similarly to the proof of Theorem 2. Therefore, the following inequality:

$$|FN(\alpha_l)-FN_{emp}(\alpha_l)|\leq\frac{1}{n_p}+\sqrt{\frac{G_{casc}(2l)-\log\frac{\eta}{4}}{n_p}} \tag{27}$$

holds with probability at least $1-\eta$, where n_p is the number of positive examples. Taking into account that $FN(\alpha_l)=1-TP(\alpha_l)$ and $FN_{emp}(\alpha_l)=1-TP_{emp}(\alpha_l)$, and substituting these equations in the left-hand side of (27), we prove (26). ■

Theorems 1, 2, and 3 make possible to construct the EROC. The idea is to plot EROCs for cascades with numbers of stages varying from one to N_s, adopting the number of stages that corresponds to the most suitable EROC, in terms of area under curve (AUC) or the most suitable shape, depending on the project demands. Beyond the empirical values of TP and FP for different values of threshold, as usual in plotting ROCs, EROC plotting requires the VC dimension of each stage, h_k, the cardinalities of both positive and negative training data sets, n_p and n_n, and the sum of both cardinalities, l (see Theorems 1, 2, and 3).

VI. Experiments

In this paper, two benchmark data sets are applied for evaluating the performance of the proposed methods; the Daimler Pedestrian Classification benchmark [3] and the INRIA Person Data set [4]. Pedestrian classification is more suitable than pedestrian detection in evaluating our learning framework, since the non-maximal suppression algorithm and the scanning settings, such as spatial and scale strides, have strong influence on the performance of the detection algorithm, making difficult to assign credit to the classification module. Therefore, we focus on the classification of cropped images into the classes pedestrian and non-pedestrian.

A. Experiments on Daimler Pedestrian Classification Benchmark

The Daimler Pedestrian Classification benchmark is composed by a collection of 29 400 images for training, divided into three training data sets, and 19,600 images for testing, divided into two testing data sets, beyond 1200 video frames without pedestrians, for extraction of additional negative training and testing examples. Samples are scaled to size 18 × 36 pixels of gray scale, as can be seen in Fig. 5.

Fig. 5. Examples of pedestrian and non-pedestrian images from the Daimler Pedestrian Classification data set [3].

TABLE I
AUCs of Classifiers on Daimler Data Set

HOG-classifier	AUC	AUC_{use}
$HOG-FLDA$	0.9817 ± 0.0184	0.7049 ± 0.0193
$HOG-SVM$	0.9808 ± 0.01751	0.7256 ± 0.0184
$HOG-SVM_{RBF}$	0.9845 ± 0.0279	0.752 ± 0.0282
$MMHOG-SVM$	0.9863 ± 0.0122	0.7851 ± 0.0124
Cascade 2 stages	0.9868 ± 0.0193	0.7885 ± 0.0203
Cascade 3 stages	0.9885 ± 0.0221	0.8219 ± 0.0240
Cascade 4 stages	0.9852 ± 0.0251	0.7786 ± 0.0266

The experiments are according to the benchmarking procedure described in [3], i.e., adjusting the punishing parameter, C, of the joint MM training via threefold cross-validation on the training data sets, generating three different classifiers, each trained by using two out of the three training data sets. Applying these classifiers to both test data sets yields six values of area under ROC curve (AUC), from which mean and variance are computed. Table I summarizes the AUC of seven different classifiers, specifically, a single MMHOG-SVM trained by the joint MM method (see Section III), cascades of MMHOG-SVM with two, three, and four stages, Fisher linear discriminant analysis, linear SVM, and SVM with RBF kernel. The last three classifiers were trained by using features extracted by the HOG descriptor[5] tunned according to our previous work [31]. We define AUC_{use} as the area under the ROC curve in the interval [0, 0.05] of FP, since the optimal operation point for ROC curves usually occurs in this interval.

Fig. 6 illustrates ROCs of a single MMHOG-SVM and of cascades with different numbers of stages. As can be seen, the best cascade configuration for the Daimler data set has three stages, in agreement with the EROC indication, as can be seen in Fig. 7.

The results reported here can be compared with previous works, such as [3], [32], and [33].

In Fig. 8, ROCs of different classifiers are superimposed. The experimental results indicate that the ensemble with three stages has the best performance of all the classifiers we

[5]http://www.mathworks.com/matlabcentral/fileexchange/28689-hog-descriptor-for-matlab

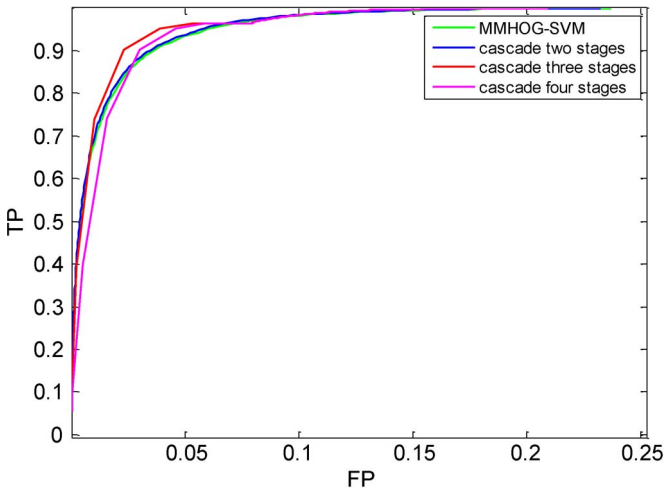

Fig. 6. ROC of a single MMHOG-SVM and ROCs of cascades with different numbers of stages.

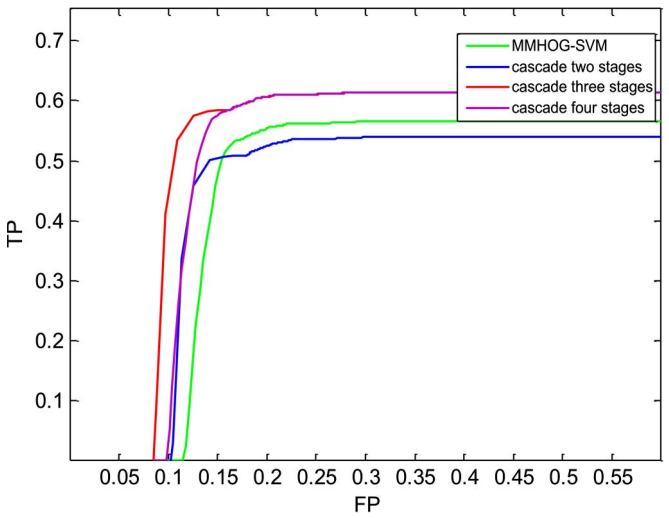

Fig. 7. EROC of a single MMHOG-SVM and EROCs of cascades with different numbers of stages.

evaluated on the Daimler data set, presenting larger AUC than the HOG-SVM with kernel RBF, specially in the case of AUC_{use}. However, the major advantage of the cascade of MMHOG-SVMs over the HOG-SVM with kernel RBF is the low processing time, around 200 times faster, because this approach does not requires nonlinear kernels or a time-expensive descriptor, i.e., due to our regularization method,[6] the MMHOG computes few features.

B. Experiments on INRIA Person Data set

The INRIA Person Data set is composed by a set of images collected from daily life photos, which are divided into training and testing data sets. The training data set contains 2416 cropped images of persons, and 1218 frames without persons, for extraction of negative examples. The testing data set contains 1132 cropped images of persons and 453 frames without persons. Person bounding-boxes are of 64×128 pixels;

[6]See Section III.

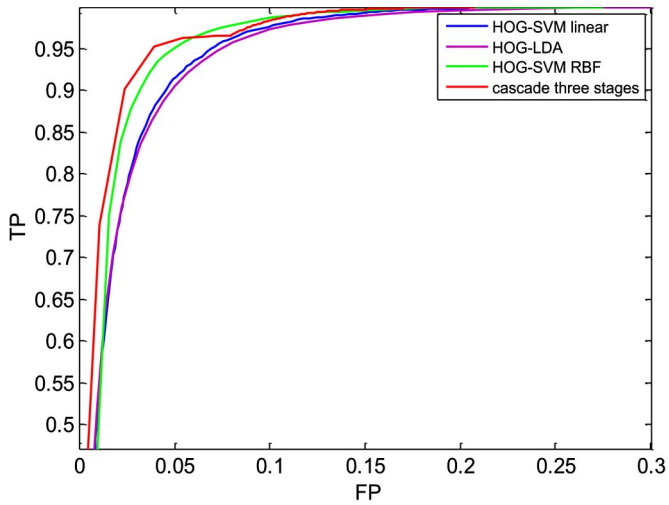

Fig. 8. Comparison between ROCs of different classifiers on Daimler data set.

Fig. 9. Examples of pedestrian and non-pedestrian images from the INRIA data set [4].

however, in the case of the training data set, the positive examples have a margin of 16 pixels in each side of the bounding-box, while in the case of the testing data set the margin is only 3 pixels.

As our focus is pedestrian classification, it was composed a training data set of negative cropped images by scaling down the images without persons under five scales, specifically 1, 1.15, 1.3, 1.56, and 1.92 times. Then, it was randomly extracted two bounding-boxes of 64 × 128 size per scaled image, i.e., ten examples per image, totaling 12,180 negative training examples. Regarding the testing data set, it was collected negative bounding-boxes of size 64 × 128 using an exhaustive sliding-window approach with horizontal and vertical steps of 64 and 128 pixels, and the same scale factors mentioned above. This procedure yielded a total of 23,289 negative testing examples. Fig. 9 illustrates some examples of pedestrian and non-pedestrian from the INRIA database.

Table II summarizes the AUCs of the classifiers evaluated in this work, while Fig. 10 illustrates ROCs of a single MMHOG-SVM and cascades with different numbers of stages. Similar to the experiments on the Daimler data set, the best cascade configuration for the INRIA database has three stages, in agreement with the EROCs of Fig. 11, which indicates an advantage of the cascade with three stages, specially for the lowest values of FP.

Fig. 12 makes possible the comparison of ROCs of different classifiers.

TABLE II
AUC OF CLASSIFIERS ON INRIA DATA SET

HOG-classifier	AUC	AUC_{use}
$HOG-FLDA$	0.9607 ± 0.02105	0.6607 ± 0.02073
$HOG-SVM$	0.9338 ± 0.0211	0.4303 ± 0.0196
$HOG-SVM_{RBF}$	0.9590 ± 0.0281	0.6211 ± 0.0289
$MMHOG-SVM$	0.9354 ± 0.0201	0.4353 ± 0.0182
Cascade 2 stages	0.9379 ± 0.0197	0.4376 ± 0.0212
Cascade 3 stages	0.9720 ± 0.0213	0.7887 ± 0.0244
Cascade 4 stages	0.9684 ± 0.0266	0.7395 ± 0.0270

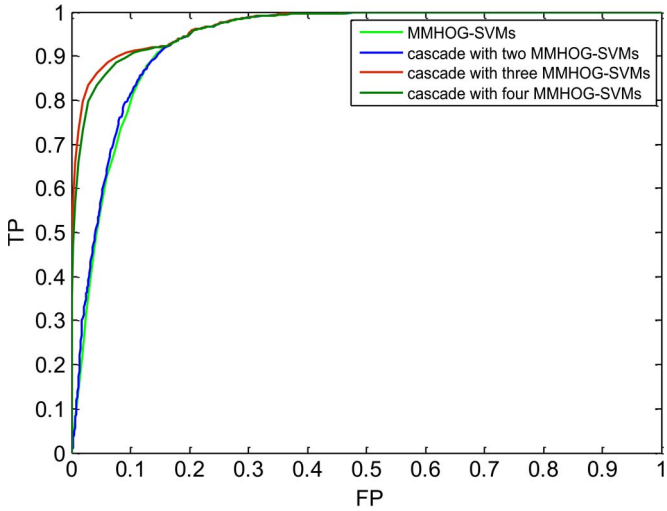

Fig. 10. ROC of a single MMHOG-SVM and ROCs of cascades with different numbers of stages.

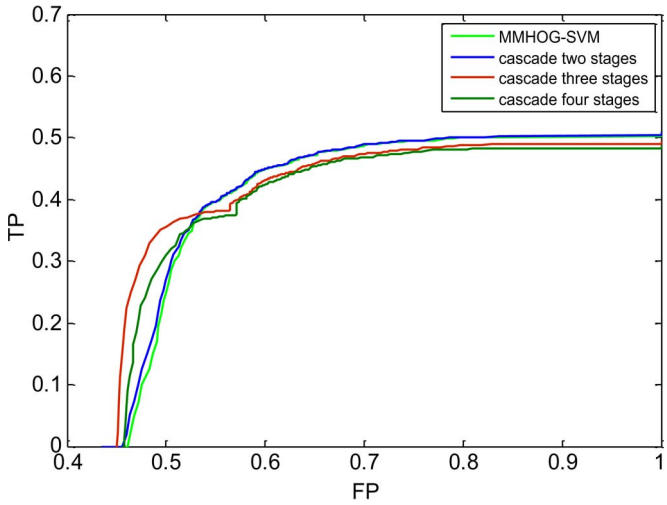

Fig. 11. EROC of a single MMHOG-SVM and EROCs of cascades with different numbers of stages.

VII. CONCLUSION

This work proposed a new training method for cascades of linear classifiers that makes possible a better control on the decision boundary, due to a joint-optimization framework based on the MM criterion, which aims at placing a nonlinear decision boundary far from the training data, minimizing the classification risk. It was also introduced a new approach in controlling the classifier complexity by automatically controlling the feature space dimension through the integration of the parameter setting of the image descriptor into a MM training framework, optimizing a tradeoff between the number of features and a SVM-like margin constraint. Since the VC dimension of liner models is well known, it was introduced a VC-style analysis that places bounds on the generalization capability of the adopted model. This theoretical analysis yielded a new project tool, the EROC, which is useful in choosing the optimal cascade structure, being also interesting to other classifier models. Regarding the optimization method, the modified GA, developed in this work, can be applied in other domains, such as feature selection or even model selection.

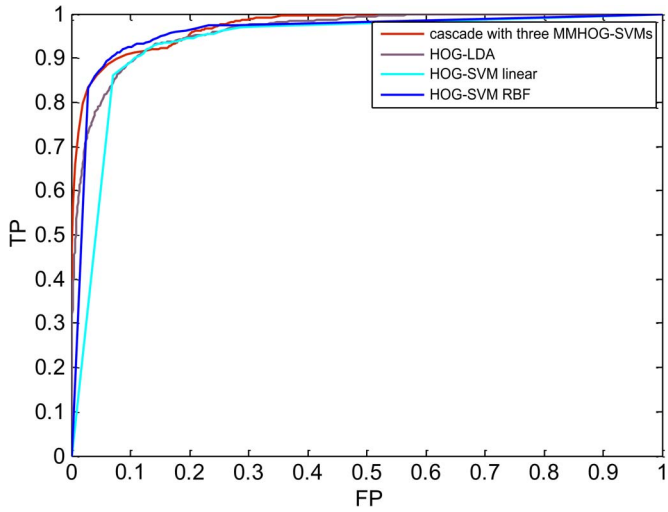

Fig. 12. Comparison between ROCs of different classifiers on INRIA data set.

The experimental results indicate that our methods can yield fast and accurate compositions of descriptor-classifier. The cascade of MMHOG-SVMs is able to perform nonlinear classification, while avoiding the high computational cost of nonlinear kernels, being around 200 times faster than the SVM-RBF. Moreover, it was observed that the cascade of MMHOG-SVMs was slightly more accurate than the SVM-RBF in our case study.

APPENDIX

This appendix briefly introduces some principles of statistical learning theory, in order to support our theorems, which require the following definition and lemmas.

Definition 1: We define $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ as a set of random vectors, generated independently and identically distributed, and $Y = \{y_1, \ldots, y_l\}$ as a set containing the respective target values (given by some supervisor that acts in the same environment of the learning machine), both generated according to some unknown (but fixed) probability distribution function $F(\mathbf{x}, y)$.

Definition 2: We define Λ as a set of parameters and $\{f(\mathbf{x}, \alpha) : X \to R, \forall \alpha \in \Lambda\}$ as a set of functions related to the classifier model, from which $f(\mathbf{x}, \alpha^*)$ is an element, i.e., a specific model in which $\alpha = \alpha^*$.

Definition 3: We define the set of indicator functions $\{Q(\mathbf{x}, y, \alpha) : X \to \{0,1\} | Q(\mathbf{x}, y, \alpha) = L(y, f(\mathbf{x}, \alpha)), \forall \alpha \in \Lambda\}$, from which $Q(\mathbf{x}, y, \alpha^*)$ is an element in which $\alpha = \alpha^*$, whose average value, $(1/l)\sum_{i=1}^{l} Q(\mathbf{x}_i, y_i, \alpha^*)$, defines some performance index, such as classification accuracy, FP, or TP.

Definition 4: The growth function is defined as the quantity

$$G(l) = \log\left(N(\mathbf{x}_1, \ldots, \mathbf{x}_l)\right) \tag{28}$$

where $N(\mathbf{x}_1, \ldots, \mathbf{x}_l) \leq 2^l$ is the largest number of different separations[7] that the functions of the set $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$ can produce on the set of vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$.

Definition 5: [8] The VC dimension of a set of indicator functions, defined as $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$, is equal to the largest number h of vectors that can be separated into two different classes in all 2^h possible ways by using this set of functions (i.e., the VC dimension, h, is the maximum number of vectors that can be shattered by the set of functions).

The following lemma bounds the growth function taking into account both the cardinality of the training data set and the capacity of the set of indicator functions, i.e., its VC dimension.

Lemma 1: [8] The growth function of a set of indicator functions, defined as $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$, satisfies the relationship

$$G(l)\begin{cases} = l\log 2 & \text{if } l \leq h \\ = \log\left(\sum_{i=0}^{h} C_l^i\right) \leq \log\left(\frac{el}{h}\right)^h = h\left(1 + \log\frac{l}{h}\right) & \text{if } l > h \end{cases} \tag{29}$$

where e is the Euler constant and C_l^i is the number of i-combinations from a given set S of l elements.

In other words, the largest number of different separations that the functions of the set $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$ can produce on the set of vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ satisfies the relationship

$$N(\mathbf{x}_1, \ldots, \mathbf{x}_l)\begin{cases} = 2^l & \text{if } l \leq h \\ = \sum_{i=0}^{h} C_l^i & \text{if } l > h. \end{cases} \tag{30}$$

A shorter version of (30) is given by

$$N(\mathbf{x}_1, \ldots, \mathbf{x}_l) = \sum_{i=0}^{\min(l,h)} C_l^i \tag{31}$$

since, if $l \leq h$, $\sum_{i=0}^{\min(l,h)} C_l^i = \sum_{i=0}^{l} C_l^i = 2^l$.

As can be inferred from Lemma 1, even if the set of functions $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$, contains infinitely many elements, only a finite number of clusters of events is distinguishable on the finite set of examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$. Therefore, in order to derive bounds on the expected risk for an infinite set of indicator functions $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$, we take advantage on the relationship stated in the following lemma:

Lemma 2: [8] Let us consider a space of random independent observations of size $2l$, $X^{2l} = \{\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{2l}\}$, which can be divided into two half-samples $X_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ and $X_2 = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{2l}\}$.

[7]in the context of this chapter, separation means the dichotomizing of the exemplars in some given universe of discourse into two groups.

For any function in the set $\{Q(\mathbf{x}, y, \alpha), \alpha \in \Lambda\}$, we determine the frequencies[8]:

$$v(\alpha, X_1) = \frac{1}{l}\sum_{i=1}^{l} Q(\mathbf{x}_i, y_i, \alpha) \tag{32}$$

on the first half-sample, X_1, and

$$v(\alpha, X_2) = \frac{1}{l}\sum_{i=1+1}^{2l} Q(\mathbf{x}_i, y_i, \alpha) \tag{33}$$

on the second half-sample, X_2. Considering the random variables

$$\rho(X^{2l}) = \sup_{\alpha\in\Lambda}\left|\frac{1}{l}\sum_{i=1}^{l} Q(\mathbf{x}_i, y_i, \alpha) - \frac{1}{l}\sum_{i=l+1}^{2l} Q(\mathbf{x}_i, y_i, \alpha)\right| \tag{34}$$

$$\pi(X_1) = \sup_{\alpha\in\Lambda}\left|\int Q(\mathbf{x}, y, \alpha)dF(\mathbf{x}, y) - \frac{1}{l}\sum_{i=1}^{l} Q(\mathbf{x}_i, y_i, \alpha)\right|. \tag{35}$$

Then, the distribution of the random variable $\pi(X_1)$ is connected with the distribution of the random variable $\rho(X^{2l})$ by the inequality

$$P\left\{\pi(X_1) > \epsilon\right\} < 2P\left\{\rho(X^{2l}) > \epsilon - \frac{1}{l}\right\} \tag{36}$$

where $P\{\cdot\}$ means probability.

In simple words, considering the worst-case realization of the parameters α, the absolute value of the difference between two realizations of some performance index, evaluated on different data sets of the same cardinality l, is given by $\rho(X^{2l})$, while $\pi(X_1)$ gives the absolute value of the deviation of that performance index evaluated on a data set of cardinality l from its expected value. We underline the inequality (36), which gives an upper bound on the probability $P\{\pi(X_1) > \epsilon\}$ as function of $\rho(X^{2l})$ that has the convenience of relying on a finite set of examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_{2l}\}$, i.e., a finite number of distinguishable events.

To complete the set of foundations required for our study, it is presented a lemma that makes possible to bound the right-hand term of (36), i.e., the following lemma defines the rate of convergence that connects two relative frequencies[9]:

Lemma 3: [8] For any fixed sample size $2l$, any fixed function[10] $Q(\mathbf{x}, y, \alpha^*)$, any $\epsilon > 0$, and any two randomly chosen subsets of half-samples, the inequality

$$P\left\{\frac{1}{l}\left|\sum_{i=1}^{l} Q(\mathbf{x}_i, y_i, \alpha^*) - \sum_{i=l+1}^{2l} Q(\mathbf{x}_i, y_i, \alpha^*)\right| > \epsilon\right\} \leq 2e^{-\epsilon^2 l} \tag{37}$$

holds true. (see Sections 4.5.3 and 4.5.4 of [8]).

[8]In the context of this work, these frequencies are associated with performance indexes, such as FP or TP.

[9]In statistics the relative frequency of an event i is the number of times the event occurred in the experiment, normalized by the total number of events.

[10]By fixed function, we mean a function whose parameters are fixed as $\alpha = \alpha^*$.

REFERENCES

[1] O. Ludwig, "Study on Non-parametric Methods for Fast Pattern Recognition with Emphasis on Neural Networks and Cascade Classifiers," Ph.D. dissertation, Univ. of Coimbra, Coimbra, Portugal, 2012.

[2] Y. Xu, X. Cao, and H. Qiao, "An efficient tree classifier ensemble-based approach for pedestrian detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 107–117, Feb. 2011.

[3] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1863–1868, Nov. 2006.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, San Diego, CA, USA, 2005, pp. 886–893.

[5] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, Nov. 1984.

[6] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probab. Appl.*, vol. 16, no. 2, pp. 264–280, Apr. 1971.

[7] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *J. Assoc. Comput. Mach.*, vol. 36, no. 4, pp. 929–965, Oct. 1989.

[8] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1999.

[9] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.

[11] O. Ludwig and U. Nunes, "Novel maximum-margin training algorithms for supervised neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 972–984, Jun. 2010.

[12] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, Jan. 1992.

[13] E. B. Kong and T. G. Dietterich, "Error-correcting output coding corrects bias and variance," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 313–321.

[14] R. Kohavi and D. H. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Proc. 13th ICML*, 1996, pp. 275–283.

[15] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: An experiment," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 2, pp. 146–156, Apr. 2002.

[16] M. M. Islam, X. Yao, and K. Murase, "A constructive algorithm for training cooperative neural network ensembles," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 820–834, Jul. 2003.

[17] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*, vol. 904, *Lecture Notes in Computer Science*, P. Vitanyi, Ed. Berlin, Germany: Springer-Verlag, 1995, pp. 23–37.

[18] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, Feb. 1994.

[19] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Ann. Stat.*, vol. 26, no. 5, pp. 1651–1686, Oct. 1998.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. CVPR*, 2001, vol. 1, pp. 511–518.

[21] L. Bruzzone and D. F. Prieto, "A partially unsupervised cascade classifier for the analysis of multitemporal remote-sensing images," *Pattern Recognit. Lett.*, vol. 23, no. 9, pp. 1063–1071, Jul. 2002.

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[23] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[24] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Conf. CVPR*, 2006, pp. 1491–1498.

[25] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse, "Pedestrian localization and tracking system with kalman filtering," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 584–589.

[26] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 807–814.

[27] M. M. Adankon and M. Cheriet, "Genetic algorithm-based training for semi-supervised SVM," *Neural Comput. Appl.*, vol. 19, no. 8, pp. 1197–1206, Nov. 2010.

[28] O. Ludwig, P. C. Gonzalez, and A. C. Lima, "Optimization of ANN applied to non-linear system identification," in *Proc. 25th IASTED Int. Conf. Modeling, Indentification, Control*, Anaheim, CA, USA, 2006, pp. 402–407.

[29] C. Cortes and V. N. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[30] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[31] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, "Trainable classifier-fusion schemes: An application to pedestrian detection," in *Proc. 12th IEEE Int. Conf. ITSC*, Oct. 2009, pp. 1–6.

[32] L. Nanni and A. Lumini, "Ensemble of multiple pedestrian representations," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 365–369, Jun. 2008.

[33] P. Dollár, Z. Tu, H. Tao, and S. Belongie, "Feature mining for image classification," in *Proc. IEEE Conf. CVPR*, 2007, pp. 1–8.

Oswaldo Ludwig received the M.Sc. degree in electrical engineering from Federal University of Bahia, Salvador, Brazil, in 2004 and the Ph.D. degree in electrical engineering from the University of Coimbra, Coimbra, Portugal, in 2012.

He is an Assistant Professor of Computer and Electrical Engineering at the University of Coimbra. His current research focuses on machine learning with application on several fields, such as pedestrian detection in the domain of intelligent vehicles and biomedical data mining.

Urbano Nunes (S'90–M'95–SM'09) received the Lic. and Ph.D. degrees in electrical engineering from the University of Coimbra, Coimbra, Portugal, in 1983 and 1995, respectively. He is a Full Professor with the Computer and Electrical Engineering Department of Coimbra University. He is also the Vice Director of the Institute for Systems and Robotics where he is the Coordinator of the Automation and Mobile Robotics Group. He has been involved with/responsible for several funded projects at both national and international levels in the areas of mobile robotics, intelligent vehicles, and intelligent transportation systems. His research interests are several areas in connection with intelligent vehicles and human-centered mobile robotics with more than 120 published papers in these areas.

Bernardete Ribeiro (M'01) received the M.Sc. degree in computer science and the Ph.D. degree in electrical engineering speciality of informatics both from the University of Coimbra, Coimbra, Portugal.

She is a Professor at the Informatics Engineering Department, Faculty of Science and Technology, University of Coimbra. Her main publications are in the areas of neural networks and their applications to engineering systems, pattern recognition, and support vector machines. Dr. Ribeiro is a member of ACM and IEEE.

Cristiano Premebida received the B.Sc. degree in electrical engineering from the State University of Santa Catarina, Florianopolis, Brazil, in 2001 and the M.Sc. and Ph.D. degrees in electrical and computer engineering from University of Coimbra, Coimbra, Portugal, in 2007 and 2012 respectively.

Currently, he is a Researcher with the Institute of Systems and Robotics, Coimbra. His research interests include mobile robotics, industrial automation, pattern recognition, perception, and autonomous/intelligent vehicles.