STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 2, 2013

PEDESTRIAN RECOGNITION BY USING KERNEL DESCRIPTORS

ANCA ANDREICA, LAURA DIOŞAN, RADU GĂCEANU, AND ADELA SÎRBU[(1)]

Abstract. Recognition of people in images is important for many applications in computer vision. This paper presents an experimental study on pedestrian classification. We investigate the recently developed kernel-based features in order to represent an image and two learning algorithms: the popular Support Vector Machine (SVM) and Genetic Programming (GP). Numerical experiments are performed on a benchmark dataset consisting of pedestrian and non-pedestrian (labeled) images captured in outdoor urban environments and indicate that the evolutionary classifier is able to perform better over SVM.

1. Introduction

Pedestrian safety is an important problem of global dimensions. A World Health Organization 2010 report describes traffic accidents as one of the major cause of death and injuries around the world, accounting for an estimated 1.2 million fatalities and 50 million injuries. Enhancing comfort and safety for the driver and the occupants of an automobile has been a major motivator in innovations associated with Intelligent Vehicles and Intelligent Transportation Systems. The European Union has been conducting several projects in collaboration with auto industry and research institutes for intelligent vehicle systems in general and pedestrian safety in particular.

One approach for pedestrian safety improvement is to develop performant recognition systems. In this paper we propose an evolutionary-based model for the learning phase of such a system. The input data for this classifier is

Received by the editors: March 25, 2013.

2010 *Mathematics Subject Classification.* 68T05,91E45.

1998 *CR Categories and Descriptors.* code I.2.6 [**Learning**]: – *Concept learning*.

Key words and phrases. Object recognition, Kernel descriptors, Support Vector Machines, Kernel selection.

This paper has been presented at the International Conference KEPT2013: Knowledge Engineering Principles and Techniques, organized by Babeş-Bolyai University, Cluj-Napoca, July 5-7 2013.

represented by features extracted from images by using different kernel descriptors.

Pedestrian detection has been an intensively studied problem, especially in the last years. There is a variety of classifiers in literature trained on different combinations of features like HOG [7], SURF [2], CSS [23], Haar walvet representation [22] and tested on datasets like INRIA, Daimler or Caltech. In [23] is performed a comparison of the existing datasets and an evaluation of several classifiers (AdaBoost, linear SVM, HIK SVM and latent SVM) based on HOG and Haar walvet representation. One of the most efficient feature extraction methods is represented by kernel descriptors [4]. In addition, these features have not been used in the case of pedestrian recognition yet, to the best of our knowledge. We, therefore, investigate their usage for this problem.

For solving the classification task, there are several approaches like Boosting [17] and statistical classification: Artificial Neural Networks (ANN) [24] and Support Vector Machines (SVM) [1]. It is known that the SVM performs better as it can avoid over-learning problem that appears in ANN training and a deficiency of a slow training speed that appears in Adaboost.

In our approach we involve a GP-based learning, as evolutionary methods are able to identify a large spectrum of decision functions (linear or non-linear). Furthermore, there are no evolutionary models for solving the pedestrian recognition problem, as far as we know.

The paper is organised as follows: Section 2 reviews the most efficient image representations and Section 3 gives a brief description of two learning algorithms (SVM and GP). The proposed framework is presented in Section 4 and followed by numerical experiments (detailed in Section 5). Finally, conclusions and ideas for future work can be found in Section 6.

2. Image representation

A highly challenging problem in computer vision is how to extract relevant image features. The features that characterise an image can be classified from many points of view. An important criterion is the area from that the feature is extracted: if the entire image is used, then some global features are computed, while if one or more image regions (patches) are utilised, then local features are determined. Another important criterion is the complexity of extraction. Image features can be extracted from scratch that means the features are extracted directly from the image (and in this case we discuss about low-level features) or can be computed based on some previously extracted features (in this case high-level features are obtained). The most popular (due to their success) low-level image descriptors are orientation and gradient histograms,

while one of the best high-level descriptor is the kernel view of orientation histograms [4]. More details about these features are given in the following.

2.1. **Histogram of Oriented Gradient (HOG) descriptors.** The first step in human detection in images is represented by feature extraction. For this stage in the recognition process, Dalal et al [7] proposed the *Histogram of Oriented Gradient (HOG)* descriptors, which show superior performance compaired to previously existing approaches (eg. SIFT [14], SURF [2], etc). HOG descriptors are briefly described in what follows.

Firstly, there is a normalization stage intended to reduce the influence of illumination effects. Power low compression is used on this purpose. There is a second stage for computing first order image gradients for detecting contour, silhouette and some texture information. The image is then divided into small regions called *cells*, for each of them accumulating a local 1-D histogram of gradient or edge orientations over all the pixels. The combined histograms give the image representation. A contrast-normalization follows in order to induce a better invariance to illumination, shadowing and edge contrast. Local groups of cells form *blocks* that might overlap and the normalised block descriptors are referred to as HOG descriptors. The descriptors from all blocks are then collected into a combined feature vector.

Static version of HOG descriptors are obtained by following the stages described above. The authors also proposed motion HOG descriptors which use oriented histograms of differential optical flow. The gradient computation is therefore replaced by flow computation and differential flow estimation [8].

2.2. **Kernel Descriptors (KD).** Kernel descriptors [4] can be seen as a generalization of orientation histograms (including HOG), which are a particular type of match kernels over *patches* (viewed as a collection of *blocks*). Moreover, kernel descriptors overcome some disadvantages of histograms based techniques, where similarity between different regions of images is computed based on their histogram. By using this approach, some quantization errors might be introduced, as individual pixel attribute values are discretized into bins and then a histogram is computed over the discrete attribute values within a patch.

Some of the kernel descriptors are the gradient match kernel (able to capture image variations) – based on a kernel of magnitudes, an orientation kernel and a position kernel –, the colour kernel (able to describe image appearance) – based on a colour kernel and a position kernel – and the local binary pattern kernel (able to capture local shape more effectively) – based on a kernel of standard deviations of neighbour pixels, a kernel of binarized pixel value differences in a local window and a position kernel.

The advantage of kernel descriptors is that they do not discretize pixel attribute values, being able to convert pixel attributes into rich patch-level features. The similarity between different images regions is therefore computed based on a match kernel function. For computation efficiency reasons, approximate, low dimensional match kernels are computed.

Kernel descriptors can be applied not only over sets of pixels, but over sets of kernel descriptors as well. In this hierarchical approach [3], kernel descriptors are recursively applied until the image features are obtained.

3. Learning algorithm

Another aspect that must be considered when the problem of object recognition has to be solved is the classification algorithm. Since the classification must be performed in an automatically manner, a machine learning algorithm can be utilised. The general problem of machine learning is to search a, usually very large, space of potential hypotheses to determine the one that will best fit the data and any prior knowledge. In supervised image classification, we are given a training set of images and their corresponding labels. The goal is to learn (based on the training set) a classifier to label unseen images. Two of the most performant algorithms are SVM and GP-based classifiers and, in what follows, we will briefly describe them.

3.1. **SVM.** SVMs are a group of supervised learning methods that can be applied to classification or regression. They use a technique known as the "kernel trick" to apply linear classification techniques to non-linear classification problems. Using a Kernel function [20], the data points from the input space are mapped into a higher dimensional space. Constructing (via the Kernel function) a separating hyperplane with maximum margin in the higher dimensional space yields a non-linear decision boundary in the input space separating the tuples of one class from another.

Suppose the training data has the following form: $D = (x_i, y_i)_{i=\overline{1,m}}$, where $x_i \in \Re^d$ represents an input vector and each y_i, $y_i \in \{-1, +1\}$, the output label associated to the item x_i. SVM algorithm maps the input vectors to a higher dimensional space where a maximal separating hyper-plane is constructed [20]. In order to construct a maximal margin classifier one has to solve the following convex quadratic programming problem, which is the primal formulation of it:

$$\begin{array}{ll} \text{minimise}_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i & \\ \text{subject to:} & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \forall i \in \{1, 2, \ldots, m\}. \end{array} \tag{1}$$

The coefficient C (usually called *penalty error* or *regularization parameter*) is a tuning parameter that controls the trade off between maximizing the margin and classifying without error. Larger values of C might lead to linear functions with smaller margin, allowing to classify more examples correctly with strong confidence. A proper choice of this parameter is crucial for SVM to achieve good classification performance.

The original optimal hyper-plane algorithm proposed by Vapnik in 1963 was a linear classifier [20]. However, in 1992, Boser, Guyon and Vapnik [5] have suggested a way to create non-linear classifiers by applying the *kernel trick*. Kernel methods work by mapping the data items into a high-dimensional vector space $\mathcal{F}$, called feature space, where the separating hyper-plane has to be found [5]. This mapping is implicitly defined by specifying an inner product for the feature space via a positive semi-definite kernel function: $K(x, z) = \phi(x)^T \phi(z)$, where $\phi(x)$ and $\phi(z)$ are the transformed data items x and z [19]. Note that all we required is the result of such an inner product. Therefore we neither need to have an explicit representation of the mapping ϕ, nor do we need to know the nature of the feature space. The only requirement is to be able to evaluate the kernel function on all the pairs of data items, which is much easier than computing the coordinates of those items in the feature space.

There is a wide choice for a positive definite and symmetric kernel K. The selection of a kernel has to be guided by the problem that must be solved. Choosing a suitable kernel function for SVMs is a very important step for the learning process.

While SVM classifiers intrinsically account for a trade off between model complexity and classification accuracy [21], the generalization performance is still highly dependent on appropriate selection of the penalty error C and kernel parameters.

3.2. **GP.** Genetic programming (GP) is a form of evolutionary computation in which the individuals in the population are computer programs, instead of bit strings [12]. GP starts from a high-level statement of what needs to be done and creates a computer program to solve the problem [12]. While genetic algorithms (GA) want to evolve only solutions for particular problems, GP evolves complex computer programs. GP individuals are represented and manipulated as nonlinear entities, usually trees in the standard approach. Any expression can be represented as a tree of functions and terminals. Depending on the problem to solve, a GP chromosome can be the syntax of an arithmetic expression, formulas in first order predicate logic or code written in a programming language [11].

The skeleton of a GP algorithm is similar to GA. It uses a population of individuals on which the standard evolutionary operators are applied. Initially a population of hierarchically arranged programs is generated by randomly combining functions and terminals. A fitness function may be applied on these programs and the best individuals are selected for reproduction. Crossover is achieved by swapping subtrees among the parents starting at two randomly selected nodes. Mutation may occur by replacing the subtree starting at a randomly selected node in the chromosome by a randomly generated subtree [12].

Flexibility is one of the main advantages of GP, and this feature allows GP to be applied for classification in many different ways. The nature of GP individuals, which include terminals (variables and constants) and nonterminals (operators and functions), gives them the ability not only to represent knowledge but also to perform computations, so that GP can be used in almost any classification-related task, not only in the core task of classifier induction, but also for preprocessing and post-processing purposes [10]. GP can be easily applied to decision tree evolution considering that chromosomes are usually represented as trees and thus they could each be regarded as decision tree classifiers. Along the same line, GP individuals can encode classification rules. A classification rule consists of two parts: the antecedent and the consequent. The antecedent states a condition over the data while the consequent contains the class label. The conditional part is usually composed of relational operators applied on various attribute data, i.e., an expression which can be represented as a tree of functions and terminals.

4. Proposed framework

Our aim is to investigate how two different learning algorithms can solve an important classification problem: the pedestrian recognition. Since the performance of a classification algorithm is strongly influenced by two ingredients (first, a suitable representation of the objects to be categorized and second, a powerful decision maker algorithm on top of this representation), we have decided to extract relevant features from images through kernel descriptors that must be analysed and processed by two classification methods: SVM and GP.

4.1. **Arguments.** *Why Kernels?* Because, by definition, they catch the similarity between arbitrary inputs, being able, in the same time, to integrate invariance (that is present in the case of image processing), to capture dependencies and to perform an efficient computation and storage.

Why SVM? It is well known that linear SVMs are currently the most frequently used classifiers in Computer Vision [13] since its training time is

approximately linear in data dimensionality and, also, approximately linear in number of training examples. Furthermore, the evaluation time (per test example) is linear in data dimensionality and is independent of number of training examples. According to [13] SVMs with non-linear kernel are commonly used for small to medium sized Computer Vision problems because their training time is typically cubic in number of training examples, while their evaluation time is typically linear in number of training examples. Furthermore, the classification accuracy is typically higher compared to linear SVMs. Shortly, linear SVMs are very fast in training and evaluation, while non-linear kernel SVMs give better results, but do not scale well (with respect to number of training examples).

Why GP? There are several arguments that sustain the utility of GP-based methods in solving classification problems, in general, and object recognition, in particular. Firstly, a GP-based method is able to perform an implicit and automatic transformation of data (the original features can be pre-processed by different methods: selection of a subset of original attributes, weighting the original attributes, construction of new features as functional expressions involving the original attributes). The feature selection is part of the evolutionary process of GP that involves individuals encoded as variable-length chromosomes, while the feature construction benefits of the GP individual's ability to combine the attributes trough different operations. Secondly, GP is able to extract various models (like classification rules or discriminant functions) from data. Furthermore, due to its capability to evolve complex discriminant functions, GP is able to solve both linear classification problems and non-linear classification problems without apriori specifying the problem type (linear or non-linear). When a linear classifier can not solve the problem, two solutions can be considered:

- a combination of several linear classifiers (as in the case of ANN or Boosting which actually encode decision functions which depend non-linearly on input data) or
- a data pre-processing step, when the original input data is transformed form the original space of representation (non-linear) into a new space that is, in general, a higher dimensional one and where the data becomes linearly separable. Usually this step is performed through the kernel functions (as in the case of SVMs). For instance a set of points can be non-linearly separable in Cartesian coordinates, but linearly separable in Polar coordinates.

Unlike other Machine Learning (ML) algorithms, GP automatically combines these two solutions during the evolution process, its individuals being able

to automatically encode both type of classifiers (linear and non-linear). Related to how GP can solve Computer Vision tasks, the discriminant functions evolved by the GP algorithm are very akin to the kind of mathematical operations and transformations usually applied to image processing. GP is flexible. It is well known that GP individuals are able to represent a great variety of learning formalisms (eg. decision trees, classification rules, discriminant functions), but also learning mechanisms (like those involved in ANNs or SVMs). Flexibility also concerns the adaptability of GP techniques to various tasks through its elements (fitness function, genetic operation, evolving mechanisms). GP ensures interpretability of the evolved classifiers since the size of GP individuals influences the comprehensibility of the model; the bigger the classifier, the harder to interpret for humans. GP is able to ensure a competitive performance.

Therefore, we study a GP-based classifier that is able to solve the given problem, obtaining improvements that concern several aspects:

- performance of the classification – a better classifier in terms of accuracy
- human-independent models – GP individuals are able to automatically decide in two very important aspects: knowledge and model representation. Instead of performing a distinct pre-processing step, the GP method is able to automatically and simultaneously select the most relevant features and construct a relevant model.
- less complex models – GP is able to automatically apply the principles of Occam's razor: if two models have the same performance (in our case if two decision algorithms ensure the same classification accuracy) the less complex model should be preferred.

Taking into account all these aspects, we have used in our numerical experiments the following methodology: several features are extracted directly from images through kernel descriptors and, afterwards, two algorithms (an SVM and a GP-based classifier) are considered in order to learn the decision model. In both cases, the learning takes place in a cross-validation framework. In k-fold cross-validation, the training data is randomly split into k mutually exclusive subsets (or folds) of approximately equal size. The decision is obtained by using $k-1$ subsets on training data and then tested on the subset left out. This procedure is repeated k times and in this manner each subset is used for testing once. Averaging the test error over the k trials gives a better estimate of the expected generalization error.

4.2. **Feature extraction by using kernel descriptors.** We considered the framework proposed by L. Bo [4] for image classification and we test different

kernel functions. We have already established that the selection of the kernel function is very important for kernel descriptor [9].

At the first step, based on the available code of Kernel descriptors developed by Xiaofeng Ren ($http://www.cs.washington.edu/ai/Mobile_Robotics/projects/kdes/$), we have tested different kernels when extracting local features from an image. Because we work only with gray images, we investigate only the kernel descriptors able to capture image variations (gradient match kernels [4]). As presented in Section 2, the Bo's gradient match kernel is composed of three kernels: a kernel of magnitudes, an orientation kernel and a position kernel.

The magnitude kernel is a linear one and its role is to measure the similarity of gradient magnitudes of two pixels. The magnitude kernel type cannot be changed since it must be an equivalent of the histogram of gradients in the feature map (a pixel has an associated feature vector obtained by multiplying the magnitude and the orientation of the pixel over all considered orientation bins).

The other two kernels involved in Ren's computation of the gradient match kernel, the orientation kernel (for computing the similarity of gradient orientations) and the position kernel (for measuring the spatial closeness of two pixels), are actually Gaussian kernels. Therefore, we have changed the implementation and we have involved more possible orientation and position kernels (Exponential, Laplacian, Euclidean) in the feature extraction process.

4.3. **SVM.** Regarding the SVM, we have considered the LibSVM tool [6] since it shines above the other tools in terms of ease of use, choice of options and features.

The dual version of the optimisation problem which arises during the SVM training was solved by Sequential Minimal Optimization (SMO) algorithm [18], since it is able to quickly solve the quadratic programming optimisation problem of SVM. We have chosen this formulation of SVM since the duality theory provides a convenient way to deal with the constraints and, in this form, the optimisation problem can be solved in terms of dot products, that allows using the kernel trick. Furthermore, SMO requires an amount of memory that increases only linearly with the training set size, being able to handle very large training sets - as in the image classification case. These aspects are different from Bo's framework [4] that is based on primal formulation of SVM and on conjugate gradient optimisation methods (in fact, Newton optimisation).

We have tried to use several kernels with different parameters during the SVM learning process in order to identify the best one: the Linear kernel, the Polynomial kernel, the Gaussian kernel and the Normalised Polynomial kernel. For the Polynomial kernel several exponents have been tested (2, 3),

for parameter $\frac{1}{2\sigma^2}$ of Gaussian kernel the following values have been checked: 0.1, 0.01, 0.001, 0.0001 and for Normalised Polynomial kernel the exponent was 2.

4.4. **GP.** For the evolutionary classifier a linear and efficient GP version is actually utilized: Multi Expression Programming (MEP)[16]. MEP uses a linear representation of chromosomes and a mechanism to select the best gene for providing the output of the chromosome. This is different from other GP techniques which use a fixed gene for output. Furthermore, no extra processing for repairing newly obtained individuals is needed.

The dynamic-output chromosome has some advantages over the fixed-output chromosome especially when the complexity of the target expression is not known. Variable-length expressions can be implicitly provided, while other techniques (such as Grammatical Evolution or Linear GP) employ special genetic operators (which insert or remove chromosome parts) in order to achieve such a complex functionality. The expression encoded in a MEP chromosome may have exponential length when the chromosome has polynomial length due to code reuse [16].

5. Numerical experiments

Several numerical experiments about how the discussed learning algorithms (an SVM and a GP-based classifier) are able to solve a particular image classification task (pedestrian recognition) are presented in this section. To evaluate the performance of the considered classifiers, the Daimler-Chrysler (DC) crop wise data sets (18×36 pixels image size) have been used as provided in [15]. For all datasets a binary classification problem was actually solved: separate the images that contain pedestrians from the images that do not. 4480 images are considered: the decision model is trained on 2240 of them, while 2240 of images are used for testing.

In order to measure the classification performance, the accuracy rate was actually computed. The accuracy rate represents the number of correctly classified items over the total number of items from a given data set. However, the accuracy rate reflects the classification performance of the learning algorithm in a confidence interval. The confidence intervals associated to the performances of the systems must be computed in order to decide if a system outperforms another system. If these intervals are disjoint, then one system outperforms the other one. A confidence interval of 95% is used in order to perform a statistical examination of our results. Therefore, the probability that the accuracy estimations are not in the confidence interval is 5% (see Equation (2)):

$$\Delta I = 1.96 \times \sqrt{\frac{Acc(100 - Acc)}{N}}\% \tag{2}$$

where N represents the number of test examples.

In Table 1 are presented the accuracy rates (and their confidence intervals) by considering different image descriptor kernels (when the image descriptors are actually constructed) and two learning algorithms (SVM and MEP). The performance measures are computed by taking into account the test images and the best identified classifiers (SVM with the best hyper-parameters and MEP with an optimal configuration).

	Exponential	Gaussian	Laplacian
SVM	0.535 ± 0.010	0.657 ± 0.010	0.599 ± 0.010
MEP	0.667 ± 0.009	0.682 ± 0.009	0.737 ± 0.009

TABLE 1. Accuracy rates (%) obtained by SVM and MEP algorithms on images represented by different kernel descriptors.

Several remarks can be done based on the results from Table 1.

Regarding the kernel descriptors, our results indicate that the Gaussian kernel seems to be able to extract the most relevant features from images when the SVM classifier is used, while the Laplacian kernel provides more significant information for MEP. Even if the Gaussian kernel is largely involved in feature extraction process [4], our results suggest that a deeper study should be performed regarding the proper selection of the kernel involved in the feature extraction process. This study might also reveal some criteria for selecting the most appropriate kernel descriptor to use for the input data of a particular problem.

Regarding the learning algorithm, the evolutionary one seems to be able to better generalise over unseen data, compared to SVM. This observation is sustained by the better accuracy rates obtained for all three considered feature extraction methods.

6. Conclusions

A study on how two learning algorithms are able to perform pedestrian recognition in images is presented in this paper. Daimler-Chrysler benchmark image dataset is involved in our numerical experiments.

The first step is to convert each image in a numerical representation relevant for the classifier. Several kernel descriptors are considered on this purpose: Exponential, Gaussian and Laplacian kernels. A statistical algorithm

— SVM — and an evolutionary approach — MEP — are used for the learning phase for which the input data is represented by the previously extracted features.

Better accuracy rates are obtained when using the evolutionary model for all considered kernel descriptors. This might be considered an indicator of the superiority of the evolutionary approach over SVM for the considered problem.

Regarding the kernel descriptors used, SVM learning indicates that the Gaussian is the best one, while MEP achieves the best results by using the Laplacian kernel. Therefore, we can not conclude which is the most efficient kernel descriptor and we intend to perform a further study of how the kernel selection influences the quality of recognition.

Our future work will also include a validation of the obtained results by considering other datasets like Caltech or INRIA.

References

[1] Alonso, I. P., Llorca, D. F., Sotelo, M. A., Bergasa, L. M., de Toro, P. R., Nuevo, J., Ocana, M., and Garrido, M. A. G. Combination of feature extraction methods for SVM pedestrian detection. *IEEE Trans. Intelligent Transportation Systems 8*, 2 (Apr. 2007), 292–307.

[2] Bay, H., Tuytelaars, T., and Gool, L. J. V. SURF: Speeded up robust features. In *ECCV* (2006), pp. I: 404–417.

[3] Bo, L., Lai, K., Ren, X., and Fox, D. Object recognition with hierarchical kernel descriptors. In *CVPR* (2011), IEEE, pp. 1729–1736.

[4] Bo, L., Ren, X., and Fox, D. Kernel descriptors for visual recognition. In *NIPS* (2010), J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., Curran Associates, Inc, pp. 244–252.

[5] Boser, B. E., Guyon, I., and Vapnik, V. A training algorithm for optimal margin classifiers. In *COLT* (1992), D. Haussler, Ed., ACM Press, pp. 144–152.

[6] Chang, C.-C., and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] Dalal, N., and Triggs, B. Histograms of Oriented Gradients for human detection. In *CVPR* (2005), C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, pp. 886–893.

[8] Dalal, N., Triggs, B., and Schmid, C. Human detection using oriented histograms of flow and appearance. In *ECCV* (2006), pp. II: 428–441.

[9] Diosan, L., and Rogozan, A. How the kernels can influence image classification performance. *Studia Universitatis Babes-Bolyai : Series Informatica LVII*, 4 (2012), 97–109.

[10] Espejo, P., Ventura, S., and Herrera, F. A survey on the application of genetic programming to classification. *Transactions on Systems, Man ans Cybernetic, Part C 40* (2010), 121–144.

[11] Grosan, C., and Abraham, A. *Intelligent systems: A modern approach*. New York, Springer, 2011.

[12] Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[13] Lampert, C. Kernel methods in computer vision. presentation during Computer Vision and Sports Summer School, Prague, 2012, 2012.
[14] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.
[15] Munder, S., and Gavrila, D. M. An experimental study on pedestrian classification. *IEEE Trans. Pattern Analysis and Machine Intelligence 28*, 11 (Nov. 2006), 1863–1868.
[16] Oltean, M. Improving the search by encoding multiple solutions in a chromosome. In *Evolutionary Machine Design: Methodology and Applications*, N. Nedjah and L. de Macedo Mourelle, Eds. Nova Publishers, 2005, ch. 4, pp. 85–110.
[17] Paisitkriangkrai, S., Shen, C. H., and Zhang, J. Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Trans. Circuits and Systems for Video Technology 18*, 8 (Aug. 2008), 1140–1151.
[18] Platt, J. Fast training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods — Support Vector Learning* (Cambridge, MA, 1999), B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., MIT Press, pp. 185–208.
[19] Schölkopf, B. The kernel trick for distances. In *NIPS* (Cambridge, MA, 2000), T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., MIT Press, pp. 301–307.
[20] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, 1995.
[21] Vapnik, V. *Statistical Learning Theory*. Wiley, 1998.
[22] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR 1* (2001), 511–518.
[23] Walk, S., Majer, N., Schindler, K., and Schiele, B. New features and insights for pedestrian detection. In *CVPR* (2010), IEEE, pp. 1030–1037.
[24] Zhao, L., and Thorpe, C. E. Stereo- and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems 1*, 3 (2000), 148–154.

(1) Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania
E-mail address: {anca,lauras,radu,adela}@cs.ubbcluj.ro