

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Classifiers Association for High Dimensional Problem: Application to Pedestrian Recognition

Laetitia LEYRIT, Thierry CHATEAU and Jean-Thierry LAPRESTE
LASMEA - UMR 6602 CNRS & Blaise Pascal University
France

1. Introduction

Machine learning is widely used for object recognition in images (Viola & Jones, 2001). Indeed, the goal is to recognize any object of the same class whatever the background, the illumination conditions, The key-point of such a method is the ability to create a generic model able to describe the huge variability of an object class. A large training set is then used so as to cover all the variations taken by the object class. For each example, a simple description provides a huge feature vector from which only a subset is relevant according to the object to be recognised.

Kernel based machines like Support Vector Machine (SVM) (Vapnik, 1998) have shown great performances for object recognition in images (Papageorgiou & Poggio, 2000). But high dimensional problem can be prohibitive for it: it implies expensive time, presence of irrelevant features can disturb the classifier and overfitting often occurs.

Various approaches were proposed in order to decrease this number of variables (Guyon & Elisseeff, 2003). They are of two types: the *filters* and the *wrappers*. The filters methods use only the training set. They process the entire data before the learning step and keep only relevant characteristics. Most widespread is the Relief algorithm, introduced by Kira and Rendell (Kira & Rendell, 1992) and improved by Kononko (Kononenko, 1994), which computes a criterion of relevance for each characteristic of the training set. Another approach presented by Hall in (Hall, 2000) uses a correlation score to reduce the training set. An extension of this method was developed in (Yu & Liu, 2003) for great dimension sets.

The wrappers methods carried out the variable choice at the same time as the training process is done. Moreover, they use the process itself to select relevant characteristics (Kohavi & John, 1997). Solutions were brought for SVMs. Weston in (Weston et al., 2000) explores parameter space by a stepped gradient descent and fix an exit threshold on the classification error. Rakotomamonjy proposes in (Rakotomamonjy, 2003) a selection criterion based on the variable influence on the decision rule of a SVM classifier. Generally, these methods, which take account of the training set and the classifier in the same time, give good results but induce expensive computing times.

For an out-line learning, the computing time is not the main problem. However, studies like (Campedel et al., 2005) showed the efficiency of variable selection to improve the classifier performances: the presence of useless data can disturb the classifier and memory is misused.

With the aim of time-saving, the ideal is to use a variable selection algorithm which can be processed independently of the learning process. But not to take account of the classifier is the major disadvantage of the filtering methods: the drawback is to select attributes which are not finally useful for this one. To guarantee the relevance of the characteristics preserved for the classifier, the best tool is this classifier used itself.

AdaBoost algorithms can also be used for feature selection. In (Viola & Jones, 2001) an AdaBoost cascade is used with Haar wavelets based descriptor. At each stage of the classifier, a Haar resolution is chosen and images are divided into several sub-windows. The classifier rejects the non-informative ones. At the follow step, the Haar resolution is increased only for sub-windows which were selected at the previous iteration. This stage of the classifier rejects also the non-informative sub-windows and the process keeps on. After several stages of the classifier, the number of sub-windows decrease quickly. Moreover the decision threshold is readjusted as the classifier progresses. An extension of this cascade method is developed in (Le & Sato, 2004) with a final SVM classifier. First stages of the classifier use AdaBoost algorithm to reduce the feature space and select relevant features. The last stage is a SVM classifier which builds a face model from the features selected previously. This both methods allow to reduce the features number in the first stages of the cascade. In this approach, the AdaBoost algorithm is only used to select relevant features from a huge set of possible ones.

We propose here an original association of a classic AdaBoost algorithm with a kernel based machine. Adaboost is an algorithm which builds a strong classifier by selecting a huge number of weak ones. It can be used for feature selection too: each feature can be seen as a weak classifier and AdaBoost selects a subset of them. Our approach consists on using the resulting subset of weak relevant classifiers (and not relevant features) as binary vectors in a kernel based machine learning classifier (like SVM).

We focus our proposal on pedestrian recognition: since pedestrians provide a large appearance variability (size, clothes, skin colour, ...) the training set used for learning must be very large. Numerous features are then used to describe correctly each sample of the training set. The association of AdaBoost and kernel machine allows to handle this high dimensional problem.

This chapter is organized as follow: section 1 describes the main features used in classification; then the classifiers methods is presented in section 2. Experiments and results of the proposed method on a pedestrian recognition task are realizes in section 3; and finally section 4 gives the conclusion.

2. Features for image description

In pattern recognition, different types of features are widely used to describe each image. For many recognition system, the goal is to realize an on-line application, and features are chosen according to that. These features have to be efficient of course but moreover to compute quickly so as to reach an on-line processing time. We present here three types of features widely used in patten recognition: Haar wavelets, histograms of oriented gradients and binary descriptors.

2.1 Haar wavelets

They are probably the most used features for pattern recognition in images. Introduced by Papageorgiou in (Papageorgiou & Poggio, 2000), they encode a local information with an intensity difference at different scales.

The overcomplete dictionary presented in (Papageorgiou & Poggio, 2000) allows a fast computation of haar wavelets in three directions: horizontal, vertical and diagonal (see figure 1) The main difficulty is to find adapted sizes for a given image. Indeed a finer scale only captures noise whereas large scale doesn't capture an object characteristics.

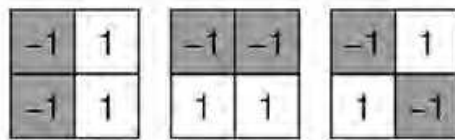


Fig. 1. The three different orientations of the Haar wavelets.

2.2 Histograms of Oriented Gradients

This image descriptor for image recognition was introduced in (Dalal & Triggs, 2005). This descriptor is based on edge orientation histograms (Freeman & Roth, 1995) and SIFT descriptor (Lowe, 2004) The idea is to count occurrences of gradient orientation in localized portions of an image called cells. We thus obtain a dense grid of uniformly spaced cells and overlapping local contrast normalization is used for improved performance. Here difficulties are to choose a correct grid to describe an object and to select a correct normalization schema.

2.3 Binary descriptors

This kind of descriptor becomes very popular used in pattern recognition, especially for low resolution images. It computes an intensity comparison of two points, which is an enough resolution for small images. Various methods were proposed to select and associated them (Lepetit & Fua, 2006; Moutarde et al., 2008). The points selection appears as the main problem of this descriptor because it gives quickly a high dimensional feature vector if all the possibilities space is covered up. We build our own binary descriptor based on the comparison result of the grey levels as presented in the figure 2. Let us note \mathbf{u} the coordinates of an image point. $I(\mathbf{u})$ returns the pixel intensity at this point, i.e. the grey level associated with these coordinates in the image. Given two points of the image, \mathbf{u}_1 and \mathbf{u}_2 , the descriptor carries out the following comparison:

$$(I(\mathbf{u}_1) \geq I(\mathbf{u}_2)) \text{ (with } \mathbf{u}_1 \neq \mathbf{u}_2) \quad (1)$$

It returns the logical value 1 if the test is true and 0 if the test is false.



Fig. 2. Descriptor of grey-levels comparisons.

3. Machine Learning

Today the machine learning used in image recognition are either Boosting method either Kernel method. This section describes these learning methods.

Here, we follow the standard notations, representing the output labels by a scalar y which can take two possible discrete values corresponding to the object class: $y = -1$ for negative examples (non-objects) and $y = 1$ for positive examples (objects). Vectors $\mathbf{x} \in \mathbb{R}^Q$ represents input features provided by images descriptors. Let $S \doteq \{(\mathbf{x}^i, y^i)\}_{i=1}^N$ denotes a training set composed by N samples of feature vectors associated to their corresponding labels.

3.1 Adaptive Boosting or AdaBoost

This method is developed by Freund and Schapire in (Freund & Schapire, 1996). It's the more common method of Boosting used for image recognition. The AdaBoost principle is straight forward. As opinions of several experts are better than only one, this algorithm combines decisions of several weak classifiers. An uniform weight is given for each data of the training set. At each iteration, a subset of the training set is drawn from S according to the weights assigned to them at the previous iteration. A weak classifier $h(t)$ is created from this subsample and the classification error ϵ_t is calculated for the entire training set. The weight vector is then updated: the weight of the elements well classified decreases, while the weight of those badly classified increases. The process is reiterated until reaching the number of required weak classifiers or until the error on the training set is lower than a given threshold.

The decision rule associated to Adaboost is then a linear combination of selected weak classifiers:

$$y = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right) \quad (2)$$

where α_t are weights estimated from the learning step. Equation (2) clearly shows that the classification rule is given by the sign of a linear equation (hyperplane) into the selected weak classifiers space.

3.2 Kernel Machine

Kernel based machines has been widely used for classification ((Suard et al., 2006),(Papageorgiou & Poggio, 2000),(Shashua et al., 2004)). The general form of the classifier is given by:

$$y = \text{sign}\left(\sum_{m=1}^M w_m \phi_m(\mathbf{x})\right) \quad (3)$$

Here, $\{\phi_m(\mathbf{x}) | m = 1 \dots M\}$ are basis functions and $\{w_m | m = 1 \dots M\}$ are the associated weights. We propose non linear basis functions:

$$\phi_m(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}^m) \quad (4)$$

where $k(\mathbf{x}, \mathbf{x}^m)$ is a kernel function. The classification rule can be written in a more compact form by the following equation:

$$y = \text{sign}(\mathbf{w}^T \phi(\mathbf{x})) \quad (5)$$

where $\mathbf{w}^T = (w_1, w_2, \dots, w_M)$ is a weight vector and $\phi(\mathbf{x}) = (\phi(\mathbf{x}^1), \phi(\mathbf{x}^2), \dots, \phi(\mathbf{x}^N))^T$. To train the model (estimate \mathbf{w}), we are given the training set $S^h = \{(\mathbf{x}^i, y^i)\}_{i=1}^N$. We use the Euclidean norm to measure y -space prediction errors, so the estimation problem is of the form:

$$\mathbf{w} := \arg \min_{\mathbf{w}} \{ \|\mathbf{w}^T \Phi - \mathbf{y}\|^2 \} \quad (6)$$

where $\Phi \doteq (\phi(\mathbf{x}^1), \phi(\mathbf{x}^2), \dots, \phi(\mathbf{x}^N))$ is the design matrix and $\mathbf{y} \doteq (y^1, \dots, y^N)^T$ is the training set label vector. The estimation of the parameter vector \mathbf{w} using the least-square criterion defined in equation (6) is given by:¹

$$\mathbf{w}_{ls} = \mathbf{y}\Phi^+ \quad (7)$$

Alternative methods can be used to estimate \mathbf{w} . A solution is to place priors over \mathbf{w} in order to set many weights to zero. The resulting model is then called sparse linear model. SVM (*Support Vector Machine*) (Vapnik, 1998) is a sparse linear model where the weights are estimated by the minimization of a Lagrange multipliers based functional. Other sparse linear models, like RVM (*Relevant Vector Machines*) (Tipping, 2001) may also be employed.

Vectors used for basis functions are usually composed by a subset of the training set S^h . It is also possible to use the entire training set and in this case $M = N$. The matrix Φ is then symmetric and system resolution can be made more efficiently using Cholesky decomposition.

We make the common choice to use Gaussian data-centred basis functions:

$$\phi_m(\mathbf{x}) = \exp \left[-(\mathbf{x} - \mathbf{x}^m(\mathbf{x}))^2 / \sigma^2 \right], \quad (8)$$

which gives us a "radial basis function" (RBF) type model from which the parameter σ must be adjusted. On one hand, if σ is too small, the "design matrix" Φ is mostly composed of zeros. On the other hand, if σ is too large, Φ is mostly composed of ones. We propose to adjust σ using a non linear optimization maximizing an empirical criteria based on the sum of the variances computed for each line of the design matrix Φ :

$$\sigma := \arg \max_{\sigma} [-C(\sigma)] \quad (9)$$

with

$$C(\sigma) = \sum_{n=1}^N \sum_{m=1}^M \left(\phi_m(\mathbf{x}) - \bar{\phi}(\mathbf{x}^{(n)}(\mathbf{x})) \right)^2 \quad (10)$$

and

$$\bar{\phi}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^M \phi_m(\mathbf{x}^{(n)}(\mathbf{x})) \quad (11)$$

The classifier thus obtained will be denoted by KHA (*Kernel Approximation Hyperplane*) in section 4.

3.3 Classifiers Association

Our goal is to be able to recognize object with a great variability. For that, we use dataset with a large number of examples, and each example is described by a huge features vector. Using all this features set is prohibitive for the computation of kernel machines. That's why we propose here to use an Adaboost algorithm to choose only the more relevant features into the training set. As AdaBoost selects the best weak classifiers for a classification task, the features selected will be relevant for a kernel machine too. Here, we propose an original approach which consists in selecting weak classifiers with AdaBoost (not relevant feature selection), and then using the selected weak classifiers as new binary input vectors to learn a kernel based classifier. This method provides non-linear separator in the weak learner space and classifies accurately more examples as shown in figure 3; positive examples are denoted by symbol +, negative examples by -. $h_1(x)$ and $h_2(x)$ are two weak classifiers selected by AdaBoost. They

¹ Φ^+ denotes the pseudo-inverse of Φ .

return value 1 for examples classified as positive and -1 for examples classified as negative. In the weak learner space, AdaBoost provides a linear separator and some examples are misclassified. Our method provides non-linear separator and classified correctly all examples. So

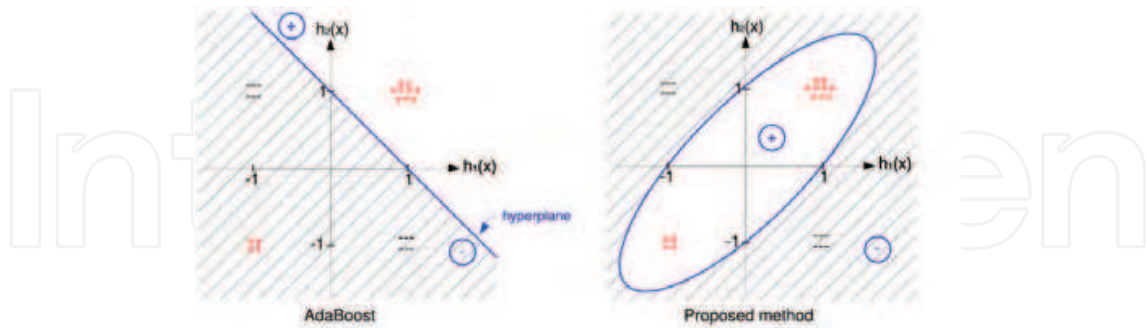


Fig. 3. Our method provides non-linear separator while AdaBoost gives linear one.

the first stage of the learning process is doing by an AdaBoost algorithm which gives a new binary training set for the kernel machines. We define $S^h \doteq \{(\mathbf{h}(\mathbf{x}^i), y^i)\}_{i=1}^N$ a new training set where $\mathbf{h}(\mathbf{x}^i) \in \mathbb{R}^T$ is a vector composed by the output of each selected classifier estimated from parameters \mathbf{x} such as $\mathbf{h}(\mathbf{x}^i) \doteq (\mathbf{h}_1(\mathbf{x}^i), \dots, \mathbf{h}_T(\mathbf{x}^i))^T$. Next the classification rule for kernel machines becomes:

$$y = \text{sign} \left(\sum_{m=1}^M w_m \phi_m(\mathbf{h}(\mathbf{x})) \right) \quad (12)$$

and

$$\phi_m(\mathbf{h}(\mathbf{x})) = k(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x})) \quad (13)$$

where $k(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x}))$ is the kernel function. The estimation problem is given by equation (6) and the resolution is done like presented in section 3.2 .

4. Experiments and Results



Fig. 4. Examples of pedestrian (first line) and non-pedestrian (second line) images.

We focus our work on pedestrian recognition in images coming from low-cost camera (see (Leyrit et al., 2008)). This work is challenging because pedestrian is a hard pattern to recognize due to the differences of clothes, size... added to classic illumination and background variations. Since pedestrian appearance provides a large variability, the training set used in

the learning stage must be very huge and each sample can be described by a huge feature vector. We used the images dataset provided by Gavrilu and Munder in (Munder & Gavrilu, 2006). This base is subdivided into five parts; each one contains 4800 positive and 5000 negative images. Each picture has a size of 36x18 pixels, in grey levels. In the positive images, the pedestrians are standing and entirely visible; they were taken in various postures, and in various illumination and background conditions. Each pedestrian picture was randomly reflected and shift a few pixels in the horizontal and vertical directions. The negative images describe the urban environment: trees, buildings, cars, road signs... This base (some examples of which are shown in figure 4) constitutes the data used for training and the validation of the proposed method. According to (Munder & Gavrilu, 2006) the three first parts are used for the training, and the two last ones are used for the validation. It assumes that the validation is doing independently of the training.

4.1 The proposed method compared to a standard AdaBoost

In this part we compare the results of a standard AdaBoost to the ones given by the proposed method. We use here the association of the binary output of the AdaBoost with a KHA kernel machine. Results are obtained for the descriptor of grey level comparisons (see paragraph 2.1). We make these comparisons between points belonging to the same line or the same column. As the image size is of 36x18 pixels, we obtain 5508 binary descriptors for the lines and 11340 for the columns.

The table 1 gives recognition errors for these experiments; the class decision rule is computed with a classical threshold set to zero. We also plot ROC curves for each experiments as shown

Classifier	Learning set number	Error on validation set 4	Error on validation set 5
Standard AdaBoost	1	27,49%	25.09%
Association with KHA	1	30,61%	24,21%
Standard AdaBoost	2	26,81%	31,69%
Association with KHA	2	16,62%	14,81%
Standard AdaBoost	3	44,79%	44,71%
Association with KHA	3	21,42%	13,48%

Table 1. Comparison between standard AdaBoost and the classifiers association using grey-level features.

in figure 5 for more precision. A ROC curve (*Receiver Operating Characteristic*) presents variations and sensitivity of a test for various values of the discrimination threshold. The x-axis represents the false negative rate (non-pedestrians classified as pedestrians) while the y-axis corresponds to the true positive rate (of the pedestrians which are well detected as such). Let us suppose a ROC curve through the point (0.1;0.9). That means that for 90% of the well classified pedestrians, 10% of non-pedestrians are badly classified. Most of the time, the classifiers association gives best results than a standard AdaBoost. In this way, we can achieve good recognition rate despite the high dimensional size of the features vectors.

4.2 Comparison between two different kernel machines

We tested the proposed method with a SVM kernel machine. Results are given in table 2 (decision threshold set to zero) and ROC curves are presented in figure 6. With a decision threshold set to zero, KHA gives better results; but we can see in figure 6 that for other decision

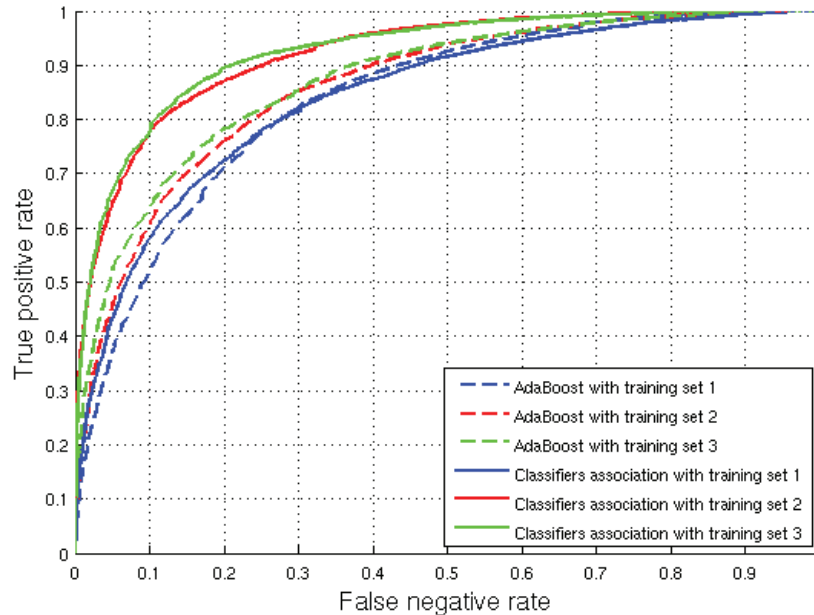


Fig. 5. Comparison between standard Adaboost and the proposed method.

thresholds, SVM reaches equivalent recognition rate. The decision threshold should be selected carefully according to each application. It depends on the rate of misclassified an application can tolerate.

Classifier	Learning set number	Error on validation set 4	Error on validation set 5
Association with KHA	1	30,61%	24,21%
Association with SVM	1	30,62%	25,70%
Association with KHA	2	16,62%	14,81%
Association with SVM	2	20,76%	21,09%
Association with KHA	3	21,42%	13,48%
Association with SVM	3	30,96%	27,86%

Table 2. Errors of the proposed method with two different kernel machines.

4.3 Comparison between three descriptors

We have implemented the three descriptors presented in section 2: the histograms of oriented gradients (HOG), Haar wavelets and our grey-level descriptor.

We chose a Haar wavelets size of 2x2 and 4x4, shifted by $\frac{1}{4}$ of the size of the wavelet in the three directions (see paragraph 2.1). It gives 1755 features.

As regards the HOG, we chose a cutting into 3x3 cells, and histograms are computed with 8 bins. We thus obtain 576 features.

For the two previous descriptors, the feature vector is relatively small and doesn't require a previous weak classifier selection. We only use the complete method for the binary descriptor. Then a KHA kernel machine is trained on each dataset.

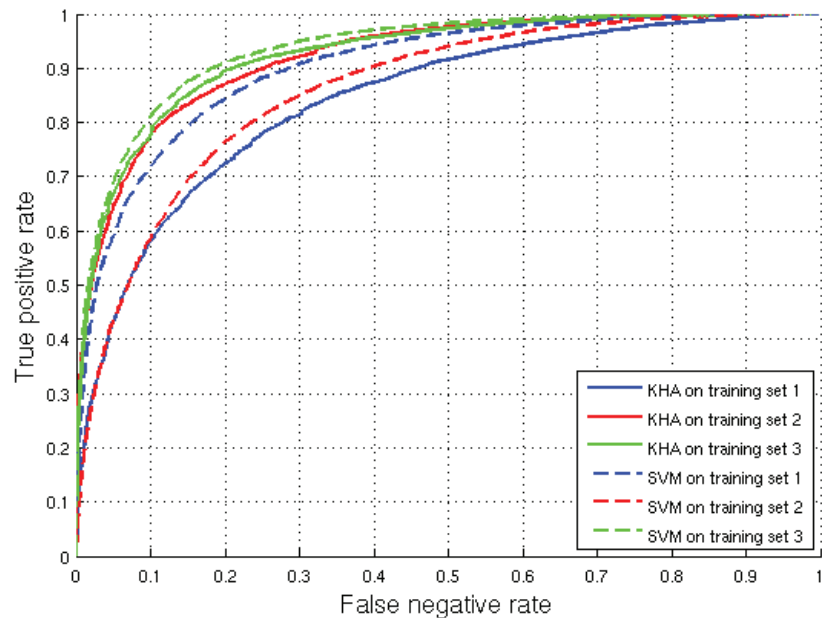


Fig. 6. The proposed method implemented with two different kernel machines.

The results presented in table 3 and in figure 7 show that these three descriptors work into almost the same range of values. With more precision, ROC curves show that histograms of oriented gradients are better for these pedestrian recognition task. The binary descriptor, despite its simplicity, achieves almost same results. Haar wavelets doesn't reach the same performances than the two others descriptors.

Descriptor	Learning set number	Error on validation set 4	Error on validation set 5
HOG	1	24,99%	21,17%
Haar wavelets	1	31,54 %	31,18%
Grey-level descriptor	1	30,61%	24,21%
HOG	2	26,17%	21,39%
Haar wavelets	2	20,35%	20,55%
Grey-level descriptor	2	16,62%	14,81%
HOG	3	18,84%	16,32%
Haar wavelets	3	21,84%	16,32%
Grey-level descriptor	3	21,42%	13,48%

Table 3. Errors with three different descriptors.

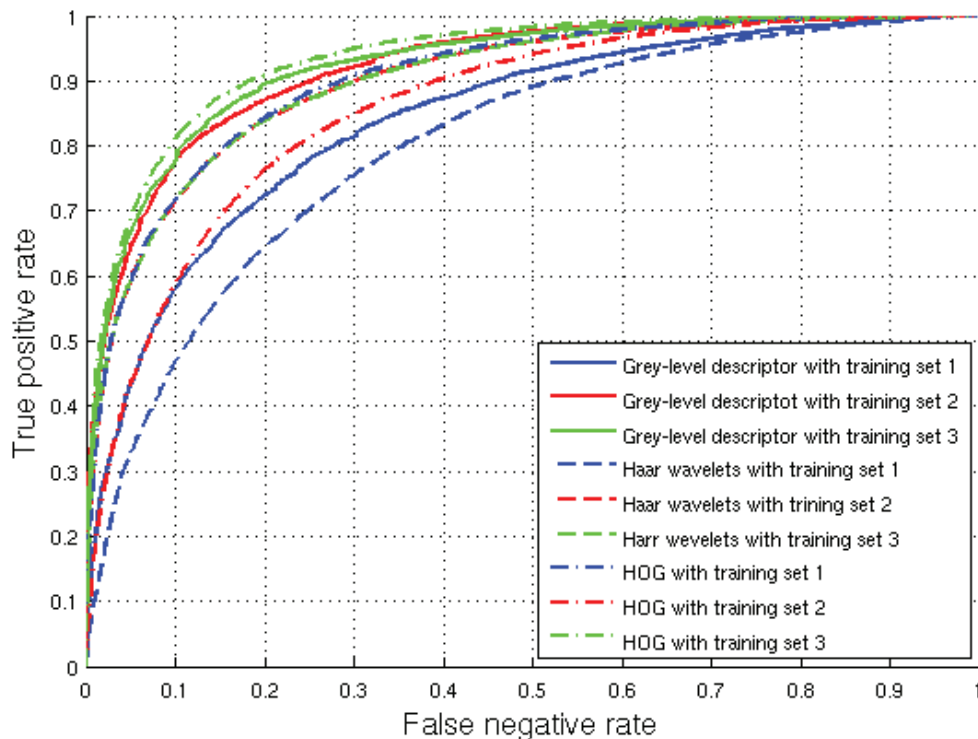


Fig. 7. Comparison between three different descriptors.

5. Conclusion and future works

We have proposed a learning based approach for high dimensional object detection. This method uses an AdaBoost algorithm to select relevant weak classifiers, which are then used as binary vectors to learn a kernel based classifier.

This method helps to solve high dimensional problem like a pedestrian recognition task; results show that the method gives good performances and outperforms standard AdaBoost.

Three popular descriptors have been tested; histograms of oriented gradients give the best results but the binary descriptors reach almost the same results, which is very interesting for a real-time application. Haar wavelets are less competitive than the two others descriptors.

We have designed our own classifier, KHA, which gives similar results than a SVM classifier. KHA is easy to develop but we explore now solutions to decrease the number of retained support vectors and to select relevant ones. It should help us to reach faster computation time.

6. References

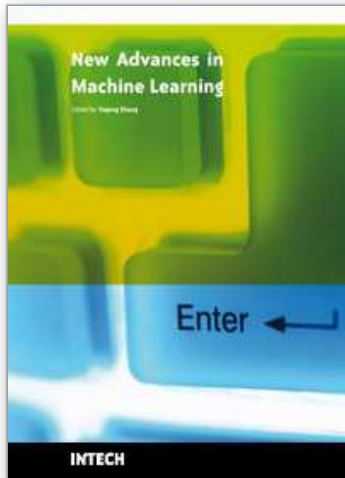
- Campedel, M., Moulines, E., Matre, H. & Dactu, M. (2005). Feature Selection for Satellite Image Indexing, *ESA-EUSC 2005: Image Information Mining - Theory and Application to Earth Observation*.
- Dalal, N. & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection, *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, California, USA, pp. 886–893.

- Freeman, W. T. & Roth, M. (1995). Orientation histograms for hand gesture recognition, *Intl. IEEE Workshop on Automatic Face and Gesture- Recognition*, Zurich, Switzerland, pp. 296–301.
- Freund, Y. & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm, *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 148–156.
- Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection, *The Journal of Machine Learning Research* **3**: 1157–1182.
- Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning, *Proceedings 17th International Conference on Machine Learning (ICML)*, Morgan Kaufmann, pp. 359–366.
- Kira, K. & Rendell, L. A. (1992). A practical approach to feature selection, *Proceedings of the 9th International Workshop on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 249–256.
- Kohavi, R. & John, G. H. (1997). Wrappers for feature subset selection, *Artificial Intelligence* **97**(1-2): 273–324.
- Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF, *Proceedings of the European Conference on Machine Learning (ECML)*.
- Le, D. D. & Satoh, S. (2004). Feature Selection by AdaBoost for SVM-based Face Detection, *Forum on Information Technology* pp. 183–186.
- Lepetit, V. & Fua, P. (2006). Keypoint Recognition using Randomized Trees, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(9): 1465–1479.
- Leyrit, L., Chateau, T., Tournayre, C. & Lapresté, J.-T. (2008). Association of AdaBoost and Kernel Based Machine Learning Methods for Visual Pedestrian Recognition, *IEEE Intelligent Vehicles Symposium (IV 2008)*, Eindhoven, Netherlands.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**(20): 91–110.
- Moutarde, F., Stanciulescu, B. & Breheret, A. (2008). Real-time visual detection of vehicles and pedestrians with new efficient adaBoost features, *Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV) of International Conference on Intelligent Robots and Systems (IROS)*, Nice, France.
- Munder, S. & Gavrilă, D. (2006). An Experimental Study on Pedestrian Classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **28**(11).
- Papageorgiou, C. & Poggio, T. (2000). A trainable system for object detection, *International Journal of Computer Vision* **38**(1): 15–33.
- Rakotomamonjy, A. (2003). Variable selection using svm-based criteria, *Journal of Machine Learning Research* **3**: 1357–1370.
- Shashua, A., Gdalyahu, Y. & Hayun, G. (2004). Pedestrian Detection for Driving Assistance: Single-frame Classification and System Level Performance, *Proceedings of the IEEE Intelligent Vehicle Symposium (IV)*, Parma, Italy.
- Suard, F., Rakotomamonjy, A., Bensrhair, A. & Broggi, A. (2006). Pedestrian detection using infrared images and histograms of oriented gradients, *Proceedings of the IEEE Conference of Intelligent Vehicles (IV)*, Tokyo, Japan, pp. 206–212.
- Tipping, M. E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine, *Journal of Machine Learning Research* **1**: 211–244.
- Vapnik, V. (1998). *Statistical Learning Theory*, Wiley.

- Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features, *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 511–518.
- Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. & Vapnik, V. (2000). Feature Selection for SVMs, *Neural Information Processing Systems*, pp. 668–674.
- Yu, L. & Liu, H. (2003). Feature selection for high-dimensional data: a fast correlation-based filter solution, *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 856–863.

IntechOpen

IntechOpen



New Advances in Machine Learning

Edited by Yagang Zhang

ISBN 978-953-307-034-6

Hard cover, 366 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

The purpose of this book is to provide an up-to-date and systematic introduction to the principles and algorithms of machine learning. The definition of learning is broad enough to include most tasks that we commonly call “learning” tasks, as we use the word in daily life. It is also broad enough to encompass computers that improve from experience in quite straightforward ways. The book will be of interest to industrial engineers and scientists as well as academics who wish to pursue machine learning. The book is intended for both graduate and postgraduate students in fields such as computer science, cybernetics, system sciences, engineering, statistics, and social sciences, and as a reference for software professionals and practitioners. The wide scope of the book provides a good introduction to many approaches of machine learning, and it is also the source of useful bibliographical information.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Laetitia Leyrit, Thierry Chateau and Jean-Thierry Lapreste (2010). Classifiers Association for High Dimensional Problem: Application to Pedestrian Recognition, *New Advances in Machine Learning*, Yagang Zhang (Ed.), ISBN: 978-953-307-034-6, InTech, Available from: <http://www.intechopen.com/books/new-advances-in-machine-learning/classifiers-association-for-high-dimensional-problem-application-to-pedestrian-recognition>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen