IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 6, November 2010
ISSN (Online): 1694-0814
www.IJCSI.org

An Enhanced Algorithm Of The Statistical Training Method In Boosting-Based Face Detection

Said Belkouch[1], Mounir Bahtat[1], Abdellah Ait Ouahman[1], M. M'rabet Hassani[2]

[1] Micro-informatics, Embedded Systems and Systems On the Chips Lab., National School of Applied Sciences, University of Cadi Ayyad, Marrakech, Morocco

[2] Faculty of Sciences Semlalia, University of Cadi Ayyad, Marrakech, Morocco

Abstract

A trained cascade for face detection with a reduced number of Haar-like features should be computationally efficient. The accurate classical scheme for selecting these Haar-like features is proposed by Viola and Jones, but the training process may take weeks. Recently, there have been several heuristics reducing the training time in a dramatic way but the selected weak classifiers are not as good as those chosen by Viola and Jones, which leads to an increased number of features in the final cascade and then decreasing detection speed. Our method is an improved version of a statistical training method; it presents both faster selections and accuracy comparable to the Viola and Jones method.

Keywords— *face detection, Haar-like feature, weak and strong classifier, statistical training.*

1. Introduction

Face detection is a fundamental task for many applications such as face recognition, surveillance, smart homes and robotics. Several frameworks have been proposed to solve this open problem, from which, the Support Vector Machines (SVM) methods and especially the Viola and Jones algorithm [1] have gained a lot of interest as it can achieve very high detection rates in an extremely fast way over all existing efficient methods [2,3,4,5,6]. Viola and Jones framework is a cascade-based face detector, composed of a number of nodes where each node represents a classifier designed for a fast rejection of likely non-face sub-windows. These classifiers are constructed using a modified version of the AdaBoost algorithm [7] which is known to be very resistant to overfitting compared to other boosting methods [8]. A set of simple Haar-like features are used to train the classifiers.

Many extensions of this set are proposed in [9, 10, 11] yielding to a trained cascade with a fewer number of features. In the present article we don't focus on which feature set is more convenient for training a face detector. The exhaustive search over the feature set that is proposed by Viola and Jones makes training a cascade of classifiers a time consuming part as the algorithm complexity in that case is $O(NTlog(N))$ where N is the number of training images and T is the number of used features. This leads to a total time of weeks to train the full cascade. A first try to reduce this complexity was proposed in [12] reducing it to $O(NT)$ using caching. This implementation is called Faster AdaBoost, but uses very huge memory space, and becomes harder to implement when the feature set is wider. The most interesting heuristic reducing this complexity is the one proposed in [13], which is a statistical method that succeeded to break up the NT factor to a complexity of $O(Nd^2+T)$, where $d=24^2$. In this case training images are scaled to a resolution of 24x24. This method is able to perform extremely faster training at the cost of slight decrease in accuracy. The statistical heuristic is not sensible to enlarging the feature set in terms of training time, which gives an opportunity to extend the feature set by more complicated ones. This was not possible using the classical method as the training time increases in an exponential way with the features number. This heuristic is one of the best known strategies to train a boosting-based cascade of classifiers taking into account the compromise of speed and accuracy. Our algorithm is an improvement over the last mentioned statistical heuristic. It is faster and more precise, and with the help of parameters adjustment our algorithm is able to choose exactly the same weak classifiers as those chosen by the precise Viola and Jones method.

The key idea of our training method is a combination technique between the two discussed methods. Indeed, we observed that the estimated feature errors that were evaluated by the statistical algorithm could not be used as a decisive comparing tool between elements of the feature

set. For this reason, we exploited the statistical heuristic as a mean to candidate some good features and treat them just afterward in a more precise way using the classical algorithm of Viola and Jones. This increases the selection accuracy, and with some additional modifications that will be explained later we make our algorithm running faster.

The remaining parts of this paper will be presented as follow: the Viola and Jones framework is exposed in section 2. In section 3 we will describe the fast selection of Haar-like features using the statistical method. Our method is presented in section 4. Associated implementation and results are presented in section 5 and finally the conclusion is given in section 6.

2. Training of a Strong Classifier Using the Viola and Jones Method

The Viola and Jones architecture is based on a cascade of strong classifiers as illustrated in fig. 1, where each one is composed of a number of weak classifiers. Each weak classifier consists of three parameters: a Haar-like feature *f* from the set shown in fig. 2, a polarity *p* and a threshold θ. The weak classifier parameters as presented in equation (1) are chosen by the AdaBoost boosting algorithm. In this equation, $h_{p,\theta,f}(x)$ is the weak classifier function and *x* is the 24 x24 input image. This function returns 1 to indicate that *x* is a face and 0 otherwise.

$$h_{p,\theta,f}(x) = \begin{cases} 1 & if\ \ p\,f(x) < p\,\theta \\ 0 & otherwise \end{cases} \quad (1)$$

The feature value *f(x)* is computed by the sum of pixel intensities in the white region of the feature f applied on x subtracted from the sum of pixel intensities in the grey one.

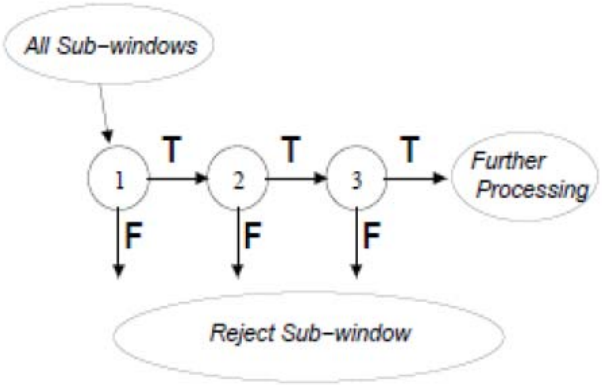

Figure 1: A cascade of classifiers

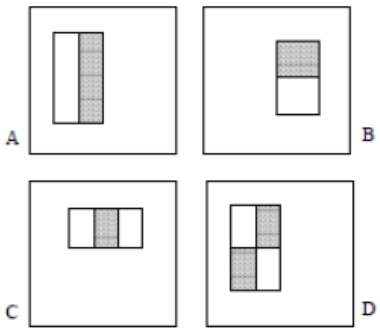

Figure 2. Four types of Haar-like features, have led to 134 736 different features

The weak classifiers are expected to have big error rates, but as long we combine more weak classifiers into the cascade architecture, the global error rate tends to 0.

The boosting algorithm requires a number of training images from two categories: face images and non-face images. In our experiments we have used 5000 face images from the LFW database [14] and a number of 10000 non-face images. AdaBoost serves to train a strong classifier for the cascade. It begins by initializing weights indicating the importance of each training image, and during each round of the algorithm a best weak classifier h_t is selected. Finally, as a preparation phase for the next round, the weak classifier error ε_t is computed and weights are updated so as the next weak classifier corrects the errors of the previous one. The final form of the trained strong classifier that contains *S* weak classifiers is given by equation (2):

$$h(x) = \begin{cases} 1 & if\ \ \sum_{t=1}^{S} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{S} \alpha_t \\ 0 & otherwise \end{cases} \quad (2)$$

Where $\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

The best weak classifier error is ε_t ; evaluated as :

$$\varepsilon_t = \sum_n \omega^{(n)} \mid h_t(x_n) - y_n \mid \quad (3)$$

Where $y_n = 1$ when the x_n image is a face and $y_n = 0$ otherwise.

The process of selecting the best weak classifier h_t is done by an exhaustive search over all possible weak classifiers, which it has a complexity of *O(NTlog(N))* for the Viola Jones algorithm. This implies that for each feature, a quick sort of N feature values is needed.

The cascade is finally constructed by a combination of many strong classifiers with a well-chosen number of contained weak classifiers in the purpose of boosting detectors speed.

3. Fast Selection of Haar-Like Features Using the Statistical Method:

The statistical method has improved the process of selecting the best weak classifier h_t in terms of training time. It succeeded to break up the *NT* factor, by treating separately training samples and features.

Instead of passing each feature on all the training images in order to determine its best parameters as it was the case in the Viola Jones algorithm, this method treats the *N* training samples only once per round. During this phase, statistics of these images are extracted and used later with each feature in order to compute the wanted parameters in a constant time. The best feature is then the one that best separates between feature values on face images and those on non-face images.

The feature value associated with the m^{th} feature is considered as a random variable that we will denote $v^{(m,c)}$ where $c = 1$ when applied on the face class or $c = 0$ otherwise.

The feature value consists of a linear expression of integral image values, then, if we denote $y^{(c)}$ the vectorized random variable form of the integral image from the c class, we will have:

$$v^{(m,c)} = y^{(c)^{\mathrm{T}}} g^{(\mathrm{m})} \tag{4}$$

Where $g^{(m)}$ is a vector describing the m^{th} feature.

The probabilistic law of the random variable $v^{(m,c)}$ is estimated to be a Gaussian distribution over training images :

$$v^{(m,c)} \sim \aleph\left(\mu^{(m,c)}, \sigma^{(m,c)^2}\right) \tag{5}$$

The mean parameter is given by:

$$\mu^{(\mathrm{m,c})} = < y^{(c)} >^{T} g^{(\mathrm{m})} \tag{6}$$

And the variance is :

$$\sigma^{(\mathrm{m,c})^2} = g^{(\mathrm{m})^{\mathrm{T}}} \delta^{(c)} g^{(\mathrm{m})} \tag{7}$$

Where $\delta^{(\mathrm{c})}$ is the covariance matrix of $y^{(\mathrm{c})}$ and is expressed by:

$$\delta^{(\mathrm{c})} = < y^{(c)} y^{(c)^{\mathrm{T}}} > - < y^{(c)} > < y^{(c)} >^{\mathrm{T}} \tag{8}$$

The mean sign <.> is computed taking into account the current weighing distribution for training images; then we have:

$$\mathrm{m_c} = < \mathrm{y^{(c)}} > = \frac{1}{\sum_{\mathrm{n,c}} \omega^{(\mathrm{n})}} \sum_{\mathrm{n}} \omega^{(\mathrm{n})} y^{(c,n)} \tag{9}$$

$$\delta^{(\mathrm{c})} = \left(\frac{1}{\sum_{\mathrm{n,c}} \omega^{(\mathrm{n})}} \sum_{\mathrm{n}} \omega^{(\mathrm{n})} y^{(c,n)} y^{(c,n)^{\mathrm{T}}} \right) - \mathrm{m_c} \mathrm{m_c}^{\mathrm{T}} \tag{10}$$

The example on fig. 3 and fig. 4 shows the estimated probabilistic density function of $v^{(m,c)}$ against the real one for a specified feature. We can see that the approximation is good enough regardless of a certain loss of precision.

Let's consider the following notations:

$$\begin{cases} \mathrm{u_1} = \sum_{\mathrm{n,c=1}} \omega^{(\mathrm{n})} & (11) \\ \mathrm{u_2} = \sum_{\mathrm{n,c=-1}} \omega^{(\mathrm{n})} & (12) \\ \mu_1 = \mu^{(\mathrm{m,c=1})} & (13) \\ \mu_2 = \mu^{(\mathrm{m,c=-1})} & (14) \\ \sigma_1 = \sigma^{(\mathrm{m,c=1})} & (15) \\ \sigma_2 = \sigma^{(\mathrm{m,c=-1})} & (16) \\ \mathrm{f_{\mu,\sigma}(x)} = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(\mathrm{x}-\mu)^2}{2\sigma^2}} & (17) \end{cases}$$

We suppose without loss of generality that:

$$\begin{cases} \mu_2 \geq \mu_1 \\ \mathrm{u_1} + \mathrm{u_2} = 1 \end{cases} \tag{18}$$

For each feature, the optimum polarity ($\mathrm{p_{opt}}$), the threshold (θ_{opt}) and the error ($\varepsilon_{\mathrm{opt}}$) are computed in constant time, in the following way:

$$\mathrm{p_{opt}} = \mathrm{sign}(\mu^{(\mathrm{m,c=-1})} - \mu^{(\mathrm{m,c=1})}) \tag{19}$$

The θ_{opt} parameter is the minimum of the error function $\varepsilon(\theta)$ which is defined below:

$$\varepsilon(\theta) = \mathrm{u_1} \int_{\theta}^{+\infty} \mathrm{f_{\mu_1,\sigma_1}(x)}\, \mathrm{dx} + \mathrm{u_2} \int_{-\infty}^{\theta} \mathrm{f_{\mu_2,\sigma_2}(x)}\, \mathrm{dx} \tag{20}$$

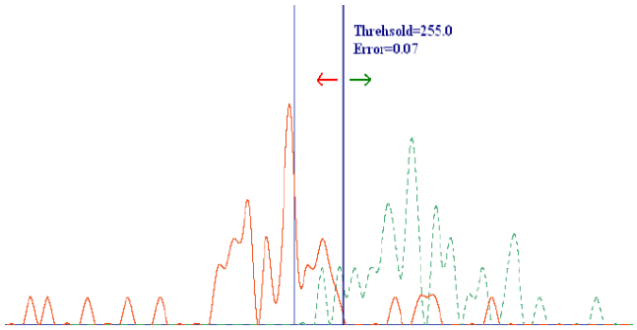

Figure 3: The real density functions for $v^{(8447,c)}$

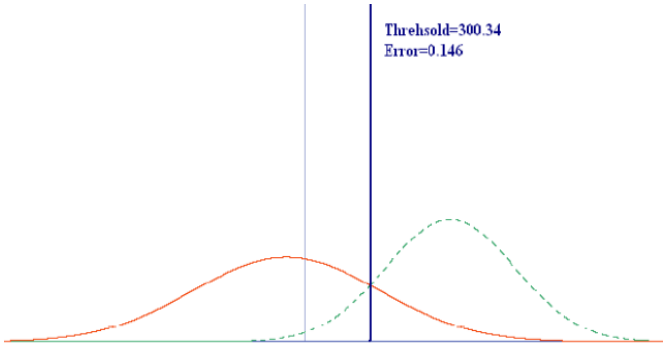

Figure 4. The estimated density functions for $v^{(8447,c)}$

The best error is then given by:

$$\varepsilon_{opt} = \varepsilon(\theta_{opt}) \quad (21)$$

The statistical selection part will then be done as the following form:

For each class c do
    Compute the vector m_c
    Compute the symmetric matrix $\delta^{(c)}$
End for
For each feature m
    For each class c
        Compute $\mu^{(m,c)}$
        Compute $\sigma^{(m,c)^2}$
        Compute p_{opt}, θ_{opt} *and* ε_{opt}
        Compare ε_{opt} *with the best* ε
    End For
End for

This has improved the complexity to $(Nd^2 + T)$; the bottleneck of this method is at computing the matrix $\delta^{(c)}$ which has a complexity of $O(Nd^2)$. It has been reported in [13] that it takes about 1.8 seconds computation for that part using the highly optimized algebra package GotoBLAS [15]. As we don't have this package we have used in our implementation the classical matrix multiplication algorithm which is less efficient than the GotoBLAS implementation. Therefore it takes more time to compute that same part of the algorithm.

4. Our Improved Algorithm For Boosting-Based Training:

The statistical training method is quite fast, but the process of weak classifiers selection terminates by choosing the wrong weak classifier due to the probability distribution estimation as shown in fig. 3 and fig. 4.

As explained in the introduction, our method combines both the statistical and the Viola Jones training methods in a cascaded architecture as shown in fig. 5. The statistical part function aims to eliminate likely non-convenient features very quickly, leaving only a small amount of features that contains most probably the best weak classifier. This smaller set is analyzed in a more precise way using the Viola and Jones algorithm, which can be also done in a short time due to the very few elements in the resulting feature set.

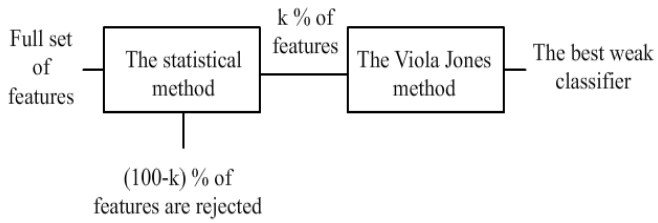

Figure 5: The architecture of our method

The reduction percentage k can be adjusted empirically; it controls a compromise between the accuracy and the speed. Then, according to our architecture, the training method is certainly going to be more accurate than the statistical method whatever k could be.

In order to select the best first k % elements, a sort operation is needed for the whole set of features according to their errors, which has a complexity of $O(T\ln(T))$; then the full complexity of the algorithm is $O(Nd^2 + Tln(T) + 10{-}2k\ NTlnN$. This complexity will take more time than the statistical method and k should be the smallest possible in order to have a short total training time. This first implementation will be denoted (α).

The first filtering block in our architecture which is the statistical algorithm doesn't make a decisive choice about the best weak classifier. It is permitted that it makes errors on evaluating each feature performance as these errors are compensated by a good choice of the k parameter. Then this architecture allows us to make some simplifications on the statistical algorithm in order to boost computation speed while a slightly modified k could remove the engendered errors. Thus, we have decided to operate on rescaled training images from 24×24 to 12×12 during the first filtering bloc. The second bloc will still be operating on 24×24 training images. This will have the effect to reducing the complexity term Nd^2 by a factor of 16. According to our experiments, if k is well-chosen, then the Nd^2 term is much greater in the complexity than other terms; then reducing it will reduce the training time significantly. This implementation will be called (β) and it is shown in fig. 6.

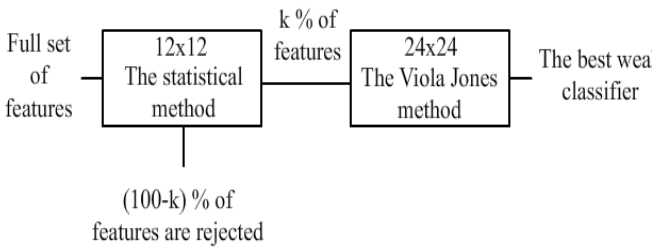

Figure 6. The (β) implementation

5. Implementation and Results

For our experiments we have used for training 5000 face images from the cropped version of the LFW (Labeled Faces in the Wild) that are distributed by the university of Massachusetts [14], and 10000 non-face images taken from the web, all training images have been rescaled to a 24×24 resolution.

Two different sets of features have been used. The first set that we will denote (1) contains 134 746 features using the 4 feature types presented in fig. 2. The second denoted (2) contains 723 448 features using 92 feature types and that some of them are presented in fig. 7.

We have developed a program that can generate the feature vectors $g^{(m)}$ after drawing the feature form manually, so that increasing the number of features can be done easily.

All experiments are done in a Core2 Quad 2.4 GHz CPU. Experimental results are presented in figs. 8 and 9. Performance is measured by comparing the evolution of the error rate in a strong classifier. Fig. 8 compares three implementations of our (α) method $(k = 1,\ 0.2\ and\ 0.01)$ with the Viola Jones implementation and the statistical implementation. Fig. 9 compares the (β) implementation in the same manner. These results are obtained using the feature set (1).

The obtained results in fig. 8 show that the Viola and Jones method converges faster than the statistical method, which means it uses only a small amount of features to reach the same performance. Indeed for example, 2.02 % error rate is reached with 20 features for Viola and Jones method while it needs 50 features for the statistical method.

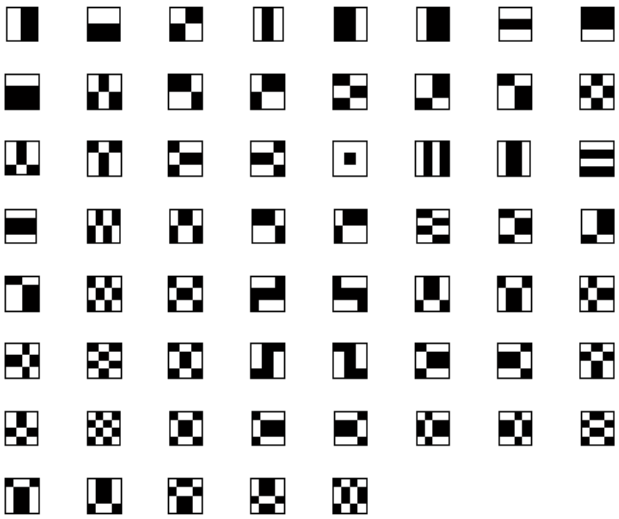

Figure 7.A part of the feature set (2)

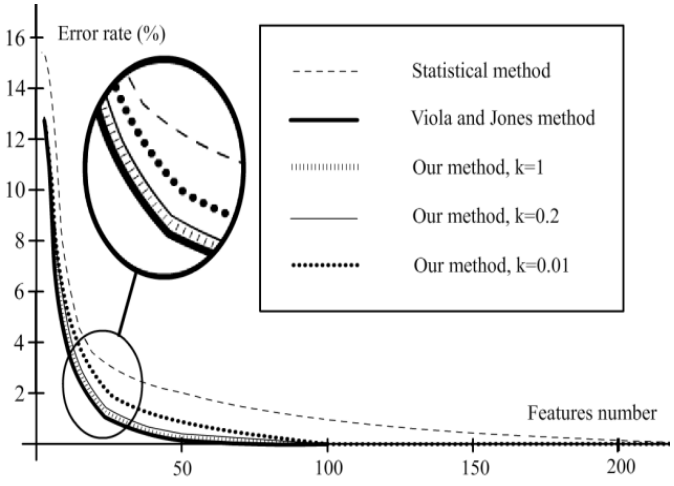

Figure 8: (α)implementation results

Our (α) implementation decreases significantly the error rate when compared to the statistical method, even when choosing $k = 0.01$ with which we have the fastest (α) implementation. In this case the full feature set is decreased by a factor of 10^4 (i.e. 0.01%) to the second part of our cascaded architecture.

In fig. 9, the (β) implementation has been used. It presents higher error rate than the (α) implementation, but still converging more quickly than the statistical method resulting in significant decrease in training time as shown in Table 1. The statistical method takes 31 seconds in our implementation, which is greater than that mentioned in [13], that's due as explained in section 3 to their use of a non-classical multiplication algorithm that's presented in the highly optimized algebra package GotoBLAS. If the algebra package had been used in our case, the (α) and (β) methods training time would further decrease as well. The (α) implementation takes slightly more training time than the statistical method as shown in table 1 but shows a huge increase in accuracy as illustrated in fig. 8. Furthermore, for k=1 our method presents almost the same accuracy as the Viola and Jones method by the approximately superposed curves while it is worth noticing that the training time takes 48 seconds in our case and 31 minutes with Viola and Jones Method. The (β) implementation loses some accuracy comparatively to (α) implementation as showed in fig. 9, but decreases the training time significantly.

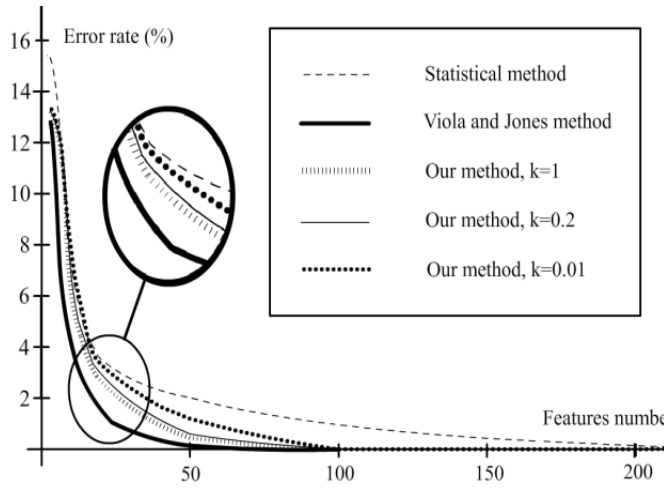

Figure 9: (β)implementation results

We notice that the (β) method has improved accuracy over the statistical method and furthermore decreases the training time as low as 2 seconds for k = 0.01. To the best of our knowledge this is the fastest training method in boosting-based face detection using Haar-like features. An equivalent statistical method using a rescaling to 12×12 would run as fast as our (β) method, but would increase error rates to huge numbers according to our experiments.

Table 1: Training time on the feature set (1)

Method	*Training time of a weak classifier*
Statistical method	31 seconds
Viola and Jones method	31 minutes
Our (α) method k=1	48 seconds
Our (α) method k=0.2	34 seconds
Our (α) method k=0.01	31seconds
Our (β) method k=1	16 seconds
Our (β) method k=0.2	5 seconds
Our (β) method k=0.01	2 seconds

Fig. 10 shows the effect of increasing the features number. The feature set (2) has been used with the (α) implementation of our method for k=1 and it is compared to the Viola and Jones method using the feature set (1). The figure shows that the error rate has been decreased in comparison with the Viola and Jones method. Besides, our method allows to increase the number of features in the training process without significant increase in the training time as shown in table 2. It is worth to remember that it does not make a sense to use Viola and Jones with feature set (2) because training then would not be praticly possible. Indeed as shown in table 2, training time of only one weak classifier takes 3.2 hours.

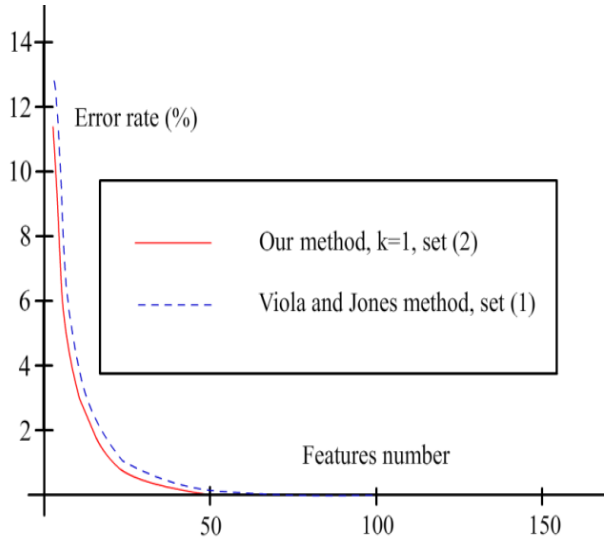

Figure 10. The effect of increasing the number of features using the (α) implementation

Table 2: Training time on the feature set (2)

Method	*Training time of a weak classifier*
Statistical method	50 seconds
Viola and Jones method	3.2 hours
Our (α) method k=1	2.6 minutes
Our (α) method k=0.2	1.1 minutes
Our (α) method k=0.01	51 seconds
Our (β) method k=1	1.6 minutes
Our (β) method k=0.2	30 seconds
Our (β) method k=0.01	14 seconds

6. Conclusion:

In this paper we present a fast and accurate selection method of Haar-like features, two implementations have been proposed according to the same architecture. The (α) implementation presents a high accuracy that can be comparable to the Viola and Jones exhaustive search method while slightly increasing the training time relatively to the statistical method, the (β) implementation loses some accuracy while still being more accurate than the statistical algorithm, besides, it decreases the training time over all known existing Haar-like selection strategies. Our method also allows enlarging the feature set so as to attain better performance in terms of convergence over the training set.

REFERENCES

[1] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), *vol. 1, pp.* 511–518, 2001
[2] K. Sung and T. Poggio, "Example-Based Learning For View-Based Face Detection," IEEE Patt.Anal. Mach. Intell., vol. 20, pp. 39–51, 1998
[3] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," IEEE Patt.Anal. Mach. Intell,vol. 20, pp. 22–38, 1998
[4] H. Schneiderman and T. Kanade, "A Statistical Method For 3D Object Detection Applied To Faces And Cars," IEEE Conference on Computer Vision and Pattern Recognition. Proceedings 2000.. , *USA, vol. 1, pp.* 746 – 751, 2000
[5] M. Yang, D. Roth, and N. Ahuja. "A Snowbased Face Detector". In NIPS-12; Conference on Advances in Neural Information Processing, Systems, pp. 855,-861. MIT Press, 2000.
[6] Y. Amit, D. Geman, and K. Wilder, "Joint induction of shape features and tree classifiers," IEEE Transaction on Pattern Analysis And Machine Intelligence , vol. 19, no. 11, pp. 1300-1305, 1997
[7] Y. Freund and R. E. Schapire. "Experiments With a New Boosting Algorithm. In Machine Learning," In Proceedings of the Thirteen International Conference In Machine Learning, Bari, pages 148–156, 1996.
[8] P. L. Bartlett and M. Traskin, "AdaBoost is consistent," Journal of Machine Learning Research, vol. 8, pp. 2347–2368, 2007.
[9] R. Lienhart and J. Maydt, "An Extended Set of Haar-Like Features For Rapid Object Detection," IEEE 2002 International Conference on Image Processing, Vol. 1, pp. 900-903, Sep. 2002.
[10] T. Mita, T. Kaneko, O. Hori, "Joint Haar-Like Features For Face Detection," Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005, Beijing. Vol. 2, pp. 1619-1626 , 2005
[11] K. Masada, Q. Chen, H. Wu and T. Wada, "GA Based Feature Generation For Training Cascade Object Detector," 19th International Conference, ICPR 2008, pp.1-4, Tampa FL, *2008*
[12] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast asymmetric learning for cascade face detection. IEEE Trans Pattern Anal Mach Intell. 2008 Mar;30(3):369-82.
[13] M.T.Pham and T.J.Cham, "Fast Training And Selection of Haar Features Using Statistics in Boosting-Based Face Detection," In Proc. International Conference on Computer Vision (ICCV 2007), Rio de Janeiro, Brazil, 2007
[14] Source "Labeled Faces in the Wild Database", Computer Visions Lab., University of Massachussets, 2010, "http://vis-www.cs.umass.edu/lfw/
[15] K. Goto and R. van de Geijn, "High-performance implementation of the level-3 Blas," FLAME Working Note #20. The University of Texas at Austin, Department of Computer Sciences. Technical Report TR-2006-23

Said Belkouch has completed his PhD in Microelectronics at University Joseph Fourier-Grenoble in France in 1989. From 1989 to 2003, he worked respectively as Assistant Researcher at University of Sherbrooke in Canada, Research Officer at National Council of Canada, and embedded ASICs Design Engineering at Tundra Semiconductor Corporation, Ottawa, Canada (Tundra acquired recently by Integrated Devices Technology Company). Since 2003, He is professor at Electrical Engineering Department, National School of Applied Sciences-Marrakech, Morocco. His area of research includes embedded systems and

microelectronics. He has published several research papers in Journals and Proceedings.

Mounir Bahtat is a Master and Engineer student at Micro-informatics, Embedded Systems and Systems On the Chips Lab

Abdellah Ait Ouahman received the doctorate thesis in Signal Processing from the University of Grenoble, France, in November 1981. His research was in Signal Processing and Telecommunications. Then he received the PhD degree in Physics Science from the University of Sciences in Marrakech , Morocco, in 1992. He is now Professor and responsible of the Telecommunications and Computer Science and Networking laboratory in the Faculty of Sciences Semlalia in Marrakech, Morocco. His research interests include the signal and image processing and coding, telecommunications and networking. Actually he is a director of National School of Applied Sciences , Marrakech. He has published several research papers in Journals and Proceedings.

Moha M'Rabet Hassani. He received the a doctorate thesis in automatic from Nice University, France, in 1982 and Ph. D degree in electrical engineering from Sherbrook university, Canada, in 1992. He is now Professor in the Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco. He heads at the University both the Doctoral Studies Centre of Science and Technology and Electronics and Instrumentation Lab. His research interests are in statistical signal processing, nonlinear system, identification and equalization fields.