Comput. Sci. Appl.
Volume 2, Number 2, 2015, pp. 64-70
Received: December 30, 2014; Published: February 25, 2015

Computer Science
and Applications
www.ethanpublishing.com

# A Convex Set Based Algorithm to Automatically Generate Haar-Like Features

César Cobos-May, Víctor Uc-Cetina, Carlos Brito-Loeza and Anabel Martin-Gonzalez

*Faculty of Mathematics, Autonomous University of Yucatan, Merida 97119, Mexico*

**Corresponding author:** Víctor Uc-Cetina (uccetina@uady.mx)

**Abstract:** Design of Haar-like features for pattern recognition is currently carried on fully empirically. Users of this kind of features can whether rely on a fix set of general purpose features, or they can design new ones that according to their own criteria could better capture the pattern to be detected. As a result of this approach, different programmers generate different features. In this paper, the authors introduce an algorithm based on convex set theory aimed at generating Haar-like features automatically. The algorithm generates Haar-like features by approximating close curves with rectangles up to some degree of precision defined by the user. The main input of the algorithm is a set of sample images already segmented, and the output is a set of Haar-like features templates ready to be applied. Although the idea is simple, it is sound, it provides good performance and most importantly, it constitutes a first step towards the automation of Haar-like features design. The authors validate experimentally the efficiency of their method by detecting the eye optical nerve in retinal images. Results of the experimental work suggest that it is possible to get a detection performance similar to that obtained by Viola and Jones using the typical Haar-like features, but with fewer and automatically generated features.

**Keywords:** Haar-like features, AdaBoost, pattern recognition, classification.

## 1. Introduction

Object detection is an essential, though challenging vision task. It plays an important role in many applications such as image search and scene understanding.

The main goal of object detectors is to find image sub-regions containing instances of the object of interest. This can be achieved, for example, by analyzing pure image intensities (pixel values), geometrical forms (edges, circles), or colors. Nevertheless, for complex objects with a strong appearance variation, Haar-like features have been widely used [1], since they represent information about the object that is not explicitly present in the image.

In 2001, Viola and Jones introduced a machine learning framework based on Haar-like features with a real-time object detection scheme [2], capable of evaluating any Haar-like feature at any scale in constant time. Such framework is based on an efficient image representation called integral image, a boosting learning algorithm, and cascade classifiers.

Although it is more than 10 years since this method was proposed, it is still being used for training robust systems which require fast patterns detection. It has been applied not only on face detection [3], but also to detect or recognize many other kind of objects such as pedestrians [4], hand gestures [5, 6], text within images [7], vehicle detection [8, 9], and even avatar face detection in virtual worlds [10].

According to the state of the art, Haar-like features are typically designed empirically based on the general shape and colors of the object to be detected.

In this paper, we introduce an algorithm based on convex set theory aimed at generating optimal Haar-like features automatically. The main input of our algorithm is a set of sample segmented images of the object of interest, and the output is a set of

Haar-like features templates ready to be applied under Viola and Jones computational scheme. In Section 2, we give an introduction of Haar-like features, their current design and use. In Section 3, a detailed description of our algorithm and its basic properties, under some reasonable constraints, are given. In Section 4, we validate experimentally the efficiency of our method by solving a pattern recognition problem involving the detection of the eye optical nerve on retinal fundus images. Finally, in Sections 5 and 6, we discuss and conclude the paper explaining how our results suggest that it is possible to get a detection performance similar to that obtained by Viola and Jones using typical Haar-like features, but with fewer and automatically generated features.

## 2. Haar-Like Features

In computer vision, a feature is some information relevant for solving a task [11]. In general, a feature could be (a) the result of an operation between neighboring pixels in a region of an image, or (b) some specific structures in the image, going from simple points and borders to more complex things such a complete objects. The feature concept is very general and their selection is highly dependent on the problem to be treated [12].

Haar-like features are a special kind of image features introduced by Papageorgiou et al., as a general framework for object detection in 1997 [13]. They received their name from their similarity to Haar wavelets already introduced by Alfred Haar in 1910 [14]. Being an over-complete set of two dimensional Haar functions, Haar-like features represent implicit object characteristics encoded in the pixel intensities of an image, indicating the presence or absence of such characteristics.

Haar-like features consist of two or more rectangular regions contained in a template (Fig. 1). Traditionally, each region has an assigned weight, such that the sum of regions weight equals zero. For example, light regions weight equals 1, and dark regions weight equals −1.

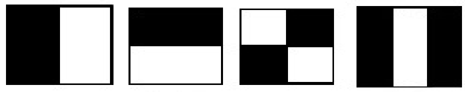

**Fig. 1 Haar-like features templates defining two-rectangle (vertical and horizontal), four-rectangle (diagonal) and three-rectangle features.**

The feature value is the result of the template convolution on the pixel image. Due to the template weights, this value is calculated as the result of subtracting the sum of all pixel intensities in one region (dark region) from the sum of all pixel intensities in the other region (light region).

The first basic Haar-like features proposed by Papageorgio et al., consisted of two-rectangle templates (vertical and horizontal) and a four-rectangle template (diagonal) [1].

Later in 2001, Viola and Jones proposed a three-rectangle feature [2], increasing the original set of features proposed by Papageorgiou. This expanded set of features, together with their novel concept of integral image and the use of the AdaBoost algorithm [15] in cascades of classifiers made possible the construction of a face detector with real-time performance.

In 2002, Lienhart and Maydt extended the set of Haar-like features with a set of rotated features by 45° [16], where its computational time was constant as well as basic Haar-like features. Few years later, new rotated Haar-like features were proposed, this time with rotation angles of 26.5° and 63.5° [17].

In 2010, Pavani et al., proposed to replace the default weights assigned to the rectangles of the Haar-like features with optimal weights, maintaining the simplicity of the traditional ones, while being more discriminative [18]. This approach was applied to find cardiac structures from magnetic resonance imaging and for face detection, obtaining better accuracy and speed of detection.

More recently, Haar-like features have also been applied with other machine learning classifiers to improve detection tasks performance. In 2013, Wei et al. used the AdaBoost algorithm, Haar-like features, HOG (histogram of oriented gradients) descriptors, and SVM classifiers to obtain a higher speed pedestrian classifier [19]. A different approach, proposed by Nasrollahi et al. [20, 21], used a large set of Haar-like features to feed a PNN (probabilistic neural networks) with sets of 115 and 61 features, obtaining good results in a biometric recognition and face detection system.

At present, Haar-like features are one of the most used concepts for real-time object detection. Nevertheless, all these have been designed empirically based on the experimental knowledge of the observer.

In the next section, we introduce and algorithm for automatically computing optimal Haar-like features based on convex set theory and the shape of the pattern to be detected.

## 3. Automatic Generation of Haar-like Features

Let $Z$ be a simple closed curve such that *int*($Z$) is a connected, convex and bounded region, and let $h$ be a horizontal line that cuts $Z$ in at least two points.

Let $z_1^h$ and $z_2^h$ be two points in the line $h$ cutting the region $Z$, with $[z_1^h]_x < [z_2^h]_x$ and such distances between $[z_1^h]_x$ and $[z_2^h]_x$ be a maximum, this is, for any other pair of points $z_3^h$ and $z_4^h$ in which $h$ cuts $Z$, we have

$$\left|[z_2^h]_x - [z_1^h]_x\right| \geq \left|[z_4^h]_x - [z_3^h]_x\right| \qquad (1)$$

We take now the vertical lines $v_1$ and $v_2$ that go through the points $z_1^h$ and $z_2^h$, respectively (Fig. 2), and consider the sets:

$A = \{a \in R | a$ is a cutting point between $v_1$ and $Z\}$

$B = \{ b \in R | b$ is a cutting point between $v_2$ and $Z\}$

We have $A \neq \emptyset \neq B$ given that at least $z_1^h \in A$ and $z_2^h \in B$.

Now, consider the sets:

$$D_A = \{\|z_1^h - a\|, \text{ for all } a \in A\} \qquad (2)$$

$$D_B = \{\|z_2^h - b\|, \text{ for all } b \in B\} \qquad (3)$$

and define

$$\hat{a} = \max_a D_A \qquad (4)$$

$$\hat{b} = \max_b D_B \qquad (5)$$

$$c = \min \{\hat{a}, \hat{b}\} \qquad (6)$$

Thus, $c = (c_1, c_2)$ for $c_1 = [z_1^h]_x$ or $c_1 = [z_2^h]_x$, and some $c_2 \in R$.

Given the horizontal line $h$, we form the rectangle $R_h$ with vertices in $z_1^h, z_2^h, \left([z_1^h]_x, c_2\right)$ and $\left([z_2^h]_x, c_2\right)$, as illustrated in Fig. 3. Notice that the area of that rectangle is

$$A(R_h) = \left([z_2^h]_x - [z_1^h]_x\right)\left|c_2 - [z_1^h]_y\right| \qquad (7)$$

Now, consider all possible horizontal lines $h$ that cut $Z$ in at least two points, and the rectangles $R_h$ associated to them, we denote

$h_{\max} = \max_h \{A(R_h) | h$ horizontal line that cuts $Z$ in at least two points$\}$

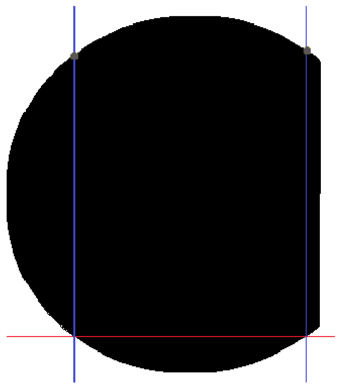

**Fig. 2 Line *h* is parallel to the *x* axis and is depicted in red, lines $v_1$ and $v_2$ are parallel to the *y* axis and are depicted in blue, and points $\hat{a}$, $\hat{b}$ are depicted in grey.**

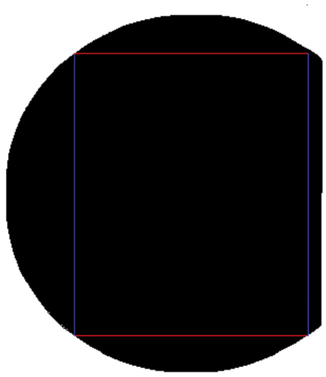

**Fig. 3 Rectangle $R_h$.**

$$R_{\max} = R_{h_{\max}} \tag{8}$$

In summary, the proposed algorithm is

Step 1. Draw a horizontal line $h$ that cuts the curve $Z$ in at least two points.

Step 2. Draw the vertical lines $v_1$ and $v_2$ that go through the first and last cutting point of $h$ with $Z$, sorted in ascending order of their $x$ coordinate.

Step 3. Pick the shortest vertical line contained in $Z$.

Step 4. Complete the rectangle that has base $h$ and height $v_1$ or $v_2$, accordingly to the selected line in Step 3.

Step 5. Repeat Steps 1 to 4 for all horizontal line $h$ that cuts in at least two points $Z$, and choose the rectangle with biggest area.

**Theorem 1.** Let $Z$ be a simple close curve such that $int(Z)$ is a convex region. $R_{\max} \subset int(Z)$.

Proof: Given that $int(Z)$ is a convex region, it is enough to prove that all four vertices of $R_{\max}$ are contained in $int(Z)$ to prove that the whole rectangle is also contained in $int(Z)$. Let $z_1^h, z_2^h, \left(\left[z_1^h\right]_x, c_2\right)$, and $\left(\left[z_2^h\right]_x, c_2\right)$ be the vertices of $R_{\max}$ and $\hat{b} = (b_1, b_2)$ constructed according to the algorithm. Without loss of generality, assume that $c_1 = \left[z_1^h\right]_x$ (the proof is analogous for the other side), then $c = \left(\left[z_1^h\right]_x, c_2\right)$. With such construction we have that $z_1^h, z_2^h, c \in int(Z)$, thus, we still need to prove that $\left(\left[z_2^h\right]_x, c_2\right) \in int(Z)$, but this is true given that $\left[z_2^h\right]_x \in \left[\left[z_1^h\right]_x, \left[z_2^h\right]_x\right]$ and $c_2 \in \left[\left[z_1^h\right]_y, b_2\right]$ (if $\left[z_1^h\right]_y \leq b_2$), or $c_2 \in \left[b_2, \left[z_1^h\right]_y\right]$ (if $b_2 \leq \left[z_1^h\right]_y$), with $b_2 \in Z$. Therefore, $R_{\max} \subset int(Z)$.

With this algorithm, given a horizontal line $h$, we have found the biggest rectangle contained in a region, having as its base the line $h$. However, that rectangle is not the biggest rectangle contained in the region. This issue can be solved by iteratively rotating the image and computing the rectangle, saving always the angle that produces the biggest rectangle.

The resulting templates when our algorithm is applied to some basic geometric forms are shown in Figs. 4-6, for one square, one triangle and one circle, respectively.

## 4. Experimental Validation

To validate our algorithm we performed pattern recognition experiments on images of retinal fundus from the Standard Diabetic Retinopathy Database -- DIARETDBI [22] from the Technological University of Lapperanta (Finland). Some examples of the images are shown in Fig. 7. The classification problem consisted of detecting the presence or absence of the eye optical nerve in the image.

We created our own database with a total of 100 images of size 174 × 192 pixels (subsamples of original images). Fifty images are optical nerves and fifty

**Fig. 4 Resulting Haar-like feature template when the input image is a square, for *n* = 1.**

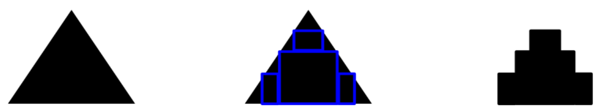

**Fig. 5 Resulting Haar-like feature template when the input image is a triangle, for *n* = 4.**

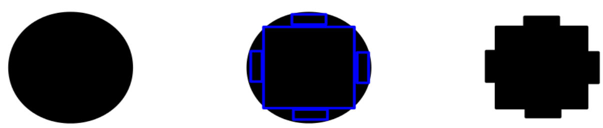

**Fig. 6 Resulting Haar-like feature template when the input image is a circle, for *n* = 5.**

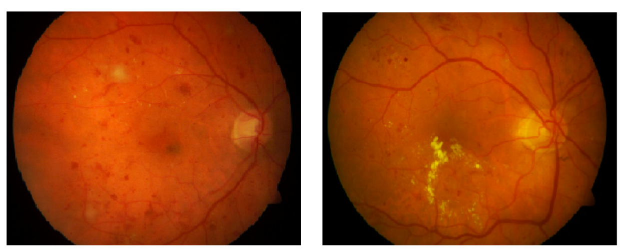

**Fig. 7 Some examples of retinal fundus images from the Standard Diabetic Retinopathy Database.**

images are other structures of the retinal fundus, but the optical nerve. Some of these images are shown in Fig. 8. A training set with 90% of the images and a test set with the remaining 10% were separated. The training set contains forty-five positive samples (optical nerves) and forty-five negative samples (other retinal regions). The test set contains five positive and five negative samples.

The experimental validation was performed using a 10-fold cross-validation [23] and the standard AdaBoost algorithm [15]. The size of the Haar-like feature templates automatically generated were computed to be the same as the size of our database images, this is 174 × 192 pixels.

In order to generate the Haar-like feature template, a representative optical nerve image from the positive sample collection was selected. This image was processed with the K-means algorithm (K = 2) to segment the main structure of the optical nerve. Later on, our method to automatically create the template was applied to the regions of the segmented image. The image selected and the resulting template are shown in Fig. 9. A mirrored version of the template is also used as a second Haar-like feature template for the detection experiments (Fig. 10). The 10-fold cross-validation results are given in Table 1.

As we can observe, 10 different partition sets were used as test sets to validate the model. The false negatives are the subset of images containing the optical nerve, but classified as non-containing it. The false positives are those images classified as non-having the optical nerve, but in fact they do contain the optical nerve.

**Fig. 8 Some images employed in our experimental work. From left to right, the first three images are optical nerves, and the remaining three are other retinal regions.**

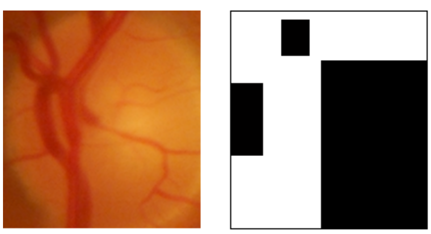

**Fig. 9 Original image and the resulting Haar-like feature template.**

With the two Haar-like features templates automatically generated, the system detection rate is 96% (mean), with sensitivity (true positive rate) of 98% and specificity (true negative rate) of 94%.

### *4.1 Comparative Experiment with Typical Haar-Like Features*

The typical Haar-like features, shown in Fig. 11, were employed to detect the optical nerve, under the same conditions as the automatically generated features presented before. The detection results are given in Table 2.

With typical Haar-like features, the boosting algorithm has a detection rate of 92% (mean), with a sensitivity of 90% and a specificity of 94%. From Tables 1 and 2, it is clear that the two features automatically generated provide a higher performance

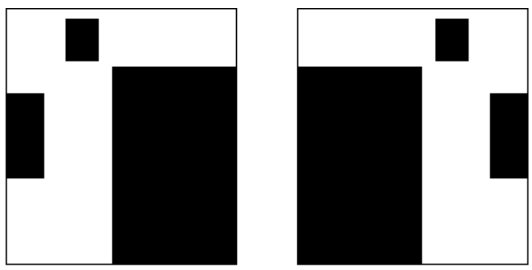

**Fig. 10 Haar-like feature template and its mirrored version.**

**Table 1 Optical nerve detection results using the set of automatically generated Haar-like features and 10-fold cross-validation.**

| Fold | False negatives | False positives | Total error |
|---|---|---|---|
| 1 | 0/5 | 0/5 | 0/10 |
| 2 | 0/5 | 0/5 | 0/10 |
| 3 | 0/5 | 0/5 | 0/10 |
| 4 | 0/5 | 0/5 | 0/10 |
| 5 | 0/5 | 0/5 | 0/10 |
| 6 | 0/5 | 0/5 | 0/10 |
| 7 | 1/5 | 0/5 | 1/10 |
| 8 | 0/5 | 1/5 | 1/10 |
| 9 | 0/5 | 2/5 | 2/10 |
| 10 | 0/5 | 0/5 | 0/10 |

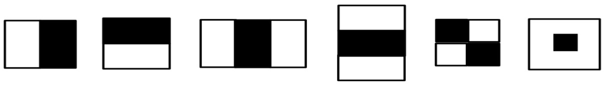

**Fig. 11 Typical Haar-like features.**

**Table 2 Optical nerve detection results using the set of typical Haar-like features and 10-fold cross-validation.**

| Fold | False negatives | False positives | Total error |
|---|---|---|---|
| 1 | 0/5 | 0/5 | 0/10 |
| 2 | 0/5 | 0/5 | 0/10 |
| 3 | 0/5 | 1/5 | 1/10 |
| 4 | 2/5 | 0/5 | 2/10 |
| 5 | 0/5 | 0/5 | 0/10 |
| 6 | 0/5 | 1/5 | 1/10 |
| 7 | 1/5 | 0/5 | 1/10 |
| 8 | 2/5 | 1/5 | 3/10 |
| 9 | 0/5 | 0/5 | 0/10 |
| 10 | 0/5 | 0/5 | 0/10 |

than using the six typical features. One key fact may be the reduced number of features needed to improve the performance. On the other hand, using two features instead of six reduces the total time of the learning process to only one third of the total time.

## 5. Discussion

According to our experimental results, the automatic generation of Haar-like features seems to be very useful and effective. However, it also has some drawbacks. Next, we summarize some advantages and disadvantages of this algorithm.

Advantages: First of all, the algorithm replaces the empirical design of the Haar-like features, which is very subjective to the decisions of the programmer, by an automatic method based on convex sets theory. Second, it gives the possibility of solving the pattern detection problem with fewer features, reducing the time used for training the classifier model. Finally, since the generation of the templates can be performed automatically, it is much more practical to apply as a general approach to different kind of patterns to be detected.

Disadvantages: First, the final Haar-like feature templates generated by our algorithm tend to contain more rectangles than the typical templates. As a result of this the computation time of the feature is increased in the number of extra rectangles to be computed. However, we must not forget that the computation of a rectangle via integral images is constant independently of its dimensions. Second, it requires a previous computation process to generate the feature templates with complexity $O(k^2)$. Finally, when the object to be detected appears in different shapes or positions, we will probably need to generate a representative Haar-like feature template for each of these shapes or positions. However, this is also a problem occurring currently with the typical templates.

We have proposed an algorithm for the automatic generation of Haar-like features. This is to the best of our knowledge the first time that this problem is formally approached. It has been a common practice for many years to generate the feature templates by hand design or some programmers simply use the typical set of features proposed by Viola and Jones in 2001 for detecting faces.

Our method consists in approximating a convex region with the biggest rectangle enclosed in it. By focusing in the biggest rectangle we reduce the final number of rectangles needed to form a template and this reduces the computation of the final feature during training and testing. Since the algorithm to generate the templates is applied to a segmented image, a natural question that rises is: What is the best method to segment the image previous to the application of our algorithm? In our experiments we segmented the images using k-means and obtained good results, but this cannot be considered as the only or the best method to segment the images.

## 6. Conclusions

We introduced an algorithm based on convex set theory to automatically generate Haar-like features. We validated the effectiveness of our method experimentally by solving a detection problem. We tested the performance of our algorithm against typical Haar-like features used in most implementations of the AdaBoost algorithm. Our 10-fold cross-validation results suggest a better performance of our automatically generated features in 4% more than the typical templates.

## References


[1] C. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: 6th International Conference on Computer Vision, 1998, India.

[2] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR 2001), 2001, pp. 511-518.

[3] M. Rezaei, H. Nafchi, S. Morales, Global Haar-like features: A New extension of classic Haar features for efficient face detection in noisy images, in: Proceedings of the 6th Pacific-Rim Symposium (PSIVT 2013), Guanajuato, Mexico, 2013, pp. 302-313.

[4] S. Zhang, C. Bauckhage, A.B. Cremers, Informed Haar-like features improve pedestrian detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 947-954.

[5] Q. Chen, N. Georganas, Hand gesture recognition using Haar-like features and stochastic context free grammar, IEEE Transactions on Instrumentation and Measurement 57 (2008) 1562-1571.

[6] M.A. Rahaman, M. Jasim, M.H. Ali, M. Hasanuzzaman, Real-time computer vision-based Bengali sign language recognition, in: Proceedings of International Conf. on Computer and Information Technology, 2014, pp. 192-197.

[7] N. Mavaddat, T. Kim, R. Cipolla, Design and evaluation of features that best define text in complex scene images, in: IAPR Conference on Machine Vision Applications (MVA2009), Yokohama, Japan, May 20-22, 2009.

[8] S.M. Elkerdawi, R. Sayed, M. ElHelw, Real-time vehicle detection and tracking using Haar-like features and compressive tracking, in: Proceedings of 1st Iberian Robotics Conference (ROBOT2013), pp. 381-390.

[9] A. Hasselhof, A. Kummert, A vehicle detection system based on Haar and triangle features, in: Proceedings of IEEE Intelligent Vehicles Symposium, 2009, pp. 261-266.

[10] D. D'Souza, R.V. Yampolskiy, Baseline avatar face detection using an extended set of Haar-like features, in: Proceedings of the 23rd Midwest Artificial Intelligence and Cognitive Science Conference, 2012, pp. 72-77.

[11] L. Shapiro, G. Stockman, Computer Vision, Prentice Hall, 2001.

[12] M. Nixon, A. Aguado, Feature Extraction and Image Processing, 1st ed., Newnes, 2002.

[13] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio, Pedestrian detection using wavelet templates, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 193-199.

[14] A. Haar, For the theory of orthogonal function systems, Mathematical Annals 69 (3) (1910) 331-371. (in German)

[15] R. Schapire, Y. Freund, Boosting: Foundations and Algorithm, Massachussets Institute of Technology, 2012.

[16] R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, in: Proceedings of International Conference in Image Processing, 2002, pp. 900-903.

[17] A. Barczak, M. Johnson, C. Messom, Real-time computation of Haar-like features at generic angles for detection algorithms, Research Letters in the Information and Mathematical Sciences 9 (2006) 98-111.

[18] S. Pavani, D. Delgado, A. Frangi, Haar-like features with optimally weighted rectangles for rapid object detection, Pattern Recognition 43 (1) (2010) 160-172.

[19] Y. Wei, Q. Tian, T. Guo, An improved pedestrian detection algorithm integrating Haar like features and HOG descriptors, Advances in Mechanical Engineering 2013 (2013) 546206.

[20] K. Nasrollahi, T.B. Moeslund, M. Rashidi, Haar-like rectangular features for biometric recognition, in: Proceedings of International Conference in Biometrics (ICB 2013), pp. 1-7.

[21] K. Nasrollahi, T. Moeslund, Haar-like features for robust real-time face recognition, in: IEEE International Conference on Image Processing, 2013.

[22] http://www2.it.lut./project/imageret/diaretdb1/ (accessed Aug. 1, 2014).

[23] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer, 2001.