Comparative Testing of Face Detection Algorithms*

Nikolay Degtyarev and Oleg Seredin

Tula State University
http://lda.tsu.tula.ru
n.a.degtyarev@gmail.com

Abstract. Face detection (FD) is widely used in interactive user interfaces, in advertising industry, entertainment services, video coding, is necessary first stage for all face recognition systems, etc. However, the last practical and independent comparisons of FD algorithms were made by Hjelmas et al. and by Yang et al. in 2001. The aim of this work is to propose parameters of FD algorithms quality evaluation and methodology of their objective comparison, and to show the current state of the art in face detection. The main idea is routine test of the FD algorithm in the labeled image datasets. Faces are represented by coordinates of the centers of the eyes in these datasets. For algorithms, representing detected faces by rectangles, the statistical model of eyes' coordinates estimation was proposed. In this work the seven face detection algorithms were tested; article contains the results of their comparison.

Key words: face detection, face localization accuracy, comparative test, face datasets

1 Introduction

Face detection tasks are becoming required more frequently in the modern world. It's caused by the development of security systems as an answer to acts of terrorism. In addition, these algorithms are widely used in interactive user interfaces, in advertisement industry, entertainment services, video coding, etc. However, many researchers mostly paid their attention to Face Recognition algorithms[6] considering Face Detection tasks (necessary first stage for all face recognition systems) to be almost solved. Thus, as far as we know, the last practical and independent comparisons of FD algorithms were made by Hjelmas et al. [4] and by Yang et al.[17] in 2001.

Nevertheless, "due to the lack of standardized tests"[4] most of such researches (including two mentioned above) "do not provide a comprehensive comparative evaluation" and contain only a summary of the originally reported performance among several face detection algorithms on the pair of small datasets. We are sure that this type of comparative testings can hardly represent "true"

* This work is supported by Russian Foundation for Basic Research (Gr. 09-07-00394).

performance, because the reported results could be based on different evaluation methods and parameters; could be adjusted to demonstrate better performance under controlled circumstances; etc.

The aim of this work is to propose parameters of FD algorithms quality evaluation and methodology of their objective comparison, and to show the current state of the art in face detection. Also it's should be stressed that a correct experiment should consists of two parts: algorithms learning on the training set and comparative testing. Unfortunately, we are not able to train all algorithms on the same data for several reasons. However, we believe that this does not diminish the correctness of this research, because our goal is to evaluate face detection systems rather than the learning methods. The following algorithms were tested in this work: Intel© OpenCV (OCV), Luxand© FaceSDK (FSDK), Face Detection Library (FDLib), SIFinder (SIF), University of Surrey (UniS), FaceOnIt(FoI), Neurotechnology© VeriLook (VL). Their brief description will be given in Section 2.

2 Algorithms' test set

2.1 Intel© Open Computer Vision library

In this work we used OpenCV 1.0, which contains the extended realization of the Viola-Jones object detection algorithm [14, 15] supporting Haar-like features.

Haar-like features, originally proposed by Papageorgiou et al. [12], valuate differences in average intensities between two rectangular regions, that makes them able to extract texture without depending on absolute intensities. However, Viola and Jones, during their work on objects detection algorithms [14], extended the set of the features and developed an efficient method for evaluating it, which is called an "integral image" [14]. Later Lienhart et al. [9] introduced an efficient scheme for calculating 45° rotated features and included it in OpenCV library.

It should be mentioned, that opposite to many of the existing algorithms using one single strong classifier, Viola-Jones algorithm uses a set of weak classifiers, constructed by thresholding of one Haar-like feature. Due to large number of weak classifiers, they can be ranked and organized into cascade.

In this work, we have tested cascade for the frontal face detection included by default in OpenCV 1.0: *haarcascade_frontalface_alt* (trained by R. Lienhart). To find trade off between FAR and FRR (see Section 4 for FAR and FRR definition), we have changed *min_neighbors* parameter, which indicates minimum number of overlapping detections are needed to decide a face is presented in the selected region; all other parameters were set by default.

2.2 SIF

This algorithm[8] has been developing in the Laboratory of Data Analysis of Tula State University. The main hypothesis consists in the eyes being dark spots in the face image, and we can immediately skip the routine scan of the image by sub-windows of different size.

At the beginning, the algorithm finds points of minimum brightness in image, then these points are sorted, some of them are discarded, and the rest are grouped in pairs. Then these fragments, containing a pair of singular points, are photometric normalized, affine transformed (for images containing only two singular points only following transformation can be applied: rotation, scale transformation (with the same scale on both axes) and displacement) and projected into the lattices of fixed size. After these transformations, the lattices are represented as a vector of features (values of brightness of nodes) and are sent to the two-class SVM-classifier, trained in advance on a large number of faces and non-faces.

As a parameter to find trade off between FAR and FRR, we changed the shift of hyperplane separating *face* and *non-face* classes in the space of features(values of brightness of the lattices nodes).

2.3 Face Detection Library

The Face Detection Library (FDLib) has been developed by Keinzle et al.[7]. Authors proposed a method for computing fast approximations to support vector decision functions (so-called reduced set method) in the field of object detection. This method creates sparse kernel expansions, that can evaluated via separable filters. This algorithm has only one tuning parameter that can control the "rigour" of face detection via changing the number of separable filters into which the reduced support vectors are decomposed.

2.4 UniS

Algorithm UniS was developed in University of Surrey and is based on various methods. To find the trade off between FAR and FRR we changed the value of the face confidence threshold for UniS.

2.5 FaceOnIt

FaceOnIt (`http://www.faceonit.ch`) is a face detection SDK developed at the Idiap research institute[13, 10]. It is based on the cascade architecture introduced by Viola-Jones and on an extension of Local Binary Patterns. *LBPs* have been proposed by Ojala et al.[11] for texture classification. But later its rotation invariance and computationally lightness were used Ahonen et al.[1] to develop effective and fast face recognition algorithm. As a parameter to find trade off between FAR and FRR, we changed the value of face confidence threshold.

2.6 FSDK and VL

FaceSDK (version 2.0) and VeriLook (version 4.0) were kindly provided by Luxand Inc. (`http://www.luxand.com`) and Neurotechnology (`http://www.neurotechnology.com`) respectively. These two algorithms are commercial products, and therefore no details of the principle of their functioning were disclosed. To find the trade off between FAR and FRR we changed the value of the face confidence threshold for VL and changed parameter of *FSDK_SetFaceDetectionThreshold* function affecting the threshold for FaceSDK.

3 Models of Faces Representations and Localization Accuracy

There are many different models of face representation in images: by the center of the face and its radius, by rectangle (OCV, FDLib, FoI), by coordinates of the centers of eyes (SIF, UniS, FSDK, VL), by ellipse, etc.

In this work we represent faces by coordinates of the centers of the eyes (i.e. centers of the pupils), because first, this representation looks to be more opportune in terms of the results comparison; second, usually face recognition algorithms require the centers of eyes matching for learning samples; third, experts mark eyes faster, easier and more precisely than they mark faces by rectangles. Thus, to unify the resulting comparison method we suggest eyes reconstruction model, which receives a face location in rectangle representation and returns estimated coordinates of the centers of eyes.

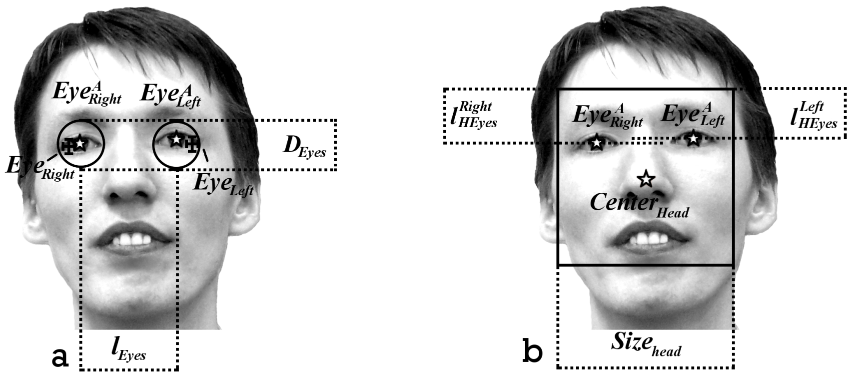

Fig. 1. Schematic face representation. Eye_{Left} and Eye_{Right} – absolute coordinates of detected left and right eye respectively; l_{Eyes} – distance between eyes' centers; l^{Left}_{HEyes}, l^{Right}_{HEyes}, l_{HEyes} – distance between top border of the face and center of the left or right eye, or the eyes respectively; $Size_{Head}$ – size of the rectangle representing face; D_{Eyes} – diameter of the area of acceptable eyes' coordinates deviation from the true eyes location Eye^A_{Right} and Eye^A_{Left}; $Center_{Head}$ – absolute coordinates of the found face.

3.1 Localization Accuracy for Algorithms Describing Faces by Centers of the Eyes

If detected faces are represented by the centers of the eyes (Fig. 1.a), let's consider them to be correctly detected, if and only if detected eyes belong the area around the true eyes location with the diameter D_{Eyes}. Which depends on the distance between eyes' centers and α, has been taken equal to 0.25 (This criterion was originally used by Jesorsky et al. [5]), and calculates as $D_{Eyes} = 2\alpha \times l_{Eyes}$.

3.2 Localization Accuracy for Algorithms Describing Faces by Rectangle

Assume there is a full face portrait image with no incline (Fig. 1.b), and the algorithm has found its center and size – ($Center_{Head}$ and $Size_{Head}$ respectively). Obviously, the eyes on this image are located symmetrically about the vertical axis (i.e., at the half the distance between them: $l_{Eyes}/2$) and at the same distance (l_{HEyes}) from the top border of the face's rectangle.

Thus the absolute coordinates of eyes can be estimated as:

$$\begin{aligned} Eye^y_{Right} = Eye^y_{Left} = Center^y_{Head} + l_{HEyes} - \tfrac{1}{2}Size_{Head} \, , \\ Eye^x_{Right} = Center^x_{Head} - \tfrac{1}{2}l_{Eyes} \, , \\ Eye^x_{Left} = Center^x_{Head} + \tfrac{1}{2}l_{Eyes} \, . \end{aligned} \tag{1}$$

Let's try to estimate the parameters of the algorithm, namely l_{Eyes} and l_{HEyes}, as an average of the huge amount of images with experts' labeled eyes. Based of such analysis, the following coefficients have been founded: A – average proportion of distance between top border of the face and center of the eyes (l_{HEyes}) to the size of the face rectangle; and B – average proportion of the distance between eyes (l_{Eyes}) to the size of the face rectangle ($Size_{Head}$). They can be estimated using information about true eyes location on the images series:

$$A = \frac{1}{N}\sum_{i=1}^{N}\frac{l^i_{HEyes}}{Size^i_{Head}}, \;\; B = \frac{1}{N}\sum_{i=1}^{N}\frac{l^i_{Eyes}}{Size^i_{Head}}, \tag{2}$$

where l^i_{HEyes}, l^i_{Eyes} and $Size^i_{Head}$ – respective parameters measured for the i-th image in the data set, containing N objects. Therefore the coordinates of the eyes for a given face size and the coefficient of proportions for the algorithms (2) are calculated according next equations:

$$\begin{aligned} Eye^y_{Right} = Eye^y_{Left} = Center^y_{Head} + Size_{Head}\left(A - \tfrac{1}{2}\right) \, , \\ Eye^x_{Right} = Center^x_{Head} - Size_{Head}\left(\tfrac{1}{2}B\right) \, , \\ Eye^x_{Left} = Center^x_{Head} + Size_{Head}\left(\tfrac{1}{2}B\right) \, . \end{aligned} \tag{3}$$

When the eyes' coordinates are estimated, we can determine its localization accuracy, as it is described in section 3.1.

If there is a full face portrait image with any incline, let's find l_{HEyes} as an average distance between center of each eye and top border of the face, i. e.:

$$l_{HEyes} = \tfrac{1}{2}\left(l^{Left}_{HEyes} + l^{Right}_{HEyes}\right).$$

It should be noted that such "conversion" of face representation could deteriorate the localization accuracy for algorithms describing faces by rectangles . However, all benefits from the "conversion" (discussed in the beginning of this section) are much bigger than risks mentioned above.

4 Parameters of Results Evaluation

The results of each algorithm have been evaluated by following parameters:

- *False Rejection Rate* (FRR) — Ratio of type I errors, which indicates the probability of misclassification of the images containing a face;
- *False Acceptance Rate* (FAR) — Ratio of type II error, which indicates the probability of misclassification of the images not containing a face;
- *Distance to the "exemplary" algorithm* (d_{exemp}). We consider a FD algorithm to be "exemplary" (exemp), if its FAR and FRR equals 0. Thus the distance between "exemp" algorithm and this one is $d_{exemp} = \sqrt{FAR^2 + FRR^2}$;
- *Speed parameters* such as mean and median of an image processing time (ms.) from the dataset. These results were obtained in following configuration: Intel Core2Duo 1.66 GHz, 2Gb RAM, Windows Vista HP.

Measure of dissimilarity of the algorithms' results is normalized Hamming distance between the pair of vectors of algorithms' errors:

$$d_H(X_i, X_j) = \frac{1}{N} \sum_{s=1}^{N} \left| x_i^{(s)} - x_j^{(s)} \right|.$$

Each vector component ($x^{(s)}$) equals 1, if the algorithm has correctly detected face region on the correspondent (s) image in the dataset. Further, we will call the matrix of such distances ($d_H \in [0, 1]$), matrix of dissimilarities or d-matrix.

5 Image dataset

All algorithms have been tested on the following manually marked datasets[1]:

1. *Face Place* contains 1247 images (480×400 pixels) of 150 people taken from different angles, `http://www.face-place.org/`;
2. *The IMM Face Database* 240 images (512×342 pixels) of 40 people, `http://www.imm.dtu.dk/~aam/`;
3. *B. Achermann's face collection, Uni of Bern* – 300 greylevel images (512×342 pixels) of 30 people., `ftp://ftp.iam.unibe.ch/pub/Images/FaceImages/`;
4. *BioID* Only 1520 grayscale images containing one face have been used, complex background;
5. *The Sheffield Face Database* (previously known as The UMIST Face Database). Only 416 images containing two eyes have been used in this test;
6. *PIE Database subset* (one picture of each individual), complex background;
7. *Indian Face Database* Only 513 images containing two eyes have been used;
8. *The ORL Database of Faces*;
9. *Laboratory of Data Analysis (Tula State University) Face Database* contains 6973 color images, complex background, 320×240 pixels (see examples at `http://lda.tsu.tula.ru/FD/TulaSU_FDB.zip`).

[1] If no URL is given, it can be found at `http://www.face-rec.org/databases/`

For correct FAR estimation we have used Non-Face images collected in *Tula State University*. This dataset contains 48211 color images, 320×240 pixels.

Total test dataset size is 59888 images: 11677 faces and 48211 non-faces. Coefficients A and B were determined on *Georgia Tech Face Database*.

6 Results

After the algorithms had been routinely tested, FRR, FAR, d_{exemp}, speed and vectors of algorithm's errors were obtained for each one. Coefficients of the model of face localization accuracy for algorithms representing faces by rectangles (see Section 3.2) are given in Table 1.

Table 1. Coefficients of the model of supposed coordinates of eyes estimation

Algorithm	FDLib	OCV	FoI
coefficient A	0.3332	0.3858	0.2646
coefficient B	0.3830	0.3666	0.5124

Changing the algorithm's parameter, plots the FAR against the FRR for all tested parameters, which yields the receiver operating characteristic (ROC) curve in standard biometric sense[3, 16]. These curves (Fig. 2) let us to determinate the optimal algorithm (with parameters) in a particular situation.

For more detailed analysis, let's fix the parameters, delivering the lowest d_{exemp} for each algorithms, find d-matrix (see Table 2), two dimensional FastMap [2] of d-matrix (Fig. 3) and other evaluation parameters (see Table 3).

Table 2. Matrix of dissimilarity of algorithms with fixed parameters (see value of parameters in the parentheses)

	OCV	SIF	FDLib	FSDK	UniS	FoI	VL	Exemp
OCV(2)	0	0.099	0.227	0.052	0.076	0.053	0.047	0.043
SIF(-3)		0	0.226	0.089	0.097	0.070	0.075	0.073
FDLib(1)			0	0.222	0.223	0.213	0.215	0.216
FSDK(5)				0	0.069	0.043	0.035	0.030
UniS(20)					0	0.051	0.053	0.047
FoI(5)						0	0.026	0.019
VL(2)							0	0.011
Exemp								0

Analysis of the elements of the d-matrix leads to the conclusion, that the most similar algorithms are FoI and VL, and the most different are FDLib and OCV. It should be mentioned, that d_H-es between almost all tested algorithms (excluding FDlib) are smaller than 0.1.

It is obvious that each of the algorithms have a unique peculiarity of detection. One way to perform their numerical evaluation is to compare the number

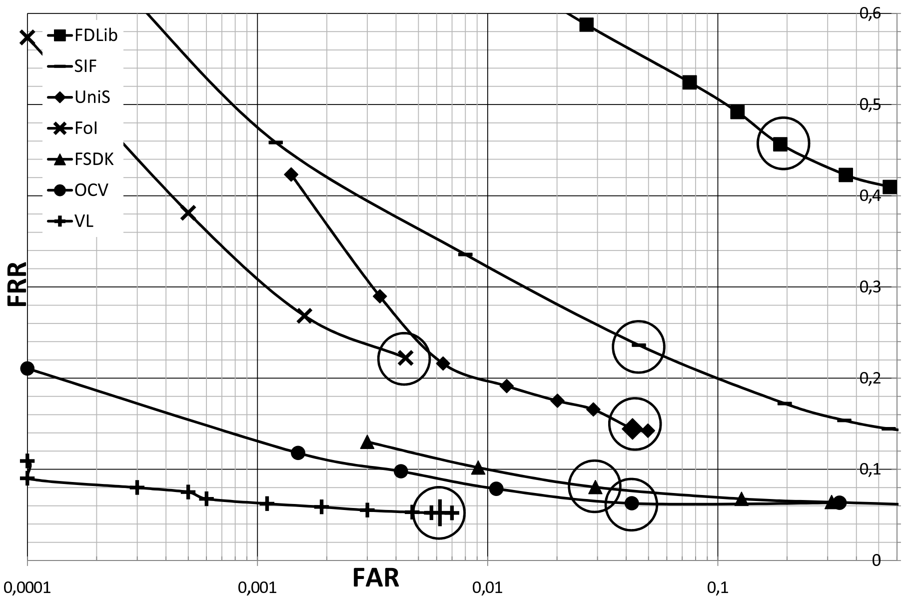

Fig. 2. The ROC plots. FAR (in log scale) against FRR. Perfect performance would be the bottom left corner: $FRR = FAR = 0$.

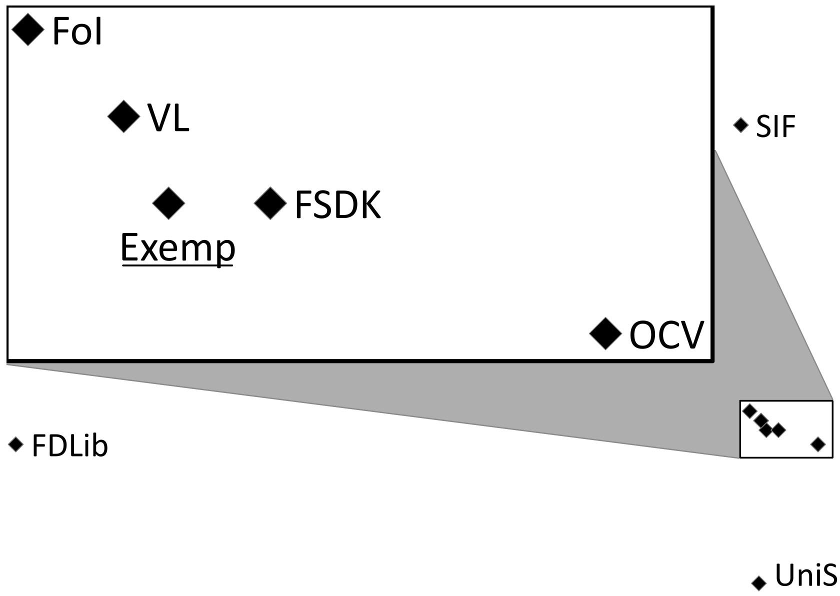

Fig. 3. Two dimensional FastMap diagram obtained on the data represented in Table 2 (the closest five algorithms are also represented on an expended scale in the frame).

Table 3. Results of algorithms' testing with fixed parameters (see value of parameters in the parentheses); Estimations of time (Mean and Median) are given in ms.

Algorithm	**FRR**	**FAR**	d_{exemp}	**Mean, ms.**	**Median, ms.**
OCV(2)	0.0628	0.0423	0.0757	90	88
SIF(-3)	0.2362	0.0454	0.2405	260	254
FDLib(1)	0.4565	0.1868	0.4932	64	62
FSDK(5)	0.0805	0.0294	0.0857	1305	1041
UniS(20)	0.1444	0.0426	0.1505	176	149
FoI(5)	0.2222	0.0044	0.2222	84	85
VL(2)	0.0523	0.0062	0.0527	47	43

of images uniquely classified by each algorithm, number of "challenging" and "easy" images (see Table 4). Here "easy" images mean images detected by all algorithms. In the opposite case, images are considered to be "challenging".

Table 4. Peculiar images distribution on the datasets (Dataset ID corresponds to the index in the image datasets' list (Section 5))

	Dataset ID									
	1	2	3	4	5	6	7	8	9	NonFaces
Number of images	1247	240	300	1520	416	68	513	400	6973	48211
Peculiar Cases										
"easy" images	315	31	121	713	28	44	4	248	2881	34093
"challenging" imag.	9	3		5	6		7		51	
only OCV	2				5		1		6	
only SIF									5	
only FDLib					1				2	
only FSDK		1			2		2		8	
only UniS	1	2					3		18	
only FoI	3						1		21	1
only VL	12			2	23		2		21	

7 Discussion and Conclusion

In this work the seven FD algorithms were tested and the statistical model of eyes' position estimation for algorithms describing faces by rectangle was proposed (see Section 3.2).

According the result of our study VeriLook has the best performance under various parameters and has the first place in the speed test (18-20 images per second). FDLib shows good speed characteristics (second place), but it demonstrates the worst performance. OpenCV – the most popular and free available FD algorithm – took the second place in the performance test and has sufficient speed. SIF developed in Tula State University has demonstrated the average performance. It's worth noting that VeriLook has the biggest number of uniquely classified images, i.e. images that were misclassified by other algorithms.

It should be noted that about 64% of images were correctly processed by all algorithms. Such images are called "easy" in this work.

Though the best algorithms demonstrated excellent performance in this test, there are as many as $5-6\%$ of FRR separating it from the desired error-free result. However, the per cent of the "challenging" images in the dataset is dramatically small – only 0.14% of our dataset. Thus there is a potential for FD algorithms refinement. Also we believe that this performance "gap" can be eliminated through the detectors combining. In the future we plan to offer an interactive web framework for FD algorithms testing. It should be emphasized, that this research has minor shortcoming: we partially used public image databases, and we have no reason to think that they were not used in a learning process of tested FD systems. We are also looking forward to collaboration with other developers of FD algorithms and with researchers who would like to share their image databases with us.

References

1. Ahonen, T.,Hadid, A. and Pietikainen, M.:Face recognition with local binary patterns, Lecture Notes in Computer Science, 469-481 (2004)
2. Faloutsos, C., Lin, K.I.: FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, 163–174 (1995)
3. Fawcett T.: An introduction to ROC analysis, Pattern Recognition Letters, Volume 27, Issue 8., 861–874 (2006)
4. Hjelmas, E., Low, B.K.: Face Detection: A Survey, Computer Vision and Image Understanding, Volume 83, Issue 3, 236-274 (2001)
5. Jesorsky, O., Kirchberg, K.J., Frischholz, R.W.: Robust face detection using the hausdorff distance, Lecture Notes in Computer Science, June, 90-95 (2001)
6. Jones, M.J.:Face Recognition: Where We Are and Where To Go From Here, IEEJ Trans. on Elect., Information and Systems, Volume 129, Issue 5, 770–777 (2009)
7. Kienzle, W., Bakir, G., Franz, M., Scholkopf, B.: Face detection – efficient and rank deficient, Advan. in neural inform. process. systems 17, 2005. – P. 673–680 (2005).
8. Krestinin, I.A., Seredin, O.S.: Excluding cascading classifier for face detection. Proc. of the 19th Int. Conf. on Computer Graphics and Vision, 380–381 (2009)
9. Lienhart, R. and Maydt, J.: An extended set of Haar-like features for rapid object detection. Proc. Intern. Conf. on Image Processing 2002, 900–903 (2002)
10. Marcel, S., Keomany, J., et al.: Robust-to-illumination face localisation using Active Shape Models and Local Binary Patterns, IDIAP-RR, V. 47, (2006)
11. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24, Issue 7, 971–987 (2002)
12. Papageorgiou, C. P., Oren, M., Poggio, T.: A general framework for object detection, Proc. of ICCV, 555–562 (1998)
13. Sauquet, T., Rodriguez, Y., Marcel, S.: Multiview face detection, IDIAP-RR (2005)
14. Viola, P. and Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple, Proc. IEEE CVPR 2001, (2001)
15. Viola, Paul and Jones, Michael J.: Robust Real-Time Face Detection, International Journal of Computer Vision, Volume 57, Issue 2, 137–154 (2004)
16. Wechsler H.: Reliable face recognition methods: system design, implementation and evaluation, Springer, 329 p.(2007)
17. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey, IEEE Transactions on Pattern analysis and Machine intel., Vol. 24, Issue 1, 34-58 (2002)