Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 7 (2015) 80-92

Half and Fully Automatic Character Identification in Movies Based on Face Detection

Gábor SZŰCS[1], Borbála MAROSVÁRI[2]

[1] Department of Telecommunications and Media Informatics,
Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics, Hungary, e-mail: szucs@tmit.bme.hu
[2] Department of Telecommunications and Media Informatics,
Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics, Hungary, e-mail: bmarosvari@yahoo.com

Manuscript received April 18, 2016

Abstract: In this paper we present our solution for character annotation in videos based on the faces of the actors. We have developed a fully automatic annotation system without any end user intervention and a half automatic one with possibility of end user's interaction. For this task we have used face detection, face recognition and we have improved them by face aligning. Another contribution is the developed clustering procedure for the extracted features of actors' faces. The key question regarding the identification of characters is whether visual information of the actors can be gathered beforehand or not. For the former case (half automatic annotation) we used Fisherfaces face recognition, as a supervised machine learning algorithm; for the latter we developed our fully automatic face identification method, and added a quick mapping and checking feature (which belongs to half automatic annotation as well) to increase the efficiency of the system by user inputs.

Keywords: annotation of film; clustering; face detection; face recognition; Fisherfaces

1. Introduction

Nowadays films are becoming increasingly complex with complicated story lines and with many characters. In the case of movies with numerous characters the individual characters are often challenging to follow, and because of this the movies can be harder to understand. Our aim was to develop an automated annotation process for following movie characters based on their faces. This system would store the appearances and disappearances of characters in the film in order to help with following fictional characters. This annotation is useful for

understanding, analyzing, archiving certain details of the film. The main scientific challenge in this problem was the face detection and recognition components to identify the different characters in the film.

An ideal face detection system should find all the faces independent of size, shape, and relative positions of the faces in the images. There are several face detection algorithms searching faces utilizing different methods. Real time face detection and tracking is already solved in normal indoor settings, but in outdoor environments (and under other special circumstances, which often occur in films) the algorithms are inaccurate [11].

Humans are able to detect, recognize, distinguish faces of different persons and recall them for the rest of their lives [14]. In computer vision finding and identifying faces are often met with difficulties. Because of different poses, different relative positions of the camera and the face, the images of the same person will look different. Depending on the perspective, the same face can be perceived as completely differently shaped objects. Some parts of the face may become completely obscured by hair, or other objects. Facial expressions strongly influence the look of the face, as well as glasses, facial hair and clothing, all of which can cause difficulties during detection. The circumstances under which the photos are taken such as lighting, focus and white balance affect the quality of the picture and the accuracy of face recognition.

2. Related works

2.1. Face detection algorithms

Face detection can be performed by observing different attributes. For finding faces skin color can be an indicator in colored pictures, in videos movement detection can be used, or generally we can look for face-like objects and structures, and of course the combinations of these. The most successful algorithms are appearance-based, without using additional cues.

Face detection algorithms can be classified into four categories, which do not have clear boundaries so they can overlap.

Knowledge based: These rule-based methods work with creating rules from human knowledge of what constitutes as a face. These rules describe the overall look of a face, details of facial features [22]. A face model can also be used, which describes the relative distances of facial features [18]. These methods are mainly used for face localization.

Feature invariant: These methods look for features that do not change when changes in illumination, position, perspective occur. Mainly used for face localization. These algorithms include: edge grouping [23], [10], texture analysis [6], skin color analysis [12], multiple feature analysis [7].

Template matching: Several patterns of a face are stored, describing the face as a whole or describing certain facial features. For face detection the correlation between the input image and stored patterns is computed. These methods can be used for face localization and detection alike. Methods include: predefined face templates [5], deformable face templates [9].

Appearance based, learning methods: These are the most successful algorithms. The models (or templates) are learned from a training set, which should capture the variability of facial appearances. Methods include: Eigenface [20], distribution based [19], Neural Network [16], Support Vector Machine [13], Hidden Markov-model [15], Bayes Classifier [17], AdaBoost learning based methods [21]. The latter are the most successful ones in terms of accuracy and speed [11].

2.2. Face recognition algorithms

For identifying faces feature extraction is essential. Depending on the face recognition system different features are required. The features needed to be found can be lines, or specific key points like the eyes, nose or mouth. Feature extraction can be performed during face detection, or after. In most cases face recognizers implement feature extraction. Feature extraction is also a key element in facial expression recognition.

Feature extraction is also necessary in systems that observe the faces in whole and create their own feature vectors. Algorithms like Eigenfaces [20] and Fisherfaces [3] need the exact positions of the eyes, nose or mouth in order to allow face normalization [24].

Face recognition algorithms can be divided into two groups, pose-dependent and pose-independent [11]. The difference between the two types is the representation of the face. Pose-dependent methods analyze faces from the viewer's point of view, while the pose-independent, object oriented methods work with 3D models of the face, thus becoming independent of the position of the face.

Pose-dependent algorithms can be further divided into two or three groups [24], [11].

Holistic algorithms: observing the faces as a whole. These include methods that use PCA like Eigenfaces and Fisherfaces. The local or analytic, feature based methods include geometric methods, HMMs, LBP histogram methods. The third class is the Hybrid methods: Elastic Bunch Graph Matching, Hybrid Local Feature Analysis, and modular Eigenfaces.

3. Improvement by face aligning

In order to align the pictures of the faces we need to determine the positions of the eyes. For finding the eyes we used two Viola-Jones detectors implemented in OpenCV [8], trained with "lefteye" and "righteye" cascade classifiers respectively [4]. We achieved the best results using these two detectors together from the OpenCV library.

These cascade classifiers were created by using 7000 positive samples for 18×12px size. The "lefteye" classifier is intended to find the left eye, as the "righteye" classifier is meant to find the right eye in the picture (as some examples can be seen in *Fig. 1*).

During testing, out of 591 cases the detectors found the eyes correctly only in one case, meaning the "lefteye" found the left eye and the "righteye" the right eye. In the majority of the test cases each detector found both of the eyes. In higher resolution pictures there were great amounts of false positives. Thus determining the true positives for each eye became difficult.

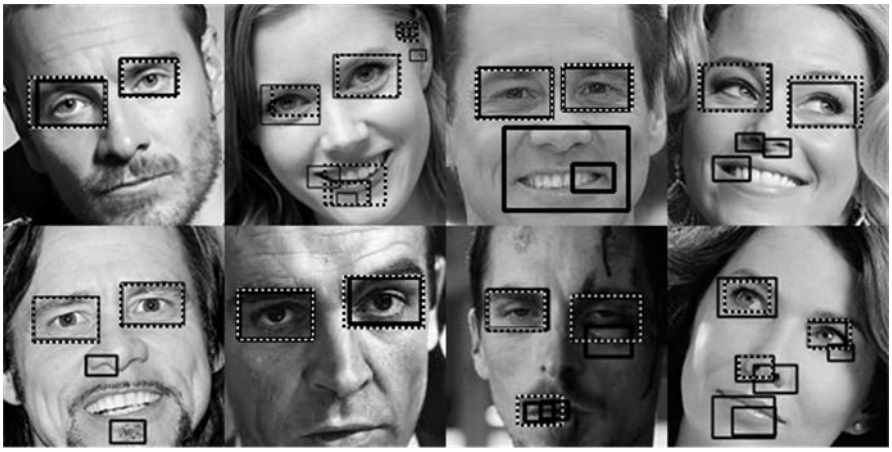

Figure 1: Results of "lefteye" and "righteye" classifiers

The eye-detectors are run in the face's upper 4/7 area; this immediately lowers the number of false positives. An eye was considered detected only when the two detectors found the same area: the center of one rectangle is within the other, and vice versa. The face is rotated when the following conditions are met: exactly 2 eyes are detected, the distance of the eyes is larger than the 1/9 of the face's width, and the rotation angle is less than ±35°. Using this method the detected eye-pairs were all true positives, out of the 591 test cases 437 correct rotations and eye-pair detections were achieved.

We tested our method on OpenCV's face recognition implementations. The test set consisted of 591 test photos of 20 persons, with one face on each photo. Beside the test set, the training set contained 10 photos for each person, aligned at eye level. The results can be seen in *Table 1*.

Table 1: Results of face recognition on 20 subjects with different preprocessing

	Correct identification	Incorrect identification	Accuracy
Eigenfaces	97	494	0.164
Fisherfaces	169	422	0.285
LBPH	129	462	0.218

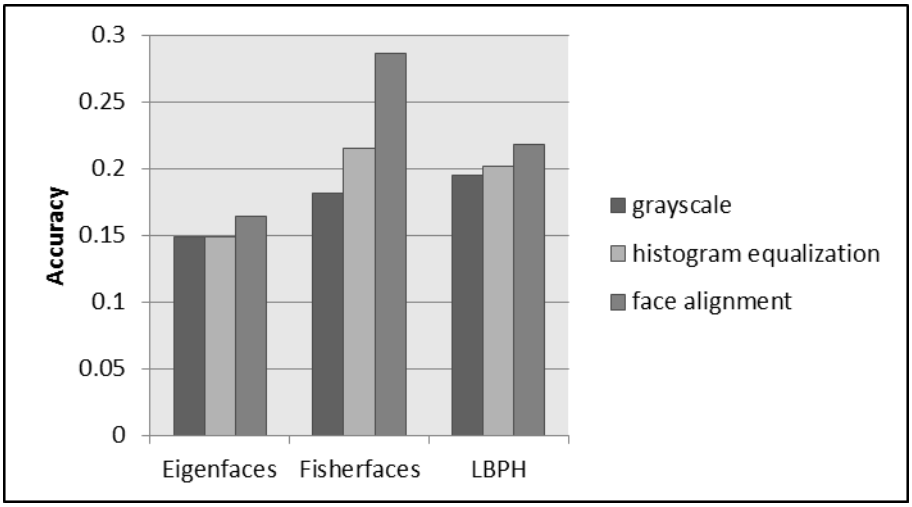

Figure 2: Results of face recognition on 20 subjects with different preprocessing

Fig. 2 shows the results of the different preprocessing methods, grayscale, histogram equalization (on grayscale images), and face alignment (on grayscale and histogram equalized images).

All three algorithms produced better results with face alignment than without. Correlation between the improvement or failure in recognition and the angle of rotation could not be observed.

4. Character identification

The key question regarding the identification of fictional characters in videos is whether there is visual information available on the characters or not. If this information can be used then the faces of actors should be gathered (e.g. from images or from the movie) and the training image set can be determined by

drawing the bounding boxes of the faces. The supervised character identification can be processed after training, and the time of appearances and disappearances of the characters can be determined in the film.

With unsupervised character identification there is no visual information about characters. In such case the identification can be achieved using information extracted only from the movie, labeling characters simply as "character A", "character B", etc. Differentiating between characters could be accomplished by using the first detected face as a training example, comparing subsequent faces to it and adding the faces to the training set to a new or existing character. But this method is practically unusable due to the low number of training images per person. Instead, in order to distinguish between different characters and to group the detected faces of same character together, our solution is based clustering. In an ideal situation the number of the clusters is equal to the number of the characters in the video. The time stamps of the clusters can be used for determining the duration of each character's presence (for annotation). For the unsupervised character identification problem we have developed a method – so called fully automatic character identification method – by clustering.

Our solution is divided into two parts; the first part extracts all the faces of the characters and stores them, the second part creates feature vectors from them, after which these vectors are clustered.

In the second part our method extracts LBP (Local Binary Patterns) feature vectors. In the basic methodology for LBP based face description (Ahonen et al., 2006) the part of the image where the face is located is divided into local regions and LBP descriptors are extracted from each region independently. The occurrences of the LBP codes in a region are collected into a histogram, which creates the local LBP descriptor. These local descriptors are then concatenated to form a global description of the face. These local feature based methods are more robust against variations in pose or illumination than holistic methods. This histogram effectively has a description of the face on three different levels of locality: the LBP labels for the histogram depict the patterns on pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face. The sample ratio parameter was 8 and the surrounding parameter of LBP was 2 in our application, which is implemented in OpenCV (Laganière, 2011).

A critical issue when clustering the LBP vectors in the next step was determining the number of clusters. The most often used clustering algorithm is the k-means, but as a parameter it requires the number of the clusters. Instead of this well-known algorithm we have used hierarchical k-means (Arai and Barakbah, 2007), which is the combination of k-means and hierarchical

clustering algorithm. The advantage of this hierarchical version of k-means is that it defines the initial centroids for standard k-means. In our solution the distance function of the clustering was the Euclidean distance (L2 norm) of LBP vector. For determining the number of the clusters we have used a rule of thumb, i.e. the root square of n/2, where n is the number of the vectors.

5. User supported and automatic annotation

Based on our fully automatic character identification method we have implemented an automatic annotation program. For the solution of the automatic annotation we have applied the method in *Fig. 3*, described in the previous section. The annotation program uses the generated clusters of the faces for the annotation of the characters. Based on the time stamps of a character's images the program collects and calculates the times of appearances and disappearances. These time durations are then stored in hh:mm:ss format, and other additional information (e.g. bounding box of the face) is also stored for possibilities of later analysis. The final annotation contains the screen time duration for each character.

With a half-automatic annotation program the user could provide external (additional) information and can improve the character identification task in the annotation, so we have developed half automatic character identification method with possibility of extra information. This extra information can be the character's name; the end user can input the names, and can map these to the list of durations. A further possibility in our half-automatic annotation program is quick mapping and checking. We have implemented an average image generating process for facial image clusters, in which the process converts the images into the same size and accumulates the pixel values in different pictures in the same positions, then divides it by the number of images in the cluster. The average image gives the possibility to quick mapping among the characters' names and the list of durations, because the end user does not need to examine numerous facial images, the average image is representative of all of them. Another advantage of using an average image is the possibility of correction. If the image cluster would contain more than one character, then the average image would not be recognized. In this case the end user can mark the unrecognized facial image, and the corresponding list of durations will be excluded from the annotation. The annotation can be improved by repairing these incorrect records in the list of durations. A further manual improvement possibility is merging a character's list of durations based on the average images.

6. Testing the fully and half automatic character identification

We have tested our character identification system on the movie „The Breakfast Club". Before the automatic annotation, we have created a manual annotation (list of durations for every character's presence, the characters' „screen time"), and these were compared to evaluate the system's performance. The length of the film is 1:37:05, the frame-rate is 23.976, the resolution is 1280×688 pixel, and we have extracted 11638 images by half second sampling.

The movie has 11 characters (14 in total, but the faces of only 11 can be seen). 7 characters (Allison, Andrew, Bender, Brian, Carl, Claire, Richard) can be seen in the whole length of the movie, 3 characters (Brian's mother, Brian's sister, Claire's father) are only present in the beginning, and 1 character (Andrew's father) can be seen in the beginning and the end of the movie.

6.1. Testing the fully automatic character identification

The *fully automatic character identification* method created 59 clusters from 7093 images, and for evaluation we have calculated the purity, Rand index and F_1 indicators of clustering. The resulted clusters are pure (have high purity score), but the images belonging to the same character are distributed into more clusters.

If we compared it with the case where we have given manually the number of clusters (11), then it can be seen (in *Table 2*) that F_1 is better, but the other two indicators are worse.

Table 2: Results of the unsupervised character identification

	purity	Rand index	F_1
59 clusters	0.588	0.791	0.183
11 clusters	0.414	0.702	0.240

By lowering the number of clusters to the correct number of clusters the purity decreased, but the F_1 improved. This is due to the increasing recall measure of the characters' images. In order to see the trends of these indicators we have constructed a simplified task, where the sampling frequency in the video was lower, thus we could investigate more clustering. In *Fig. 3* the trends of the purity, Rand index and F_1 indicators can be seen, where the number of clusters was the parameter of our evaluation.

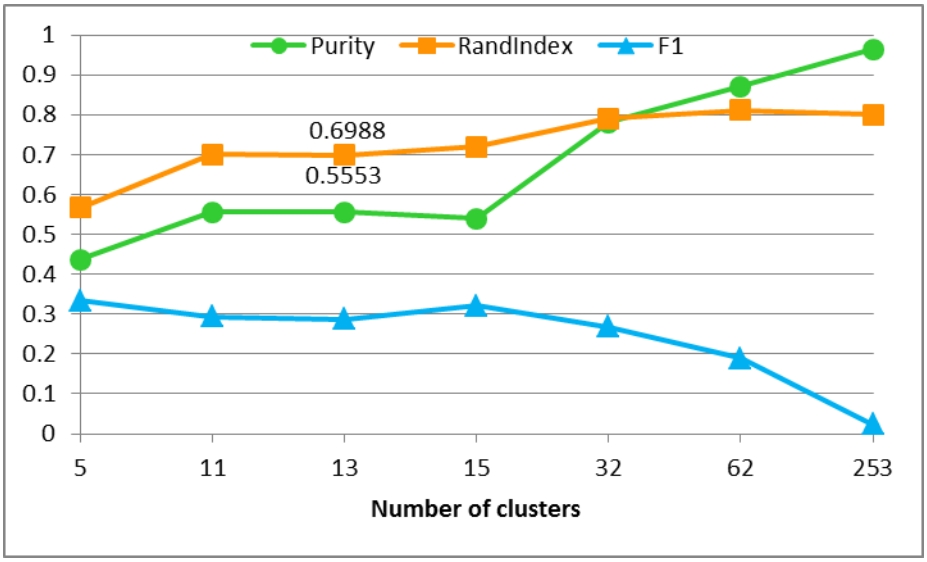

Figure 3: Comparison of clustering with different number of clusters

6.2. Half automatic character annotation – testing the supervised character identification

In order to test the supervised character identification method we created a learning set, in which there are 10 pictures for each character. These pictures were extracted from the movie to be tested, simulating the aforementioned method of drawing the bounding boxes of the characters and marking the positions of the eyes. These images were cropped and aligned using OpenCV's crop_face.py program to create the training set for the supervised character identification.

At the evaluation (as can be seen in *Table 3*) the manual annotation and the results of our character annotation system were compared. At the evaluation we have calculated the recall, the precision, and the harmonic mean of them (F_1), as the most frequent used indicators in the literature. Regarding a character the sum of the intersection of time of manual and automatic annotation system is the TP (true positive), the sum of only the manual ones is the P (i.e. this screen times consists of the sum of the true positives and false negatives: TP+FN), and the sum of intervals at only our system is the predicted P (i.e. TP+FP); F_1 can be determined from them.

Table 3: Results of the supervised character identification

Character	P	Recall	Precision	F_1
Allison	0:13:29	0.0816	0.1119	0.0944
Andrew	0:20:53	0.1979	0.7209	0.3106
Andrew's father	0:00:15	0.5333	0.0050	0.0099
Bender	0:22:51	0.2531	0.6686	0.3672
Brian	0:17:15	0.2676	0.4009	0.3210
Brian's mother	0:00:06	0.8333	0.0515	0.0971
Brian's sister	0:00:03	0	0	0
Carl	0:01:57	0.3419	0.5063	0.4082
Claire	0:13:10	0.3278	0.5396	0.4079
Claire's father	0:00:21	0.0476	0.0357	0.0408
Richard	0:07:04	0.4198	0.3090	0.3560

In order to compensate for the possible errors in the manual annotation, the detected faces were also individually inspected to calculate precision. Out of 7093 detections 6370 were correctly detected faces, the rest were the face detector's FP cases. Out of the 6370 faces 3085 were correctly identified by the Fisherfaces face recognizer, which means a precision score of 0.484.

Analyzing the *Table 3* and the collected faces we can assess our findings as follows. Brian's sister, the character with the shortest screen time was completely missed by the face detector, despite of her screen time being continuous, thus the sampling frequency being adequate, and also another character was detected on the same frame where Brian's sister was visible.

Brian's mother's and Andrew's father's recall values show that the program correctly identified their faces, the low precision scores show that there were many faces incorrectly identified as them. Claire's father's low recall value can be explained by that the face detector could only find his face 4 times during 21 seconds with 0.5 second sampling rate. Out of these 4 detected faces only 1 was correctly identified by the face recognizer, hence the low precision value. Allison's low scores can be explained by her appearance, her hair was usually obstructing her face, which made detecting and recognizing her face hard. The characters with the longest screen times, Andrew's and Brian's low recall values come from the error of the face detector, because their precision score is quite high, so when their faces were found, they were also identified correctly.

6.3. Half automatic character annotation with quick mapping and checking

Based on our *fully automatic character identification* method the half-automatic annotation program generates average images for facial image clusters for *quick mapping and checking*. An end user, who has seen the movie, is asked to map the average images to the characters. At the end of the mapping the images of the same characters were aggregated into one cluster, as shown in *Fig. 4* on the character called Bender. Only a few average images were mapped to nothing, because they were unrecognizable.

Figure 4: Average images of clusters mapped to a character (Bender)

After the user supported mapping the effectiveness of the system is expected to be better.

Table 4: Results of the half-automatic annotation after user supported mapping

Character	Number of mapped clusters	P	TP	TP+FP	Recall	Precision	F1
Allison	10	0:13:29	0:05:15	0:07:12	0.3894	0.7292	0.5077
Andrew	13	0:20:53	0:05:19	0:07:22	0.2546	0.7217	0.3764
Andrew's father	0	0:00:15	0:00:00	0:00:00	0	0	0
Bender	10	0:22:51	0:07:12	0:09:04	0.3151	0.7941	0.4512
Brian	5	0:17:15	0:02:13	0:03:18	0.1285	0.6717	0.2157
Brian's mother	0	0:00:06	0:00:00	0:00:00	0	0	0
Brian's sister	0	0:00:03	0:00:00	0:00:00	0	0	0
Carl	1	0:01:57	0:00:17	0:00:20	0.1453	0.8500	0.2482
Claire	10	0:13:10	0:04:47	0:07:51	0.3633	0.6093	0.4552
Claire's father	0	0:00:21	0:00:00	0:00:00	0	0	0
Richard	5	0:07:04	0:02:13	0:02:59	0.3137	0.7430	0.4411
Summarized	**54**	**1:37:24**	**0:27:16**	**0:38:06**	**0.2799**	**0.7157**	**0.4025**

After the user supported mapping the clusters of unrecognized average images are filtered out, as it can be seen in Table 4; from 59 clusters only 54 clusters remained. The purity values of the 5 omitted clusters are low: 0.28, 0.42, 0.19, 0.3, 0.35. It can be concluded that during filtering clusters the false decisions (images not containing faces) are also filtered out, thus the system is more accurate (F_1 reaches higher value).

7. Summary

In books, as well as in films it is important to evoke reader sympathy in order to draw your audience into the story. But if the audience cannot follow the many characters in the movie, then this sympathy will not develop in the audience. Our work on character detection and identification (for annotation) in films is a large step in understanding and analyzing the story told.

In this paper we have presented a character annotation solution for videos. We have developed a fully automatic annotation system without any end user intervention and a half automatic one with possibility of end user's interaction. For this task we have used face detection, face recognition and we have improved them by face aligning. Another contribution is the developed clustering procedure for the extracted features of actors' faces.

For the half automatic annotation we have used Fisherfaces face recognition, as a supervised machine learning algorithm. For fully automatic annotation the possible method is clustering, and we have developed a *fully automatic face identification* method using it. In order to increase the efficiency of the system by user inputs a new feature, so called *quick mapping and checking* feature was added to the system.

References

[1] Ahonen, T., Hadid, A., Pietikäinen, M., "Face Description with Local Binary Patterns: Application to Face Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(12), 2006, pp. 2037–2041.

[2] Arai, K., Barakbah, A. R., "Hierarchical K-means: an algorithm for centroids initialization for K-means", *Reports of the Faculty of Science and Engineering*, 36(1), 2007, pp. 25–31.

[3] Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J., "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 1997, pp. 711–720.

[4] Castrillón-Santana, M., Déniz-Suárez, O., Antón-Canalís, L., Lorenzo-Navarro, J., "Face and Facial Feature Detection Evaluation", *Third International Conference on Computer Vision Theory and Applications*, VISAPP08, 2008.

[5] Craw, I., Tock, D., Bennett, A., "Finding Face Features", *Proc. Second European Conf. Computer Vision*, 1992, pp. 92–96.

[6] Dai, Y., Nakano, Y., "Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scene", *Pattern Recognition*, Vol. 29, No. 6, 1996, pp. 1007–1017.
[7] Kjeldsen, R., Kender, J., "Finding Skin in Color Images", Proc. *Second Int'l Conf. Automatic Face and Gesture Recognition*, 1996, pp. 312–317.
[8] Laganière, R., "OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision", Packt Publishing Ltd., 2011.
[9] Lanitis, A., Taylor, C. J., Cootes, T. F., "An Automatic Face Identification System Using Flexible Appearance Models", *Image and Vision Computing*, vol. 13, no. 5, 1995, pp. 393–401.
[10] Leung, T. K., Burl, M. C., Perona, P., "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching", *Proc. Fifth IEEE Int'l Conf. Computer Vision*, 1995, pp. 637–644.
[11] Li, S. Z., Jain, A. K., "Handbook of Face Recognition", New York: Springer, 2005.
[12] McKenna, S., Gong, S., Raja, Y., "Modelling Facial Colour and Identity with Gaussian Mixtures", *Pattern Recognition*, Vol. 31, No. 12, 1998, pp. 1883–1892.
[13] Osuna, E., Freund, R., Girosi, F., "Training Support Vector Machines: An Application to Face Detection", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 130–136.
[14] Parr, L. A., "The evolution of face processing in primates", *Philosophical Transactions of the Royal Society B: Biological Sciences*., Vol. 366 No. 1571, 2011, pp. 1764–1777.
[15] Rajagopalan, A., Kumar, K., Karlekar, J., Manivasakan, R., Patil, M., Desai, U., Poonacha, P., and Chaudhuri, S., "Finding Faces in Photographs", Proc. *Sixth IEEE Int'l Conf. Computer Vision*, 1998, pp. 640–645.
[16] Rowley, H., Baluja, S., Kanade, T., "Neural Network-Based Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, 1998, pp. 23–38.
[17] Schneiderman, H., Kanade, T., "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 45–51.
[18] Shinn-Ying, H., Hui-Ling, H., "Facial Modeling from an Uncalibrated Face Image Using a Coarse-to-Fine Genetic Algorithm", *Pattern Recognition*, Vol. 34, No. 9, 2001, pp. 1015–1031.
[19] Sung, K.-K., Poggio, T., "Example-Based Learning for View-Based Human Face Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, 1998, pp. 39–51.
[20] Turk, M., Pentland, A., "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, 1991, pp. 71–86.
[21] Viola, P. and Jones, M., "Rapid object detection using a boosted cascade of simple features", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, pp. 12–14.
[22] Yang, G., Huang, T. S., "Human Face Detection in Complex Background", *Pattern Recognition*, Vol. 27, No. 1, 1994, pp. 53–63.
[23] Yow, K. C., Cipolla R., "Feature-Based Human Face Detection", *Image and Vision Computing*, Vol. 15, No. 9, 1997, pp. 713–735.
[24] Zhao, W., Chellappa, R., Phillips, P. J., Rosenfeld, A., "Face Recognition: A literature Survey", ACM Computing Surveys, Vol. 35. No. 4, 2003.