Real time Face Recognition using Local Ternary Patterns with Collaborative Representation based Classification for Mobile Robots

Duc My Vo and Andreas Zell

Chair of Cognitive Systems, Computer Science Department, University of Tübingen, Sand 1, D-72076 Tübingen, Germany
{duc-my.vo,andreas.zell}@uni-tuebingen.de
http://www.cogsys.cs.uni-tuebingen.de

Abstract. The ability to recognize faces is a crucial element for human-robot interaction. In this paper, we present an algorithm for mobile robots to detect, track and recognize human faces accurately, even when humans go through different illumination conditions. We track faces using a tracker that combines the algorithm of an adaptive correlation filter with a Viola-Jones object detection. This tracker adapts to scale changes and to rotation of the face, and its occlusion. It also adapts to complex changes of background and illumination. Recognizing the tracked face is established by using an algorithm that combines local ternary patterns and collaborative representation based classification. This combination enhances the efficiency of face recognition under different illumination and noisy conditions. Our method achieves high recognition rates on challenging face databases and can run in real time on mobile robots.

Keywords: local ternary patterns, collaborative representation, face recognition, face tracking, mobile robot.

1 Introduction

Face recognition for mobile robots still remains a challenging task. First, the face image is often taken under different conditions of illumination. Most existing methods are accurate for recognizing faces in constrained illumination conditions, but their performance is much worse in recognizing faces under uncontrolled illumination conditions. Second, while both the humans and the robot move in front of complex backgrounds, the face changes with wide variations of poses and scales. Therefore, it is not easy for the mobile robot to track the face correctly. Third, the captured images may include a great amount of noise that significantly degrades face recognition performance. Noise may result from environmental conditions and illumination. Fourth, many approaches to face recognition are time-consuming to be able to run on a mobile robot.

Recently collaborative representation based classification (CRC) [1] has been presented, which is not only as accurate as other state-of-the-art algorithms such as sparse representation based classification (SRC) [2], but it is also less time-consuming. This algorithm achieves a high accuracy when tested on challenging datasets, but it can be degraded when the cropped face image is misaligned or the mobile robot captures the face image under varying illumination. Fortunately, we found an efficient descriptor, local ternary patterns [3], which is able to significantly reduce the influence of uncontrolled illumination, in shady as well as in bright areas. In addition, it is not only insensitive to random noise in face images but also relatively robust to misalignment. In this paper, we propose an algorithm for recognizing faces based on the combination of local ternary patterns and collaborative representation based classification. This combination enhances the efficiency of collaborative representation based classification in face recognition, which can help mobile robots to recognize human faces even when humans move freely under different illumination and noisy conditions. Furthermore it significantly reduces computational costs to help the robot run in real time. In our method, face tracking is an early and critical step that finds the face in images, from which we can extract relevant features to improve the accuracy of face recognition. The more precisely the face can be tracked, the more accurately it can be cropped and recognized. Therefore we use the method of face tracking mentioned in our previous research [4], to adapt the changes of the face well as to adapt to complicated changes of illumination.

The remaining parts of this paper are organized as follows. In Section II we present state-of-the-art algorithms of face recognition which motivated our research. In Section III, our method is presented in detail. In Section IV the experimental results obtained from databases are presented. We conclude this paper, mentioning our intentions with regard to our future work in Section V.

2 Related work

Although there are many approaches to face recognition for human-robot interaction [5], these methods are only used to recognize faces in constrained environments. Cruz [6] has developed an approach to help service robots to be able to recognize faces in unconstrained environments. This method, in which SIFT descriptors are applied for extracting facial features, is robust to local affine distortions of face images. Thus it achieved significant progress in face recognition under different viewpoints and some environmental conditions. However, its performance is worse under uncontrolled illumination conditions. Furthermore, this method is too time-consuming to run in real time.

Recently, sparse representation, which was developed from the theory of sparse coding, has been applied in face recognition. In this theory, a face is represented as a combination of the training faces on the overall training dataset. Then this face is classified based on the least representation residual. Recent research has developed new algorithms of face recognition motivated from sparse representation and they have achieved significant progress [2]. Although sparse representation has shown high accuracy of face recognition, its computational cost is very expensive. For the field of face recognition for human-robot interaction, the computational complexity of this algorithm has not met the requirement of real time performance. Therefore, the collaborative representation was proposed by [1], using non-sparse l_2-regularization instead of l_1-norm sparse regularization. This improvement makes a significant difference between collaborative representation and sparse representation. Collaborative representation is nearly as accurate as sparse representation while it is much less time-consuming. Collaborative representation achieved a high accuracy when tested on some challenging datasets, but it is less successful when the cropped face is misaligned or the mobile robot runs under uncontrolled illumination conditions.

Other promising approaches for face recognition are descriptor based algorithms such as local binary patterns [7], which use both shape and texture information to represent the face image. The key advantages of local binary patterns are that they are invariant to gray-scale changes and their computational cost is very low. Thus they achieved a considerable success in uncontrolled face recognition [9]. However, in practice the efficiency of local binary patterns deteriorates significantly due to random noise in the areas surrounding the face. Tan *et al.* [10] presented local ternary patterns which not only inherit the advantages of local binary patterns but also significantly reduce noise sensitivity. The method of local ternary patterns is tested on challenging databases chosen to compare the algorithms of face recognition under complicated illumination conditions. All the tests demonstrated that local ternary patterns outperform local binary patterns in dealing with difficult illumination conditions.

3 Approach

In this section, we present in detail our approach to real time face recognition for human-robot interaction. Figure 1 shows the simple flowchart representing our approach to face recognition on mobile robots. It consists of seven steps: First, we apply a method of face detection described in our previous work [11], in order to quickly locate the position of the face in the initial step of the face tracker. In the second step, the face is tracked based on a method using the MOSSE filter [12] and a Viola-Jones face detection [13]. This method, which helps mobile robots to track the human face efficiently, has been presented in detail in our previous paper [4]. It is also used for the next steps when the tracked face is frontal. Thirdly, after successfully tracking the face we detect and track two eyes. After that, in the fourth step, the face image is aligned and cropped based on the two tracked eyes. In the fifth step, we apply the local ternary patterns operator to encode the textures of the facial regions. In the sixth step, the face image is divided into cells from which local ternary patterns histograms are extracted and concatenated into an advanced histogram. The technique of principal component analysis (PCA) is applied to transform this advanced histogram from a high-dimensional space to a low-dimensional space. The output of PCA is the facial feature vector. In the last step, the recognition is attained by using collaborative representation based classification, of which the input is the facial feature vector.

3.1 Face detection

As mentioned in our previous work [11] a method of face detection is used to find quickly the position of the face in the initial step of the face tracker. The information of geometric constraints, navigation and the technique of depth-based skin color segmentation are provided to make our face detector much faster and more accurate. Our

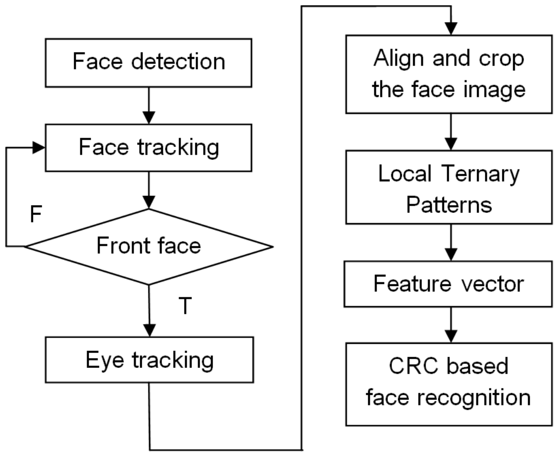

Fig. 1. Flow chart of our approach.

face detection involves three basic steps: First, in order to reduce computational costs we use a set of sampling points spanning the whole image to collect the information of color, texture and depth. Second, the constraints of geometry and navigation information are used to remove the background. Finally, the techniques of skin detection and depth-based skin colour segmentation are applied around filtered sampling points to find the potential regions in which the face detector is able to localize the face position. In addition, we can speed up face detection by limiting the range of facial scales, as mentioned in [11].

3.2 Face tracking

After detection, the face is tracked by using the method of face tracking presented in [4]. The MOSSE filter plays the role of an adaptive tracker which models the face appearance by training on-line the face samples from previous frames to adapt to the changes of its poses as well as the sudden changes of illumination. The face is tracked by correlating this filter over a search window. The correlation output indicates the relative position of the face in the current frame with respect to the previous one, which is the area corresponding to the maximum value in the correlation output. Thus we can efficiently find the next position of the face in image coordinates. The correlation output is computed as follows:

$$M = N \odot F^* \tag{1}$$

where M, N and F are the 2 D Fourier transforms of the correlation output M, training image N and the filter F, respectively. After moving the search window to the new face position, the MOSSE filter is updated on-line with the current search image. In frame i, it takes the form:

$$F_i^* = \frac{U_i}{V_i} \tag{2}$$

$$U_i = \gamma M_i \odot N_i^* + (1-\gamma)U_{i-1} \tag{3}$$

$$V_i = \gamma N_i \odot N_i^* + (1-\gamma)V_{i-1} \tag{4}$$

where γ is the learning rate.

The face detector in this paper plays an important role to correct the tracking position and eliminate drift. In addition the face size is estimated by using depth information while the face is being tracked.

(a) (b) (c) (d)

Fig. 2. Examples of face tracking through occlusion and drift. We compare our face tracker, which is marked by the red rectangle, and the original MOSSE filter, which is marked by the black rectangle. 2(a), 2(b), 2(c): The drift problem occurs when the human turns around or rotates. 2(d): The face is occluded.

3.3 Eye tracking

While detecting the face, only the frontal faces are selected for recognition. Since the size of the face is estimated by using depth information, we can apply a bilinear interpolation algorithm to scale the face into an 120×120 image. Because the size of two eyes are easily estimated they are detected and tracked based on the algorithm mentioned in [4]. Figure 3 shows the result of eye tracking.

Fig. 3. Example of eye tracking and face cropping. White circles indicate locations of the eyes. The white rectangle indicates the location of the cropped face.

3.4 Face alignment and cropping

The roll angle of the human face, θ, is calculated simply as follows

$$\theta = tan^{-1}\left[\frac{y_2 - y_1}{x_2 - x_1}\right] \tag{5}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the left eye and the right eye, respectively. In order to align the face, the face image is rotated at an angle of θ degrees. After that we crop the face by using a window with a fixed size of 76×84. The center point of the window is the tracking point of the face. Figure 3 also shows the cropped face which is indicated by the white rectangle.

3.5 Local Ternary Patterns

Local ternary pattern [10] is an advanced version of local binary pattern [7], which is used to summarize local gray-level structure. The local ternary patterns operator works in a 3×3 pixel block of a face image in which the difference between the center pixel and the neighboring pixel is encoded into a trinary code. We denote l_c as the gray level of the center pixel, and l_p as the gray level of the neighbors in which $p = 0, 1, ..., 7$. Thus the LTP code is computed as follows:

$$LTP = \sum_{p=0}^{7} f(l_p, l_c, th)3^p \tag{6}$$

Here $f(l_p, l_c, th)$ is the threshold function

$$f(l_p, l_c, th) = \begin{cases} 1, \ l_p \geq l_c + th \\ 0, \ |l_p - l_c| < th \\ -1, \ l_p \leq l_c - th \end{cases} \tag{7}$$

where th is a threshold. In our paper, th is equal to 5. In order to reduce the dimensionality, the LTP code is split into positive and negative LBP codes as follows:

$$f_p(l_p, l_c, th) = \begin{cases} 1, \ l_p \geq l_c + th \\ 0, \ otherwise \end{cases} \tag{8}$$

$$f_n(l_p, l_c, th) = \begin{cases} 1, \ l_p \leq l_c - th \\ 0, \ otherwise \end{cases} \tag{9}$$

As a result, we applied the local ternary patterns operator to generate two positive and negative LBP images for extracting the face features.

3.6 Facial feature extraction

In order to keep the local information and spatial locations of the face, we proposed the extraction of LTP features from the face image by dividing the face image into cells. In our research every cell is fixed at the size of 8×8 pixels. Figure 4 illustrates all the basic steps of our algorithm. In every cell, we extract an histogram of LTP codes. All these histograms are concatenated into an advanced feature histogram. Since the advanced histogram consists of a large number of bins, the technique of principal component analysis (PCA) is applied to the advanced histogram to reduce its dimensionality. The output of PCA is a 30-dimensional vector called the facial feature vector.

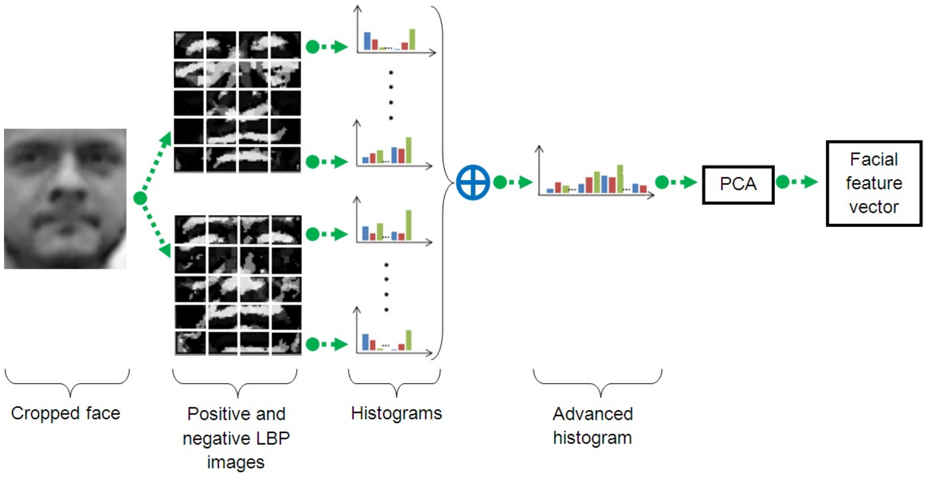

Fig. 4. Diagram of facial feature extraction

3.7 Collaborative representation based classification

We apply the collaborative representation based classification (CRC) with the regularized least squares for face recognition which codes a facial feature vector as a linear combination of the training vectors on the whole dataset instead of each subset. We denote the subset of the i^{th} class as T_i of which the number of columns is same as the number of training images. Furthermore, we denote the set of K classes of identities as $T = [T_1, T_2, T_3, ..., T_K]$ and every facial feature vector $v \in R^m$ is coded over T by using the regularized least square method as follows:

$$(\hat{\gamma}) = \arg\min_{\gamma} \left\{ \|v - T \cdot \gamma\|_2^2 + \lambda \|\gamma\|_2^2 \right\} \tag{10}$$

where λ is the regularization parameter. The solution of the collaborative representation based classification with regularized least squares is analytically derived as follows:

$$\hat{\gamma} = \left(T^T T + \lambda \cdot I\right)^{-1} T^T v \tag{11}$$

In addition we compute the regularized residuals of classes as follows:

$$r_i = \|v - T_i \cdot \hat{\gamma}_i\|_2 \, / \, \|\hat{\gamma}_i\|_2 \tag{12}$$

where $\hat{\gamma_i}$ is the coefficient vector of class i. By finding the minimal regularized reconstruction error, the identity of v is computed as follows:

$$\text{Identity}(v) = \text{arg min}_i \{r_i\} \tag{13}$$

4 Experimental Setup

In our experiments, the algorithm of local ternary patterns with collaborative representation based classification is denoted as LTP-CRC. For evaluating the performance of our face recognition algorithm, we used three challenging databases. First, we used the AR database [14] to test the accuracy and the processing time of our face recognition method and its competing methods in controlled environments. For evaluating the accuracy of face recognition in uncontrolled environments, we used the LFW-a database mentioned in [8]. In this database all the training and testing faces were aligned by using the algorithm of face alignment in [8]. Thus we did not need to apply the technique of face alignment in the preprocessing step. Finally, the Tuebingen database for face recognition was built by us to test the algorithms of face recognition on mobile robots in our own office environment.

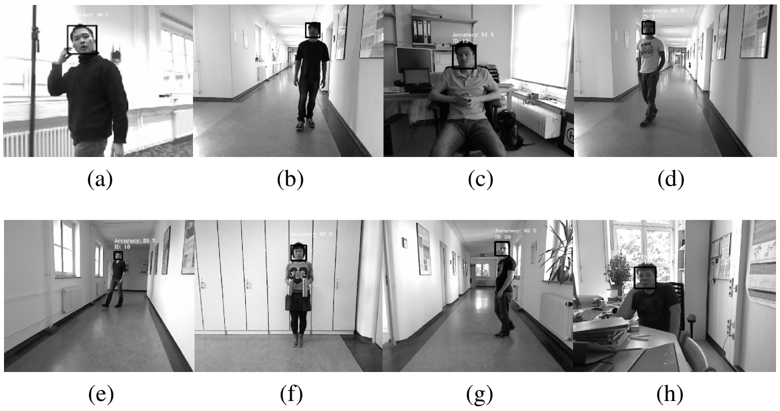

Fig. 5. Sample images from the Tuebingen dataset. The human face moves through different conditions of illumination and noise.

The experiments carried out on both the AR database and the LFW-a database used Matlab on a PC with 2.5 GHz Intel Core i5 CPU. Additionally, the experiment on the Tuebingen database were implemented using C++ on a PC with 2.5 GHz Intel Core i5 CPU. For the methods of LTP-CRC and CRC we set the parameter λ as 0.001 in the experiments. Figure 5 shows a sample of some images extracted from our dataset.

4.1 AR database

We used the AR database, which consists of 50 males and 50 females, to perform experiments under controlled environments. For each subject we used seven images for training, which are different in illumination and expression, and used the other seven images for testing.

We did two experiments with this database. In the first experiment, the images were cropped and resized to 60×43 pixels. And in the second experiment, the images were resized to 32×32 pixels to show how the algorithms work in the database of low-resolution images. We compared our algorithm LTP-CRC with its competing algorithms which are collaborative representation based classification (CRC), sparse representation based classification (SRC), support vector machine (SVM) and nearest neighbor (NN). In our algorithm and CRC, PCA is applied to reduce the dimension of the images to 300.

Table 1 shows the recognition rates of LTP-CRC, CRC, SRC and NN on the AR database in the first experiment. It shows that the result of LTP-CRC outperforms CRC, which is 5.2 % more accurate, and is significantly better than the other methods. It proves that local ternary patterns give a high contribution to the performance of face recognition due to its robustness with regard to different forms of illumination.

Table 1. Recognition rate and processing time on the AR database in the first experiment

	Recognition rate	Time
NN	0.713	**0.0013 s**
SRC	0.933	1.7878 s
CRC	0.937	0.0024
LTP-CRC	**0.989**	0.0026 s

Table 1 also shows the comparison of the processing time between our algorithm with the state-of-the-art ones including CRC and SRC. We can see that the recognition rate of LTP-CRC is higher than CRC and SRC, while its processing time is just 1.1 times slower than CRC, but is 688 times faster than the SRC. This implies that LTP-CRC is more advantageous in real time application of face recognition.

Table 2. Recognition rate on the AR database in the second experiment

NN	SRC	CRC	LTP-CRC
69.3 %	90.1%	88.4 %	**98.3 %**

In the second experiment we compared the accuracy of our algorithm with the others in lower-resolution images. Their results are shown in the Table 2 which proves that our algorithm is the best of the algorithms mentioned in this experiment. We can also see that the accuracy of our algorithm is only slightly reduced in comparison with the previous experiment, while all others are significantly degraded.

4.2 LFW-a database

The LFW-a database is the second one we used to compare our algorithm with some challenging methods including CRC, SRC, and NN. This database is collected for research of unconstrained face recognition. It consists of 158 different individuals of different races, ages and genders. For each of these individuals, we collected 5 training and 2 testing images. All the faces in these images were cropped to 32×32 pixels, and those from the same individual differed in pose, expression and illumination. In this experiment we used PCA for our algorithm and CRC to reduce the dimension of the images to 300.

Table 3. Recognition accuracy on the LFW-a database

NN	SRC	CRC	LTP-CRC
13.6 %	46.4 %	43.7 %	**59.5 %**

The accuracy of face recognition on the LFW-a database is shown in Table 3. Since the LFW-a database is a very challenging one the accuracy of all algorithms in this database is less than those of the AR database. However, our algorithm achieves the best accuracy in comparison to other algorithms mentioned above.

4.3 Tuebingen dataset

The Tuebingen dataset consists of 22 log files of 22 people recorded from a Microsoft Kinect camera mounted on a mobile robot SCITOS G5, as shown in Figure 6. Each of these log files recorded color and depth images at 30 frames per second at a resolution of 640×480 pixels. For each subject we used only five images for training which were cropped to 76×84 pixels. Our goal was to evaluate the performance of face recognition in indoor environments in which both the humans and the robot move under different illumination conditions and the faces change a variety of poses. We compared the accuracy and the processing time of our method with its competitors which are collaborative representation based classification (CRC), sparse representation based classification (SRC) and nearest neighbor (NN). The processing times of these algorithms were measured, including face detection, face tracking, eye tracking, face alignment and cropping in addition to face recognition. By using PCA, the dimension of images was reduced to 30. In addition, on this dataset we also evaluated the performance of face recognition with and without face alignment in order to demonstrate the effectiveness of our alignment technique.

In Table 4 we evaluated the recognition rate and the processing time of our method, LTP-CRC, with SRC and NN. It shows that LTP-CRC outperforms CRC by 12 % and NN by 33 %. Although the computational cost of LTP-CRC is slightly higher than the other methods, it is nevertheless fast enough to run in real time on mobile robots. We can see that LTP-CRC is a fast and reliable face recognition algorithm for mobile robots running in realistic environments. This is due to the fact that CRC is a relatively accurate and fast method and LTP is a powerful feature descriptor which is insensitive to noise and is resistant to lighting effects.

Table 5 shows the role of face alignment in the performance of face recognition. Our technique of face alignment significantly improved the recognition rate of LTP-CRC, CRC and NN due to its robustness to changes in face pose as well as in varying degrees of illumination. It contributes 2.73 %, 3.71 % and 3.83 % to the recognition rate of LTP-CRC, CRC and NN, respectively.

Table 4. Recognition rate and processing time on the Tuebingen database

	Recognition rate	Time
NN	67.55 %	**9 ms**
CRC	80.46 %	10 ms
LTP-CRC	**89.75 %**	12 ms

Table 5. Face recognition results with and without alignment on the Tuebingen database

	Unaligned	Aligned
NN	63.72 %	67.55%
CRC	76.75 %	80.46%
LTP-CRC	87.02 %	89.75%

5 CONCLUSION AND FUTURE WORK

In order to improve the performance of face recognition we proposed local ternary patterns with collaborative representation based classification. Our experimental results show that this algorithm achieved high recognition rates, and it is suitable for face recognition on mobile robots under uncontrolled illumination conditions. The proposed face recognition system requires on average 12 ms per frame on on a PC with a 2.5

Fig. 6. Our mobile robot SCITOS G5.

GHz Intel Core i5 CPU. Thus, it is able to run at video frame rate on mobile robots. For future development, we intend to develop this algorithm for recognizing faces across poses using the technique of face pose estimation mentioned in [4]. Recognizing the face in arbitrary poses will be more difficult in uncontrolled environments under varying illumination. Nevertheless our approach is expected to be able to recognize the face with large variations of face appearance.

References

1. L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 471–478.
2. J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE, Special Issue on Applications of Compressive Sensing and Sparse Representation*, vol. 98, no. 6, pp. 1031–1044, 2010.
3. X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.
4. M. Vo-Duc and A. Zell, "Real time face tracking and pose estimation using an adaptive correlation filter for human-robot interaction," in *European Conference on Mobile Robots (ECMR 2013)*, Barcelona, Catalonia, Spain, September 2013.
5. Y. Zhang, K. Hornfeck, and K. Lee, "Adaptive face recognition for low-cost, embedded human-robot interaction," in *Intelligent Autonomous Systems 12*, Advances in Intelligent Systems and Computing, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds., vol. 193. Springer Berlin Heidelberg, 2013, pp. 863–872.

6. C. Cruz, L. Sucar, and E. Morales, “Real time face recognition for human-robot interaction,” in *FG ’08. 8th IEEE International Conference on Automatic Face Gesture Recognition* , 2008, pp. 1–6.
7. T. Ahonen, A. Hadid, and M. Pietikinen, “Face recognition with local binary patterns,” in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds., vol. 3021. Springer Berlin Heidelberg, 2004, pp. 469–481.
8. L. Wolf, T. Hassner, and Y. Taigman, “Similarity scores based on background samples,” in *Proceedings of the 9th Asian Conference on Computer Vision - Volume Part II*, ser. ACCV’09. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 88–97. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12304-7_9
9. L. Wolf, T. Hassner, and Y. Taigman, “Descriptor based methods in the wild,” in *Faces in Real-Life Images Workshop in ECCV. (2008)*.
10. X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.
11. M. Vo-Duc, A. Masselli, and A. Zell, “Real time face detection using geometric constraints, navigation and depth-based skin segmentation on mobile robots,” in *2012 IEEE International Symposium on Robotic and Sensors Environments*, 2012.
12. B. A. D. David S. Bolme, J. Ross Beveridge and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.
13. P. Viola and M. Jones, “Robust real-time object detection,” in *International Journal of Computer Vision*, 2001.
14. A. Martinez and R. Benavente, “The AR face database,” in *CVC Technical Report 24*, June 1998.