A face recognition method in the Internet of Things for security applications in smart homes and cities

Conference Paper · April 2018

DOI: 10.1109/SGCF.2018.8408934

CITATIONS
29

READS
2,072

2 authors, including:

Nashwan Adnan Othman
Knowledge University
11 PUBLICATIONS **169** CITATIONS

SEE PROFILE

A Face Recognition Method in the Internet of Things for Security Applications in Smart Homes and Cities

Nashwan Adnan OTHMAN
Department of Computer Engineering
Firat University
Elazig, TURKEY
Nashwan.adnan92@gmail.com

Ilhan AYDIN
Department of Computer Engineering
Firat University
Elazig, TURKEY
iaydin@firat.edu.tr

***Abstract*—In recent years, the security constitutes the most important section of the human life. At this time, the cost is the greatest factor. This system is very useful for reducing the cost of monitoring the movement from outside. In this paper, a real-time recognition system is proposed that will equip for handling images very quickly. The main objective of this paper is to protect home, office by recognizing people. For this purpose, the PIR sensor is used to detect movement in the specific area. Afterwards, the Raspberry Pi will capture the images. Then, the face will be detected and recognized in the captured image. Finally, the images and notifications will be sent to a smartphone based IoT by using Telegram application. The proposed systems are real-time, fast and has low computational cost. The experimental results show that the proposed face recognition system can be used in a real time system.**

***Index Terms*--Internet of things, Computer Vision, Raspberry Pi 3, Face recognition, Telegram.**

I. INTRODUCTION

Today, the security system field is a very important area in smart cities, offices, and homes. Security of the house and the family is important for everybody. Likewise, smart systems can provide Internet of Things (IoT) [1-2]. The IoT can be applied in smart cities in order to give various benefits that enhance citizens [3]. In other terms, smart homes can be made by utilizing the IoT. It has the ability to control and automate exact things of houses such as lights, doors, fridges, distributed multimedia, windows and irrigation systems [4-6]. The IoT is becoming popular in many sides of life, such as smart security, smart cities, healthcare, smart transportation, smart grids and online business. The objectivity of utilizing IoT is to share information and knowledge with everyone in everywhere around the world [7].

Computer vision can present more security system in the IoT platform for smart houses. It has abilities to recognize a person in the incorrect area and at the wrong time because this person may be a malicious one for the environment [8]. Face recognition system grow to be one of the most active research areas especially in recent years. It has an assortment of large applications in the ranges: public security, access control, credit card verification, criminal identification, law enforcement commerce, information security, human-computer intelligent interaction, and digital libraries. Generally, it recognizes persons in public areas such as houses, offices, airports, shopping centers and banks. This mechanism permits secure access to the house by detecting motion controlled by the embedded system.

The face is the most important part of human's body. So, it can reflect many emotions of a person. Long year ago, humans were using the non-living things like smart cards, plastic cards, PINS, tokens and keys for authentication, and to get grant access in restricted areas like ISRO, NASA and DRDO. The most important features of the face image are nose, eyes and mouth which are related to facial extraction [9-10]. Face detection and recognition system is simpler, cheaper, more accurate, and non-intrusive process as it is compared to other biometrics. The system will fall into two categories; face detection and face recognition. There are many methods to implement face detection such as Haar-like features, Eigen-face and Fisher-face. Then, analyzing the geometric features of facial images, such as, distance and location amongst eyes, nose and mouth were provided by several face recognition techniques [8]. There are a few techniques for fetching the most important features from face images to implement face recognition. One of these feature is extraction technique called Local Binary Pattern (LBP). LBP technique was produced by Ojala et al. [11]. LPB describes the shape and texture of a digital image. This technique provides good results and efficient for real-time applications. Haar-like features and LBP are robust when compared to the others. According to many studies [12-14] to get fast discriminatory performance and good results, LBP technique was chosen for face recognition. LBP generates the binary code that describes local texture pattern. From the LBP face image, the nose and eyes area are extracted, and for each image's pixel the LBP histograms will be drawn [9-10].

In this paper, Raspberry Pi 3 is utilized and Raspberry Pi camera is connected to it. The system will take an image when PIR sensor detects any movement. Then, computer vision is applied to the captured images. Subsequently, the system sends the images to a smartphone via the Internet. In this case, IoT based Telegram application is utilized to see the activity

978-1-5386-4478-2/18/$31.00 ©2018 IEEE

and get the images and notifications. In the paper, the Raspberry Pi single-board computer is a heart of the embedded face recognition system. It controls each of the peripherals.

II. METHODOLOGY

In the proposed system, a camera is utilized to achieve the image when a movement detected by PIR sensor. Then, computer vision module is applied to the captured images to detect and recognize the human faces. Then, it will send it to a smartphone. This system is very useful and important to secure a place. If there is no movement detected, the program will not go to face detection and recognition. Fig. 1 illustrates the flow chart for the proposed system.

In Fig. 1, the motion detection module detects any motion by using PIR sensor. Afterward, the algorithm will search for human faces and then face recognition will be processed. Then, the image will be sent to the smartphone.

Face recognition can be described as classifying a face either known or unknown via comparing a face and putting away known persons in the database. This can be finished by comparing the invariant features got from the strategies that catch the delegate variability of the faces or the structure.

Face recognition system is classified into steps; face detection and face localization according to Haar-like features. Through using weighted LBP algorithm, face features will be extracted [15-17]. By utilizing face recognition, the procedure of the person identification can be classified into three main stages as illustrated in Fig. 2.

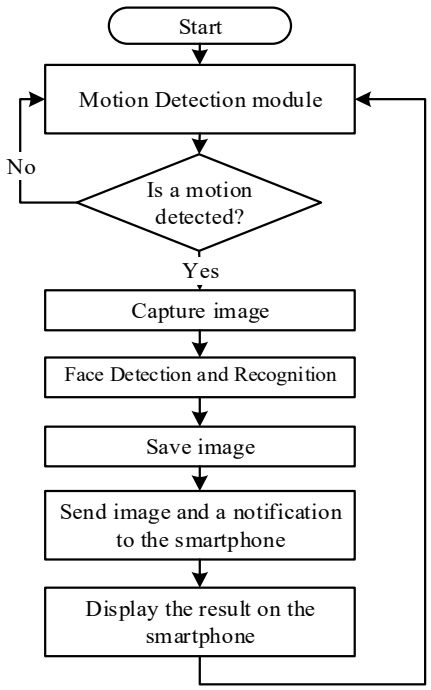


Figure 1. Flowchart for face recognition system

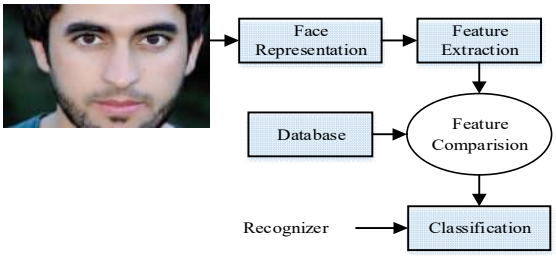


Figure 2. General steps for identification and recognition procedure

A. Face Representation

The initial step, face representation characterizes in how to display a face and check which algorithms which can be utilized for detection function. Haar-like feature and AdaBoost classifier are embraced to realize face detection [18]. There are a lot of factors of the captured image, like shine, many redundant data, contrast and image size which affect the precision of face recognition system. So, it is more vital to prepare the caught picture before face recognition. For this purpose, the input face is converted to grayscale and normalize it.

B. Feature Extraction

Face image has the most useful and special features which are extracted in the feature extraction stage. Face image with the images from the store will be compared to obtain the features. Feature extraction phase is the core of face recognition system. For the sampling of the face and to measuring matches among images, these features can be utilized.

There are some techniques in face recognition to achieve and bring the most important features from face images. For describing the shape and texture of a digital image, LBP can be used. This technique provides good results and efficient for real-time applications. Haar-like features and LBP are robust when compared to the others. LBP technique is finished via partitioning a picture into a number of small regions as shown in Figure 3. The features are extracted from every individual region. To portray the surroundings of pixels in the areas, these features are coded into binary patterns. Each area is processed to calculate the features. These calculated features from all areas are combined to a single feature histogram, which forms a demonstration of the picture.

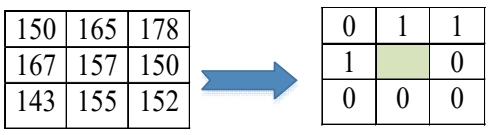
In this stage, LBP is calculated for each pixel. Generally, LBP works on grayscale pictures with the 3*3 neighborhood. First, LBP computes the binary relations among every pixel in the picture and its local neighbor points in the grayscale. After that, the binary relationship is weighted into an LBP code according to a confident set of laws. I (x_i, y_i) is any pixel surrounded by a local area of a picture, I(x_c, y_c) is the middle of the 3×3 window, the other 8 points are g_0, g_1..., g_7. For describing the local area texture as T = t (g_c, g_0..., g_7), other 8 pixels inside the window is used to set he threshold. The LBP code is computed for the middle pixel with coordinate (x, y) via:

$$T \approx t(s(g_0 - g_c),...,s(g_7 - g_c)) \quad (1)$$

Figure 3. A face image divided into 64 areas

LBP value is computed as follows. There are eight neighboring pixels in the window of the image. In the grayscale image, to compute the LBP value for pixels, the pixel is compared to all of its eight neighbors on its left-middle, left-top, left-bottom, right-bottom and right-top. When the middle pixel's value is bigger than the neighbor's value, write zero. Something else, write one. This offers an 8-digit binary number. Binary value normally stored in the center pixel location of the output picture like a decimal number. The entire process is illustrated in Figure 4. For example, the current pixel has value 157. The comparison starts from the neighboring pixel where the its label is zero. The value of the neighboring pixel with label zero is 150. As it is smaller than the current pixel value which is 157, the 0^{th}-bit position will set as zero in the 8-bit binary array. After that, it will be repeated in the counter clockwise direction. The next label positions one have value 165 which is bigger than the current pixel value. So set the first-bit position in the array is set to one. If the value of the neighboring pixel is equal to the current pixel value, write 1.

After calculating LBP operator, local features are created through calculating histograms of LBP over local image areas in local feature extraction process. When LBP mask is computed for every pixel, the feature vector of the picture can be constructed by computing the LBP histogram for each cell. This histogram can be viewed as a 256- dimensional feature vector. After that, the standardization of the LBP histogram and concatenate histograms of every cell will be completed. This offers a feature vector for the whole image. The length of the feature vector minimizes from 256 to 59 for a single cell. A face image is divided into 64 sections.



Pattern: (01100001)

LBP Code for Center pixel (157):

$$0*2^0+1*2^1+1*2^2+0*2^3+0*2^4+0*2^5+0*2^6+1*2^7$$

$$=2+4+128=134$$

Figure 4. Example of LBP calculation

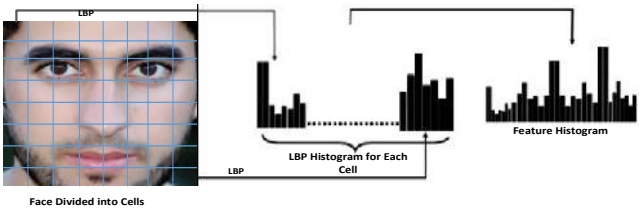


Figure 5. An example of extracting LBPH feature vector

A histogram with every possible label is built for each section. This means that each bin in a histogram characterizes a pattern and holds the quantity of its appearance in the section. The feature vector is subsequently constructed through concatenating the regional histograms to one large histogram. Finally, these histograms are used to measure the match among the pictures via computing the distance among the histograms. LBP codes of every pixel for an input picture are gathered into a histogram like a texture descriptor. The histogram of the picture LBP (x, y) can be characterized as:

$$LBPH(i)=\sum_{x,y} I\{i, LBP(x,y)\}, \quad i=0,...,n-1 \qquad (2)$$

In (2), n is the number of various labels created via the LBP operator.

C. *Classification*

Face image with the images from the store will be compared to obtain the features in the classification stage. The comparison is performed utilizing the local features acquired in the earlier phase of the algorithm. According to the face image, the maximum matching score is going to be the output of the classification portion. After the LBPH feature vector is extracted, the face recognition is performed via resorting to the K-Nearest Neighbor (KNN) classifier based on the histogram matching methods. For example, the commonly utilized Chi-square measure:

$$d_{x^2}(M,S)=\sum_{i=1}^{B}\frac{(M_i-S_i)^2}{M_i+S_i} \qquad (3)$$

In (3), M and S is the gallery and explores histogram objects. B is the number of bins in the histogram. To compare two face images, a sample (S) and a model (M), the diversity among the feature vectors have to be measured. This can be done with histogram features. After the distance between features calculated by Chi-square function, then KNN classifier will apply to identify faces. KNN is a data classification technique that can be utilized in the face recognition system. Fig. 6 illustrates the schematic diagram of nearest neighbor.

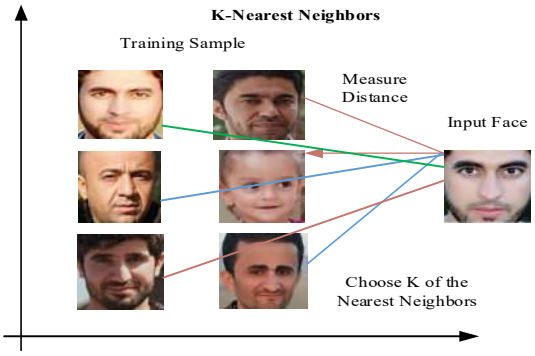


Figure 6. Graphic diagram of KNN classification

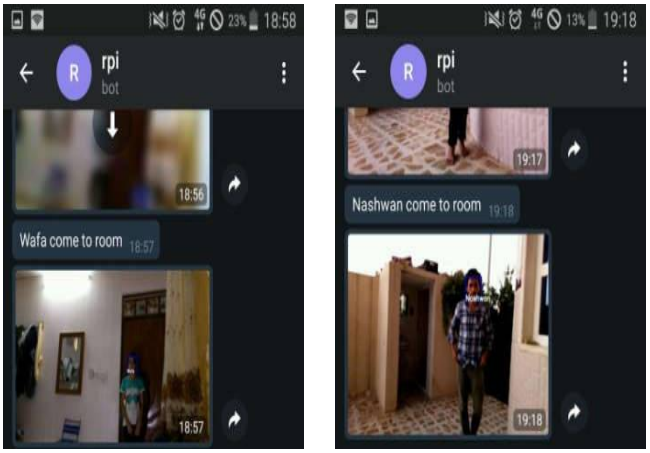
In KNN, input face will be compared with each example of training data. The meaning of the nearest neighbor technique is the select of uniqueness faces. It works based on the idea that is minimum distance from the query case to the training examples. All pixel in face denotes unique data.

III. EXPERIMENTAL RESULTS

Smartphone application was integrated with the proposed system to develop a smart movement and face recognition for a personal area. The Raspberry Pi camera successfully captures the image when PIR sensor detects any movement. After that, face detection and recognition are implemented. The system was able to successfully identify the faces in the captured images. The algorithm has been applied to all the images. The real-time face detection is done by means of Haar-like features and real-time face recognition is done by means of local binary pattern (LBP). The face detection and recognition algorithm are more reasonable for real-time because they need less CPU resource and low costs.

This application performs four operations such as motion detection, capturing images, face detection and recognition with sending output images and notifications to the user's smartphone. The system will be activated when a motion is detected. At the same time, the camera captures the events. The notifications and the images will be sent to a smartphone application "Telegram" as shown in Fig. 7. Raspberry Pi 3 has a Wi-Fi wireless technology and a Bluetooth. This is useful to view activity and show images immediately on the smartphone gadgets. The received output images and notifications on the smartphone are given in Fig. 8.

Figure 7. Screenshot of received notification on the smartphone



(a) wafa face (b) nashwan face

Figure 8. The output of the algorithm for different face recognition

In Figure 8, the algorithm detects and recognizes human faces successfully for known faces that stored in the database. If unknown face enters the room, the system will notify unknown person is coming to the room as appearing in Fig. 9.

Currently, OpenCV gives three technique for face recognition: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Local Binary Patterns Histograms (LBPH). In this paper, LBPH is used for face recognition. Also, the result of LBPH algorithm is compared with PCA and LDA algorithm. PCA as well called Eigenfaces and LDA called Fisherfaces. All of the three previously mentioned techniques utilizes the training set a bit differently. Both PCA and LDA techniques are one of the well-known techniques for face recognition. They are known to be very sensitive to pixel level variations like facial expression, illumination and pose variations. PCA and LDA find a mathematical description of the most predominant features of the training set as a whole. PCA and LDA are both affected by brightness in the actual life.

LBPH analyzes every face in the training set independently and separately. While the PCA and LDA technique looks at the dataset as a whole. LBPH will perhaps implement better in various situations and light conditions. Nevertheless, it will depend on training and testing data sets. Compared to PCA and LDA techniques, LBPH method gives better result under lighting and pose variations. LBPH accomplishes the highest accuracy in various probe sets in the bellow tests.

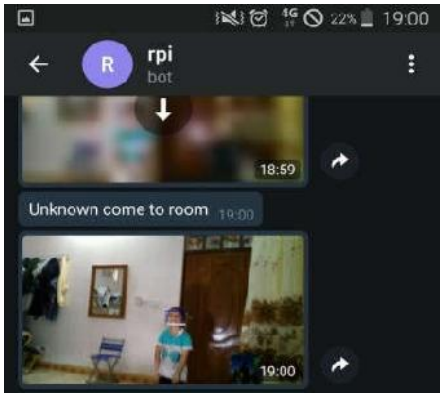


Figure 9. The output of the algorithm for unknown face

In the proposed system, the Raspberry Pi 3 platform is used. Raspberry Pi 3 is based running at 1.2GHz, 64-bit quad-core processor with 1 GB RAM. TABLE I shows time execution comparison for each of three algorithms on the Raspberry Pi 3.

TABLE I compare the time execution of three face recognition algorithms. As shown in TABLE I, the proposed LBPH algorithm outperforms from others. Generally, LBPH algorithm robustness and makes the best performance result. The accuracy of each algorithm is shown in TABLE II.

As seen in Table II, the accuracy of the LBPH algorithm is better than PCA and LDA. Also, PCA is better than LDA, but when the number of a sample is large, LDA outperforms PCA.

IV. CONCLUSIONS

In this paper, an embedded face detection and recognition with smart security system are designed to be able to capture an image and send it to a smartphone. So, when a face is detected and recognized, the system will notify the user by using a smartphone and displays who is he in that area. By adding the face recognition system, people will be easily recognized and a safer city will be built. Also, a possible solution is proposed to utilize computer vision in the IoT in this paper. Smartphone is the main benefit of this paper which is utilized by the client to obtain notifications with the captured images. This system helps to enhance and automate the security of industries, cities, homes and towns. In this paper, LBPH algorithm is used to recognize faces. Also, the result of LBPH algorithm is compared with PCA and LDA algorithm. The results show that LBPH method gives better results under lighting and pose variations.

TABLE I. TIME EXECUTION COMPARISON FOR FACE RECOGNITION ALGORITHMS

Method	1 image (seconds)	22 images (seconds)	65 images (seconds)
LBPH	0.71±2.8	15.04±4.3	35.21±9.6
PCA	2.07±4.5	43.58±6.9	113.17±11.2
LDA	1.93±4.2	39.44±5.8	98.90±10.6

TABLE II. DETECTION RATE COMPARISONS OF ALGORITHMS

Method	Test(1) 22 image	Test (2) 65 image
	Detection Rate %	*Detection Rate %*
LBPH	90.90±0.9	93.8±1.7
PCA	81.80±2.9	81.5±3.1
LDA	77.02±2.3	84.6±2.6

REFERENCES

[1] N. A. Othman, I. Aydin, "A New IoT Combined Face Detection of People by Using Computer Vision for Security Application," in Proc. 2017 IEEE International Conference on Artificial Intelligence and Data Processing (IDAP17), pp. 1-5.

[2] N. A. Othman, I. Aydin, "A New IoT Combined Body Detection of People by Using Computer Vision for Security Application," in Proc. 2017 IEEE International Conference on Computational Intelligence and Communication Networks (*CICN 2017*), pp.1-5.

[3] H. Gu, D. Wang, "A content-aware fridge based on RFID in smart home for homehealthcare," in Proc. 2009 11th Advanced Communication Technology Conference, pp. 987–990.

[4] S. Luo, H. Xia, Y. Gao, J.S. Jin, R. Athauda, "Smart fridges with multimedia capability for better nutrition and health," in Proc. 2008 IEEE International Symposium on Ubiquitous Multimedia Computing, pp. 39–44.

[5] M. Rothensee, "A high-fidelity simulation of the smart fridge enabling productbased services," in Proc. 2007 3rd IET International Conference on Intelligent Environment, pp. 529–532.

[6] E. Borgia, "The Internet of things vision: Key features, applications and open issues," Computer Communications. Vol. 54, 1–31, December 2014.

[7] Vienna University of Technology, European Smart Cities, 2015.http://www.smart-cities.eu (accessed 28.09.16).

[8] P. N. Belhumeur, J.P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. pp. 711–720, July 1997.

[9] T. Ojala and M. Pietikainen. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971-987, July 2002.

[10] T. Ahonen, A. Hadid, M. Pietikainen, "Face description with LBP: Application to face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pp. 2037–2041, December 2006.

[11] T. Ojala, M. Pietikainen and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," Pattern Recognition vol. 29, pp. 51-59, January 1996.

[12] H. Xiaohua, S.J. Wang, X. Liu, G. Zhao, X. Feng, X., & Pietikainen, M. (2017). "Discriminative Spatiotemporal Local Binary Pattern with Revisited Integral Projection for Spontaneous Facial Micro-Expression Recognition," IEEE Transactions on Affective Computing, in press.

[13] T. Ahonen, A. Hadid, M. Pietikainen and T. M Aenpaa. "Face recognition based on the appearance of local regions", In Proc. 2004 17th IEEE International Conference on Pattern Recognition, pp. 1051-4651.

[14] R. Gottumukkal and V.K. Asari, "An Improved Face Recognition Technique Based on Modular PCA Approach," Pattern Recognition Letters, vol. 25, pp. 429- 436, Mar. 2004.

[15] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in Proc. 2002 IEEE Image Processing, pp.1522-4880.

[16] M. Turk and A. Pentland, "Eigenfaces for recognition," Cognitive Neuroscience, vol. 3, January 2001.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted classifier of simple features," in Proc. 2001 IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518.

[18] T. Hastie, S. Rosset, J. Zhu, H. Zou, "Multi-class adaboost," Statistics and its Interface, vol. *2*, pp. 349-360, January 2006.