Internet of Things Approach for Face Detection & Face Recognition using Raspberry Pi

Swati Sahu[1] Prof. Vijay Sharma[2]
[1]M.E. Student [2]Assistant Professor
[1,2]Department of Electronic and Communication Engineering
[1,2]Swami Vivekanand College of Engineering, Indore, India

Abstract— In this paper, we are presenting a proposed system for Smart Surveillance model in which we are implementing facial monitoring system by embedded face detection and face tracking algorithm and it consists of three steps namely: facial detection, feature extraction and recognition by using Haar classifier and Eigen face approach. The purpose of the project is to make a system, which would detect and take snapshots and videos of the Human motion when detected and upload to an external server.

Key words: Face detection, Face recognition, Face authentication, Raspberry Pi, IoT

I. INTRODUCTION

Nowadays people keep on moving around the world because of their job and for other reason. So monitoring about what happening in their home or office has become a must needed one today, there is a pressing need to design and develop authentication technology in view of increasing vicious crimes and terrorist threats that can assist individual and safe living of people. Biometric authentication is one of the best automatic identification or identity verification of an individual based on physiological or behavioural characteristics such as fingerprint, iris, face, vein, voice, and so on. These kinds of authentications are most commonly used to safeguard international borders, control access to facilities, and enhance computer network security.

"Face authentication has fascinating approach that other biometrics does not have; facial images can be captured from a distance, any special actions are not always required for authentication, and a crime-deterrent effect can be expected because the captured images can be recorded and we can see who the person is at a glance. Due to this approach, the face recognition technique is expected to be applied extensively not only to security applications such as video surveillance but also to image indexing, image retrievals and natural user interfaces."

"The authors have previously developed face detection and recognition technologies [1-2]. In previous work researchers considered head and shoulder of the object to detect the more appropriately its motion, since it is the most unvarying part of human body. The range of monitoring in this study is up to five meters hence, it is difficult to detect face at a long distance since the face will be small and blurry [3]. In the [3-4] paper the researchers objective to detect the motion. In this project it used a Raspberry Pi Model B to connect the web camera to capture the footage sending an email on detection of motion [4-8]. The python script matches the last frame and the present frame of the live video, if there is any difference then the motion flag is set, triggering all the events"[4].

"Subsequently, there is a pressing need to design and improvements on the previous technologies. This paper describes such advances in the authors' face detection and recognition technologies. For face detection, a hierarchical scheme for combined face and eye detection has been developed [5-6]. For face recognition, the Eigen features of the face for tracking its position are detected using openCV [7].

To develop an active surveillance camera that has the capability of identifying the context of the scene being monitored and able to give notification or alarm on event occurrence. In order to achieve this it's necessary to choose the hardware wisely. "

The Functionality of this system is mainly categorized in following steps

1) The first step is to have a good Dataset of sample images of faces with multiple images for each individual
2) The next step is to detect faces from the sample images and use them to train the face.
3) Methodology for recognizing and distinguishing between the known faces and unknown faces are then designed (Using Eigen Face).
4) If the user is Unknown then gives notification the owner.

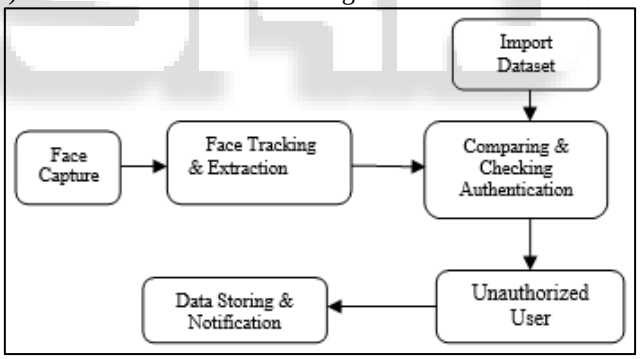

Fig. 1: System's overview

The rest of this paper is organized as follows: Face detection technology and face recognition technology are described in Section 2 and Section 3, respectively. Section 4 discussed the system model. Experimental results are given in Section5, and conclusions are presented in Section 6.

II. FACE DETECTION

"The human face is a dynamic object that comes in many forms and colors. There are various different algorithms exist to perform face detection, each has its own weaknesses and strengths. Some use flesh tones, some use contours, and other are even more complex involving templates, neural networks, or filters. These algorithms experience from the same problem; they are computationally expensive [2]. An image is only a collection of color and/or light intensity values. Analyzing these pixels for face detection is time consuming and difficult to accomplish because of the wide variations of

shape and pigmentation within a human face. Pixels often require reanalysis for scaling and" precision. Viola and Jones give an algorithm, called Haar Classifiers, to fastly detect any object, including any object with higher accuracy, using AdaBoost classifier cascades that are based on Haar-like features and not pixels [9].

A. Haar Cascade Classifiers

"Detecting human facial features, such as the mouth, eyes, and nose require that Haar classifier cascades first be trained [1]. In order to train the classifiers, this recommend AdaBoost algorithm and Haar feature algorithms must be implemented. Fortunately, Intel developed an open source library devoted to easing the implementation of computer vision related programs called Open Computer VisionLibrary (OpenCV).
The Face detection technology operate using the following stages

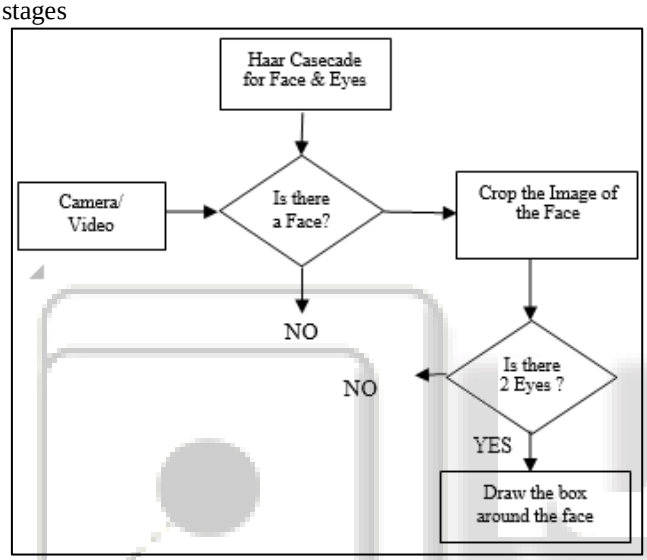

Fig. 2.1: Face detection flow operation

"Haar Cascading Classifiers are trained with several hundred "positive" sample views of a particular object and arbitrary "negative" images of the same size. After that classifier is trained it can be applied to area of an image and detect the object in question. To search for the object in the entire frame, the search window can be moved across the image and check every location for the classifier. This process is most commonly used in image processing for object detection and tracking. OpenCV contains many pre-train classifier for full-body, mouth, nose, eyes many more, all these classifier files are XML files."In this project we are using haarcascade_frontalcatface.xml and haarcascade_eye.xml for face detection. The result of face Detection using Haar classifier cascades shown in Figure 5.2.

III. FACE RECOGNITION

"Eigenfaces [9] measured as 2-D face recognition problem, faces will be mostly upright and frontal. That's why 3-D information about the face is not required that reduces complexity of our project. It convert the face images into a set of basic functions which essentially are the principal components of the face images seeks directions" in which it is more efficient to represent the data.

A. Algorithmic Description

"The algorithm for the facial recognition using eigenfaces technique is described in figure3.1. First, the raw or input images of the training set are transformed into a set of Eigenfaces E. Afterwards; the weights are calculated for each and every image of the training set and stored in the set W.

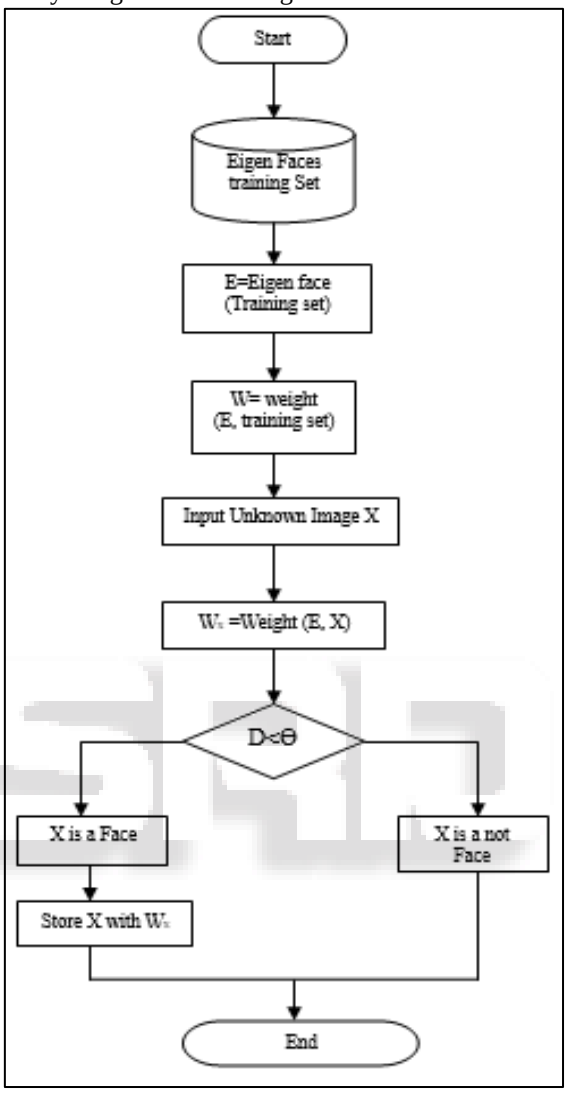

Fig. 3.1: High-level functioning principle of the Eigenface based facial recognition algorithm.

If observing an unknown image X, the weights are calculated for that particular image and stored in the vector W_X. After that, W_X is compared with the weights of images, of which one knows for certain that they are faces (the weights of the training set W). One the best way to do it would be to regard each weight vector as a point in space and calculate an average distance D between the weight vectors from W_X and the weight vector of the unknown image W_X (the Euclidean distance [9] be a measure for that).

The problem with the image representation we are given is its high dimensionality to achieve accuracy. Two-dimensional p×q grayscale images span a m=pq dimensional space vector, so an image with 10×10 pixel lies in a 10,000 image space already. The question is: Are all dimensions equally useful for us? We can only make a decision if there's any variance in data, so what we are looking for are the components that account for most of the information. The

Principal Component Analysis (PCA) was independently proposed by [9] to turn a set of possibly correlated variables into a smaller set of uncorrelated variables. The idea is, that a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The PCA method finds the directions with the greatest variance in the data, called principal components [9-10].

Let $X = \{x_1, x_2, \ldots\ldots\ldots, x_n\}$ be a random vector with observations $x_i \in R^d$.

Compute the mean μ

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Compute the covariance matrix S

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)\ (x_i - \mu)^T$$

Compute the Eigen value $\boldsymbol{\lambda_i}$ of S

$$S\,\boldsymbol{V_i} = \boldsymbol{\lambda_i}\,\boldsymbol{V_i} \qquad \{i=1, 2,\ldots, n\}$$

Order the eigenvectors descending by their eigenvalue. The k principal components are the eigenvectors corresponding to the k largest Eigenvalues. The k principal components of the observed vector x are then given by

$$y = W^T(x - \mu)$$

Where, $W = (\boldsymbol{V_1}, \boldsymbol{V_2}, \boldsymbol{V_3}, \ldots\ldots, \boldsymbol{V_n})$

The reconstruction from the PCA basis is given by

$$x = Wy + \mu$$

Where, $W = (\boldsymbol{V_1}, \boldsymbol{V_2}, \boldsymbol{V_3}, \ldots\ldots, \boldsymbol{V_K})$

The Eigenfaces method then performs face recognition by

Projecting all training samples into the PCA subspace. Projecting the query image into the PCA subspace finding the nearest neighbor between the projected training images and the projected query image. Still there's one problem left to solve. Imagine we are given 400 images sized 100×100 pixel. The Principal Component Analysis solves the covariance matrix $S = XX^T$, where is given below in this example

$$size(X) = 10000 \times 400$$

You would end up with a 10000×10000 matrix, roughly 0.8GB. Solving this problem isn't feasible, so we'll need to apply a trick. From your linear algebra lessons you know that M×N matrix with M>N can only have N-1 non-zero eigenvalues. So it's possible to take the eigenvalue decomposition $S = X^TX$ of size $N \times N$ instead:

$$X^TXv_i = \lambda_i vi$$

And get the original eigenvectors of $S = XX^T$ with a left multiplication of the data matrix:

$$XX^T(Xv_i) = \lambda_i(Xv_i)$$

The resulting eigenvectors are orthogonal, to get ortho-normal eigenvectors they need to be normalized to unit length. I don't want to turn this into a publication, so please look into [9-10] for the derivation and proof of the equations

B. Preparing Dataset

For Face recognition we have to create a good Dataset of sample images of faces with multiple images for each individual person. In first user Enter his/her name and then camera enables and takes taking 25 images and crop the face in the images. Now create the training set data using Eigen Face recognizer (set threshold 15) for face recognition

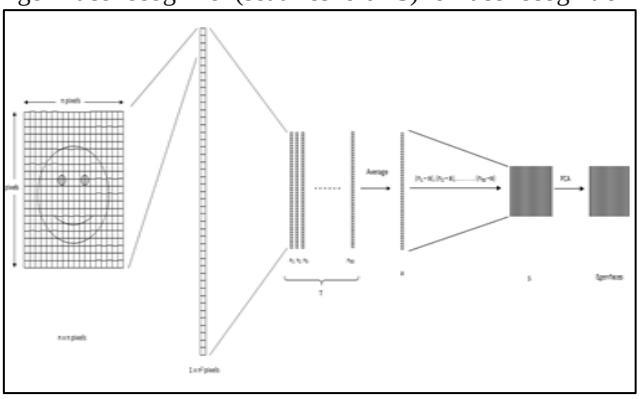

Fig. 3.2: Splitting the Face

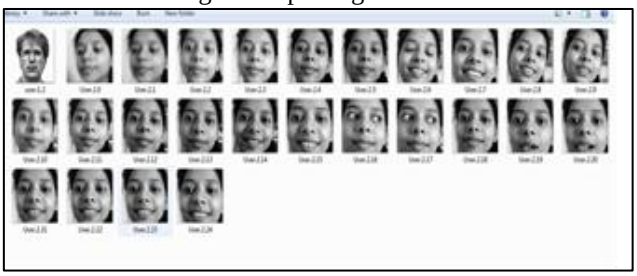

Fig. 3.3: Created the Dataset

IV. SYSTEM MODEL

"To develop an active surveillance camera that has the capability of identifying the context of the scene being monitored and able to give notification or alarm on event occurrence. In order to achieve this it's necessary to choose the hardware wisely".

The Process steps of this system is mainly categorized in following steps

1) Initially a set of images needed to create the database which is shown in Fig4.1.
2) Pre-set algorithms, logic and computational methods are used to identify images and find out the pixel points in train of images (Discuss in section2 & 3).
3) Once the train of images is ready then apply computational methods to identify the points and recognize gestures of 2D images
4) Methodology for recognizing and distinguishing between the known faces and unknown faces are then designed.
5) Alert system and database updating dynamically is created for further course of action.

A. System Flow Diagram

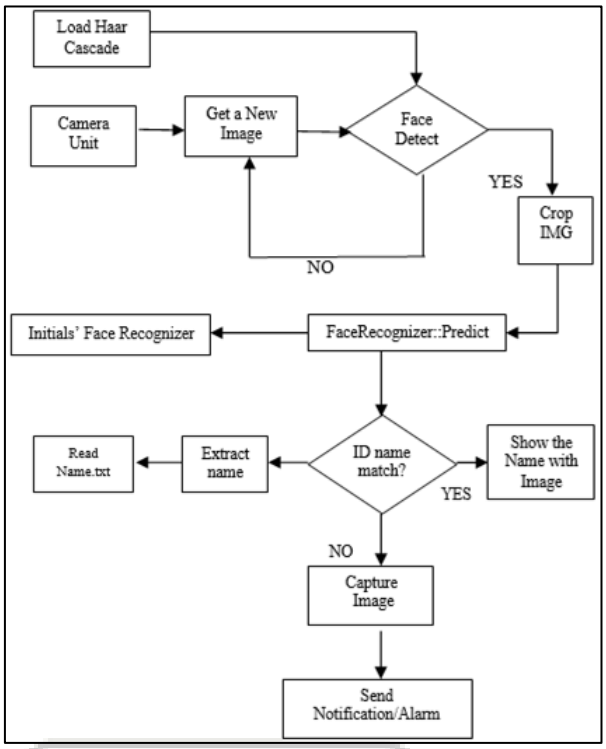

Fig. 4.1: Face Recognition Flow chart

B. Hardware Design

1) Raspberry Pi

The Raspberry Pi which is a single board minicomputer developed in UK. The Raspberry Pi3 Model B is used in this project. It is a high specification, minicomputer embedded on a chip having 802.11n Wi-Fi, Bluetooth 4.0 and a quad-core 64-bit ARM cortex A53 running at 1.2 GHz.

2) Camera Unit

In this system USB camera module is used.

To perform the entire task (i.e. face detection and face recognition) code for the Hardware setup, python language is used to access the editor Linux terminal is required. The Face detection and recognition part is carried out by Python Script. General System flowchart is shown in Figure 4.3

V. RESULT

Figure5.4 will give the unauthorized user image that is read to upload an external server and notify the authority. The result can be visible in the below set of images:

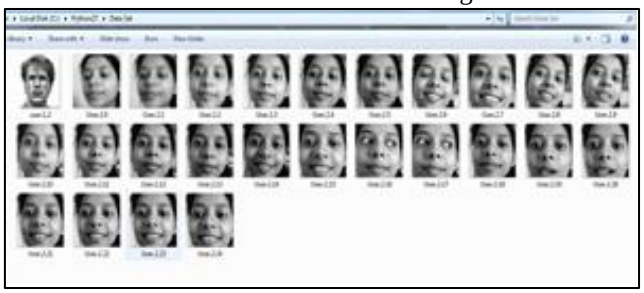

Fig. 5.1: Prepare Dataset

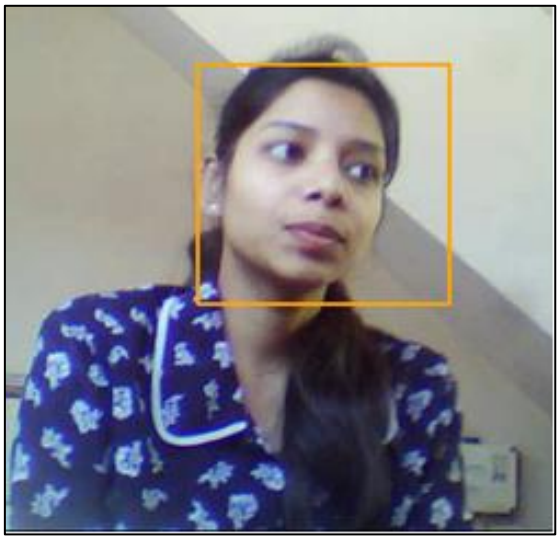

Fig. 5.2: Result of Face Detection (Haar Casecade)

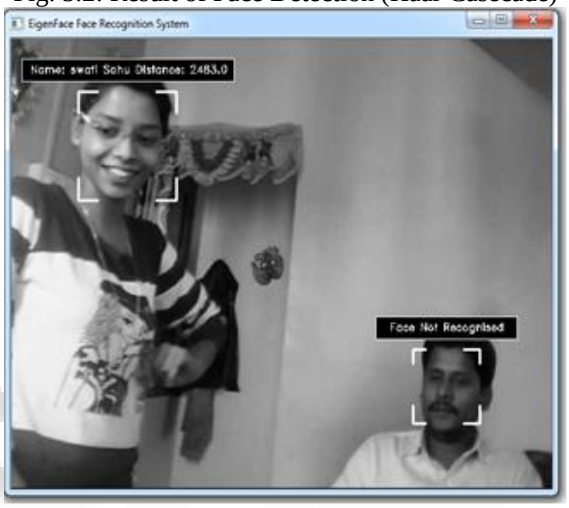

Fig. 5.3: Result of Face Recognition (Eigenface)

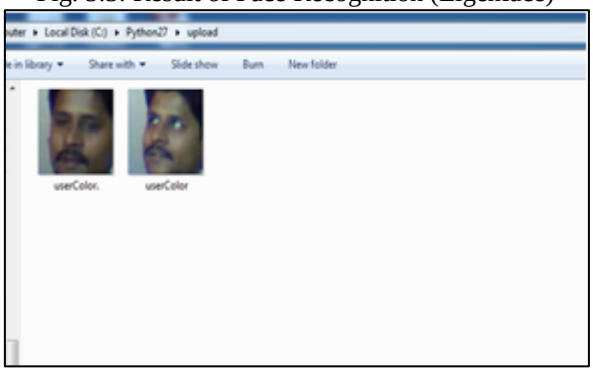

Fig. 5.3: Image of Unauthorized User

VI. CONCLUSION

With the objective of designing a very fast and efficient alert and monitoring system that can be used for a wide range of application, we have used an approach which is unique in nature with very less computational time , precise accuracy in detection ,recognizing and create notification system dynamically in no time. The functionality and effectiveness of this system can be termed in a single word "APT" when we talk about its responsiveness in distinguishing between an authorized and an unauthorized one and notify the authority concerned.

REFERENCES

[1] P. Viola and M. Jones. Robust Real-time Object Detection. International Journal of Computer Vision, 57(2):137–154, 2002.
[2] A.M.Patil, Dr. Satish R. Kolhe, Dr. Pradeep M. Patil (2009), "FaceRecognition by PCA Technique", Second InternationalConference on Emerging Trends in Engineering and Technology, ICETET-09.
[3] Wilson Feipeng Abaya, Jimmy Basa, Michael Sy Electronics "Low Cost Smart Security Camera with Night Vision Capability Using Raspberry Pi and OpenCV" The Institute of Electrical and Electronics Engineers Inc. (IEEE) – Philippine Section 12-16 November 2014 Hotel Centro, Puerto Princesa, Palawan, Philippines.
[4] Aamir Nizam Ansari \ Mohamed Sedkyl, Neelam Sharma2, Anurag Tyagil lFaculty of Computing, Engineering and Sciences, Staffordshire University, Stoke-on-Trent, United.
[5] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, "Face recognition by independent component analysis," IEEE Transactions on Neural Networks, vol. 13, no. 6, pp. 1450–1466, 2002. Name of Author(s), "Title of the research", Citation Details, year.
[6] IEEE Review and Comparison of Face Detection Algorithms Cloud Computing, Data Science & Engineering - Confluence, 2017 7th International Conference on December.
[7] International Conference on ICT For Smart Society (ICISS Implementation of Face recognition algorithm for biometrics based time attendance system by Adrian Rhesa Septian Siswanto; Anto Satriyo Nugroho; Maulahikmah Galinium.
[8] Kingdom" An Internet of Things Approach for Motion Detection using Raspberry Pi" presented at (IC1T),2015
[9] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces" Proceedings. 1991 IEEE Computer Society Conference on Computer Visionand Pattern Recognition,Maui, HI, 1991, pp. 586-591
[10] https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html.

