Improved Intruder Detection System for Restricted Areas Using Internet of Things

[1]Dr.C.Kalyana Chakravarthy, [2]Dr.M.Chandrasekhar

[1]Department of CSE, [2]Department of IT
MVGR College of Engineering (Autonomous)
Vizianagaram, India

***Abstract*- Security is one of the major issues in the world. Many locations, like research centers, Government agencies, Prisons, ATMs etc. required restricted access. These sensitive locations must be well secured so that when any known intruder enters the location, authorities must be notified. Sometimes human surveillance will not be sufficient to secure these locations. Hence there need an automated system to fulfill the job of controlling the intrusion. The automated system must be able to detect the intrusion and notify the authorities immediately such that any preventive actions can be taken. The system must be able to identify the persons entering the area and check for any discrepancies. Biometric-based techniques have emerged as the most promising option for recognizing individuals in recent years. Face recognition appears to offer several advantages over other biometric methods, as some these require some voluntary action by the user, i.e., the user needs to place his hand on a hand-rest for fingerprinting or hand geometry detection and has to stand in a fixed position in front of a camera for iris or retina identification. Our system can be used in these sensitive locations where human surveillance is not sufficient. The system uses facial recognition to identify the persons. The system is equipped with a camera, proximity sensor and alarm (LED for representation). The camera is placed such that the person's face who wish to enter the area can be captured. The captured person is then identified to be intruder or not and thus following actions can be taken. The identification of the captured image is done using some Face recognition techniques. The intrusion is notified to the respective authorities over email and also the alarm will be turned on immediately.**

***Keywords*- Intrusion; Automated System;Surveillance; facial recognition.**

I. Introduction

Many face recognition systems are currently being used in various fields. It is used for person detection, automated attendance systems, gender classification, age detection etc. Many security systems use face recognition but have some limitations. The current Face Recognition Systems and applications in the market have deficiencies that range from reliability problems, reduced recognition accuracies in certain environment [1], complicated feature extraction, high setup costs and performance issues. Existing systems are not completely automated and require humans at some point of functionality of the system. These systems do not have any provision to send intrusion details in real time over the internet to the authorized persons and at the same time alerting the local security where the intrusion took place. Also, many systems continuously capture video and try to identify faces
from various frames of the video. This involves recording and
storing very large sets of data. The problem with these systems is that they do not have any trigger to identify any person
approaching and thus have to record all the time. The proposed system overcomes these limitations. LBPH face recognizer is used, which is better than EigenFaces [2] or FisherFaces [3][4] in cases where illumination is not proper. Also, LPBH recognizer works fine with even less number of images of a person in the database which is not the case in the other two algorithms [5].

By integrating the concept of “Internet of Things (IOT)”, a better system which communicates over internet and also with the other devices (proximity sensor, camera, alarms) within the system [6] was developed. A proximity sensor is used to detect the approach of any person which eliminates the requirement of continuous capturing and saves a lot of memory. Only the images of the persons approaching are captured. Another important factor of the proposed system is the method of notification regarding the intrusion. The system not only alerts the local security, but also sends email to the authorities immediately. This way a proper coherence can be established between the local security and the authorities who may not be locally present.

II. Existing SysteMS

A. Eigen Faces

Eigenfaces is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition. The Eigenfaces method take a holistic approach to face recognition: A facial image is a point from a high-dimensional image space and a lower-dimensional representation is found, where classification becomes easy. A **set of eigenfaces** can be generated by performing a mathematical process called principal component analysis (PCA) on a large set of images depicting different human faces. The eigenfaces that are created will appear as light and dark areas that are arranged in a specific pattern. This pattern is how different features of a face are singled out to be evaluated and scored.

These eigenfaces can now be used to represent both existing and new faces: we can project a new (mean-subtracted) image on the eigenfaces and thereby record how that new face differs from the mean face. The eigenvalues associated with each eigenface represent how much the images in the training set vary from the mean image in that direction.

B. Fisher Faces

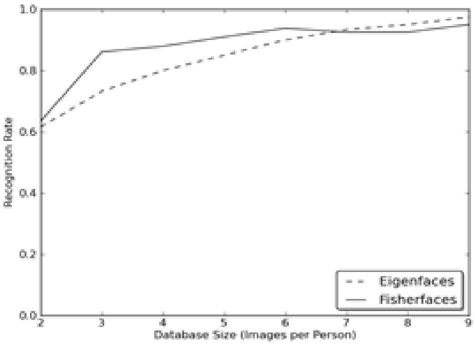
The Principal Component Analysis (PCA), which is the core of the Eigenfaces method, finds a linear combination of features that maximizes the total variance in data. While this is clearly a powerful way to represent data, it doesn't consider any classes and so a lot of discriminative information *may* be lost when throwing components away. Imagine a situation where the variance in your data is generated by an external source, let it be the light. The components identified by a PCA do not necessarily contain any discriminative information at all, so the projected samples are smeared together and a classification becomes impossible.

The Linear Discriminant Analysis performs a class-specific dimensionality reduction and was invented by the great statistician Sir R. A. Fisher. He successfully used it for classifying flowers in his 1936 paper *The use of multiple measurements in taxonomic problems*. In order to find the combination of features that separates best between classes the Linear Discriminant Analysis maximizes the ratio of between-classes to within-classes scatter, instead of maximizing the overall scatter. The idea is simple: same classes should cluster tightly together, while different classes are as far away as possible from each other in the lower-dimensional representation. This was also recognized by Belhumeur, Hespanha and Kriegman and so they applied a Discriminant Analysis to face recognition.

C. Fisher Local Binary Pattern Histogram

Eigenfaces and Fisherfaces take a somewhat holistic approach to face recognition. You treat your data as a vector somewhere in a high-dimensional image space. We all know high-dimensionality is bad, so a lower-dimensional subspace is identified, where useful information is preserved. The Eigenfaces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components with a maximum variance over all classes aren't necessarily useful for classification. So to preserve some discriminative information we applied a Linear Discriminant Analysis and optimized as described in the Fisherfaces method.

Now real life isn't perfect. One simply can't guarantee perfect light settings in your images or 10 different images of a person. So what if there's only one image for each person? Our covariance estimates for the subspace *may* be horribly wrong, so will the recognition. The following graph is based on the results performed on AT&T database extracted from the OpenCV documentation.



a.

Fig. 1. Recognition Rate Variation with Database Size

So in order to get good recognition rates you'll need at least 8(+-1) images for each person and the Fisherfaces method doesn't really help here.

So some research concentrated on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector, but describe only local features of an object. The features we extract this way will have a low-dimensionality implicitly. A fine idea! But we'll soon observe the image representation we are given doesn't only suffer from illumination variations. Think of things like scale, translation or rotation in images - your local description has to be at least a bit robust against those things. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. We'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called *Local Binary Patterns* or sometimes referred to as *LBP codes*. The first LBP operator described in literature actually used a fixed 3 x 3 neighborhood just like this:

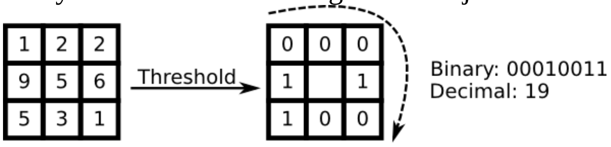


Fig. 2. LBP Code Generation

In order to enhance the robustness of illumination changes, expression change and attitude deflection, inspired by median filtering, a LBPH algorithm based on pixel neighborhood gray median(MLBPH) is proposed. Compared with the LBPH algorithm, the improvement of the MLBPH algorithm is that when the LBP eigenvalues are calculated, the central pixels are replaced by the median values of the sampled values of their neighborhood sampling points.

The definition of the median is shown in below formula

X(1+n)/2

X(n/2) +X(n/2+1)

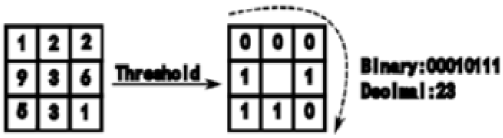


Fig. 3. MLBP Code Generation

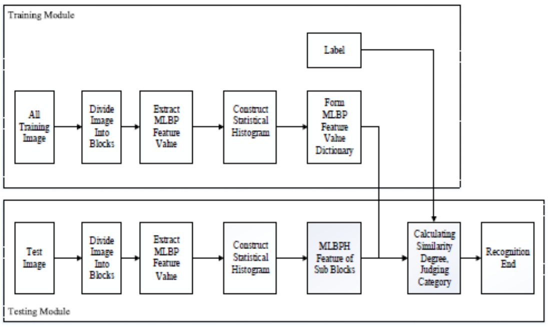

Training Module
Label
All Training Image
Divide Image Into Blocks
Extract MLBP Feature Value
Construct Statistical Histogram
Form MLBP Feature Value Dictionary
Test Image
Divide Image Into Blocks
Extract MLBP Feature Value
Construct Statistical Histogram
MLBPH Feature of Sub Blocks
Calculating Similarity Degree, Judging Category
Recognition End
Testing Module

Fig. 4. Training MLBP Images

III. PROPOSED SYSTEM ARCHITECTURE

The proposed system basically identifies intrusion and notifies the event to the authorities. It's functions include the following :

A. *Determining the approaching person*

The primary function of the system is to determine whether a person is approaching the system or not. In case the person approached the system, an immediate trigger must be sent to activate camera to capture the image. If the approach is not detected, the camera must be continuously capturing. To overcome the memory overhead, this functionality of the system is quite necessary.

B. *Capturing the person approached and saving the image to the disk*

The next function for the system is to capture the image of the person who approached the system. The trigger from the previous function is used to enable the camera and capture the image. This captured image must be saved into the disk. The processing of this image can be done without saving it to the disk, but for future reference and other details, the system stores the captured image.

C. *Training the face database*

In order to compare faces and identify them, the pre-given database of faces must be trained using some face recognizer (LPBH or Fisher face). The images in the database must be done some preprocessing, like cropping the face, converting to grayscale etc, and then trained against a recognizer. This recognizer is then used to recognize faces with the trained dataset.

D. Face recognition

The major functionality of the system is to perform face recognition. The captured image must be done required preprocessing and the has to be compared with the existing dataset of faces. This is done using the face recognizer and the trained dataset. The “confidence” value is retrieved which is used as threshold for matching the person. From this process the person is matched or not is identified and whether to notify authorities or not is decided.

E. Sending Email

The immediate action for the system when an intrusion is identified is to notify the authorities. The system must be able to communicate over the internet regarding the intrusion happened. For this the system sends email notifications to the registered email addresses along with the image of the intruder and other details regarding the event. With the increased use of smartphones in the recent times, email notification is an effective way of notifying for events like these which require immediate reaction.

F. Notifying local security

It is not only important to notify authorities regarding the intrusion event, but also the place where the intrusion happened must be alarmed, so that immediate action can be

taken avoiding any kind of loss. For this the system must alarm the local security persons as soon as any intrusion is detected.

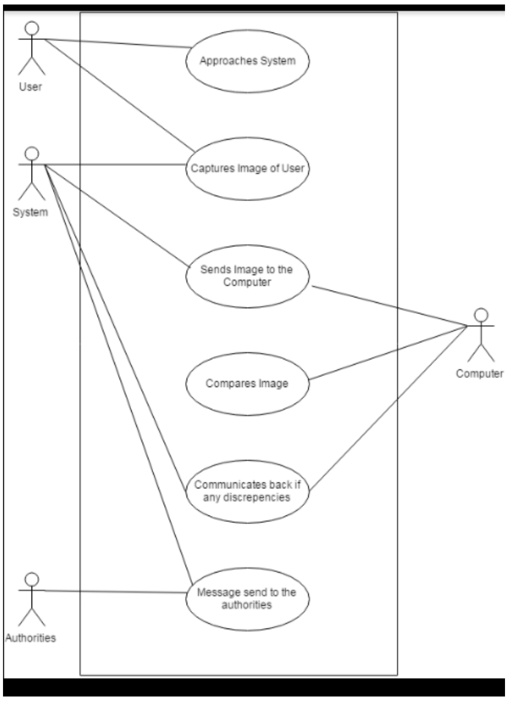

User
System
Authorities
Computer
Approaches System
Captures Image of User
Sends Image to the Computer
Compares Image
Communicates back if any discrepencies
Message send to the authorities

Fig. 4. Proposed System Use Case Diagram

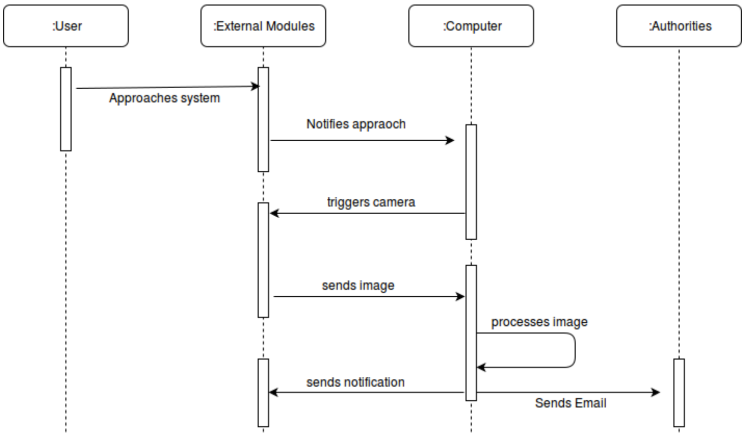


Fig. 4. Proposed System Sequence Diagram

When the user approaches the system, the System gets notified. The System triggers the camera connected to take a picture. The camera captures the image and stores it into the disk. The program processes the image and compares it with the database and sends notification to start an alarm and also sends email simultaneously.

IV. PROPOSED SYSTEM IMPLEMENTATION

A. Working with the camera

Once the distance is received by the program, the camera is triggered. The web camera is identified by it's device id and an *VideoCapture* object is created for the camera. The image is then captured and is stored as a matrix. The matrix is written into the disk as a jpg file using imread() function of OpenCV. The camera will be turned off as soon as the image has been captured.

B. Setting up Arduino with the Computer to communicate serially

The Arduino is connected to the computer via USB. The arduino IDE is used to upload the code to the arduino. To receive and transfer data serially for our system, a SerailClass.h header file is written in C++ , which included functions like serialRead() and serialWrite() for reading and writing data serially. Using serialRead() function, the data from the sensor is written into a buffer. This distance is used as a trigger for our camera.

Code:

```
#ifndef SERIALCLASS_H_INCLUDED
#define SERIALCLASS_H_INCLUDED

#define ARDUINO_WAIT_TIME 2000

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>

class Serial
{
private:

 //Serial comm handler
  HANDLE hSerial;
  //Connection status
  bool connected;
//information about the connection
  COMSTAT status;
  DWORD errors;

public:
  //Initialize Serial communication with the given COM port
  Serial(const char *portName);
“   //Close the connection
  ~Serial();
  int ReadData(char *buffer, unsigned int nbChar);
  //Check the connection
  bool IsConnected();
};
#endif // SERIALCLASS_H_INCLUDED
```

C. Processing the image and comparing it with image database

The data set of facial images are given as input to the program using a csv file format, which specifies the path to the images along with a label denoting the person in the image. Initially these images are read from the csv file into a vector of matrices along with their labels. The captured image is read from the disk and the faces are detected in the image. This done by CascadeClassifier object using the “haarcascade_frontalface_alt.xml” provided by OpenCV.

On the other hand, we define a FaceRecognizer model, for our system we are using LBPHFaceRecognizer taking into account it's capability to recognize faces in variable illuminations with less data available when compared to Eigen faces model or Fisher faces model. Also that this model doesn't require it's database to be of equal sized images which is quite difficult for real time applications. The model is used to train the data set of images by passing the vector of matrices as argument. Now the detected face is then predicted against the trained data. The prediction is done against a gray scaled image, hence it is necessary to convert the captured image into a grayscale image.

When the match found, the label of match is returned as a value from the predict function. The predict function also gives the "confidence" with which the match has been made. A '0' confidence indicates a perfect match. Using this confidence as parameter , we can confirm the match and that an intruder has been identified.

```
Ptr<face::FaceRecognizer> model = face::createLBPHFaceRecognizer();
model->train(images, labels);
model->predict(face_resized, prediction, pred_confidence);
```

D. Notifying the authorities through email and start the alarm

When the confidence of the prediction is enough to confirm a match, the authorities must be notified regarding the intrusion. This is done by sending a mail to the registered email addresses. For this process, we wrote the code in PHP using external library called "PHPMailer" and called the PHP script for sending email from our C++ program using "system()". We went ahead with this one because, C++ doesn't have any native libraries for sending email using SMTP. Along with it, the arduino is given a signal to start alarm immediately, this is done by serial transfer to the board. A LED is used to denote the alarm.

V. TESTING PROPOSED SYSTEM

A. Module Testing

The system has been developed module by module and the working of each module have been tested and later integrated the modules to form the proposed system.

➢ **Programming proximity sensor with the Arduino board**

The code was uploaded using Arduino IDE and the arduino board started receiving data from the sensor. The sensor output was observed i.e, distance of the obstacles, using the serial monitor. The system was tested with various obstacles at various angles and distances. The results were good and this was then integrated with the image detection process.

➢ **Setting up arduino board with the computer**

The serialRead() function is used to capture the serial data from the arduino and stored it into a buffer. The functionality was tested by comparing the results displayed by the program with the serial monitor values of Arduino IDE. The results were matching and the serialRead() is working fine.

➢ **Capturing the image using Webcam**

The image capturing using webcam is done using C++ code. Initially the images were not captured due to some deprecated libraries. Later, the code was modified and the images were captured pretty good. The images captured were displayed in a window and also saved into the disk.

➢ **Processing the image and comparing it with pre defined database**

The AT&T database was used for faces initially to test the code. Using fisher faces we weren't able to use our own face database as it requires every image to be of equal size on the disk. The recognizer model was therefore modified to enable the use of our own data base. The module was tested and displayed the test image in a window with a rectangle around the face, denoting the working of face detector and the prediction and confidence values are used.

➢ **Sending notification to the administration**

The e-mailing code was written in php directly using php and then the script was called from the C++ program. The results were successful. The code was tested by integrating files and the code works fine.

Couple of modules each were developed at a time and then complete integration was done and the system was tested against some of the test cases. The results were satisfactory.

B. System Testing

Once the system is completely integrated, the whole was tested by designing the following test cases.
The test cases are executed and the results are noted.

VI. TEST SCENARIOS AND RESULTS:

The System is given a set of images which are assumed to be those of the intruders.

■ TEST SCENARIO 1:

1. Test Case Id : test001.
2. Description : No user approaches the system.
3. Scenario : The system will not be approached by any user. The system will be facing this scenario most of the time.
4. Input : -
5. Expected Output : No processing should be done.
6. Actual Output : No processing is Done

Remarks : PASS

■ TEST SCENARIO 2:

1. Test Case Id : test002.
2. Description : An assumed intruder is approached to the system.
3. Scenario : A person whose image is in the data set provided to the computer approaches the system(comes closer than 20cm).
4. Input : Image of the intruder.
5. Expected Output : Mail to the registered email id is sent regarding intrusion. Alarm is set on.
6. Actual Output : Mail is sent to the registered email id and alarm is set on

Remarks : Intrusion detected successfully. PASS.

■ TEST SCENARIO 3:

1. Test Case Id : test003.
2. Description : Intruder approaches the system, but stands at distance more than 50cm. Test for the proximity sensor.
3. Scenario : A person whose image is in the data set provided to the computer approaches the system (comes not closer than 20cm).
4. Input : -
5. Expected Output : No processing should be done.
6. Actual Output : No processing is done

Remarks : PASS

■ TEST SCENARIO 4:

1. Test Case Id : test004.
2. Description : A person with access into the area is approached to the system.
3. Scenario : A person whose image is not in the data set provided to the computer approaches the system (comes closer than 20cm) and has access to the restricted area.
4. Input : Image of the person.
5. Expected Output : No mail is sent to the registered email id.
6. Actual Output: No mail is sent to the registered email id.

Remarks : PASS

■ TEST SCENARIO 5:

7. Test Case Id : test005.
8. Description : A person with access into the area is approached to the system.
9. Scenario : A person whose image is not in the data set provided to the computer approaches the system (comes not closer than 20cm) and has access to the restricted area.
10. Input : -
11. Expected Output : No processing should be done
12. Actual Output: No processing is done.

Remarks : PASS

VII. CONCLUSION AND FUTURE DIRECTIONS

The primary objective of the project is to provide an efficient system in restricted areas for securing it from intrusion, and also sending reports over the internet to authorities not present locally. At the same time the memory overhead must be reduced by capturing only when any person approaches the system. With this objective we developed the system and tested its functionalities. All the modules are working and communicating as per requirement and thus the whole system is efficiently working and is up to the mark in performing its functionalities. The developed system has a lot of scope to get developed further with many other essential functionalities added. The following work can extend the system to a much better system :

- A database can be created to store the captured images. This can be useful as the number of images can be grown exponentially over days. To efficiently retrieve the data later, a database could be useful
- A single camera capturing the image of intruder may not give efficient results with 100 percent confidence. A set of cameras at different angles and places to capture a single person can be an effective solution. The network of cameras send images and all the images are then processed for better identification.

REFERENCES:

[1] Ahsan, M.M.; Li, Y.; Zhang, J.; Ahad, M.T.; Gupta, K.D. Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions. *Technologies* 2021, *9*, 31. https://doi.org/10.3390/technologies9020031

[2] Dar, I.G., Khan, A., Sharma, M. (2018). Feature Recognition of Face with Real-Time Variations Using Eigen Face Approach Methodology with PCA Algorithm. In: Singh, R., Choudhury, S., Gehlot, A. (eds) Intelligent Communication, Control and Devices. Advances in Intelligent Systems and Computing, vol 624. Springer, Singapore. https://doi.org/10.1007/978-981-10-5903-2_85I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[3] Wang, H., Li, P. & Zhang, T. Histogram feature-based Fisher linear discriminant for face detection. *Neural Comput & Applic* 17, 49–58 (2008). https://doi.org/10.1007/s00521-006-0081-7

[4] Mustamin Anggo and La Arapu 2018 *J. Phys.: Conf. Ser.* 1028 012119

[5] Zhao, X. and Wei, C. (2017) A Real-Time Face Recognition System Based on the Improved LBPH Algorithm. 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 4-6 August 2017,72-76. https://doi.org/10.1109/SIPROCESS.2017.8124508

[6] Gaddipati, M.S.S., Krishnaja, S., Gopan, A., Thayyil, A.G.A., Devan, A.S., Nair, A. (2021). Real-Time Human Intrusion Detection for Home Surveillance Based on IOT. In: Senjyu, T., Mahalle, P.N., Perumal, T., Joshi, A. (eds) Information and Communication Technology for Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies, vol 196. Springer, Singapore. https://doi.org/10.1007/978-981-15-7062-9_49