

**BHARATI VIDYAPEETH’S INSTITUE OF COMPUTER APPLICATIONS AND MANAGEMENT**

**NEW DELHI**

**FACE DETECTION SECURITY SYSTEM**

**(C# Programming)**

**Team Details:**

**Sanjna Katyal**

**(02135304419)**

**Abhijeet Kumar**

**(00435304419)**

**Synopsis**

**PROJECT TITLE:**

AUTOMATED FACE DETECTION AND RECOGNITION SECURITY SYSTEM

**PROBLEM STATEMENT:**

Today’s institutions are facing major security issues;

consequently, they need several specially trained personnel to

attain the desired security. These personnel, as human beings,

make mistakes that might affect the level of security.

A proposed solution to the aforementioned matter is a Face

Recognition Security System, which can detect intruders to

restricted or high-security areas, and help in minimizing human

error. This system is composed of two parts: hardware part and

software part. The hardware part consists of a camera, while the

software part consists of face-detection and face-recognition

algorithms software. When a person enters to the zone in

question, a series of snapshots are taken by the camera and sent

to the software to be analyzed and compared with an existing

database of trusted people. An alarm goes off if the user is not

recognized.

Today’s institutions are facing major security issues;

consequently, they need several specially trained personnel to

attain the desired security. These personnel, as human beings,

make mistakes that might affect the level of security.

A proposed solution to the aforementioned matter is a Face

Recognition Security System, which can detect intruders to

restricted or high-security areas, and help in minimizing human

error. This system is composed of two parts: hardware part and

software part. The hardware part consists of a camera, while the

software part consists of face-detection and face-recognition

algorithms software. When a person enters to the zone in

question, a series of snapshots are taken by the camera and sent

to the software to be analyzed and compared with an existing

database of trusted people. An alarm goes off if the user is not

recognized.

Today’s institutions (whether a huge corporate or a small education institution) are facing major security issues; consequently, they need several specially trained personnel to attain the desired security. These personnel, as human beings, make mistakes that might affect the level of security. Also, securing large premises and maintaining a keen eye on intruders on such large scales is prone to human error. A proposed solution to the aforementioned matter is an automated way of recognizing the differences between a legit and an intruder. A Face Recognition Security System, based on ML Algorithms, which can detect intruders to restricted or high-security areas can help in minimizing human error.

**IDEA BEHIND THE PROJECT:**

Although many existing solutions to the problem mentioned above have been developed and are under further advancements, this project was chosen by us to enhance our knowledge about how these things work in actual real-life scenario as well implementing the knowledge thus gained to produce a reasonable product that can do things in an innovative way and which can serve the purpose well.

**OBJECTIVE OF THE PROJECT:**

* To develop an automated solution using various machine learning and image recognition algorithms.
* It is non-intrusive and requires no physical interaction on user end. Thus saving a lot of time that is wasted on physical verifications of individuals.
* Keeping the cost of deployment as minimum as possible by enhancing the existing infrastructure present and incorporating our solution within it.

**SCOPE OF THE PROJECT:**

**(TARGET AUDIENCE)** The project solution thus developed can be implemented in various types of institutions like universities, business corporates, banks, and many more. With a little modification and enhancement, the project can serve multiple purposes, like:

* It can be used as a security and surveillance system at business corporates to allow restricted access to people at specified floors.
* It can be used as a surveillance system at various examination centers to reduce the cases where people with false identification may write papers on someone else behalf.
* The project solution can further be connected to the local authority networks to alert police and authorities in case of intrusion to places of restricted access.

Thus, the scope and target audience of our project solution is extensive and adaptable to situations as required.

**PROPOSED WORKING METHODOLOGY:**

This system is composed of two parts: hardware part and software part. The hardware part consists of a network of surveillance cameras, while the software part consists of face-recognition algorithms software. When a person enters to the zone in question, a series of snapshots are taken by the camera from the live footage and sent to the software to be analyzed and compared with an existing database of trusted people. If the person is in the system’s databases, he is identified and the screen tracks him while showing his identification above him. An alarm goes off if the user is not recognized and a marker is assigned by the system to the intruder that keeps a track of his/her movement via multiple live video feeds from the surveillance network.

**TECHNOLOGIES INVOLVED:**

C#, .NET Framework, OpenCV and EmguCV libraries, ML algorithms, etc.

**LIMITATIONS:**

* Poor Image Quality Limits Facial Recognition's Effectiveness.
* Small Image Sizes Make Facial Recognition More Difficult.
* Different Face Angles Can Throw Off Facial Recognition's Reliability.
* Data Processing and Storage Can Limit Facial Recognition Tech.

**PROJECT IMPACT:**

It will provide an automated and affordable solutions to small scale institutes while allowing them to reduce cost on security in long term as well making the surveillance effective.

**CONCLUSION:**

Face recognition systems are going to be used more and

more in the future for security reasons because they provide

better performance over other security systems.

Face recognition systems are going to be used more and

more in the future for security reasons because they provide

better performance over other security systems.

Face recognition systems are going to be used more and

more in the future for security reasons because they provide

better performance over other security systems.

Face recognition systems are going to be used more and more in the future for security reasons because they provide better performance over other security systems. At the end of completion of this project, we will have an automated solution for Facial Recognition and Security System. Real-time detection and recognition of people in a camera setup. A face detector integrated with a people tracker for higher accuracy and faster detections is proposed.

**Chapter 1**

**What is facial recognition?**

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time. Law enforcement may also use mobile devices to identify people during police stops. In the near future, face recognition technology will likely become more ubiquitous. But face recognition data can be prone to error, which can implicate people for crimes they haven’t committed.

**Who uses facial recognition?**

* Government for security purposes.
* Mobile phone companies
* Social media websites(facebook or instagram)
* Airlines at international departures

**How facial recognition is done?**

To identify the face of a family member, friend, or acquaintance you should be familiar with their facial features — their eyes, nose, mouth etc. That’s how a facial recognition system works. It stores that data and access it, for example in security systems.

**Step 1**. A picture of your face is captured from a photo or video. Your face might appear alone or in a crowd. Your image may show you looking straight ahead or nearly in profile.

**Step 2**. Facial recognition software reads the geometry of your face. Key factors include the distance between your eyes and the distance from forehead to chin. The software identifies facial landmarks — one system identifies 68 of them — that are key to distinguishing your face. The result: your facial signature.

**Step 3**. Your facial signature — a mathematical formula — is compared to a database of known faces.

**Step 4**. A determination is made. Your faceprint may match that of an image in a facial recognition system database else your face is registered as new entry.

**Methods of facial recognition**

**The Eigen Classifier**

The Eigen recogniser takes two variables. The 1st, is the number of components kept for this Principal Component Analysis. There’s no rule how many components that should be kept for good reconstruction capabilities. It is based on your input data, so experiment with the number. OpenCV documentation suggests keeping 80 components should almost always be sufficient. The 2nd variable is designed to be a prediction threshold; this variable contains the bug as any value above this is considered as an unknown. For the Fisher and LBHP this is how unknowns are classified however with the Eigen recogniser we must use the return distance to provide our own test for unknowns. In the Eigen recogniser the larger the value returned the closer to a match we have.

**The Fisher Classifier**

The Fisher recogniser takes two variables as with the Eigen constructor. The 1st, is the number of components kept Linear Discriminant Analysis with the Fisherfaces criterion. It’s useful to keep all components, this means the number of your training inputs. If you leave this at the default (0), set it to a value less than 0, or greater than the number of your training inputs, it will be set to the correct number (your training inputs - 1) automatically. The 2nd Variable is the threshold value for unknowns, if the resultant Eigen distance is above this value the Predict() method will return a -1 value indicating an unknown. This method works and the threshold is set to a default of 3500, change this to constrain how accurate you want results. If you change the value in the constructor the recogniser will need retraining.

**The Local Binary Pattern Histogram (LBPH) Classifier**

The LBPH recogniser unlike the other two takes five variables:

* radius – The radius used for building the Circular Local Binary Pattern.
* neighbors – The number of sample points to build a Circular Local Binary Pattern from. An value suggested by OpenCV Documentations is ‘8’ sample points. Keep in mind: the more sample points you include, the higher the computational cost.
* grid\_x – The number of cells in the horizontal direction, 8 is a common value used in publications. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.
* grid\_y – The number of cells in the vertical direction, 8 is a common value used in publications. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.
* threshold – The threshold applied in the prediction. If the distance to the nearest neighbour is larger than the threshold, this method returns -1.

The final variable, the threshold value works as it does in the Fisher method. If the Eigen Distance calculated is above this value the Predict() method will return a -1 value indicating an unknown. This method works and the threshold is set to a default of 100, change this to onstrain how accurate you want results. If you change the value in the constructor the recogniser will need retraining.

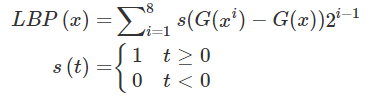
LBPH algorithm was commonly used for facial recognition. This algorithm is based on the local-binary-operator, broadly implemented in face recognition, due to its discriminating strength and calculation easiness.

Face recognition is performed by employing the Local Binary Pattern Algorithm. The LP operator is applied for local binary features by considering the Local Binary patterns which helps to shorten the local special features of the face image. The LBP is the binary ratio of pixels intensities within the centre pixel. And it's around eight pixels. The mathematical description is described in the below equation.



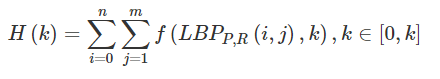
Center pixel is shown by tc and (pc, qc) represents the surrounded eight pixels, it is very though very useful to determine the face feature. In face matrix feature extracted from the image to compare the values with center pixel values to finally generate binary code.

The mathematical formulation of LBP operator is given by:

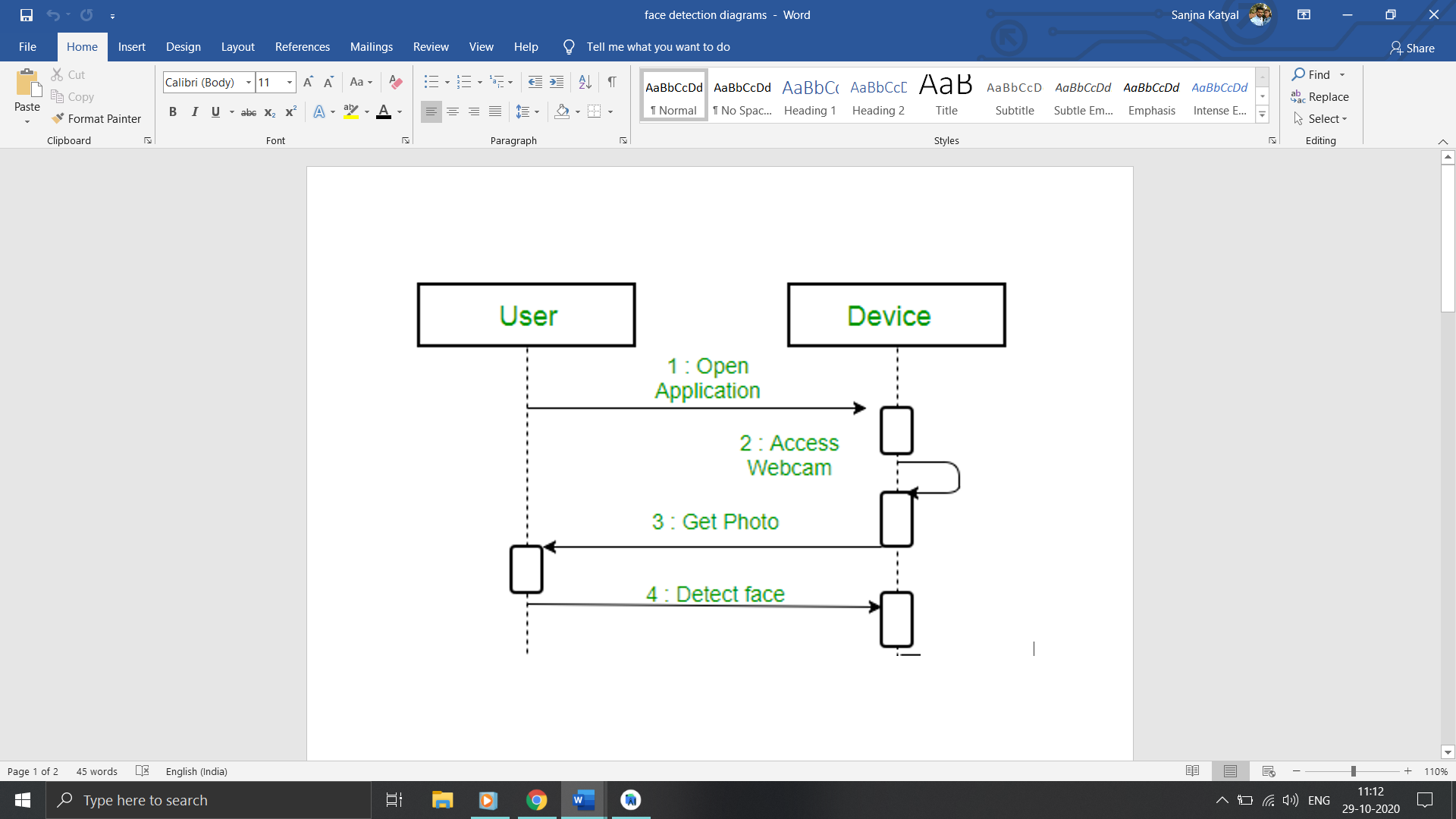


We purposed LBPH for image recognition and face detection in the surveillance camera in a specific area. Having obtained good results from various experimental analyzes of this technique, they also provide valid results for occlusion, pose variation, and illumination. Therefore, the proposed system allows recognition and recognition of faces in a controlled environment. As machine learning is very important nowadays, there are many areas where this work can be expanded. In implementing this project, we have identified some areas for improvement, such as Limitations of distance, the maturity of algorithms and camera qualities, even using DNN techniques. Accuracy can be improved in the future, which is more directly related to our work.

We used a modified LBP operator called uniform pattern. The pattern is the number of bitwise transitions from 1 to 0 or vice versa. The LBP is called uniform if its uniformity measure is at most 2. For example, the patterns 11111111 (0 transitions), 01111100 (2 transitions) and 11000111 (2 transitions) are uniform, while the patterns 10001000 (3 transitions) and 11010011 (4 transitions) are not. For dimension reduction, we used the histogram to reduce the image features from a 256-dimensional decimal to a 59- dimensional histogram, which contains information about the local patterns. The histogram uses a separate bin for each uniform pattern, and one separate bin for all non-uniform patterns. In the 8-bit binary number, we have 58 uniform patterns; therefore, we used 58 bins for them and one bin for all non-uniform patterns. The global description of the face image is obtained by concatenating all regional histograms. The overall value of LBPH can be represented in a histogram as:



where P is the sampling points and R is the radius.



**Limitations of Facial Recognition System**

* Security
* Mistaken identity
* Safety
* Basic freedom

**Using OpenCV and EmguCV files for Facial Recognition**

Training the Classifiers OpenCV enables the creation of XML files to store features extracted from datasets using the FaceRecognizer class. The stored images are imported, converted to grayscale and saved with IDs in two lists with same indexes. FaceRecognizer objects are created using face recogniser class. Each recogniser can take in parameters that are described below:

cv2.face.createEigenFaceRecognizer()

1. Takes in the number of components for the PCA for crating Eigenfaces. OpenCV documentation mentions 80 can provide satisfactory reconstruction capabilities.

2. Takes in the threshold in recognising faces. If the distance to the likeliest Eigenface is above this threshold, the function will return a -1, that can be used state the face is unrecognisable 6.

cv2.face.createFisherfaceRecognizer()

1. The first argument is the number of components for the LDA for the creation of Fisherfaces. OpenCV mentions it to be kept 0 if uncertain.
2. Similar to Eigenface threshold. -1 if the threshold is passed.

cv2.face.createLBPHFaceRecognizer()

1. The radius from the centre pixel to build the local binary pattern.

2. The Number of sample points to build the pattern. Having a considerable number will slow down the computer.

3. The Number of Cells to be created in X axis.

4. The number of cells to be created in Y axis.

5. A threshold value similar to Eigenface and Fisherface. if the threshold is passed the object will return -1.