Home Lock Authentication

with

Face Recognition

Lokesh Bejjagam

Reshmi Chakradhara

Department of Computer Science Engineering

Blekinge Institute of Technology

Karlskrona

Sweden

2021

Abstract

In today's generation, smart technology has become an integrated part of the world. The security of an individual's home or personal space is to be considered with utmost priority. With a smart lock system, we can design, develop security system for many purposes. This system is used to avoid data theft and illegal use of PIN or IDs by unauthorized users to gain access. The purpose of this system is to control the door lock access using advanced technology called face detection.

Why should we use face detection technology when we have PIN lock systems is of major concern. This major concern is to be solved which is the prime concern of our project.

PIN and pattern thefts by unauthorized users can lead to access breach which is highly risky and illegal. Protection of property is very essential. To provide more security to the digital home lock system, we come up with an advanced idea of a smart lock system with face detection which will provide high security to the users. Detecting face characteristics and using it along with PIN, patterns or biometrics will make it more precise and accurate and hence access cannot be breached. This security system is implemented with web cameras.

The project algorithm has been tested on the subjects successfully. Access through the door is given to the owners whose face is detected by the camera and is matched with the database.

**Keywords:**

Convolutional Neural Network, Facial Recognition, Facial Feature Extraction, Home Lock Security, Keyless entry

Acknowledgement

We would like to express our heartfelt gratitude to Dr. Wlodek Kulesza, Professor of our course, for his great flexibility on agreeing to make this project, for providing adequate infrastructure and a pleasant environment to complete our course, for providing invaluable guidance at every stage of this project work for which we are deeply grateful.

Finally, we would like to express our heartfelt appreciation and gratitude to our dear parents and members of our family for their moral support and heartfelt cooperation in carrying out the main project.

Reshmi Chakradhara  
Lokesh Bejjagam

Contents

Abstract 2

Acknowledgements 3

Contents 4

List of figures 5

List of acronyms 6

1 Chapter: Introduction 7

2 Chapter: Survey of related work 8

3 Chapter: Problem statement, objectives and main contribution 9

4 Chapter: Solution 10

4.1 Modeling 10

4.2 Implementation or Application 11

4.3 Validation 15

5 Chapter: Conclusion and future work 16

Reference 17

List of figures

[Fig 1 Block diagram of the Home Lock Authentication System 10](file:///C:\Users\lokesh\Downloads\Template%20_Milestone%203_ht21%20(3)%20(1).docx#_Toc87115784)

[Fig 2 Importing the Libraries 11](#_Toc87115785)

[Fig 3 Loading the Images 11](#_Toc87115786)

[Fig 4 Finding the location of the faces 12](#_Toc87115787)

[Fig 5 Comparing the faces 12](#_Toc87115788)

[Fig 6 Importing Images 13](#_Toc87115789)

[Fig 7 Computing the Encoded Image 13](#_Toc87115790)

[Fig 8 Detecting an Image through webcam 13](#_Toc87115791)

[Fig 9 Encoding the webcam Image 14](#_Toc87115792)

[Fig 10 Finding the Image match 14](#_Toc87115793)

[Fig 11 The Verification of webcam along with the algorithm 15](#_Toc87115794)

List of acronyms

|  |  |
| --- | --- |
| **Acronym** | **Unfolding** |
| LBPH | Local Binary Patterns Histograms |
| SURF | Speed Up Robust Features |
| SIFT | Scale Invariant Feature Transform |
| CNN | Convolutional Neural Network |

# Chapter: Introduction

The project is developed to increase the security level called as " Home Lock Authentication with Face Recognition”. It is designed in such way to open the door using a web camera and to give access only to authorized person. This effective system provides access control to the door and security system which is based on a face recognition pattern [1].

Face recognition based on geometric features of a face is most likely the most intuitive method of face recognition. A feature vector (distance between the points, angle between them, etc.) was constructed using marker points (position of eyes, ears, nose, etc.). The Euclidean distance between the feature vectors of a probe and reference image was calculated to perform the recognition. Because of the nature of the method, it is resistant to changes in illumination.

Facial biometrics [2] systems have been used as a security measure in the most prestigious institutions and workplaces to ensure that there is no possibility of vandalism. This type of software eliminates all room for human error and is a huge help. The software performs geometric and photometric recognition in seconds using only a set of algorithms.

Because of its ease of use and low cost, these facial biometrics system has emerged as the king of all recognition software. Its non-contact nature is its best feature, as it can recognize a person using facial recognition even in a crowded environment, given that the images are saved in the database.

# Chapter: Survey of related work

According to the statistics from Statistica website, there were 7,369 reported burglaries against people in Sweden in 2020. As we see the number of cases has been increasing from the past few years, there is an urge to increase the security measures with utmost priority. Thus, our project has taken this as its prior motive and moved ahead with this thought of face recognition as a security measure [3].

Nearly 60 million Americans have been affected by identity theft, according to a 2018 online survey by The Harris Poll. That same survey indicates nearly 15 million consumers experienced identity theft in 2017. While there are many stories of identity theft in the news, what we tend to hear more about are data breaches—in which a company or other organization’s customer’s records, which may include full names, Social Security numbers, and other personal information, are accessed fraudulently. In 2017, the Identity Theft Resource Centre counted a new record high of 1,579 data breaches, exposing more than 178 million records. The big one—involving Equifax, one of the three major credit reporting agencies—received a lot of attention. Not only was the number of potential victims quite large at 147.9 million, but the kind of information also exposed was significant. It included names, Social Security numbers, birth dates, addresses and, in some instances, driver’s license numbers.

Our main motive is to provide security to home doors with the simplest mechanism possible at our disposal. In this regard, we have studied different algorithms for face recognition such as Eigen faces, Local Binary Patterns Histograms (LBPH), Fisher faces, Scale Invariant Feature Transform (SIFT), Speed Up Robust Features (SURF).[4] Considering better computational capacity and short runtimes, we chose to implement the algorithm using Convolutional Neural Networks (CNN).

# Chapter: Problem statement, objectives, and main contribution

Today, we are living in an era where security is of major concern which plays a vital role in maintaining the safety of every individual’s property and personal life. This security can be implemented in many ways which varies based on the latest technology. PIN lock system, RFID tags and passwords are one way of implementing the security, but they can be easily shoulder surfed and it paves a way to gain unauthorized access to people.

The main objective of our project is to increase the security of our home doors using face recognition system. Automatic face recognition is all about extracting meaningful features (face edges, nose, eyes, lips, eyebrows) from an image, putting them into a useful representation and performing classification on them.

The main contribution of the project is face recognition. While humans can easily recognize faces, facial recognition is a difficult pattern recognition problem in computing. Based on a two-dimensional image, facial recognition systems attempt to identify a human face, which is three-dimensional and changes appearance depending on lighting and facial expression. Facial recognition systems go through four steps to complete this computational task. Face detection is used first to separate the face from the image background. The segmented face image is aligned in the second step to account for face position, image size, and photographic properties such as illumination and grayscale. The alignment process's goal is to allow the accurate localization of facial features. In the third step, facial feature extraction, accurate localization of facial features is required. To represent the face, features such as the eyes, nose, and mouth are pinpointed and measured in the image. The face's established feature vector is then matched against a database of faces in the fourth step [5].

# Chapter: Solution

## Diagram Description automatically generatedModeling

Fig 1 Block diagram of the Home Lock Authentication System

Three basic steps are used to develop a robust face recognition system: (1) face detection, (2) feature extraction, and (3) face recognition. The face detection step is used to detect and locate the human face image obtained by the system. The feature extraction step is employed to extract the feature vectors for any human face located in the first step. Finally, the face recognition step includes the features extracted from the human face to compare it with all template face databases to decide the human face identity [6].

*Face Detection*: The face recognition system begins first with the localization of the human faces in a particular image. The purpose of this step is to determine if the input image contains human faces or not. The variations of illumination and facial expression can prevent proper face detection. To facilitate the design of a further face recognition system and make it more robust, pre-processing steps are performed. Also, the face detection step can be used for video and image classification, object detection, region-of-interest detection, and so on.

*Feature Extraction*: The main function of this step is to extract the features of the face images detected in the detection step. This step represents a face with a set of features vector called a “signature” that describes the prominent features of the face image such as mouth, nose, and eyes with their geometry distribution. Each face is characterized by its structure, size, and shape, which allow it to be identified.

*Face Recognition*: This step considers the features extracted from the background during the feature extraction step and compares it with known faces stored in a specific database. There are two general applications of face recognition, one is called identification and another one is called verification. During the identification step, a test face is compared with a set of faces aiming to find the most likely match. During the identification step, a test face is compared with a known face in the database to make the acceptance or rejection decision.

## Implementation

The steps for implementing the algorithm are as follows:

*Installations:*

First, we must obtain a C++ compiler. This can be accomplished by installing Visual Studio. We will install the necessary packages here. The list is provided below.

cmake

dlib

face recognition

numpy

opencv-python

*Face Recognition* :First we will import the relevant libraries. The Process of Recognition can be divided into 3 steps.



Fig 2 Importing the Libraries

*Loading Images and Converting to RGB*: The Face Recognition package consists of a load image function that loads the image. Once imported the image must be converted to RGB.

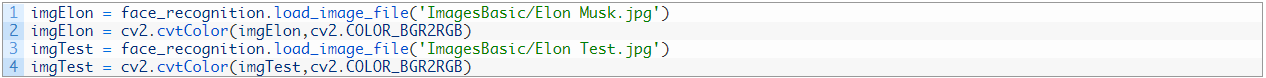


Fig 3 Loading the Images

*Find Faces Locations and Encodings*: In this step we will use the true functionality of the face recognition library. First, we will find the face in the image.[7]This is done using HOG (Histogram of Oriented Gradients) at the backend. Once we have the face, they are warped to remove unwanted rotations. Then the image is feed to a pretrained neural network that out puts 128 measurements that are unique to that face. The parts that the model measures are not known as this is what the model learns by itself when it was trained. Once we have the face locations and the encodings, we can draw rectangles around our faces.

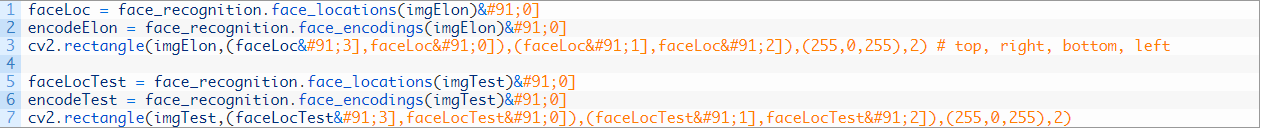


Fig 4 Finding the location of the faces

*Compare Faces and Find Distance:*

Once we have the encodings for both faces, then we can compare these 128 measurements of these two faces to find similarities. To compare the package uses one of the most common machine learning methods linear SVM classifier. We can use the compare faces () function to find if the faces match. This function returns True or False. Similarly, we can use the face distance () function to find how likely are the faces match in terms of numbers. The lower the distance the better the match.



Fig 5 Comparing the faces

*Importing Images*

As we have imported before we can use the same load\_image\_file () function to import our images. But when we have multiple images, importing them individually can become messy. Therefore, we will write a script to import all images in each folder at once. For this we will need the os library so we will import that first. We will store all the images in one list.  
 Graphical user interface, text, application

Description automatically generated

Fig 6 Importing Images

*Compute Encodings*

Now that we have a list of images, we can iterate through those and create a corresponding encoded list for known faces. To do this we will create a function. As earlier we will first convert it into RGB and then find its encoding using the face encodings () function. Then we will append each encoding to our list.[8]

Graphical user interface, text, application

Description automatically generated

Fig 7 Computing the Encoded Image

*The While loop*

The while loop is created to run the webcam. But before the while loop, we must create a video capture object so that we can grab frames from the webcam.

*Webcam Image*

First, we will read the image from the webcam and then resize it to quarter the size. This is done to increase the speed of the system. Even though the image being used is 1/4th of the original, we will still use the original size while displaying. Next, we will convert it to RGB.



Fig 8 Detecting an Image through webcam

*Webcam Encodings*

Once we have the webcam frame, we will find all the faces in our image. The face location’s function is used for this purpose. Later we will find the face encodings as well.



Fig 9 Encoding the webcam Image

*Find Matches*

Now we can match the current face encodings to our known faces encoding list to find the matches. We will also compute the distance. This is done to find the best match in case more than one face is detected at a time. Once we have the list of face distances, we can find the minimum one, as this would be the best match. Now based on the index value we can determine the name of the person and display it on the original Image. [9]





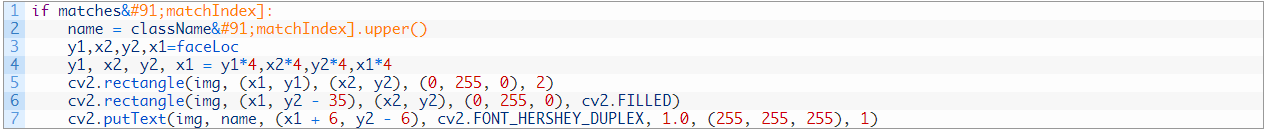


Fig 10 Finding the Image match

*Software*: This algorithm is implemented using Intel core i7 processor with a web camera resolution of still image of 0.92 megapixel and video upto 1280 x 720 (HD) at 30 fps.[10]

*Hardware*: The hardware used for implementation is our personal laptop

*Color:* It can be implemented on both variations: black and white as well as color scale.

*No of subjects*: We have implemented the algorithm on 2 subjects.

## Validation

The tests on the subject have been done and the final output shows the detected face after running the algorithm successfully. The owner of the detected face will be given access through the door.

A screenshot of a person

Description automatically generated with medium confidence

Fig 11 The Verification of webcam along with the algorithm

# Chapter: Conclusion and future work

The system is self-sufficient and can decide whether to allow the visitor. Only the known visitor has permission from the owner. This smart lock can help with security management. The system controls the main entrance access workplace. It can help you save time by notifying and managing the user. Customers and employees using this system can avoid unwanted visits and confrontation.

Future works have many possible leads as we can develop the technique of retinal scan for more accurate security and aging factor. One can also increase the scope of the project to an industrial level providing security to design rooms, development rooms etc on a large scale. Owner can receive alerts via mobile in the future when there’s an access breach which is a safety measure.

Reference

[1] N. A. Othman and I. Aydin, “A face recognition method in the Internet of Things for security applications in smart homes and cities,” in *2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, Apr. 2018, pp. 20–24. doi: 10.1109/SGCF.2018.8408934.

[2] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, “Middlewares for Smart Objects and Smart Environments: Overview and Comparison,” in *Internet of Things Based on Smart Objects: Technology, Middleware and Applications*, G. Fortino and P. Trunfio, Eds. Cham: Springer International Publishing, 2014, pp. 1–27. doi: 10.1007/978-3-319-00491-4\_1.

[3] “An empirical examination of consumer adoption of Internet of Things services: Network externalities and concern for information privacy perspectives - ScienceDirect.” https://www.sciencedirect.com/science/article/abs/pii/S0747563216302990 (accessed Nov. 06, 2021).

[4] Q. Yin, X. Tang, and J. Sun, “An associate-predict model for face recognition,” in *CVPR 2011*, Jun. 2011, pp. 497–504. doi: 10.1109/CVPR.2011.5995494.

[5] T. Ahonen, A. Hadid, and M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.

[6] L. Wiskott, N. Krüger, N. Kuiger, and C. von der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, Jul. 1997, doi: 10.1109/34.598235.

[7] 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, Jun. 2010, doi: 10.1109/TIP.2010.2042645.

[8] Z. Cao, Q. Yin, X. Tang, and J. Sun, “Face recognition with learning-based descriptor,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 2707–2714. doi: 10.1109/CVPR.2010.5539992.

[9] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith, “Fully automatic pose-invariant face recognition via 3D pose normalization,” in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 937–944. doi: 10.1109/ICCV.2011.6126336.

[10] M. K. Müller, M. Tremer, C. Bodenstein, and R. P. Würtz, “Learning invariant face recognition from examples,” *Neural Networks*, vol. 41, pp. 137–146, May 2013, doi: 10.1016/j.neunet.2012.07.006.