Bachelor of Science in Computer Science & Engineering

Smart Surveillance System with Facial Recognition Using IoT

by

Md. Saydul Kader

ID: 1504090

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

20th May, 2021

Smart Surveillance System with Facial Recognition Using IoT

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Md. Saydul Kader

ID: 1504090

Supervised by

Dr. Asaduzzaman

Professor

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

The thesis titled '**Smart Surveillance System with Facial Recognition Using IoT'** submitted by ID: 1504090, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Dr. Asaduzzaman
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md. Mokammel Hoque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Md. Mokammel Hoque
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

The journey of writing this thesis towards its completion has only been possible though all the support and supervision from many individuals. This has been a fulfilling experience, both personally and professionally.

I would like to express my heartfelt gratitude towards my supervisor Dr. Asaduzzaman by whose guidance this thesis becomes a reality. I am thankful for his continuous guidance and critical questions that has encouraged me to think out of my comfort zone and has levered my capabilities to take challenges that I never thought could be done by me.

I am thankful to the panel members for providing the approval of my work. I am grateful to my family members for their patience in bearing with me throughout this whole time. Their unconditional love and encouragement has provided support in every aspect of my life.

Abstract

Protecting our homes and offices from invaders has always been a key priority in our everyday lives, and introducing technology into this element, like anything else, will enhance our current security measures. In this project a smart surveillance system with facial recognition is developed using IoT devices like Raspberry Pi. The system consists of Raspberry Pi, Pi camera module and android operating system. For the face detection process, the RPi uses the Haar Cascade algorithm to extract facial features from provided images. Using a machine learning technique and the Eigenfaces algorithm, the program is learned to identify the features of a face from a trained model already fed to the device. For facial recognition, the device uses the OpenCV library to equate the new image with the facial dataset already stored on the system. The result of the face recognition is fed to the alert system. The alert system consists of an android app and alert notification via email. The system generates an email alert when any unrecognized person is detected. The android app has features like real time video streaming, previous activity monitoring to monitor camera premises. One of the problems with today's surveillance systems is the amount of storage required. Our designed system attempts to address the storage problem by only saving photos when a human face is identified. In addition, the system applies a filter to the images and compresses them to ensure faster storing of images in the database. Another major challenge encountered throughout the project's development is ensuring rapid transmission of data between the RPi and the database. The more time required in the data transmission process the more vulnerable the system becomes.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The concept of facial recognition has been widely researched and developed since its conception in the 1960s, pioneered by Woody Bledsoe, Helen Chan Wolf, and Charles Bisson. The applications of facial recognition are abundant but its employment in the areas of biometric security has been emphasized on the most [1]. When it comes to human faces, Facial Recognition has always been one of the most interesting and intriguing innovations. For this thesis project, we have tried to take this large-scale system and compress it into a user and cost friendly system to be used in corporate offices or in household. This concept also has applications in other areas besides security and surveillance. This very system can be used to register attendance in classrooms or can also replace or assist in the use of ID cards for identification in universities or office spaces. If applied properly and used correctly, this design may prove to be an easy and cost friendly technological update or addition to similar existing systems. The overview of the proposed smart surveillance system and also the challenges encountered in the process, will be discussed in this chapter. The chapter also discusses the motivation and contribution of this specific thesis.

1.2 Face Detection and Recognition with Feedback System

A facial recognition system is a technology that can compare a human face from a visual image or a video frame against a database of faces. It operates by pinpointing and calculating facial features from a given image, and is commonly

used to authenticate users via ID authentication services. [1] The major task of face recognition process starts with a comparatively difficult task which is face identification. Face detection process faces several challenges due to different circumstances. So, the major concern before performing recognition process is to gain accuracy in face detection process. After the detection and recognition of faces then the challenge comes to give feedback to the administrator of the system in an efficient and user friendly manner. For gaining efficiency in facial recognition and providing a complete solution to security concern we have proposed a smart surveillance system using IoT device. It can be proved to be a cost and user friendly solution in this concern. The main flow of work of the proposed smart surveillance system can be discussed as-

1. Face identification process is performed by cascading two classifiers to gain more accuracy.
2. Face recognition process is performed by using the most efficient recognizer method createEigenRecognizer() from OpenCV.
3. An alert or feedback system is developed along with an integrated android app to provide realtime feedback to the system administrator.

The block diagram of the smart surveillance system with facial recognition is shown as follows:

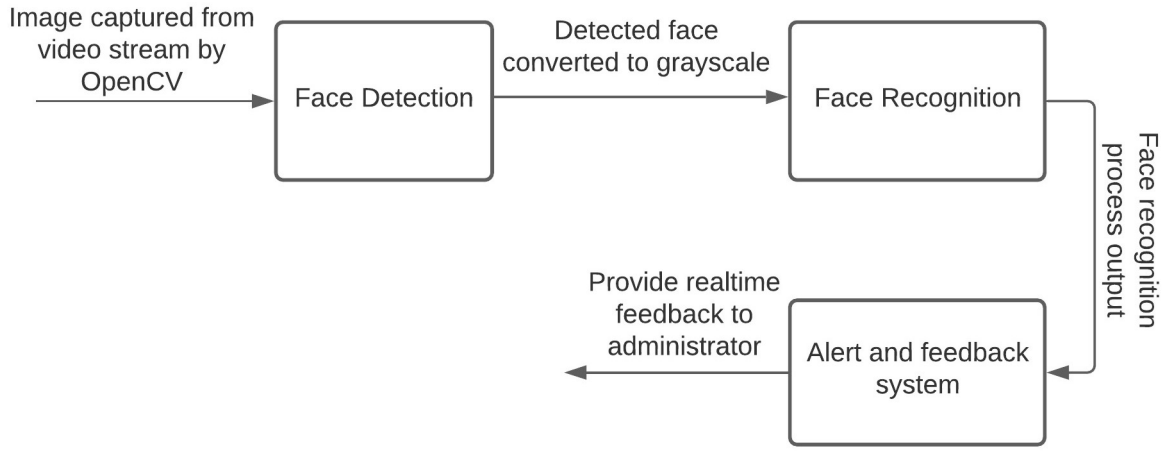

Figure 1.1: Block diagram of the smart surveillance system

1.3 Difficulties

Growing commercial interest in face recognition is promising, but it also proves to be a difficult endeavor when it comes to problems associated with it, which have continuously hampered its standard of execution. These difficulties occur when conditions are uncooperative, resulting in the various facial appearances and expressions. The major challenges in detecting and recognizing a face are enlisted as follows:

1. Light variations: A minor alteration in lighting conditions poses a major obstacle to automatic face recognition and can have a significant effect on its performance.
2. Pose: Pose patterns are very sensitive to facial recognition systems. When a person's head movement and viewing angle change, the posture of his or her face changes. For this results of recognition process can be changed.
3. Occlusion: It means blockage of one or many parts of the face. Thus whole face is not available as an input image. Occlusion is one of the most difficult problems of facial recognition systems.
4. Low resolution: Low quality cameras like CCTV cameras in streets, ATM cameras, supermarket security cameras capture low resolution images. A low resolution image doesn't provide much information as most of them are lost. It can be a big challenge in the process of recognizing the faces.

1.4 Applications

Facial recognition is indeed an area of bio-metrics where several technologies meet. As investments in face recognition technologies increase and technologies mature, we see facial recognition usage increasing in some use cases, including new ones. Also, The facial recognition technology market gets driven by the COVID-19 pandemic. Facial recognition systems can contribute to many aspects of our daily lives. Such as:

1. Increasing home and office security.
2. Preventing retail crime by identifying the intruder.

3. It can be used to find missing children and victims of human trafficking.
4. Diagnosing diseases that cause detectable changes in appearance.

1.5 Motivation

The motivation of working on this project originated from the urge to increase the safety of the citizens of this country by using available technology. Additionally, there is always room for improvement in security and surveillance.
Surveillance systems using CCTV cameras for general purposes simply record the camera's surroundings. Generally a real-time feedback system is not implemented. Exiting systems with real-time feedback costs a significant amount. This also pushes for a cost effective, user-friendly surveillance system with smartphone support.
The key objectives during the development of this project can be enlisted as follows:

1. To detect a human face in camera surroundings with more accuracy.
2. To recognize a human face by matching it with stored datasets for controlling the entry of unknown persons.
3. To develop a user friendly and efficient feedback system along with an integrated android app to provide realtime security update to the system administrator.

1.6 Contribution of the Thesis

Thesis or Research work is performed, to achieve a specific set of goals, whether it is to define a new methodology or to improve the existing ones. In this thesis, the main focus was given to improve the face detection accuracy and to develop an efficient and user friendly android app to be used as the feedback system. The primary contribution of this thesis is the following:

1. Development of a cost effective facial recognition based surveillance system consist of multiple cameras with Raspberry Pi and Pi camera module.
2. Development of an android app that is capable of monitoring multiple cameras surroundings in real-time and which provides features such as real time

video streaming, previous activity monitoring by storing images of human face with specific time.

1.7 Thesis Organization

The rest of this thesis report is organized as follows:

i. Chapter 2 gives a brief summary of previous research works in the field of face detection and recognition system.

ii. Chapter 3 describes the proposed methodology for the smart surveillance system with facial recognition feature using IoT devices. In face detection process we have cascaded two classifiers named Haar cascade classifiers and Bayesian classifiers to improve face detection accuracy. For face recognition purposes the most efficient recognizer method of OpenCV is used.

iii. Chapter 4 provides the description of the working dataset and analysis of the performance measure for the proposed method.

iv. Chapter 5 contains the overall summary of this thesis work and provides some future recommendations as well.

1.8 Conclusion

In this chapter, an overview of is provided. Along with the difficulties, the summary of the smart surveillance system with facial recognition is described in this chapter. The motivation behind this work and contributions are also stated here. In the next chapter, background and present state of the problem will be provided.

Chapter 2

Literature Review

2.1 Introduction

There are a lot of contribution in facial recognition based surveillance system in recent years. Various kinds of system have built up on face detection and recognition concept. Though there are a few works where facial recognition works efficiently with an elaborate feedback system for the purpose of security. Now most of the IoT based applications have android app support. So it is also necessary to modernize the system by integrating a user friendly and efficient android app in the system.

Facial recognition process is divided into two major part which is face detection and face recognition. A detailed description of both these parts is provided in this chapter.

2.2 Related Literature Review on Facial Recognition Based Surveillance System

As compared to voice, fingerprint, iris, retina eye scan, and hand geometry, face recognition is an effective technique and one of the most common biometric methods for identifying and verifying individuals. This has necessitated the creation of many facial recognition methods over the years, making it one of the most researched areas of computer vision. One of the key reasons it is still a fast-growing research area is its use in unconstrained settings, where most current strategies do not work optimally.

2.2.1 Face Detection and Recognition Process:

In Goyal et al. [2] different algorithms of face detection were compared. As the face recognition accuracy mostly depends upon the results of face detection process, choosing the right algorithm is very much important. In this work, the author has compared the accuracy of three different face detection algorithm. They are Finding via motion, Cam shift algorithm and Haar Cascade. Among them, Haar Cascade algorithm's efficiency and performance is much better than the other two. Also, this work compares the benefits of using MATLAB and OpenCV for face detection purposes. OpenCV proved to be the better solution than MATLAB based on execution time.

Khan et al. [3] has proposed a face recognition sytem with OpenCV with some specific algorithms. For the face detection process they have used Haar cascade classifiers. For face recognition process they have implemented three different recognizer method from OpenCV. They are createEignenFaceRecognizer(), createFisherFaceRecognizer() and createLBPHFaceRecognizer().

Mrs. Madhuram et al. [4] has used OpenCV computer vision library for developing a face recognition system. They also used Haar cascade classifiers for detecting face from a video stream. In this work they have proposed a method for improving the accuracy of the recognition process during the low light condition. They have used a preprocessing filter to do so. By this filter an image is converted into a grayscale image and then histogram equalization is applied. It is a very simple method of automatically standardizing the brightness and contrast of facial image.

S. V. Tathe et al. [5] discussed about the implementation of face recognition and tracking in videos. In this work, they have compared two different feature extraction method, Gabor filter and Eigen method. Their work shows that the Eigen method provides more accuracy in recognition process.

Face detection is a fundamental step of all facial analysis algorithms including face recognition. Chen et al. [6] discussed some techniques used for the purpose of detecting face. In this work they used feature extraction methods such as cascading, hard example mining, and the use of loss functions. They also have discussed the evaluation matrix ROC curve which is widely used to measure the

performance of the face detector.

D. Gaikawad et al. [7] introduced a new algorithm which can effectively solve the problem of large scale video face recognition. Their proposed work is capable of recognizing face rapidly in real time. Also the proposed system is robust to pose, illumination and shape. The algorithm also provides a way to normalize the noise on a detected face image.

Facial recognition is a major challenge in the field of computer vision. A. M. Jagtap et al. [8] implemented three different facial recognition algorithms such as LBPH, EigenFace and FisherFace. In this work, they have compared these three algorithms with the same facial datasets. Their finding shows that the LBPH recognizer performs best when the dataset consists of more than 50 images. On the other hand, Eigenface and FisherFace show almost the same accuracy in every occasion.

2.2.2 Facial Recognition Based Surveillance System

I. Gupta et al. [9] developed a system with raspberry pi to perform facial detection and recognition. In this work, OpenCV has used for the purpose of recognition process. For detection process HAAR Cascade classifier is used. Their proposed system used Eigen face approach for recognition process. In the feedback system they have used a DC motor which rotates clockwise or anticlockwise according to the result.

A. Nandurkar et al. [10] have done a survey on internet of things approach for surveillance system. In this work, the methodology of a face recognition system is explained briefly. Also they implemented a simple feedback system using GSM module. However, there were no discussion on the accuracy of recognition process in different challenging situations like low lighting conditions.

N. Sugumaran et al. [11] implemented a system that works on motion detection and face recognition but there is no direct approach for face recognition and it is also not efficient as it can't detect face from live video streaming. There is an alert system that sends email to the authorized person. But there is no android app implementation.

K. Kusnandar et al. [12] developed a prototype of a face recognition based access control system. For face recognition purposes they used the OpenCV library. For face detection purposes they used the Haar Cascade Classifier. The system consists of a solenoid door lock and an integrated android app. The android has a manual control system for door lock but the system lacks automatic access after successful recognition process. Also it doesn't have any live monitoring feature. The importance of home security is increasing rapidly in recent years. The scope of it has become even bigger by integrating smartphones in this. D. Sri et al. [13] introduced a face recognition based security system which also has a smartphone based feedback system. The system can capture the image of any detected face and upload it to a telegram bot. But it doesn't provide any feature like real time video streaming and monitoring previous data.

2.3 Conclusion

A detailed description of the literature review is provided in this chapter. The discussion was divided into two parts depending on the procedure of facial recognition based surveillance system. The different strategies and methods used by the researchers were described here. The next chapter contains an explanation of the methodology of the smart surveillance system with facial recognition in detail.

Chapter 3

Methodology

3.1 Introduction

Face recognition is a technology that can identify or verify a subject based on an image, video, or other audiovisual element of his face. This identification is typically used to gain access to an application, system, or service. It is a biometric identification method that uses body measurements, in this case the face and head, to verify a person's identity through its facial biometric pattern. The objective of face recognition is, from the incoming image, to find a series of data of the same face in a set of training images in a database. The great difficulty is ensuring that this process is carried out in real-time. Facial recognition is dependent upon face tracking. After identifying the face, then the system would start the recognition process. The first challenge is to identify the face from an image dataset. To overcome the challenge Haar-cascade classifier is used in this project. For the recognition process we have used OpenCV computer vision library. The various steps of proposed methodology are described in this chapter in detail.

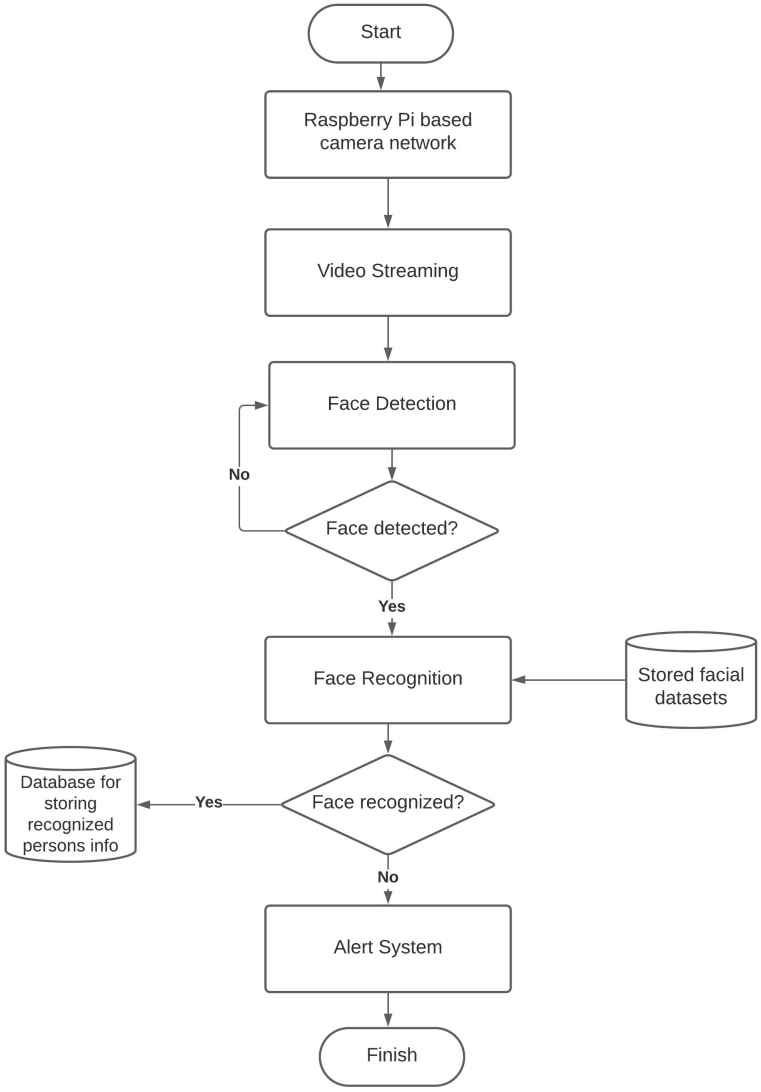

Figure 3.1: Methodology of the proposed surveillance system

3.2 Steps of Proposed Smart Surveillance System

Figure 3.1 shows the steps of proposed smart surveillance system that helps us to recognize a human face and alert the system administrator when an unknown face is detected. In the processing step, video streaming from the Pi camera module is taken as input. OpenCV reads the camera stream frame by frame in real time. In every frame OpenCV tries to detect a face by the Haar Cascade object detection algorithm. If a face is detected in the camera stream, then the system will try to match the face with the existing facial dataset in raspberry pi. If a detected face is not matched with the existing dataset then the system will generate an

warning message to the administrator by email. The system administrator can monitor the live streaming of the pi camera by an android app. Also he can see the previous activity log of the camera premises in the app.

3.3 Raspberry Pi Based Camera Network

The Pi camera module is a small, lightweight camera that works with the Raspberry Pi. It connects with the Raspberry Pi through the MIPI camera serial interface protocol. It is most commonly utilized in image processing, machine learning, and surveillance projects. Because the payload of the camera is so small, it is widely employed in surveillance drones also. Apart from these modules, Pi may also utilize standard USB cameras that are used in a PC.
The camera network consists of two Raspberry Pi camera modules. The purpose of this camera network is to provide real-time video stream to the OpenCV library. The Pi camera is integrated in Raspberry Pi via CSI port. It also provides real time video streaming feature for the android app.

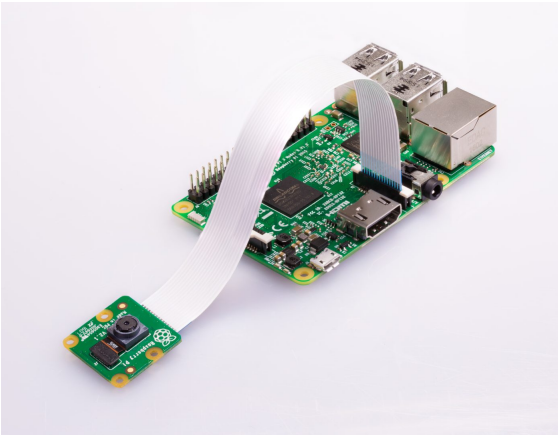

Figure 3.2: Pi camera installed in Raspberry Pi

3.4 OpenCV Computer Vision Library

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed

product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects etc. [14]

In this project we have used the OpenCV library for detecting and identifying faces. The version that is used in this project is 3.1.0.

3.5 Face Detection

The very first process in this system is face detection. The camera module is to detect a face from a running feed and capture a photograph in that instant. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images which is then used to detect objects in other images. For detecting a face, the program needs images with faces in them (positive images) and images without faces (negative images). The program then extracts the features from them. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle. [15]

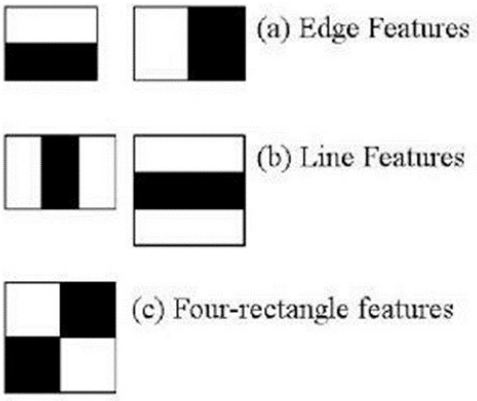

Figure 3.3: Extractable features

A multitude of attributes are determined by utilizing the various sizes and locations of each kernel. This is accomplished by employing integral pictures, which reduce the pixel computation to a four-pixel procedure. Not all of the

many attributes that may be estimated are useful for detecting objects (faces) in every scenario. The collection's most important features must be extracted. This is accomplished by using all of the characteristics on all of the training examples. The attributes with the lowest error rate are picked, implying that they best classify the face and non-facial pictures. Even after additional categorization, the ultimate number of characteristics required to detect a face in an image is still quite huge, which prolongs the procedure. However, because the majority of an image's region is non-face region, a solution to this problem is conceivable. The notion of Cascade of Classifiers has been proposed to help with this. The characteristics are organized into distinct stages of classifiers and applied to the window one by one. If the first stage fails, the window is discarded completely. Its remaining attributes are not taken into account. If it passes, the procedure is completed and the next stages of features are applied. A face region is a window that travels through all stages. The following block diagram simplifies the procedures to perform this procedure in OpenCV:

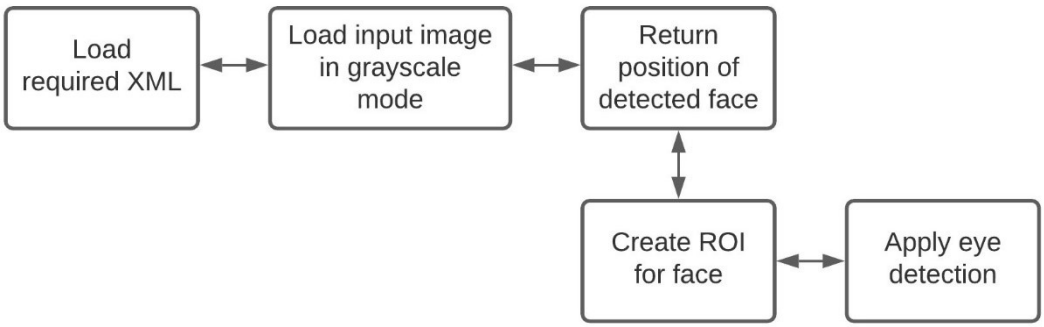

Figure 3.4: HAAR Cascade process

Once a face region is identified, the picture is instantly sent on to the next phase of the process to determine whether or not the discovered face matches a stored facial dataset.

3.6 Face Recognition

Facial Recognition is a computer application or process which is capable of identifying a person from an image or video feed. This is achieved by cross examining a given picture with an already available image on a database. The FaceRecognizer class for face recognition that comes with OpenCV includes the following

algorithms:

- Eigenfaces (createEigenFaceRecognizer())
- Fisherfaces (createFisherFaceRecognizer())
- Local Binary Patterns Histograms (createLBPHFaceRecognizer())

[16] The algorithm used for this system is the Eigenfaces method. The Eigenfaces method described took 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. The lower-dimensional subspace is found with Principal Component Analysis, which identifies the axes with maximum variance. While this kind of transformation is optimal from a reconstruction standpoint, it doesn't take any class labels into account. Imagine a situation where the variance is generated from external sources, let it be light. The axes with maximum variance do not necessarily contain any discriminative information at all, hence a classification becomes impossible. So, a class-specific projection with a Linear Discriminant Analysis was applied to face recognition. The basic idea is to minimize the variance within a class, while maximizing the variance between the classes at the same time. [16]

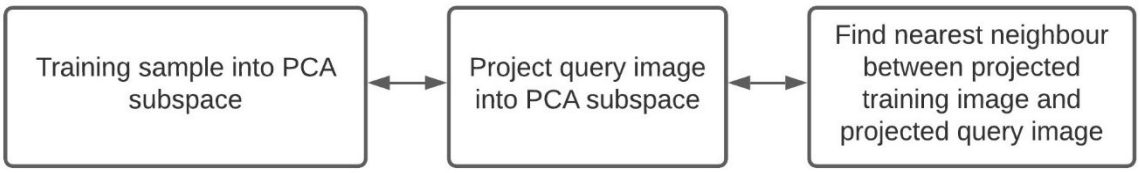

Figure 3.5: Eigenface process

Eigenface in OpenCV:

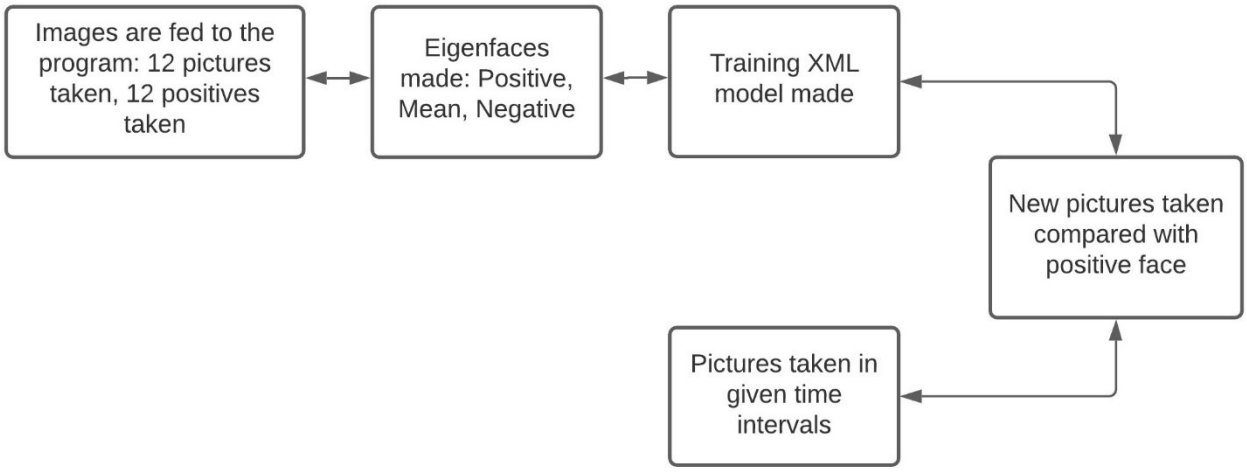

Figure 3.6: Eigenface process in OpenCV

3.7 Alert System with Android App Integration

After the face recognition process the alert system will provide feedback to the administrator about the similarity of the detected face with stored facial dataset. If an unknown face is detected then the system will notify the administrator via an email. Administrators then can access the app and see the live streaming of the Pi camera. If the detected face is recognized by the system, then information about the person is stored in the "Firebase Realtime Database". Administrators can also see the activity log or previous activity information in the app dashboard.

3.7.1 Alert Notification Via Email

If a detected face is not matched with the existing facial datasets then the system will generate a warning message and send it to the administrators gmail account. The notification system via email is implemented by using SMTP mail server. SMTP stands for Simple Mail Transfer Protocol and it defines the method that handles the process of email exchange and delivery across IPs.

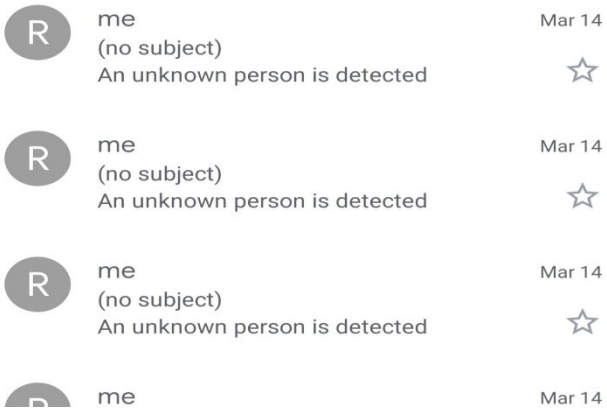

Figure 3.7: Warning notification via email

3.7.2 Android App Integration in the System

There has been a paradigm shift towards mobility solutions because of the ability of mobile applications in launching personalized communication with the users.

A recent study shows that amongst all the smartphones sold globally devices had Android operating systems equipped with them. So, to provide a complete and user friendly surveillance solution it is very much important to integrate an android app in the system. To develop the system the tool listed is used.

3.7.2.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices

Android Studio version 4.1.3 is used for development of the app.

3.7.3 Integrated Android App's Backend

Firebase Realtime Database is used as the backend of the app where all activity history of the particular camera premises is stored. The system administrator can monitor the activity of the camera premises.

3.7.3.1 Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of the clients share one Realtime Database instance and automatically receive updates with the newest data. The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how the data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it. The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database

API is designed to only allow operations that can be executed quickly [17].

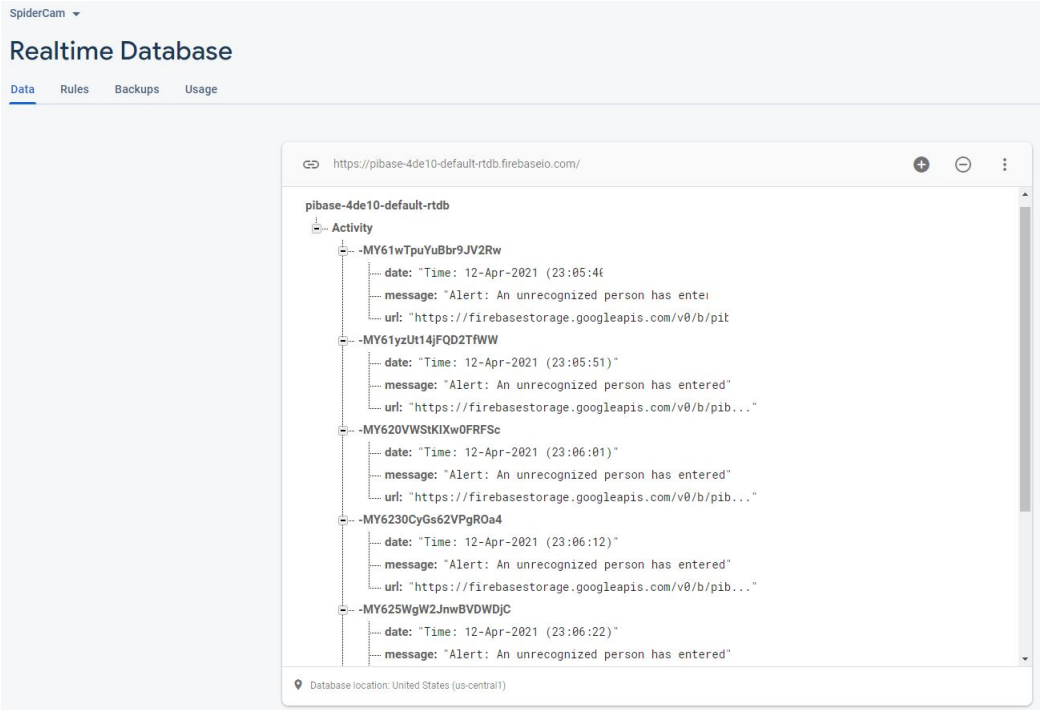

Figure 3.8: Android app's backend

3.7.4 Integrated Android App's Features

The integrated android app's user interface is pretty simple and straight forward. User can access to the app dashboard after authentication process. From dashboard live streaming of the cameras can be accessed. Also user can monitor the activity history in the app.

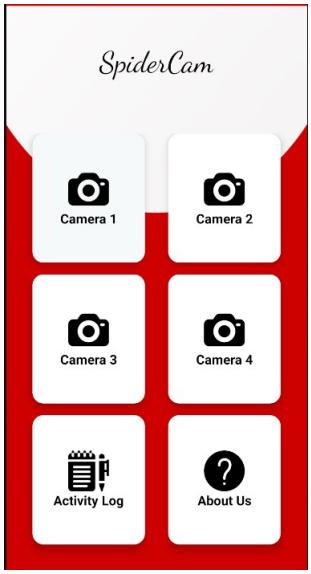

Figure 3.9: Android app dashboard

3.7.4.1 Realtime Video Stream from Camera

User of the app can access realtime video stream of the camera at any time by clicking the camera icon in the app dashboard. Primarily two cameras are integrated in the system. If an unknown person is detected in the camera, administrator can access the videos streaming and monitor the activity of that unrecognized person. Also they can access the camera streaming at any time to monitor the camera premises.

Figure 3.10: Real time video streaming of cameras

3.7.4.2 Activity History

The administrator of the system can see the previous activity of the camera surroundings by clicking on the activity log icon of the dashboard. Whenever any face is detected in the camera surroundings the system will automatically restore the image of the detected face in Firebase Realtime Database. User can see those images in the activity log which is retrieved from that database.

Figure 3.11: Previous activity monitoring in the app

3.8 Conclusion

A methodology for smart surveillance system with facial recognition has been discussed in this chapter. Face detection and recognition with the feedback to an android app is implemented in this system. The next chapter is about the experimental result analysis of the proposed methodology.

Chapter 4

Results and Discussions

4.1 Introduction

In the previous chapter, a detailed description of the overall surveillance system and different modules of the system were discussed. The performance of the proposed facial recognition based surveillance system on various facial datasets will be discussed in this chapter. Our main challenge is to observe how the proposed method handles the face recognition task in real life scenarios.
The proposed system is implemented with Raspberry Pi model 3B which has 1.2 GHz Arm Cortex-A53 CPU and 1 GB of RAM. The facial dataset was built by capturing images of 13 different individuals.

4.2 Experimental Dataset Description

The dataset used in this chapter was collected by capturing images of different individuals in different experimental circumstances. Totals of 13 individual's facial images have been collected. The images were captured in different lighting conditions to gain more accuracy in the recognition process. Also partial and total occlusion scenarios were considered while building the facial dataset. Different poses of faces were also considered while capturing the images. Another experimental decision on building the dataset was to consider the distance of the faces from the camera.
Each of these different experimental circumstances played an important role in the result generation of the face detection and recognition process. The camera quality also plays an important role in the results as we lost many facial data in low resolution images.

4.3 Evaluation of Facial Recognition Based Surveillance System

After taking photos from the Pi camera stream, the HAAR Cascade algorithm attempts to detect faces in each collected image. Image characteristics are viewed as numerical information taken from images that can distinguish one image from another in face detection. On all training photos, we apply every algorithm feature. At the start, each image is given equal weight. It determines the appropriate threshold for categorizing the faces as positive or negative. Face recognition is accomplished by computing the Euclidean distance between the feature vectors of a probing picture and a reference picture. The retrieved histogram is used to represent each image from the training dataset once the recognition algorithm has been trained. For the fresh image, we repeat the procedures and get a new histogram. We only need to match two histograms and return the picture with the closest histogram to identify the picture that matches the provided picture. There are several methods for comparing histograms (calculating the distance between two histograms), such as Euclidean distance, chi-square, absolute value, and so on. To compare the histograms, we may use the Euclidean distance formula. The program will output ID from the image with the closest histogram. The algorithm should also return the computed distance, which is referred to as the confidence measurement. If the confidence value is less than the threshold, the algorithm has correctly recognized the face.

4.4 Evaluation of Performance

Performance of a system is measured by its ability to handle challenging situations in an effective manner. Considering the challenges encountered during face detection and recognition, our process of performance evaluation has been carried out considering four criteria,

1. Change in illumination
2. Occlusion of face
3. Change in different facial expressions or poses

4. Amount of storage required

The accuracy of the proposed method is evaluated using F1-score. The F1-score is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative' [18]. F1-score in represented as the harmonic mean of precision and recall. It represents the overall performance of the system and can be calculated as follows:

$$F1score = 2 * \frac{precision * recall}{precision + recall} \tag{4.1}$$

Furthermore, accuracy is measured using matrices as follows:

$$Precision = \frac{\sum_{i=1}^{n} TP}{\sum_{i=1}^{n} TP + FP} \tag{4.2}$$

$$Recall = \frac{\sum_{i=1}^{n} TP}{\sum_{i=1}^{n} TP + FN} \tag{4.3}$$

4.4.1 Performance Evaluation for Change in Illumination

To check robustness of the system in case of change in illumination the Pi camera was set up in different places of testing environments with variable lighting conditions and the results of face recognition are observed. It is seen that, with the change of illumination the accuracy of recognition process varies.

When the illumination is high on the camera surroundings, the face recognition system provide very good results as the face detection process purely depends on lighting conditions. In low light conditions createEigenFaceRecognizer() method provides the best result in recognition process that the other two method mentioned above.The recognition process was performed 50 times in different lighting conditions to evaluate accuracy of the system for changing of illumination.

Method	Recall TP/(TP+FN)	Precision TP/(TP+FP)	Specificity TN/(TN+FN)	FPR FP/(FP+TN)	Accuracy (TP+TN)/ (ALL)
Proposed System	0.8649	0.9143	0.7692	0.2308	0.8400

Table 4.1: Performance evaluation for change in illumination

Figure 4.1: Performance evaluation for changing in illumination

4.4.2 Performance Evaluation for Occlusion of face

Occlusion refers to the partial or total disappearance of the object from the visible region for a time being. Occlusion is a big challenging factor during the recognition process. To observe the impact of occlusion of face in the results, different percentage of partial occlusion is considered in the recognition process. In the result analysis 50 images were captured with partial occlusion of face to calculate the accuracy of the system.

Figure 4.2: Performance evaluation for occlusion of face

Method	Recall TP/(TP+FN)	Precision TP/(TP+FP)	Specificity TN/(TN+FN)	FPR FP/(FP+TN)	Accuracy (TP+TN)/ (ALL)
Proposed System	0.8056	0.8529	0.6429	0.3571	0.7600

Table 4.2: Performance evaluation for occlusion of face

4.4.3 Performance Evaluation for Change in Facial Expression or Poses

Facial poses are very sensitive to facial recognition systems. When a person's head moves and their viewing angle changes, the posture of their face changes. Changes in face appearance caused by head motions or different camera POVs inevitably trigger intra-class differences, lowering automatic face recognition rates dramatically. It becomes difficult to tell who the actual person is. It may result in faulty recognition or no recognition if the stored

facial dataset only has the frontal view of the face. The face recognition process was performed 50 times with different facial poses for accuracy measurement.

Figure 4.3: Performance evaluation for different poses

Method	Recall TP/(TP+FN)	Precision TP/(TP+FP)	Specificity TN/(TN+FN)	FPR FP/(FP+TN)	Accuracy (TP+TN)/ (ALL)
Proposed System	0.8750	0.9722	0.9000	0.1000	0.8800

Table 4.3: Performance evaluation for different poses

4.4.4 Amount of Storage Required

All visual and auditory data is now stored digitally in today's environment. Every new day in surveillance necessitates more terabytes of storage capacity for the thousands of cameras positioned throughout numerous cities, major towns, and regional locations. However, problems arise as storage expands at the same time as recording time. Short-term storage is one solution to this challenge. The first in, first out (FIFO) principle is used to recycle available storage space in short-term storage. In other words, if recorders only have a 14-day storage capacity, an occurrence that happened three weeks ago cannot be recovered since it would

CCTV camera recording size of 1 hour (640*480, 15 FPS)	Images stored in DB (proposed system) (640*480)
2.3 GB	50 KB

Table 4.4: Required storage comparison

have been overwritten by a fresh recording made within the last 14 days.
We attempted to overcome this storage issue in our proposed system by storing photos of detected faces rather than continuous recordings of camera premises. Our system generates a warning notification anytime any unidentified face is discovered, so it's a fair trade-off. Here is a comparison of the amount of storage space required by our proposed system against general CCTV recordings.

4.5 Conclusion

This chapter shows the result of facial recognition based surveillance system on different real life circumstances. Performance of the proposed surveillance system is also discussed here. In the next chapter, the conclusion is drawn on this thesis work.

Chapter 5

Conclusion

5.1 Conclusion

The improvement of technology in the recent years have helped in the automation of most of the tasks carried out by humans. With that view, CCTV cameras have been installed in almost all the infrastructures in city areas. These CCTV cameras are viewed as a means of providing security to a building that records the entry and exit of people in different times of the day. If a person can be recognized by his face by the camera, then we can control and monitor the entry point of camera premises.

However, there are some major challenging factors in the face recognition process. Considering all these factors, it can be deduced that the facial recognition based surveillance system will need various optimization techniques with the improvement of Computer vision. The task of facial recognition in surveillance systems is chosen in particular because of raising security concern in our daily lives. Also due to Covid-19 outbreak the necessity of a hands free human identification process has increased.

The proposed system addresses the difficulties of recognizing human faces in case of change of lighting conditions, occlusion of faces, different facial expressions and low resolution of images as the major challenges. Also cost effectiveness is another major challenge while developing this kind of system.

With the aim of making the facial recognition based surveillance system more efficient and cost effective we have developed the project using Raspberry Pi which reduces the cost of development and also provides required computational power. For making the system user friendly and to provide real time feedback, we have developed an android app with many exciting features for real time monitoring of the camera premises.

5.2 Future Work

Regardless of the effectiveness of various algorithms and libraries, facial recognition in surveillance system still remain a problem due to different unwanted conditions which can't be solved completely. But we can always try to increase the efficiency of the process by using different classifiers and optimizing methods of computer vision libraries.

The proposed system can be improved further by gaining more accuracy in recognition process with more improved classifiers. And also there is room for improvement in the recognition process during low light conditions and partial occlusion. Finally, the work can be extended more to cope with the recognition process in this Covid-19 pandemic situation since we have to wear mask in regular basic. It would be a great achievement if a system can recognize human wearing mask by their body movement pattern.

References

[1] [Online]. Available: https://en.wikipedia.org/wiki/Facial_recognition_system#History_of_facial_recognition_technology (cit. on pp. 1, 2).

[2] R. K. K. Goyal Kartikey Agarwal, 'Face detection and tracking: Using opencv,' *International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, pp. 474–478, 2017 (cit. on p. 7).

[3] M. Khan, S. Chakraborty, R. Astya and S. Khepra, 'Face detection and recognition using opencv,' in *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2019, pp. 116–119 (cit. on p. 7).

[4] M. Madhuram.M, B. P. Kumar, L. Sridhar, N. Prem and V. Prasad, 'Face detection and recognition using opencv,' *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, 2018 (cit. on p. 7).

[5] S. P. N. Swapnil Vitthal Tathe Abhilasha Sandipan Narote, 'Face recognition and tracking in videos,' *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, 2017 (cit. on p. 7).

[6] C. C. Loy, 'Face detection,' in *Computer Vision: A Reference Guide*. Cham: Springer International Publishing, 2020, pp. 1–5, ISBN: 978-3-030-03243-2 (cit. on p. 7).

[7] P. D. S. A. D. Gaikawad, 'An efficient video surveillance system using video based face recognition on real world data,' *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 5, 2016 (cit. on p. 8).

[8] A. Jagtap, V. Kangale, K. Unune and P. Gosavi, 'A study of lbph, eigenface, fisherface and haar-like features for face recognition using opencv,' Feb. 2019, pp. 219–224 (cit. on p. 8).

[9] I. Gupta, V. Patil, C. Kadam and S. Dumbre, 'Face detection and recognition using raspberry pi,' in *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2016, pp. 83–86 (cit. on p. 8).

[10] P. G. N. Amit A. Nandurkar, 'An internet of things approach for security surveillance using raspberry-pi–a survey,' *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, 2017 (cit. on p. 8).

[11] E. N.Sugumaran G.V.Vijay, 'Smart surveillance monitoring system using raspberry pi and pir sensor,' *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 4, pp. 11–15, 2017 (cit. on p. 8).

[12] A. Najmurrokhman, K. Kusnandar, A. B. Krama, E. C. Djamal and R. Rahim, 'Development of a secured room access system based on face recognition using raspberry pi and android based smartphone,' *MATEC Web of Conferences*, vol. 197, p. 11 008, 2018 (cit. on p. 9).

[13] D. S. S. Mahesh, T. Reddy, A. S. Yaswanth, C. Joshitha and S. S. Reddy, 'Facial detection and recognition system on raspberry pi with enhanced security,' in *International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–5 (cit. on p. 9).

[14] Nov. 2020. [Online]. Available: `https://opencv.org/about/` (cit. on p. 13).

[15] [Online]. Available: `http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html` (cit. on p. 13).

[16] [Online]. Available: `http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-recognition-%20with-opencv` (cit. on p. 15).

[17] [Online]. Available: `https://firebase.google.com/docs/database` (cit. on p. 18).

[18] [Online]. Available: `https://deepai.org/machine-learning-glossary-and-terms/f-score` (cit. on p. 23).