ISSN 1533-9211

DOI 10.5281/zenodo.7059664

A STUDY ON APPLICATION OF FACIAL RECOGNITION FOR DEVELOPING AN ANDROID APP TO TRACK ATTENDANCE

[1]Dr. SESHAIAH MERIKAPUDI, [2]ANIRUDDHO MITRA, [3]Dr. MURTHY SVN and [4]Dr. MANJUNATHAKUMAR BH

[1]Associate Professor, [2]Scholar, [3]Associate Professor, [3]Professor

[1,2,3,4] Computer Science and Engineering, SJC Institute of Technology, Chickballapur, India.

Abstract:

If we take a break from our daily lives and have a look around us then we would be surprised to find how much everything around us has changed in the past few decades. Among all the sectors we have observed from medical science to business and from education to technology, one of the domains which has evolved drastically in the recent years has been the advancements and the innovations in technology and education. Both of them has been marvellously blended together to provide us with a wonderful e-learning tools like the online meetings applications which has proven to be the backbone of the organisations during the pandemic era. Notwithstanding all these advancements, we still find that the attendance system has been the same for several decades i.e., the old school way of using the pen and paper to mark the attendance after calling out the roll codes or names is still being used in almost every institution. This leads to not only wastage of time but also many other resources. Henceforth, I present this paper with the motive of finding the possibility of developing an android application which would free the world of the traditional pen and paper-based attendance system and instead would use face recognition libraries and APIs to detect as well as recognise the faces and use the preloaded data to mark the attendance of the respective students or employees. This would also eliminate the most prevalent issue of proxy attendance marking and provide a transparent attendance mechanism.

Keywords: Face net, Android Operating System, Machine Learning, Library, API, Student Attendance System, Facial Recognition, Face Detection, Mobile Application, Tensor Flow, and Neural Networks.

1. Introduction

People have always had the ability to recognize and distinguish two different faces, while computers have begun to show the same ability not very long ago. By the mid-1960s, researchers began the work on the recognition of human faces using the computer.

Face recognition has been one of the hot fields in computer vision and biometrics, since it is a challengeable work to identify a face image with varying expression, occlusion, disguise, and illumination. Along with fingerprint match and eye retina match, they become a prominent method in biometric technology. The task of face recognition is to compare a query face against multiple faces in the database in order to identify whether it belongs to a person in a database. In some identification applications, one needs to find the most similar face. In a more advanced application, the requirement is not only finding the most similar faces, but also the data associated with detected person as well as the notification system in terms of showing a suspicious person. In the face recognition systems, the facial detection is an initial step to capture an image of the user to be processed. The facial detection usually uses a light proximity sensor that consists of photodiode components. However, these sensors can only detect faces

ISSN 1533-9211

DOI 10.5281/zenodo.7059664

horizontally to the line. For a wider view of the face image, the use of webcam can provide more capability than the proximity sensor.

The proposed system employs the phone's camera to capture the facial image. Then, the image will be processed further by a certain algorithm to get the best results. The captured image will be compared to the user database in the system. If the person's face is present in the database as well as in the captured image, then he/she will be marked present or else absent. Such mechanism was made to assure that only authorized person's attendance is marked and minimise the possibility of proxy attendance.

Taking and monitoring student's attendance in classes have been a long-standing issue amongst lecturers and the school management as a whole because of the numerous complications associated with the conventional paper and pen approach of taking attendance. It is a very stressful method of keeping record with important data and information getting lost easily, thus making the process inefficient for monitoring student's attendance in classes. The conventional approach has over time proven inefficient as students can easily write attendance for their friends who are absent, and the fact that records can be misplaced or destroyed easily.

The credit of enriching the lives of every individual user goes to mobile applications that have altogether brought a drastic transformation globally. At the same time, it has provided ample scope to the developers to showcase their hidden talents and creativeness and make a lot of tasks easier and more accessible. The mobile app technologies have taken a curvature that is beyond imagination.

Several reasons exist for selecting a mobile app over creating a website or implementing on other IoT devices:

1) **Mobile apps are budget-friendly**: With the advancements in the field of technology new methods and tools are getting developed and the process of app development has become both affordable and budget friendly. Thus, any person even a student can learn App Development within a few weeks and can develop and maintain apps like these to help their organisations and institutions.

2) **24/7 Availability**: Mobile apps are available all the time for use. This aspect would be very useful in the situations when the teacher forgets to bring the register to the classroom. Since everybody has access to a mobile phone all the time, it is just a matter of opening the app and capturing the images of the students and the app would automatically update the attendance of the students who are present in the class as it is not a difficult task to click a picture and a few buttons so anyone can do it.

3) **Remote Access**: In case the teacher is unable to take the class then he/she can assign the class to any substitute teacher and they can update the attendance using the app and the teacher who is absent can access the attendance data on his own device easily.

ISSN 1533-9211

DOI 10.5281/zenodo.7059664

2) Analysis

Facial recognition is a way of using software to determine the similarity between the two face images in order to evaluate a claim. The technology can be used for a variety of purposes, from signing a user into their phone to searching for a particular person in a database of photos. Facial recognition uses computer generated filters to transform face images into numerical expressions that can be compared to determine their similarity. These filters are usually generated by using Deep Learning which uses Artificial Neural Networks to process data.

Each human face has many distinctive landmarks, different deck peaks that make up facial features. Each face has approximately 80 nodal points of which the following are used for facial recognition Distance between the eyes, Width of the nose, Depth of the eye sockets, the shape of the cheekbones, and the length of the jawbones.

Face Detection vs Face Recognition: First of all, let's see what "face detection" and "face recognition" means. While many people use both terms interchangeably, they are actually two very different problems.

Face Detection, in short is: given an input image, to decide if there are people's faces present in that image. And for each present face, to know where each face is located (e. g. a bounding box that encloses it) and possibly, also to know the position of the eyes, the nose, the mouth (known as face landmarks).

Face recognition: given an image of a person's face, identify who the person is (from a known dataset of registered faces). Let's say we have a dataset with the registered faces of the input image (Bach, Beethoven and Mozart). For any new face image, we want to know who the face belongs to.

The reason for not using the Google's ML kit is that it provides the feature of face detection and not face recognition yet. But still for the first part of the problem that is face detection we will use the ML kit and some other approach for the face recognition part.

ISSN 1533-9211
DOI 10.5281/zenodo.7059664

3) Design

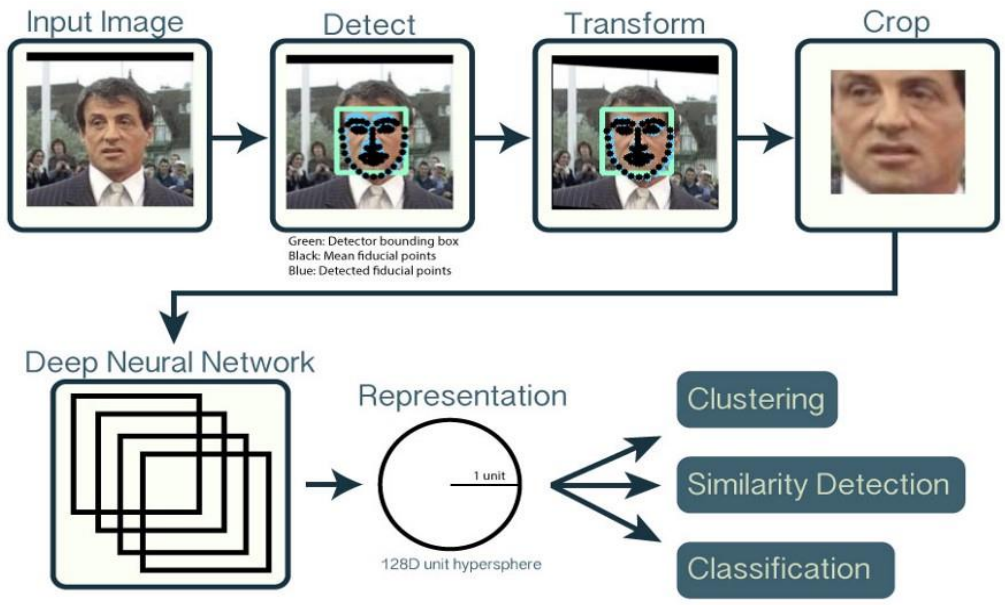

Figure 1: Deep Newral Network Architecture

In the first step, the face is detected on the input image.

In the second step the image is warped using the detected landmarks to align the face (so that all cropped faces have the eyes in the same position).

In the third step the face is cropped, and properly resized to feed the recognition Deep Learning model. Also, some image pre-processing operations are done in this step (e. g. normalizing and "whitening" the face)

In the fourth step the most "juicy part", is the one depicted as "Deep Neural Network". We are going to focus more on this step.

The main idea is that the deep neural network **DNN** takes as input a face **F** and gives as output a **D** =128 dimensions' vector (of floats). This vector **E** is known as embeddings. These embeddings are created such as the similarity between the two faces **F1** and **F2** can be computed simply as the Euclidean distance between the embeddings **E1** and **E2**.

$$Similarity(F1, F2) = \|DNN(F1) - DNN(F2)\|_2 = \sqrt{\sum_{i=1}^{D}(E1_i - E2_i)^2}$$

We can now compare two faces **F1** and **F2**, by computing its Similarity, and then check it against some **threshold.** If lower, we can say that both faces are from the same person.

ISSN 1533-9211

DOI 10.5281/zenodo.7059664

4) Literature Survey

FaceNet is a face recognition system developed in 2015 by researchers at Google that achieved the state-of-the-art results on a range of face recognition benchmark datasets **(99.63% on the LFW)**. This work introduces the novel concept of triplet loss. They proposed an approach in which it generates a high-quality face mapping from the images using deep learning architectures such as ZF-Net and Inception. Then it used a method called triplet loss as a loss function to train this architecture.

It uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. To train, Facenet uses triplets of roughly aligned matching / non-matching face patches generated using a novel online triplet mining method. The benefit of this approach is much greater representational efficiency and also it has achieved state-of-the-art face recognition performance using only 128-bytes per face. On YouTube Faces DB it achieves 95.12%.

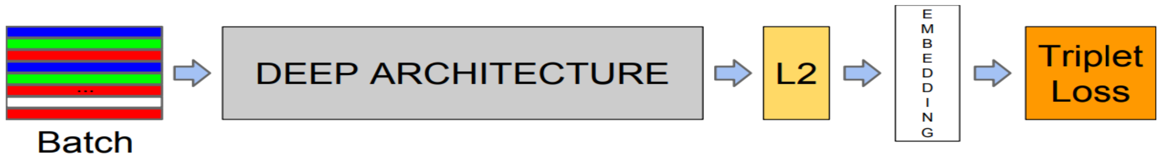

Figure 2: Triplet Loss Function

FaceNet employs end to end learning in its architecture. It uses ZF-Net or Inception as its underlying architecture. It also adds several 1*1 convolutions to decrease the number of parameters. These deep learning models outputs an embedding of the image f(x) with L2 normalization performed on it. These embeddings then passed into the loss function to calculate the loss. The goal of this loss function is to make the squared distance between two image embedding is independent of image condition and pose of the same identity is small, whereas the squared distance between two images of different identity is large. Therefore, a new loss function called Triplet loss is used. The idea of using triplet loss in our architecture is that it makes the model to enforce a margin between faces of different identities.

The implementation initially thought upon was by David Sandberg's FaceNet Implementation. Using his implementation, we can achieve the execution time of around 3.5 seconds, the comparison is accurate and we can compare faces offline on a mobile. But in order to get faster speed and lightness we can use a Mobile DNN Architecture.

Using the MobileFaceNet: It is a great work by researchers at Watch data Inc. in Beijing, China. They presented a very efficient CNN model specifically designed for high-precision real-time face verification on mobile devices. They achieved impressive speeds with very high accuracy with a model of just 4.0 MB. The accuracy they obtained is very similar to that of other heavier models (such as FaceNet). After for some MobileFaceNet implementation to bring it to Tensor Flow Lite. Most available implementations are for PyTorch, which could be

ISSN 1533-9211

DOI 10.5281/zenodo.7059664

converted using the ONNX conversion tool. But since this tool is still in early stages of development, The implementation opted is a MobileFaceNet implementation on Tensor Flow, from Sirius-ai(https://github.com/sirius-ai). The resulting file was very light-weight only 5.2 MB really good for a mobile application.

5. Updating attendance on recognising face

After the face has been recognised by the app the next task is to update the attendance of the person whose face has been detected. The first step is to decide what database and tools to make use of. For keeping our application more secure we are going to use an offline solution through which all the attendance will be stored in the device itself. Among the most widely used database solutions we have SQLite which can provide the same reliability and flexibility like SQL but for mobile devices.

We are going to modify the default working of the application as in the default mode the only data which is stored will be the student's name. What we are going to do is that after scanning each student's face we will ask for the student's roll number instead of the name so that the database will make the tables and add the roll numbers of the students to the table and we will consider them as the primary keys of our table to uniquely identify each row. We cannot use names for this purpose as two students can have the same name, although we can keep a field for storing the student names. In order to maintain consistency, initially all the student's attendance data has to be stored in the database so that the database can create as many numbers of rows as required. Later if any student unenrolls from the school or college then we can simply delete that student's data from the table. Further in case the number of students increase then we can simply scan the new student's face and the database will add that student's data to the backend database. Also, initially the database won't have the full calendar, it will add the date's everyday whenever there is a working day as this approach will save the extra space which the database may require if we also save the holidays and some other days in which there were some events and the classes didn't take place.

The teacher is going to take a picture of the entire classroom and the app is going to scan and detect all the faces in the picture as they would have been pre-fed into the application and the application will mark the attendance of the students whose faces have been detected and it will mark absent for the students whose faces are not there in the photograph. Another advantage of this method is that the teacher can also export and transfer the file into a computer and then can use the appropriate SQL commands to view all the contents of the table. Every teacher will have the app installed in his/her phone and everyone can have the attendance record of each student of their respective subjects. And if some other teacher comes as a substitute in place of the assigned teacher, then the substitute teacher can simply share the SQL file with the assigned teacher and he/she can get that day's attendance record very easily. They need not to go through the substitute teacher's register and mark the attendance individually. All he has to do is to get the SQL file and join that day's attendance record in the table with his already existing table of his database and voila the work is done within a few minutes and a few commands. SQL provides these unique features like JOIN in order to manage the database more efficiently and effectively. The table names will be of the subjects for which the class will be going on.

ISSN 1533-9211

DOI 10.5281/zenodo.7059664

Through this approach we can uniquely identify each table since each subject code will be different from one another.

6) Future Enhancements and conclusion

1. Converting the TensorFlow Dataset into SQL dataset or vice versa: Currently all the facial data is being run and analysed using the TensorFlow platform and libraries and also the face recognition data is being stored in the form of TensorFlow Datasets. HashMaps and Array List are being used in order to store the facial data. The Hash Map stores the facial data in the form of key value pairs in which the key is the name and the value is the facial data associated with the name i.e., the facial features, distances and depths of different points on the face and so on. But in order to make our application work as intended we need to find a method to convert the existing TensorFlow Lite Datasets into SQL dataset in which the facial data would be stored in the form of tables consisting of rows and columns. This would not only make the app more functional by making it easier to implement the attendance system but also it would be easier for developers to manage and maintain the databases and there are some more advantages which has been discussed below.

2. Portability of the attendance system: Although the SQL file containing all the attendance records can be transferred from one android phone to another but apart from this, we also have to design the app in such a way so that after transferring the SQL file to another device it can reload the existing attendance data from the old device and update it on the new device in order to let the invigilator continue/manage the attendance from where it was left off on the previous device. Currently our system is not designed in such a way so that it can restore the attendance record from the old device to a new device but it certainly can be achieved with some modifications to the app as well as to the SQL file in the way the attendance data is stored so that when the file is transferred to a new device then it can adapt to the new device as well as restore the old attendance record. This is one of the advantages of storing the data locally in a SQL file so that it can increase the portability of the system.

3. Changing of the attendance status: There can be many instances when the attendance record of a person has to be updated. It may involve making it present to absent or making it from absent to present. This situation may arise if a person or student is absent from work or class due to some health or other unavoidable reasons and later provides a valid medical certificate or document. In such a situation their attendance can be updated or if someone arrives late and has been marked as absent then he/she can inform the respective person and get their attendance updated. But with our current attendance system the process of changing the attendance status is not available. This is another limitation with our system which can be rectified later. The changing of the status from present to absent can be done in order to rectify the false positive cases in case any face which is not present physically in class is scanned and is marked absent. Although the chances of this happening are very low but a machine developed using AI and ML currently is not 100% fool proof and thus, we must also keep this under consideration.

With the huge advancements in the fields of technology and education the ed-tech sector is booming with new opportunities and a whole new era is coming which would not only change

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7059664

the way we teach and learn but also it would transform our way of thinking, lifestyle and also our outlook towards life. In today's world where time is the most valuable resource a person can have, the idea of this attendance application based on face recognition will save a lot of time of not only the students but also millions of people working at various places and of all the organisations and people related to them. The saved time can then be utilised in other productive works thereby providing a new direction and definition to our lives. Finally, I would like to conclude my paper with the golden words of Sir Stephen R. Convey: "The key is not in spending time, but investing in it". So let us not waste any further time and invest it in doing something good not only for ourselves but also for our society.

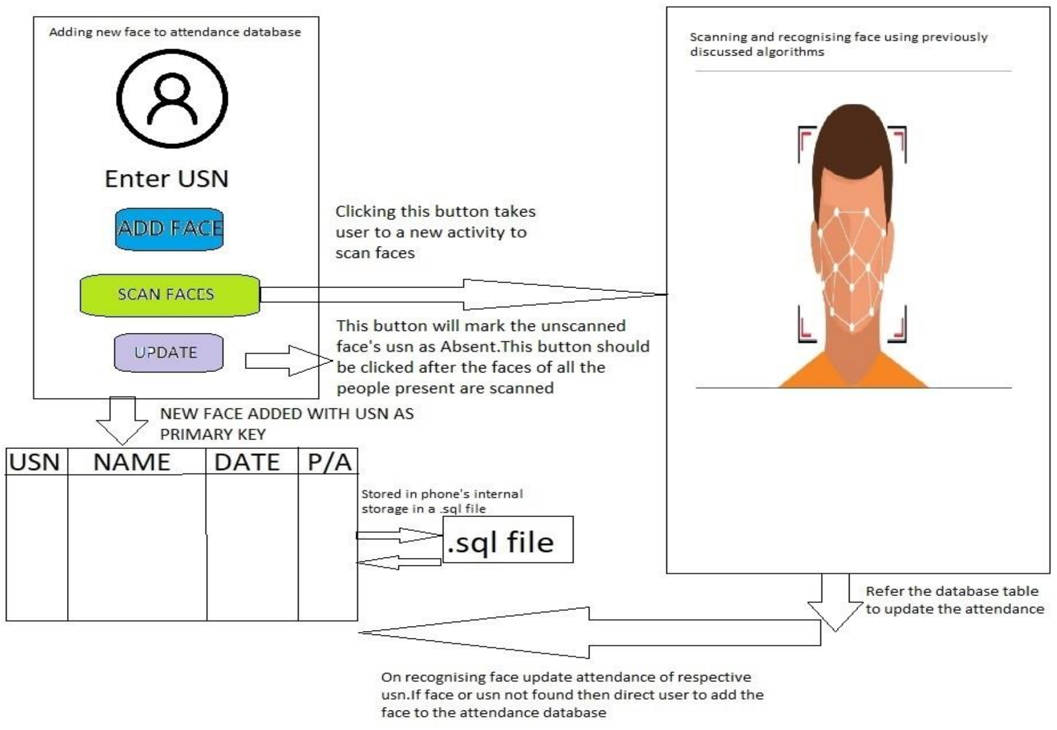
Adding new face to attendance database
Enter USN
ADD FACE
SCAN FACES
UPDATE
Clicking this button takes user to a new activity to scan faces
This button will mark the unscanned face's usn as Absent.This button should be clicked after the faces of all the people present are scanned
NEW FACE ADDED WITH USN AS PRIMARY KEY
USN
NAME
DATE
P/A
Stored in phone's internal storage in a .sql file
.sql file
Scanning and recognising face using previously discussed algorithms
Refer the database table to update the attendance
On recognising face update attendance of respective usn.If face or usn not found then direct user to add the face to the attendance database

Figure 3: Attendance Flow Diagram

REFERENCES

[1]Folasade O. Isinkaye,Jumoke SoyeAmi,Olamide I. Arowosegbe, "An Android-based Face Recognition System for Class Attendance and Malpractice Control",International Journal of Computer Science and Information Security (IJCSIS),Vol. 18, No. 1, January 2020

[2]Octavian DOSPINESCU1 Iulian POPA2, "Face Detection and Face Recognition in Android Mobile Applications",Informatica Economică vol. 20, no. 1/2016

[3]: Sheng Chen, Yang Liu, Xiang Gao, Zhen Han, et al.["MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices] — Chinese Conference on Biometric Recognition. — Apr, 2018

[4]: F. Schroff, et al. ["FaceNet: A unified embedding for face recognition and clustering,"] — 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815–823 — Jun, 2015

ISSN 1533-9211

DOI 10.5281/zenodo.7059664

[5]: Satya Mallick, at al. [Face Recognition: An Introduction for Beginners]—learnopencv(Apr, 2019)

[6]: Adrian Rosebrock. [Face Alignment with OpenCV and Python] — pyimagesearch(May, 2017)

[7]: Adrian Rosebrock. [OpenCV Face Recognition] — pyimagesearch(Sep, 2018)

[8]: Jason Brownlee. [How to Develop a Face Recognition System Using FaceNet in Keras] machine learning mastery (June, 2019)

[9] https://github.com/sirius-ai

[10] https://github.com/davidsandberg/facenet

[11] https://github.com/tensorflow/examples/tree/master/lite/examples/object_detection/android

[12] https://medium.com/@estebanuri/real-time-face-recognition-with-android-tensorflow-lite-14e9c6cc53a5

[13] How Does Facial Recognition Work? A Primer By William Crumpler and James A. Lewis