INTELLIGENT FACULTY TRACKING SYSTEM IN EDUCATIONAL INSTITUTIONS*

BY

RUDRA DUTT, PRANAY NARANG, AKSHAT SRIVASTAVA, TARUN ADITYA KUSUPATI,

DR BISWAJEET CHAMPATY♣

Ajeenkya DY Patil University, Charholi Budruk, Pune, Maharashtra

Email: rudra.dutt@adypu.edu.in, pranay.narang@adypu.edu.in,

akshat.srivastava@adypu.edu.in tarun.kusupati@adypu.edu.in,

biswajeet.champaty@adypu.edu.in

ABSTRACT

In a large organization, the faculty fraternity is much occupied due to their multiple administrative roles, teaching loads, research experiments, and other creative co-curricular activities. Hence, they are hardly found at their designated places. This creates difficulty for those students who want to meet the faculty for any administrative or academic advice. Eventually, this creates a delay in the process, surpassing deadlines and loss of student's interest in academics. Therefore, this current study proposes an intelligent tracking system to ensure the availability of faculties at any instant time. The app-based system captures the facial image at any given instance of time, processes it using deep learning techniques, and sends feedback to the user about the availability of the concerned faculty. This process enhances the productive working culture in educational institutions.

KEYWORDS

Deep Learning, Model Training, Facial Data, Intelligent Tracking System, Recycler View.

I.INTRODUCTION

In the Contemporary world, one of the most commonly faced problems by the student's is to check the availability of a faculty in his/her cabin, sometimes they have to travel through buildings to check the presence of the faculty to get their work done, this sometimes leads to student's wasting their time, we decided to find a solution to this problem as this not only leads to time and energy-wasting of the students but can also disturb the faculty if he/she is busy in some important work. To find a solution to it we decided to develop a system that would let you

* Received 22 September 2021, Accepted 09 October 2021, Published 24 October 2021
♣ Corresponding Author

know the presence of faculty in his/her cabin through your mobile. In our system, we would be developing an application that list's all the departments of the college and then within the departments the list of all the faculty members, along with the name of each faculty member there will be a check button which when clicked would display a loading icon and at the same time our backend API would run via the HTTP request received after the click and grab a photo of the faculty in the room assigned to him, the backend API would then run facial recognition over the clicked image and use the face data that would be collected from the college of all the faculty member, to process the image and find faces, once the API succeeds in finding the faces it would send the output as a JSON response containing the names of the faces detected in the image back to the application the response would be then processed, and it would compare the response name with the faculty name and if it matches then it would show a tick otherwise a cross is what we plan to develop to counter the problem. The backend API would be hosted on a server built on-site inside the campus for easy deployment and maintenance with a basic configuration and would be connected to all the local cameras placed inside the staffrooms or specific faculty cabins and would be only visible on the local network to ensure security.

II.FRONTEND

The Application would begin with a login page in which the student needs to enter the id provided to him by the college. On Clicking the login button a GET request will be sent to the API which in response will redirect the student to the next activity which will contain the data of all the schools under their college in a list format. When the student will select their respective school, another activity will startup which will contain cards in vertical view format and each card will contain a name, image, and a check button for each faculty member under that school. When the check button is clicked for a faculty member some backend processing will take place and in just a matter of seconds, it will display the desired result whether the faculty member is present in his/her cabin or not.

For displaying the data of schools and faculty members we have used Card View inside Recycler View. The reason for selecting the RecyclerView over ListView is that RecyclerView is an optimization over ListView. RecyclerView view only creates spaces for the custom objects which are displayed on the mobile screen of the user, in a scroll view if one custom object leaves the screen its space is deallocated and space same is allocated to the next custom

object which will enter the screen. On the other hand, ListView creates spaces for all the objects in the memory, which leads to some memory wastage.

III.BACKEND

(a)PRIOR WORK

As we all know whenever we start to develop an idea we have to go through rigorous reading and research to build onto it similar was the case with us, once our idea was ready we went through a similar process, but there was not a single research paper matching our idea but there have been many advancements in the field of Facial Recognition and it is being done through several methods, but in our case, we are considering deep learning because it's the most efficient one. Here are some of the researches that have been done in this field.

1.Adam Geitgey's "Modern Face Recognition with Deep Learning" [1].

2.Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications "[2].

(b)METHODOLOGY

IFTS (Intelligent Faculty Tracking System) is a system that consists of numerous components, one of which is Face Recognition, which plays a critical part in the overall system.

(b.1) Face Recognition [3] –

Facial recognition is a method of recognizing or verifying an individual's identity by glancing over their face. People can be identified in pictures, films, or real-time using facial recognition systems.

Face Recognition systems can vary, but in general, they tend to operate as follows:

(b.1.1) Face detection –

The camera recognizes and tracks the image of a face, whether it is alone or in a crowd. The person in the image could be staring straight ahead or in profile.

(b.1.2) Face analysis –

Since it is easier to match a 2D image with public photos or those in a database, most facial recognition technology uses 2D images rather than 3D images. One's face's geometry is read by the system. The distance between your eyes, the depth of your eye sockets, the distance between your forehead and chin, the curve of your cheekbones, and the contour of your lips, ears, and chin are all important considerations. The goal is to figure out what facial landmarks are important for differentiating your face.

(b.1.3) Converting the image to data –

Based on a person's facial traits, the face capture method converts analog information (a face) into a set of digital information (data). The examination of your face is effectively reduced to a mathematical formula. Faceprint refers to the numerical code. Each person has their faceprint, similar to how thumbprints are unique.

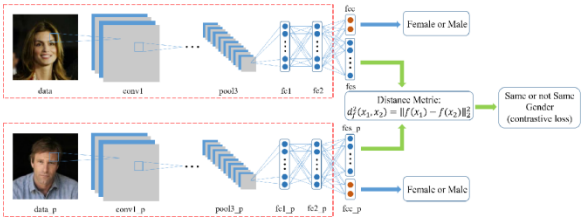

Fig. Reference

As stated above there are various models/systems for recognizing faces, after thorough research, we considered Face Net. Face Net is a face recognition system developed in 2015 by researchers at Google that achieved state-of-the-art results on a range of face recognition benchmark datasets.

(b.2) Face Net [4] –

Face Net is a uniform embedding that is used for face recognition, verification, and clustering. It converts each face image into Euclidean space with distances that correlate to face similarity. Face Net differentiates itself from other systems as it learns the mapping from the photos and builds embeddings instead of relying on a bottleneck layer for recognition or verification. Once the embeddings are created, the rest of the tasks, such as verification and recognition, can be completed utilizing typical domain approaches using the newly generated embeddings as the feature vector. Face Net is based on a deep convolutional neural network (DCNN). The network is trained to match face similarity to squared L2 distance between embeddings.

Face Net's loss function is another integral component. It makes use of the triplet loss function [5].

$$Loss = \sum_{i=1}^{N} \left[\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha \right]_+$$

Fig. Triple Loss Formula

The loss function optimizes for the largest distance between the anchors (Fig. Loss FN) and negative sample and the smallest distance between the positive and anchor sample. It cleverly combines both metrics into one loss function. It can optimize for both cases simultaneously in one loss function. If there is no negative sample, the model will not be able to differentiate different people and vice versa.

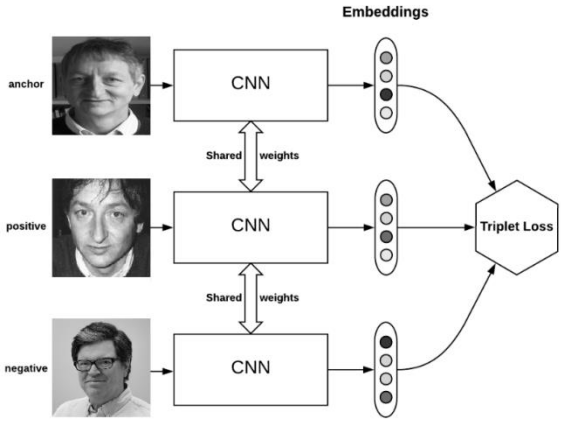

Fig. Loss FN

(c)Process Operation -

On the implementation side, IFTS can be considered as a collection of tools and interfaces implemented in both C++ and Python languages.

Fig. Request

Step by step process for Request (Fig. Request) –

1. Client sends a request to the server.

2. The server running flask receives the request and forwards it to the Database running MongoDB.

3. MongoDB checks the number assigned and forwards the request to the dedicated camera.

4. The camera takes a snapshot and sends it back to the server for further processing.

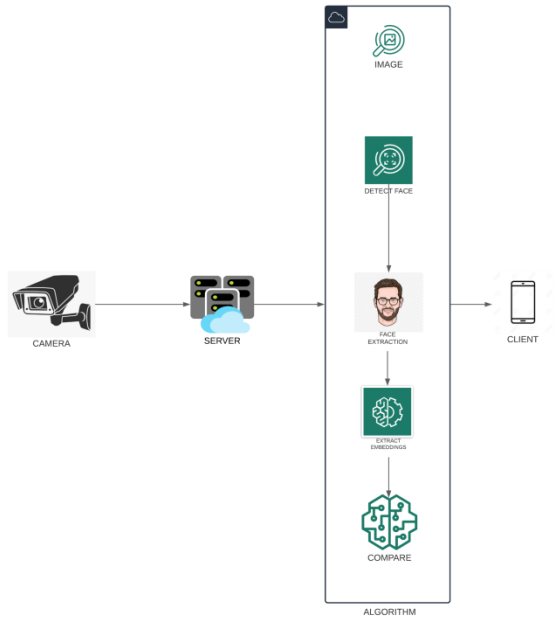

Fig. Response

Step by step process for Request (Fig. Response) –

The sequence is reversed in response where the image is sent to the service which takes care of the evaluation.

The server runs the image through the algorithm which -

a.Detects Face -

Detects all the faces present in the image.

b.Extracts Face -

Extract all the faces from the image for features.

c.Extract Embeddings -

Given a picture of a face, it will extract high-quality features from the face and predict a 128 element vector representation of these features.

d.Compare-

The features are compared with the trained model and a response is sent back to the client.

(d)Process Summary –

When a client requests a specific faculty, the server runs through a list of numbers assigned to that faculty in our database and then sends a request to the camera assigned to that faculty for a snapshot.

Once the camera catches a glimpse and uploads it to the server, our algorithm examines the face embeddings from the training model to see if the faculty is present or absent. Then the server sends a response to the client after evaluating the image.

IV.CONCLUSION

In this paper, we have taken a subtle approach that demonstrates the advantages of using IFTS (Intelligent Faculty Tracking System) for a particular purpose. Our approach of running the system in-house gains a valuable amount of trust and reduces unwanted risks of the cloud which hampers the role of privacy. IFTS (Intelligent Faculty Tracking System) will continue to be extended in various directions. Firstly, we will expand the implementations to ARM and mobile devices for operations/processing which could largely help reduce time, running costs and help in mobility. The Addition of new tools and frameworks with time is planned.

We look forward to continuing to develop IFTS (Intelligent Faculty Tracking System), in the pursuit of faster and efficient model training and deployment and ever-increasing reproducible research.

REFERENCES

[1]Adam Geitgey's, "Modern Face Recognition with Deep Learning," https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78.

[2] Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications" http://reportsarchive.adm.cs.cmu.edu/anon/anon/2016/CMU-CS-16-118.pdf

[3] KasperskyTeam, "What is Facial Recognition – Definition, and Explanation" https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition.

[4]Clement Hui, "Explanation of Facenet Model for face recognition, "https://ai.stackexchange.com/questions/16361/detailed-explaination-of-facenet-model-for-face-recogniton. Dhairya Kumar, "Introduction to FaceNet: A Unified Embedding for Face Recognition and Clustering." https://medium.com/analytics-vidhya/introduction-to-facenet-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02. Florian Schroff, Dmitry Kalenichenko, James Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering."

https://arxiv.org/abs/1503.03832. Jason Brownlee, "How to Develop a Face Recognition System Using FaceNet in Keras"https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/. Luka Dulčić, "Face Recognition with FaceNet and MTCNN" https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/.

[5] Clement Hui, "formula used to calculate the loss in the FaceNet model.",https://ai.stackexchange.com/questions/16311/what-is-the-formula-used-to-calculate-the-loss-in-the-facenet-model.

[6]Armin Ronacher and The Pallets Projects, "Flask", https://palletsprojects.com/p/flask/

https://flask.palletsprojects.com/en/2.0.x/index.html

[7]Roy Ficlding, "Representational State Transfer", https:/restfulapi.net/.

[8]App development material, https://developer.android.com/docs