Automated Facial Recognition System

Akash Dhasade[1], Saket Joshi[2] and Dr.Rama Krishna Gorthi[3]

***Abstract*— This B.Tech project aims at researching practical aspects of facial recognition systems. The end goal is to build a complete a attendance system for fully automated attendance in all classrooms at the Institute. In this report, we specify the data acquisition steps, discuss the dataset generation algorithm, summarize the complete facial recognition pipeline, present a modified classifier adhered to the real data and finally state the proposed future work. An in place working system is currently being tested under various conditions in the classrooms at the Institute.**

I. INTRODUCTION

The accuracy of face detection and recognition networks has been growing over time. Deep convolutional neural networks outperform all previous methods. More recent trend is towards training networks to naturally rank similarity between inputs Ex. Siamese networks [3]. FaceNet [2], a network to generate unique 128-d facial embedding for every unique face has been by developed by researchers at Google Inc. Similar to Siamese networks, FaceNet is trained to generate embedding such that the euclidean distance between the embedding of two similar faces is below a certain threshold in the 128-d euclidean space while two different faces lie farther away in space. The facial recognition task reduces simply to finding out euclidean distances and conforming similarities between two faces. While this seems to easy, the networks are often trained on specific datasets. Good results on the direct usage of the network for specific applications like classroom attendance cannot be guaranteed. In this project, we explore practical considerations of utilizing FaceNet for classroom attendance and demonstrate a full working pipeline of the facial recognition system with adaptions to real classrooms. We present an end to end system starting from data acquisition to classification algorithm adhered to the real data. Through a web application, the system is currently being tested in classrooms at IIT Tirupati.

II. DATA ACQUISITION

In order to implement the system in real time classroom environments, we collected the images of 33 students in various poses. The poses included top, left, right, front and down orientations. Images are taken in groups of four to reduce acquisition time. We then use a clustering algorithm defined in Section III to group faces corresponding to a single person. A sample image in shown Fig. 1.

Fig. 1. Dataset acquisition in groups of four. Right orientation shown.

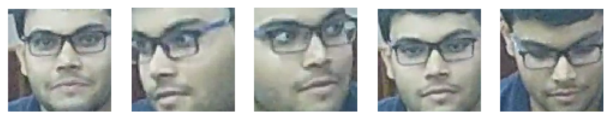

Fig. 2. Sample bin from the dataset.

III. DATASET GENERATION ALGORITHM

We apply dlib's face detection network [1] to detect faces followed by use of FaceNet [2] to generate 128-d facial embedding for every face. The detection and embedding generation are explained in section IV. The embeddings corresponding to 20 faces (5 poses X 4 persons) are clustered to save human effort of sorting faces. The clustering generates a unique bin for every person with five faces as shown in Fig. 2.

The algorithm begins with a unique bin for every person in the image. These bins are initialized with faces from a single image. For every remaining face, we assign a per bin score defined as the mean distance from all faces in each bin. The choice of face for addition to a bin is done as follows:

1) Consider the minimum score from all bin scores for each face.
2) Among all faces, assign the face with the minimum score obtained in step-1 to the respective bin. Call it as the assigned-bin.
3) Update the bin score corresponding to the assigned-bin for each remaining face and repeat until no face remains.

IV. FACIAL RECOGNITION PIPELINE

An illustrative figure for the pipeline in shown in Fig. 3. Next we describe the elements of the pipeline.

This work was done at Indian Institute Of Technology Tirupati.

[1]Akash Dhasade is a student at the Department of Computer Science & Engineering, Indian Institute Of Technology, Tirupati tee15b007@iittp.ac.in

[2]Saket Joshi is a student at the Department of Computer Science & Engineering, Indian Institute Of Technology, Tirupati tce15b020@iittp.ac.in

[3]Dr. Rama Krishna Gorthi is with Department of Electrical Engineering, Indian Institute Of Technology, Tirupati rkg@iittp.ac.in

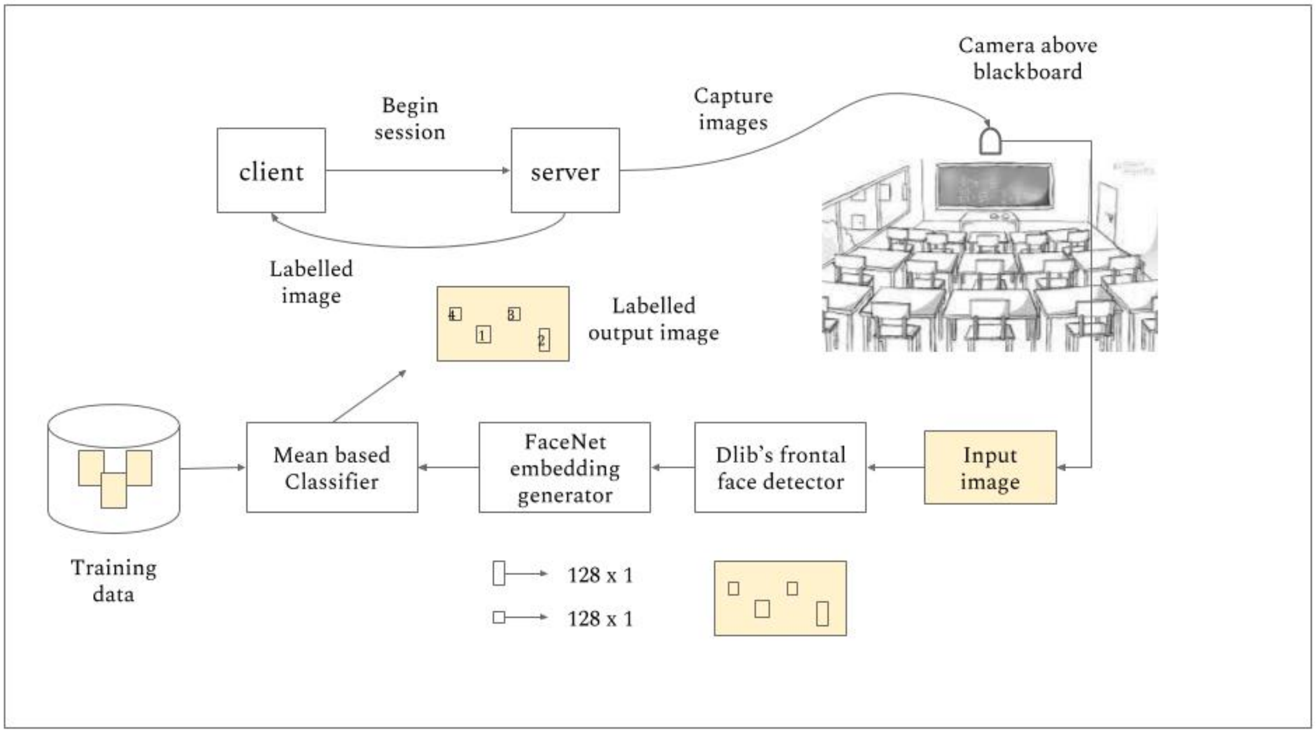

Fig. 3. Automated facial recognition system pipeline.

A. Training Data

The training data consists of a total of 165 face embeddings corresponding to the 5 pose variations of 33 students in the classroom. The data that needs to be stored is very minimal.

B. Face Detection

We use dlib's [1] state of the art frontal face detector to detect faces in the image. This deep CNN based face detector gives 100% precision with no false positives. The detection speed is more than one minute per frame when run on a CPU. However, we run the system on an NVIDIA Tesla K40c GPU with 12GB memory where the detection speed is 0.5s per frame.

C. Face recognition

The embeddings generated by FaceNet form unit spheres in the 128-d feature space. In other words, the embeddings of a single person are clustered around a hypothetical centre. We can assume the centre to be the mean of the data when the dataset is large. Most classifiers ignore this important piece of information about the prior distribution of data. The commonly used multiclass support vector machine classifier, decision tree classifier and also naive bayes classifier perform poorly on this data. Hence, the need of designing a new classifier. We consider the following two classification schemes:

1) **Simple mean classifier** Assign the face to the class with the nearest mean.
2) **Weighted mean classifier** Consider k nearest samples to the input sample for every class. Find the per class mean using the above k nearest samples and assign the face to the class with the nearest mean. k is a parameter that needs to be tuned.

The classifier design can be improved by considering the fact that we can achieve higher precision at the cost of skipping classification of some faces. The skipping can be effectively achieved by setting appropriate thresholds for the distance from the mean. Any two faces within the threshold will belong to the same class while faces above the threshold will be skipped from labelling. This scheme also allows us to label new faces of students (who are not a part of dataset) as "unknown" without making any extra changes to training or dataset as the scheme effectively skips labelling them. The classification scheme is illustrated in Fig. 4.

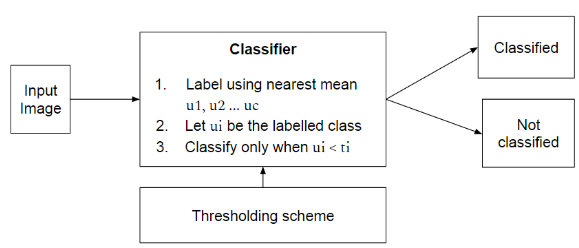

Fig. 4. Designing the classifier.

Figuring out thresholds While the classifier seems reasonably easy, setting the right threshold is equally difficult. The thresholds can be of two types,

1) Static thresholds: Common threshold for all classes.
2) Dynamic thresholds: Varying threshold for every class.

Still in the process of development of appropriate scheme for dynamic thresholding, we look forward to utilizing the answers to the following questions for devising the scheme.

1) How data points within class are spaced with respect to each other.
2) How data points within class are spaced with respect to the mean of the class.

One idea is to fix a constant precision P that we demand from each class and trade off recall for the precision demanded. This could be obtained by studying precision-recall curves for varying thresholds.

V. Web Application

We use the Amcrest pan and tilt PTZ camera with a python API [4] for our classroom attendance system. A simple web application was developed using the flask microweb framework. The application serves the purpose of sending requests to camera for recording the session, shows live processing of images and updates the list of students marked as present. A snapshot of the application is shown in Fig. 5. Still in prototype stage, the application is built with features sufficient enough for testing and verification.

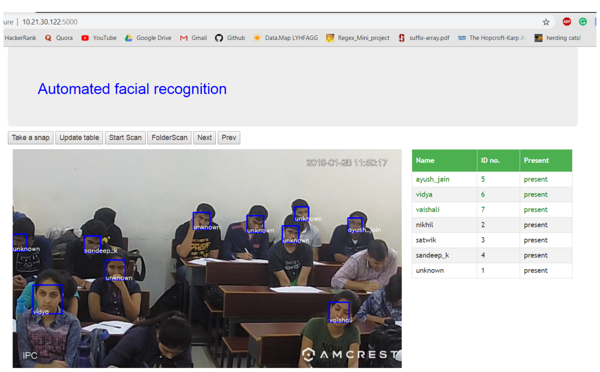

Fig. 5. A snapshot of the running application. Entries in green correspond to newly detected students as the session is being recorded. Faces marked as "unknown" correspond to faces whose distance was above the threshold as specified in the classifier.

VI. Future Work

As mentioned earlier, the system is being tested in the classrooms at IIT Tirupati. In the current stage, the facial recognition pipeline is able to mark 90% of the students in the class with a precision of 100%. Some results are shown in Fig. 6 and Fig. 7. Note that this result is only with respect to static thresholds and simple mean based classifier. The following tasks remain to be worked out,

1) Identify an appropriate criteria for dynamic thresholding. Test the classifier using dynamic thresholds. We expect better results using this scheme.
2) Defining the right aggregation strategy for granting attendance. For example, a simple aggregation strategy would be to mark as present only if the face is detected at least 10 times.
3) Run the tests using a camera with a better resolution.
4) Augment the dataset with extra faces as more and more sessions are recorded by designing an online algorithm for dataset augmentation.

Finally, we plan to devise a new architecture for one shot detection and recognition since the two tasks complement each other.

Fig. 6. Result Image-1

Fig. 7. Result Image-2. The four students on the first bench are not a part of the dataset and are correctly marked as "unknown" by the classifier.

Acknowledgment

We thank third year Computer Science students for their kind cooperation in dataset acquisition. We also thank the Computer Science & Engineering Department as well as the Electrical Engineering Department for the infrastructure facilities to set up cameras and GPU's.

References

[1] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009
[2] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. arXiv
[3] G Koch, R Zemel, and R Salakhutdinov. Siamese neural networks for one-shot image recognition. In ICML Deep Learning workshop, 2015.
[4] https://github.com/tchellomello/python-amcrest
[5] Fu, R., et al.: University classroom attendance based on deep learning. In: 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA). IEEE (2017)