Optimizing the design of semi-supervised facial recognition framework

submitted in partial fulfilment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

AKASH DHASADE CS15B031

SAKET DATTATRAY JOSHI CS15B034

Supervisor(s)

Dr. Rama Krishna Gorthi

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

APRIL 2019

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/-data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 10-05-2019

Signature
Akash Dhasade
CS15B031

Place: Tirupati
Date: 10-05-2019

Signature
Saket Dattatray Joshi
CS15B034

BONA FIDE CERTIFICATE

This is to certify that the thesis titled **Practical face recognition**, submitted by **Mr. Akash Dhasade** and **Mr. Saket Dattatray Joshi**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 19-05-2019

Dr. Rama Krishna Gorthi
Guide
Associate Professor
Department of Electrical Engineering
IIT Tirupati - 517501

ACKNOWLEDGEMENTS

Foremost, we would like to thank our project advisor, Dr. Rama Krishna Gorthi for guidance and support, which allowed us to work on and finish a project that we are proud of. We would like to acknowledge all faculty members for their contribution and suggestions which were very valuable.

We are thankful to the 2016 Computer Science & Engineering batch students and 2017 Electrical Engineering batch students for their kind cooperation in data acquisition. We would also like to thank the engineering unit and administration staff at IIT Tirupati for their cooperation and prompt support in installing required hardware for our project.

We are always grateful to our family for their support and the institute for providing us this opportunity and the all facilities required.

ABSTRACT

KEYWORDS: Face recognition (FR); Face detection; Attendance system.

In this work, we present the design of a complete face recognition (FR) system to automate classroom attendance. We first present a novel classroom dataset with 11056 face images of 61 distinct identities. To the best of our knowledge, the dataset is a first such dataset collected in practical scenarios like lecture halls. This dataset includes a separate training set collected in constrained settings with protocol, and an operation set collected over 12 lecture sessions in unconstrained settings serving as a benchmark to evaluate practical systems. We propose the binning algorithm and utilize clustering to perform semi-supervised data acquisition and training, thus minimizing human in the loop. A mathematical bound on relation between face verification and recognition accuracy is presented with a comprehensive discussion of factors at play in obtaining better recognition accuracies. We then propose a new KMeans Classifier adhered to the problem setting, providing comparable performance to traditional classifiers but with additional advantages. We perform a comprehensive evaluation of the state-of-the-art FR models and report the face verification and face identification accuracies on the classroom dataset. Using the best model, the designed KMeans classifier and a final aggregation scheme to grant attendance, we report the classroom attendance over all 12 sessions. Through a web-app, the system is operational in three classrooms at IIT Tirupati. The system is shown to be practically feasible with a low effort of implementation, and high performance and scalability.

The intention of this project is to bridge the gap between the state-of-the-art research in FR technology and practical application. We thus make both theoretical contributions while working on a scalable solution to a pragmatic use-case for the society.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

FR	Face recognition
SENet	Squeeze and excitation network
DL	Deep Learning
NN	Neural Network

CHAPTER 1

INTRODUCTION

Over millions of years, the human brain has evolved a keen ability to detect and recognize faces. In computer science, facial recognition is a unique problem in it's own right. There are interesting facets to the problem which have allowed it to enjoy an abundance of attention from the industry, academia and the public. The problem definition is not too esoteric for public understanding, and presents a myriad of applications. At the same time, it stood for a long time as one of the most difficult problems in computer vision, standing as a benchmark problem for the potential of machine learning. The gift of face recognition technology came with the spring of machine learning, as key advancements and discoveries such as convolutional neural networks were made. However, hitherto the technology is largely limited to large companies and research settings. There are several challenges to be addressed in implementing a practical system. The primary objective of our project is to systematically identify and address these challenges from a research perspective. We study and work with the state-of-the-art research and methodologies available, specifically two deep networks, squeeze and excitation network (SENet) and a deep convolutional neural network, FaceNet. Further, we exemplify our research by implementing a complete system to automatically recognize people and grant attendance in lecture halls or similar settings.

Granting attendance to students has always been cumbersome due to issues like proxy attendance, time consuming roll calls, etc. In proxy attendance, we find students signing for their friends in their absence. On the other hand, roll calls consume significant amount of total time over the entire course. Five daily minutes scale to 200 minutes over 40 lectures implying that for every course at least three lecture hours are overlooked, let apart the fact that five minutes is the minimum considered time. We address this issue by using deep learning to grant attendance through automated face detection followed by face recognition. The system we propose includes a camera at the centre of the class which takes burst images or a short video of the class while the lecture is in progress. The images are processed to detect faces of all the students in class and matched against a pre-stored database to recognize individual students. A web-app enables the professor to

verify attendance at the end of the class and mark/unmark students in peculiar uncertain cases. The attendance can be compiled and accessed by administrators.

A surrogate approach to this problem employs fingerprint-scan based bio-metric for identification. This system has been popularly employed in offices, as a part of a security system which is mounted near the entrance and controls door access. However, such a system has several limitations, especially for lecture attendance. A major limitation is that in universities the system is very easy to cheat or manipulate, since the system cannot ensure that the person attended the lecture. An alternative is to circulate the machine during the lecture. However, the FR based system we present is differentiated since it is non-intrusive and completely automated. Cameras are being increasingly installed in premises for security purposes, the required hardware is getting cheaper and serves multiple utilities.

There are several technical challenges in building a complete attendance system. The accuracy of face detection and face recognition frameworks have been growing over time, however, there is a dearth of literature on training, employment, design aspects, and performance analysis of these techniques in practical environments. The classroom environment presents an unconstrained setting where students do not pose for the camera explicitly. As a result of which facial recognition becomes challenging with side profiles, dropped heads, small sizes of detected faces, etc. This is in contrast with the norm of testing new models on data-sets predominantly composed of images compiled from the internet. Our primary focus in this project is to deal with above-mentioned issues and build a highly robust attendance system.

1.1 Objectives and scope of the thesis

This project involves challenges in several different domains, including; statistical and mathematical analysis, computer vision and machine learning, system design and performance evaluation, software development, and practical work for hardware and data collection. In this section, we outline the objective, approach and contributions for the project.

Objective : The theme of the project is to contribute practical and research aspects allowing facial recognition technology to be employable in practical use cases. This

includes designing a complete facial recognition pipeline, meticulous experimentation and evaluation of the pipeline followed by building a robust system for automated attendance in the lecture halls.

Sub-problems : Following points in order summarize our sub-problems and approach -

1. Standard datasets differ greatly from production environments. We begin with a creation of classroom dataset gathered over several lectures. The dataset includes separate train and test set allowing for extensive experimentation in practical settings.
2. State-of-the-art facial recognition models have reached new heights of performance. Two such famous models: Facenet and SENet form our direct baseline. We evaluate the models, verifying their robustness and present algorithmic changes adhered to the problem setting to improve performance.
3. Exploration of several individual components forming our system pipeline. This includes clustering for data generation, systematic procedures for data acquisition and labelling, understanding verification and recognition systems, aggregation scheme for granting attendance, etc.
4. Designing a system that integrates all the above-mentioned components. A web-app to serve as an end-to-end application with functional implementation in IIT Tirupati lecture halls.

CHAPTER 2

Literature Review

The task of face recognition (FR) has a rich history in techniques and performance. Deep learning made a great impact, and largely dominated the field of FR. Since the introduction of DeepFace in 2014 [14], FR has seen continuous innovation and improvement with the introduction of new deep learning based techniques. In this literature review, we summarize the important techniques and components which shape the state-of-the art in FR.

2.1 Face detection and alignment

Face detection and alignment is requisite to a wide variety of applications. Viola Jones face detection algorithm [15] is considered to be the first algorithm to have achieved reasonable accuracy and computational efficiency in face detection. The paper introduced the notion of integral images for calculating haar features in constant time while it utilized a cascade of classifiers for efficient detection. Recent techniques employ convolutional neural networks for the task, first generating candidate bounding boxes, followed by refining the candidates and calibrating the bounding boxes. These techniques have improved performance under occlusions, variations in illumination conditions and poses [17]. Another popular method is based on histogram of oriented gradients (HOG) which offers higher computational efficiency at the cost of slightly reduced performance [3] [2].

Face alignment is also an important pre-processing technique employed before images are fed to neural networks. Multiple studies have shown improved performance by using face alignment [16]. There are two popular categories of approaches used, regression-based, and template fitting. K. Zhang, Z. Zhang, Z. Li and Y. Qiao introduced a new framework to integrate the tasks of face detection and alignment using unified cascaded CNNs by multi-task learning (MTCNN) [20]. The CNN is designed to be lightweight allowing real-time execution. We experimented with MTCNN and dlib's deep CNN based face detector [8].

2.2 Face recognition

The field of face recognition first gained traction with the introduction of the Eigenface approach in the late 1990s. The field landscape then changed with the introduction of AlexNet in 2012 and DeepFace in 2014, moving from handcrafted features to a deep learning based approach and presenting pronounced performance improvements over previous methods. In 2018 paper [17], Wang and Deng explicate the development of deep learning in FR, and review it's effect on aspects such as algorithm designs, data-sets and evaluation protocols.

We begin with presenting the sizes of standard datasets used by industry and academia to train FR models in Table-1. The largest dataset procured for face recognition comprised of 200M images of 8M distinct identities by Google. Several of the other datasets also range in the order of few millions and were mostly owned by industry giants. Publicly available datasets are two orders of magnitude smaller than these datasets. Thus, dataset was one big bottleneck for academic researchers preventing further experimentation and improvement of recognition models. The Visual Geometry Group from the University of Oxford procured a dataset that featured wide variations in pose, age groups, ethnicity, etc, while keeping the size within limits for academic experimentation. The VGG dataset [11] with 2.6M images was released in 2015. It was followed by an improved dataset featuring more variations, VGGFace2 [1] with 3.3M images in 2018. VGGFace2 is the current state-of-the-art publically available dataset for training FR models. The traditional datasets for testing are LFW [5] and Youtube Faces DB [18]. While most papers report only face verification accuracies on these two datasets, the IARPA Janus benchmark (IJB) dataset is considered to be more difficult for both face verification and face identification. [9].

Dataset	Identities	Images
LFW	5749	13233
WDRef	2995	99773
CelebFaces	10177	202599
VGG	2622	2.6M
Facebook	4030	4.4M
Google	8M	200M
VGGFace2	9131	3.3M

Table 2.1: Face recognition datasets.

We next describe in further detail the two candidate FR models which were part of our experimentation.

FaceNet In 2015, researchers from Google introduced the FaceNet architecture [12]. In contrast to a majority of previous approaches based on deep networks using a classification layer over a known set of face identities, FaceNet is trained to output a 128 dimensional embedding using a triplet based loss function. The embedding can be used to compare faces based on euclidean distance. This approach provides several advantages by separating the classifier training from the feature embedding generation. The model was trained using a private data set comprising of 200M images as mentioned in Table-2.1. The architecture improved state-of-the-art results on the popular LFW (99.63%) and Youtube Faces DB (95.12%) datasets. Implementations of the architecture are popularly used, hence we chose it as one of our candidate FR model.

ResNet-50 with squeeze-and-excitation blocks The researchers from the Visual Geometry Group trained a ResNet-50 network with and without Squeeze-and-Excitation blocks (called SENet) on the VGGFace2 dataset. The training procedure comprised of a softmax loss training for all the different classes followed by training using triplet loss. Prior training using softmax ensured faster convergence. SENet stood as the winner of the ILSVRC 2017 challenge. Subsequently, even for the face recognition task, the model exceeded the previous best performance on the IJB datasets. The authors report an identification accuracy of 98.2% for Rank-1 identification on the IJB-A dataset. Three important reasons made this model a prime candidate for our study:

1. VGGFace2 includes a large number of distinct identities and images featuring wide variations in ethnicity, age and poses.
2. The model represents the latest state-of-the-art models at the time of our study.
3. The model is evaluated for both *face verification* and *face recognition*; and the model and data set are publicly available.

2.3 Attendance systems

Attempts to build attendance system include [10], [7] and [4]. In [10], discrete Wavelet transform and discrete Cosine transform are used to extract facial features while radial basis function is applied for recognition task. The recognition accuracy is reported to be 82% on their collected data set. The majority of the previous work is evaluated in

small constrained settings or reports limited success. Few works can be found employing current state-of-the art techniques in unconstrained environment settings. [7] Suggests the use of continuous observation to improve accuracy. The face detection rate over a period of 79 minutes is evaluated to be 80% while the precision score for attendance is reported to be 70% on their data set. Rong Fu & et. al., developed a system integrating multi-task cascade CNN with Center-Face face recognition to grant classroom attendance [4]. The system could process one frame in 100 ms while the accuracy of facial recognition model was obtained to be 98.87%. However, the evaluation is based on a constrained environment with a small number of subjects. Most of these works do not discuss the systematic steps to be followed in building a practical facial recognition pipeline.

In addition, there are several industrial or company based products for FR, however, we could not find implementation or evaluation for a similar task.

CHAPTER 3

Proposed design of face recognition system

The system designed for classroom attendance is currently installed in three classrooms at IIT Tirupati. This section provides a brief overview of the end-to-end system pipeline, introduces all major components and describes implementation details. In Chapter-4 and Chapter-6, we further explicate the design decisions and motivating factors behind the key components of the system.

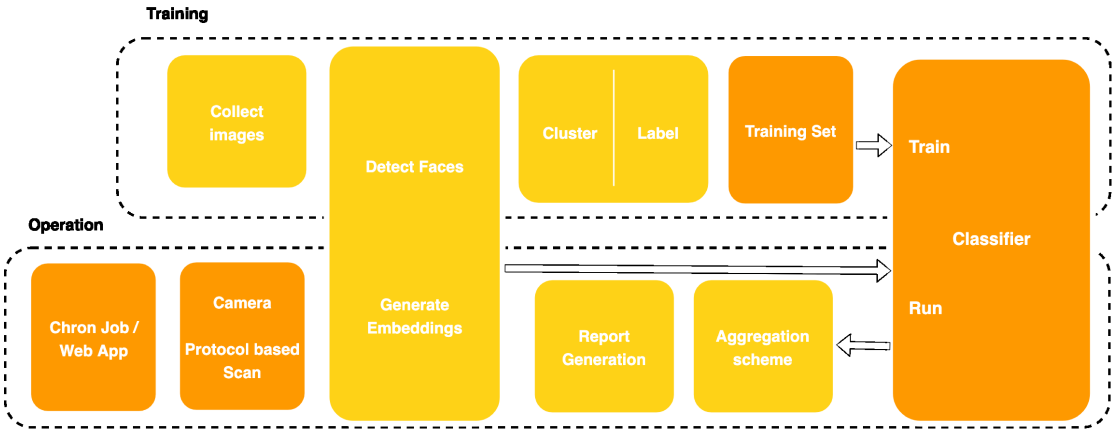


Figure 3.1: System pipeline

System pipeline: The system operates in two phases: training and operation phase. The design objective is to ensure the best performance while designing a streamlined protocol following all constraints. Fig. 3.1 depicts the system pipeline. For an institute with several hundred students, it is necessary that the process of data acquisition be streamlined and efficient. We design the training phase with this objective in our mind. The operation phase, as the name suggests runs the end-to-end system in real time. Several modules overlap between the two phases. Each module is described next.

3.1 Image collection

The image collection module in the training phase employs different strategy than the camera protocol scan in the operation phase. The target here is to gather several facial images of students to form the firsthand training dataset. Note that we use the word

'firsthand' here to indicate that the training data will be updated as more and more data becomes available. The students are seated in groups of around four people and pose for image capture. The process is then repeated for the next set of students. We discuss in detail the advantage of this strategy and rationale behind group image capture in Chapter-4.

3.2 Face detection and embedding generation

This module is common to both the training and operation phase. It consists of a face detection neural net followed by a face recognition model to generate face embedding. We experiment with MTCNN and Dlib's deep face detector for face detection and Facenet and SENet for embedding generation. In the training phase, the generated embeddings are used to train a classifier while in the operation phase, the embeddings are fed to classifier for labelling.

3.3 Clustering and labelling

The actual speedup in data acquisition is obtained by several design decisions in the clustering and labelling module. As mentioned earlier, we obtain face embeddings for students in groups. An unsupervised clustering algorithm then segregates the face embeddings for each student with the number of clusters made equal to the number of students. The clusters are then labelled with student ids. This prevents repeating the same procedure individually for each student. The number of students in the group is subject to camera location, acquisition conditions and accuracy of clustering algorithm. We achieve a minimum of 4x time speedup by employing clustering in the data acquisition process.

3.4 Training set

The segregated embeddings labelled with student ids form the training set. The constructed training set has a direct correlation with the performance of the system. The variations captured by training set affect the final accuracy of recognition. Hence, it

forms a crucial component of the entire pipeline. The statistics of the training set are presented in Chapter-5.

3.5 Classifier

The training set is used to build the classifiers for face recognition. Properties of the embeddings generated such as the similarity metric, dimensionality, the size of dataset affect the performance of different classifiers. We experiment with several classifiers and also present the design of a new classifier: KMean Classifier. The classifier module uses the best classification scheme evaluated empirically. For every input face embeddings, the module returns a face label and the corresponding confidence score for the label.

3.6 Web App

To ensure user friendliness, we made a web application which handles all functionality end-to-end. The web-app supports initiating the camera protocol scan and running the system during a lecture hour to displaying aggregate attendance for the lecture. It forms the single point of contact for the users of the system. We discuss implementation and functional details of the web-app in Chapter-7.

3.7 Camera protocol scan

This module is responsible for collecting images in the operation phase. We installed a camera at the center head of each classroom to capture images of sessions in progress. There are two main objectives to satisfy. The first is that the camera should be able to capture all students. For large diameter classrooms, the camera must thus have pan and tilt capabilities. The second objective is that we should get a good angle of the face to facilitate facial recognition. This means the camera should be installed so that all faces are visible (at least at some point of time during the scan). If the camera is positioned too high, we tend to get headshots, whereas if too low, the visibility range of students is obstructed. We found the center of the blackboard to be an ideal location for the installation of the camera. We use an Amcrest PTZ camera with 2048px resolution and

pan-tilt-zoom capabilities. A higher resolution may be important for a larger classroom to ensure sufficient image quality for students seated at the back.

Furthermore, we employ a camera protocol that operates in a grid fashion. The protocol captures one row spanning the full width of the classroom in one run. In the next run, the camera is zoomed appropriately to capture the next row. The process is repeated until all rows of the classroom are captured with sufficient face resolution for recognition purposes.

3.8 Aggregation scheme & report generation

The images obtained from the camera scan are fed into the detection-embedder module followed by classifier for labelling. The robustness of attendance system depends upon the performance of classifier. Note that many faces will be captured for every student and not all faces will be labelled correctly by the classifier. Hence, it becomes important to devise an aggregation scheme based on the labels and scores returned by the classifier. We explore and implement one such scheme that maximizes the precision and recall for granted attendance. The scheme is presented in Chapter-8: Performance and Results. A final report that marks each student as present or absent is then generated along with an aggregate confidence score for the session. Based on these scores, the teacher can then check for one or two uncertain cases.

In this Chapter, we presented a brief overview of each module in the system pipeline. Over the next few chapters: Chapter-4 and Chapter- 6, we venture into the algorithmic details of each involved module. In Chapter-5 we present our classroom dataset and in Chapter-7 we discuss the implementation of the web-app. Finally in Chapter-8, we present an extensive performance analysis of each module on our dataset. We conclude with limitations and future work in Chapter-9.

CHAPTER 4

Data acquisition and clustering

4.1 Need for clustering

The training data for face recognition models is a set of labelled faces. While we can assume that such-labelled data is present in most practical scenarios, it might be difficult to acquire training data that is similar to the true data that the model will be fed in real production environments. For instance, it is easy to acquire random face pictures for every person, but difficult to acquire face pictures of the person in the same environment, i.e in the lecture hall. For an automated attendance system, this gap becomes crucial since the facial recognition pipeline has to operate in a classroom environment with low quality images. Clustering is an attempt to reduce this gap by allowing the generation of more training samples which adhere to the test samples being fed to the model. Unlabelled faces acquired from classroom images are clustered to group face images belonging to the same person. The generated dataset can be used as a first hand training data or can be augmented to the original training data with least possible human interference.

4.2 Clustering algorithms

We formally state the problem next and then describe an algorithm for effective clustering, which we refer to as the Binning algorithm. Traditional clustering algorithms require the number of clusters to be known apriori, for ex. KMeans clustering. We impose this condition in a slightly stronger form by assuming that there exists at least one image where all faces are detected. Each face belonging to that image then represents a unique class, acting as a seed for the algorithm. Secondly, we require the algorithm to cluster such that no two faces belonging to the same image be labelled the same. Hereafter, we use the word bin and class interchangeably. The problem setup is as follows: let f_j^i denote the j^{th} face obtained from i^{th} image. The goal is to assign an integer k to each face such that all faces belonging to the same person are assigned with the same integer. f_j^i is a face object with the following attributes:

- *enc* - Fixed length encoding or face descriptor of the face obtained using modern networks like Facenet or SENet.
- *bin_scores* - Holds the mean scores for every bin.
- *max_index* - Index of bin with max mean score.
- *score* - Max mean score among all bins.
- *img_index* - Index of image to which the face belongs.

Binning algorithm: The binning algorithm is an agglomerative (bottom-up) clustering algorithm, particularly an average linkage clustering algorithm. Similar to UPGMA (unweighted pair group method with arithmetic mean) [13], the algorithm merges two nearest clusters to form a higher-level cluster. The definition of *nearest* varies from algorithm to algorithm. In UPGMA, *nearest* distance between two clusters X and Y is defined as the average of all distances $d(x,y)$ between pairs of objects $x \in X$ and $y \in Y$. Thus, we have $D(X,Y) = \frac{\sum_{x \in X} \sum_{y \in Y} d(x,y)}{|X|+|Y|}$. There exist slight differences between binning algorithm and UPGMA which are listed next.

1. Unlike UPGMA, in each iteration of binning algorithm, a new face is merged with one of the face clusters already formed. Since we begin with the assumption that one element from each different cluster is known before the start of the algorithm, the face clusters are not merged themselves.
2. An additional constraint that no two faces from the same image are labelled the same is imposed during the clustering process.

Notation

I: Set of all images; C: Total number of classes

F: Set of all faces; N: Total number of images

Constraint No two faces from the same image are labelled the same.

$label(f^i_j) \neq label(f^i_k)$ *if* $j \neq k$

Condition for Binning algorithm There exists at least one image where faces corresponding to all C classes are detected.

$\exists I^n \in I$ s.t $|\{f^i_j \; where \; f^i_j \in F \; \& \; i = n\}| = C$

A similarity matrix between all pairs of faces can be precomputed before the be-

ginning of the algorithm. The similarity measure could either be cosine similarity or euclidean distance. For each unlabelled face, an average score corresponding to every bin is maintained throughout the clustering process. The index of the bin with the max score represents a probable assignment for the face. Out of all such unlabelled faces, the face with the maximum score for most probable assignment is merged with the respective bin. Average scores for all faces corresponding to the recently merged bin are then updated. An auxiliary list named *images* is maintained to check the constraint stated above. The algorithm repeats until all faces are merged. The pseudo code for the algorithm is presented next. We also experiment with KMeans clustering, COP KMeans, and agglomerative clustering algorithms. Performance of the algorithm in comparison to other clustering algorithms is presented in Section-8.2.

Algorithm 1: Binning algorithm: SIM FUNCTION

\\Cosine similarity
def *sim(f, f')*:
    return $\frac{f.enc^T * f'.enc}{|f.enc| * |f'.enc|}$

Algorithm 2: Binning algorithm: INITIALIZATION

bins $\leftarrow$ List of C empty lists
imgs $\leftarrow$ List of N empty lists
\\Necessary condition
Choose $I^n \in I$ s.t $|\{f_j^i$ where $f_j^i \in F \ \& \ i = n\}| = C$
for *b in* $0 : C-1$ **do**
    bins[b] $\leftarrow$ **append**(*bins[b]*, f_b^n)
    imgs[n] $\leftarrow$ **append**(*imgs[n]*, *b*)
endfor
$F \leftarrow F \setminus \{f_j^i$ where $f_j^i \in F \ \& \ i = n\}$
for $f \in F$ **do**
    for *b in* $0 : C-1$ **do**
        f. bins_score[b] $\leftarrow$ **avg**{**sim**(*f, f'*) $\forall$ $f' \in$ *bins[b]*}
    endfor
    f. score $\leftarrow$ **max** *f. bins_score*
    f. max_index $\leftarrow$ **argmax** *f. bins_score*
endfor

4.3 Acquisition of training data

Dataset collection is an important facet of the system pipeline. The process used determines the type, quality and quantity of training data available, which in turn

Algorithm 3: Binning algorithm: MAIN LOOP

```
1  while not empty(F) do
2      F ← sort(F, key = \f → f.score)
3      F ← reverse(F)
4      top ← head(F)
5      F ← tail(F)
6      b ← top. max_index
7      i ← top. img_index
8      if b in imgs[i] then
9          top. bins_score[b] ← −top. bins_score[b]
10         top. score ← max top. bins_score
11         top. max_index ← argmax top. bins_score
12         F ← append(F, top)
13         continue
14     else
15         bins[b] ← append(bins[b], top)
16         imgs[i] ← append(imgs[i], b)
17         for f ∈ F do
18             prev ← f. bins_score[b]
19             if prev ≥ 0 then
20                 n ← len(bins[b])
21                 f. bins_score[b] ← (prev * (n − 1) + sim(f, top)) / n
22                 if f. bins_score[b] > f. score then
23                     f. score ← f. bins_score[b]
24                     f. max_index ← b
25                 end
26             end
27         endfor
28     end
29 end
```

may bear strong correlation to the performance of the system. In practical application scenarios, there may be several limitations to the training data available. In addition, there may be non-conformity or incongruence in the conditions under which training data was collected and where the system must operate. There are two primary objectives,

1. Minimize the manual interference required.
2. Maximize and ensure sufficient quality and quantity of data

One possible solution is to collect images during session. To label all faces, clustering can be employed, followed by manual correcting and labelling. However, we found two issues with this approach: (1) The clustering performance degrades very quickly with the number of people. We also tried constraint based clustering, adding the constraint that faces in the same image belong to different people. However, we found that the obtained performance still requires a large amount of manual correction and labelling. (2) The quality of images collected is very low since the images are collected in unconstrained settings. In addition, there is no standardization on the number of images collected per person.

Figure 4.1: Dataset acquisition in groups of four. Right orientation shown.

We now describe the protocol we followed to collect a training dataset. More time consuming the process, the difficult it is to scale to several batches or classrooms of students. Since we need several images for each person, the time required to acquire face images of thirty students may be significantly high. The problem amplifies when scaled to an Institute with several hundred students. In order to reduce the time spent in acquisition, we employ clustering to group faces. The steps used are outlined next.

1. Images are taken in several poses, including top, down, right, left, extreme right,

etc. for a group of students (generally four) at once. Fig-4.1 depicts a sample image.

2. Faces are detected and 128-d embeddings are generated for all faces using Facenet.
3. With $k = 4$, KMeans clustering is used to cluster faces using the face embeddings obtained in step-2.
4. Face clusters are written to separate directories and verified manually for errors. Fig-4.2 depicts five faces in different poses obtained post clustering.
5. Directories are relabelled with student ids.

Figure 4.2: Sample faces from a cluster in dataset.

KMeans clustering gives 100% accuracy with the above procedure. The steps are then repeated for another group of four students. With the use of clustering, the acquisition time reduces by a factor of four. The size of the group can be varied subject to the camera location, acquisition conditions and accuracy of clustering algorithm. In the following chapter, we present our novel classroom dataset.

CHAPTER 5

Dataset

In this chapter, we present a novel classroom dataset with 11056 faces of 61 distinct identities. We describe the details of *train-set* and *operation-set* that constitute the dataset and then discuss several applications of the dataset. We begin with the need of procuring such a dataset.

5.1 Need and essence of dataset

The majority of standard datasets available today [5, 11, 1], popularly used to report face verification and face identification accuracies are based on celebrity images found on the internet. However, there hardly exist any datasets which were collected in settings such as lecture hall. The goal of the dataset is not just to obtain faces in a new setting for validating facial recognition models, but also to mark attendance for students using a finite set of images captured during the lecture hour. There are several implicit assumptions with the acquired dataset when the later goal is sought. They include,

1. Captured images include at least a few non occluded frontal faces for every student present in the class. Camera positioned too high or too low might fail to give frontal faces.
2. The camera protocol is good enough to ensure equal distribution of capture. Too many images from a single section of classroom can affect final accuracy.
3. The resolution of capture is sufficient to meet recognition requirements. Camera must be zoomed in to acquire faces from last benches in the classroom.

The final accuracy obtained for attendance task is dependent not just on the robustness of the facial recognition model but also on the crucial assumptions listed above. Failing to meet these requirements would result in indirection of efforts towards bettering facial recognition model with little improvements in accuracies whatsoever. The acquired dataset is consistent with the listed assumptions and facilitates bench-marking of facial recognition as well as attendance systems. The overview of dataset is presented next followed by several applications and research directions facilitated by the dataset.

5.2 Overview of dataset

The dataset includes two batches of students: batch-1 with 33 students and batch-2 with 32 students. For each batch, the dataset includes a *train-set* and an *operation-set*. The *train-set* features variations under controlled conditions and is acquired using the procedure defined in Section 4.3. The *operation-set* includes seven sessions of classroom lectures for batch-1 and five sessions for batch-2 captured on different days. The notable characteristics of the dataset are as follows,

1. The sessions vary in number of students present in the class, position of students and classroom conditions.
2. The images in *operation-set* are captured without prior knowledge to students. In other words, students do not pose for the camera explicitly. Thus, the dataset presents a fully uncontrolled setting.
3. In order to study whether new students joining the class randomly affects the accuracy scores for primary students, we include only 28 out of 32 students in the *train-set* of batch-2. Thus, the remaining four students (if present) constitute *new faces* unseen by the model. Sessions of batch-1 are also randomly joined by new students not a part of dataset.

The statistics of the *train-set* and the *operation-set* are summarized in Table-5.1 and Table-5.2 respectively. The dataset was labelled by the authors of the paper using a mix of automation and manual labor. Firstly, the best classification scheme was run to generate probable labels to each face. Secondly, the labels were corrected in several iterations through semi-automated programs. All *new faces* were labelled using the same unique id. In summary, the dataset contains 11056 images of 61 distinct identities obtained directly from the classroom environment.

Batch	Students	Average faces per student
Batch-1	33	29
Batch-2	28	37

Table 5.1: Details of *train-set*.

5.3 Applications of the dataset and possible research directions

- Analyze the performance of the state of the art facial recognition models trained on standard datasets in a real world usecase.

	Batch-1				Batch-2			
Session No.	Images	Faces	Present students of 33	Total students	Images	Faces	Present students of 28	Total students
Session 1	99	722	32	34	230	862	26	29
Session 2	119	1165	32	36	230	1113	28	32
Session 3	127	412	31	31	230	897	26	29
Session 4	155	603	29	29	230	1152	26	30
Session 5	86	850	27	27	230	1304	27	32
Session 6	108	669	30	30	-	-	-	-
Session 7	149	1307	32	35	-	-	-	-

Table 5.2: Details of *operation-set*.

- Explore and resolve several problems in an unconstrained setting.
- Bridge the gap between small and constrained train data and unconstrained test data by experimenting with data augmentation, online learning algorithms, etc. over multiple sessions of image capture.
- Bench-marking of facial recognition based attendance systems.

5.4 Further details

All images were taken with a prior consent of the subjects involved. The consent form can be found in the Appendix section. The dataset is not made publicly available yet. Please contact the authors for more information.

CHAPTER 6

Face verification to recognition : Proposed approach

As part of our work, we explored the general relation between face verification and recognition performance. Face verification is the more popular problem used for evaluation in literature. This chapter discusses the difference between the two problems, and introduces a performance bound if a verification system is exclusively used to perform recognition. This analysis also draws insight into the design of Facenet's harmonic triplet loss function, and the advantage of a euclidean distance based embedding. In the case of VGG's SENet, the cosine similarity measure acts as a verification system.

6.1 Difference between verification and recognition

A system trained on a training set T of labelled faces can be asked two different questions.

1. **Face verification** - "Given two facial images, do they belong to the same person?"
2. **Face recognition** - "Given a facial image, who does it belong to?"

The difference between the two is not subtle for both performance evaluation and using one system for the other task. Popular datasets such as LFW have a testing protocol which measures accuracy as follows.

---

(1) $T(d) = \{(i,j) \in P_{same},\ with\ D(x_i,x_j) \leq d\ or\ (i,j) \in P_{diff}\ with\ D(x_i,x_j) > d\}$

(2) $Accuracy\ = \frac{|T(d)|}{|P_{same}|+|P_{diff}|}$

P_{same} and P_{diff} denote the set of pairs belonging to the same and different classes respectively. Variable d denotes the threshold chosen for classification and $D(x_i,x_j)$ denotes the euclidean distance between the pairs x_i and x_j. To measure recognition accuracy, consider a machine M which produces label y_i for an input sample x_i. If y_i^* is the correct class label for sample x_i and T denotes the test set, then the recognition accuracy is given by:

(3) $S = \{x_i, \forall x_i \in T \text{ where } M(x_i) = y_i^*\}$

(4) $Accuracy = \frac{|S|}{|T|}$

Accuracies in equation (2) and equation (4) for a system are related, yet fundamentally different. To create a face recognition system, a classifier is trained on the embeddings generated. In Facenet, the triplet loss function is designed such that generated embeddings of the same person should cluster in euclidean space. In the case of ResNet, we use the cosine distance as similarity metric, and the classifier we train essentially works on multiple validation problems. It is thus interesting to investigate the relational bounds on recognition accuracy which can be obtained from a system which has access to a validation system.

6.2 From verification to recognition: bound analysis

In this section, we derive an accuracy bound on the recognition problem for a system which works on a verification machine.

The verification machine: For a set of facial images X, Let matrix M be defined as, then $\forall x_i, x_j \in X$,

$$M(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ belong to the same person} \\ 0, & \text{otherwise} \end{cases}$$

We then define the verification machine $M_v(f_1, f_2)$ which takes two inputs f_1, f_2, outputting 1 if the inputs belong to the same person, and 0 otherwise. The reported accuracy of this machine is α_v, which is the probability that $M_v(f_1, f_2) == M(f_1, f_2)$

The recognition machine: From the verification machine M_v, we build the ideal recognition machine using Bayesian probability theory. Below problem describes the recognition machine for the simple case where each person class consists of only one for training.

Problem A - Given test sample x, n images from different classes Y, which class does x belong to?

We use the machine M_v to compare the test sample to images of each class. The following two cases arise:

1. The machine works correctly on all inputs, producing 1 for the correct class, and 0 for the incorrect class. The probability of this is α^n.
2. The machine incorrectly produces 0 for the correct class, or 1 for 1 or more incorrect class. There is no information to discriminate between the positive-tested classes better than natural odds. We will consider this case as a fail for the recognition system.

The accuracy of this system exponentially decreases with the number of classes. Even for a high accuracy of $\alpha_v = 0.98$, the recognition system reaches an accuracy of 0.54 for 30 classes.

We now consider the case where multiple test samples per class are available. This problem illustrates the significant difference which may manifest in the accuracies.

Problem B - Given m known samples for each class, C classes, classify an unknown sample x. Again, we compute $M_v(x,t)\ \forall t \in C_i \forall i$. The highest accuracy classification scheme classifies x into the class with the most positive matches.

Let j be the true label of sample x. Then,

$$P(len(\{M_v(x,t) == 1, \forall t \in C_j\}) == b) = \binom{m}{b} \alpha_v^b (1-\alpha_v)^{m-b} \tag{6.1}$$

For all other classes,

$$P(len(\{M_v(x,t) == 1, \forall t \in C_i, i \neq j\}) == b) = \binom{m}{b} (1-\alpha_v)^b \alpha_v^{m-b} \tag{6.2}$$

Using the classification scheme, the probability of the correct class yielding *more* positive results than all other classes is then,

$$\begin{aligned} \alpha_r &= P(len(\{M_v(x,t) == 1, \forall t \in C_j\}) > len(\{M_v(x,t) == 1, \forall t \in C_i, i \neq j\})) \\ &= \sum_{i=1}^{i=m} \left(\sum_{j=0}^{j=i-1} \binom{m}{j} (1-\alpha_v)^j (\alpha_v)^{m-j} \right)^{c-1} \alpha_v^i (1-\alpha_v)^{m-i} \binom{m}{i} \end{aligned} \tag{6.3}$$

The given equation relating α_r and α_v is based on an interpretation of the verification machine similar to naive bayes. There is assumed to be no correlation or biases in

the system. The error in the outcome is assumed to be independent and equal in true positive and true negative samples. However, the relation is important since it illustrates the dynamics between the number of training samples per class, and the number of classes. As the number of classes increase, a greater number of training samples per class are required to maintain the same precision. This equation shows how much detection and recognition accuracy can vary. Fig-6.1 charts values of α_r for different values of α_v and classes (c) for $m = 1$ and $m = 5$. For $m = 1$, the recognition accuracy expeditiously degrades with the number of classes as expected. Plot for $m = 5$ shows that small difference in A_v can significantly change A_r as number of classes increase. Note that there is a precarious balance between the three variables; the verification accuracy, number of classes, and the number of training samples per class. If there is high correlation between training samples, effectively reducing the quality of training data, this leads to a reduction in verification accuracy and a decrease in the effective number of training samples, consequently leading to deteriorated performance.

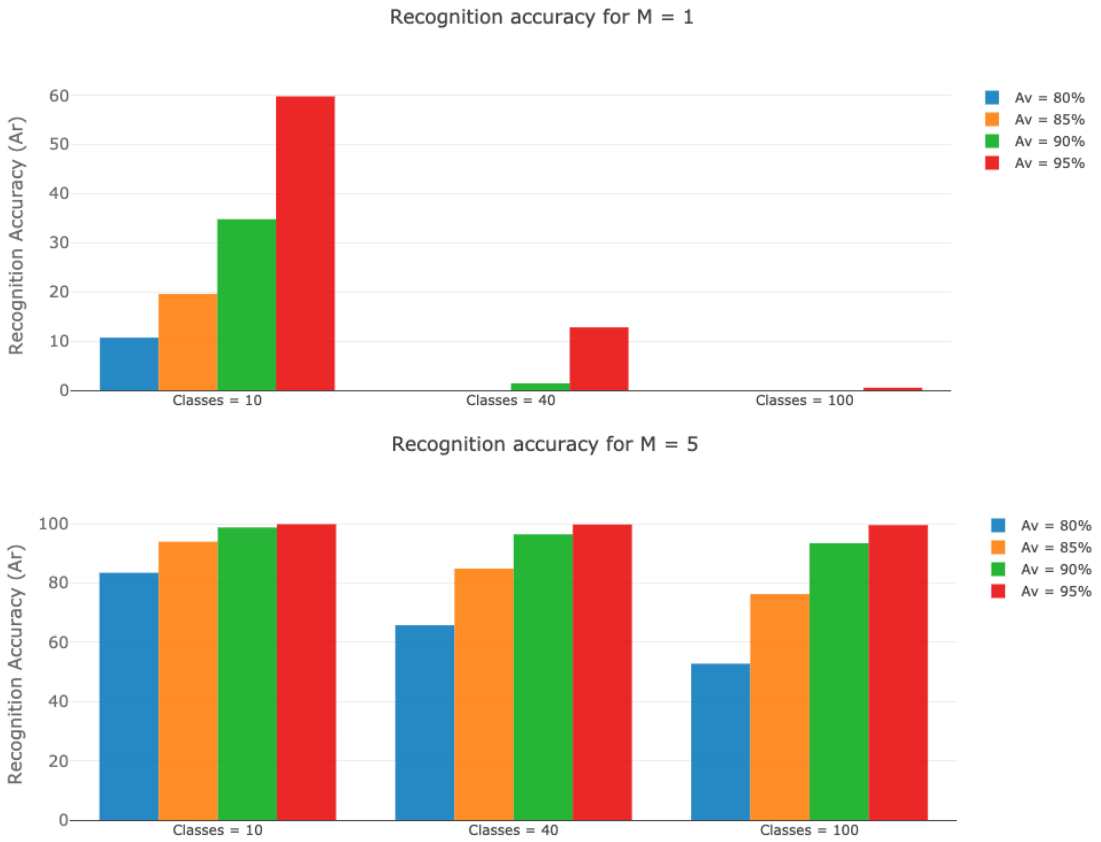


Figure 6.1: Comparison of α_r and α_v for different number of classes

6.3 Design of classifiers

The embeddings generated using facial recognition models are fed into classifiers as training data. The embeddings are such as to allow the embeddings belonging to the same person cluster in a finite dimensional space. This information is crucial when it comes to the classification of a test sample. Traditional classifiers build their own decision boundaries by learning from the data. But as mentioned earlier, the training data might not be large enough to learn generalized decision boundaries. Distance from training samples can directly be used to label a test sample. As a result, K-Nearest-Neighbors becomes the algorithm of choice. We propose a new classifier which provides slightly improved performance over all classifiers when the number of classes is large. The design roots in KNN algorithm with a change in the way the k nearest neighbors are considered. For every class, a similarity score is obtained by considering only the k nearest neighbors instead of considering k most nearest neighbors from the entire training set like KNN. The class similarity score is the mean of the similarity score between k most nearest samples and the test sample. The rationale behind the classifier is that the embeddings for alike poses lie closer in space. For a person posing left, it might be sufficient to consider most similar poses from the training set belonging to different classes. Considering all training samples for every person might affect the decision negatively. The classifier is thus resilient to poses and provides improved performance gains. The following pseudo code summarizes the algorithm.

Algorithm 4: KMeans Classifier

Input face f, integer k
Output Class label
def *kmean_clf(f, k)*:
    for i *in* $0 : C-1$ **do**
        $nearest_i$ = **sort**$\{\text{dist}(f, f_j) \forall f_j \in \omega_i\ )\}$
        $score_i$ = **mean**$(nearest_i[:k])$
    endfor
    return **argmax** $(score_i)$

In this work, we also experiment with several traditional classifiers including multi-class SVM, decision trees and random forests. Accuracy scores using several classifiers are presented in Chapter-8.

CHAPTER 7

Implementation

As discussed briefly in Chapter-3, three classrooms in IIT Tirupati have functional installations of the automated attendance system. In this section, we discuss the implementation details and usage instructions of the system. The hardware part of the system consists of the cameras and computational machine. For the camera we use the Amcrest PTZ 2048p camera with pan-tilt-zoom capabilities. It comes with an API handle to the camera which can be accessed via a network. The computational machine is a central computer or server which runs a host application, controls the camera, and performs all computation centrally. For real time performance, access to a GPU is recommended. We use the GTX 1080Ti for running the neural network for facial detection and embeddings generation.

We built automated scripts for handling the camera, and an E2E browser based application which is hosted on the main server. The software stack for the web application uses Python, Flask, Amcrest API, and PyTorch. The web application was built with software engineering principles in mind to ensure modularity, extensibility, and ease of maintainability. We build multiple libraries and a suite of scripts which allows easy development. There is a well defined API which operates between the front-end and back-end. We next describe the usage of the web application in detail and tours the full functionality of the application.

7.1 Web Application

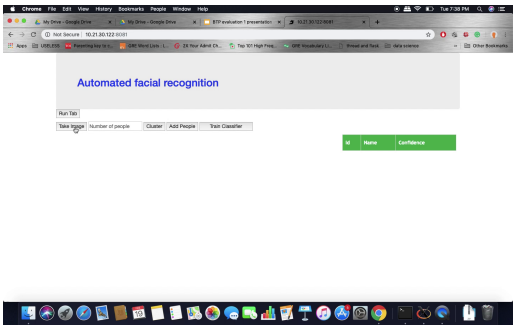
The web application operates in two modes, *Training* and *Operation*. Figure-7.1 shows a sample view of the application. The application interacts with a defined API which we here on refer to as the webAPI, to interact with the python-flask based back end. The back end runs on the server machine with access to GPU, and performs all the computationally intensive tasks. The application has a connection to the camera through the Amcrest API. The application comes with a settings page for configuration settings such as camera IP address.

7.2 Training mode

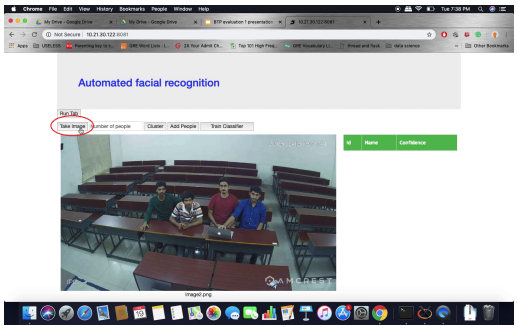
To streamline the training operation, the protocol we established is built into the web application, which handles the complete process from collecting training data to training the classifier. The functionality also includes the clustering based labelling discussed in Chapter-4. The trained model can be saved and restored. Figure-7.1 below shows the process of training a classifier from the web application.

7.3 Operation mode - Running attendance system

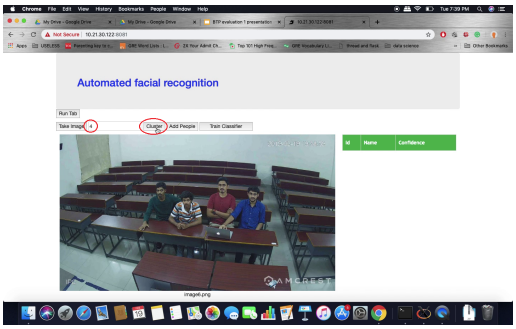
The classifier model used in operation mode can be either pre-loaded from configuration or trained as described in the earlier section. Figure-7.2 below outlines the process of capturing attendance during lecture. We also have API based invocation of the process, thus allowing the setup of Chron jobs for schedule based automated running.



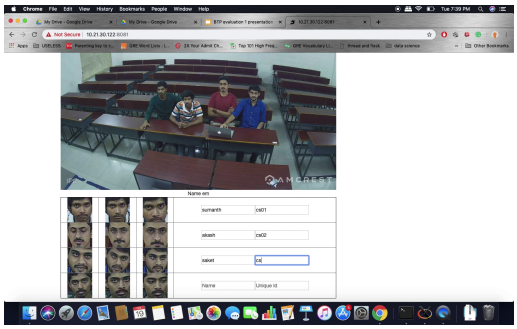
(a) The training mode of the application can be found as shown in figure.



(b) Take Images from using the web app connected to the camera.



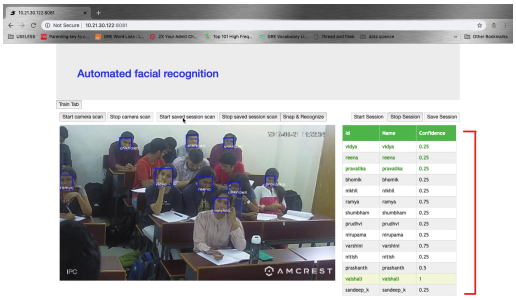
(c) Once multiple training images are taken, the user enters the number of people and clicks on *cluster*. The web application calls the server which detects faces, generates embeddings and clusters all the faces from the images.



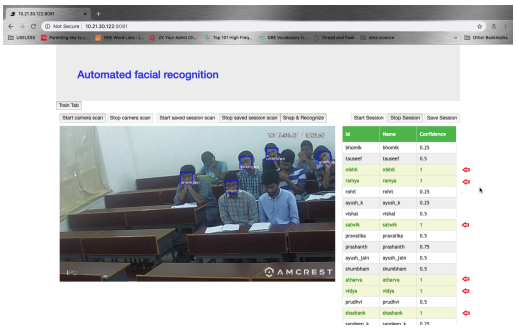
(d) The clustered faces are presented for labelling. Steps (b) and (c) can be repeated to register all students. Finally, on clicking *Train* button, a classifier is trained and the model is saved for default use.

Figure 7.1: Training Mode

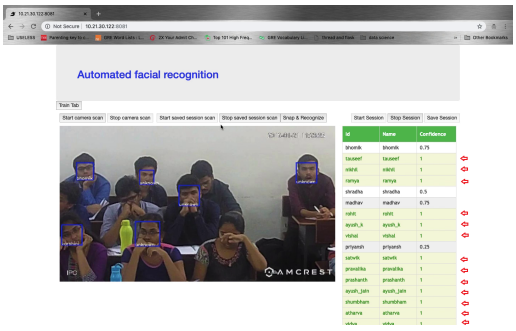
(a) Operation is the default mode on starting the web application. The classifier model used is loaded as specified in configuration.



(b) On clicking Camera scan, the camera starts a scanning routine for the attendance hall. Images taken are processed by the server, and the images can be seen in real time through the application.



(c) The final confidence scores for students are updated in the results on the right. On reaching sufficient confidence, the rows are highlighted in green.



(d) On completion of the process, the results are shown on the right and can be downloaded or used as required. The process time depends on the length of scan. For our classrooms, we see a typical time of 20 minutes.

Figure 7.2: Operation Mode

CHAPTER 8

Performance analysis and results

In this chapter, we present an extensive analysis of the major components governing the system pipeline. We begin with a discussion of face detection and clustering followed by robustness of embeddings generated by Facenet and SENet. We then analyze the face verification performance using the two models on two batches. Further, we present face recognition results on our own classroom dataset. The end goal is to assess the performance of the system in granting accurate attendance with the individual components glued together. In the last section, we present the system performance over several sessions.

8.1 Face detection

The first step of the face recognition process is face detection. Evaluation of face detection is performed on two aspects, the detection rate, and the accuracy of the bounding boxes generated. We experimented with two popular face detectors, dlib CNN face detector, and an implementation of multi-task cascaded convolutional neural network (MTCNN). Both networks produce output with confidence scores, and a threshold is used to negotiate the FPR(False positive rate) and TPR(True positive rate). On experimentation, we found that both detectors performed well in detecting faces with occlusion involved. These images, however lead to reduced performance in face recognition classifier. In a classroom environment, we find several faces are occluded at different time frames. However, we have access to several images, and hence recognition precision plays a more significant role in the accuracy of our final task. Using this rationale, we set a higher threshold to filter out faces with high occlusion or low confidences. Of the two networks, our final application employs the MTCNN network for both detection and alignment of faces. With a 0.0% false detection rate obtained by setting a higher threshold, we do not explicitly evaluate face detection model for several reasons as follows:

1. The state of the art face detectors [20, 6, 19] have been verified to achieve benchmarking results on most datasets.
2. Detecting smaller faces is not the main goal since smaller the face, more difficult it is to recognize.
3. The images are captured using zoomed in camera for better recognition.

8.2 Clustering

Clutering algorithm	Purity	Rand Index (RI)
K-means	0.9489	0.9691
Agglomerative clustering (linkage=complete, affinity=cosine)	0.9635	0.9970
Binning algorithm (s=1)	**0.9870**	**0.99742**

Table 8.1: Comparison of clustering performance

For evaluation of clustering performance, $purity$, $NMI(Normalizedmutualinformation)$ and $RandIndex(RI)$ metrics are commonly used. We employed K-means clustering, agglomerative clustering and the binning algorithm seeded with one random image from the testing set to cluster faces from session 1(862 faces). Table-8.1 compares $purity$ and $RandIndex$ metrics. The performance of binning algorithm can be further improved if more seed images are provided.

The $purity$ metric is defined as,

$$\begin{aligned} purity(\Omega,\mathbb{C}) &= \frac{1}{N}\sum_{k} max_j|\omega_k \cap c_j| \\ &\text{where } \Omega = \{\omega_1,\omega_2,..\omega_K\} \text{ is the set of clusters,} \\ &\mathbb{C} = \{c_1,c_2..c_J\} \text{ is set of classes} \end{aligned} \tag{8.1}$$

The $RandIndex$ is defined as,

$$RI = \frac{TP+TN}{TP+TN+FP+FN}$$

where TP (True Positive) is pairs of samples with same label in same cluster

TN (True Negative) is pairs of samples with different label in different clusters

FP (False Positive) is pairs of samples with different labels in the same cluster

FN (False Negative) is pairs of samples with same label in same cluster

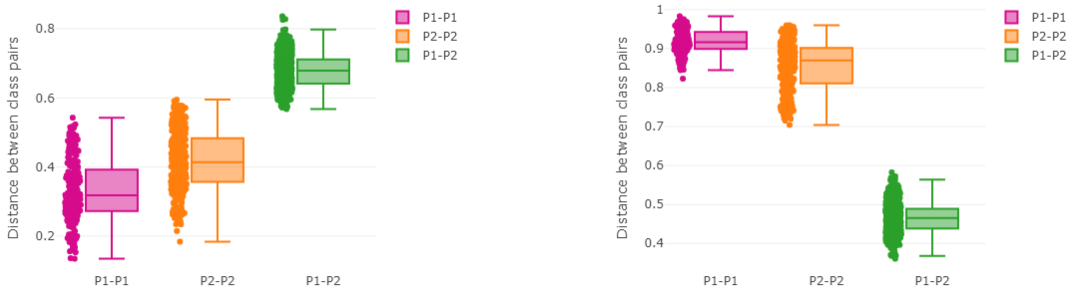
(8.2)

8.3 Robustness of embeddings

The embeddings generated by Facenet are such to allow clustering of faces in the 128-d euclidean space. We visualize the distance between pairs of same class and different class through a box-plot shown in Fig-8.1. Euclidean distance is used as the distance measure where smaller the distance, more alike are the pairs. P1 and P2 denote the randomly picked first person and second person for analysis. The box-plot titled P1-P1 depicts distances between all pairs of faces belonging to the first person. The mean distance between any two pairs belonging to P1 is close to 0.3. Similarly, for any two faces belonging to P2, the mean distance is close to 0.4. The mean distance between any two pairs (p1, p2) where p1 $\in$ P1 and p2 $\in$ P2 is close to 0.7 as depicted by the green colored box-plot in Fig-8.1a. Thus, the embeddings are robust enough to allow a thresholding scheme to separate any two classes.

We obtain similar plots for 2048-d embeddings generated by SENet. The similarity or distance measure used is cosine similarity where higher the value, more alike are the pairs. Using the same P1 and P2 in Facenet, Fig-8.1b depicts the box-plots for SENet. The average cosine similarity between any two pairs for P1 is greater than 0.9 while it is greater than 0.85 for P2. For any pair (p1, p2) where p1 $\in$ P1 and p2 $\in$ P2, the average cosine similarity is close to 0.5. The maximum similarity between any such pair is still less than 0.6 demonstrating the high robustness of SENet embeddings. The interclass separation achieved by SENet is far more than that obtained by Facenet indicating superior performance of SENet embeddings.

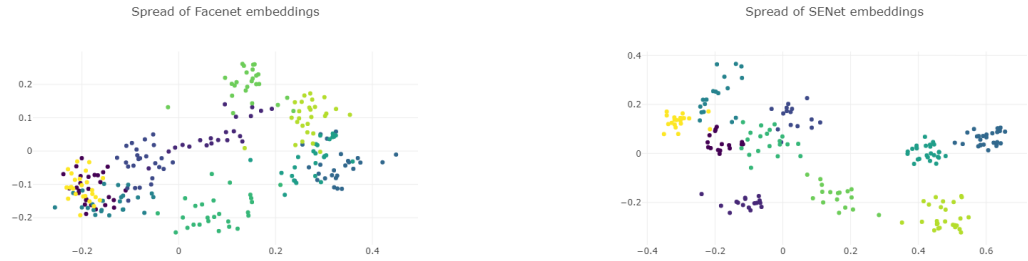
We perform principal component analysis on the embeddings generated by Facenet and SENet in order to visualize the face clusters in two dimensions. Fig-8.2a and



(a) Facenet embeddings (b) SENet embeddings

Figure 8.1: Intraclass and Interclass separation

Fig-8.2b depict the face clusters for Facenet and SENet respectively with all samples from 10 different classes. The class colors are consistent across the plots. Facenet clusters intersect while SENet clusters form distinct groups again indicating superior performance of the generated embeddings.



(a) Facenet clusters (b) SENet clusters

Figure 8.2: 2-D visualization of face clusters for 10 classes using PCA.

8.4 Face verification performance

The problem of face verification was described in Section-5. Verification deals with answering the question: "Given two facial images, do they belong to same person?" Accuracy for face verification is reported as the number of pairs answered correctly divided by total number of pairs. We evaluate the verification accuracies on the *train-set* using Facenet and SENet for both batch-1 and batch-2 separately. The accuracies are presented in Table-8.2. The ROC curve for batch-1 and batch-2 is shown in Fig-8.3a and Fig-8.3b respectively. It is evident from the plots that SENet performs much better than

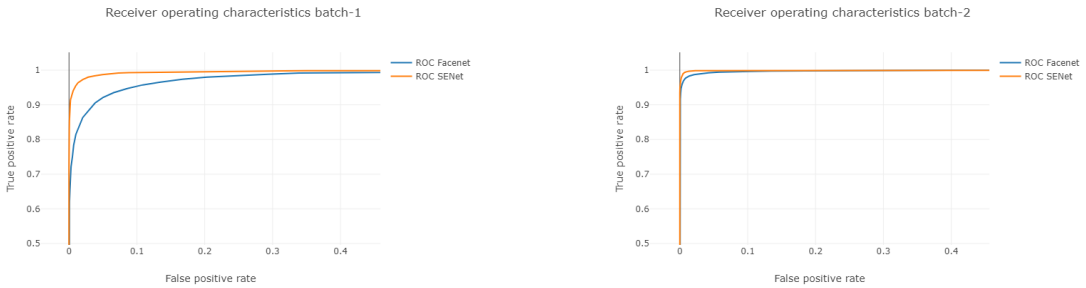
Facenet for batch-1 and slightly better for batch-2. Between batch-1 and batch-2, the models perform better on batch-2 not because of any specific reason but a possible few which include:

- Training images are procured with a higher resolution camera.
- Training images are procured under better illumination conditions.
- Number of students in batch-2 is less than that in batch-1.

From the box-plots of embeddings, 2D cluster visualization and verification accuracies presented above, empirically, SENet performs better than Facenet. We choose SENet for our classroom implementation and present face recognition and aggregate attendance results only for SENet.

Model	Batch-1	Batch-2
Facenet	0.937	0.985
SENet	**0.979**	**0.993**

Table 8.2: Verification accuracies of Facenet and SENet on *train-set*. The *train-set* was gathered under controlled conditions.



(a) Batch-1 (b) Batch-1

Figure 8.3: Face verification ROC.

8.5 Face Recognition performance

8.5.1 Classifier

In this section, we report the performances of popular classifiers for FR on our dataset. As described in Chapter-5, the dataset consists of *train-set* and *operation-set* for each of the two batches separately. Table- 8.3 compares the precision accuracy of some of the classifiers. The models were built using *train-set* and tested on *operation-set* of batch-2.

The accuracies are reported post hyper-parameter tuning. Also *new faces* (faces not a part of dataset) in every session are excluded from evaluation.

Classifier	Accuracy
One vs. rest SVM (gamma:scale)	89.73%
K-means classifier	88.35%
K-nearest neighbours (for best K=8)	88.04%
Random Forest	86.33%
Ada Boost with Decision tree base classifier	37.60 %

Table 8.3: Comparison of recognition accuracy of different classifiers for five sessions together. Models were built using *train-set* and tested on *operation-set* of batch-2. The *train-set* and *operation-set* were gathered under controlled and uncontrolled conditions respectively.

We observe that SVM and K-nearest neighbours, and designed K-means classifier give comparable performance. In Table-8.4, we perform a 50% train-test split of the *operation-set* of batch-2 to train and test different classifiers.

Classifier	Accuracy
One vs. rest SVM (gamma:scale)	93.23%
K-means classifier (for K=5)	98.78%
K-nearest neighbour (for best K=8)	98.70%

Table 8.4: Comparison of recognition accuracy for 50% train-test split of the *operation-set* of batch-2. The *operation-set* was gathered under uncontrolled conditions.

In traditional machine learning schemes, the train data and test data are similar where models are often tested on train-test split of a larger dataset. But for systems using facial recognition, the acquired train data does not correlate well to the real data that the model will operate on. The *operation-set* is a representative of the real world data while the *train-set* represents data collected under controlled conditions. The initial faces can only be acquired by explicit registration of students similar to the acquisition procedure used for gathering *train-set*. But the classroom conditions where the system is run will be uncontrolled, similar to *operation-set*. The gap becomes crucial when it comes to system performance. As we see from Table-8.3 to Table-8.4, the performance of classifier improves by changing the training dataset from the *train-set* to using a 50-50 split of *operation-set* as training data. The environmental correspondence in the training and test set improves performance, since both are sampled from the same data. Thus, to better the system performance, it becomes essential to augment data to the training

set as it becomes available over several sessions. We study the system performance under an online learning scheme in the next section. With reference to Table-8.4, we see that both KMeans classifier and K-nearest neighbor give similar performance and both outperform SVM classifier. However, the design of KMeans classifier has a few advantages (as discussed in section-6.3); it provides a better confidence score allowing an efficient trade-off between precision and recall.

8.5.2 Online learning

As we noted from Table-8.3 and Table-8.4, there is a significant performance improvement if images captured during lecture sessions are employed in the training set. We follow an online learning based approach to incrementally augment the training set using images captured during session. The initial classifier is built using the *train-set* of batch-2, and after every session, the training set is augmented by up of 10 faces per person which were classified with a confidence score of $> 95\%$. The evaluation set consists of five lecture sessions $\{S1, S2, S3, S4, S5\}$. The *train-set* is referred to as $S0$, and the set of new images added after each session $S\{i\}$ are referred by $A_{S\{i\}}$. Table 8.5 reports the accuracy results evaluated over five sessions using the KNearest Neighbors classifier.

Training Set (T)	Evaluation Set				
	S1	S2	S3	S4	S5
$T_0 = S0$	90.40%	85.97%	91.67%	86.21%	87.08%
$T_1 = T_0 \cup A_{S1}$		90.44%	97.74%	87.92%	86.53%
$T_2 = T_1 \cup A_{S2}$			98.09%	91.73%	94.59%
$T_3 = T_2 \cup A_{S3}$				92.68%	94.13%
$T_4 = T_3 \cup A_{S4}$					94.96%

Table 8.5: Comparison of recognition accuracy with an online learning scheme. The top 10 faces per student after each session are augmented to the training data. T_i represents training data after i^{th} session. S1 to S5 represent five classroom sessions of batch-2 used for evaluation.

A significant improvement in classifier performance can be observed as the training data is augmented. The improvement attenuates, and then appears to converge, thus training data from only a few sessions needs to be added. Evaluating different online learning methods over a longer duration is an interesting future direction for this work.

8.6 Evaluation of attendance

In this section, we present the results of aggregate attendance obtained by gluing together several components of the system. We begin the description of the procedure followed for evaluating attendance.

A session comprises of a set of images taken during the lecture hour. The images span the entire classroom in a grid fashion with multiple faces contained in one image. The camera protocol was described in detail in the Chapter-3. In our experiments, 100-250 images are captured in every session with an average of six faces in every image. Thus, we obtain around 600-1500 total number of faces for every session. Let clf denote the classifier which outputs a label l and a confidence score s for every input face f. Thus, $clf(f) = l, s$. After labelling every face using the classifier, we consider top k faces sorted by confidence scores for every class $c \in C$. The top k faces for every class $c \in C$ are written to a directory and manually verified for false identification. The rationale behind using top k faces is that to mark the attendance of a student, it is enough to verify whether the best k faces are correctly labelled as that student considering the recognition accuracies of the model.

A histogram depicting the result of evaluation scheme outlined above for one of the sessions is shown in Fig-8.4. The histogram was obtained by using MTCNN for face detection, SENet for face descriptors, KMean Classifier as the classifier and cosine similarity as the similarity measure. The blue region for each bar corresponds to true positives for the respective bin while yellow region corresponds to false positives. The scores on the top of each bar indicate *mean_score* for the respective bin considering top $k = 10$ faces. The true *present/absent* labels for each student are indicated at the bottom of the bar using the alphabets *p/a*.

Inference The pipeline is quite robust with low scores (< 0.72) for absent students and high scores (≥ 0.72) for present students. The mean scores for the absent students: *pranav, atharva* and *saran_teja* are 0.0, 0.67 and 0.71 respectively. Note that the total number of faces labelled as absent students from among $\sim$800 faces detected for the session is less than 10 for most absent students. This demonstrates the extensive training of SENet and robustness of KMeanClassifer scheme to label images. The system is not fully perfect though with low *mean_score* of 0.62 for the present student *madhav* with

all five false detections.

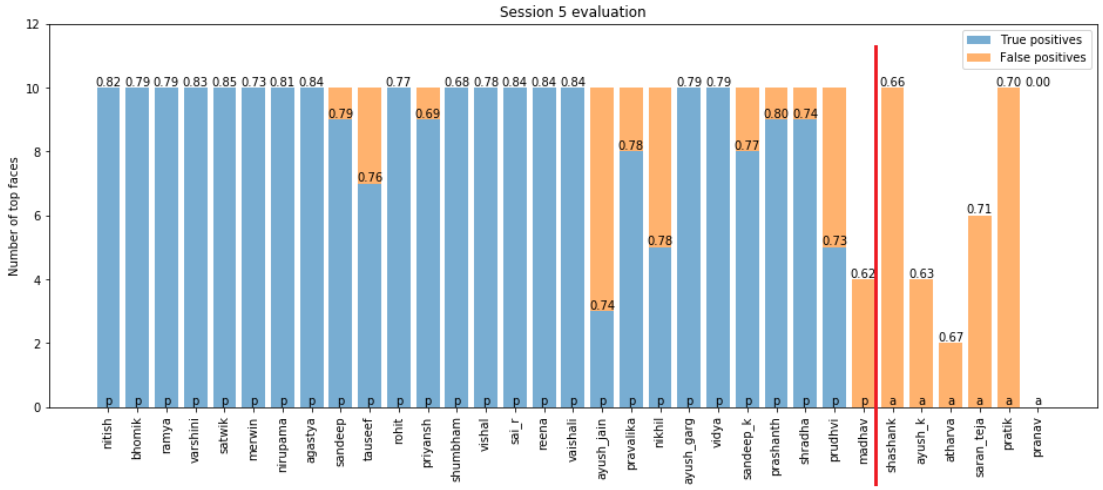


Figure 8.4: Histogram depicting system performance on batch-1. The bars to the right of the red line correspond to absent students. The true present/absent labels are marked at the bottom of each bar. Mean cosine scores for absent students are lower as compared to present students as indicated by the numbers on the top of each bar.

Note that all evaluations are done using MTCNN for face detection and SENet model with KMean Classifier for labelling. We now use the scheme defined above to obtain the *mean_score* considering only top $k = 10$ faces and grant attendance to students over several sessions. A threshold on the *mean_score* then determines whether the student is marked as present or absent. The best threshold is determined empirically. Table-8.6 summarizes the aggregate attendance for batch-2 over 5 sessions using the threshold of 0.69.

true-pred	Session1	Session2	Session3	Session4	Session5
P-P	26	28	26	24	25
A-A	1	0	1	2	1
P-A	0	0	0	2	2
A-P	1	0	1	0	0

Table 8.6: Final aggregate attendance batch-2 using best threshold = 0.69. The true label and predicted label for each row is indicated in the first column.

The system performs very well on batch-2 with a maximum of 2 out of 28 students left incorrectly marked over any session. Most present students are correctly marked as present by the system. Also, *new faces* (faces not a part of dataset) in the lecture do not affect the system performance on the faces of the primary students. The peculiar or uncertain cases can be considered by the teacher at the end of the class. These few peculiar cases can be attributed to several reasons which include: (1) Number of images

captured (2) Camera protocol (3) Students remaining undetected due to dropped heads. The details on number of images captured were presented in Table-5.2 in Chapter-5. Notice the constant number of 230 images captured across all sessions of batch-2.

A larger scan duration and further optimization of camera scan protocol can bring the system to near perfection. We also chart the precision, recall, F1 and accuracy scores corresponding to batch-2 in Fig-8.5. All four measures stay over 92% across all sessions while precision is 100% over three sessions. The numbers in Table-8.6 and Fig-8.5 demonstrate the robustness of the pipeline indicating that a practical implementation is feasible using the methodologies discussed in this work.

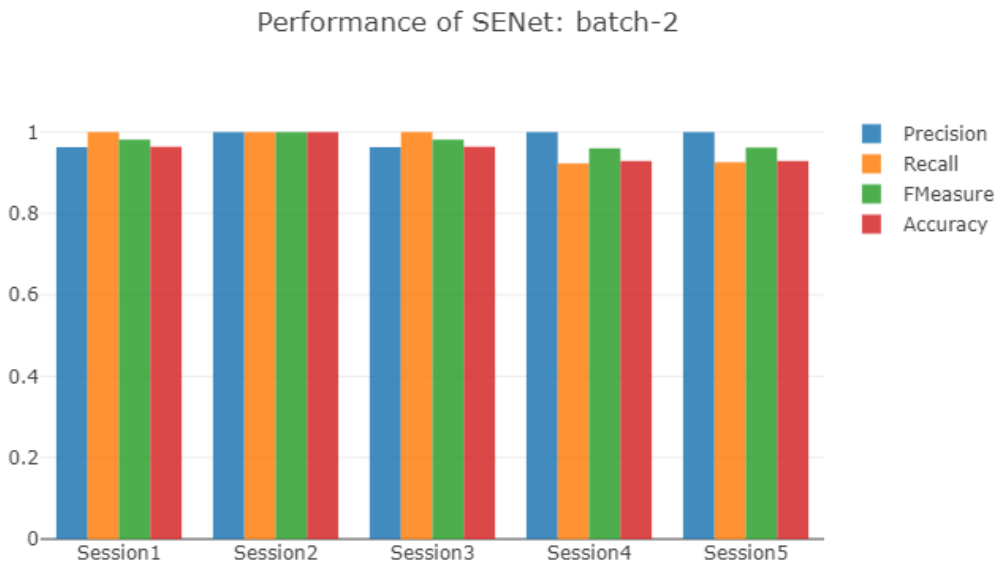


Figure 8.5: Precision, recall, F1 and accuracy scores for batch-2 using the best threshold = 0.69. The precision is 100% for three different sessions while all remaining scores stay over 92% across all five sessions.

CHAPTER 9

Limitations, Future work and Conclusion

In this chapter, we first discuss the current limitations and future research opportunities corresponding to the work we presented in this thesis, and then outline the conclusion of our project.

9.1 Limitations

A few limitations of the work presented in this thesis is discussed below.

Visibility of person in camera's field of view: There are two associated issues stemming from this problem, scaling the system to a large lecture hall and ensuring all attendees are visible for a significant time period in the camera's field of view. We believe the former can be solved by using multiple cameras for large lecture halls. In our experiments, we employ cameras with tilt, pan and zoom capabilities. The latter issue can originate if the camera is improperly positioned, or if a person is never detected because the subject is always occluded, sleeping or never looks at the camera. This issue is sensitive to the environment and people, and depending on experience, the attendees may be asked to ensure they look at the camera at-least at some point during the lecture. In our experiments, we found this issue to be limited to only 2-3 out of sixty-one students, who may be notified.

9.2 Future work

1. **Identifying sources of errors:** The recognition system performs much better at identifying some people compared to others, due to the heuristic nature of the process involved in generation of face embeddings. Hence, the errors predominantly originate from a set of few people. This insight is key to reducing the errors in a system. There is scope for introducing a mechanism which diagnoses the classifier to identify people who are more likely to be unrecognized or falsely recognized.

This information can then be used to improve the system by training the classifier on more data of the person, transforming the embedding space, or transfer learning on the embedding generating neural network.

2. **Expanding dataset:** The presented dataset represents a valuable contribution of our work in view of the features described in Chapter-5. The dataset is uniquely positioned especially for evaluation of online learning systems. There is scope for expansion over time to a significant size face dataset as the system is employed across lecture halls at IIT Tirupati.
3. **Enhancing face embedding generated:** We found significant deterioration in FaceNet and SENet FR accuracy in our application compared to FR datasets. We believe biases learned by the neural network are one attribute. There is scope for improving quality of the embeddings generated by transfer learning using Indian faces dataset.
4. **Emotion recognition:** The system may be augmented to extract richer information from lecture sessions, such as by implementing built-in emotion recognition and pupil-tracking.

9.3 Conclusions

In this work, we explored several aspects that govern application of facial recognition systems in practical scenarios. We focused primarily on classroom attendance and presented the design of a complete automated facial recognition system. We proposed the binning algorithm for semi-supervised generation of training data and then constructed a novel dataset for bench-marking practical systems. We explored the relation between verification and recognition accuracies followed by the design of new KMeans classifier that provides improved performance. Finally, we presented an extensive analysis of several components of the system starting from face detection to granting final attendance. Through a web-app, the system was thoroughly tested in three classrooms at IIT Tirupati. The work stands as a good move towards non-intrusive identification, smarter classrooms and practical application of face recognition technology.

APPENDIX A

Students consent form

भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI

Consent Letter April 2th , 2019

To Students,

Under the guidance of Dr. Rama Krishna Gorthi, we are experimenting and researching an automated facial recognition based system. One of the possible applications is a lecture attendance mechanism. For the same, we are collecting image frames during lectures for training the Deep Learning system. The data collected will be used strictly for research purposes. Please provide your details below to approve consent as a participant in the study, and give permission to collect image frames which may contain your faces.

For any queries, or if you are interested in the project, you may contact either of us. Thank you for your cooperation.

Thank you,
Saket Joshi - tce15b020@iittp.ac.in,
Akash Dhasade - tee15b007@iittp.ac.in

Sr No.	Name	Roll No	Signature
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

REFERENCES

[1] **Cao, Q.**, **L. Shen**, **W. Xie**, **O. M. Parkhi**, and **A. Zisserman**, Vggface2: A dataset for recognising faces across pose and age. *In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018.

[2] **Dalal, N.** and **B. Triggs**, Histograms of oriented gradients for human detection. *In international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1. IEEE Computer Society, 2005.

[3] **Déniz, O.**, **G. Bueno**, **J. Salido**, and **F. De la Torre** (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, **32**(12), 1598–1603.

[4] **Fu, R.**, **D. Wang**, **D. Li**, and **Z. Luo**, University classroom attendance based on deep learning. *In 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*. IEEE, 2017.

[5] **Huang, G. B.**, **M. Ramesh**, **T. Berg**, and **E. Learned-Miller** (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

[6] **Jiang, H.** and **E. Learned-Miller**, Face detection with the faster r-cnn. *In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017.

[7] **Kawaguchi, Y.**, **T. Shoji**, **L. Weijane**, **K. Kakusho**, and **M. Minoh**, Face recognition-based lecture attendance system. *In The 3rd AEARU workshop on network education*. Citeseer, 2005.

[8] **King, D. E.** (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, **10**, 1755–1758.

[9] **Klare, B. F.**, **B. Klein**, **E. Taborsky**, **A. Blanton**, **J. Cheney**, **K. Allen**, **P. Grother**, **A. Mah**, **M. Burge**, and **A. K. Jain**, Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. ISSN 1063-6919.

[10] **Lukas, S.**, **A. R. Mitra**, **R. I. Desanti**, and **D. Krisnadi**, Student attendance system in classroom using face recognition technique. *In 2016 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2016.

[11] **Parkhi, O. M.**, **A. Vedaldi**, and **A. Zisserman**, Deep face recognition. *In* **M. W. J. Xianghua Xie** and **G. K. L. Tam** (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2015. ISBN 1-901725-53-7. URL https://dx.doi.org/10.5244/C.29.41.

[12] **Schroff, F.**, **D. Kalenichenko**, and **J. Philbin**, Facenet: A unified embedding for face recognition and clustering. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[13] **Sokal, R. R.** and **C. D. Michener** (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, **38**, 1409–1438.

[14] **Taigman, Y.**, **M. Yang**, **M. Ranzato**, and **L. Wolf**, Deepface: Closing the gap to human-level performance in face verification. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[15] **Viola, P.** and **M. J. Jones** (2004). Robust real-time face detection. *International journal of computer vision*, **57**(2), 137–154.

[16] **Wagner, A.**, **J. Wright**, **A. Ganesh**, **Z. Zhou**, **H. Mobahi**, and **Y. Ma** (2012). Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(2), 372–386.

[17] **Wang, M.** and **W. Deng** (2018). Deep face recognition: A survey. *arXiv preprint arXiv:1804.06655*.

[18] **Wolf, L.**, **T. Hassner**, and **I. Maoz** (2011). Face recognition in unconstrained videos with matched background similarity. *CVPR 2011*, 529–534.

[19] **Yang, S.**, **P. Luo**, **C.-C. Loy**, and **X. Tang**, Wider face: A face detection benchmark. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[20] **Zhang, K.**, **Z. Zhang**, **Z. Li**, and **Y. Qiao** (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, **23**(10), 1499–1503.