# Applications of Visual Identification & Verification

Prepared By:

**Lekha Agrawal**

**14MCEC15**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**INSTITUTE OF TECHNOLOGY**
**NIRMA UNIVERSITY**
**AHMEDABAD**

May 2016

# Applications of Visual Identification & Verification

**Major Project**

Submitted in partial fulfillment of the requirements
for the degree of

**Master of Technology in Computer Science & Engineering**

Submitted By:
**Lekha Agrawal**
**14MCEC15**

Guided By:
**Prof. Vipul Chudasama**
**Mr. Harinarayanan K K**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**INSTITUTE OF TECHNOLOGY**
**NIRMA UNIVERSITY**
**AHMEDABAD**

May 2016

# Declaration

I, **Lekha Agrawal, 14MCEC15**, a student of semester IV, Master of Technology in Computer Science & Engineering, Nirma University, Ahmedabad, hereby declare that the project work **Applications of Visual Identification & Verification** has been carried out by me under the guidance of **Prof. Vipul Chudasama**, Department of Computer Science and Engineering, Nirma University, Ahmedabad. This Project has been submitted in the partial fulllment of the requirements for the award of degree Master of Technology (M.Tech.) in Computer Science & Engineering Information and Network Security, Nirma University, Ahmedabad during the year 2014 - 2016.

I have not submitted this work in full or part to any other University or Institution for the award of any other degree.

| | |
|---|---|
| **Date:** | **Lekha Agrawal** |
| **16th May 2016** | **14MCEC15** |

# Certificate

This is to certify that the major project entitled ”Applications of Visual Identification & Verification” submitted by Lekha Agrawal(14MCEC15), towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science & Engineering of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, havent been submitted to any other university or institution for award of any degree or diploma.

**PROF. VIPUL CHUDASAMA**
Guide
Institute of Technology
Nirma University

**MR. HARINARAYANAN K K**
External Guide
Aindra Systems

**DR. SANJAY GARG**
Head of Dept., CSE
Institute of Technology
Nirma University

**DR. PRIYANKA SHARMA**
Coordinator, M.Tech (CSE)
Institute of Technology
Nirma University

**DR. P. N. TEKWANI**
Director
Institute of Technology
Nirma University

**Aindra Systems Private Limited**

Registered Office:
26, Gunasheela Layout
19th Cross, 24th Main
JP Nagar 5th Phase,
Bangalore – 560078
aindra.in

Phone: +91-80-42051729

May 12, 2016

**To Whomsoever This May Concern**

This is to certify that, **LEKHA AGRAWAL, 14MCEC15**, pursuing her final year, Master of Technology, Computer Science & Engineering (CSE) at Nirma University, Ahmedabad, has successfully completed the project work titled **"APPLICATION OF VISUAL IDENTIFICATION & VERIFCATION"** as a part of her M.Tech Course. This project was executed during the period of 25th May 2015 to 25th May 2016.

This is a record of bonafide work carried out by **LEKHA AGRAWAL** under my supervision and guidance at **Aindra Systems Private Limited, Bangalore**. During the course of the project her code of conduct was good.

For, **Aindra Systems Private Limited.**

AIndra Systems Private Limited

[illegible]

**Abhishek Mishra**
**Co-Founder & Head of Business Dept.**
**Aindra Systems**

# Acknowledgement

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Vipul Chudasama**, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

A special thank you is expressed wholeheartedly to **Prof. Vijay Ukani** & **Prof. Priyank Thakkar**, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for their constant support & valuable guidance thoughout the post graduation program.

I would also like to thank **Dr. Sanjay Garg**, Honb́le Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for providing basic infrastructure and healthy research environment.

I would also like to thank **Dr. P. N. Tekwani**, Honb́le Director, Institute of Technology, Nirma University, Ahmedabad for the motivation he has extended throughout course of this work.

I would also like to thank **Mr. Harinarayanan K K** & **Mr. Nirmal Jith**, Aindra Systems for the guidance and support provided in my internship & the project work.

I would like to thank my all faculty members for providing encouragement, exchanging knowledge during my post-graduate program.

I also owe my family & classmates in the special thanks for helping & supporting me on this path and for making project more enjoyable.

**Lekha Agrawal**
**14MCEC15**

# Abstract

The objective of the project is to develop an application that can be used in organizations, hospitals and educational institutions to keep record of daily attendances/security. The application is based on Face Recognition & Verification techniques using computer vision & machine learning. The algorithm learns using the faces captures daily in the system. Another advantage is that face recognition works well for both: images of single person or a group. There are three major components in this project: 1.Client Application (Desktop and Mobile to capture/upload images), 2.Server (Cloud where actual face detection/recognition algorithms reside) and 3.Web interface (to display result statuses). There is no need for any external hardware installation. Only a device with a camera (laptop with webcam, pc with webcam, smart phone) is required for capturing/uploading images to server. The application's newer version is going to be launched with the new architecture containing the improved algorithm, APIs, and a new web interafce using the APIs. The new algorithm is made using improved layers in the neural network, classifiers and verifiers. The APIs are created to removed the tight coupling between database and server components which was existing in the older architecture, as well as third party APIs will help to create custom applications for enterprises.

# Abbreviations

| | |
|---|---|
| **ROC** | Receiver Operating Characteristic |
| **AUC** | Area Under Curve |
| **LFW** | Labelled Faces in Wild |
| **DNN** | Deep Neural Network |
| **HOG** | Histogram of Oriented Gradients |
| **FR** | Face Recognition |
| **ORM** | Object Relational Model |
| **SVM** | Support Vector Machine |
| **REST** | Representational State Transfer |
| **SOAP** | Simple Object Access Protocols |
| **XML** | Xtensible Markup Language |
| **API** | Application Programming Interface |
| **FRT** | Face Recognition Techniques |

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Listings

# Chapter 1

# Introduction

**Smart Attendance**, & **Smart Verification** are applications based on Artificial Intelligence, which can be used in handheld devices with equipped with cameras like computers, smartphones, tablets & provide the functionality to detect and identify/verify people. These devices are used to campture images and which are sent to the cloud based servers & then processed by the intelligent algorithms to detect and recognize/verify human face. As part of the results, useful data visualizations are created for particular stakeholders in an organization.

The system is also able to collect demographic data & can generate the contextual data which can be available for market consumption. These intelligent algorithms are capable to use video streamings as well, which scales the degree of flexibility to leverage the existing infrastructure.

## 1.1 Statement of Purpose

The objective of the project is to develop an application that can be used in organizations, health care industry and educational institutions to keep record of attendances of individuals in organization and is based on the concepts of computer vision and machine learning. It aims to create smart attendance campus as well as smart attendance industries.

The applicable ares are:

- **Enterprise:** It can serve as a solution to track & monitor the employeesṕresence on-field or off-field with storage of timestamps & location. This is capable of capturing single person attendance or group attendance.
- **Education:** It can serve as a solution for biometric, proxy free attendance management system to update the institute's ERP system. This is capable of generate and store digitaized data on servers.
- **Health Care:** In terms to verify referral doctors or remote doctors, to ensure only intended person turns up at referral sites, as well as to serve as security purposes, to ensure that only authorized person has access to resources.

- **Security:** In terms to verify the authenticate the validity of only the authorized person in the organization where security is needed, to ensure the correct person gets the access, verification can be used.

## 1.2 Scope

In organizations it would be useful for managers to supervise their employees attendance and which is important in generating their monthly salary, monitor the activities etc. In educational institutions it can be used to mark attendance for various sessions in a day and which is crucial in determining the students eligibility to appear for exams. Another advantage is that face recognition works well for both: images of single person or a group.

There is a wide scope in terms of security aspects as well. Any process, where the biometric security is crucial for the system to ensure that correct person is autorized, or to detect any individual to be searched for, this is a useful application.

For e.g., to verify that the correct person is giving exams, in the examination systems, to verify that the authrized doctor has access to confidential areas or important machines in the hospital, to verify the criminal faces with proof images etc. Hence, the scope of visual identification & verification is very wide, given the accuracy of the system is high enough to be trusted. The scope lies in authenticating the validity of the person.

## 1.3 Comparison with Other Biometric Technologies

Facial recognition may not be the most effective & reliable among the othe biometric methods available. But, one key advantage is that it allows mass identification of test subject. Other tchniques such as, finger print recognition, speech recognition, iris scanning etc, need individual cooperations.

Moreover, since the application can be used in government training courses or labors' identification, the identifocation using the finger print recognition is not useful. As their fingers and thumbs skin is quite damaeged because of the nature of the work done by them, the finger prints can't be reocrded accurately & hence can't be vaildated. Such biometric technologies fail in these cases. Hence, facial identification & verification will be useful here.

Though, in case of mass identifications, questions are raised in case of security applications on public places as it may fail to reach accuracy where huge number of people are roaming and recognition has to be made from video streaming.[1]

## 1.4 Organization of Thesis

Rest of the thesis is organized as follows:
Chapter 2 (Background) gives an overview of background for such technology and purposes

of use. It narrates the methodology of the system proposed. In methodology, the system components & systems process are discussed. A brief overview of system architecture is given. Further, the limitations & assumptions of the systems are discussed.

Chapter 3 (Literature Survey) deals with the studies made in process of the project flow. It gives detailes of the case studies which have been made, i.e., basically on "FaceNet" paper. It also discusses OpenFace & the neural networks provided by it. The sizes, accuracies & performance data is given. The disadvantage & advantages of the system are discussed.

Chapter 4 (Smart Verification) addresses the task of facial verification, done in the prdouct, Smart Verification. It discusses the usecases and system methodoly of the product including components, process, architecture. The system is built using OpenFace. It also explores the implementation technologies used, proposed algorithms & the acquired results.

Chapter 5 (Smart Recognition) involves the complete description of Smart Recognition starting from old architecture to new architeture. Smart Recognition's new blackbox is built on using OpenFace. The new algorithm is proposed. The results got with experiments using new algorithm are presented. It also tells about the full architecture and applications of the system.

Chapter 6 (Third Party API Integeration with Web Interface) summarizes the use of third party API & the methodology used. The APIs are dynamic in nature. The web interface is created as a rest client for the APIs & is also dynamic in nature.

Chapter 7 (Cancer Cells Annotation UI) address the new solution to cervical cancer detection problem using artificail intelligence. In the whole process, the requirement to create and annotate the cancerous regions in the cells was made by pathologist, to fulfill that, this software has been created. It uses dynamic canvas operations.

Chapter 8 (Conclusions and Future Work): In this chapter, conclusions drawn from various implementations mentioned above are discussed. Future scope of work in this area is also mentioned in the chapter.
The Bibliography section consists of related research work cited in the thesis work.

# Chapter 2

# Background

## 2.1 Overview

For initiation & promotion of a deep, powerful & positive practices in an organization, it's an imperative requirement to Identify and Recognize people who are the critical stakeholders in the organization. They could be students, employees or customers. For attendance or security purposes, the face recognition & verification based applications will make the process automated & faster, generate digitized data, manage the data on servers and learn with the data generated.

Thus, Face Detection, Face Verification and Face Recognition are utilized in extensive applications such as security in an organizaiton, marking attendance of the employees or students, verifying and authenticating faces. Illumiantion, Heavy Computation, Image Quality, Image Position are some important issue of such development.

## 2.2 Methodology

### 2.2.1 Components

There are three major components in this project namely:

1. Client Application (Desktop & Mobile)
2. Server
3. Web Interface

These components are further classified into categories of sub-components which are described below.

**Client**

**"Client"** is the component of "Smart Verification" which is installed on the machines of the customers who are also the users of the product. There are two types of client applications to sophisticate the users:

1. Desktop Application
2. Mobile Application

Any of the applications can be used to enroll in the system or to send the attendance.

**Server**

Server is the backbone for all the processes and activities. In fact, it is the critical component of the product. Before getting into how things work on the server, lets first get acquainted with the organization of files on the server. Basically, the server, is a linux (Ubuntu) operating system. So, the file system and its organization would be the same as in a typical Linux Operating System. The APapche server is installed on it. Apart from that, we can categorize the file system into two from a product perspective: On the server side there are programs having algorithms for facial detection/ verification. The OpenCV library is used in server programs for image processing. The registered images will be matched during verification to verify if they match. The php programs on server are the ones that populate and query the database.

**Web Interface**

The web interface (http://aindra.in/attendance) is a User Interface which presents data to the users in varying granular levels and it has been built using Django. In deed, Django is a free and open source web application framework, written in Python, which follows the model view controller architectural pattern. The new web interface going to be released sooner, is being built using PHP and JavaScript.

### 2.2.2 Processes

Using the client application, there can be two types of processes:

- **Training:** This is a one time process and each employee has to be trained individually. To say in a lucid manner, images containing faces of the employee to be enrolled are captured in various different orientations based on which FR algorithms are run and added to the dictionary.
- **Recognition:** This is a regular (mostly, daily basis) process. Employees have to upload their image for recognition. Eventually, FR algorithms are run on those images and matches are found with the help of the dictionary created during enrollment. The results of recognition are populated on the web interface (GUI), where the employees, managers and admin can view their attendance and generate reports, as and when required.

- **Verification:** This is required at time process.. Employees have to upload their image for verification. Eventually, FR algorithms are run on those images and matches are found with the help of the source image given during enrollment. The results of verification are populated on the web interface (GUI), where the employees, managers and admin can view their verification status and generate reports, as and when required.

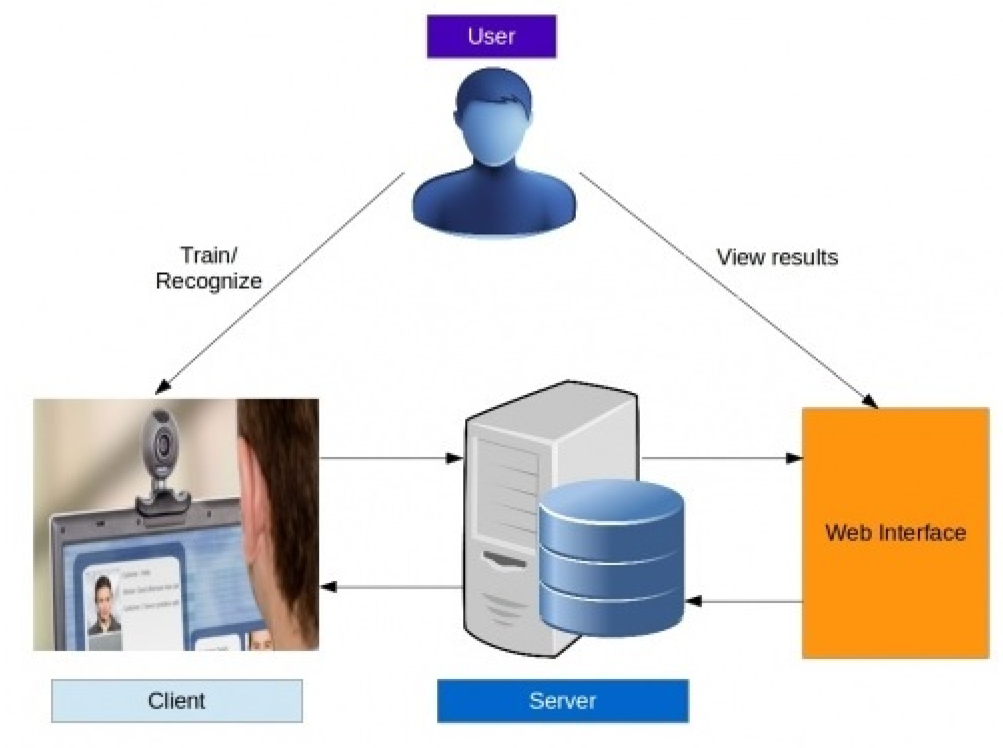


Figure 2.1: System Components

There is no need for any external hardware installation. Only a device with a camera (laptop with webcam, pc with webcam, smart phone) is required for capturing/uploading images to server. The actual process of face recognition happens on server, where algorithms are written to do this. The client application invokes the server programs on the cloud. Once the server programs nish execution, the results are displayed on the Web in terface. The web interface is interactive and user friendly.

The status of each person for that day will be shown on the web interface. Along with it, attendance data of the organization in various forms, reports will be provided for further processes and calculative help in organization records.

## 2.3 Assumptions

The following assumptions have been made for this study:

- The stakeholders in an organization or students in an educational institute have to give daily attendance & they are enabled/provided with a device which is equipped with a camera. They will also be able to see their records given credentials for the application web interface.
- The images they send for attendance and verification are taken in a particular format, e.g., in particular orientation, distance from camera, direction etc.

## 2.4 Limitations

The limitation of this project is that it cannot be applied for a real time scenario where the results are required immediately. This is because all the proessing has to be done on the server side and hence not on the client machine itself. High Computation power for image processing and generating result matrix reduces the speed of getting the results. Also, the illumination & posing in the images are issues of such systems.

# Chapter 3

# Literature Survey

This chapter provides an overview of the research done in the eld of Face Recognition. The rst section provides a brief overview of the FR techniques. The later part of this chapter discusses the significance of intelligent FR algorithms, architetures used in the systems, limitations of these and design issues.

## 3.1 Case Study:

### 3.1.1 FaceNet: A Unified Embedding for Face Recognition and Clustering

This study has been made using the FaceNet paper written by Google developers in 2015. In the area of face recognition implementing face verification and recognition, despite the recent significant advanceds, efficiently at large scale presents important challenges to current approaches being made. A system called FaceNet is presented in this study, which directly make & learns a particular mapping from facial images to a general compact & specific Euclidean spacing where the correspondance of distances is directly denoting to the measure of face similarity.[2] After we have the Euclidean space produced, such tasks as face verification, recognition and clustering may become easily available to implement with the use of the standard available techniques which can be applied with FaceNet embeddings as feature vectors.

The FaceNet method makes use of a deep convolutional network which is pretrained to straightly optimize the current itself's embedding, instead of having a intermediate bottleneck layer which was used in previous all such deep learning approaches. For trainig, they are using triplets of the roughly aligned face patches which can be either matching or non matching & are generated with the use of a triplet mining function. The major benefit of FaceNet based approach is representing a better efficiency: they have achieved state-of-the-art face recognition performance by making use of only 128-bytes per face detected.[2] On the massively used Labelled Faces in the Wild (LFW) dataset, their model is achieving an accuracy of 99.63 %, which is a new record. On YouTube Faces DB the accuracy achieved by them is 95.12 %, their system cuts down the rate of error in direct comparison to the results on both datasets by 30 % on which are best published yet.[3]

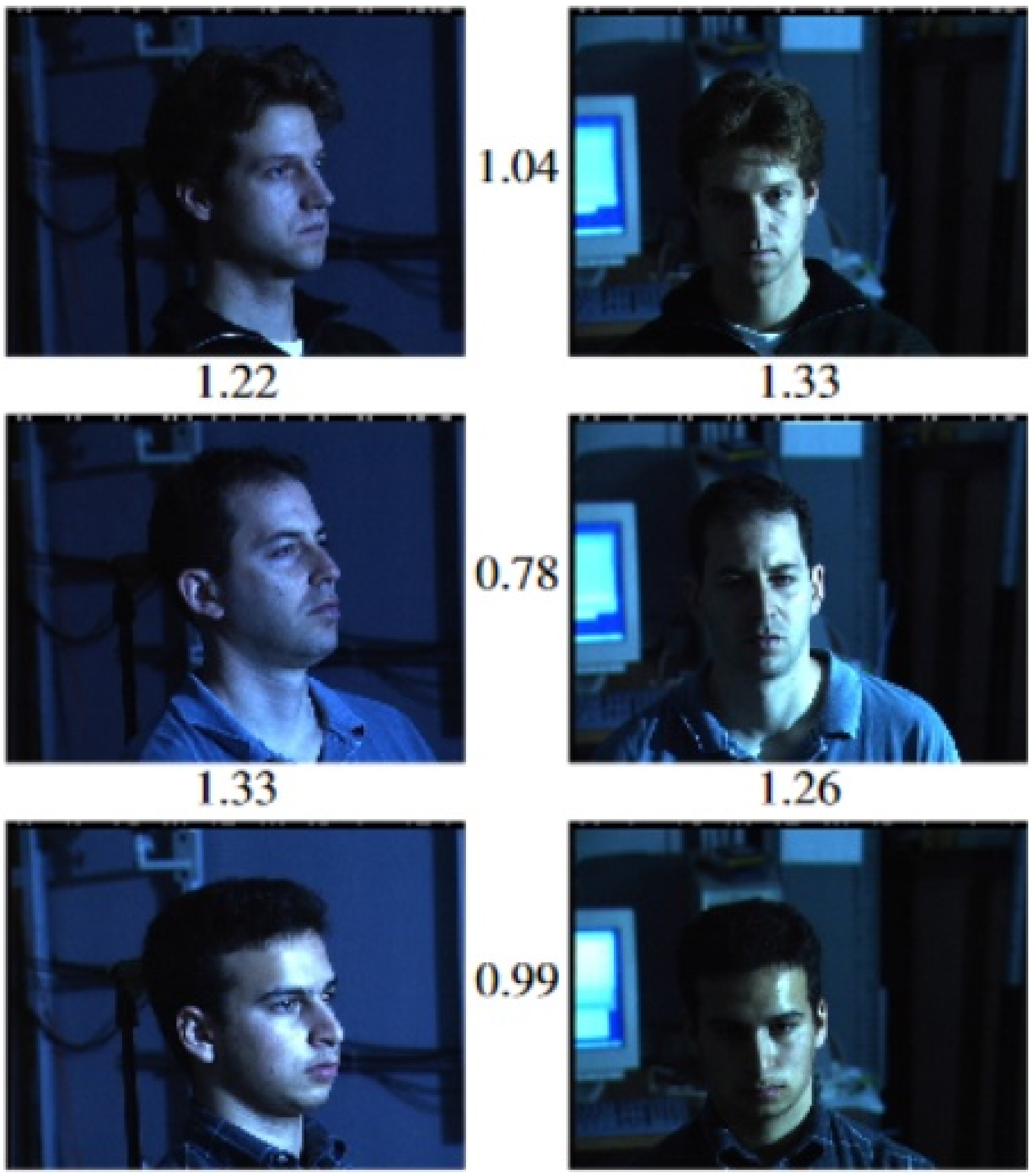


Figure 3.1: Illumination & Pose Variance

In face recognition, standard problems are pose and illumination. Here are shown the distances in output of Fcenet between the pairs of facial images of a different and the same person having various poses and illumination combinations. A distance saying 0.0 denotes the faces are same, whereas 4.0 denotes to the opposite data spectrum, giving two distinct identities. Here, a threshold of 1.1 is classifying every pair with correctness.

#### FaceNet Methodology

FaceNet is a system which is unified & is created for face verification (whether this is the correct person), recognition (finding who is the person) and clustering (to find similar persons among a group of persons). This approach is based on making & generating a Euclidean embedding of each image which is done by using a deep convolutional neural network. The training of network is done in sucgh a way that the squared L2 distances in the Euclidean embedding space are corresponding straightly to face similarity. The facial images of the same person have smaller distances compariltvely & the facial images of different people have larger distances.

After producing these embeddings, the task or techniques can be applied based on applications: for e.g., face verification involves the thresholding of the distance between the two embeddings & recognition can be done by applying classification techniques, likewise, clustering can be done using techniques such as k-means or agglomerative method of clustering.[2]

### 3.1.2 OpenFace Neural Network Models

OpenFace is a Python and Torch implementation with deep neural networks for facial recognition & verification, & is based on the CVPR 2015 paper FaceNet: A Unified Embedding for Face Recognition and Clustering.[4] In OpenFace, the Models are created using parameters is some numbers.These parameters are with embeddings which are 128 dimensional. Following table gives out a summary.

| Model | Number of Parameters |
|---|---|
| nn4.small2 | 3733968 |
| nn4.small1 | 5579520 |
| nn4 | 6959088 |
| nn2 | 7472144 |

Table 3.1: OpenFace Models Definitions

#### Pre-trained Models

To train the models, different methods with different datasets can be used. The pre-trained models released by openface are having versions and are relesed under relative model's definition. The currently existing models are given training with multiple conbinations of the largest two datasets which are for face recognition & are publically available: FaceScrub and CASIA-WebFace.[4]

The differences between alignment of models are given below:

| Model | Alignment (landmarkIndices) |
|---|---|
| nn4.v1 | openface.AlignDlib.INNER_EYES_AND_BOTTOM_LIP |
| nn4.v2 | openface.AlignDlib.OUTER_EYES_AND_NOSE |
| nn4.small1.v1 | openface.AlignDlib.OUTER_EYES_AND_NOSE |
| nn4.small2.v1 | openface.AlignDlib.OUTER_EYES_AND_NOSE |

Table 3.2: OpenFace Models Alignment Difference

### Performance

The performance is measured using Tesla K40 GPU & an 8 core 3.70 GHz CPU using OpenBLAS.[4]

The differences between alignment of models are given below:

| Model | Runtime (CPU) | | Runtime (GPU) | |
|---|---|---|---|---|
| nn4.v1 | 75.67 ms | 19.97 ms | 21.96 ms | 6.71 ms |
| nn4.v2 | 82.74 ms | 19.96 ms | 20.82 ms | 6.03 ms |
| nn4.small1.v1 | 69.58 ms | 16.17 ms | 15.90 ms | 5.18 ms |
| nn4.small2.v1 | 58.9 ms | 15.36 ms | 13.72 ms | 4.64 ms |

Table 3.3: OpenFace Models Performance

### Accuracy on LFW Benchmark

Benchmarking the standard LFW dataset, the accuracy appears to be remarkably high. They have used versions of deep funneled for 58 of total images in a number of 13233 as the dlib was failing to detect[4]

The differences between alignment of models are given below:

| Model | Accuracy | | AUC |
|---|---|---|---|
| nn4.small2.v1 (Default) | 0.9292 | 0.0134 | 0.973 |
| nn4.small1.v1 | 0.9210 | 0.0160 | 0.973 |
| nn4.v2 | 0.9157 | 0.0152 | 0.966 |
| nn4.v1 | 0.7612 | 0.0189 | 0.853 |
| FaceNet Paper (Reference) | 0.9963 | 0.009 | not provided |

Table 3.4: OpenFace Models Alignment Difference

### ROC Curves

The true positive rate of nn4.small1.v1 is higher compared to others like eigen faces, Deepface Ensemble etc. The graph shows the AUC and true positive v/s true false positive rates for different models.

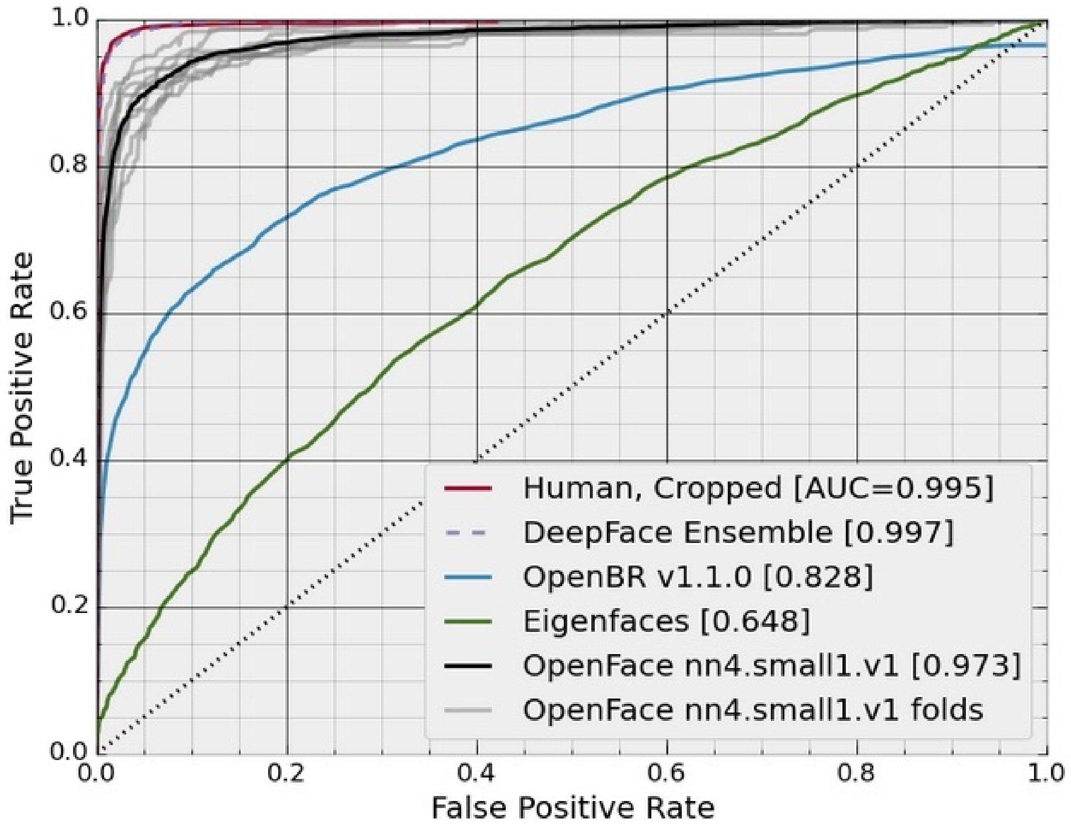


Figure 3.2: ROC Curve for nn4.small.v1

## 3.2 Advantages & Disadvantages

### 3.2.1 Weakness

Facial recognition techniques struggle to perform under certain conditions and enviroments. Such conditions which makes it harder to achieve success include facial accessories such as sunglasses, poor lighting, low resolution of images, long hair, beards & other object which may partially cover the face of subject. One more discovery concludes that facial recognition systems lack accuracy when facial expressions vary. Hence, at many places, neutral face expression is asked in passport photos. Some times, organization like Google, Nikon, Flicker are criticised for the inability of their softwares to recognize faces which have darker skin colors.The datasets which are being used by researchers also have inconsistency. It is imoportant to have a standard dataset to be used universally.

### 3.2.2 Effectiveness

**Case Study:** Critics of the technology claim that such systems having face recognition could be useful for implementation of procedures and ensure the discipline. For e.g., the London Borough of Newham scheme, 2004 was made to recognize criminals, using a database of hundreds of criminals living in Borough & the system was allowed to run for several years. The Newhamś automatic facial recognition system, though spotted live target for a few number of times, the system was credited for 34% reduction in crime. [1] The effectiveness was due to the face detection & recognition of the live criminals as well as by the notion when the common people were reguralry told that they are under consistent video surveillance which is equipped with advanced face recognition technology. Thus, this fear along with live spotting, both proved effective in reduction of crime rate. Thus, the technology & the usersṕerception of the technology make it more effective.

### 3.2.3 Privacy Issues

The use of Facial Recognition is not limited to just identify an individual but is also capable to unearth the other individual and personal data which is associated with her/him. Such as the pictures which are featuring the person, social networking profiles, internet behavior, their whreabouts and activities etc., which can be done through the facial features. Moreover, the people who have covered faces, these systems limites that ability and is crucial to privacy. This is fundamental change in the daily privacy preservation, by enabling the government, markets, agencies or any random person to secretly gather the indentities & the adjoining personal information & data of captured by such systems.

This lead the social websites such as **Facebook**, have been gathering very large number of pictures of people which are annonated by their names. Such big storage represents a database which could be used potentially by government/agencies or market for different purposes. Resulting, a hearng was held in July 2012 before the the Law of the Committee on the Judiciary & Subcommittee on Privacy, Technology, United States Senate [1], aiming to address issues which surround the meaning of this facial recognition technology for privacy and civil right and liberties.

# Chapter 4

# Smart Verification

**Smart Verification**, is a product which uses computer vision & machine learning to verify facial data to support particular applications.

## 4.1 Overview

**Smart Verification** is a unique way to ensure transparency between the registered and actual person while in an ongoing process. Once the process is triggered, our solution starts the verification of given image with the registered image. And the result is channelled to the authorized person. By this way the illegal authoriztion in particular programs can be prevented. Our solution plays major role in crusading such attempts by which nonvisual method of invalid authorization are carried out.

In a hospital scenario for a regular patient in a hospital tracking the medical history at right moment and minimum effort is tiresome. Our solution identifies and stacks the biometric traits of the patient in a safe location. When the patient seek medical support for the next time, he will be identified automatically and his medical history is briefed to the doctor.

### 4.1.1 Definition

**Smart Verification** is a software to record & verify biometric facial data using Computer Vision & Machine Learning. It works on one to one image mapping, i.e., it maps the actual data to expected data with some probability. It needs the target and source image both to create mapping. Generally, this software is to be used on requirement, not on routine basis (daily).

- The stakeholders in an organization or students in an educational institute are provided with handheld devices with inbuilt cameras like smartphones, tablets and laptops, the ability to record & verify their images.
- Images that are captured using such devices are sent to our cloud based server.
- They are processed by intelligent algorithms to detect and verify faces.

### 4.1.2 Problem Statement

There can be number of area, where this product can serve as solutions. Few of such are being discussed here:

- **Government Training Courses:**
  - In government provided training courses, such as computers, machinery, stitching etc., people do enroll to get certified but they do not attend the daily sessions.
  - At the time of examination, a fake candidate can come instead pf the real candidate who had enrolled and take the exam to get passing certificate.
  - To identify if the candidate was the same person who was enrolled, verification can be used.
- **Police or Security Organizations:**
  - In security organizations, there is data recorded for criminals in their databases.
  - Anytime if they get any image of any suspect and they want to verify that he is the same criminal.
  - To verify the identity of the criminal, verification can be used.
- **Healthcare Industry:**
  - In healthcare industry, the access to particular rooms and devices are given to only authorzed doctors. Also in some treatments like telepathy and teleradiology etc, only authorized doctors can give service.
  - Similarly in some cases, it can be used to verify patients also, where the health checkups are scheduled or where the doctor needs the history of patients.
  - To verify the identity of the doctor or patients, verification can be used.

### 4.1.3 Use Cases

1. When a user enrolls in a course, he/she gives a number of pictures in different orientations and are saved in folder on a daily/one time basis.
2. When the user comes to take exam for the respective course, his/her images are verified against the enrollment images.
3. This is to make sure that the correct person is taking exams.
4. The application can be used to verify persons for different needs. E.g., For criminal face verification.

**Use Case 1: Enrollment Process**

**Enrollment Process**, is a one time process when the user data is stored in the database. The images of the user is take in different orientations (neraly 45 images) and are stored on server.

In case of daily attendance in government certified courses or healthcare industry, daily images are also added in this storage to increase the accuracy of verification. Such images are stored in a folder dedicated to that particular person, called his/her source folder.

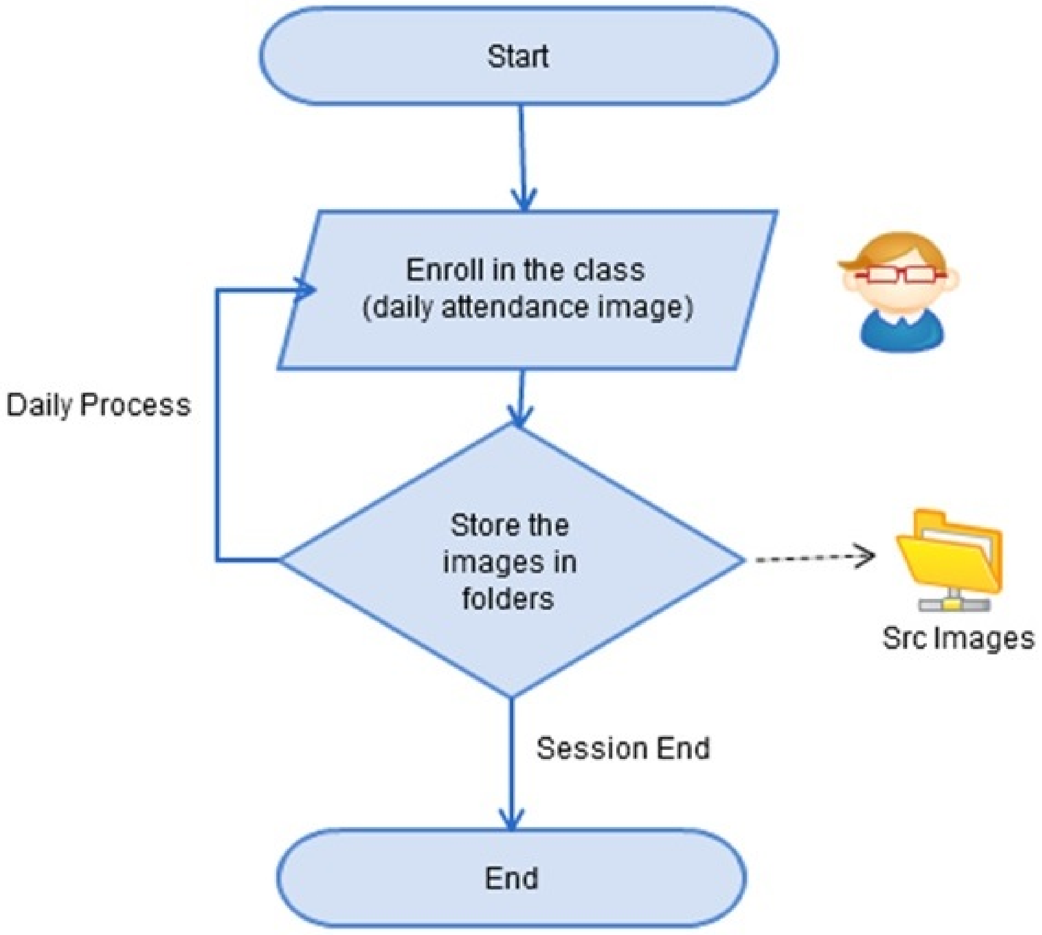


Figure 4.1: Smart Verification Scenario 1: Enrollment Process

**Use Case 2: Verification Process**

**Verification Process**, is a process when the user data has to be verified with the stored data. The images of the actual person is taken & are compared to the expected data(stored images).

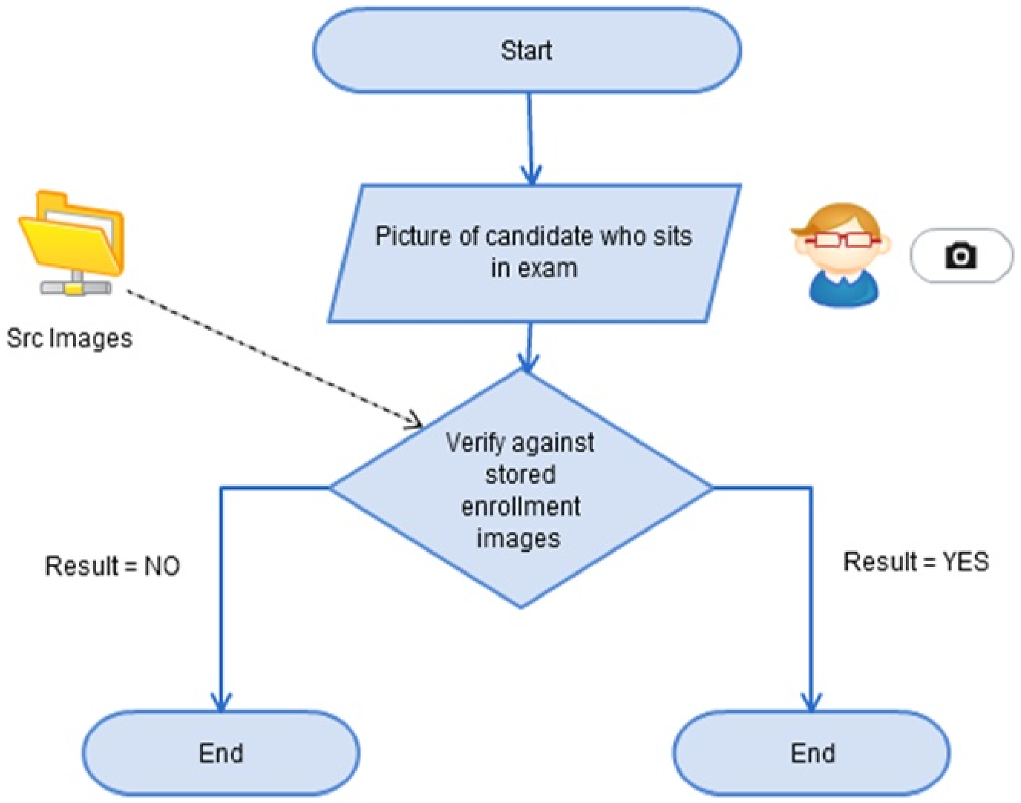

Start
Picture of candidate who sits in exam
Src Images
Verify against stored enrollment images
Result = NO
Result = YES
End
End

Figure 4.2: Smart Verification Scenario 2: Verification Process

## 4.2 System Architecture

### 4.2.1 Methodology

- In our method, we use a deep convolutional neural network and then learn a Euclidean embedding for each image.
- The network used are having pre-training such that the squared L2 distances given by the Euclidean embedding space straightly denote to probability of face similarity.
- Smaller distances mean that the faces are of same person and the larger the distance, higher the dissimilarity.
- In our verification system, the aim is to simply threshold the distance between the given Euclidean embeddings.

### 4.2.2 System Design

#### Components

The system is built using various components for interaction with user. There are three main system components are:

1. Client
2. Server
3. Web Interface

These components are further classified into categories of sub-components which are described below.

<u>**Client**</u>

**"Client"** is the component of "Smart Verification" which is installed on the machines of the customers who are also the users of the product. There are two types of client applications to sophisticate the users:

1. Desktop Application
2. Mobile Application

Any of the applications can be used to enroll in the system or to send the attendance.

**Desktop Application:** Client machine can be a computer system or a smart phone. The desktop application is developed using java swing. The two main functions of client application are to capture images of users and upload them to the server for new registration & verification respectively.

**Mobile Application:** Mobile application can be used on smartphones of tablets. The latest version of mobile application is built using android. The two main functions of client application are to capture images of users and upload them to the server for new registration & verification respectively. It also allows to record the latitiude & longitude of the user's location, which can be used to determine the location.

<u>**Server**</u>

Server is the backbone for all the processes and activities. In fact, it is the critical component of the product. Before getting into how things work on the server, lets first get acquainted with the organization of files on the server. Basically, the server, is a linux (Ubuntu) operating system. So, the file system and its organization would be the same as in a typical Linux Operating System. The APapche server is installed on it. Apart from that, we can categorize the file system into two from a product perspective:

| Activities | |
|---|---|
| Activity | Description |
| Login | User has to get the password before the registration process. This OTP is sent to the user via mail and verified later during first time registration. |
| Configure | When the app is configured to some other branch, location or organization by mistake or if the user wants to change the same for some reason, this activity will update the config.xml file to reflect the changes. |
| Registration | This is a one time or daily activity to register into system with user's images. |
| Verification | This is an activity by which user send her/his images and the algorithm verifies the face to validate the identity. |

Table 4.1: Smart Verification: Activities on Client

1. Core Server Operations
2. Web Interface Operations

On the server side there are programs having algorithms for facial detection/ verification. The OpenCV library is used in server programs for image processing. The registered images will be matched during verification to verify if they match. The php programs on server are the ones that populate and query the database.

**Cron Jobs:**
Cron jobs are time-based job scheduler in Unix-like computer operating systems. On the server, there are cron jobs running which take database backups every day at midnight (00:00:00).

**Web Interface**

The web interface (http://aindra.in/attendance) is a User Interface which presents data to the users in varying granular levels and it has been built using Django. In deed, Django is a free and open source web application framework, written in Python, which follows the model view controller architectural pattern. The new web interface going to be released sooner, is being built using PHP and JavaScript.

**Processes**

1. **Upload**

   The upload process takes place for two cases: Registration & Verification. The activities included in the process are:

| Group of Users | |
|---|---|
| User | Description |
| Admin | Admin is at the top of the hierarchy and he has all the privileges. An admin can generate license key for an organization, remove organizations, add/remove/edit managers, users, accept/reject the manual tag requests, if any etc. |
| Super User | This is the super user of the organization. He can add/remove/update/assign managers or add/remove/updat users in the organization at any level. He can also add/remove/update the levels of the hierarchy. |
| Manager | There can be managers at each level of the organization hierarchy. He is next to the admin in the user hierarchy and can Add or Remove employees under him apart from Approve/ Reject tag requests. |
| User | These users are normal employees (neither managers nor admin) having least privileges. They can view their verification status. |

Table 4.2: Smart Verification: Group of Users

- Creation of target folder: The target folder for the training images or verification images are created.
- Uploading the images: The image for which this upload.php is called, is uploaded and saved inside the respective target folder created on the server.
- Populating the web interface: Based on updated database, the web interface in populted with information.

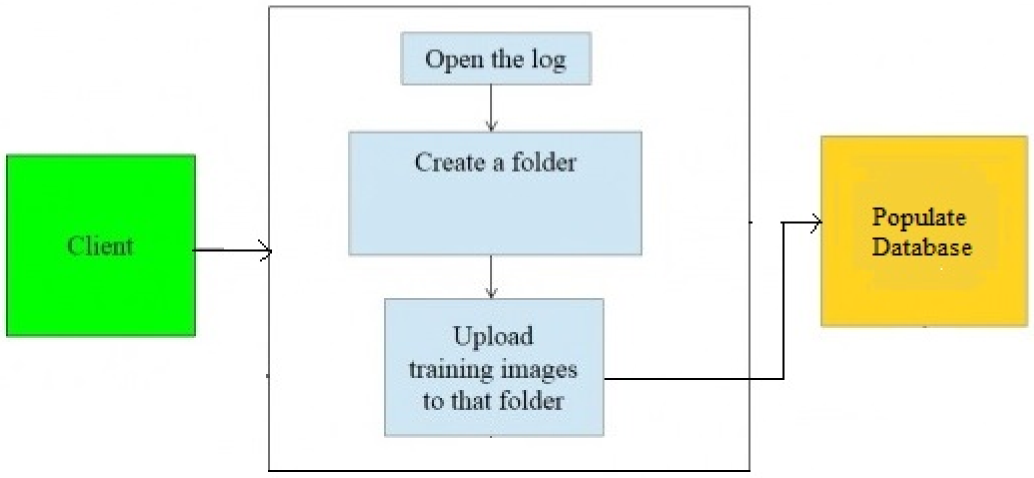


Figure 4.3: Upload process

2. **Registration**

   Registration, as the name insists, is the image registering phase of the Smart Verfiication system. It is a one time/daily process depneding on the application, whose component wise sequence of actions are explained as follows.

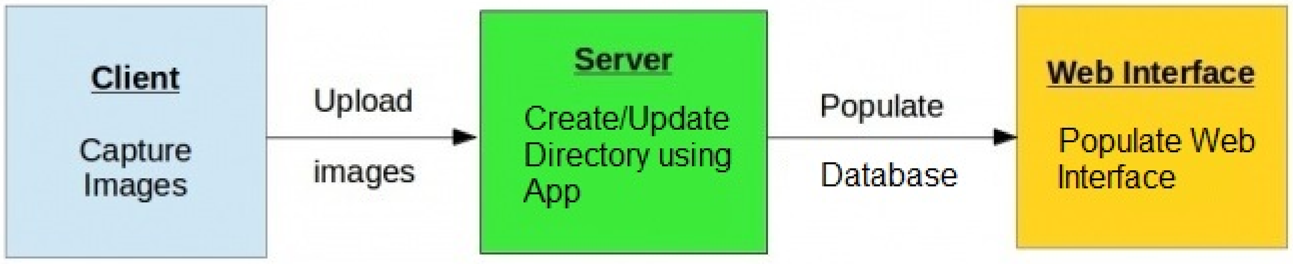


Figure 4.4: Registration process

   The activities included in the process are:

   - Notify the admin by email: This mechanism is to notify the admin and a mail is sent to ”XXXXX@aindra.in” with the organization, location and branch, session, time etc., details.
   - A source folder is created/updated for the particular person.
   - The path of the folder is stored in DB for Verification process usage.

3. **Verification**

   Verification is carried out whenever authorization validation has to be made. Like Registration, this also has to be initiated from the Client. The following diagram gives a broad picture of the interrelationship among the components during Verification.

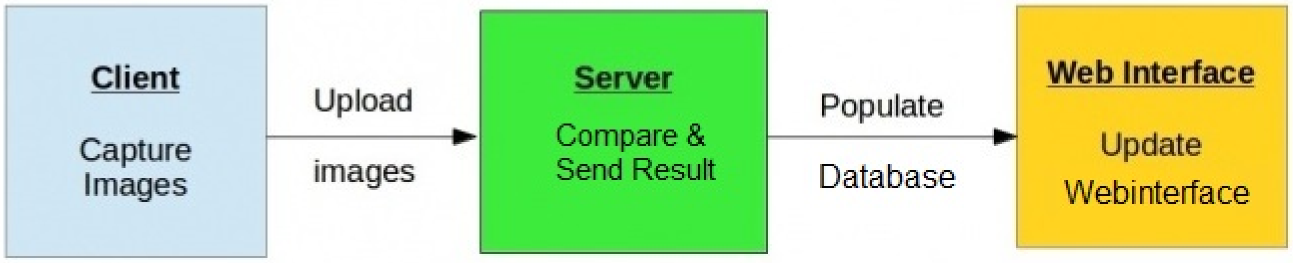


Figure 4.5: Verification process

   The activities included in the process are:

- Notify the admin by email: This mechanism is to notify the admin and a mail is sent to "XXXXX@aindra.in" with the organization, location and branch, session, time etc., details.
- Take each image from source folder.
- Run verification process with each of the image & based on the threshold for confidence, send result as "verified" or "failed".

#### System Dependencies

Following system dependencies are there:

- OpenCV
- OpenFace
- dlib
- torch
- python

## 4.3 Implementation

### 4.3.1 OpenFace

**OpenFace** is a Python and Torch implementation of face recognition with deep neural networks and is based on the CVPR 2015 paper **FaceNet: A Unified Embedding for Face Recognition and Clustering** by Florian Schroff, Dmitry Kalenichenko, and James Philbin at Google.[4] Torch allows the network to be executed on a CPU or with CUDA.

This research was supported by the National Science Foundation (NSF) under grant number CNS-1518865. Additional support was provided by the Intel Corporation, Google, Vodafone, NVIDIA, and the Conklin Kistler family fund.[4] Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and should not be attributed to their employers or funding sources.

#### Overview

Given is the overview of workflow using OpenFace for a single input image of Sylvestor Stallone from the LFW dataset which is publically available. Processing of images using OpenFace is similar to this.

1. Using the pre-trained models from dlib or OpenCV, detect the faces.
2. Transform the face for the neural network. This repository uses dlib's real-time pose estimation with OpenCV's affine transformation to try to make the eyes and bottom lip appear in the same location on each image.

3. Use a deep neural network to represent (or embed) the face on a 128-dimensional unit hypersphere. The embedding is a generic representation for anybody's face. Unlike other face representations, this embedding has the nice property that a larger distance between two face embeddings means that the faces are likely not of the same person. This property makes clustering, similarity detection, and classification tasks easier than other face recognition techniques where the Euclidean distance between features is not meaningful.

4. Apply your favorite clustering or classification techniques to the features to complete the recognition task.

**Workflow**

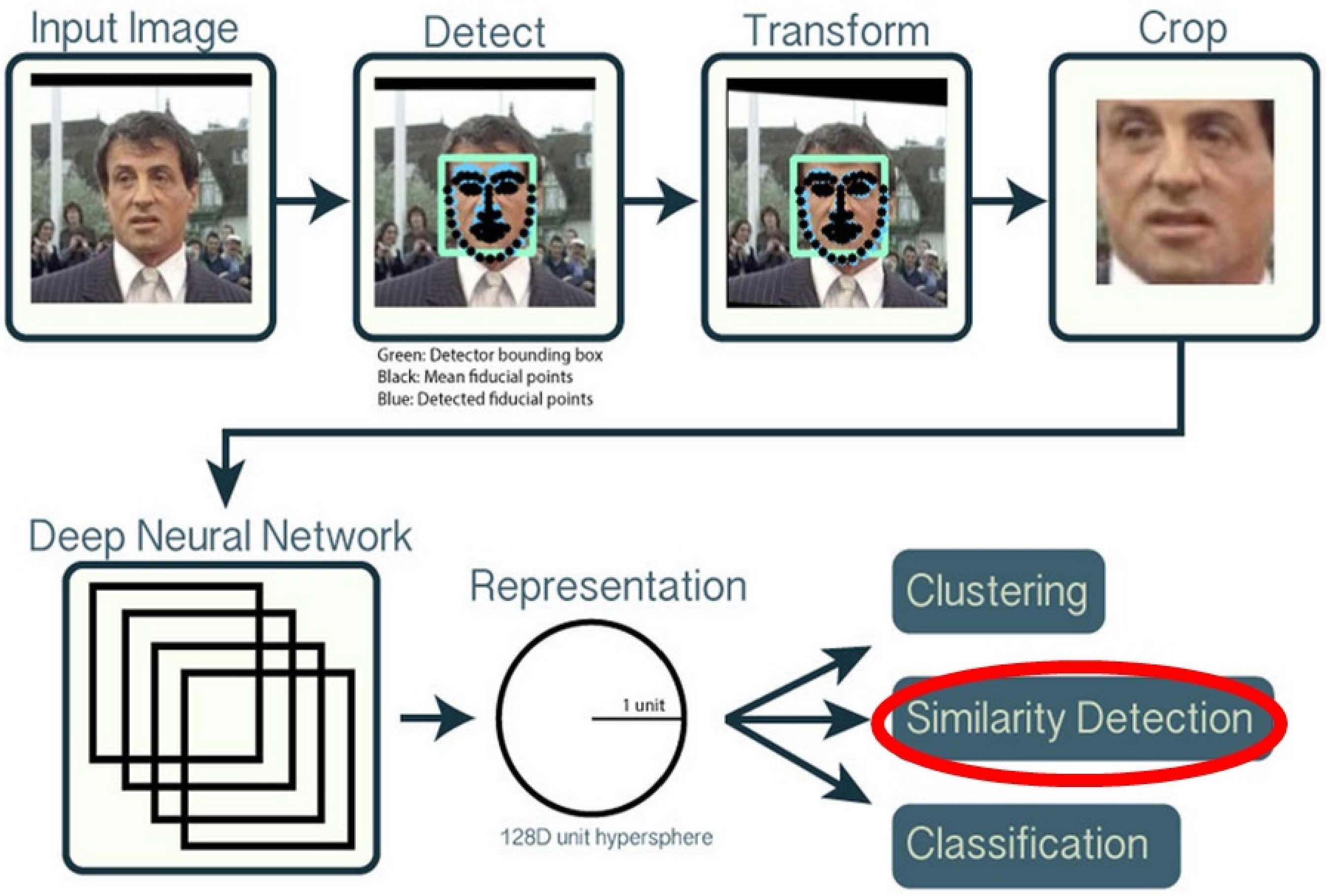


Figure 4.6: OpenFace WorkFlow

### 4.3.2 Workflow

**System Workflow**

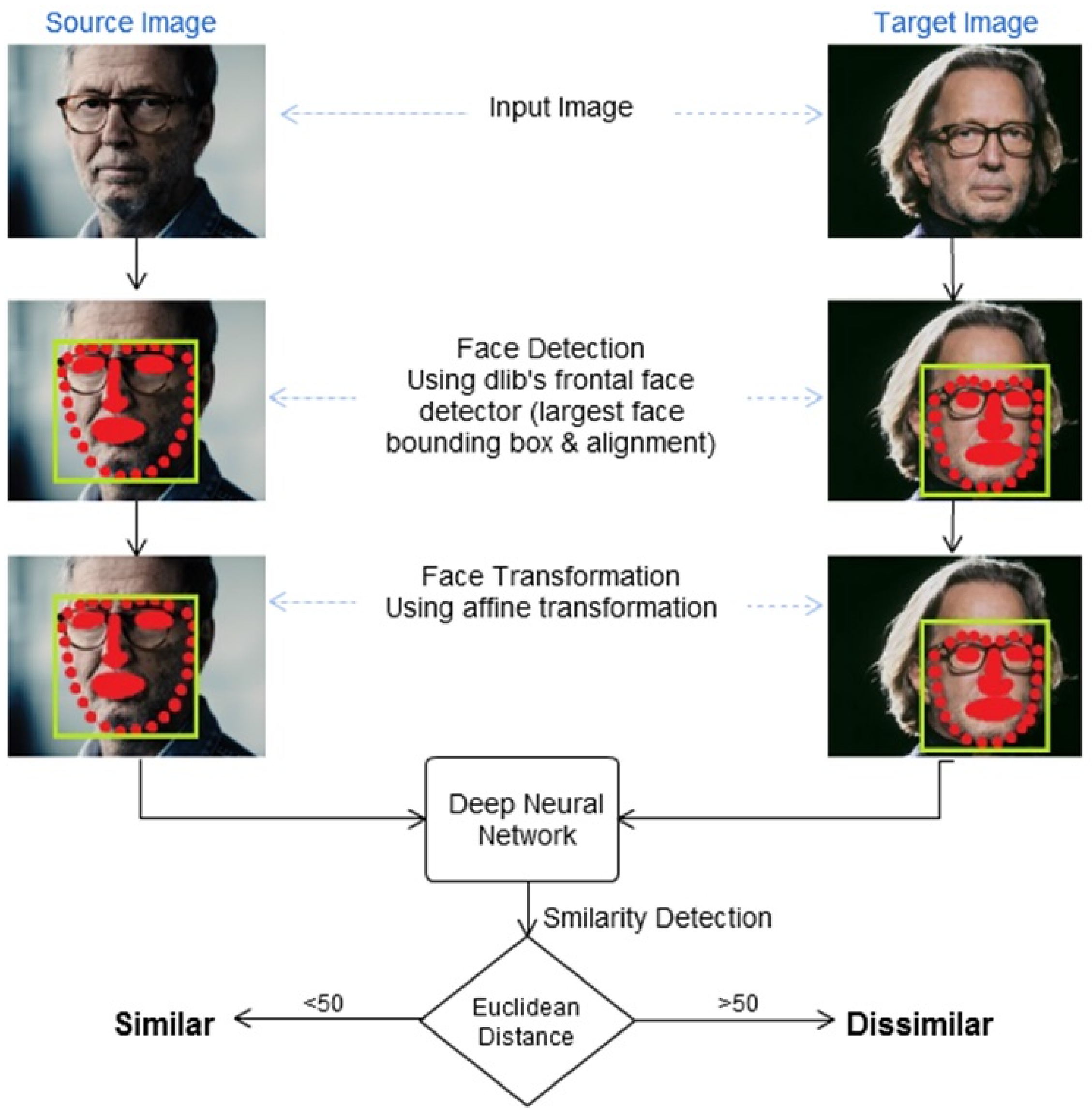


Figure 4.7: SmartVerification WorkFlow

**Face Detection Using Dlib**

- Tool for creating histogram-of-oriented-gradient (HOG) based object detectors.
- Technique for detecting semi-rigid objects in images.
- Tool that makes training HOG detectors super fast and easy.
- Needs as input is a set of images and bounding boxes around faces.

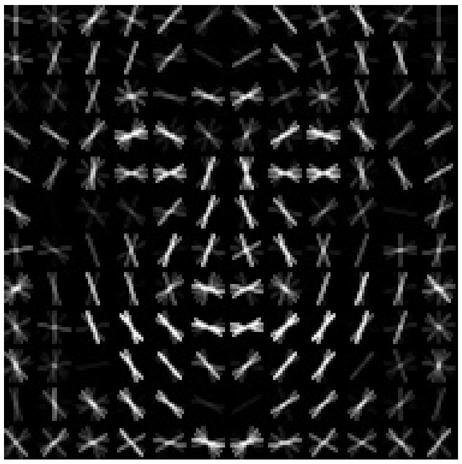


Figure 4.8: Face Detector

- It takes about 6 seconds to do its training using the example face data provided with dlib.
- Once finished it produces a HOG detector capable of detecting faces.
- Compare this to the time it takes to train OpenCV's popular cascaded haar object detector, which is generally reported to take hours or days to train and requires you to fiddle with false negative rates and all kinds of spurious parameters. HOG training is considerably simpler.

**Picture alignment**

- Dlib's implementation of the paper: One Millisecond Face Alignment with an Ensemble of Regression Trees.

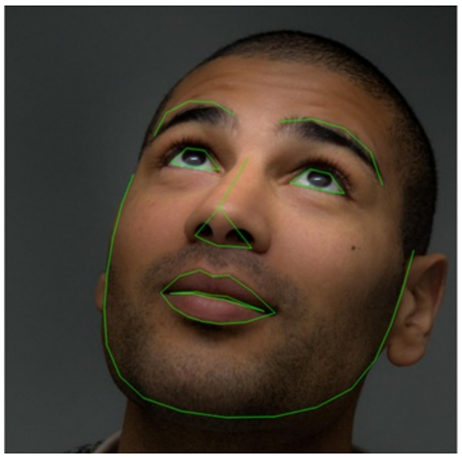


Figure 4.9: Face Alignment

- If you give an image of someone's face it will add this kind of annotation
- This is the output of dlib's new face landmarking example program on one of the images from the HELEN dataset.[5]

**Affine Transformation**

Affine transformations is any transformation that can be expressed in the form of a matrix multiplication (linear transformation) followed by a vector addition (translation).

We can use an Affine Transformation to express:

- Rotations (linear transformation)
- Translations (vector addition)
- Scale operations (linear transformation)

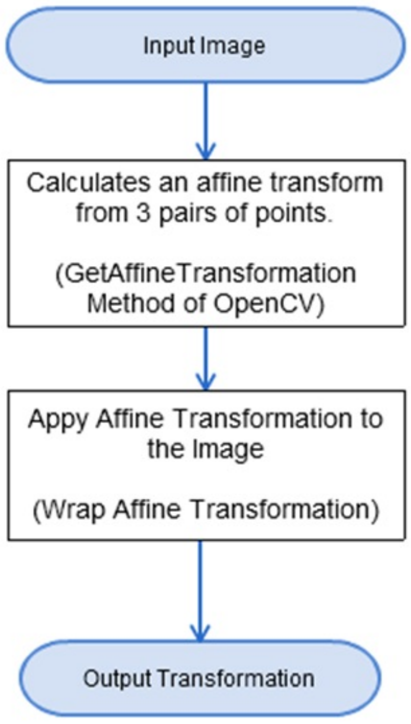

Input Image
Calculates an affine transform from 3 pairs of points.
(GetAffineTransformation Method of OpenCV)
Appy Affine Transformation to the Image
(Wrap Affine Transformation)
Output Transformation

Figure 4.10: Affine Transformation

### 4.3.3 Algorithm

For the complete process, two jobs are required: Enrollment Job & verification job.

### Enrollment Job

The algorithm for enrollment job is as follows:

**Algorithm 1** Verification: Enrollment Job

```
Create a directory on server with person name.
Add the source folder path next to the person's source column in database.
for each file do
    Upload the file to directory.
end for
```

### Verification Job

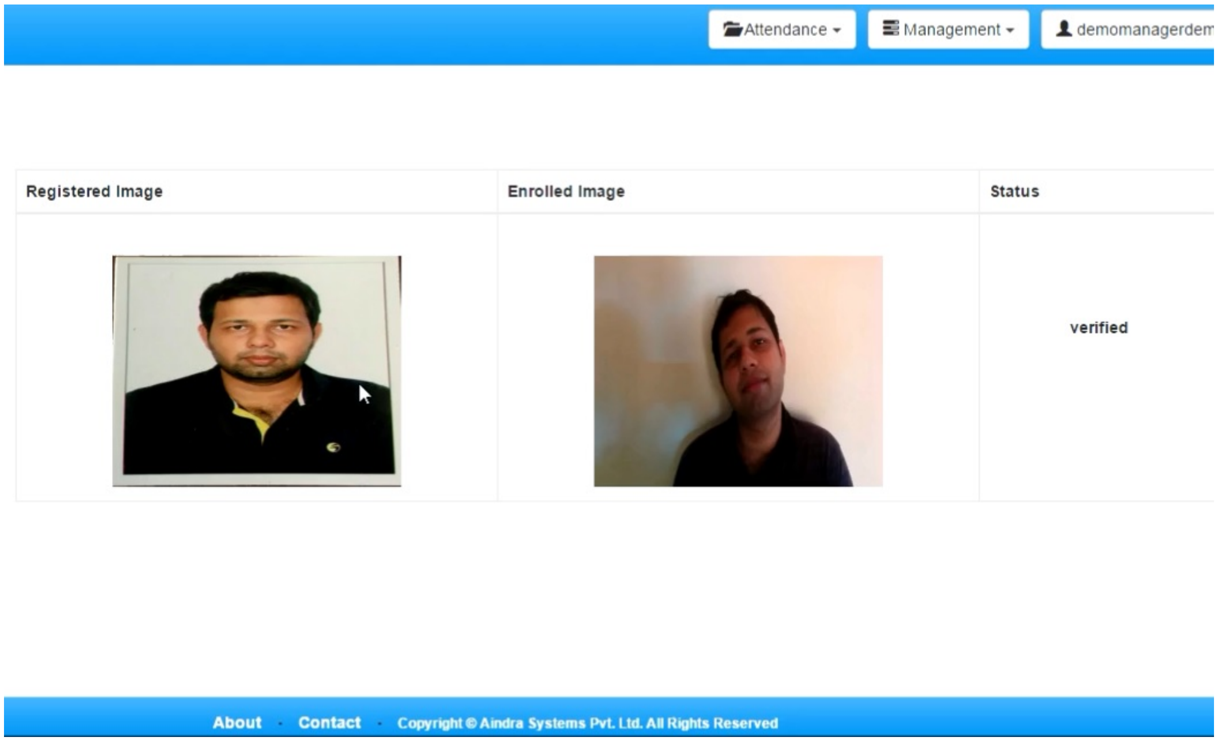
The algorithm for verification job is as follows:

**Algorithm 2** Verification: Verification Job

```
Get the largest face from target image using dlib's face_landmarks.
Transform and align the face extratcted.
Get the representation by passing that face to forwardImage()
Get the list of files from source directory
$positive \leftarrow 0$
$negative \leftarrow 0$
for file in list of files do
    $source\_image \leftarrow file$
    Get the largest face from source_image using dlib's face_landmarks.
    Transform and align the face extratcted.
    Get the representation by passing that face to forwardImage()
    Computer dissimalirty between both representations using norm.
    if $dissimilarity >= 50$ then
        $positive \leftarrow positive+1$
    else
        $negative \leftarrow negative+1$
    end if
end for
if $positive >= ngative$ then return True
elsereturn False
end if
```

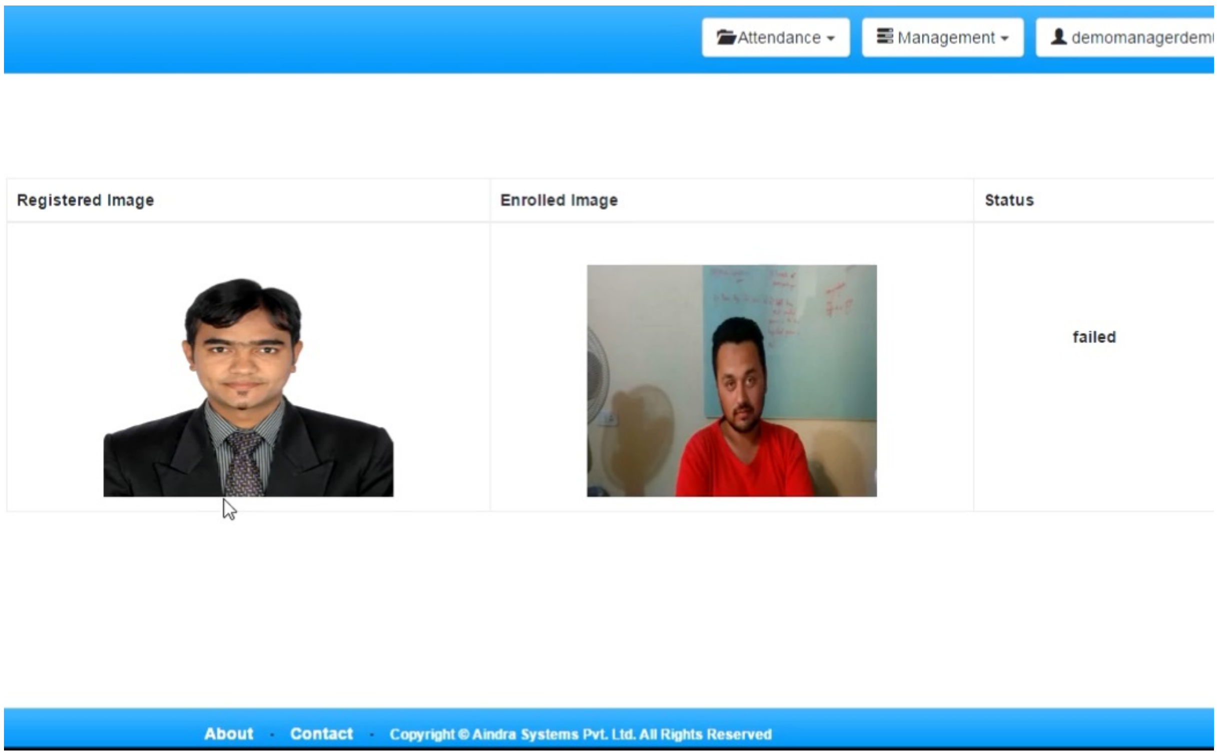
### SmartVerification GUI

Following are the GUI examples for verified and failed cases.


Attendance
Management
demomanagerdem
About · Contact · Copyright © Aindra Systems Pvt. Ltd. All Rights Reserved

| Registered Image | Enrolled Image | Status |
|---|---|---|
| | | verified |

Figure 4.11: Smart Verification: GUI(Verified Case)


Attendance
Management
demomanagerdem
About · Contact · Copyright © Aindra Systems Pvt. Ltd. All Rights Reserved

| Registered Image | Enrolled Image | Status |
|---|---|---|
| | | failed |

Figure 4.12: Smart Verification: GUI(Failed Case)

## 4.4 Results

### 4.4.1 OpenFace Experiment: Comparing Two Images

In Openface, the result carried out by comparing two images for two persons, both having slight variations are shown here. The distances between both show the similarity or dissimilarity probability.

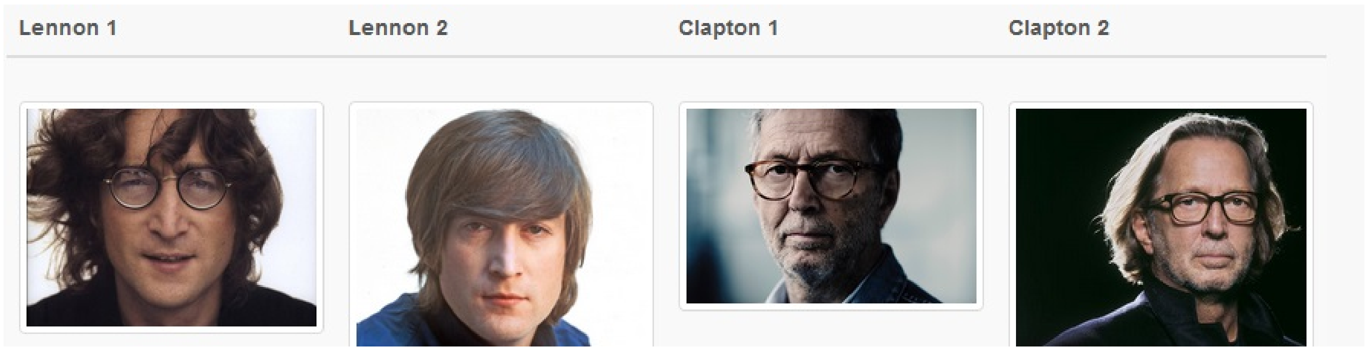


Figure 4.13: OpenFace Experiment: Comparing Two Images

| Image 1 | Image 2 | Distance |
|---|---|---|
| Lenon1 | Lennon 2 | 0.763 |
| Lenon1 | Clapton 1 | 1.132 |
| Lenon1 | Clapton 2 | 1.145 |
| Lennon 2 | Clapton 1 | 1.447 |
| Lennon 2 | Clapton 2 | 1.521 |
| Clapton 1 | Clapton 2 | 0.318 |

Table 4.3: OpenFace Experiment Results

### 4.4.2 Accuracy

The accuracy said in the FaceNet paper is remarkably high on the LFW datasets. Also, it gives a big magnitude on the Youtube data sets. 3

Given are the results acquired on by FaceNet & our experiments, respectively:

- **Ideal (FaceNet)**:
  - On the widely used Labeled Faces in the Wild (LFW) dataset: 99.63%
  - On YouTube Faces DB: 95.12%
- **Smart Verification:** On around 150 image pairs (both negative & positive cases): 89.63%

## 4.5 Applications

The applications can be all such areas where the access to a particular process or resource needs authorization. It can handle fraud cases. For e.g.,

- Candidate authorization before exams
- Healthcare Industry
- Security organizations
- Prospective Industries

# Chapter 5

# Smart Attendance

By applying the Face Recognition Techniques (FRT), this cloud based people recognition system will be developed which reduces the load on the client side while running the algorithms on the server. The first step is to capture and store images locally using a camera. Then upload these images to cloud using internet where image processing and face recognition happens on servers.

The final step is to display the results of recognition. This is our ultimate goal of face recognition. The accuracy of results is determined by light illumination, data sets and poses variation.

## 5.1 Overview

**Smart Attendance** is a unique way to ensure that the correct person was present at the given time and location. Once the process is triggered, our solution starts the recognition of given image with the pool of registered images. And the result is channelled to the authorized person. By this way the illegal attendance in organizations can be prevented. Our solution plays major role in crusading such attempts by which nonvisual method of invalid attendance are carried out.

In any organization, tracking of correct attendance at right time and minimum effort is tiresome. Our solution identifies and stacks the biometric traits of the person with location. When the users give attendance, they are identified against their training images and attendance is automatically marked, stored & briefed to the manager/teacher.

### 5.1.1 Definition

**Smart Attendance** is a software to record & identify biometric facial data using Computer Vision & Machine Learning. It works on one to many image mapping, i.e., it maps the actual data to available all the data with some probability. It needs only the target image to create mapping with training image. Generally, this software is to be used on routine basis, for attendance.

- The stakeholders in an organization or students in an educational institute are provided with handheld devices with inbuilt cameras like smartphones, tablets and laptops, the ability to record & verify their images.
- Images that are captured using such devices are sent to our cloud based server.
- They are processed by intelligent algorithms to detect and identify faces.
- Attendance is marked and stored in database, to populate web interface and briefed to higher authority.

### 5.1.2 Problem Statement

A general statement of the problem of machine recognition of faces can be formulated as follows: given still images of a scene, identify or verify one or more persons in the scene using a stored database of faces. The solution to the problem involves segmentation of faces (face detection) from cluttered scenes, feature extraction from the face regions, recognition, or verification. In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verfication problems, the system needs to confim or reject the claimed identity of the input face.

There can be number of area, where this product can serve as solutions. Few of such are being discussed here:

- **Educational Organiztions:**
  - To automate attendance of students and teachers.
  - Aiming to create Smart Education System
- **Industrial Organiztions:**
  - To automate attendance of employees.
  - Aiming to create Smart Industries
- **Healthcarel Organiztions:**
  - To automate attendance of patients and staff.
  - Aiming to create Smart Healthcarel Industry

### 5.1.3 Use Cases

1. When a user enrolls in a class or company, he/she gives his/her training images to system.
2. These training images are processed by blackbox for training.
3. On routine basis, user giver attendance images.

4. These images are processed by blackbox, the faces are detected and recognized using the already trained system.

5. Attendance is marked on basis of recognized images and database is populated and thus, is web interface.

**Use Case 1: Enrollment Process**

**Enrollment Process**, is a one time process when the user data is stored in the database. The images of the user is take in different orientations (neraly 45 images) and are stored on server.

In case of daily attendance in organization, system keeps learning using attendance images also. Such images are stored in a folder dedicated to that particular person, called his/her source folder.

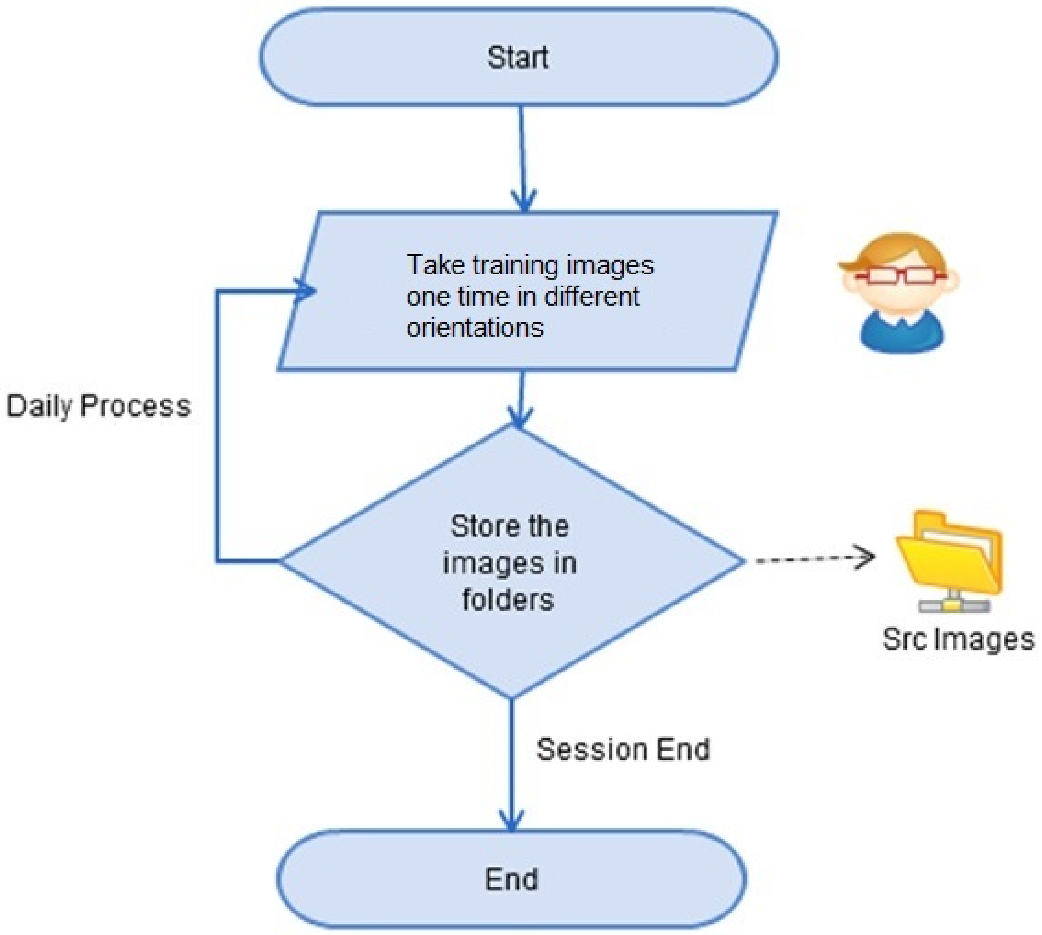

Start
Take training images one time in different orientations
Daily Process
Store the images in folders
Src Images
Session End
End

Figure 5.1: Smart Attendance Scenario 1: Enrollment Process

**Use Case 2: Recognition Process**

**Recognition Process**, is daily process of giving attendance. The images of the user are processed to detect faces and once detected to recognize among all users in that branch.

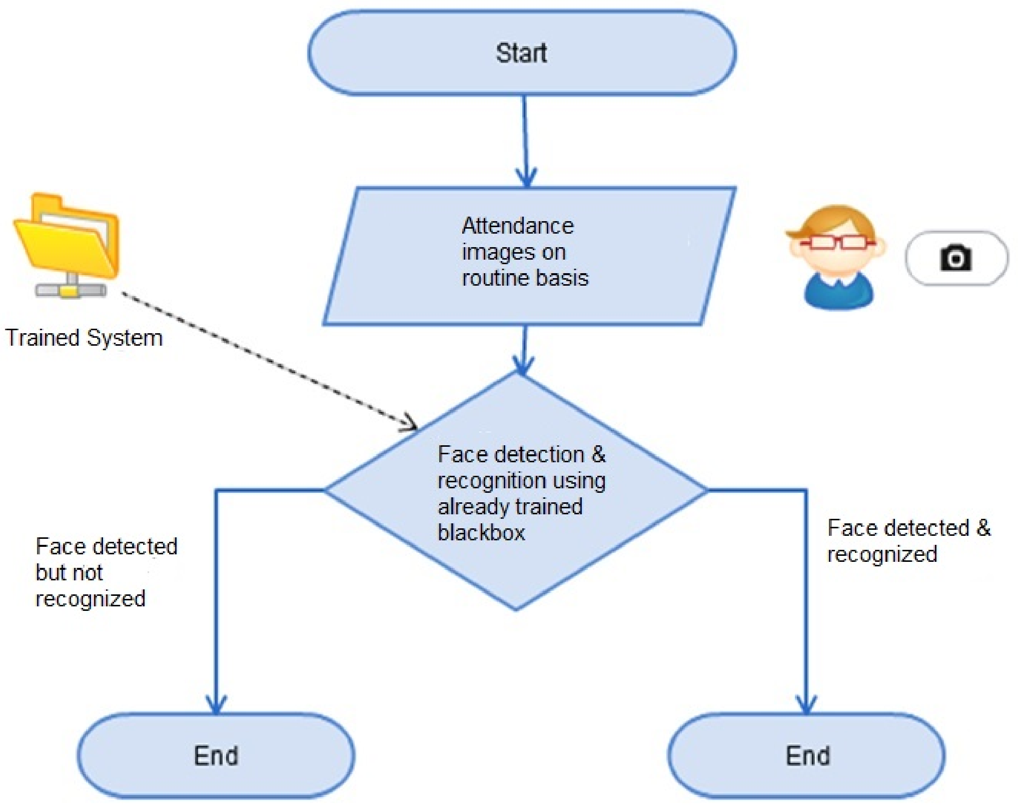

Start
Attendance images on routine basis
Trained System
Face detection & recognition using already trained blackbox
Face detected but not recognized
Face detected & recognized
End
End

Figure 5.2: Smart Attendance Scenario 2: Recognition Process

## 5.2 System Architecture

### 5.2.1 Methodology

- Our method is based on creating a classifier which is basically an SVM.
- Using training images of each person, representations are created. —item Using those representations, an SVM is created.
- This SVM is used to classify the testing image. It takes representation of face and makes a prediction with some probability and confidence.

### 5.2.2 Old Architecture

The system is currently using the old architecture and is very soon about to be relaunched with the new architecture. The old architectre used django framework to be used as MVC. The database was tightly coupled by the web interface by django and which was not able to provide much felibility. Moreover, the market demand arised for standalone APIs so that organizations could build their own web interface according to their requirements and usage. The old architecture allowed only one web interface, in which the users of every organization had to login and do the operations. To meet the requirements, the need to change the django architecture arised. Currently, the last phase of old architecture is running. Earlier, the hierarchy level of the organization was limited to three levels, (Organization-Location-Branch), now thr new architecture consists of unlimited levels in hierarchy.

### 5.2.3 New Architecture

In the new architecture, the Java RESTful APIs are created to interact with database and which are standalone. Hence, they can be used by the customers and they can utilize them according to their needs, build their own interface, applications etc. The web interface for "Smart Attendance" is also rebuilt with PHP integerating the APIs to interact with database. This new web interface is sooner to be launched. Since the hierarchy is unlimited and undefined in this architecture, the APIs behaviour is dynamic in two dimensions.

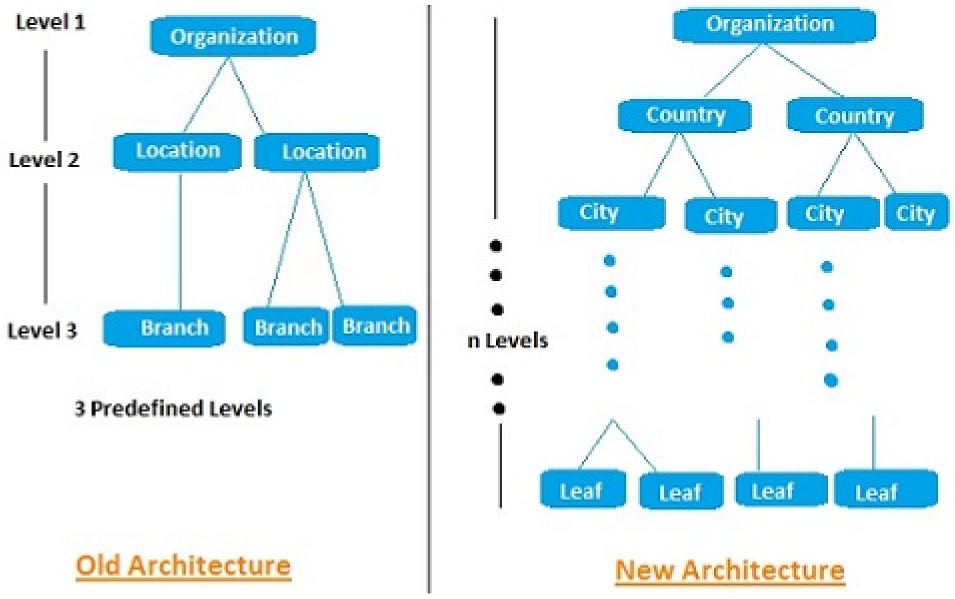


Figure 5.3: Comparison between Hierarchy Levels in New & Old Architecture

#### Multitier Architecture

The multitier architecture using Application layer(Presentation Layer, Business Logic Layer & Data Access Layer) & Data Store Layer is shown following:

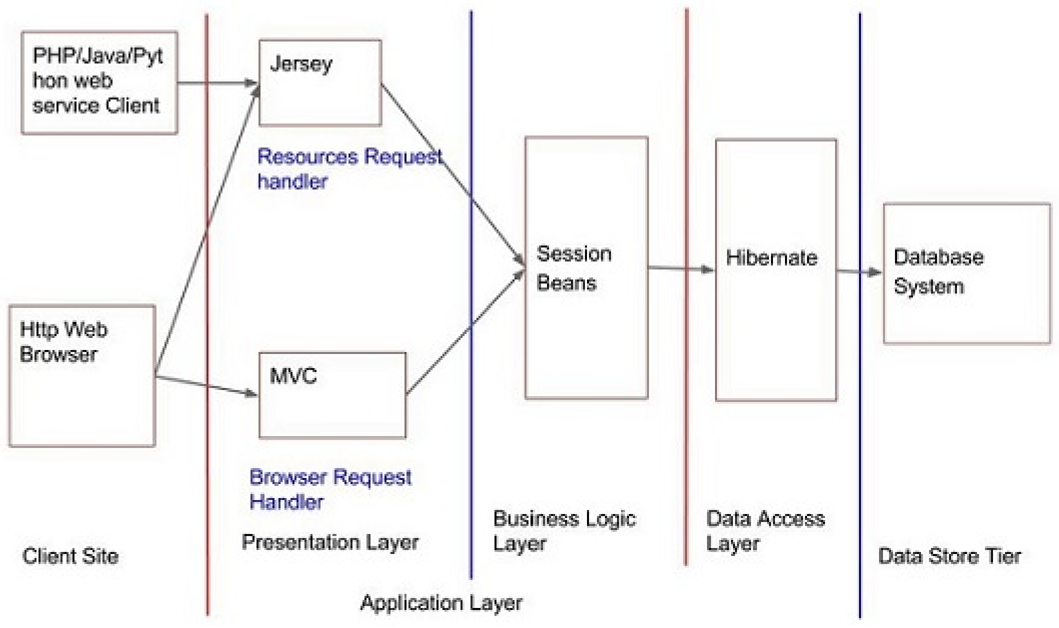


Figure 5.4: Multitier Architecture

### 5.2.4 System Components

The system is built using various components for interaction with user. There are three main system components are:

1. Client
2. Server
3. Web Interface

These components are futher classified into categories of sub components which are described below.

#### <u>Client</u>

**"Client"** is the component of "Smart Attendance" which is installed on the machines of the customers who are also the users of the product. There are two types of client applications to sophisticate the users:

1. Desktop Application
2. Mobile Application

Any of the applications can be used to enroll in the system or to send the attendance.

**Desktop Application:** Client machine can be a computer system or a smart phone. The desktop application is developed using java swing . The two main functions of client application are to capture images of users and upload them to the server for new enrollment or recognition respectively.

**Mobile Application:** Mobile application can be used on smartphones of tablets. The latest version of mobile application is built using android. The two main functions of client application are to capture images of users and upload them to the server for new enrollment or recognition respectively. It also allows to record the latitiude & longitude of the user's location, which can be used in onfield attendance.

| Activities | |
|---|---|
| Activity | Description |
| Login | User has to get the password before the enrollment process. This OTP is sent to the user via mail and verified later during enrollment. |
| Configure | When the app is configured to some other branch, location or organization by mistake or if the user wants to change the same for some reason, this activity will update the config.xml file to reflect the changes. |
| Enrollement | This is a one time activity to train the algorithm with user's images. |
| Recognition | This is a daily activity by which user send her/his images and the trained algorithm recognizes the face to mark the attendance. |

Table 5.1: Activities on Client

The two main activities include Enrollment & Recognition.

1. **Enrollement:** Enrollement process consists following steps:
    - Receive OTP (One Time Password)
    - Verify Password
    - Capture Images
    - Store in apt directory
    - Trigger upload.php for each image
    - Trigger Train.php once ]
    - Dictionary created

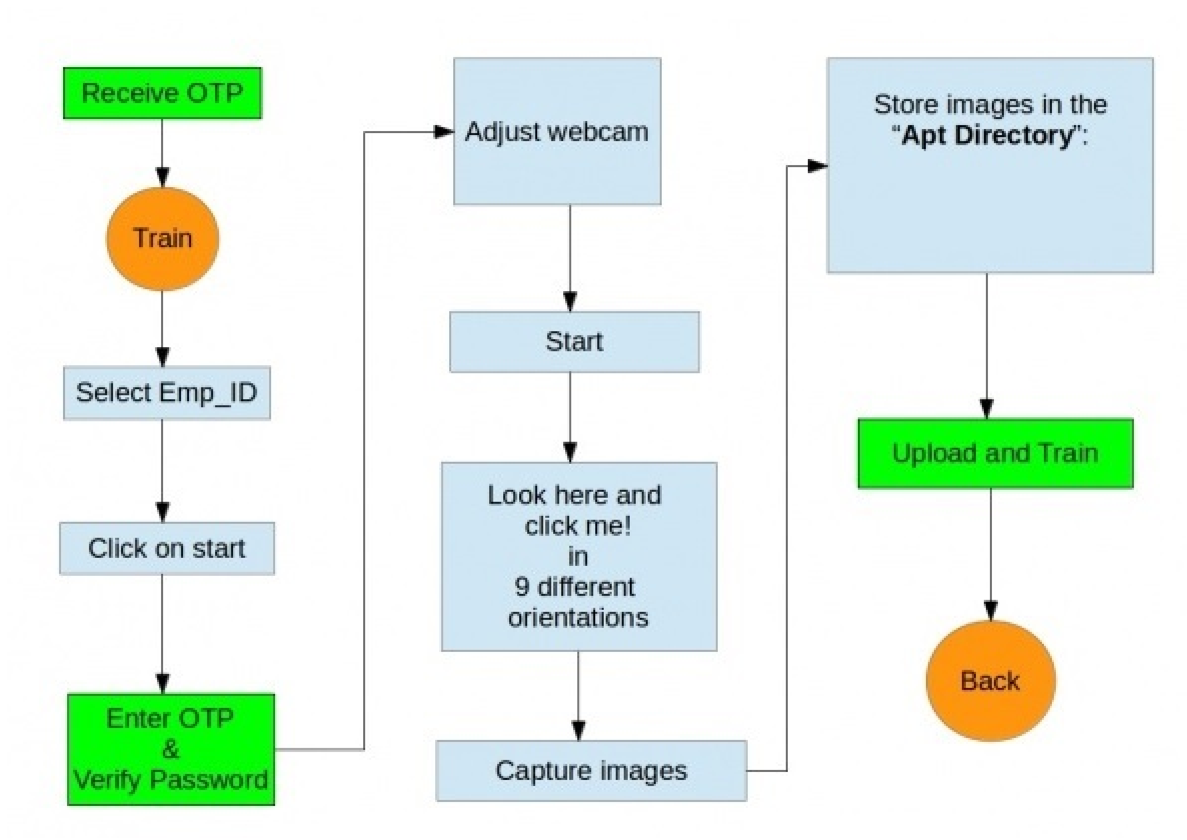

Receive OTP
Train
Select Emp_ID
Click on start
Enter OTP & Verify Password
Adjust webcam
Start
Look here and click me! in 9 different orientations
Capture images
Store images in the "Apt Directory":
Upload and Train
Back

Figure 5.5: Enrollment

2. **Recognition:** It is a daily activity to give attendance. The following figure depicts the series of activities during the process of Recognition related to the Client machine. Recognition process consists following steps:

    - Login into application
    - Select Hierarchy
    - Select Session
    - Capture Pictures
    - Store in apt directory
    - Trigger upload.php for each image
    - Trigger Recognize.php once

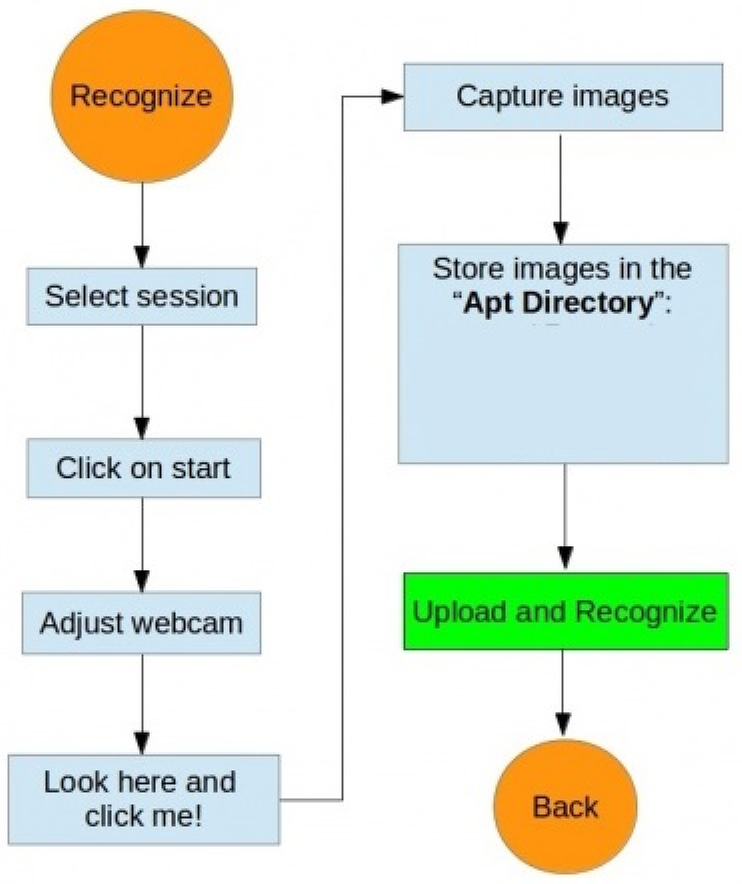


Figure 5.6: Recognition

**Server**

Server is the backbone for all the processes and activities. In fact, it is the critical component of the product. Before getting into how things work on the server, lets first get acquainted with the organization of files on the server. Basically, the server, is a linux (Ubuntu) operating system. So, the file system and its organization would be the same as in a typical Linux Operating System. The APapche server is installed on it. Apart from that, we can categorize the file system into two from a product perspective:

1. Core Server Operations
2. Web Interface Operations

On the server side there are programs having algorithms for facial detection/ recognition, to train the images sent during enrolment for the rst time and recognize the images sent during recognition against dictionary of images for a particular user for marking attendance later. All of these vital data is stored in appropriate tables in the database as objects. The OpenCV library is used in server programs for image processing. The training or enrolment is a one-time process done initially to create a database with known faces. These images will be matched during recognition to verify if they match, to mark their attendance. The php programs on server are the ones that populate and query the database.

**Cron Jobs:**
Cron jobs are time-based job scheduler in Unix-like computer operating systems. On the server, there are cron jobs running which take database backups every day at midnight (00:00:00).

The following diagram will give an insight about the number and the sequence of file triggers on the server.

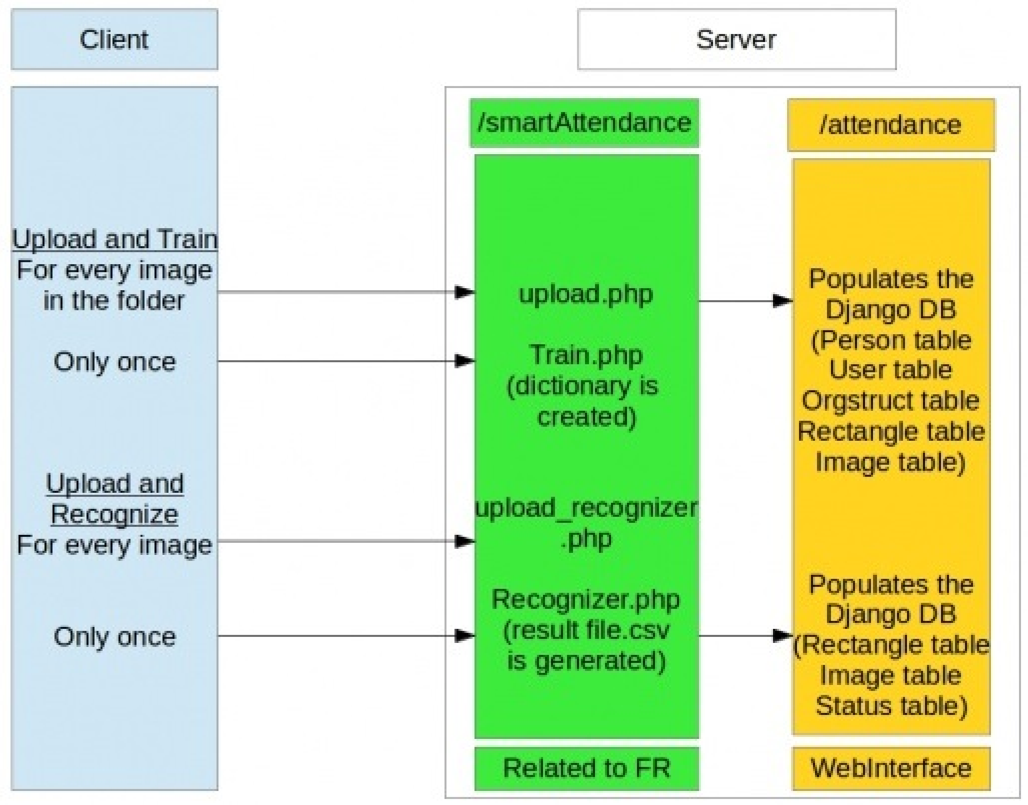

Client
Server
/smartAttendance
/attendance
Upload and Train
For every image in the folder
Only once
Upload and Recognize
For every image
Only once
upload.php
Train.php (dictionary is created)
upload_recognizer .php
Recognizer.php (result file.csv is generated)
Populates the Django DB (Person table User table Orgstruct table Rectangle table Image table)
Populates the Django DB (Rectangle table Image table Status table)
Related to FR
WebInterface

Figure 5.7: Interaction with Server

## Web Interface

The web interface (http://aindra.in/attendance) is a User Interface which presents data to the users in varying granular levels and it has been built using Django. In deed, Django is a free and open source web application framework, written in Python, which follows the model view controller architectural pattern. The new web interface going to be released sooner, is being built using PHP and JavaScript.

The activities on the web interface include:

| Group of Users | |
|---|---|
| User | Description |
| Admin | Admin is at the top of the hierarchy and he has all the privileges. An admin can generate license key for an organization, remove organizations, add/remove/edit managers, users, accept/reject the manual tag requests, if any etc. |
| Super User/Principal | This is the super user of the organization. He can add/remove/update/assign managers or add/remove/updat users in the organization at any level. He can also add/remove/update the levels of the hierarchy. |
| Manager/Teacher | There can be managers at each level of the organization hierarchy. He is next to the admin in the user hierarchy and can Add or Remove employees under him apart from Approve/ Reject tag requests. Managers can view the attendance in Self and Employee modes. |
| Employee/Student | These users are normal employees (neither managers nor admin) having least privileges. They can view their attendance. |

Table 5.2: Group of Users

- Login & Logout
- Select hierarchy level
- View attendance summary
- View individual attendance
- Select view type (Daily/Monthly/Yearly)
- Date navigation
- Generate reports
- Change enrollment image
- Generate tag request if required
- Add/update/remove user
- Add/update/remove/assign manager
- Add/update/remove hierarchy

### 5.2.5 System Processes

#### Upload

The upload process takes place for two cases: Training & Recognition. The activities included in the process are:

- Creation of target folder: The target folder for the training images or recognition images are created.
- Uploading the images: The image for which this upload.php is called, is uploaded and saved inside the respective target folder created on the server.
- Creating/Opening text file: A text file is creates at the time of training to be used in recognition process.
- Triggering process: Based on action, it triggers Train.php or Upload.php
- Populating the web interface: Based on updated database, the web interface in populted with information.

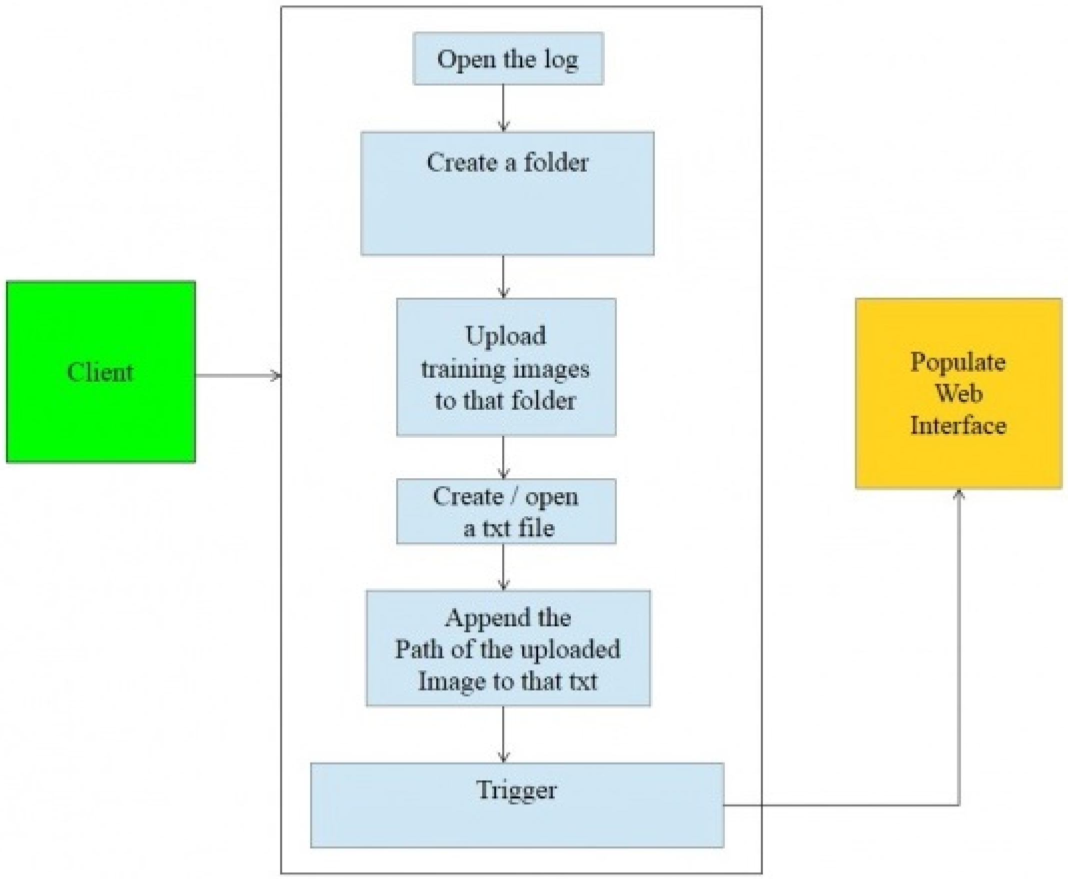

Open the log
Create a folder
Client
Upload training images to that folder
Populate Web Interface
Create / open a txt file
Append the Path of the uploaded Image to that txt
Trigger

Figure 5.8: Upload process

### Train

Training, as the name insists, is the learning phase of the Face Recognition system. It is a one time process whose component wise sequence of actions are explained as follows.

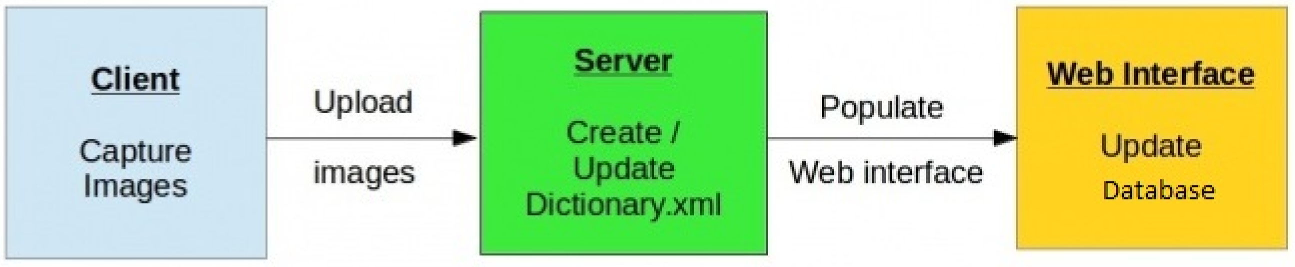


Figure 5.9: Training process

The activities included in the process are:

- Notify contact@aindra.in: This mechanism is to notify the admin and a mail is sent to "contact@aindra.in" with the organization, location and branch, session, time etc., details.
- Open dictionary file: The dictionary file is opened to be updated.
- Run "Trainer.exe": By passing the txt file (created by upload.php) and dictionary.xml of the correspoding branch as arguments, "./Trainer" is executed. In short, this dictionary is updated with some values pertaining to the newly enrolled employee by running Face Recognition (FR) Algorithms.
- Delete the txt file: The txt file created by upload.php is deleted.

### Recognition

Recognition is a daily process. Like Training, this also has to be initiated from the Client. The following diagram gives a broad picture of the interrelationship among the components during Recognition.

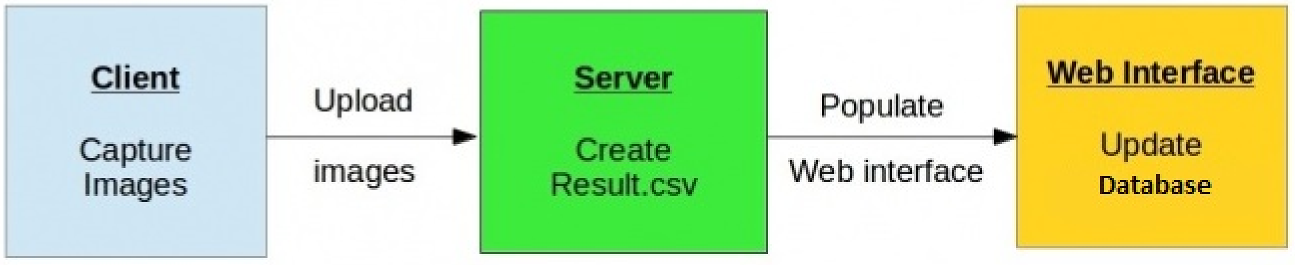


Figure 5.10: Recognition process

The activities included in the process are:

- Notify contact@aindra.in: This mechanism is to notify the admin and a mail is sent to "contact@aindra.in" with the organization, location and branch, session, time etc., details.
- Open dictionary file: The dictionary file is opened to be used.
- Run "Recognizer.exe": By passing the txt file (created by upload.php) and dictionary.xml of the correspoding branch as arguments, "./Recognizer" is executed. In short, this dictionary is used with some values pertaining to the enrolled employee by running Face Recognition (FR) Algorithms.

#### Stubbing

A method [stub] or simply stub in software development is a piece of code used to stand in for some other programming functionality. A stub may simulate the behavior of existing code (such as a procedure on a remote machine) or be a temporary substitute for yet-to-be-developed code.

Say, we have made some changes in the algorithm (FR algorithm either in training or in recognition part). This change will be reflected only for the newly coming images in future. But, what to do for those present on the server already? Will redoing training/recognition manually be a solution? Here comes the job of the stubs to ease our job. In general, stubs can be written for any purpose and in any desired programming language depending on the need. Sometimes, they will be required for testing also (without waiting for the whole process to complete, we can stub for the required portion alone).

## 5.3 BlackBox

**Black Box** is the name given to the system component which contains all the learning algorithms. It's the intelligent component of the system. First it's trained with user images while enrollment process and then leraning on daily basis through attendance takes place. Black box's older version had a limit to the accurcy. Thus, the new algorithm is created to make it more accurate, efficient and faster.

### 5.3.1 Old Black Box

The old black box was fully written in C++. It had two features, training & recognition. It used eigenfaces & fishefaces bases for computations.

### 5.3.2 New Black Box

The new black box is written using openface in python. Along with training & recognition, it also has a verification feature. It trains a classifier which is basically an SVM & uses it to classify the images. The implementation of new blackbox is given in the next section.

## 5.4 Implementation

### 5.4.1 System Dependencies

Following system dependencies are there:

- OpenCV
- OpenFace
- dlib
- torch
- python

#### OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.[6]

#### dlib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is open source software and licensed under the Boost Software License.[7] We are using it for multiclass classification, a multiclass SVM, a tool for solving the optimization problem associated with structural support vector machines, structural SVM tools for sequence labeling, structural SVM tools for object detection in images & SVM classification algorithm.

#### Torch

Torch is a scientific computing framework with wide support for machine learning algorithms that puts GPUs first. It is easy to use and efficient, thanks to an easy and fast scripting language, LuaJIT, and an underlying C/CUDA implementation.[8]

### 5.4.2 OpenFace

**OpenFace** is a Python and Torch implementation of face recognition with deep neural networks and is based on the CVPR 2015 paper **FaceNet: A Unified Embedding for Face Recognition and Clustering** by Florian Schroff, Dmitry Kalenichenko, and James Philbin at Google.[4] Torch allows the network to be executed on a CPU or with CUDA.

This research was supported by the National Science Foundation (NSF) under grant number

CNS-1518865. Additional support was provided by the Intel Corporation, Google, Vodafone, NVIDIA, and the Conklin Kistler family fund.[4] Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and should not be attributed to their employers or funding sources.
The detailed workflow for openface was shared in chapter 3. Here only the technique after representation changes to classification.

**Workflow**

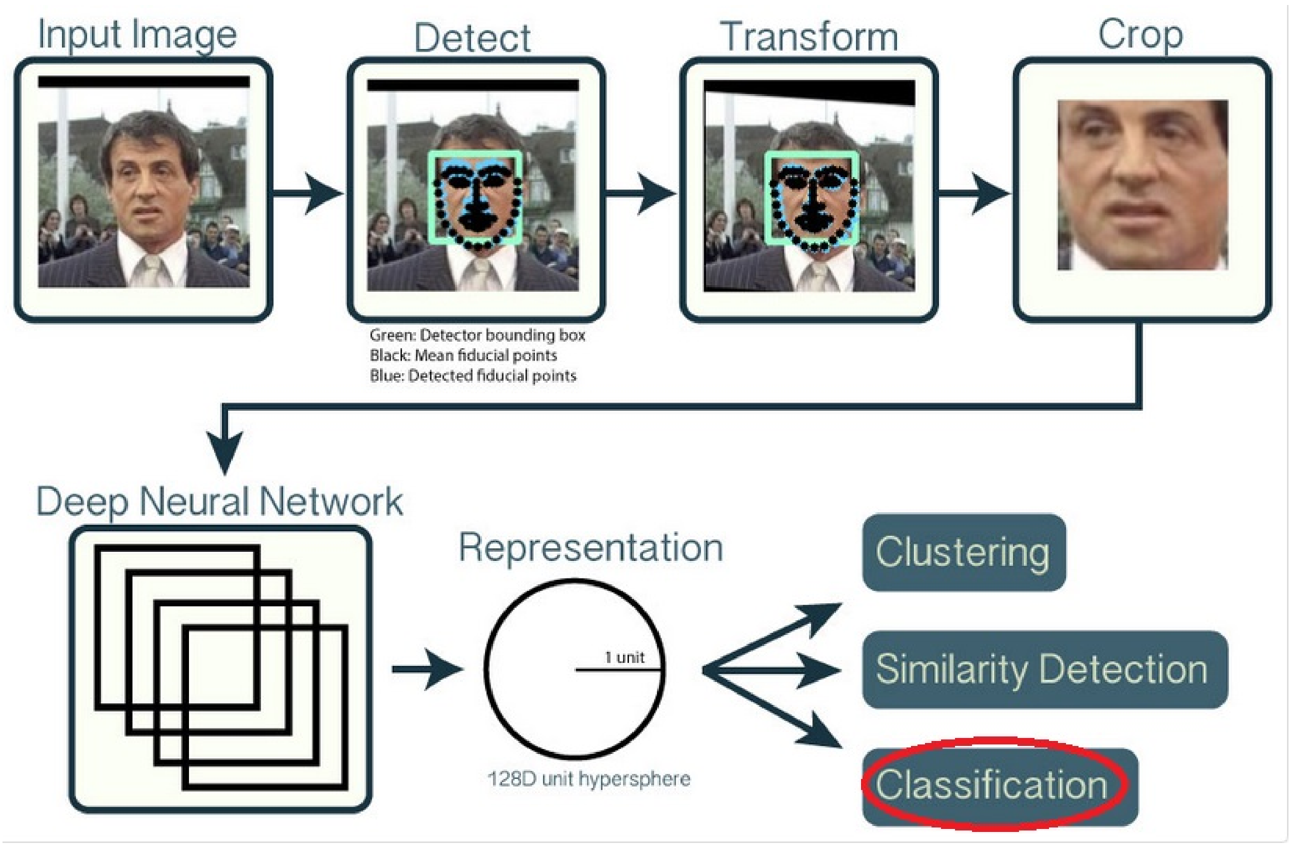


Figure 5.11: OpenFace WorkFlow

### 5.4.3 Workflow

Following diagram summarizes the workflow of the blackbox. The training images are given in a folder, for a person. Similarly, testing images are also given using a folder and as a result, result.csv file is generated.

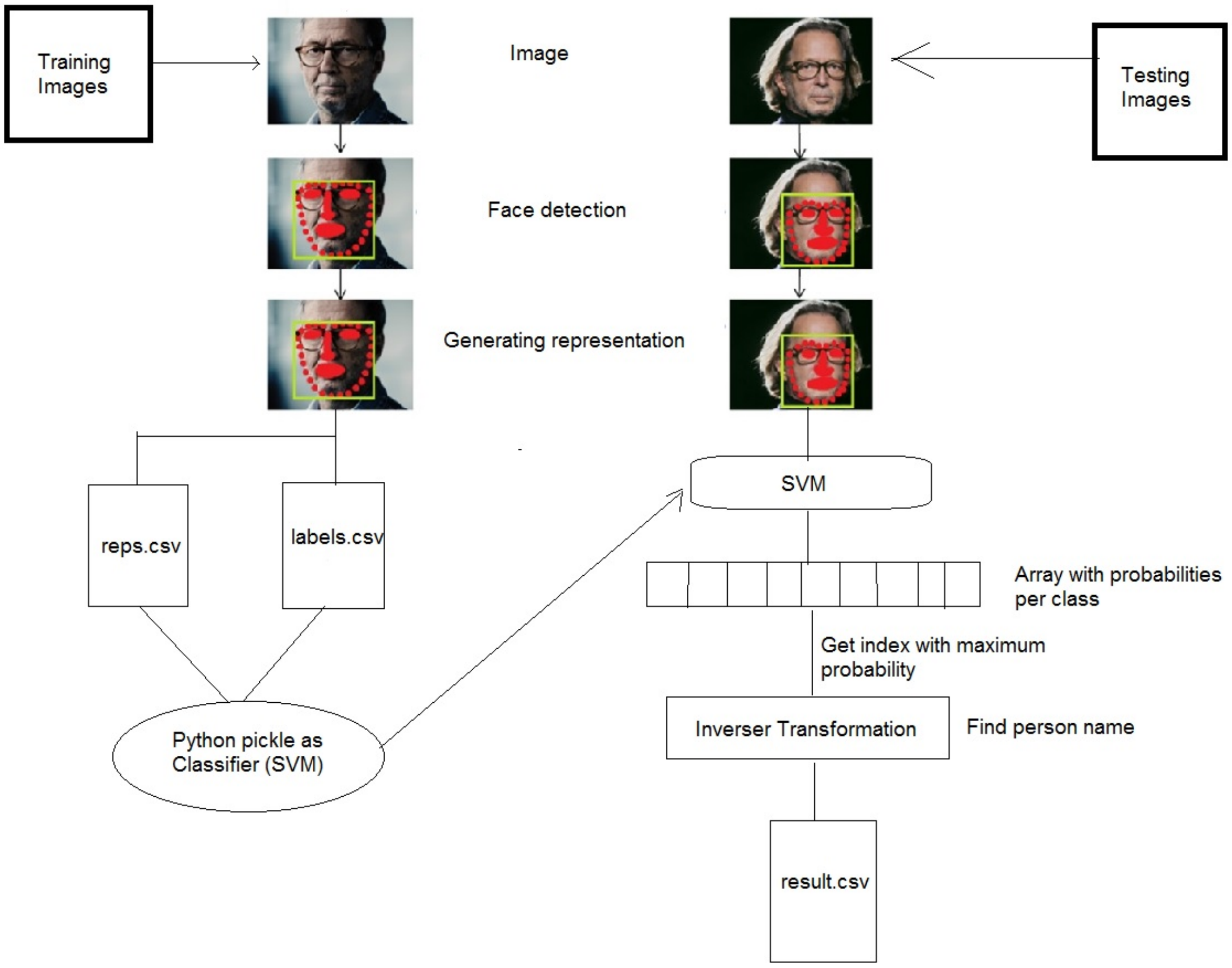

Training Images
Image
Face detection
Generating representation
Testing Images
reps.csv
labels.csv
Python pickle as Classifier (SVM)
SVM
Array with probabilities per class
Get index with maximum probability
Inverser Transformation
Find person name
result.csv

Figure 5.12: Smart Attendance: Blackbox WorkFlow

**Enrollment Job**

During enrollment, the images are taken and faces are detected in them. The detected faces are transformed into representations & these representations are stored into a csv file. Accordingly, a labels.csv file is generated in which a label is assigned to each representation. Using these reps.csv and labels.csv, a python pickle is generated which is basically an SVM classifier.

The algorithm for enrollment job is as follows:

**Algorithm 3** Recognition: Enrollment Job

```
Get the file containing enrollment images paths for a person.
Go to the feature directory and search for "reps.csv" & "labels.csv"
if reps.csv & labels.csv already exist then
    open & fetch the last label
else
    Create the reps.csv & labels.csv
end if
for file in list of files do
    Get the largest face from target image using dlib's face_landmarks.
    Transform and align the face extratcted.
    Get the representation by passing that face to forwardImage()
    Append the representation into reps.csv and new label into labels.csv
end for
Run GridSearchCV to create an SVM classifier and train it by fitting reps.csv and labels.csv as parameters
Dump the SVM as a python pickle and save it to the directory.
```

**Recognition Job**

In recognition, the testing images are taken one by one, and faces are detected into them. These faces are transformed to representation. The earlier created python pickle is loaded and the representation is passed to it. The pickle returns an array which is contaning probabilities against each class which was trained. We take the highest probability from it and using the highest probability's index, we get the person name by inverse transformation. At end, a result.csv file is generated which has the prediction and the confidence, which is nothing but the respective probability.

**pickle.load()**
This method is used from python package "pickle". The python pickle generated in enrollment process is used here to load.

- Read a string from the open file object file and interpret it as a pickle data stream, reconstructing and returning the original object hierarchy. This is equivalent to Unpickler(file).load().
- The file must have two methods, a read() method that takes an integer argument, and a readline() method that requires no arguments. Both methods should return a string.[9]

**sklearn.svm.libsvm.predict_proba()**
It computes probabilities for number of samples in X, for the all possible outcomes. The requirement is to have the model containing probability information which is computed at time of training: that means fit with the attribute probability is set to True.

- **Parameters:**

  - X : array-like, shape (n_samples, n_features)
  - For kernel="precomputed", the expected shape of X is [n_samples_test, n_samples_train]

- **Returns:**
  - X : array-like, shape (n_samples, n_features)
  - Returns the probability of the sample for each class in the model. The columns correspond to the classes in sorted order, as they appear in the attribute classes.

The algorithm for recognition job is as follows:

**Algorithm 4** Recognition: Recognition Job

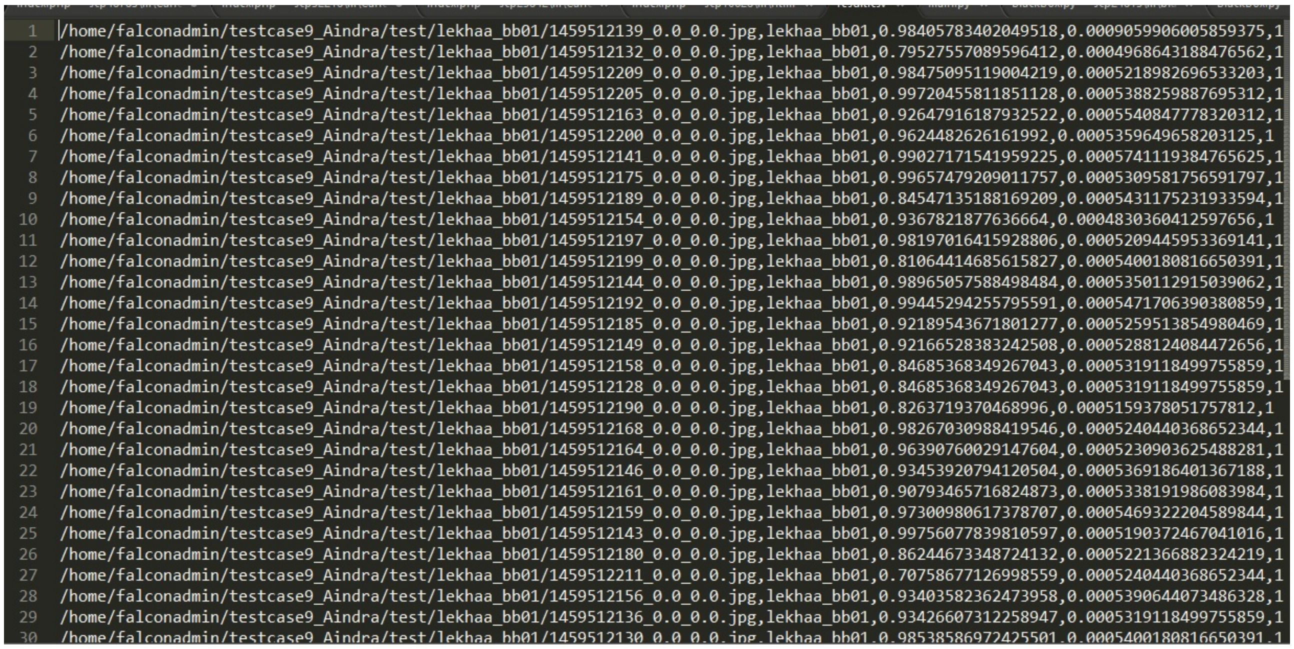
```
Load the SVM from python pickle in the feature directory.
Create & open result.csv in feature directory.
for file in list of files do
    Get the largest face from target image using dlib's face_landmarks.
    Transform and align the face extratcted.
    Get the representation by passing that face to forwardImage()
    Start time calculation.
    Pass the representation to already trained SVM's predict_proba() method.
    Get the array of prediction against each class with which the SVM is trained.
    Find the maximum prediction out of the array and get the index.
    Using the Inverse Transform function, by passing the index to it.
    Get the corresponding label or person's name.
    Get the prediction value from prediction array at the acquired index.
    This is the confidence for prediction
    Calculate current time & subtract start time from it to get the time duration in prediction.
    Append source image path, predicted person name, confidence & time duration into result.csv
end for
```

## 5.5 Results

### 5.5.1 Experiments

Many experients have been carried out using different nn models and varying data to check the accuracy of created blackbox. The blackbox gives predictions for each person with some confidence. In our experiments, we found that nn4.v1.t7 gives better accuracy on our dataset compared to nn4.small2.v1.t7 model.

For each test a nn model is used, number of persons, number of training imagesequal for each person, number of testing images & result file is acquired.
The result.csv generated contains items in following order: source path of test image, prediction, confidence, time taken in seconds, correct or incorrect(1 or 0)



```
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512139_0.0_0.0.jpg,lekhaa_bb01,0.98405783402049518,0.0009059906005859375,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512132_0.0_0.0.jpg,lekhaa_bb01,0.79527557089596412,0.0004968643188476562,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512209_0.0_0.0.jpg,lekhaa_bb01,0.98475095119004219,0.0005218982696533203,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512205_0.0_0.0.jpg,lekhaa_bb01,0.99720455811851128,0.0005388259887695312,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512163_0.0_0.0.jpg,lekhaa_bb01,0.92647916187932522,0.0005540847778320312,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512200_0.0_0.0.jpg,lekhaa_bb01,0.962448262616199 2,0.000535964965820312 5,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512141_0.0_0.0.jpg,lekhaa_bb01,0.99027171541959225,0.0005741119384765625,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512175_0.0_0.0.jpg,lekhaa_bb01,0.99657479209011757,0.0005309581756591797,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512189_0.0_0.0.jpg,lekhaa_bb01,0.84547135188169209,0.0005431175231933594,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512154_0.0_0.0.jpg,lekhaa_bb01,0.9367821877636664,0.0004830360412597656,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512197_0.0_0.0.jpg,lekhaa_bb01,0.98197016415928806,0.0005209445953369141,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512199_0.0_0.0.jpg,lekhaa_bb01,0.81064414685615827,0.0005400180816650391,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512144_0.0_0.0.jpg,lekhaa_bb01,0.98965057588498484,0.0005350112915039062,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512192_0.0_0.0.jpg,lekhaa_bb01,0.99445294255795591,0.0005471706390380859,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512185_0.0_0.0.jpg,lekhaa_bb01,0.92189543671801277,0.0005259513854980469,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512149_0.0_0.0.jpg,lekhaa_bb01,0.92166528383242508,0.0005288124084472656,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512158_0.0_0.0.jpg,lekhaa_bb01,0.84685368349267043,0.0005319118499755859,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512128_0.0_0.0.jpg,lekhaa_bb01,0.84685368349267043,0.0005319118499755859,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512190_0.0_0.0.jpg,lekhaa_bb01,0.8263719370468996,0.0005159378051757812,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512168_0.0_0.0.jpg,lekhaa_bb01,0.98267030988419546,0.0005240440368652344,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512164_0.0_0.0.jpg,lekhaa_bb01,0.96390760029147604,0.0005230903625488281,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512146_0.0_0.0.jpg,lekhaa_bb01,0.93453920794120504,0.0005369186401367188,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512161_0.0_0.0.jpg,lekhaa_bb01,0.90793465716824873,0.0005338191986083984,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512159_0.0_0.0.jpg,lekhaa_bb01,0.97300980617378707,0.0005469322204589844,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512143_0.0_0.0.jpg,lekhaa_bb01,0.99756077839810597,0.0005190372467041016,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512180_0.0_0.0.jpg,lekhaa_bb01,0.86244673348724132,0.0005221366882324219,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512211_0.0_0.0.jpg,lekhaa_bb01,0.70758677126998559,0.0005240440368652344,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512156_0.0_0.0.jpg,lekhaa_bb01,0.93403582362473958,0.0005390644073486328,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512136_0.0_0.0.jpg,lekhaa_bb01,0.93426607312258947,0.0005319118499755859,1
/home/falconadmin/testcase9_Aindra/test/lekhaa_bb01/1459512130_0.0_0.0.jpg,lekhaa_bb01,0.98538586972425501,0.0005400180816650391,1
```

Figure 5.13: Result.csv Format

Following table gives results for experiments done with new blackbox with various data.

| No. of Experiment | nn Model | No. of persons | No. of Training Images | No. of Testing Images | Correct Predictions | Incorrect Predictions | Accuracy |
|---|---|---|---|---|---|---|---|
| 1 | nn4.v1.t7 | 5 | 150 | 5 | 4 (Confidence: 0.698879) | 1 (Confidence: 0.395678) | 90% |
| 2 | nn4.v1.t7 | 18 | 360 | 606 | 285 (Confidence: 0.759456) | 321 (Confidence: 0.652345) | 47% |
| 3 | nn4.v1.t7 | 2 | 200 | 100 | 93 (Confidence: 0.913458) | 9 (Confidence: 0.512385) | 93% |
| 4 | nn4.small2 .v1.t7 | 2 | 200 | 100 | 43 (Confidence: 0.567895) | 57 (Confidence: 0.578954) | 43% |
| 5 | nn4.v1.t7 | 2 | 200 | 100 | 99 (Confidence: 0.999879) | 1 (Confidence: 0.687458) | 99% |
| 6 | nn4.small2 .v1.t7 | 2 | 277 | 100 | 64 (Confidence: 0.785699) | 36 (Confidence: 0.712563) | 64% |
| 7 | nn4.v1.t7 | 3 | 300 | 120 | 104 (Confidence: 0.982456) | 16 (Confidence: 0.672536) | 86.66% |
| 8 | nn4.v1.t7 | 5 | 300 | 100 | 92 (Confidence: 0.975686) | 8 (Confidence: 0.612536) | 92% |

Table 5.3: Smart Attendance: New Blackbox Results Summary

### 5.5.2 Confusion Matrix

A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix.[10]
Following is the confusion matrix for test case 8:

**Test Case 8:**

- **Model Used:** nn4.v1.t7
- **No. of persons:**5
- **Training Images:** 5 X 60 each = 300
- **Testing Images:** 5 X 20 each = 100
- **Results:**
  - Correct Predictions: 92 (Average Confidence: 0.975686)
  - Incorrect Predictions: 8 (Average Confidence: 0.612536)
  - **Total Accuracy: 92%**

| | Person A | Person B | Person C | Person D | Person E |
|---|---|---|---|---|---|
| **Person A** | 9 | 0 | 0 | 0 | 0 |
| **Person B** | 0 | 10 | 0 | 0 | 0 |
| **Person C** | 0 | 0 | 7 | 0 | 0 |
| **Person D** | 0 | 0 | 0 | 8 | 0 |
| **Person E** | 0 | 0 | 0 | 0 | 8 |

Table 5.4: Confusion Matrix

# Chapter 6

# Third Party API Integeration With Web Interface

The term API stands for Application programming interface (API), & is basically defined as: The process which allows programmes to access, connect and more imporatantly, to communicate with other programmes. Their APIs are their languages.[11]

## 6.1 Overview

A third party API integration is a document which is provided by the vendors to using organization, this document contains all the instructions & technical details like input/output data formats, API url etc., for programmers to itegrate the software into their custom application. This API documentation supplied gives a chance to create a useful custom application according to the requirements and avoiding all unnecessary features.

Representational State Transfer (REST) & Simple Object Access Protocol (SOAP) are solutions to access web services. The choice between two may be difficult. Both of these are having the similarities over HTTP protocols. In comparison to REST, SOAP is a more rigid collection of patterns for messaging. Without following the rules in SOAP, the level of standardization can't be achieved. On the other hand, REST is naturally more flexible and as an architectural style, it does not need much processing. However, both of these are established on basis of rules which are used for exchanging the information. [12]

### 6.1.1 RESTful API

#### REST- Representational State Transfer

It is a simple method to send and receive the data in between client & server without many standards defined for such method. The data to be sent and received could be anything, XML, JSON or even plain text. Thus, in comparison to SOAP, it is considered "Light Weighted". Instead os using the cumbersome XML structures, it relies on a simple URL. The most of web services which use REST, rely completely on getting the required information using the approach of URL. REST has options to use types of t HTTP 1.1 verbs (GET, POST, PUT, DELETE) to do operations.

REST doesn't need to exclusively use XML to provide the response. REST based web services can give the output in various forms as, CSV (Comma Seperated Values), JSON (Javascript Object Notion) & RSS (Really Simple syndication). Hence, basically, we can get the output in a form which is easy to parse in the language used for the application. [12] As an example to create with REST, we can create a URL for some service. The page for API documentation shows a URL. We can use the browser to interact with the web service, hence making it very easy to make the correct URL & verify the output which is needed to be parsed with the application.

### 6.1.2 SOAP API

**SOAP- Simple Object Access Protocol**

It's a way to send and receive data or small amount of information, over the internet. In this, the data is formatted into XML, & generally are sent using HTTP(Hypertext Transfer Protocol). This is made so, to be responsible for encoding the XML information such that, it is collected by an operating system and is understood over any type of networking protocol.

To provide services for messaging, SOAP relies fully on XML. This XMl which is used for making requests and in turn receive responses, can be extremely comples. Sometimes, the requests have to be built manually. That can be tough as SOAP is intolerant of errors. The other languages, however can use the easier ways, which are provided by SOAP; which can reduce the efforts needed to make the request & to parse the received response. [12] The main point says that, SOAP is very highly extensible, but we only use the parts which are required for a particular task. E.g., while using a web service which is public, and freely available, there is no need for WS-Security.

The most important SOAP feature is the built-in error handling. If any problem is there with the request, the received response will bring the error information which can be used to solve the problem. Such error reports also give the standardized codes to automate the error handling processes and tasks in the code. One more interesting feature is that, we dont́ compulsarily need tp use it with HTTP transport. There is a specification for using SOAP over SMTP(Simple mail Transfer Protocol).

Extremely few web service support both REST and SOAP.

In **Smart Attendance** System, we are using RESTful API, as we need the data to be sent and received in various forms & cant rely exclusively on XML. As we are using JSON format for input and output in mobile app and XML format for input and output in web interface. Hence, to make it easier to create the client and use suaitable inout/output formats, we selcted RESTful APIs.

| SOAP | REST |
|---|---|
| Independent of platform, language & transport. | Rest requires use of HTTP. |
| In distributed enterprise environment, it performs well. | REST makes assumption of direct point to point communication. |
| Standardized | Not fully standardized. |
| It has extensibility which is prebuilt and significant by WS standards. | It doesn't have such lists of standards. |
| Built in error handling. | No robust built in error handling. |
| Tools needed to interact with web services. | No such expensive tools required. |
| Uses XML for all messages. | Capable of using smaller message formats. |
| Slower, some enstensive processing required. | Faster, no such processing needed. |

Table 6.1: Activities on Client

## 6.2 Rest Client Creation

To use these APIs into web interface & make request & receive response from them, we created rest client using PHP and JavaScript in the web pages. These client access the web services, receive output XML, parse it & use the data to show in the particular format on web interface.

**PHP REST Client**

The PHP RESTClient looks like:

Listing 6.1: PHP Rest Client

```
$post_xml = (string) $_POST['xml'];

$url = $apihost."updateorg";

$ch = curl_init();      // initialize curl handle

curl_setopt($ch, CURLOPT_VERBOSE, 1);

curl_setopt($ch, CURLOPT_URL, $url); // set url to post to

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

curl_setopt($ch, CURLOPT_USERPWD, "$username:$password");
```

```
        curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);

        curl_setopt($ch, CURLOPT_HTTPHEADER, Array("Content-Type: application

        curl_setopt($ch, CURLOPT_HEADER, 1);

        curl_setopt($ch, CURLOPT_TIMEOUT, 15);

        curl_setopt($ch, CURLOPT_POST, 1);

        curl_setopt($ch, CURLOPT_POSTFIELDS, $post_xml);

        $data = curl_exec($ch);

        $header_size = curl_getinfo($ch, CURLINFO_HEADER_SIZE);

        $header = substr($data, 0, $header_size);

        $body = substr($data, $header_size);

        curl_close($ch);

        $doc = new DOMDocument();
        $doc->loadXML($body);
```

**JavaScript REST Client**

The REST Client in JavaScript is made using Ajax call. The e.g. in JavaScript is as follows:

Listing 6.2: JAVA Rest Client

```
$.ajax({
        url: 'curate_getabsentpool.php',
        async: false,
        type: 'POST',
        data: {xml:str},
        success: function(data) {
        console.log(data);
        var send = data;
        parser1 = new DOMParser();
        xmlDoc1 = parser.parseFromString(send,"text/xml");
        xmlLoader = new DOMParser();
        var x1 = xmlDoc1.documentElement;
        }
});
```

## 6.3 Process flow

### 6.3.1 API Calling Flow

API calling process flow includes follwoing steps:

- Client Send the Request to call API with given URL and input xml.
- Jersey Servlet Receive API Request URL.
- Servlet URL Mapping.
- After that HttpGofer Class receives input xml and send output xml.
- API gofer function call.
- Main API will call.
- API Response goes to API gofer function.
- API gofer function make xml of API response.
- API gofer response goes to HttpGofer class.
- Final Response send to Client side(API Calling side) in form of xml.

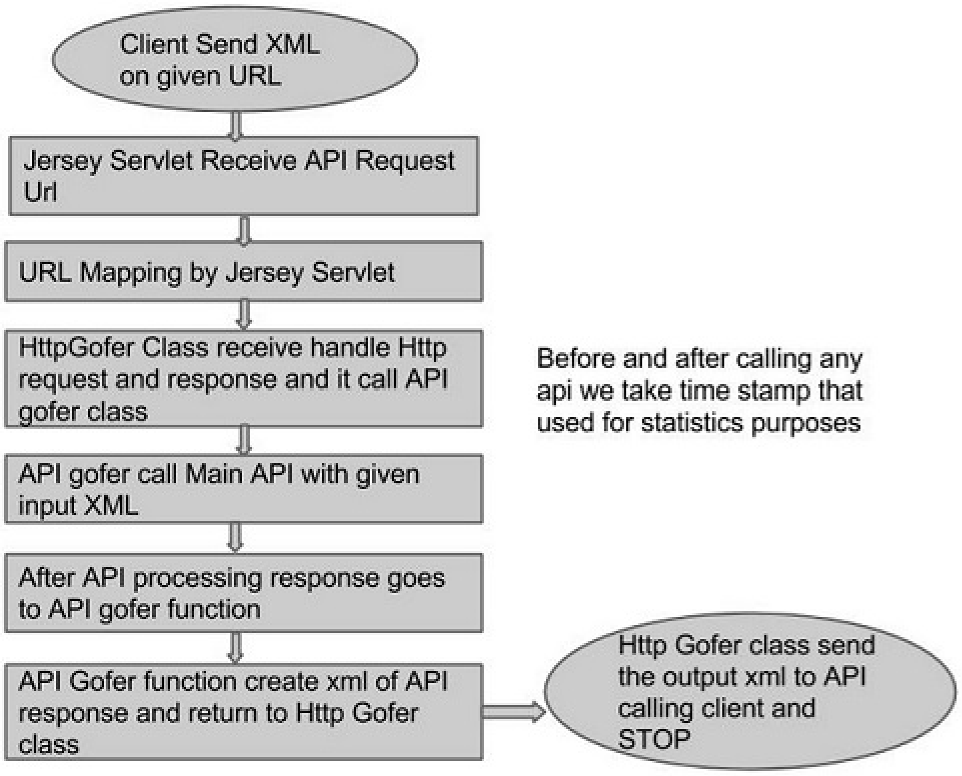

Client Send XML on given URL
Jersey Servlet Receive API Request Url
URL Mapping by Jersey Servlet
HttpGofer Class receive handle Http request and response and it call API gofer class
API gofer call Main API with given input XML
After API processing response goes to API gofer function
API Gofer function create xml of API response and return to Http Gofer class
Http Gofer class send the output xml to API calling client and STOP
Before and after calling any api we take time stamp that used for statistics purposes

Figure 6.1: API Calling Process Flow

## 6.4 Hibernate

### 6.4.1 Hibernate Configuration

Hibernate used for Object/Relationship mapping. Hibernate ORM enables developers to more easily write applications whose data outlives the application process. As an Object/Relational Mapping (ORM) framework, Hibernate is concerned with data persistence as it applies to relational databases (via JDBC).

### 6.4.2 Hibernate Mapping Files

#### POJO Class

POJO classes are just Java classes and it used for mapping the Hibernate XML file with Database Tables. POJO class contains the all attributes that are available in Database tables. All POJO class contains Setter or getter methods only. All POJO class are available at src/..../java/entity package. For Every Table in the database that are used in our project we create a POJO class. This is one sample POJO class apiList.

Listing 6.3: POJO class example

```
public class apiList
{
        private Integer api_id;
        private String api_name;

        public Integer getApi_id()
        {
                return api_id;
        }

        public void setApi_id(Integer api_id)
        {
                this.api_id = api_id;
        }

        public Integer getApi_name()
        {
                return api_name;
        }

        public void setApi_name(Integer api_name)
        {
                this.api_name = api_name;
        }
}
```

- The mapping document is an XML document having ¡hibernate-mapping¿ as the root element which contains all the ¡class¿ elements.
- The ¡class¿ elements are used to define specific mappings from a Java classes to the database tables.
- The Java class name is specified using the name attribute of the class element and the database table name is specified using the table attribute.
- The ¡id¿ element maps the unique ID attribute in class to the primary key of the database table.
- The name attribute of the id element refers to the property in the class and the column attribute refers to the column in the database table.
- The type attribute holds the hibernate mapping type, this mapping types will convert from Java to SQL data type.
- The ¡generator¿ element within the id element is used to automatically generate the primary key values.
- The ¡property¿ element is used to map a Java class property to a column in the database table.
- The name attribute of the element refers to the property in the class and the column attribute refers to the column in the database table.
- The type attribute holds the hibernate mapping type, this mapping types will convert from Java to SQL data type.

**XML file**

This is the Hibernate Mapping XML file that maps POJO class to Database.

Listing 6.4: Hibernate Mapping XML

```
<?xml version = "1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "=//Hibernate/Hibernate␣Mapping␣DtD␣3.0//E
"http://hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
        <class name = "com.aindra.java.entity.apiList" table="api_list">
                <id name = "api_id" type = "int">
                        <column name = "api_id" />
                        <generator class = "increment" />
                </id>
                <property name = "api_name" type = "java.lang.String">
                        <column name = "api_name"></column>
                </property>
        </class>
</hibernate-mapping>
```

## 6.5 Dynamicity of API Call

Since, the new architecture supports unlimited hierarchy on both dimensions, the APIs are extremely dynamic in nature. The input & output XML structures, tags, attributes are undefined and are created on the fly. Thus, the web interface is also built based on the data fecthed from parsed output XML. The API itself is not dynamic. This dynamicity comes through The actual behaviou represented by the system. What makes a Dynamic API dynamic is expanding the scope of its contract to define an interaction-model between the systems that is designed to cope with change. [13]

### 6.5.1 Featuring APIs

The APIs created as a part of this project work are for basically made for curation process. (Manual tagging or Untagging) Hence, the APIs for curation process are discussed as below:

#### Get Present Pool

This API is called to get the present pool images for particular days attendance. By present pool here, we mean by those images in which face are detected by the BlackBox & recognized with someone. Thus, to approved the matches done by the blackbox, we need to get all those images and persons with whom they are marked.

This is called with an input xml, which must contain the date for which present pool matches are required. This API gives user details, & the images details which are maked with him. Also, it gives rectangle parameters of the detect face on image which is marked with the respective user.

#### Curate Present Pool

This API is called to curate the present pool images for particular days attendance. By curating present pool here, we mean by approving the matches in those images in which face are detected by the BlackBox & recognized with someone. Thus, to approved the matches done by the black box, we need to get all those images and persons with whom they are marked & call this API to approve the matches.

This is called with an input xml, which must contain the date, session number, status IDs & hierarchy for which present pool matches are approved. This API status id of matches which are marked with the image. Also, it gives ACKs based on successful or failure response.

#### Get Absent Pool

This API is called to get the absent pool images for particular days attendance. By absent pool here, we mean by those images in which face are detected by the BlackBox & not recognized with anyone or those images in which face are not detected by the BlackBox. Thus, to create the rectangles manually and mark it with someone or to mark already detected rectangles with someone, we need to get all those images and rectangles if detected any.

This is called with an input xml, which must contain the date & hierarchy for which absent pool matches are required. This API gives absent user details, & the images details. Also, it gives rectangle parameters of the detect face on image if detected any.

#### Curate Absent Pool

This API is called to curate the absent pool images for particular days attendance. By curating absent pool here, we mean by changing the curated status of all the unmarked rectangles to YES or else you might want to use the same to tag it with the list absent users corresponding to the absent pool. This is called with an input xml, which must contain the date, session number & hierarchy for which absent pool is requested. Also, it gives ACKs based on successful or failure response.

#### Curate Create Rectangle

This API is called to create a rectangle on an image. The recangle coordiantes are stored and sent to API. This API needs data provided by Get Absent Pool API. This is called with an input xml, which must contain the hierarchy where the respective image is. This API gives success or failure ACK in response.

#### Curate Mark Rectangle

This API is called to mark a rectangle with a person. The recangle could be a manually made rectangle or a rectangle detected by BlackBox. This API needs data provided by Get Absent Pool API. This is called with an input xml, which must contain the hierarchy where the respective person and images are. This API gives success or failure ACK in response.

#### Curate Unmark Rectangle

This API is called to unmark a rectangle with a person. The recangle could be a manually made rectangle or a rectangle detected by BlackBox. This API needs data provided by Get Attendance API. This is called with an input xml, which must contain the hierarchy where the respective person and images are. This API gives success or failure ACK in response.

#### Get Reviews

This API is called to get reviews of the manual tagging done by the teacher or the manager. The to be reviewed tags are stored in a table and populated on web interface. The API send all the reviews with image and rectangle coordinates for the selected hierarchy and the session.

#### Mark Reviews

This API is called to mark reviews of the manual tagging done by the teacher or the manager. The to be reviewed tags are stored in a table and populated on web interface. The API recieves all the reviews with image and rectangle coordinates for the selected hierarchy and the session, with mark accpet or reject.

## 6.6 Dynamic User Interface

### 6.6.1 Dependencies

The Web interface for Smart Attendance is made using HTML, CSS, JAVSCRIPT, PHP. The PHP pages serve as rest client to call the Java APIs to get and send the data. The XML is created on UI based on user inputs and is sent to API. API in turn retrieve data from database and send back in XML format to client. This XML is parsed and dynamically web interface is poplulated based on the type of data. The web interface is highly dynamic in nature.

### 6.6.2 Design

#### Authentication

User has to login in the system to view attendance and do other operations.

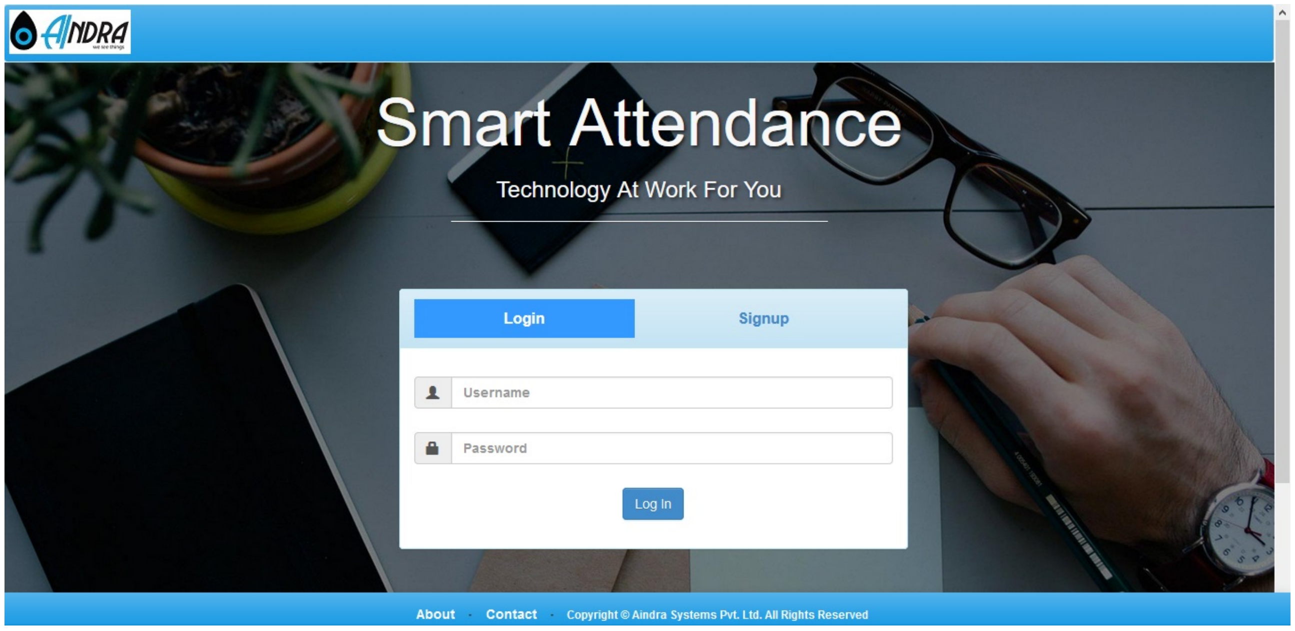
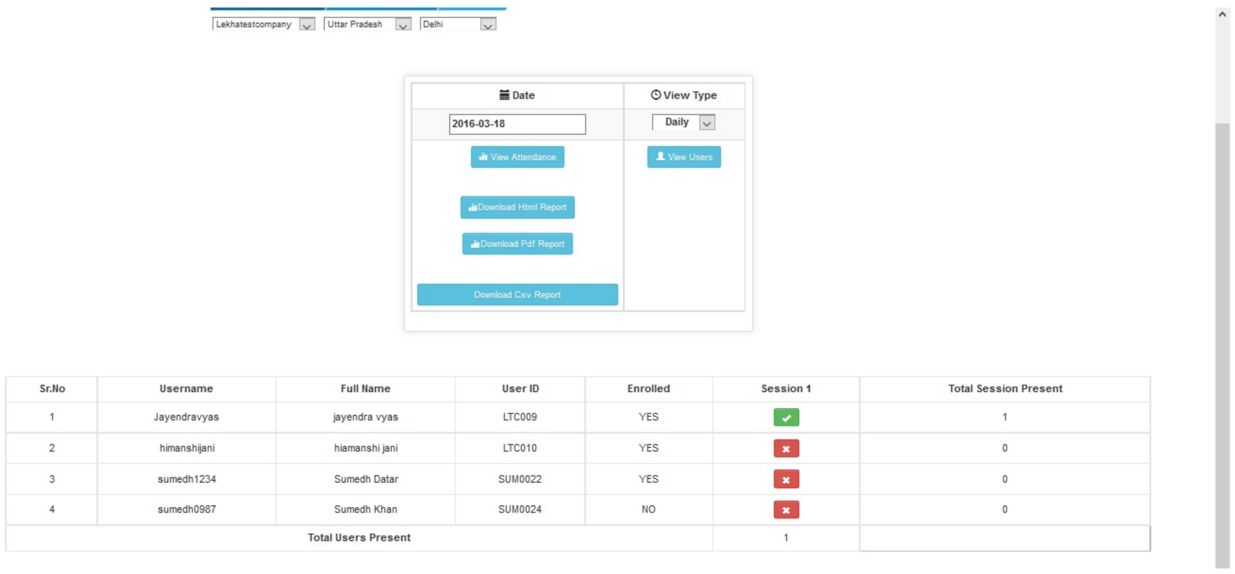

AINDRA
Smart Attendance
Technology At Work For You
Login
Signup
Username
Password
Log In
About
Contact
Copyright © Aindra Systems Pvt. Ltd. All Rights Reserved

Figure 6.2: Smart Attendance: Login

#### Attendance

To view the attendance in daily, monthly and weekly format, attemdance page is created. User at any level in hierarchy of users, can see it. On click of any green or red mark, user's present marked image or pool of absent image resepectively, can be seen.



Lekhatestcompany | Uttar Pradesh | Delhi

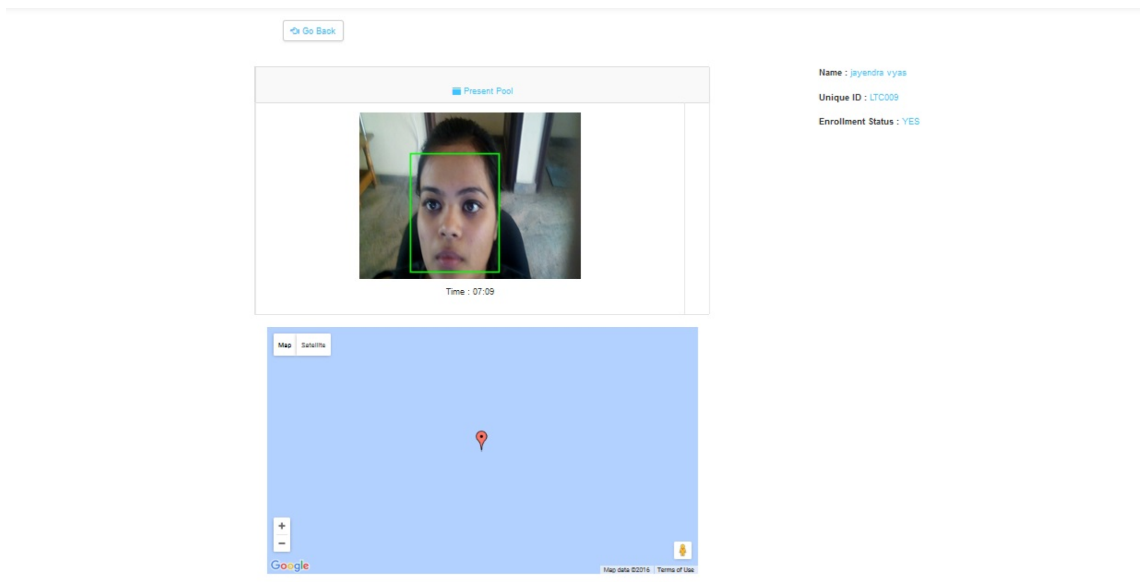
Date: 2016-03-18

View Type: Daily

| Sr.No | Username | Full Name | User ID | Enrolled | Session 1 | Total Session Present |
|---|---|---|---|---|---|---|
| 1 | Jayendravyas | jayendra vyas | LTC009 | YES | ✓ | 1 |
| 2 | himanshijani | hiamanshi jani | LTC010 | YES | ✗ | 0 |
| 3 | sumedh1234 | Sumedh Datar | SUM0022 | YES | ✗ | 0 |
| 4 | sumedh0987 | Sumedh Khan | SUM0024 | NO | ✗ | 0 |
| Total Users Present | | | | | 1 | |

Figure 6.3: Smart Attendance: Attendance

- **Present Marked Image**
  The present marked images of that user in that session are shown. The time stamp is also shown with the images. Also, if images were sent from mobile application, the location with map (lattitude and longitude) is shown.



Time : 07:09

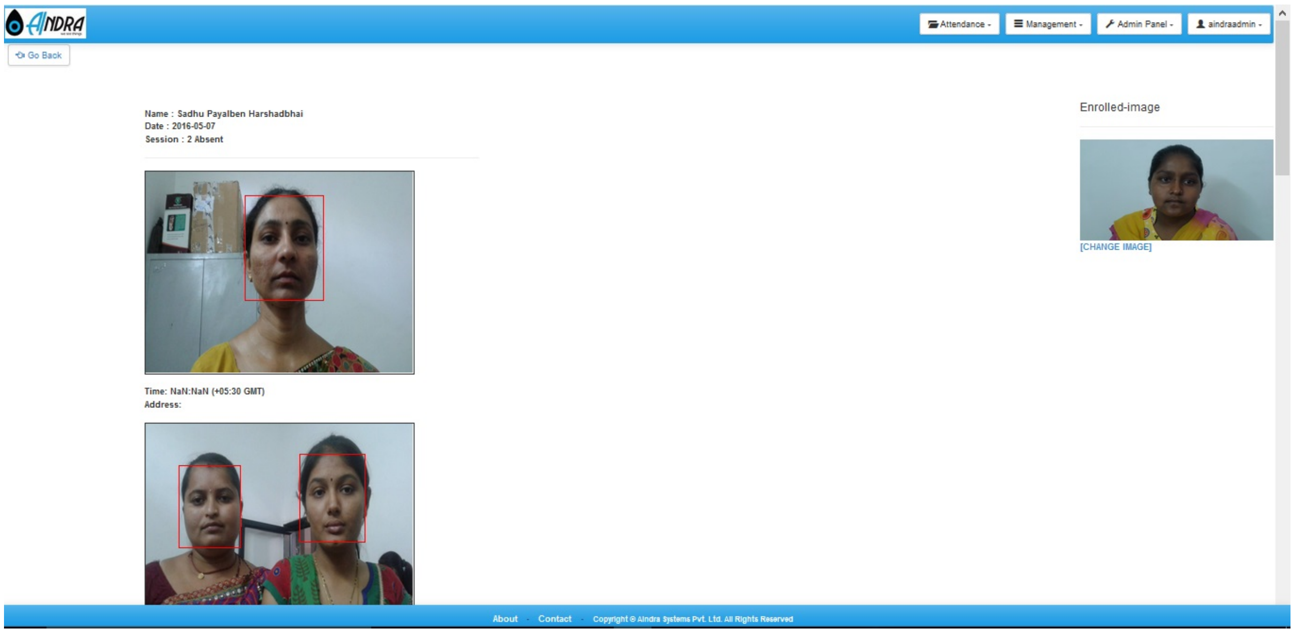
Name : jayendra vyas

Unique ID : LTC009

Enrollment Status : YES

Figure 6.4: Smart Attendance: Present Marked Image

- **Pool of Absent Images**
  If user is absent, then click on red mark will lead to pool of absent images which containes all the images of that day and that session in that branch of organization.



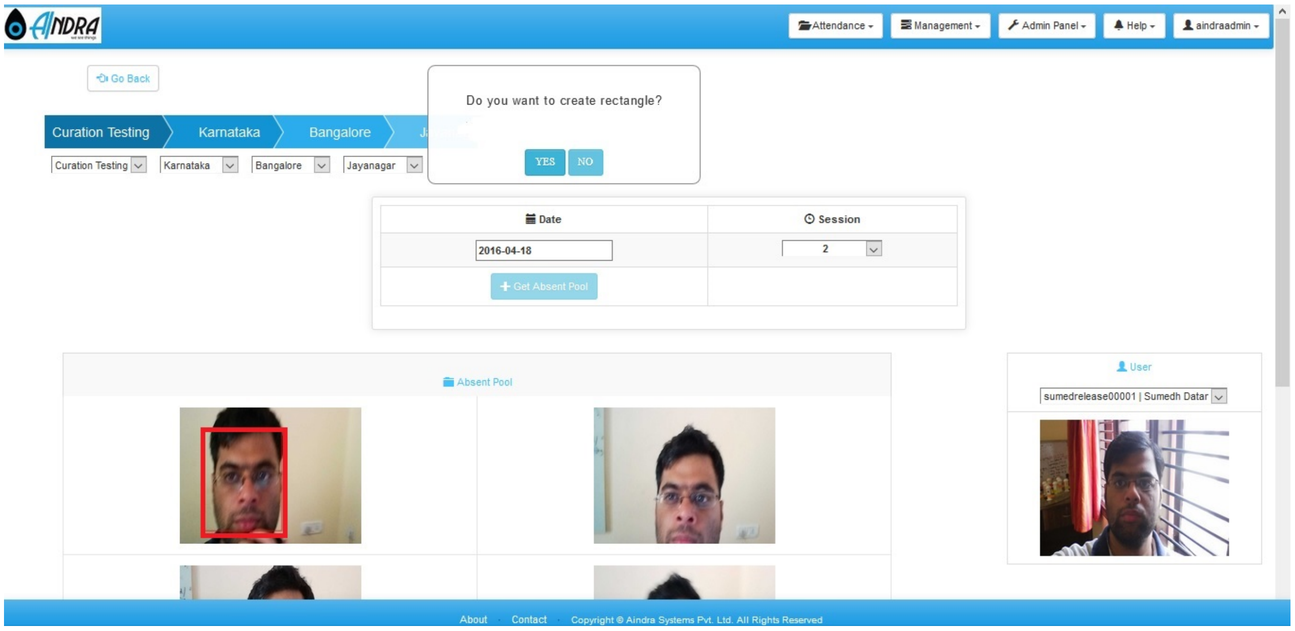
Name : Sadhu Payalben Harshadbhai
Date : 2016-05-07
Session : 2 Absent

Time: NaN:NaN (+05:30 GMT)
Address:

Figure 6.5: Smart Attendance: Pool of Absent Images

### Manual Tagging

On cases, when a person's face is left undetected or unmarked with any user, the senior user or manager can generate a manual tagging request. In this process, a rectangle is created over the face of the person & marked with a user. This request goes to the admin at AIndra.



Do you want to create rectangle?

Curation Testing | Karnataka | Bangalore | Jayanagar

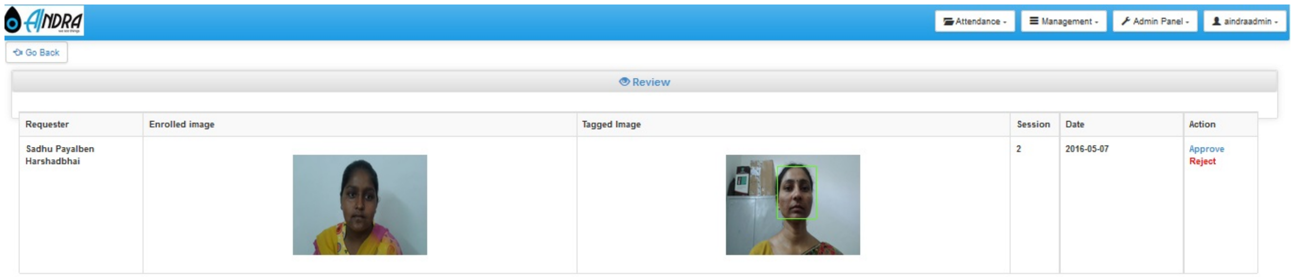
| Date | Session |
|---|---|
| 2016-04-18 | 2 |

sumedrelease00001 | Sumedh Datar

Figure 6.6: Smart Attendance: Manual Tagging

### Tagging Reviews

When a manager at any organization generates a tag request for a user, admin at AIndra has a right to accept or reject it.



| Requester | Enrolled image | Tagged Image | Session | Date | Action |
|---|---|---|---|---|---|
| Sadhu Payalben Harshadbhai | | | 2 | 2016-05-07 | Approve Reject |

Figure 6.7: Smart Attendance: Tagging Review

# Chapter 7

# Cancer Cells Annotation UI

The new product, **Cancer Detection** is going through the research phase right now. This product is aiming to tackle the cervical cancer's screening rates whihc are very low.

Typically, cervical cancer is unlike other cancers, not very agressive types, the gestation period is quitre long, which is of atleast 10-15 years, & still in records, because of cervical cancer, one woman dies in every seven and a half minutes in our country. One solution is to automate the porocess for gargantuan problem which can be done by AI. In screening, when a sample which is taken from a far flung area & then is sent to a tertiary oncology centre for the further process of screening, the expected results take long time span of atleast five to six weeks to be recieved.

Technically, the deviceis banked on the pap smear test. the pap smear test is a function of cervical screening by which we can detect cancerous cells which are potential in nature. To do this, the cells are smeared on a glass slide initially and the a chemical reagent is used to stain it (pap stains) and the slide is kept under a digital microscope. A board computing unit is there in setup, which is used to classify the sample whether they are normal or abnormal and then a pathologist receives it to cofirm the findings.

Pathologist can either accept or reject the view of the machine. This response is fed to the system again which does re-inforced learning and next time used the learning for better accuracy. The system is being built using artificial intelligence which will be trained by such cancerous cells images. The system will be fed resoponses everytime the pathologist accept or rejects a potential cancerous region. The initial data for the training of system is collected from hospitals. On the testing data, the pathologist response will be served as answer. Thus, this is an example of supervised learning. .

The whole process take very less time compared to the conventional tests. Hence, the product is expected to reduce the death rate of women by cervical cancer.

## 7.1 Requirement

For the pathologist, to see the cell images and annotate the nucleus, a GUI is created. When the cells are smeared on the glass slide and then are stained wirh a chemical reagent, they are put under microscope and the image is taken with camera. When this image is stored, an XML file is created for each image to collect data about the cells.

The fucnctional requirements in UI for the pathologist are:

- To view the marked regions.
- To view the annotated regions & view their nucleus names.
- To create regions on cell images.
- To annotate regions.
- To re-annotate regions.

## 7.2 Design

### 7.2.1 Dependencies

The system has been built using HTML, CSS, PHP, JAVASCRIPT. XML files are used to store data of cancer images, including the region coordinates, nucleus names etc. The system reads XML files and parses it to show the regions on the respective image on interface and show the information if it's annotated. After creating a new region on the image the inofrmation is added to XML files stored on the server.

### 7.2.2 Functionalities

1. **View the annotated and non-annotated regions on image:**
   When the user selects any category, he sees the images of four types:

   - Fully Annotated: Images where all the regions are annotated.
   - Partially Annotated: Images where some regions are annotated and rest are not annotated.
   - Non-annotated: Images where none of the regions are annotated.
   - Not Marked: Images where no regions are marked.

   After selecting an image, it shows all the regions, annotated with green boundaries and non-annotated with red boundaries, based on XML's coordinates data for regions. Canvas drawing is used to darw the regions. On click on any green region, a pop up shows the nucleus name for it.

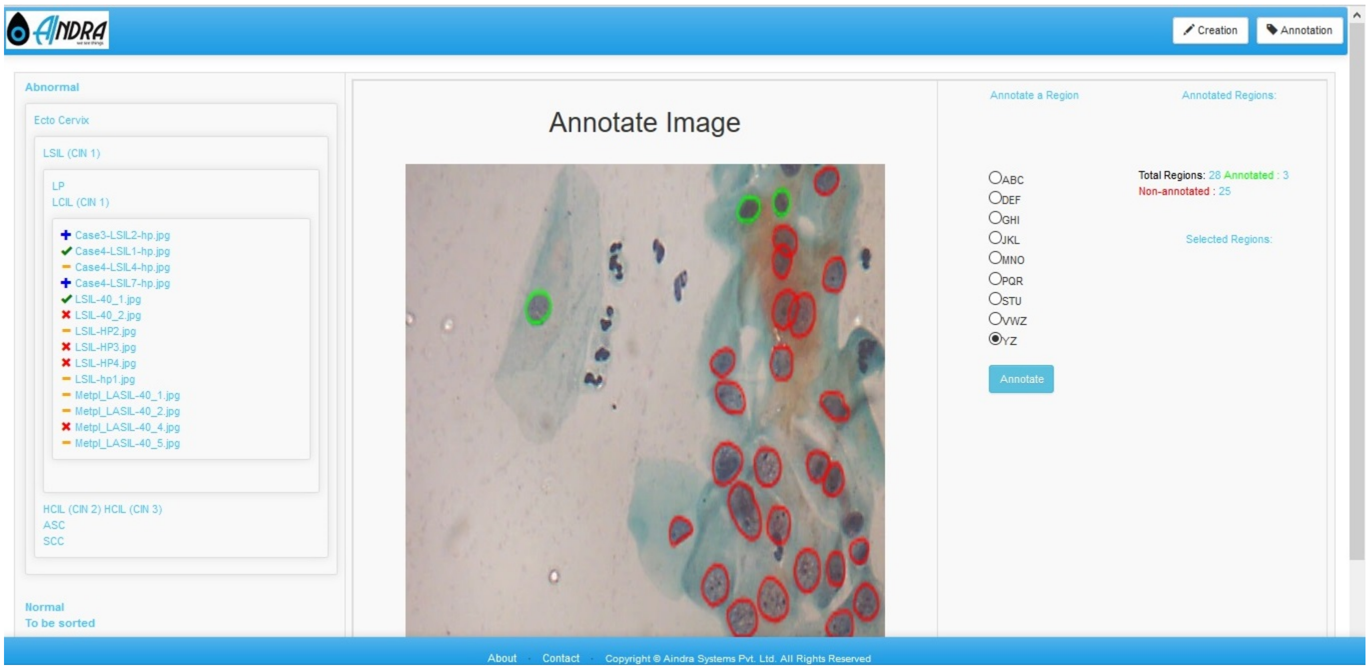


Figure 7.1: Cancer Annotation GUI: Viewing Regions

2. **Creation of new regions:**
   In this, the image will be used as a canvas background and no regions are drawn on it. The original image with all regions is shown on the right side for refernece. On click of any any annotated region there, will show the name.

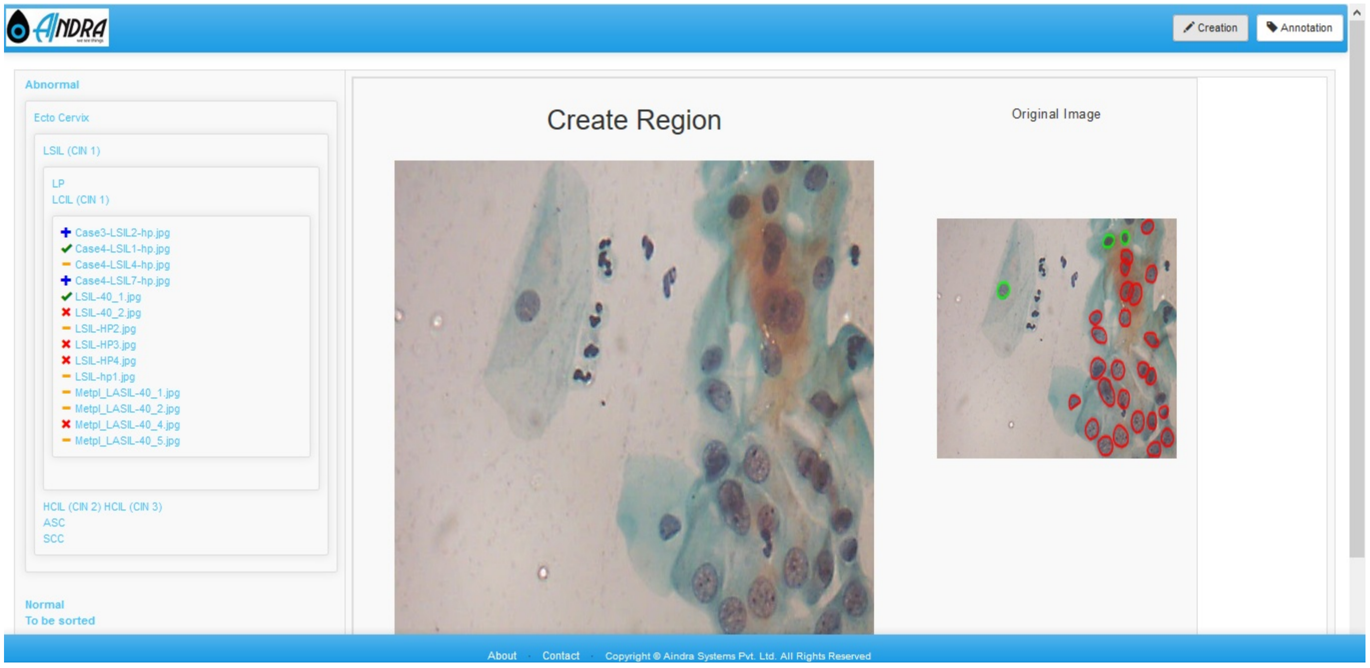


Figure 7.2: Cancer Annotation GUI: Canvas to Create Region

To create a region, the user has to make points around the shape. the points will keep connecting after each other, completing the boundary.

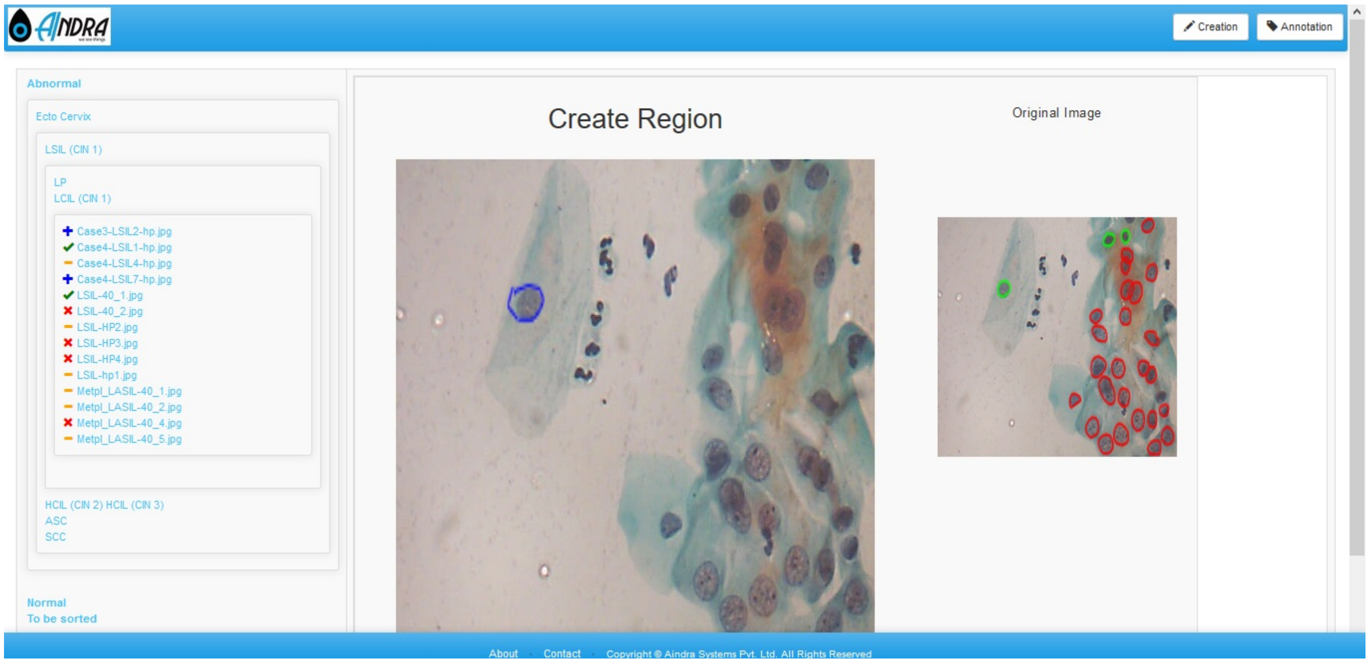


Figure 7.3: Cancer Annotation GUI: Drawing Boundary

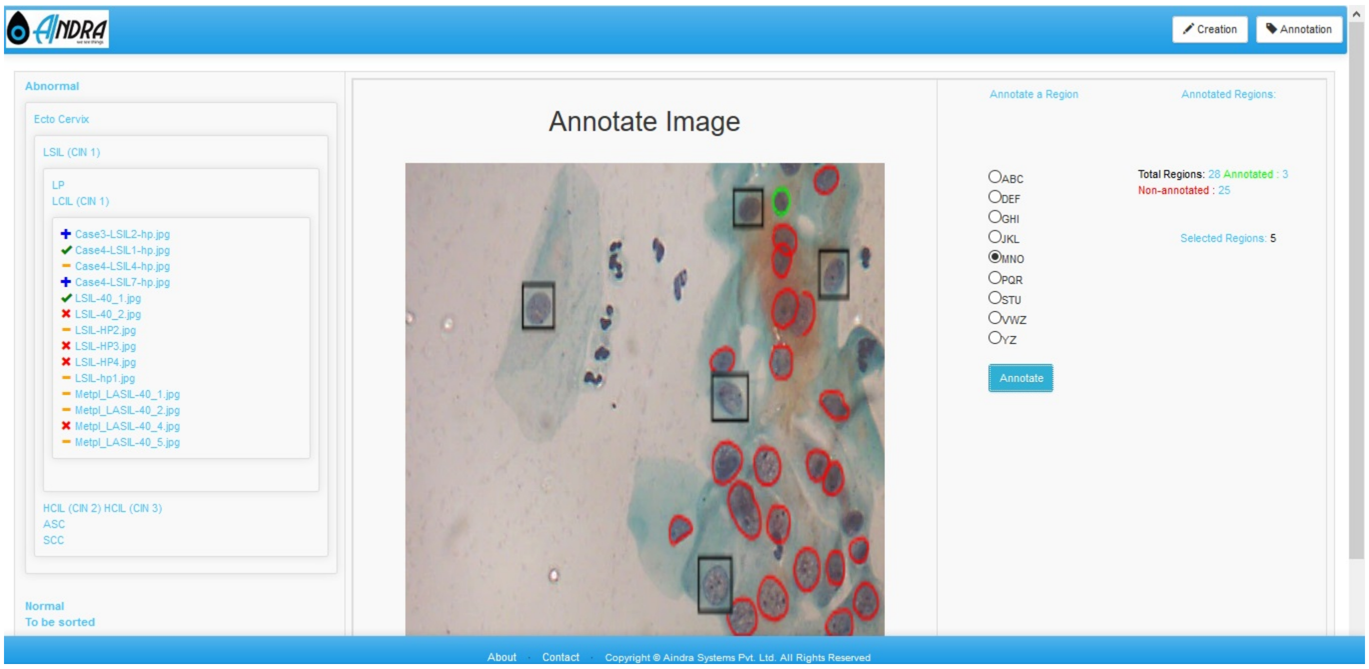
If any wrong point is made, there is option to clear the canvas.

Once the boundary is drawn, the user can create the region and all the points' x & y coordinates are stored in the XML corresponding to the image with a new region node. This region can be annotated by user in annotation functionality.

3. **Annotation:**
For annotation, user can select any numeber of regions. Multiple region selection is provided, i.e., green and red both, can be selected. Everytime a region is selected, the regions will be shown selected with a black rectangle around them. The potential cancerouse regions inside them will be visible.

User can also deselect the regions. The black rectagles are removed and original green and red boundaries are visible on deselection.

Once selection is done, user can select any name from the list of given names and annotate them. The text value from the radio buttons list is taken sent to a php file to add it into the XML file of the image. Along with it, the array containing list of regions to be annotated is also sent. Hence, using this, non-annotated regions can be annotated and already annotated regions' annotations can be updated.



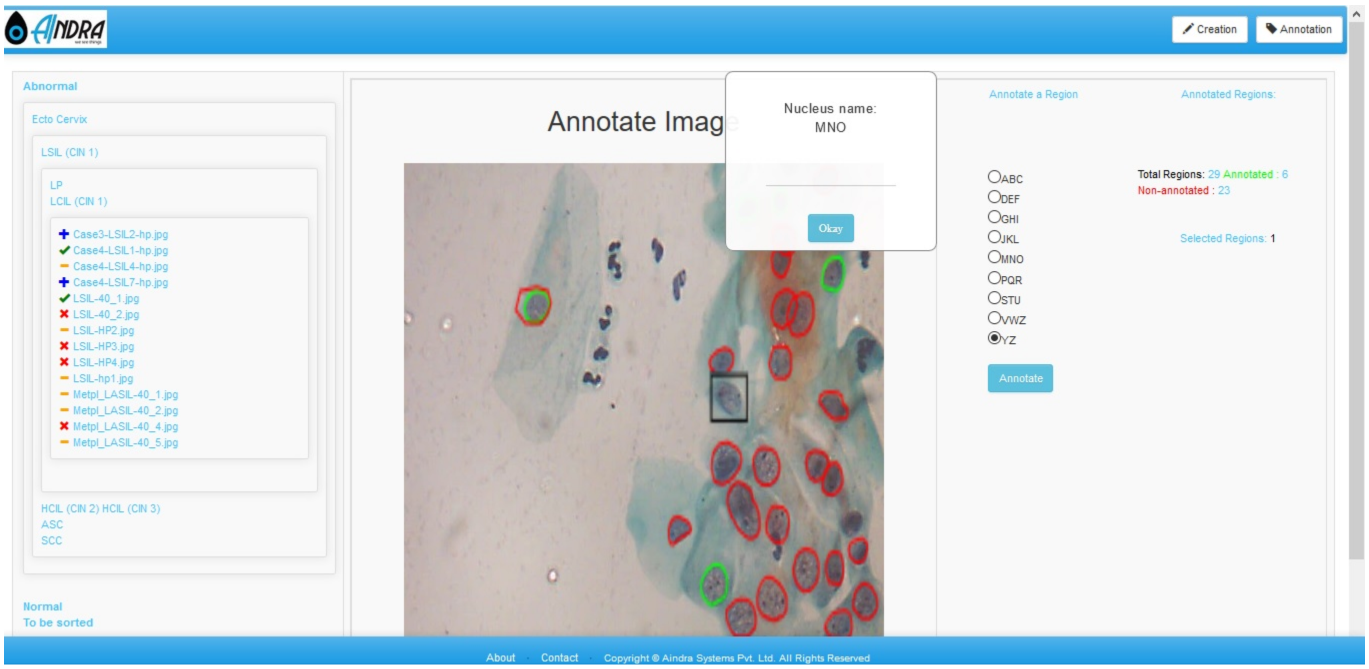
ABC DEF GHI JKL [✓ MNO] PQR STU VWZ YZ

Total Regions: 28 Annotated : 3
Non-annotated : 25

Selected Regions: 5

Figure 7.4: Cancer Annotation GUI: Region Selection

4. **Check Annotation:**
   To check the nucleus name for any annotated region, user can click on a any green region and a pop up will appear with the nucleus name for that region.



Nucleus name:
MNO

ABC DEF GHI JKL MNO PQR STU VWZ [✓ YZ]

Total Regions: 29 Annotated : 6
Non-annotated : 23

Selected Regions: 1

Figure 7.5: Cancer Annotation GUI: Annotation Check

# Chapter 8

# Conclusion & Future Work

## 8.1 Conclusion

Thus an intuitive Face Recognition System has been developed which makes use of the Face Recognition Techniques (FRT) to train (learn) itself and uses it to recognize people from the dictionary it created during learning process. Once Recognition is done, the result is forwarded to the web interface where managers and employees can view the consolidated attendance in various levels of granularity. When a person is wrongly marked absent, with the help of manual tagging, reinforced learning is also made possible so that the machine can recognize that person properly in the future. The new architecture is going to be implemented with the new black box, APIs & new web interface.

## 8.2 Future Work

The new architecture is going to be implemented with the new black box, APIs & new web interface. The illumination conditions and face postures are important factors impacting the results. The eciency can be increased by improving the core algorithm which takes care of the recognition process for a group of people. The future tasks include stablizing the core learning algorithm, APIs & add verify option in the system along with train & recognize.

## 8.3 Challanges

### 8.3.1 Smart Verification

- If more than one person is captured in the target iamge, to handle this condition.
- For relatively blurr/lower quality images.
- Since detection happens on eyes first, if in any case, eyes are closed or due to eye glass tint, not detected.
- For setting a benchmarked threshold for decision making.

- Use the results to make system learn for future purposes.

### 8.3.2 Smart Attendance

- If more than one person is captured in the target iamge, to handle this condition.
- For relatively blurr/lower quality images.
- Since detection happens on eyes first, if in any case, eyes are closed or due to eye glass tint, not detected.
- Use the results to make system learn for future purposes.

The work in future will also focus on the detailed understanding of cases of errors, moreover on improvement of the system, & reduction of the size of the model, as well to reduce the CPU requirements. The requirement is to look into the ways for improvement of at present longer time taken in training.

# Bibliography

[1] "Facial recognition system." https://en.wikipedia.org/wiki/Facial_recognition_system/.

[2] J. P. G. I. Florian Schroff, Dmitry Kalenichenko, "Facenet: A unified embedding for face recognition and clustering." IEEE Xplore, Computer Vision Foundation http://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf, 2015.

[3] X. W. Y. Sun and X. Tang, "Deeply learned face representations are sparse, selective and robust." CoRR, abs/1412.1265, 2014.

[4] "Openface, free and open source face recognition with deep neural networks." http://cmusatyalab.github.io/openface/.

[5] D. K. Florian Schroff and J. P. at Google, "Real-time face pose estimation." http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html, 2014.

[6] "Opencv." http://opencv.org/about.html.

[7] "Dlib, c++ library." http://dlib.net/, 2015.

[8] "Torch, a scientific computing framework for luajit." http://torch.ch/.

[9] P. S. Foundation, "pickle python object serialization." https://docs.python.org/2/library/pickle.html, 2016.

[10] "Confusion matrix." http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html.

[11] M. Welch, "What is api integration?." https://support.sparkpay.com/hc/en-us/articles/203285490-What-Is-API-Integration-, 2015.

[12] J. Mueller, "Understanding soap and rest basics and differences." http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/, 2013.

[13] D. Duggal, "Dynamic apis: Negotiating change." http://www.dataversity.net/dynamic-apis-negotiating-change/, 2015.