SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Facial Image Quality Estimation

SUBMITTED TOWARDS THE
FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING

(Computer Engineering)
BY

Aditya Deshpande	B120384236
Mrunmayi Deshpande	B120384237
Darshana Gadre	B120384241
Alisha Shahane	B120384325

Under The Guidance of

Prof. Dr. Siddhivinayak Kulkarni

DEPARTMENT OF COMPUTER ENGINEERING
MIT College of Engineering
Kothrud, Pune-410038

MIT College of Engineering
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Facial Image Quality Estimation

Submitted by

Aditya Deshpande	B120384236
Mrunmayi Deshpande	B120384237
Darshana Gadre	B120384241
Alisha Shahane	B120384325

is a bonafide work carried out by Students under the supervision of Prof. Dr. Siddhivinayak Kulkarni and it is submitted towards the fulfilment of the requirement of Bachelor of Engineering (Computer Engineering).

Prof. Dr. Siddhivinayak Kulkarni.
Internal Guide
Dept. of Computer Engg.

Prof Dr. Bharati Dixit
H.O.D
Dept. of Computer Engg.

Prof. Dr. Anil S Hiwale
Principal
MIT College of Engineering

Signature of Internal Examiner

Signature of External Examiner

PROJECT APPROVAL SHEET

Facial Image Quality Estimation

Is successfully completed by

Aditya Deshpande	B120384236
Mrunmayi Deshpande	B120384237
Darshana Gadre	B120384241
Alisha Shahane	B120384325

at

DEPARTMENT OF COMPUTER ENGINEERING
(MIT COLLEGE OF ENGINEERING)

SAVITRIBAI PHULE PUNE U NIVERSITY, PUNE
ACADEMIC YEAR 2017-2018

Prof. Dr. Siddhivinayak Kulkarni.
Internal Guide
Dept. of Computer Engg.

Prof Dr. Bharati Dixit
H.O.D
Dept. of Computer Engg.

Acknowledgments

*It gives us great pleasure in presenting the project report on **'FACIAL IMAGE QUALITY ESTIMATION'**. It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.*

*We would like to take this opportunity to thank our internal guide **Prof. Dr. Siddhivinayak Kulkarni** for giving us all the help and guidance we needed. We are grateful to him for his support. Our numerous discussions with him were extremely helpful and fruitful for us. His valuable suggestions and constant guidance were very invaluable during the course of this project.*

*We are also grateful to **Prof. Dr, Bharati Dixit**, Head of Computer Engineering Department, MITCOE and **Prof. Dr. Anil S Hiwale**, Principal, MIT College of Engineering, for making available all resources such as library, laboratory with all needed software platforms and creating a congenial environment to work in for completing our project.*

*We would also like to extend our sincere gratitude to **Dr. Bhushan Garware**, our company mentor. We would also like to thank our sponsor **Persistent Systems Ltd.** for providing us the necessary resources, guidance and thorough support throughout the course of this project.*

*Lastly, we would like to express our heartfelt thanks to Assistant Prof. **Sukhada Bhingarkar** and Assistant Prof. **Suresh Kapare** for their constant support and encouragement in the preparation of this report as our project co-ordinators.*

Aditya Deshpande
Mrunmayi Deshpande
Darshana Gadre
Alisha Shahane
(B.E. Computer Engg.)

Abstract

We propose a framework for facial image quality estimation in order to address the limitation of real-time applicability of facial recognition. This framework determines whether an image is suitable for facial recognition. We first exploit machine learning algorithms to map the relationship between image quality features and performance of facial recognition. We extract a variety of features (like focus measure, brightness, obscured face) and study their influence on the accuracy of face recognition. After examining the results of this approach, we then used deep learning to build a binary classifier which accepts or rejects images before sending them for actual facial recognition. This decision is taken based on the probability of the facial recognition framework correctly matching a face from the image. We used images from the Chokepoint dataset, and OpenFace- an open source facial recognition software, for building our framework.

FACIAL IMAGE QUALITY ESTIMATION

Contents

List of Figures

List of Tables

Chapter 1

Synopsis

1.1. Project Title:

Facial Image Quality Estimation

1.2. Project Option

External Project

1.3. Internal Guide

Prof. Dr. Siddhivinayak Kulkarni

1.4. External Guide

Dr. Bhushan Garware

1.5. Technical Keywords

Classification, Machine Learning, Deep Learning, Edge Computing

1.6. Problem Statement

To design a framework for matcher specific facial image quality estimation.

1.7. Abstract

This project proposes a framework for facial image quality estimation in order to address the limitation of real time applicability of facial recognition software. This framework determines whether an image is suitable for facial recognition. We first exploit machine learning algorithms to map the relationship between image quality features and performance of a facial recognition software. We extract a variety of features (like focus measure, brightness, obscured face) and study their influence on the accuracy of a facial recognition software. After examining the results of this approach, we then used deep learning to build a binary classifier which accepts or rejects images before sending them to the actual facial recognition system. This decision is taken based on the probability of the facial recognition system correctly matching a face from the image. We used images from the Chokepoint dataset, and OpenFace- an open source facial recognition software, for building our framework.

1.8. Goals and Objective

The goal of this project is to build a system which will pre-process the images before performing facial recognition. The system will act as predictor and estimate facial image quality. This estimation will be used to determine which images to send to the facial recognition software. Thus ensuring that it processes only those images for which it can definitely produce a match.

1.9. Relevant mathematical model associated with the Project

Let the system be S = { DD, NDD, i/p, o/p, f, S, F } where-
DD : Deterministic Data = { vector features }
features = { x1, x2, x3, … , xn }
NDD : Non-Deterministic Data = {ŷ }
i/p : input = { image }
o/p : output = { quality estimate }
f : functions = { extract_features(), extract_matchScore(), classification_model() }
S : Success = quality estimate is close to matcher's score
F : Failure = large difference between quality estimate and match score

1.10. Names of Conferences / Journals where papers can be published

IEEE 4th International Conference for Convergence in Technology (I2CT)

1.11. Review of Conference/Journal Papers supporting Project idea

Accepted without any changes

1.12. Plan of Project Execution

Please refer Annexure D for details.

Chapter 2

Technical Keywords

2.1. Area of Project

Machine Learning, Deep Learning

2.2. Technical Keywords

Classification, Machine Learning, Deep Learning, Edge Computing

Chapter 3

Introduction

3.1. Project Idea

A face recognition system is a computer application capable of identifying or verifying a person from a digital image or a video frame from a video source.
Each facial recognition software employs a different approach and faces its own set of challenges. Currently facial recognition softwares have high time complexity. The real time performance of such softwares is not up to the mark.
Recognizing faces in low light images, in foggy weather conditions, or images having obscured facial features are some of the issues encountered by facial recognition softwares.
The goal of this project is to build a system which will pre-process the images before performing facial recognition. The system will act as predictor and estimate facial image quality. This estimation will be used to determine which images to send to the facial recognition software. Thus ensuring that it processes only those images for which it can definitely produce a match.
Moreover, the performance of each facial recognition software might vary in case of different types of images. To improve the confidence level of facial recognition result, multiple such softwares can be integrated into an ensemble. The system we propose to build, will analyse an image and deliver the best match for it from all the softwares in the ensemble . Such a system implementing an ensemble as well as a pre-processing estimator will help improve the accuracy of facial recognition.
The proposed system will exploit machine learning techniques for the building the predictor.
Potential applications of this project include CCTV monitoring and real time analysis, criminal identification, automated attendance system, etc;

3.2. Motivation of the project

Currently facial recognition softwares are very slow because of their high time complexity. The performance of real time application of such softwares is not up to the mark. Moreover, a single classifier is unable to classify all sorts of faces. Some of the main obstacles that these classifiers face are images with low light in the background, blurred facial features, fog, different hairstyles of the same person. Our project aims to improve the applicability of existing facial recognition softwares by ensuring that the images sent as input to such software will always be matched successfully.

3.3. Literature Survey

One of the main issues for face recognition is processing the large amount of images generated from video data. This affects the feasibility of real-time face recognition. An approach to solve this issue is to evaluate a set of common face recognition methods and form a cascade of those methods [9]. The initial stages of the cascade process all the data, while the later computationally expensive stages of the cascade have less data to process. The mean absolute time difference between the given time limit and the actual time required is 16.9%.

The effectiveness of facial recognition is hampered if parameters like illumination, pose, occlusion and expression are modified. A significantly lower error rate of 0.3% [10] is reported on a high quality Visa dataset as compared to 18% [11] on the LFW dataset. Selecting only high-quality images from the acquired multiple images may improve overall performance but this is not a good practice when the dataset is dominated by low-quality images. Pruning all the low-quality images can cause considerable information loss. One solution involves face image quality assessment, in which multiple feature fusion and learning to rank can be used [12]. A quality assessment function can be defined wherein the goal is to assign the correct rank weights. Such a function is called Rank based Quality Score and can be used for improving and evaluating the system.

Another concern in real time face recognition is the discrepancy in ROC provided by vendors for an off-the-shelf face recognition system and the ROC obtained from real-time images. This makes them unreliable for practical application of verification systems. A proposed approach is a model that predicts the verification performance based on image quality [7]. The uncertainty whether a low match score indicates a non-match or poor image quality [13] makes similarity scores futile as a performance feature. Using a probability density function, the relationship between image quality and face recognition performance can be modelled. Since the model is based solely on image quality features, such a model allows performance prediction even before the actual recognition has taken place.

The authors of [14] apply Multi-Dimensional Scaling (MDS) to model the relationship between quality features and match score distribution while in [15], the authors use regression to model the relationship between quality partition (good, bad and ugly) and image quality features. Our work relates to the work of [16] which uses a Generalized Linear Mixed Model (GLMM) to model the relationship between image quality (like focus, head tilt, etc) and the False Non-Match Rate (FNMR) at a given False Match Rate (FMR). Their analysis focused on investigating the impact of each quality metric on recognition performance.

Chapter 4

Problem Definition and Scope

4.1 Problem Statement

4.1.1. Goals and objectives

The goal of this project is to build a system which will pre-process the images before performing facial recognition. The system will act as predictor and estimate facial image quality. This estimation will be used to determine which images to send to the facial recognition software. Thus ensuring that it processes only those images for which it can definitely produce a match.

4.1.2. Statement of scope

This project aims to create a framework that will be implemented before actual facial recognition takes place and try to predict the performance of that facial recognition by estimating image quality. The framework can be used in CCTV monitoring and real time analysis. It can also be used for criminal identification or detection of trespassers. The input image resolution, type might affect the performance of the framework.

4.2. Major Constraints

- Training the CNN is resource imperative as it requires GPUs.
- Since the OpenFace matchscore range is narrow, determining threshold for different classes is a challenging task.

4.3. Methodologies of Problem solving and efficiency issues

Our framework is a classification model. We implement the machine learning algorithms using the python scikit learn library. We use tensorflow to implement the model using deep learning. Such a model is built using keras. We have also used Nvidia Digits to train the model using transfer learning, with pretrained weights.

4.4. Outcome

- Matcher specific facial image quality estimator
- Reduced communication overhead
- Reduced latency as a result of maximum processing at edge

4.5. Applications

- CCTV Monitoring and Real Time Analysis
- Criminal Identification / Detecting Trespasser
- Automated Attendance System
- E-commerce Payments
- Prevent Underage Drinking
- Social Media Tagging
- Smart Phones

4.6. Hardware and Software Resources Required

- Face Recognition Software: OpenFace
- Languages: Python
- Libraries: OpenCV, numpy, scipy, scikit, pandas
- Frameworks: Viola-Jones, Amazon Rekognition
- Platform: Ubuntu 16.04, Amazon EC2
- Tools: Jupyter Notebooks

Chapter 5

Project Plan

5.1. Project Estimates

The proposed project model comes under the Waterfall Model category. It undergoes several phases starting from Requirement Analysis till its Deployment and Maintenance. We mapped our estimates with the steps in the waterfall model. Below are some estimates-

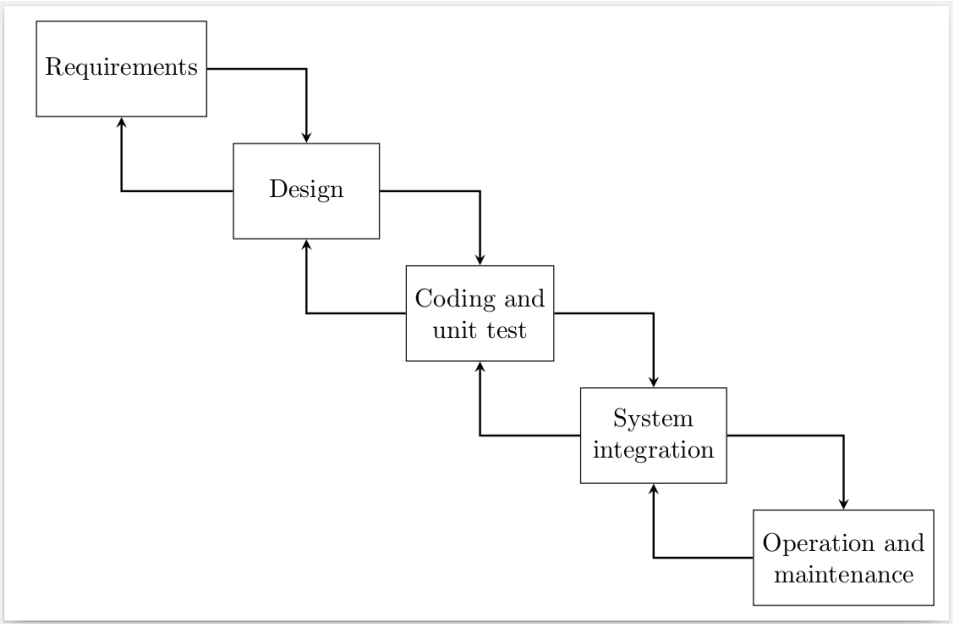


Fig 5.1 Waterfall Model

5.1.1. Reconciled Estimates

5.1.1.1. Cost Estimate

Phase	Cost	Estimate
Phase I	Documentation, Printing Report,	Rs. 500
Phase II	AWS Instance Costs, NVIDIA Digits GPU charges	Rs. 500

Table 5.1 Cost Estimation

Above table gives detailed information about the cost per phase as described in annexure B.

5.1.1.2. Time Estimate

The total detailed time estimation is given in the Annexure B and the figure below depicts the total time required along with the date.

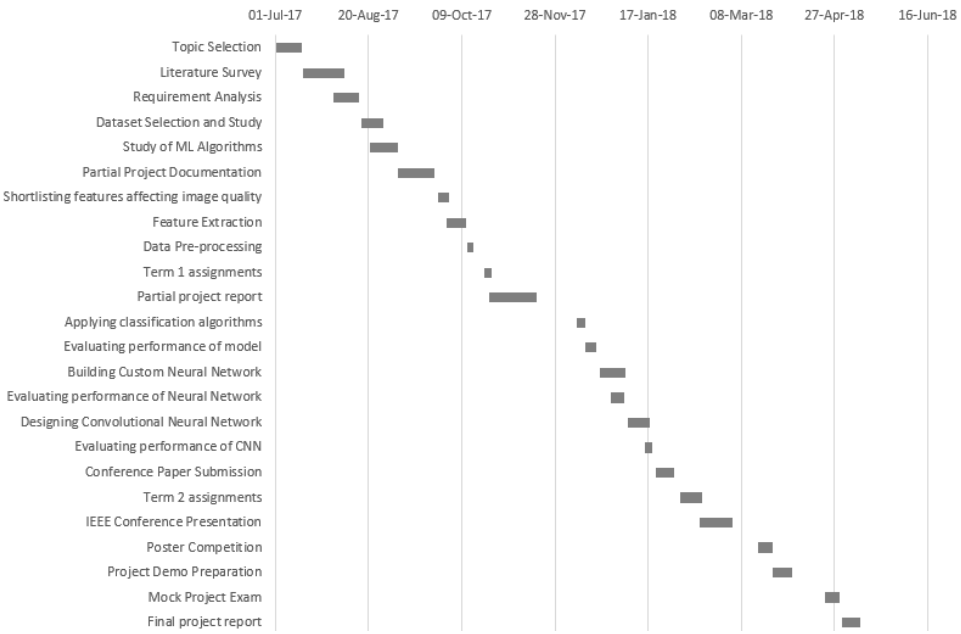


Fig 5.2 Timeline Chart

5.1.2. Project Resources

The major hardware requirement for this project is a Linux 64 bit machine with NVIDIA GPUs. Various software resources used include -

- CUDA
- TenserFlow with Keras
- NVIDIA Digits
- OpenFace
- Amazon Rekognition
- AWS EC2 Services
- Anaconda with Jupyter Notebooks

5.2. Risk Management w.r.t. NP Hard analysis

Refer Annexure B for the detailed proof of NP-completeness.

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem that might happen, or it might not. But, regardless of the outcome, it is a really good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

Steps : Recognizing what can go wrong is the first step, called risk identification. Next, each risk is analyzed to determine the likelihood that it will occur and the damage that it will do if it does occur. Once this information is established, risks are ranked, by probability and impact. Finally, a plan is developed to manage those risks with high probability and high impact. There are two basic strategies to deal with the risks that occur/may occur in the project: proactive and reactive strategies. The reactive strategy means not worrying about the problem until it happens. At best, a reactive strategy monitors the project for likely risks. Resources are set aside to deal with them, should they become actual problems. More commonly, the software team does nothing about risks until something goes wrong. Then, the team flies into action in an attempt to correct the problem rapidly. This is often called a fire fighting mode. When this fails, crisis management takes over and the project is in real jeopardy. A considerably more intelligent strategy for risk management is to be proactive. A proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability and impact are assessed, and they are ranked by importance. Then, the software team establishes a plan for managing risk. The primary objective is to avoid risk, but because not all risks can be avoided, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner. Throughout the remainder of this chapter, we discuss a proactive strategy for risk management.

5.2.1.1. Risk Identification

Risk identification is a systematic attempt to specify threats to the project plan (estimates, schedule, resource loading, etc.). By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.
There are two distinct types of risks : generic risks and product-specific risks. Generic risks are a potential threat to every software project. Product-specific risks can be identified only by those with a clear understanding of the technology, the people, and the environment that is specific to the project at hand. To identify product specific risks, the project plan and the software statement of scope are examined and an answer to the following question is developed: ”What special characteristics of this product may threaten our project plan?”
One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories:

- Product size – risks associated with the overall size of the software to be built or modified.

- Business impact – risks associated with constraints imposed by management or the marketplace.
- Customer characteristics – risks associated with the sophistication of the customer. and the developer's ability to communicate with the customer in a timely manner.
- Process definition – risks associated with the degree to which the software process has been defined and is followed by the development organization.
- Development environment – risks associated with the availability and quality of the tools to be used to build the product.
- Technology to be built – risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- Staff size and experience – risks associated with the overall technical and project experience of the software engineers who will do the work.

The following questions have derived from risk data obtained by surveying experienced software project managers in different part of the world. The questions are ordered by their relative importance to the success of a project.

1. Have top software and customer managers formally committed to support the project?
2. Are end-users enthusiastically committed to the project and the system/product to be built?
3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end-users have realistic expectations?
6. Does the software engineering team have the right mix of skills?
7. Are project requirements stable?
8. Is the number of people on the project team adequate to do the job?
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

5.2.2. Risk Analysis

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Damage to surveilling entity	Low	Low	Medium	High
2	Failure of server - client connectivity	Low	Low	Medium	Medium
3	Sudden increase in image load leading to loss of data	Low	Low	High	Low

Table 5.2 Risk Table

Probability	Value	Description
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 − 75%
Low	Probability of occurrence is	< 25%

Table 5.3 Risk Probability Definitions

Impact	Value	Description
High	> 10%	Schedule impact or Unacceptable quality
Medium	5 − 10%	Schedule impact or Some parts of the project have low quality
Low	< 5%	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.4 Risk Impact Definitions

5.2.3. Overview of Risk Mitigation, Monitoring, Management

Risk ID	1
Risk Description	Damage to surveilling entity
Category	External Factors
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Replace damaged entity as soon as possible
Strategy	Keep tabs from time to time on working of surveilling entity
Risk Status	Identified

Risk ID	2
Risk Description	Failure of server - client connectivity
Category	Link Failure
Source	This was identified during early development and testing
Probability	Low
Impact	Medium
Response	Keep images in a buffer till connection is re-established
Strategy	Before discarding image from cache, it must be checked whether it is successfully forwarded to server
Risk Status	Identified

Risk ID	3
Risk Description	Sudden increase in image load leading to loss of data
Category	Implementation
Source	This was identified during testing.
Probability	Low
Impact	Low
Response	Provide additional buffer space to save images temporarily
Strategy	Keep track of free cache size
Risk Status	Identified

5.3. Project Schedule

5.3.1. Project Task Set

Major Tasks in the Project stages are:

- Task 1: Study of Existing Systems
- Task 2: Results are discovered problems with existing systems.This results serve as ground for design choices
- Task 3: Plan for alternatives to the existing systems.
- Task 4: Present the concept of local and sequential text scanning with continuous auditory and tactical feedback.
- Task 5: Demonstrate validity of design conducting early user studies with visually impaired to assess it.
- Task 6: Checks to see if it serves its practical and real time use.

5.3.2. Task Network

Project tasks and their dependencies are noted in this table form. The below snapshot shows different tasks along with their timeline. These tasks will be done sequentially and they are dependent on each other.

Task	Start Date	End Date	Duration
Topic Selection	01-Jul-17	15-Jul-17	14
Literature Survey	16-Jul-17	07-Aug-17	22
Requirement Analysis	01-Aug-17	15-Aug-17	14
Dataset Selection and Study	16-Aug-17	28-Aug-17	12
Study of ML Algorithms	21-Aug-17	05-Sep-17	15
Partial Project Documentation	05-Sep-17	24-Sep-17	19
Shortlisting features affecting image quality	26-Sep-17	02-Oct-17	6
Feature Extraction	01-Oct-17	11-Oct-17	10
Data Pre-processing	12-Oct-17	15-Oct-17	3
Term 1 assignments	21-Oct-17	25-Oct-17	4
Partial project report	24-Oct-17	18-Nov-17	25
Applying classification algorithms	10-Dec-17	14-Dec-17	4
Evaluating performance of model	14-Dec-17	20-Dec-17	6
Building Custom Neural Network	22-Dec-17	05-Jan-18	14
Evaluating performance of Neural Network	28-Dec-17	05-Jan-18	7
Designing Convolutional Neural Network	06-Jan-18	18-Jan-18	12
Evaluating performance of CNN	15-Jan-18	20-Jan-18	4
Conference Paper Submission	21-Jan-18	31-Jan-18	10
Term 2 assignments	03-Feb-18	15-Feb-18	12
IEEE Conference Presentation	14-Feb-18	03-Mar-18	17
Poster Competition	17-Mar-18	25-Mar-18	8
Project Demo Preparation	25-Mar-18	05-Apr-18	10
Mock Project Exam	22-Apr-18	30-Apr-18	8
Final project report	01-May-18	11-May-18	10

Fig 5.3 Project Tasks

5.3.3. Timeline Chart

A project timeline chart is presented. This includes a timeline for the entire project. Refer Annexure D for the same.

5.4. Team Organization

The team for B.E. final year project consists of a team of students, a college professor as an internal college guide and industry professionals (in case of industry sponsored project) as external guides making collaborative efforts for the fulfillment and implementation of the project problem statement.

5.4.1. Team Structure

The team of students consists of 4 final year students of Department of Computer Engineering, MITCOE.

Roll Number	Name
3142023	Aditya Deshpande
3142024	Mrunmayi Deshpande
3142026	Darshana Gadre
3142062	Alisha Shahane

Table 5.5 Team Structure

Each and every member of team is responsible for identification of the problems and constraints, proposing problem solving methodologies, breaking the problems into components, identifying the approaches for implementation and documentation.

Dr. Siddhivinayak Kulkarni, Professor, Department of Computer Engineering, MITCOE is the internal college guide for providing thorough domain guidance, resolving doubts, suggesting and approving solution approaches and ensuring timely completion of activities.

The project is sponsored by Persistent Systems Ltd. and the external guide (Persistent Project mentors) guiding us is Dr. Bhushan Garware. The Industry Project Mentors guide the student team as per the company requirements and expected outcome from the project, through their technical expertise and industry experience help the student team in various technical difficulties and ensure that strict software engineering practices are followed for developing a high quality product.

5.4.2. Management Reporting and Communication

To ensure smooth functioning of various activities during the project development, the following communication strategy has been adopted:

- The team members of the student team communicate and meet together in every two days to divide the assigned work efficiently among the members

and identify which of the assigned tasks are completed and which of them are yet to be completed or under completion.

- The student team reports in person to the internal college guide every week and discuss on the previously assigned activities and tasks are assigned for the period before the next meet, all the activities discussed during this meeting are updated in the project log book.
- The student team also reports in person to the Industry Project Mentors from Persistent Systems Ltd., in the premises of Persistent itself with a frequency of once in every 7 to 10 days depending on the task deadlines for every task.
- There is a shared Google group consisting of student team members, internal guide and external guides which is used for intermediate communication, all the resources and documents of tasks completed are shared via Google drive and when the core development of project starts the code will be maintained by using the version control tool - git.

Refer Annexure C for more details .

Chapter 6

Software Requirement Specification

6.1. Introduction

6.1.1. Purpose and Scope of Document

Currently facial recognition softwares are very slow because of their high time complexity. The performance of real time application of such softwares is not up to the mark. Moreover, a single classifier is unable to classify all sorts of faces. Some of the main obstacles that these classifiers face are images with low light in the background, blurred facial features, fog, different hairstyles of the same person. Our project aims to improve the applicability of existing facial recognition softwares by ensuring that the images sent as input to such software will always be matched successfully.

This project aims to create a framework that will be implemented before actual facial recognition takes place and try to predict the performance of that facial recognition by estimating image quality. The framework can be used in CCTV monitoring and real time analysis. It can also be used for criminal identification or detection of trespassers. The input image resolution, type might affect the performance of the framework.

6.1.2. Overview and responsibilities of developer

- The developer should iteratively develop:
    - The deployable solution
    - Models required for the properly controlled development of the solution
    - Models and documentation required for the purpose of supporting the Solution
- Testing the output of their own work prior to independent testing
- Carrying out all types of technical testing of the solution as a whole
- Creating testing products, e.g. test cases, plans and logs

6.2. Usage Scenario

6.2.1. User roles

Framework: The framework will provide a probability based on which, a decision can be made whether the image should be sent to the server for matching or not. The framework will be placed a step before the actual facial recognition system.

6.2.2. Use Case

Sr. no.	Use Case	Description	Actor	Assumptions
1	Detect Faces	From the captured image, detect the face and preprocess it(crop and grayscale conversion).	Framework	Captured image is a facial image
2	Extract Features	From the preprocessed image, extract various features.	Framework	Image is pre processed successfully
3	Estimate Quality	Generate quality estimate by passing the image through the trained CNN	Framework	The quality will be within the matcher range

Table 6.1 Use Cases

6.2.3. Use Case View

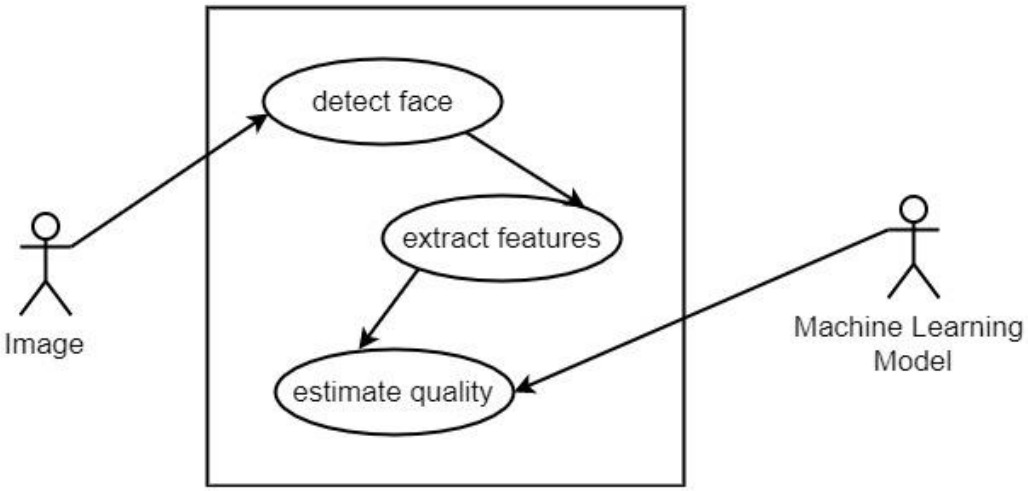


Fig 6.1 Use Case Diagram

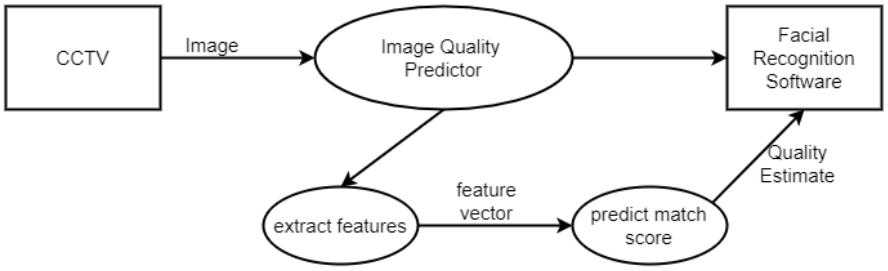
6.3. Functional Model and Description

There are three main functions:

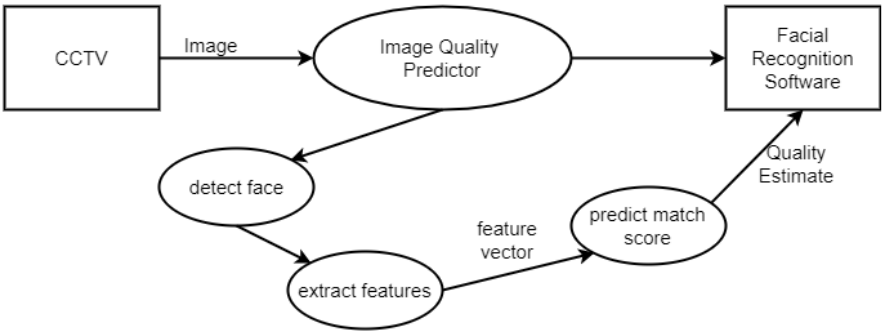
- Image Preprocessing
- Feature Extraction
- Quality Estimation

6.3.1. Data Flow Diagram

Level 0 Data Flow Diagram



Level 1 Data Flow Diagram



Level 2 Data Flow Diagram

Fig 6.2 Data Flow Diagram

6.3.2. Description of functions

- Image Preprocessing: After capturing a facial image, it needs to be preprocessed. It is cropped and converted to grayscale for further operations.

- Feature Extraction: Once the image is preprocessed, its features need to extracted in order to estimate its quality.

- Quality Estimation: Based on the features extracted and the classification model the quality of an image is estimated. This quality is responsible for accepting or rejecting an image. If accepted the image is sent for actual facial recognition.

6.3.3. Activity Diagram

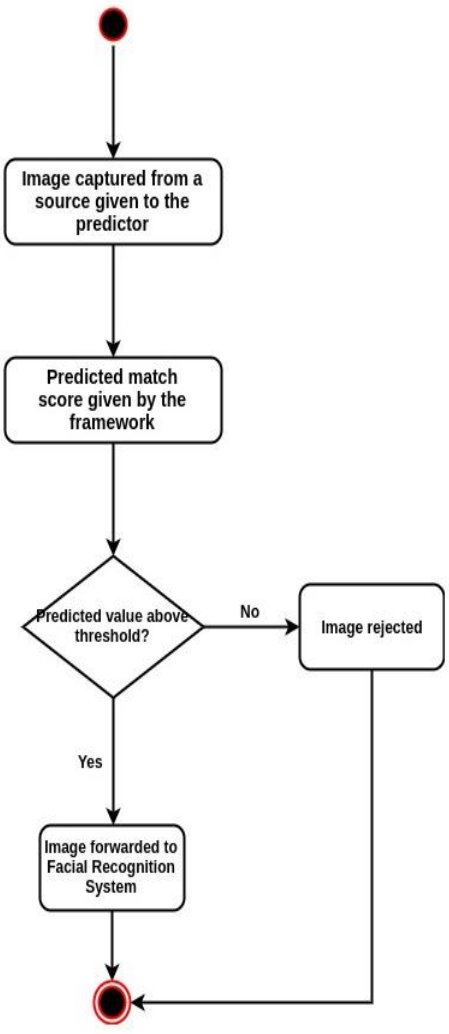


Fig 6.3 Activity Diagram

6.3.4. Non Functional Requirements

- Interface Requirements:
    - The interface between client and server must be completely transparent.
    - The image should be forwarded to the framework from cctv as it is, without any loss.
- Performance Requirements:
    - Response time should be low as it finds its application in the security domain.
    - Should be able to handle high workload.
- High Availability: One of the main purpose of such a framework is to increase real time applicability of a facial recognition system. Thus it should be reliable and available all the time.
- Modifiability:
    - Portable: The framework can be customized for different matchers.
    - Scalability: Performance should be consistent with the increase in the number of images captured.

6.3.5. State Diagram

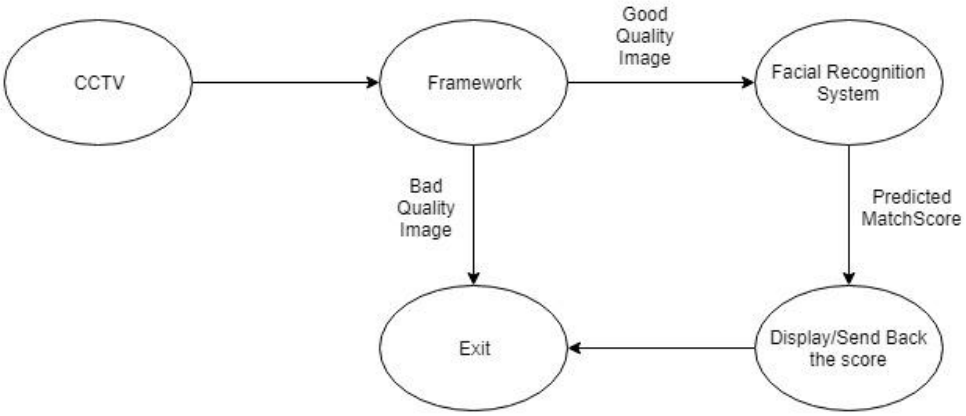


Fig 6.4 State Diagram

6.3.6. Design Constraints

- The threshold for good and bad quality images differs for every matcher.
- Input image requirements for every matcher is different (eg: .jpg or .png, grayscale or coloured)
- It is necessary to ensure that at least one image of a person is forwarded to the facial recognition.

6.3.7. Software Interface Description

The framework will run before the actual facial recognition takes place. The captured images will be passed through the framework and only good quality images will be forwarded.

Chapter 7

Detailed Design Document using Appendix A and B

7.1. Introduction

This document specifies the design that is used to solve the problem of Product. This document also gives the detailed view of architecture of the device.

7.2. Architectural Design

The program architecture can be considered as a 2-tier architecture. The CCTVs form the 1st Tier which process the images. The framework runs on the CCTVs itself and makes a decision regarding which images are to be forwarded to the facial recognition software which resides in the 2nd Tier.

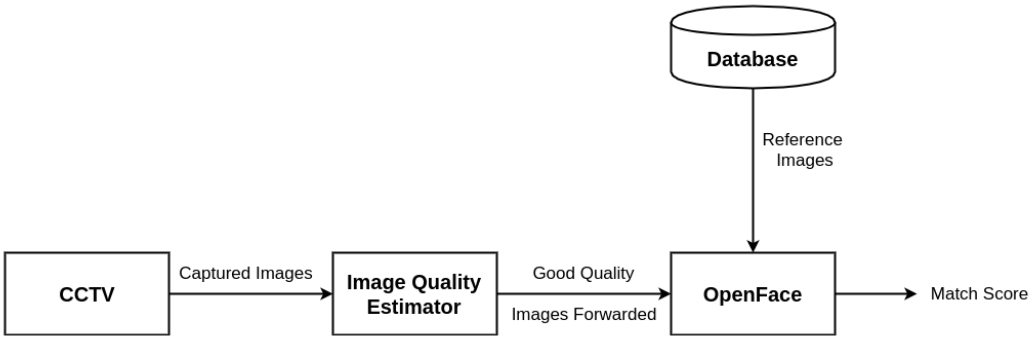


Fig 7.1 Architecture Diagram

7.3. Data design (using Appendices A and B)

This section contains a description of all data structures including internal, global, and temporary data structures, database design (tables), file formats.

7.3.1. Internal software data structure

In the machine learning approach we have used DataFrames to store the feature vector as well as class labels. A DataFrame is a 2-dimensional 36uclidea data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table, or a dict of Series objects. It is generally the most commonly used pandas object. Like Series, DataFrame accepts many different kinds of input. Along with the data, you can optionally pass index (row labels) and columns (column labels) arguments. If you pass an index and / or columns, you are guaranteeing the index and / or columns of the resulting DataFrame.

In the deep learning approach, we have used Numpy arrays to store image data. An ndarray is an N-dimensional array type. It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index. Every item in an ndarray takes the same size of block in the memory. Each element in ndarray is an object of data-type object (called dtype). Any item extracted from ndarray object (by slicing) is represented by a Python object of one of array scalar types.

7.3.2. Global data structure

The convolutional neural network was trained for 120 epochs. At the end of this training stage, the weights of the neurons in the network were stored in a HD5 file. Hierarchical Data Format (HDF) is a set of file formats (HDF4, HDF5) designed to store and organize large amounts of data. Originally developed at the National Center for Supercomputing Applications, it is supported by The HDF Group, a non-profit corporation whose mission is to ensure continued development of HDF5 technologies and the continued accessibility of data stored in HDF.

7.3.3. Temporary data structure

During the extraction of genuine match scores N x N matrices were used to store the scores of all images with respect to each other. This matrix was used to determine the best image as the row of the matrix having the least sum. Then the math scores of all other images with respect to this image were stored as the genuine scores.

7.3.4. Database description

Chokepoint dataset [3] is specifically designed for experiments in person identification / verification under real-world surveillance conditions. An array of cameras was used to capture a sequence of images from different angles as subjects walked by. The images captured through this staged experiment have considerable variations in terms of illumination conditions, pose, sharpness, as well as misalignment. Thus this database is ideal for our problem which aims to improve facial recognition performance in CCTV footage. This database contains a total of 63,570 images of 29 subjects. We used a balanced portion of this dataset of around 30,000 images to train the Convolutional Neural Network with equal number of images below and above the threshold of 0.84.

7.4. Component Design

7.4.1. Algorithm

Steps followed in the task of facial image quality estimation:

1. Obtain images captured by a CCTV.
2. Crop faces from the images and discard the rest.
3. Convert the cropped facial images into grayscale.
4. Run the image through the CNN.
5. Depending on the quality estimate given by the CNN either discard the image or forward it to the server for face recognition.

7.5. Component Description

One of the main components of this framework are the image pre-processing module which crops faces from images captured by the CCTV and converts them into grayscale. Secondly, this framework contains a Convolutional Neural Network that runs on the CCTVs which are the logical extremes of the network as per Edge Computing principles. The CNN accepts cropped, grayscale facial images as input and returns a class label indicating whether the quality of the image is good enough for it to be forwarded to the actual face recognition system which runs on the server side. The face recognition system compares the forwarded images against reference images stored in a database and tries to match them.

Chapter 8

Project Implementation

8.1. Introduction

Our framework is basically a two class classification model. In order to build the dataset for classification we used OpenFace and ChokePoint dataset.

OpenFace: It is a general purpose open source matching library. It is implemented using python and torch and has a deep neural network. Working:

- Detect faces through a pretrained model.
- Transform face: Position the eyes and lips for all images at same location.
- Using DNL represent or (embed) face in a 128-dimensional unit hypersphere.
- Larger distance between embeddings denotes that images are likely not of the same person.
- This makes clustering, classification easier where 39uclidean distance between features is not meaningful.

Using the "compare" API of OpenFace, a matchscore can be generated between two images. This matchscore ranges between 0 to 4 where 0 indicates best match and 4 indicates a bad match or a no match.

ChokePoint: It is a dataset designed for experiments in person identification/verification under real-world surveillance conditions. An array of cameras were placed which captured the images if subjects passing through. When a subject walks through a series of images were captured (face set). Due to automatic face localization/detection, these images have variations in terms of illumination, sharpness, and misalignment of face. The experiment was done on 29 subjects and a total of 64,204 images were captured. Since our problem is a 2 class classification problem, we used a balanced dataset of 29,022.

To get this balanced dataset, we obtained the matchscores of all images using openface. To do so the first step was to find the best images for every subject which was considered as the best image. All other images of a subject were compared to this reference image to get the matchscore. These are the genuine matchscore since we compared the images of same subject. We found equal number of imposter scores as well. As shown in the distribution, the threshold is 0.84 where FAR is 0.09 and FFR 0.24. The threshold indicates that images with matchscore below 0.84 are labelled as 'good' images and those having a score above it are 'bad' images. This is how we created a balanced dataset.

We propose a framework which will run before the actual facial recognition, using which only the good quality images will be forwarded for FR. Such a framework will help increase the real time applicability of FR since it has to process only on good quality images for which it can definitely produce a match. And this system/framework will work in line with the principles of edge computing which suggests that maximum processing should be done at the source of data, which here are CCTVs. In order to implement such a framework we took various approaches.

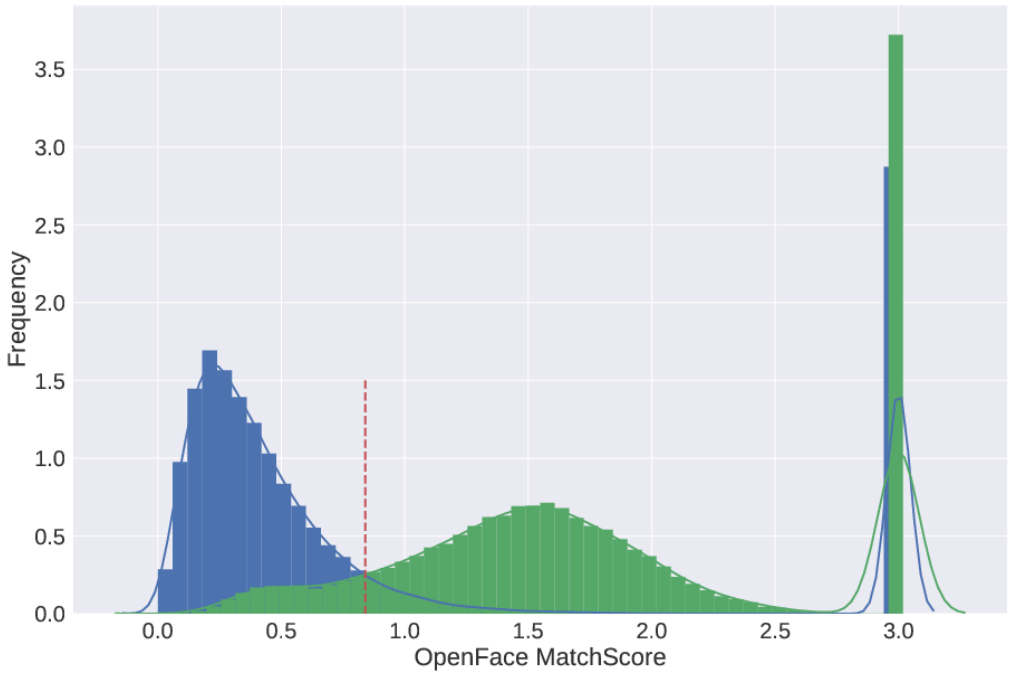


Fig 8.1 Genuine Imposter Matchscore Distribution

8.2. Tools and Technologies Used

8.2.1. Tools

OpenFace, AWS, Jupyter Notebooks, Nvidia Digits

8.2.2. Technologies

Keras, tensorflow, anaconda, viola-jones, matplotlib, h5py, scikit-learn

8.3. Methodologies/Algorithms Details

Feature Extraction: We studied the behaviour of OpenFace by obtaining match scores of many different facial images in order to determine which factors potentially alter match scores. We observed that many image quality metrics such as brightness, contrast, sharpness directly affected match score. Apart from these we also observed that the area covered by a face in an image also influenced match score. Lastly, obstruction of face by sunglasses, eyeglasses, caps, facial hair, different hairstyles, all impacted the match score. So these features were extracted and a feature vector of such features was created.

Machine Learning Approach: After completing the data pre-processing and standardization, we divided the dataset into 70-30 ratio for training and testing of the classification model respectively. In the training phase, the model was built using the feature vector as the independent variable of classification and the

corresponding quality as the class labels. To build the model we used various classification algorithms. The results are shown in the table below.

Algorithm	**f1-score**
Gaussian NB	0.66
Decision Tree	0.68
Random Forest	0.62
SVM	0.66
KNN	0.66

Table 8.2 F1 Scores

The same results in the form of a ROC-

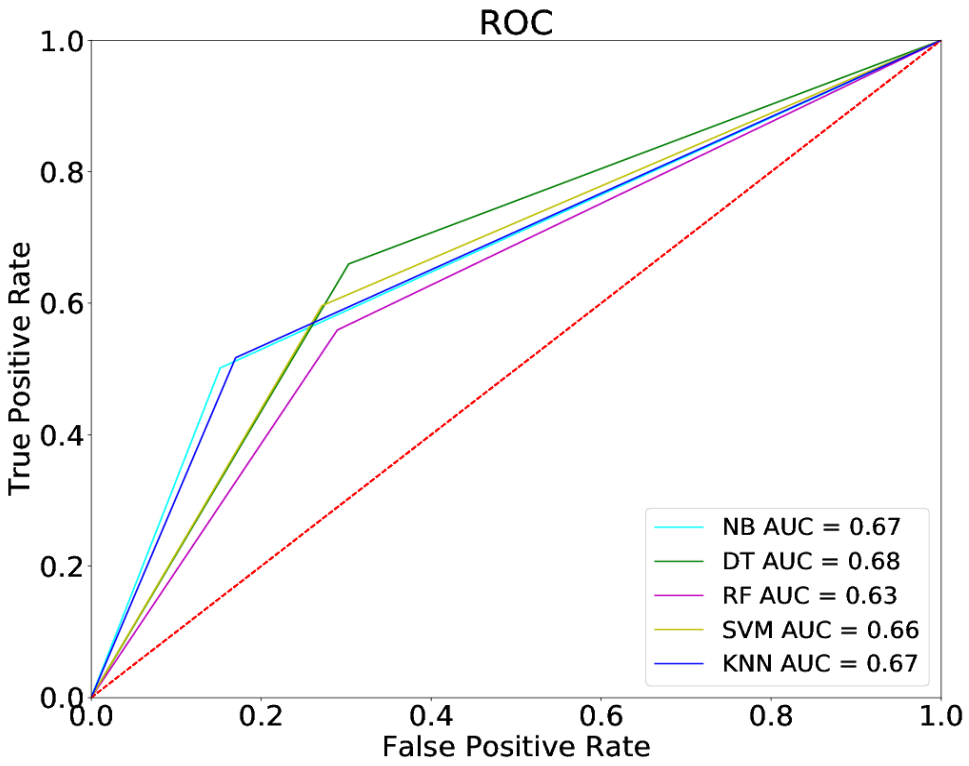


Fig 8.2 ROCs

We also built the model on a custom Neural Network with which we could achieve a highest accuracy of 67.74%. The model has two densely connected hidden layers and was trained for 1000 epochs. The model summary and result,

Layer Output Shape Param	Layer Output Shape Param	Layer Output Shape Param
dense 1(Dense) (None, 10) 110	dense 1(Dense) (None, 10) 110	dense 1(Dense) (None, 10) 110
dropout 1(Dropout) (None, 10) 0	dropout 1(Dropout) (None, 10) 0	dropout 1(Dropout) (None, 10) 0
dense 2(Dense) (None, 5) 55	dense 2(Dense) (None, 5) 55	dense 2(Dense) (None, 5) 55
dropout 2(Dropout) (None, 5) 0	dropout 2(Dropout) (None, 5) 0	dropout 2(Dropout) (None, 5) 0
dense 3(Dense) (None, 2) 12	dense 3(Dense) (None, 2) 12	dense 3(Dense) (None, 2) 12
dropout 3(Dropout) (None, 2) 0	dropout 3(Dropout) (None, 2) 0	dropout 3(Dropout) (None, 2) 0
dense 4(Dense) (None, 1) 3	dense 4(Dense) (None, 1) 3	dense 4(Dense) (None, 1) 3

Table 8.3 Neural Network Architecture

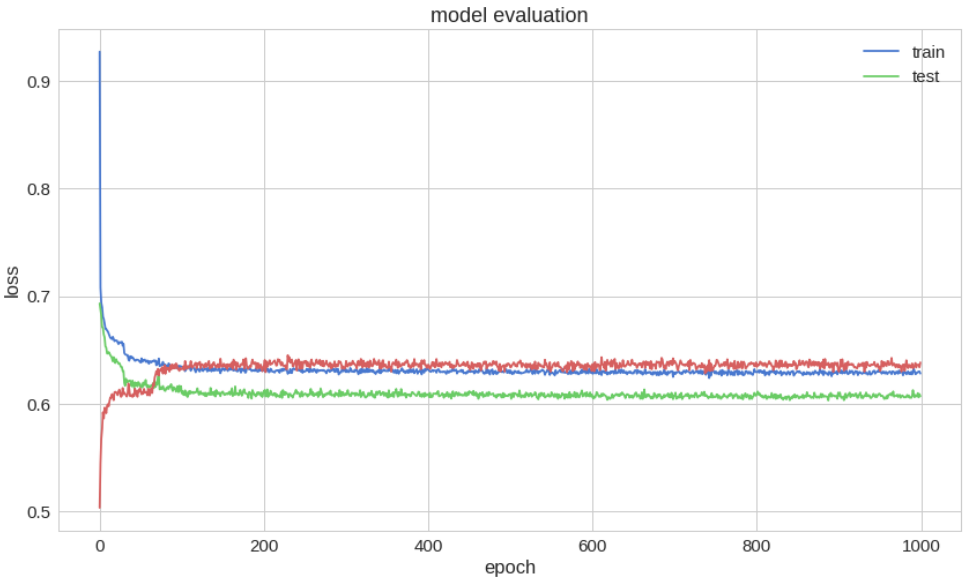


Fig 8.3 Neural Network Result

Deep Learning Approach: We used a convolutional neural network to build the framework. We based the architecture of our convolutional neural network on AlexNet. We modified the output layer to implement binary classification.

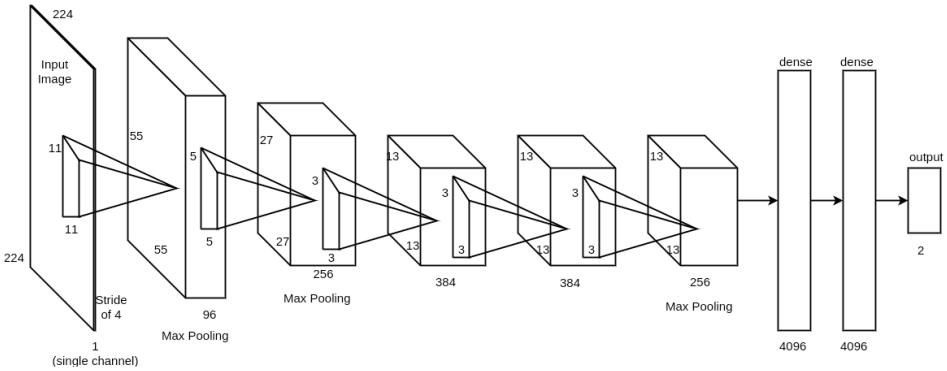


Fig 8.4 CNN Architecture

The input shape to our CNN was a single channel since the dataset contains grayscale images. The input image data to the network was pre-processed by performing mean image subtraction and normalization of pixel values from [0, 255] into [0, 1] range. Mean subtraction involves subtracting the mean across every individual feature in the data and has the geometric interpretation of centering the cloud of data around the origin along every dimension. Normalization refers to normalizing the data dimensions so that they are approximately of the same scale. The CNN was trained for 130 epochs, at which point the accuracy was constant. The accuracy of the deep learning framework is 92.274%

8.4. Verification and Validation for Acceptance

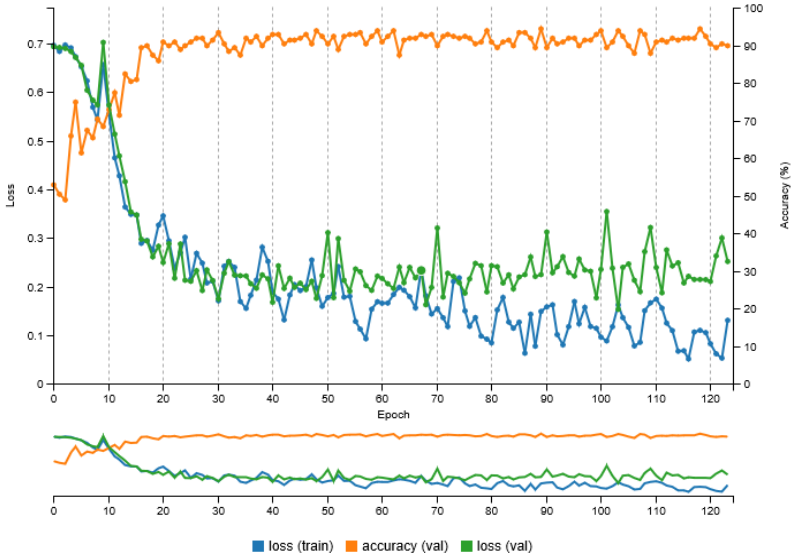


Fig 8.5 CNN Result

Chapter 9

Software Testing

9.1. Type of Testing Used

Black Box Testing

Black Box Testing is often categorised as functional testing, but can, to some extent, be seen as a type of User Acceptance Testing. Its, basically, a method of software testing which analyses certain functionalities without letting the tester see the internal code structure of the software. Therefore, Black Box Testing can also be applied to User Acceptance Testing, because Black Box Tests do share the same principles as User Acceptance Tests. During Black Box Tests the user isn't aware of any code base, but only about the requirements which the software should meet.

White Box Testing

White-box testing also known as clear box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system,as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually.

Integration Testing

Integration testing (sometimes called integration and testing, abbreviated IT) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Regression Testing

It is testing the application as a whole for the modification in any module or functionality. It is re-testing of a program after doing the modification. It helps in ensuring that faults have not been introduced or uncovered as a result of the changes made and that the modified system still meets its requirements.

Load Testing

It is a performance testing to check system behaviour under load. Testing an application under heavy loads, such as testing of a website under a range of loads to determine at what point the systems response time degrades or fails.

9.2. Test Cases and Test Results

9.2.1. Black Box Testing

Sr. No.	Test Case	Description	Expected Output	Actual Output
1	Image with match score greater than threshold is given to the framework	Framework should predict class of image	Image should be discarded	Image not passed to the matcher for further processing
2	Image not containing a face is given as input	Framework should predict class of image	Image should not be passed to the matcher for further processing	Image pruned at edge

Table 9.1 Black Box Testing

9.2.2. White Box Testing

Sr. No.	Test Case	Description	Expected Output	Actual Output
1	ROC Curves	Graphical plot that illustrates the diagnostic ability of a binary classifier system	Monotonically increasing, start at (0,0)(0,0) and arrive at (1,1)(1,1)	Monotonically increasing, start at (0,0)(0,0) and arrive at (1,1)(1,1)
2	F1-Score	F1-score is the harmonic average of the precision and recall. F1-Score lies between 0 and 1.	F1-Score should be closer to 1.	F1-Score ranging from 0.62 to 0.68
3	Accuracy	Accuracy is the fraction of predictions our model got right.	Accuracy should be closer to 100%.	Accuracy of 92.27% for CNN.
4	Loss	Loss value implies how well a certain model behaves after each iteration of optimization.	Reduction of loss after each, or several, iteration(s).	Reduction of loss after each, or several, iteration(s).

Table 9.2 White Box Testing

Chapter 10

Results

10.1. Screenshots

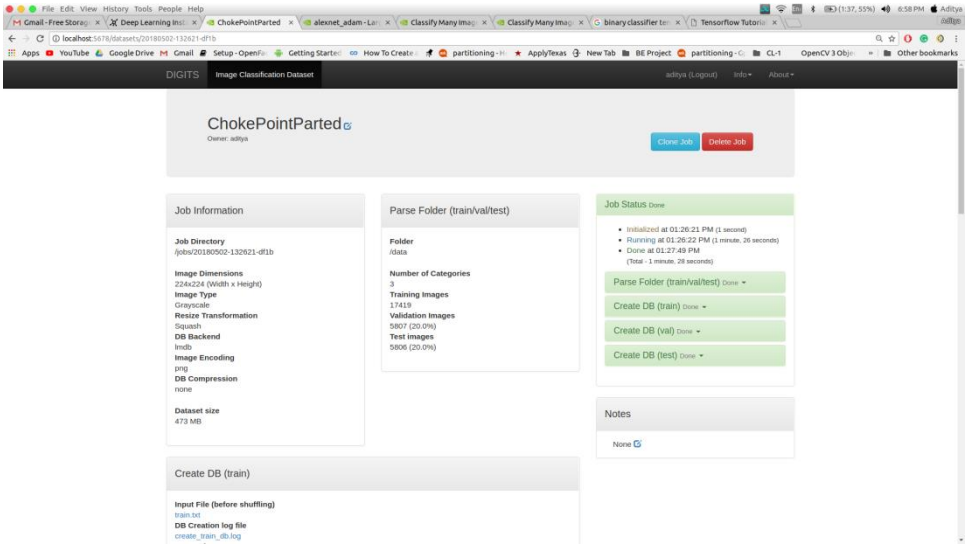


Fig 10.1 Dataset Creation

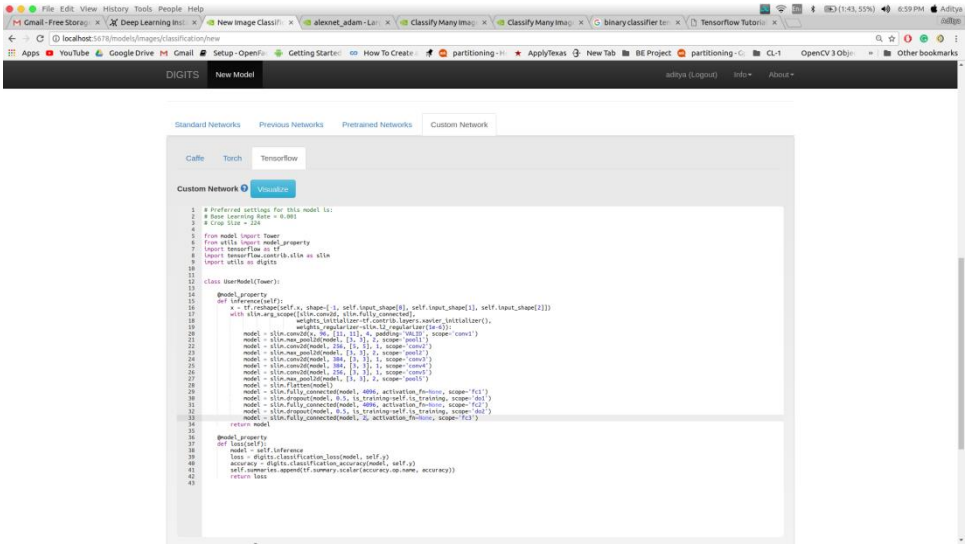


Fig 10.2 Model Creation

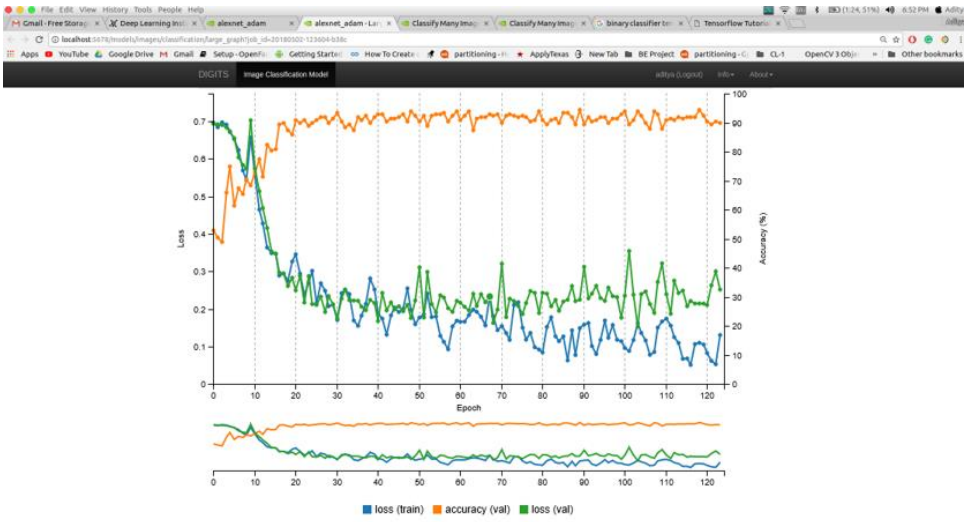


Fig 10.3 CNN

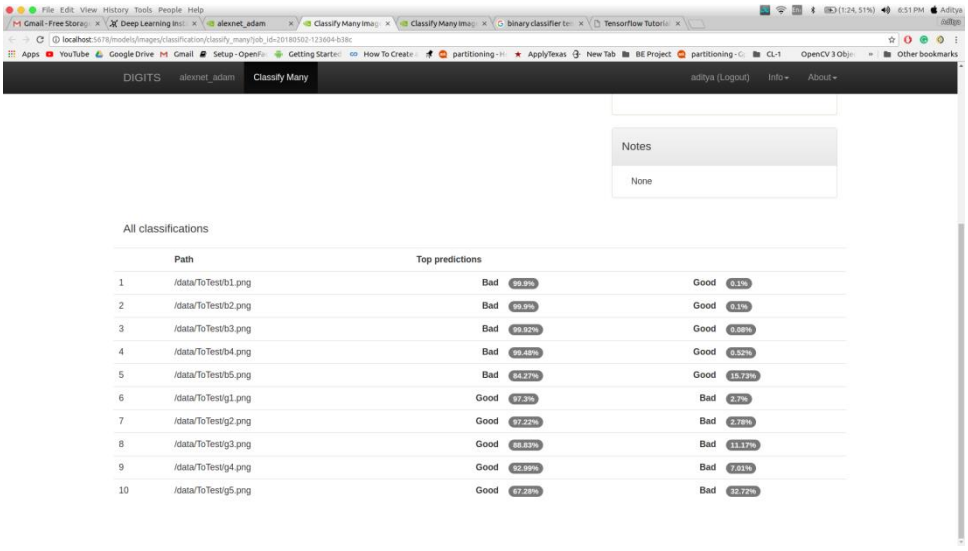


Fig 10.4 Prediction

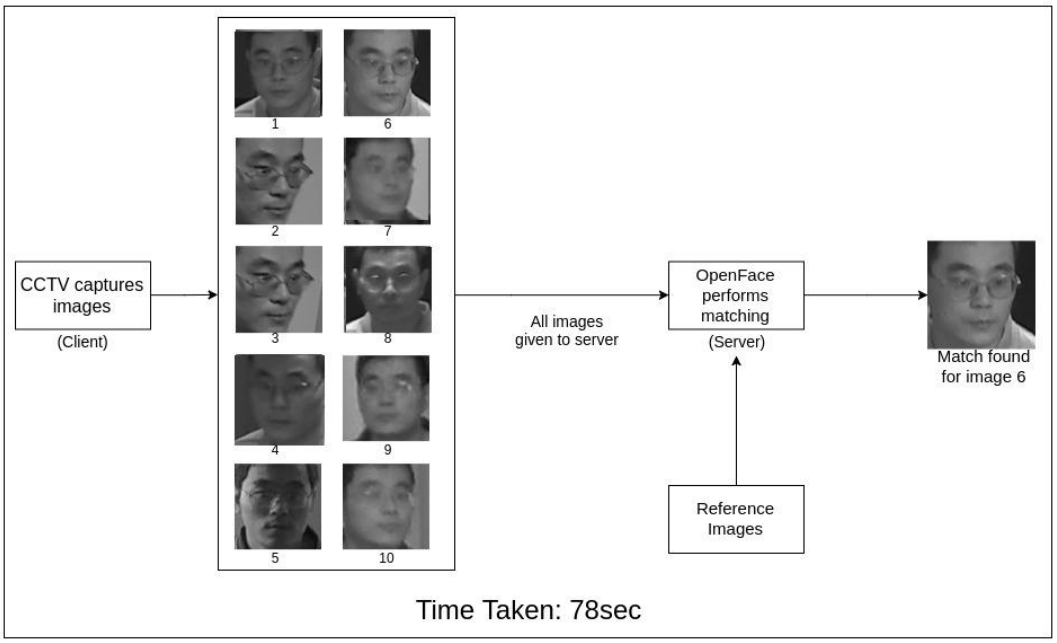


Fig 10.5 Demo [without our framework]

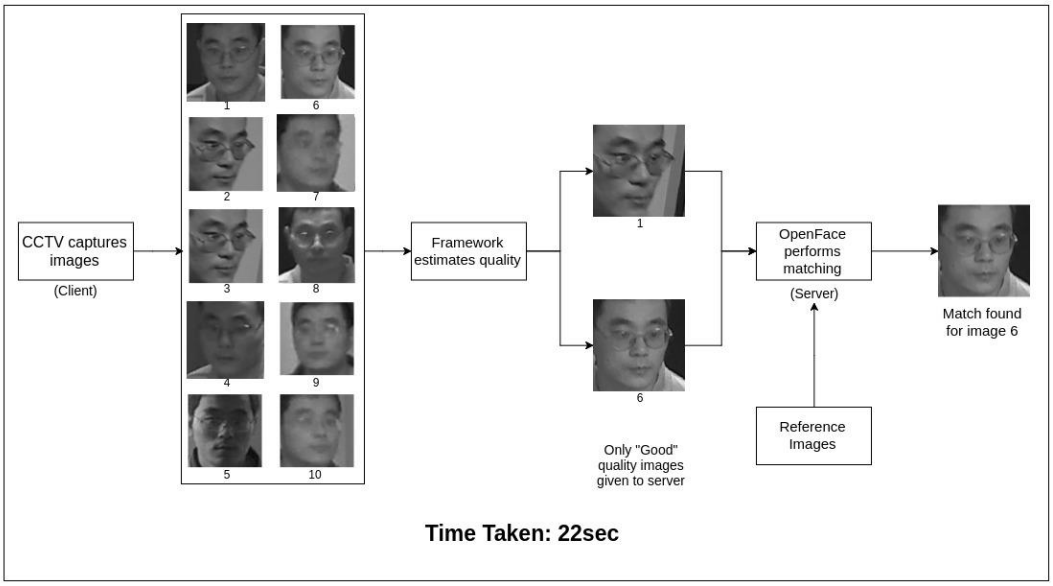


Fig 10.6 Demo [with our framework]

Chapter 11

Deployment and Maintenance

11.1. Installation and Uninstallation

The user should install all the dependencies needed. The dependencies needed are as follows:

- Anaconda
  bash Anaconda3-5.0.1-Linux-x86_64.sh
  source ~/.bashrc
- Ipython Notebook
  Sudo apt-get install ipython-notebook
- Tenserflow
  conda create -n tensorflow pip python=2.7
  source activate tensorflow
- Keras
  pip install keras
- Numpy
  sudo apt-get install python-numpy
- H5py
  sudo pip install h5py
- OpenCV
  sudo apt-get install python-opencv

For building the machine learning and deep learning models installations needed are as follows:

- Openface
- The dependencies needed for installation of openface are:
  - Gcc
  - Cmake
  - OpenCV
  - boost

- Scikit-learn
  pip install -U scikit-learn
- Matplotlib
  python -mpip install -U matplotlib
- Graphviz
  sudo apt-get install graphviz
- Seaborn
  pip install seaborn
- Amazon Web Services

Chapter 12

Conclusion and Future Scope

12.1. Conclusion

The proposed framework maps the relationship between facial image quality and performance of facial recognition using deep learning. This framework can be embedded at the logical extremes of the network which are CCTVs, as per Edge Computing Principles. This approach can be a solution towards making facial recognition feasible for real-time security applications. Such applications include surveillance and monitoring systems, intrusion detection systems, etc.

12.2. Future Scope

This framework can also be customized for any other matcher instead of OpenFace. Currently, our framework classifies a facial image as acceptable or not based on its quality. Rejecting all unacceptable images is not a good practice as it leads to information loss. Rather than simply rejecting these images, some efforts need to be taken to enhance the quality of these images. It would also be interesting to apply the same model to predict the performance of other biometric systems.

Annexure A

References

[1] - Margaret Rouse, Peter Loshin, Michael Cobb. Retrieved 28 December 2017. http://searchsecurity.techtarget.com/definition/biometrics

[2] - Margaret Rouse, Peter Loshin, Michael Cobb. Retrieved 28 December 2017. http://searchsecurity.techtarget.com/definition/biometrics

[3] - Y. Wong, S. Chen, S. Mau, C. Sanderson, B.C. Lovell, "Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition"
In *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81-88. IEEE, June 2011.

[4] - B. Amos, B. Ludwiczuk, M. Satyanarayanan,"Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016. Retrieved 28 December 2017. http://reports-archive.adm.cs.cmu.edu/anon/anon/2016/CMU-CS-16-118.pdf

[5] - B. Amos, B. Ludwiczuk, M. Satyanarayanan,"Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016. https://cmusatyalab.github.io/openface/demo-2-comparison/
Retrieved 28 December 2017.

[6] - P. Grother and E. Tabassi, "Performance of biometric quality measures," In IEEE Trans . Pattern Anal. Mach. Intell., vol. 19, no. 4, pp. 531–543, Apr. 2007

[7] - Abhishek Dutta, Raymond Veldhuis, Luuk Spreeuwers. "Predicting Face Recognition Performance Using Image Quality". *arXiv:1510.07119v1 [cs.CV] 24 Oct 2015*

[8] - Jiansheng Chen, Member, IEEE, Yu Deng, Gaocheng Bai, and Guangda Su. "Face Image Quality Assessment Based on Learning to Rank". In *IEEE SIGNAL PROCESSING LETTERS, VOL. 22, NO. 1, JANUARY 2015*

[9] - Christian Herrmann, Jürgen Beyerer. "Maximizing face recognition performance for video data under time constraints by using a cascade". In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*

[10] - P. J. Grother, G. W. Quinn, and P. J. Pillips, "Report on the evaluation of 2D still-image face recognition algorithms," In NIST Interagency Report 7709, 2010.

[11] - H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou, "Learning deep face representation," arXiv:1403.2802, 2014

[12] - Jeffrey Edgell , Andrew Trimpe. "4 Limitations of Facial Recognition Technology". Retrieved 28 December 2017.
https://fedtechmagazine.com/article/2013/11/4-limitations-facial-recognition-technology

[13] - A. Dutta, R. N. J. Veldhuis, and L. J. Spreeuwers. "Can facial uniqueness be inferred from impostor scores?" In Biometric Technologies in Forensic Science, BTFS

2013, Nijmegen, Netherlands, Nijmegen, October 2013. Centre for Language and Speech Technology, Radboud University.

[14] - G. Aggarwal, S. Biswas, P. J. Flynn, and K. W. Bowyer. “Predicting performance of face recognition systems: An image characterization approach”. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on, pages 52–59, 2011.

[15] - G. Aggarwal, S. Biswas, P. J. Flynn, and K. W. Bowyer. “Predicting good, bad and ugly match Pairs”. In Applications of Computer Vision (WACV), In 2012 IEEE Workshop on, pages 153–160, 2012.

[16] - J. R. Beveridge, G. H. Givens, P. J. Phillips, B. A. Draper, and Yui Man Lui. “Focus on quality, predicting FRVT 2006 performance”. In Automatic Face Gesture Recognition, 2008. FG 08. 8th IEEE International Conference on, pages 1–8, 2008.

[17] - Hanczar, Blaise; Hua, Jianping; Sima, Chao; Weinstein, John; Bittner, Michael; and Dougherty, Edward R. (2010); “Small-sample precision of ROC-related estimates, Bioinformatics” 26 (6): 822–830

[18] - A. Krizhevsky, I. Sutskever, G.E. Hinton; “ImageNet Classification with Deep Convolutional Neural Networks”; In Advances in neural information processing systems, 1097-1105, 2012

[19] - http://cs231n.github.io/neural-networks-2/ Retrieved 28 December 2017

[20] - Chokepoint Dataset http://arma.sourceforge.net/chokepoint/

Annexure B

Laboratory assignments on Project Analysis of Algorithmic Design

B1. Idea Matrix

I	D	E	A
Increase	Drive	Educate	Accelerate
Improve	Deliver	Eliminate	Avoid
Ignore	Decrease	Evaluate	Associate

Table B1 Idea Matrix

- **Increase**
- To increase real time applicability of facial recognition systems
- To increase security measure through facial recognition

- **Improve**
- To improve real time performance of facial recognition systems
- To improve face image extraction from bulk CCTV data

- **Ignore**
- To ignore all images below the threshold

- **Drive**
- To drive the real time applicability of existing facial recognition softwares
- To drive the focus away from facial recognition systems and towards the image quality.

- **Deliver**
- To deliver a prediction of facial image quality before actual facial recognition takes place.

- **Decrease**
- To decrease the excessive use of computational resources for performing facial recognition on all images captured by CCTV
- To decrease space requirement of saving the entire CCTV footage

- **Educate**
- To educate the security personnel who monitors the CCTV footage
- To educate cyber forensic analysts of CCTV data

- **Eliminate**
- To eliminate the loss of image due to connection failure by saving images locally till connection is re-established

- **Evaluate**
- To evaluate reliability of framework
- To evaluate accuracy of framework

- **Accelerate**
- To accelerate the real-time performance of facial recognition system

- **Avoid**
- To avoid damage to CCTV by any entity

- **Associate**
- To associate the domain of machine learning with cyber security and embedded systems

B2. Knowledge Canvas

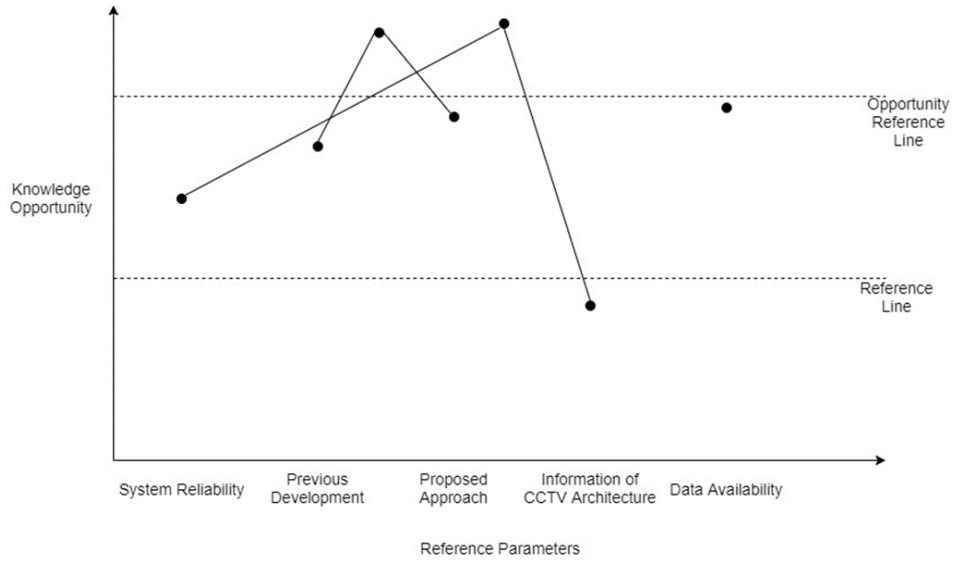


Fig B1 Knowledge Canvas

B3. Mathematical Modelling

Let the system be S = { DD, NDD, i/p, o/p, f, S, F } where-

DD : Deterministic Data = { vector features }

features = { x1, x2, x3, … , xn }

NDD : Non-Deterministic Data = {ŷ }

i/p : input = { image }

o/p : output = { quality estimate class }

f : functions = { extract_features(), extract_matchScore(),classification_model() }

S : Success = quality estimate class is close to matcher's score

F : Failure = large difference between quality estimate class and match score

B4. NP-Hard Analysis and Proof of NP-Completeness

The task is to match two given images, that is comparing an observed image to given references.

We show that it is an algorithmically hard problem to decide whether a matching between two images exists with costs below a given threshold. We show that the problem image matching is NP-complete by means of a reduction from 3-SAT, which is a common method of demonstrating a problem to be intrinsically hard.

We formalize the problem of image matching as

Let the images be given as (without loss of generality) square grids of size M x M with grayvalues (respectively node labels) from a finite alphabet $\hat{G} = \{1, \ldots, G\}$

To define the problem, two distance functions are needed, one acting on grayvalues
$d_g : \hat{G} \times \hat{G} \rightarrow N$, measuring the match in grayvalues, and one acting on displacement differences
$d_d : Z \times Z \rightarrow N$, measuring the distortion introduced by the matching.

For these distance functions we assume that they are monotonous functions (computable in polynomial time) of the commonly used squared Euclidean distance, i.e.

$$d_g(g_1, g_2) = f_1(\| g_1 - g_2 \|^2) \text{ and } d_d(z) = f_2(\| z \|^2)$$

with f1, f2 monotonously increasing.

Now we call the following optimization problem the image matching problem
(let $\hat{M} = \{1, \ldots, M\}$)

Instance: The pair (A,B) of two images A and B of size M x M.

Solution: A mapping function $f : \hat{M} \times \hat{M} \rightarrow \hat{M} \times \hat{M}$.

Measure:

$$c(A, B, f) = \sum_{(i,j)\in \hat{M}x\hat{M}} d_g\left(A_{ij}\ ,\ B_{f(i,j)}\right) + \sum_{(i,j)\in\{1,\ldots M-1\}x\hat{M}} d_d\left(f((i,j)+(1,0)) - (f(i,j)+(1,0))\right) + \sum_{(i,j)\in \hat{M}x\{1\ldots M-1\}} d_d\left(f((i,j)+(1,0)) - (f(i,j)+(1,0))\right)$$

Goal: $\min_f c(A, B, f)$

In other words, the problem is to find the mapping from A onto B that minimizes the distance between the mapped grayvalues together with a measure for the distortion introduced by the mapping. Here, the distortion is measured by the deviation from the identity mapping in the two dimensions. The identity mapping fulfils f(i, j)=(i, j) and therefore f((i, j)+(x, y))=f(i, j)+(x, y).

The corresponding decision problem is fixed by the following
Question: Given an instance of image matching and a cost c', does there exist a mapping f such that c(A, B, f)<= c'?

In the definition of the problem some care must be taken concerning the distance functions. For example, if either one of the distance functions is a constant function, the problem is clearly in P (for d_g constant, the minimum is given by the identity mapping and for d_d constant, the minimum can be determined by sorting all possible matchings for each pixel by grayvalue cost and mapping to one of the pixels with minimum cost). But these special cases are not those we are concerned with in image matching in general.

We choose the matching problem of Uchida and Sakoe (1998) to complete the definition of the problem. Here, the mapping functions are restricted by continuity and monotonicity constraints: the deviations from the identity mapping may locally be at most one pixel (i.e. limited to the eight-neighbourhood with squared Euclidean distance less than or equal to 2). This can be formalized in this approach by choosing the functions f_1, f_2 as e.g.
f_1 = id,
f_2 = step(x) := $\begin{cases} 0, & x \leq 2, \\ (10G)^{M \cdot M}, & x > 2 \end{cases}$

Reduction from 3 SAT:
3-SAT is a very well-known NP-complete problem, where 3-SAT is defined as follows:

Instance: Collection of clause C = {c_1, …. , c_K} on a set of variables X = {x_1, ... , x_L} such that each c_k consists of 3 literals for k = 1, …, K. Each literal is a variable or the negation of a variable.
Question: Is there a truth assignment for X which satisfies each clause c_k , k = 1, …, K

The dependency graph D(Φ) corresponding to an instance Φ of 3-SAT is defined to be the bipartite graph whose independent sets are formed by the set of clauses C and the set of variables X. Two vertices c_k and x_l are adjacent iff c_k involves x_l or $\overline{x_l}$.

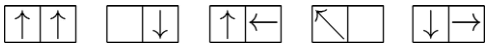
Given any 3-SAT formula Φ, we show how to construct in polynomial time an equivalent image matching problem $_l$(Φ) = (A(Φ), B(Φ)). The two images of $_l$(Φ) are similar according to the cost function (i.e. ∃f : c((A(Φ), B(Φ), f) <= 0) iff the formula Φ is satisfiable. We perform the reduction from 3-SAT using the following steps:
• From the formula Φ we construct the dependency graph D(Φ).
• The dependency graph D(Φ) is drawn in the plane.
• The drawing of D(Φ) is refined to depict the logical behaviour of Φ, yielding two images (A(Φ), B(Φ)).

For this, we use three types of components: one component to represent variables of Φ, one component to represent clauses of Φ, and components which act as interfaces

between the former two types. Before we give the formal reduction, we introduce these components.

For the reduction from 3-SAT we need five components from which we will construct the instances for image matching, given a Boolean formula in 3-DNF, respectively its graph. The five components are the building blocks needed for the graph drawing and will be introduced in the following, namely the representations of connectors, crossings, variables, and clauses. The connectors represent the edges and have two varieties, straight connectors and corner connectors. Each of the components consists of two parts, one for image A and one for image B, where blank pixels are considered to be of the background color.

We will depict possible mappings in the following using arrows indicating the direction of displacement (where displacements within the eight-neighbourhood of a pixel are the only cases considered). Blank squares represent mapping to the respective counterpart in the second image. For example, the following displacements of neighbouring pixels can be used with zero cost:



On the other hand, the following displacements result in costs greater than zero:

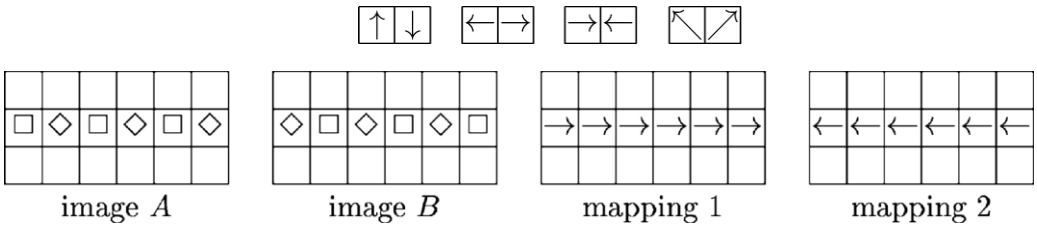


Fig. 1. The straight connector component with two possible zero cost mappings.

Fig. 1 shows the first component, the straight connector component, which consists of a line of two different interchanging colours, here denoted by the two symbols } and . Given that the outside pixels are mapped to their respective counterparts and the connector is continued infinitely, there are two possible ways in which the coloured pixels can be mapped, namely to the left (i.e. f(2, j) = (2, j-1)) or to the right (i.e. f(2, j) = (2, j+1)), where the background pixels have different possibilities for the mapping, not influencing the main property of the connector. This property, which justifies the name connector , is the following: It is not possible to find a mapping, which yields zero cost where the relative displacements of the connector pixels are not equal, i.e. one always has . f(2, j) – (2,j) = f(2, j') – (2,j'), which can easily be observed by induction over j'. That is, given an initial displacement of one pixel (which will be ∓ 1 in this context), the remaining end of the connector has the same displacement if overall costs of the mapping are zero. Given this property and the direction of a connector, which we define to be directed from variable to clause, we can define the state of the connector as carrying the

true truth value, if the displacement is 1 pixel in the direction of the connector and as carrying the false truth value, if the displacement is)1 pixel in the direction of the connector. This property then ensures that the truth value transmitted by the connector cannot change at mappings of zero cost.

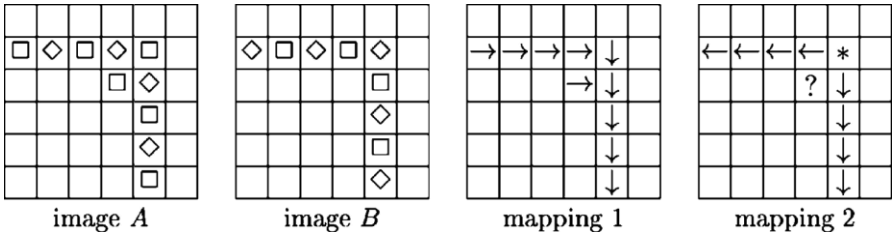


Fig. 2. The corner connector component and two example mappings.

For drawing of arbitrary graphs, clearly one also needs corners, which are represented in Fig. 2. By considering all possible displacements which guarantee overall cost zero, one can observe that the corner component also ensures the basic connector property. For example, consider the first depicted mapping, which has zero cost. On the other hand, the second mapping shows, that it is not possible to construct a zero cost mapping with both connectors leaving the component. In that case, the pixel at the position marked '?' either has a conflict (that is, introduces a cost greater than zero in the criterion function because of mapping mismatch) with the pixel above or to the right of it, if the same color is to be met and otherwise, a cost in the grayvalue mismatch term is introduced.

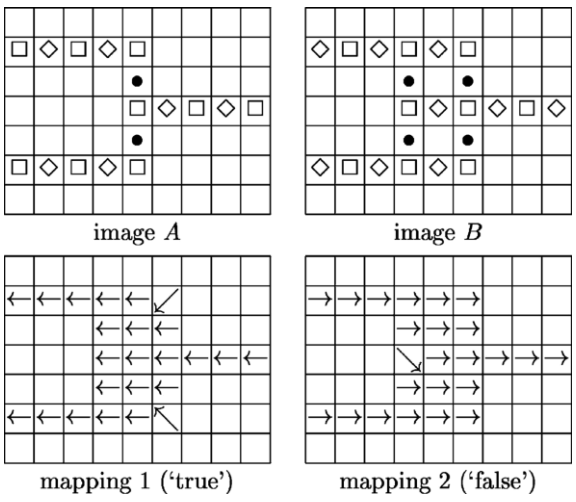


Fig. 3. The variable component with two positive and one negated output and two possible mappings (for true and false truth value).

Fig. 3 shows the variable component, in this case with two positive (to the left) and one negated output (to the right) leaving the component as connectors. Here, a fourth color is used, denoted by '.' This component has two possible mappings for the coloured pixels

with zero cost, which map the vertical component of the source image to the left or the right vertical component in the target image, respectively. (In both cases the second vertical element in the target image is not a target of the mapping.) This ensures 1 pixel relative displacements at the entry to the connectors. This property again can be deducted by regarding all possible mappings of the two images. The property that follows (which is necessary for the use as variable) is that all zero cost mappings ensure that all positive connectors carry the same truth value, which is the opposite of the truth value for all the negated connectors. It is easy to see from this example how variable components for arbitrary numbers of positive and negated outputs can be constructed.

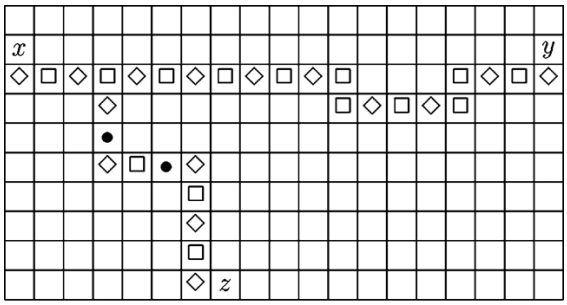


image A

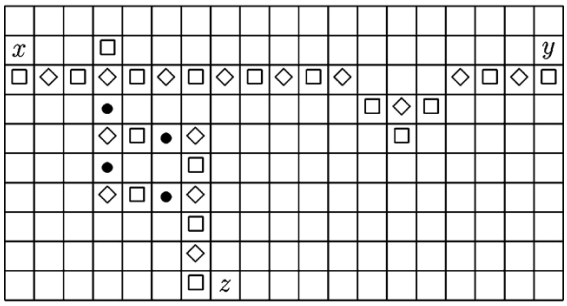
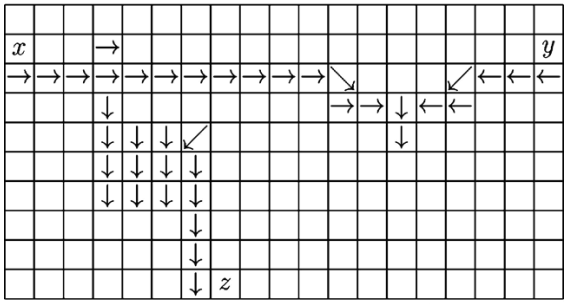
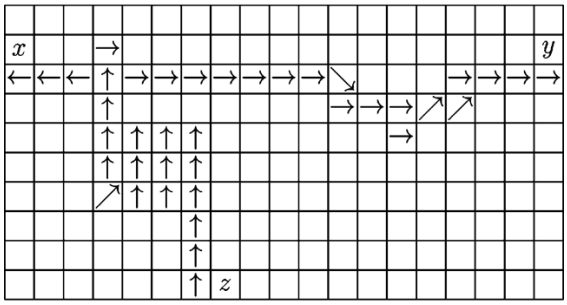


image B



mapping 1 (true, true, false)



mapping 2 (false, false, true)

Fig. 4. The clause component with three incoming connectors x, y, z and zero cost mappings for the two cases (true, true, false) and (false, false, true).

Fig. 4 shows the most complex of the components, the clause component. This component consists of two parts. The first part is the horizontal connector with a bend in it to the right. This part has the property that cost zero mappings are possible for all truth values of x and y with the exception of two false values. This two input disjunction can be extended to a three input dis-junction using the part in the lower left. If the z connector carries a false truth value, this part can only be mapped one pixel downwards at zero cost. In that case the junction pixel (the fourth pixel in the third row) cannot be mapped upwards at zero cost and the two input clause behaves as de-scribed above. On the other hand, if the z connector carries a true truth value, this part can only be mapped one pixel upwards at zero cost, and the junction pixel can be mapped upwards, thus allowing both x and y to carry a false truth value in a zero cost mapping. Thus there exists a zero cost mapping of the clause component iff at least one of the input connectors carries a true truth value.

The described components are already sufficient to prove NP-completeness by reduction from planar 3-SAT (which is an NP-complete sub-problem of 3-SAT where the additional constraints on the instances is that the dependency graph is planar), but in order to derive a reduction from 3-SAT, we also include the possibility of crossing connectors.

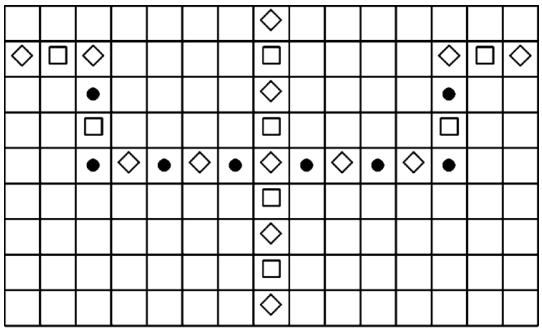


image A

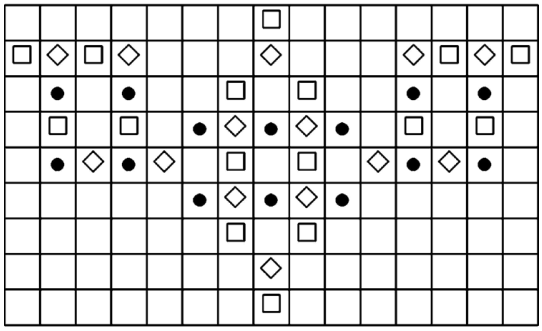
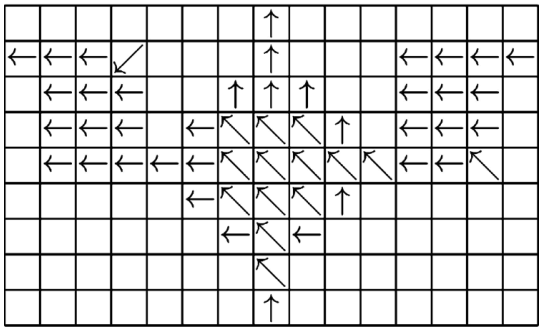


image B



zero cost mapping

Fig. 5. The connector crossing component and one zero cost mapping.

Fig. 5 shows the connector crossing, whose basic property is to allow zero cost mappings if the truth–values are consistently propagated. This is assured by a color change of the vertical connector and a flexible middle part, which can be mapped to four different positions depending on the truth value distribution.

Reduction:

Using the previously introduced components, we can now perform the reduction from 3-SAT to image matching.

Proof of the claim that the image matching problem is NP-complete:

Clearly, the image matching problem is in NP since, given a mapping f and two images A and B, the computation of c(A, B, f) can be done in polynomial time. To prove NP-hardness, we construct a reduction from the 3-SAT problem. Given an instance of 3-SAT we construct two images A and B, for which a mapping of cost zero exists iff all the clauses can be satisfied.

Given the dependency graph D, we construct an embedding of the graph into a 2D pixel grid, placing the vertices on a large enough distance from each other (say $100(K+L)^2$). From this image of the graph D we construct the two images A and B, using the components described above. Each vertex belonging to a variable is replaced with the respective parts of the variable component, having a number of leaving connectors equal to the number of incident edges under consideration of the positive or negative use in the respective clause. Each vertex belonging to a clause is replaced by the respective clause component, and each crossing of edges is replaced by the respective crossing component. Finally, all the edges are replaced with connectors and corner connectors, and the remaining pixels inside the rectangular hull of the construction are set to the background grayvalue. Clearly, the placement of the components can be done in such a way that all the components are at a large enough distance from each other, where the background pixels act as an insulation against mapping of pixels, which do not belong to the same component. It can be easily seen, that the size of the constructed images is polynomial with respect to the number of vertices and edges of D and thus polynomial in the size of the instance of 3-SAT, at most in the order of $(K+L)^2$. Furthermore, it can obviously be constructed in polynomial time, as the corresponding graph drawing algorithms are polynomial.

Let there exist a truth assignment to the variables $x_1 , \ldots , x_L$, which satisfies all the clauses $c_1 , \ldots , c_K$. We construct a mapping f , that satisfies c(A, B, f) = 0 as follows.

For all pixels (i, j) belonging to variable component l with A(i,j) not of the background color, set f(i,j)=(i,j-1) if x_l is assigned the truth value ‘true’, set f(i,j)=(i,j+1), otherwise. For the remaining pixels of the variable component set f(i,j)=(i,j) if A(i,j)=B(I,j), otherwise choose f(i,j) from {(i,j+1), (i+1,j+1), (i-1,j+1) } for x_l ‘false’ respectively from {(i,j-1), (i+1,j-1), (i-1,j-1) } for x_l ‘true’, such that A(i,j)= B(f(i,j)).This assignment is always possible and has zero cost, as can be easily verified.

For the pixels (i,j) belonging to (corner) connector components, the mapping function can only be extended in one way without the introduction of nonzero cost, starting from the connection with the variable component. This is ensured by the basic connector property. By choosing f(i,j)=(i,j) for all pixels of background color, we obtain a valid extension for the connectors. For the connector crossing components the extension is straightforward, although here—as in the variable mapping— some care must be taken with the assignment of the background value pixels, but a zero cost assignment is always possible using the same scheme as presented for the variable mapping.

It remains to be shown that the clause components can be mapped at zero cost, if at least one of the input connectors x, y, z carries a 'true' truth value. For a proof we regard all seven possibilities and construct a mapping for each case. In the description of the clause component it was already argued that this is possible, and due to space limitations we omit the formalization of the argument here.

Finally, for all the pixels (i,j) not belonging to any of the components, we set f(i,j)=(i,j), thus arriving at a mapping function which has c(f, A, B)=0, as all colors are preserved in the mapping and no distortion of squared Euclidean distance greater than 2 is introduced.

On the other hand, let there exist a mapping f with c(f, A, B)<=0. This means that c(f, A, B)=0 since there are only positive cost terms involved in the term for c. Furthermore, we can conclude, that

$\forall(i,j)\in \hat{M}x\hat{M}\ d_g(A_{ij}, B_{f(i,j)}=0,$

$\forall(i,j)\in M\{1,\ldots,M-1\}x\hat{M}\ d_d(f((i,j)+(1,0))-(f(i,j)+(1,0)))=0,$

$\forall(i,j)\in \hat{M}xM\{1,\ldots,M-1\}\ d_d(f((i,j)+(0,1))-(f(i,j)+(0,1)))=0$

This means that f maps all pixels onto pixels of the same color and that the difference in mapping between neighboring pixels has at most squared Euclidean distance 2.

These facts now ensure a number of basic properties:

(1) The basic elements of the components and the overall represented graph are mapped to their corresponding parts, because of the monotonicity and continuity assured by the maximum local mapping difference.

(2) All the variable components are mapped such that the leaving connectors all carry consistent truth values (i.e. all the positive outputs carry the same truth value, and all the negative outputs carry the negated truth value).

(3) All (corner, crossing, straight) connectors are mapped such that the basic connector property of propagated truth values is fulfilled, since no neighboring pixels are mapped into opposite directions.

(4) Each clause component has at least one entering connector carrying a truetruth value, as otherwise a mapping with zero cost is not possible.

Now we construct an assignment of truth values to the variables $x_1,\ldots,x_L$, which satisfies all the clauses $c_1,\ldots,c_K$. We set x_l to 'true', if f(i,j)=(i,j+1) for the pixels of the corresponding variable component, which are not background pixels. We set x_l to 'false', otherwise. By means of the properties stated above we can conclude that this assignment satisfies all the clauses $c_1,\ldots,c_K$, which concludes the proof.

Annexure C

Laboratory Assignments on Project Quality and Reliability Testing of Project Design

C1. Object Oriented Analysis

C1.1 Objects/Entities:

Image, Predictor, Facial Recognition System

C1.2 Relations:

Image given to predictor, Facial recognition system recognizes face in image, Predictor estimates quality of image, Predictor decides which images to send to FR system

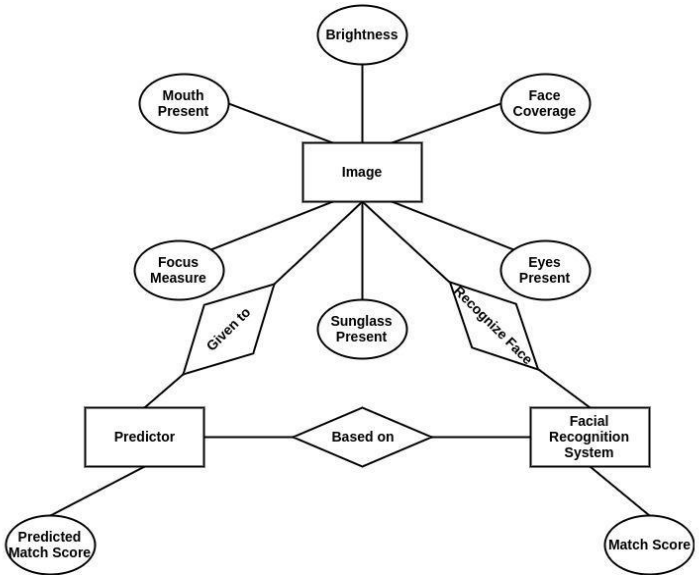


Fig C1 ER Diagram

C1.3 Functional Relation

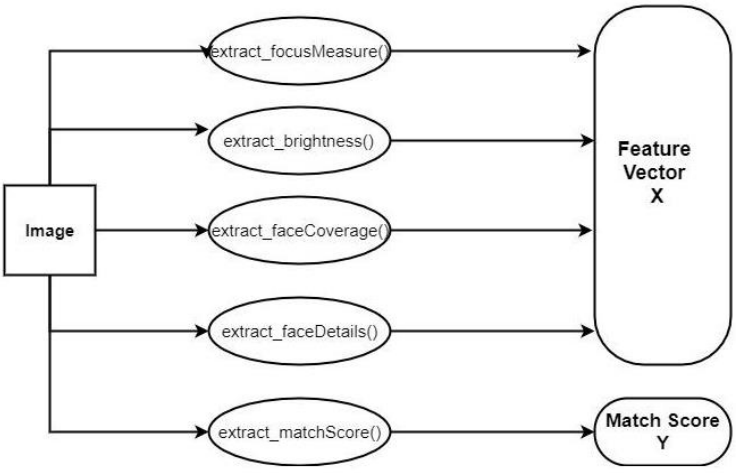


Fig C2 Functional Relations

C2. Functional Dependency Graph

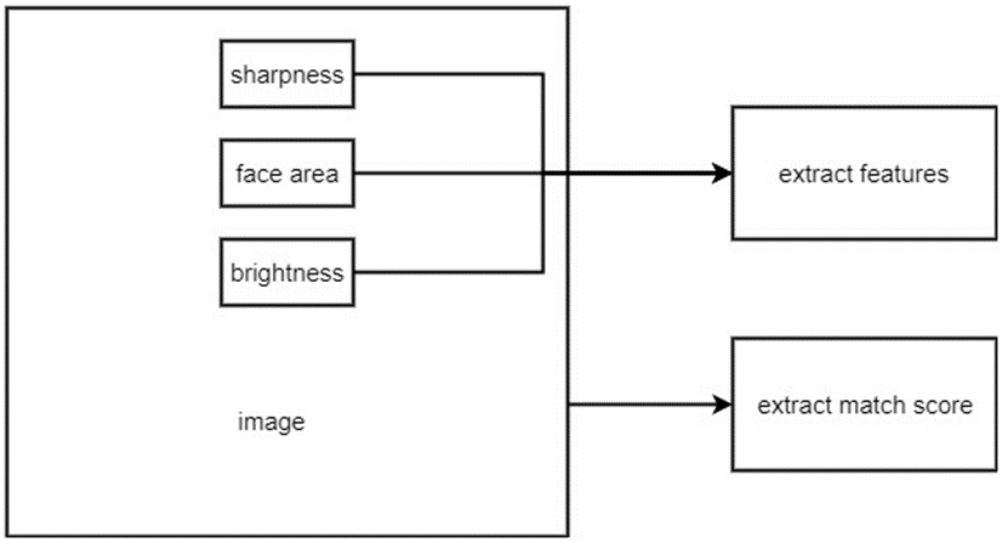


Fig C3 Functional Dependency Graph

C3. UML Diagrams

C3.1 Use Case Diagram

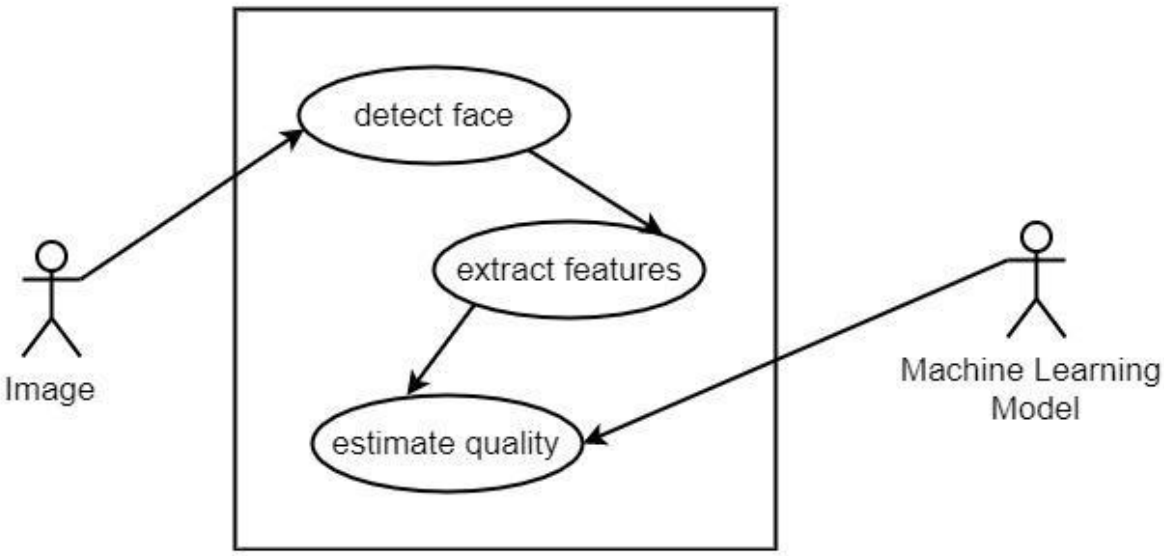


Fig C4 Use Case Diagram

C3.2 Activity Diagram

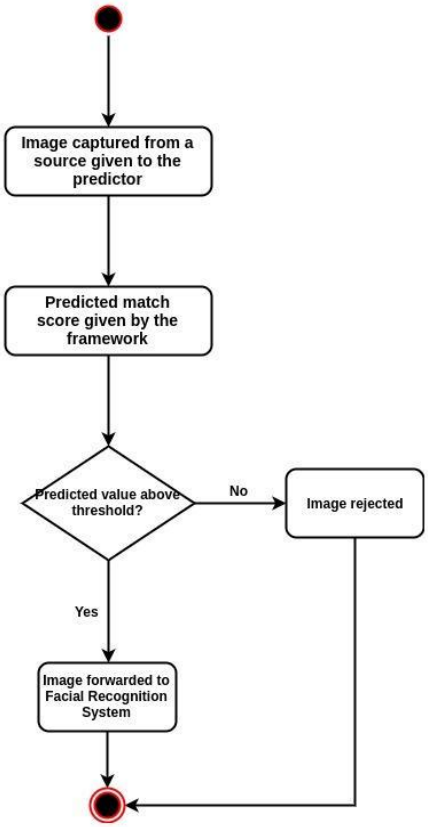
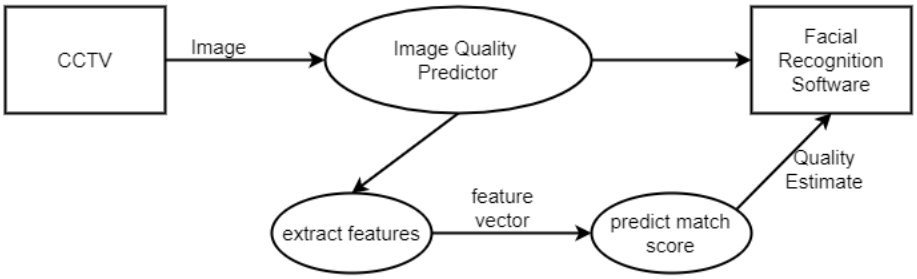


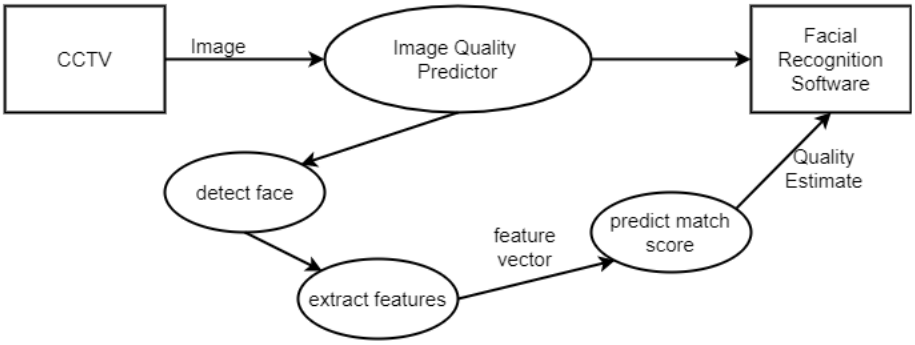
Fig C5 Activity Diagram

C3.3 Data Flow Diagram

Level 0 Data Flow Diagram



Level 1 Data Flow Diagram



Level 2 Data Flow Diagram

Fig C6 Data Flow Diagram

C4. Testing

C4.1 Black Box Testing

Sr. No.	Test Case	Description	Expected Output
1	Image with match score greater than threshold is given to the framework	Framework should predict class of image	Image should be discarded
2	Image not containing a face is given as input	Framework should predict class of image	Image should not be passed to the matcher for further processing

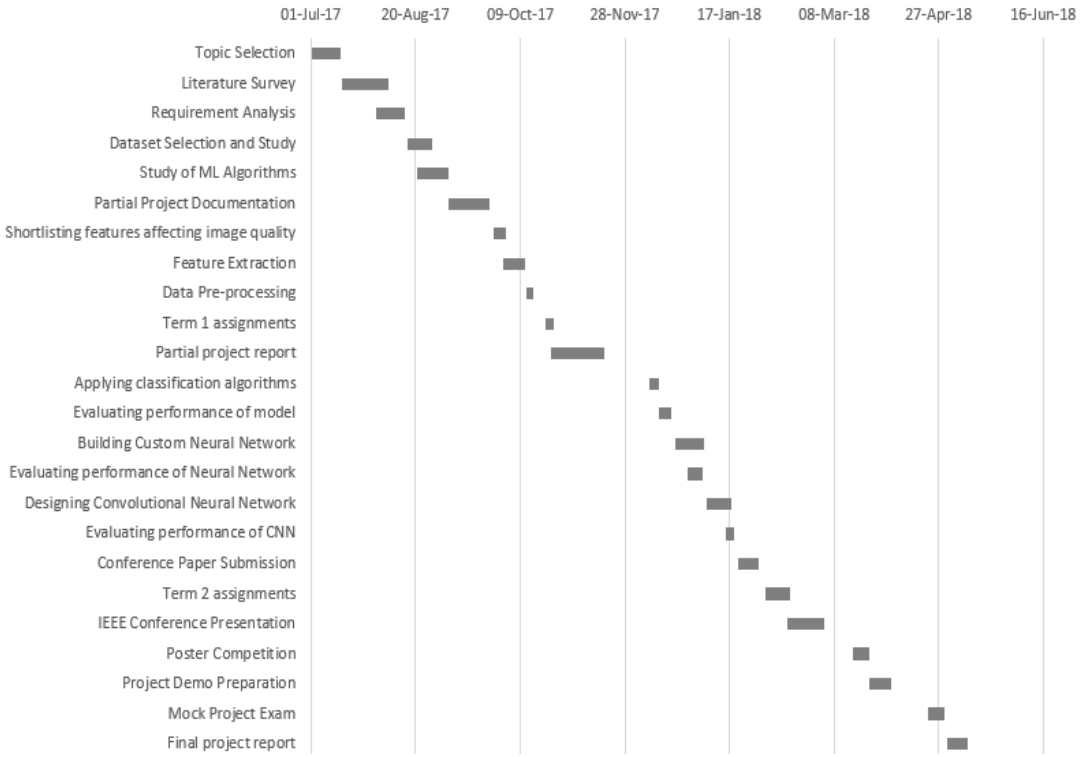
C4.2 White Box Testing

Sr. No.	Test Case	Description	Expected Output
1	ROC Curves	Graphical plot that illustrates the diagnostic ability of a binary classifier system	Monotonically increasing, start at (0,0)(0,0) and arrive at (1,1)(1,1)
2	F1-Score	F1-score is the harmonic average of the precision and recall. F1-Score lies between 0 and 1.	F1-Score should be closer to 1.
3	Accuracy	Accuracy is the fraction of predictions our model got right.	Accuracy should be closer to 100%.
4	Loss	Loss value implies how well a certain model behaves after each iteration of optimization.	Reduction of loss after each, or several, iteration(s).

Annexure D

Project Planner

D1. Gantt Chart



D2. Tasks List

Task	Start Date	End Date	Duration
Topic Selection	01-Jul-17	15-Jul-17	14
Literature Survey	16-Jul-17	07-Aug-17	22
Requirement Analysis	01-Aug-17	15-Aug-17	14
Dataset Selection and Study	16-Aug-17	28-Aug-17	12
Study of ML Algorithms	21-Aug-17	05-Sep-17	15
Partial Project Documentation	05-Sep-17	24-Sep-17	19
Shortlisting features affecting image quality	26-Sep-17	02-Oct-17	6
Feature Extraction	01-Oct-17	11-Oct-17	10
Data Pre-processing	12-Oct-17	15-Oct-17	3
Term 1 assignments	21-Oct-17	25-Oct-17	4
Partial project report	24-Oct-17	18-Nov-17	25
Applying classification algorithms	10-Dec-17	14-Dec-17	4
Evaluating performance of model	14-Dec-17	20-Dec-17	6
Building Custom Neural Network	22-Dec-17	05-Jan-18	14
Evaluating performance of Neural Network	28-Dec-17	05-Jan-18	7
Designing Convolutional Neural Network	06-Jan-18	18-Jan-18	12
Evaluating performance of CNN	15-Jan-18	20-Jan-18	4
Conference Paper Submission	21-Jan-18	31-Jan-18	10
Term 2 assignments	03-Feb-18	15-Feb-18	12
IEEE Conference Presentation	14-Feb-18	03-Mar-18	17
Poster Competition	17-Mar-18	25-Mar-18	8
Project Demo Preparation	25-Mar-18	05-Apr-18	10
Mock Project Exam	22-Apr-18	30-Apr-18	8
Final project report	01-May-18	11-May-18	10

Annexure E

Reviewers Comments of Paper Submitted

- **Paper Title**: Towards Designing an Adaptive Framework for Facial Image Quality Estimation at Edge
- **Conference**: IEEE International Conference for Convergence of Technology (I2CT), Pune-2018
- Paper accepted and published in Institute of Electrical and Electronics Engineers (IEEE) Xplore
- **Corrective actions** (if any): None

Annexure F

Plagiarism Report

Plagiarism Scan Report

Summary	
Report Genrated Date	02 May, 2018
Plagiarism Status	**95% Unique**
Total Words	1000
Total Characters	6323
Any Ignore Url Used	

Content Checked For Plagiarism:

The proposed project model comes under the Waterfall Model category. It undergoes several phases starting from Requirement Analysis till its Deployment and Maintenance. We maped our estimates with the steps in the waterfall model. Below are some estimates

Introduction
Our framework is basically a two class classification model. In order to build the dataset for classification we used OpenFace and ChokePoint dataset.

OpenFace: It is a general purpose open source matching library. It is implemented using python and torch and has a deep neural network. Working:
Detect faces through a pretrained model.
Transform face: Position the eyes and lips for all images at same location.
Using DNL represent or (embed) face in a 128-dimensional unit hypersphere.
Larger distance between embeddings denotes that images are likely not of the same person.
This makes clustering, classification easier where euclidean distance between features is not meaningful.
Using the “compare” API of OpenFace, a matchscore can be generated between two images. This matchscore ranges between 0 to 4 where 0 indicates best match and 4 indicates a bad match or a no match.

The experiment was done on 29 subjects and a total of 64,204 images were captured. Since our problem is a 2 class classification problem, we used a balanced dataset of 29,022.

To get this balanced dataset, we obtained the matchscores of all images using openface. To do so the first step was to find the best images for every subject which was considered as the best image. All other images of a subject were compared to this reference image to get the matchscore. These are the genuine matchscore since we compared the images of same subject. We found equal number of imposter scores as well. As shown in the distribution, the threshold is 0.84 where FAR is 0.09 and FFR 0.24. The threshold indicates that images with matchscore below 0.84 are labelled as ‘good’ images and those having a score above it are ‘bad’ images. This is how we created a balanced dataset.

We propose a framework which will run before the actual facial recognition, using which only the good quality images will be forwarded for FR. Such a framework will help increase the real time applicability of FR since it has to process only on good quality images for which it can definitely produce a match. And this system/framework will work in line with the principles of edge computing which suggests that maximum processing should be done at the source of data, which here are CCTVs. In order to implement such a framework we took various approaches.

Feature Extraction: We studied the behaviour of OpenFace by obtaining match scores of many different facial images in order to determine which factors potentially alter match scores. We observed that many image quality metrics such as brightness, contrast, sharpness directly affected match score. Apart from these we also observed that the area covered by a face in an image also influenced match score. Lastly, obstruction of face by sunglasses, eyeglasses, caps, facial hair, different hairstyles, all impacted the match score. So these features were extracted and a feature vector of such features was created.
Machine Learning Approach: After completing the data pre-processing and standardization, we divided the dataset into 70-30 ratio for training and testing of the classification model respectively. In the training phase, the model was built using the feature vector as the independent variable of classification and the corresponding quality as the class labels. To build the model we used various classification algorithms.The results are shown
We also built the model on a custom Neural Network with which we could achieve a highest accuracy of 67.74%. The model has two densely connected hidden layers and was trained for 1000 epochs. The model summary and result,

Deep Learning Approach: We used a convolutional neural network to build the framework. We based the architecture of our convolutional neural network on AlexNet. We modified the output layer to implement binary classification.

The input shape to our CNN was a single channel since the dataset contains grayscale images. The input image data to the network was pre-processed by performing mean image subtraction and normalization of pixel values from [0, 255] into [0, 1] range. Normalization refers to normalizing the data dimensions so that they are approximately of the same scale. The CNN was trained for 130 epochs, at which point the accuracy was constant. The accuracy of the deep learning framework is 92.274%
Verification and Validation for Acceptance

7 Detailed Design Document using Appendix A & B
7. 1 Introduction
This document specifies the design that is used to solve the problem of Product. This document also gives the detailed view of architecture of the device.
7.2 Architectural Design
The program architecture can be considered as a 2-tier architecture. The CCTVs form the 1st Tier which process the images. The framework runs on the CCTVs itself and makes a decision regarding which images are to be forwarded to the facial recognition software which resides in the 2nd Tier.
This section contains a description of all data structures including internal, global, and temporary data structures, database design (tables), file formats.
5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk ID
1
Risk Description
Damage to surveilling entity
Category
External Factors
Source
Software requirement Specification document.
Probability
Low
Impact
High
Response
Replace damaged entity as soon as possible
Strategy
Keep tabs from time to time on working of surveilling entity
Risk Status
Identified

Risk ID
2
Risk Description
Failure of server - client connectivity
Category
Link Failure
Source
This was identified during early development and testing
Probability
Low
Impact
Medium
Response
Keep images in a buffer till connection is re-established
Strategy
Before discarding image from cache, it must be checked whether it is successfully forwarded to server
Risk Status
Identified

Risk ID
3
Risk Description
Sudden increase in image load leading to loss of data
Category
Implementation
Source
This was identified during testing.
Probability
Low
Impact
Low
Response
Provide additional buffer space to save images temporarily
Strategy

Annexure G

Term-II Project Laboratory Assignments

G1. Review of Design and Necessary Corrective Actions Taking Into Consideration the Feedback Report of Term Assessment

Initial Work

We used machine learning classification algorithms like Gaussian Naïve Bayes, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbour to estimate facial image quality and decide whether to forward an image to a CCTV or not. We got a maximum accuracy of 68% by using decision tree classification algorithm.

Reviewer's Comments

According to the reviewer's an accuracy of 68% was not enough for our project as its main application lies in the security domain. The reviewers suggested that we try to improve the accuracy by exploring other approaches.

Changes Incorporated in Work

After reviewer's feedback in Term-1 and with inputs from our project guide and mentor at Persistent Systems Ltd., we decided to try other approaches to increase accuracy. We built a neural network with two densely connected hidden layer and trained it on the extracted features of the chokepoint dataset for 1000 epochs. We got an accuracy of 67.74% only. Then we used the deep learning approach to estimate the image quality. We built a custom neural network based on the architecture of Alexnet by modifying the output layer to implement binary classification. We gave the pre-processed images as an input to the CNN and trained it for 130 epochs to get an accuracy of 92.274%.

G2. Project Workstation Selection, Installation Along with Setup

The user should install all the dependencies needed. The dependencies needed are as follows:

- Anaconda
  bash Anaconda3-5.0.1-Linux-x86_64.sh
  source ~/.bashrc
- Ipython Notebook
  Sudo apt-get install ipython-notebook
- Tenserflow
  conda create -n tensorflow pip python=2.7
  source activate tensorflow
- Keras
  pip install keras
- Numpy
  sudo apt-get install python-numpy
- H5py
  sudo pip install h5py
- OpenCV
  sudo apt-get install python-opencv

For building the machine learning and deep learning models installations needed are as follows:

- Openface

The dependencies needed for installation of openface are:

  - gcc
  - cmake
  - OpenCV
  - boost

- Scikit-learn
  pip install -U scikit-learn
- Matplotlib
  python -mpip install -U matplotlib
- Graphviz
  sudo apt-get install graphviz
- Seaborn
  pip install seaborn
- Amazon Web Services

G3. Testing

Black Box Testing

Sr. No.	Test Case	Description	Expected Output	Actual Output
1	Image with match score greater than threshold is given to the framework	Framework should predict class of image	Image should be discarded	Image not passed to the matcher for further processing
2	Image not containing a face is given as input	Framework should predict class of image	Image should not be passed to the matcher for further processing	Image pruned at edge

White Box Testing

Sr. No.	Test Case	Description	Expected Output	Actual Output
1	ROC Curves	Graphical plot that illustrates the diagnostic ability of a binary classifier system	Monotonically increasing, start at (0,0)(0,0) and arrive at (1,1)(1,1)	Monotonically increasing, start at (0,0)(0,0) and arrive at (1,1)(1,1)
2	F1-Score	F1-score is the harmonic average of the precision and recall. F1-Score lies between 0 and 1.	F1-Score should be closer to 1.	F1-Score ranging from 0.62 to 0.68
3	Accuracy	Accuracy is the fraction of predictions our model got right.	Accuracy should be closer to 100%.	Accuracy of 92.27% for CNN.
4	Loss	Loss value implies how well a certain model behaves after each iteration of optimization.	Reduction of loss after each, or several, iteration(s).	Reduction of loss after each, or several, iteration(s).

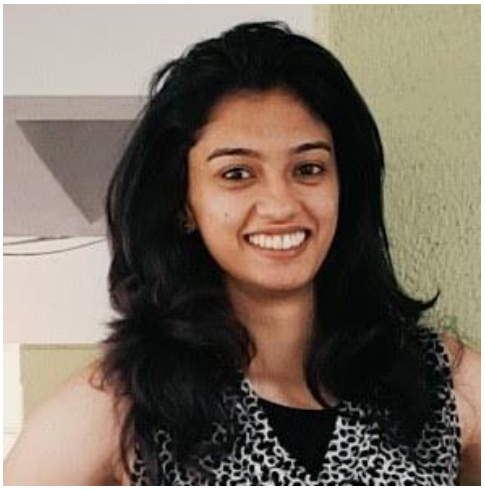
Annexure H

Information of Project Group Members

1. Name : Aditya Deshpande

- Date of Birth :08/12/1996
- Gender : Male
- Permanent Address : B-1/101, Shivsagar City, Sun City Road, Anandnagar, Sinhagad Road, Pune 411051
- E-Mail : adityamd812@gmail.com
- Mobile/Contact No. : 8600997096
- Placement Details : Mu Sigma
- Paper Published : Yes(IEEE I2CT)

2. Name : Alisha Shahane



- Date of Birth :05/09/1996
- Gender : Female
- Permanent Address : Plot No:53, Pramathesh Society, behind Mahatma Society, Kothrud, Pune 38
- E-Mail : alisha.shahane@gmail.com
- Mobile/Contact No. : 9689930573
- Placement Details : Mu Sigma
- Paper Published : Yes(IEEE I2CT)

3. Name : Darshana Gadre

- Date of Birth :24/10/1996
- Gender : Female
- Permanent Address : Flat No. 16, ’20 Oaks’, Mayur Colony, Kothrud, Pune-411038
- E-Mail : darshana.gadre@gmail.com
- Mobile/Contact No. : 9604535131
- Placement Details : Softlink International
- Paper Published : Yes(IEEE I2CT)

4. Name : Mrunmayi Deshpande

- Date of Birth :26/01/1996
- Gender : Female
- Permanent Address : 3/10 Tara Residency Kothrud Pune-411038
- E-Mail : mrunmayi96@gmail.com
- Mobile/Contact No. : 9552012300
- Placement Details : EQ Technologic
- Paper Published : Yes(IEEE I2CT)