Software Design Document for Face Recognition Based Attendance System

Himansh Sheoran, Maanik Arora, Nausheen Khan
Madhvi Panchal, Meenu Menon, Abhishek Trivedi
Supervised by: Prof Venkataramanan Subramanian

April 7, 2020

1 Introduction

1.1 Purpose

The main purpose of this project is to develop a **Face Recognition Based Automated Student Attendance System**. We will be building a system that will be able to detect and recognize frontal faces of students in classroom, and then mark their respective attendance. The system is exclusively designed for marking attendance in various educational institutes like in colleges and schools. The aim of this project is to eliminate the traditional system of **Manual Attendance System** which is usually a time consuming process with chances of human error and replace it with a **Automated Attendance System**.

1.2 Scope

The scope of the project is to design an automated system which will be able to find and recognize human faces in a fast and precise manner from images and videos that are captured through a surveillance camera and subsequently mark their attendance.
The main objectives of the project are

- To detect and precisely locate the face segments in a video frame.
- To count the face instances in a frame for crowd/students numbering.
- To extract the useful invariant features of the detected face.

Figure 1: Block Diagram of the General Framework

- To classify the features for subsequent face recognition through feature matching.
- To record the attendance of the identified student.

1.3 Overview

As we know that technology aims to impart tremendous technical innovations these days. Machine Learning, Computer Vision, Deep Learning have gained prominence in the recent years which has led to solving many problems like Image Classification, Object Detection, Image Segmentation and so on. From the very beginning, attendance is considered as a crucial component in education system which makes it a target area to bring some kind of automated innovation. This can help discard the traditional tiresome physical methods of marking attendance in many oriental institutions.
This project therefore hopes to deploy an easy and a secure system capable of marking attendance of students entering a classroom, by using Computer Vision and Deep Learning based face recognition techniques. The system is going to include some data and model paralleling/distribution techniques for fast computation and detection of faces.

2 System Overview

The two main building blocks of this system are **Face Detection** and **Face Recognition**.

2.0.1 Face Detection

Face detection is a computer technology being used in a variety of applications and algorithms that detects and locates human faces in digital images. It is a subset problem of challenging object detection tasks in computer vision. The extended task in the proposed system is to detect and locate all faces present in an image with the notion of their different sizes. The task is a bit challenging because of the variability present across human faces such as difference in lighting, facial expressions, skin tone, image resolution, facial position, facial hair etc.

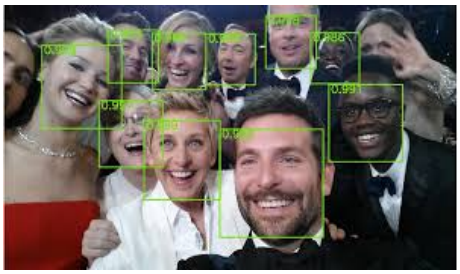

Figure 2: Automatic face detection with OpenCV

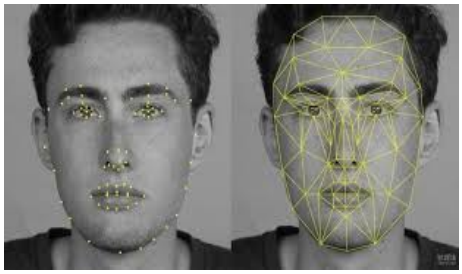

Figure 3: Automatic face detection with OpenCV

2.0.2 Face Recognition

A facial recognition system is a technology capable of identifying a person present in an image or video. There are dedicated algorithms for the task where the general idea is selection of various face features from a given input image and comparing it with the features of image present in the existing database. Our subsystem is a part of recent trend of Biometric Artificial Intelligence based applications that can uniquely identify a person by analyzing patterns based on the person's facial textures and shape.

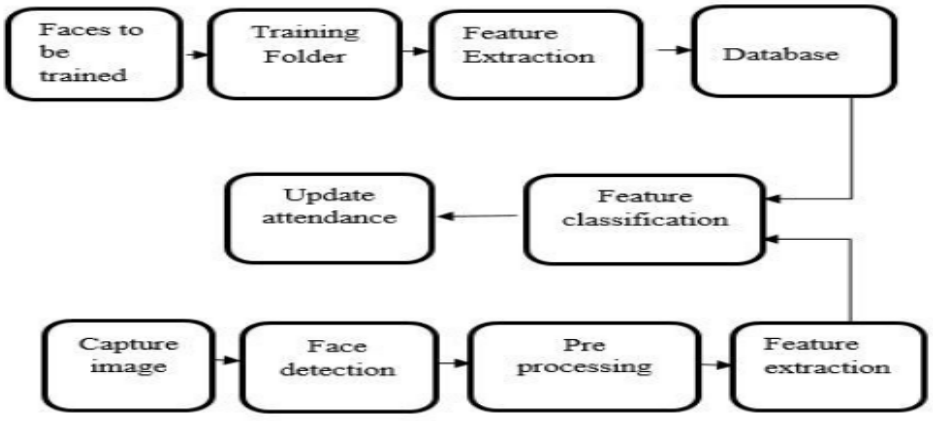

Figure 4: Face Recognition Block Diagram

3 System Architecture

3.1 Architectural Design

Our automated Attendance system is composed of different modules for different sub-tasks which bring together the main objective of fast and reliable attendance of students in a sequential manner.

- The image captured through the video camera is passed through the face detection module. This module has two sub-modules working in a coordinated manner to precisely locate the faces. The face detector is trained on images from face database to predict the bounding boxes around faces for further downstream processes.
- We are also preparing a system of eye database and eye localizer. The presented module helps in feature extraction of different facial structures by computing absolute location of eyes and relative facial positions of other sense organs.
- The facial feature extraction module helps in robust features extraction from images of different subjects/students. The module eventually builds a strong base for feature matching between input and target images.
- The face recognition module is composed of face recognizer and face database containing information of students. The recognizer sub-module is trained on face database to classify the input image to a target permissible student id in database through feature matching. The outcome is Recognized **Face Id** or **Reject** alert.

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don t go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

3.2 Decomposition Description

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object oriented description. For a functional description, put top level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

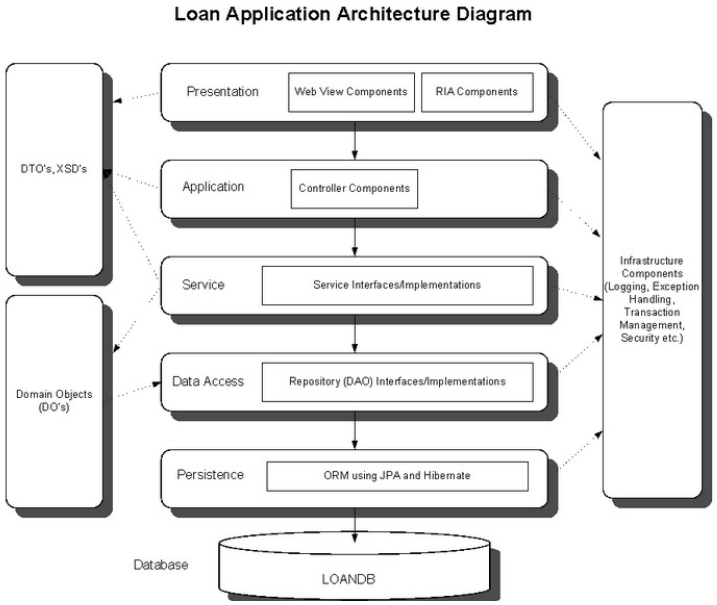

Figure 5: Architectural Design

3.3 Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

4 Data Design

4.1 Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

4.2 Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

5 Component Design

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudo-code. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudo code. Describe any local data when necessary.

6 Human Interface Design

6.1 Overview of User Interface

Describe the functionality of the system from the user s perspective. Explain how the user will be able to use your system to complete al the expected features and the feedback information that will be displayed for the user.

6.2 Screen Images

Display screenshots showing the interface from the users perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible.

6.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

7 Requirements Matrix

Provide a cross reference that traces components and data structures to the requirements in your SRS document. Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

8 APPENDICES

This section is optional. Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

9 References

References