

Automatic attendance system using Facial Recognition

Team-15

Aayush Upadhyay, Aditya Agarwal, Aniruddha Deshpande,
Jeevesh Kataria , P.V. Haridatta , V.R.N Sricharan Priyatham

1 Abstract

The project deals with the automatic face recognition attendance system. The camera is placed outside the classroom. For every person who enters the class, the camera captures the image and sends the image to the verification server, where the facial features are extracted from the sent image and will be passed to the trained model, which can identify the person and mark the attendance. In addition to attendance this model also detects drowsiness of the student and gives a little penalty in attendance accordingly.

2 Solution Overview

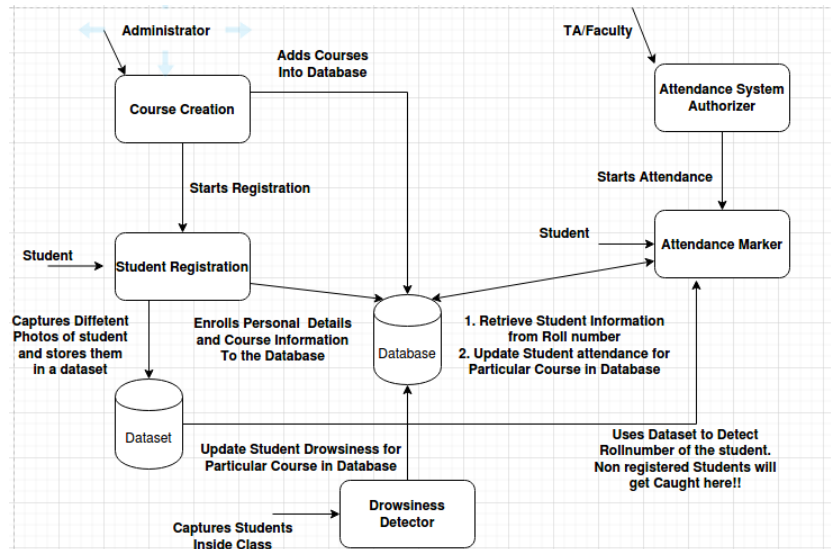


Figure 1: Solution Overview

As shown in the above figure, Process starts with admin adding new courses into the database. admin also sets a passwords for each course to concerned faculty or TA. These courses are then offered to students at the time of registration. Students will then provide there credentials and photos to register to this system. At the starting of class TA/Faculty will start attendance process by providing correct Course ID and Pass Key pair. The attendance marker which

had a model trained during registration, will then detect the corresponding student from camera and mark attendance. Also, there is a drowsiness detection module which detects the sleeping student(s) inside the class and gives penalty in attendance accordingly.

3 Description of Modules

3.1 Course Creation

This module can be used by the administrative department for adding the Courses that Students can take in that particular semester to the database.

3.2 Student Registration

It helps students to register to the courses offered by the institute. The students should provide relevant details required for the registration. Registration mainly includes the following phases

- Storing personal details about the students in a database.
- Storing the subjects chosen by the student in a database
- Storing the different images corresponding to student in a disk under a folder named with their corresponding roll number.

3.3 Authorization of Attendance System

The Cameras are configured in such a way that,they will collect details(Course Id and passwords) to start the attendance,so that only the authorized people can start the attendance. The camera collects these details and forwards the same to verification server, which stores hashed passwords against each course ID created. After verifying credentials camera will start functioning.

3.4 Marking of Attendance

As show in the figure2, Face Detector will detect any face in the input video. It includes Face localization, which detects where the faces are located in the input video. Feature Extractor will extract important features using Face-Net(a pre-trained CNN with triplet loss function over millions of images). The extracted features are then given to a classification model based on Random Forest which is trained on dataset created during registration which matches a face based on a trained data set of pictures. Then the probability for the captured image being that particular person on the data set is calculated and fed to Face recognition model which if identified finally returns the unique roll number(Assigned during the time of registration).

3.5 Drowsiness Detection

The module takes responsibility of identifying Students found Sleepy in the class.To implement this,the Cameras are placed inside the Classroom, which will continuously take the snapshots of areas they are intended to cover and

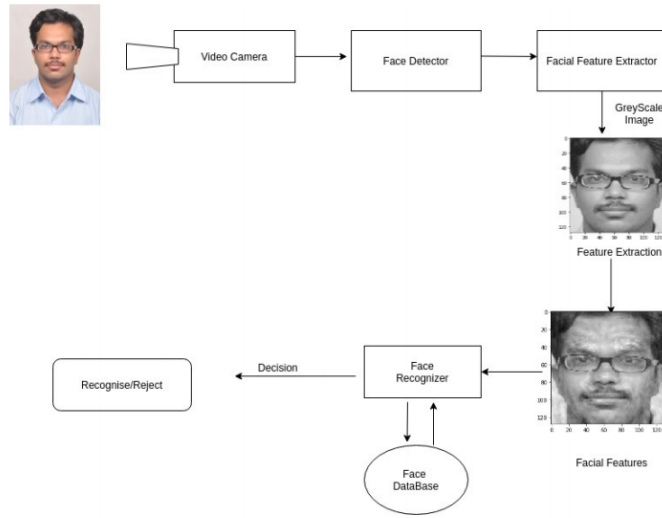


Figure 2: Face Recognition System Framework

send to the servers for processing further. The server iterating through all the frames sent, will capture the faces of Students in it. Once the faces are captured, it detects whether the person is feeling drowsy or not. If it found that the person is feeling drowsy for some (configurable) continuous frames, it will identify the person with his/her facial features and will make an entry in the database regarding this and the students will get penalized in their attendance.

4 Data Flow

We collect the photographs of students with several orientations and emotions to make our training model robust. We pre-process the images, by doing mean subtraction and scaling to deal with the illumination changes in the images we collected. We feed all the collected images to our model to learn, which can be finally used for recognition when facial features are given as input.

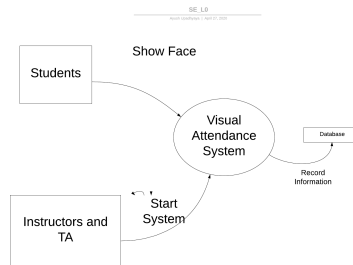


Figure 3: 0 Level DFD

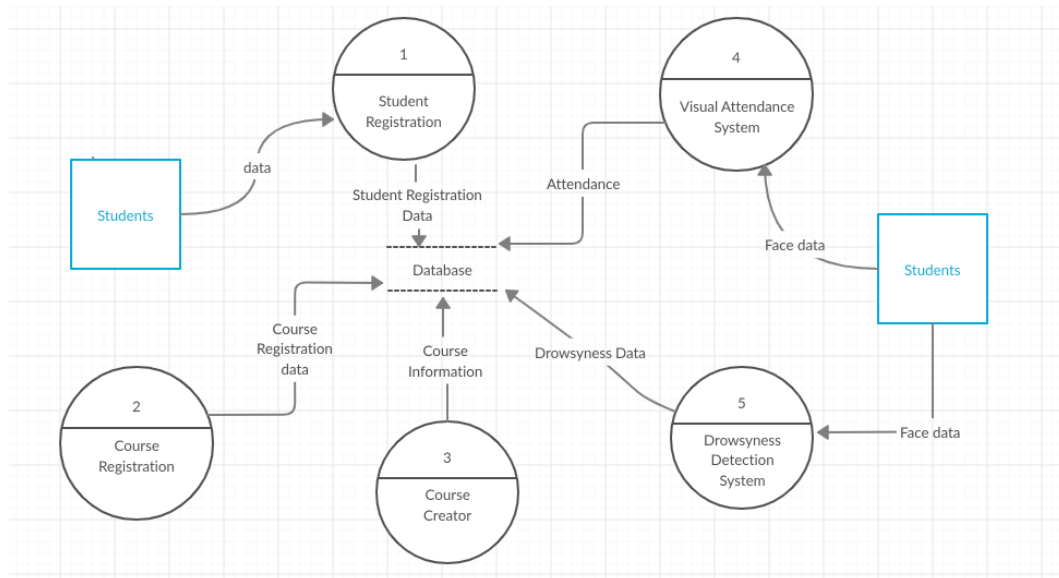


Figure 4: Level 1 DFD

4.1 Database Description

Our database consists of the following tables

- **students**(roll_number,full_name,created_at)-Maintains registration data of students.
- **courses**(course_id,course_name):Maintains information about courses offered by college.
- **course_reg**(course_id,roll_number,semester_type,semester_year): Maintains information about which students have registered for what courses in a particular semester.
- **attendance**(course_id,roll_number,Semester_year,Semester_type,Date_of_class,full_name,time_of_class,class_room):Maintains attendance record of students.
- **course_credentials**(course_id,pass_key):For each course that corresponds to a course id, a password in the form of SHA-1 is stored in this table.This password shall be known to instructors and TA of a course, who can start the attendance system.
- **drowsy_data**(course_id,roll_number,date_of_class):This maintains drowsiness data of students sleeping in a particular class.

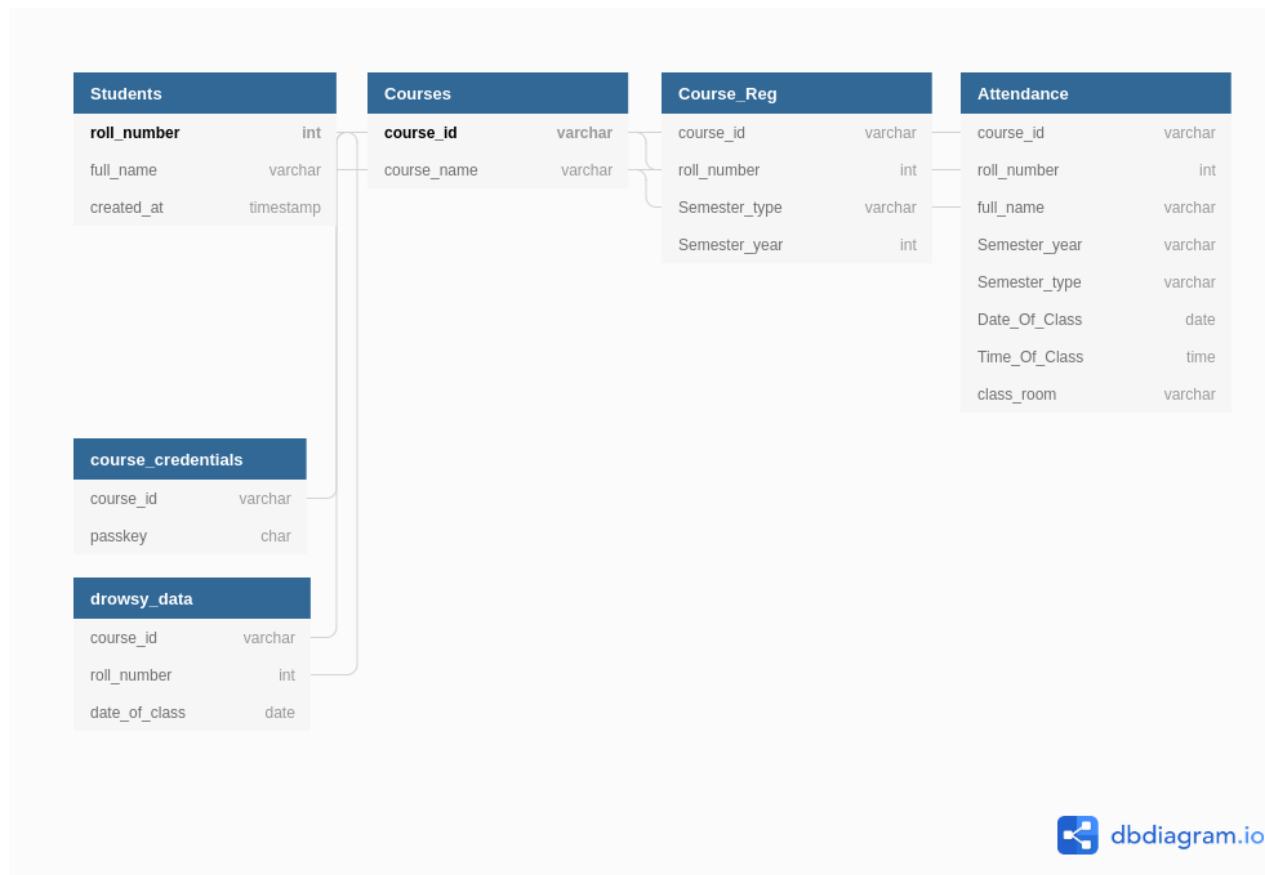


Figure 5: Database Layout

5 Process Flow for Attendance Recognition System:

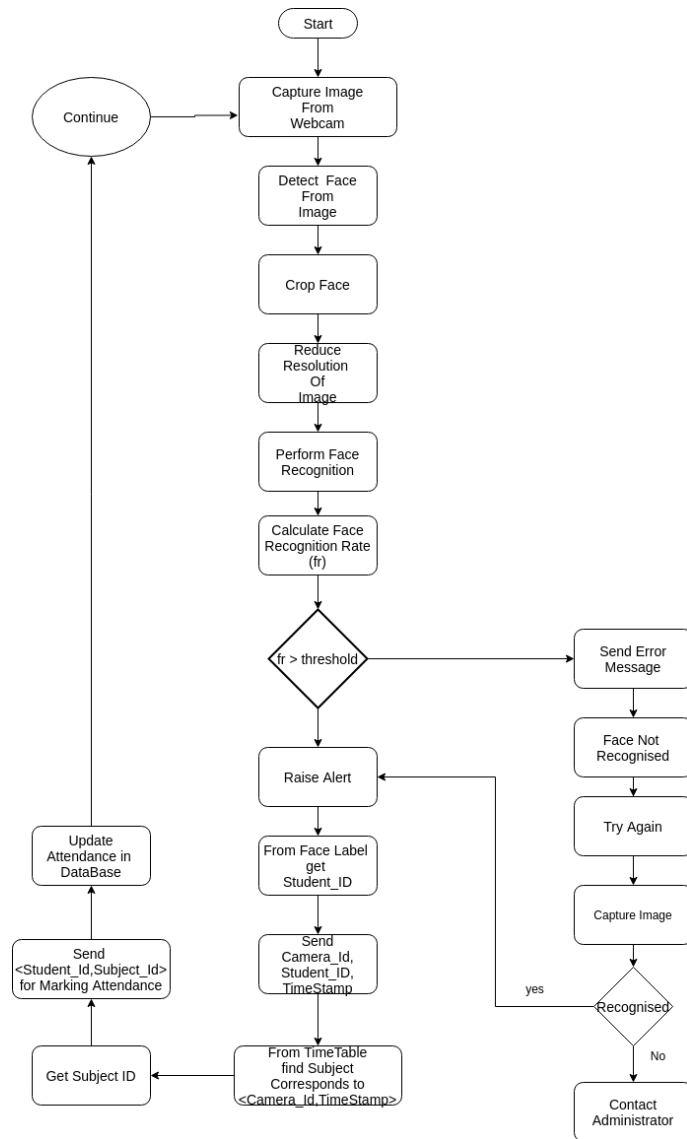


Figure 6: Testing Model

6 Innovation

6.1 Detecting Drowsiness of a Student

We capture the class image , and if any student is found sleeping in the class we mark that event , if same student found sleeping for more than 4 times for the same subject then we decrement the attendance of that student by one class.

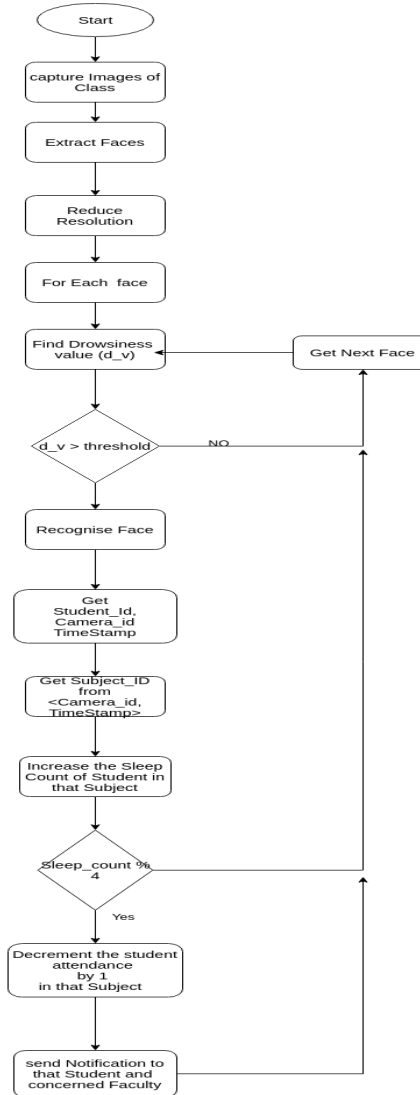


Figure 7: Drowsiness detector model

7 Code Description

The Figure8(DirectoryStructure) represents directory structure showing main files/folders. The description of each code file is in the subsequent subsections in a brief way.

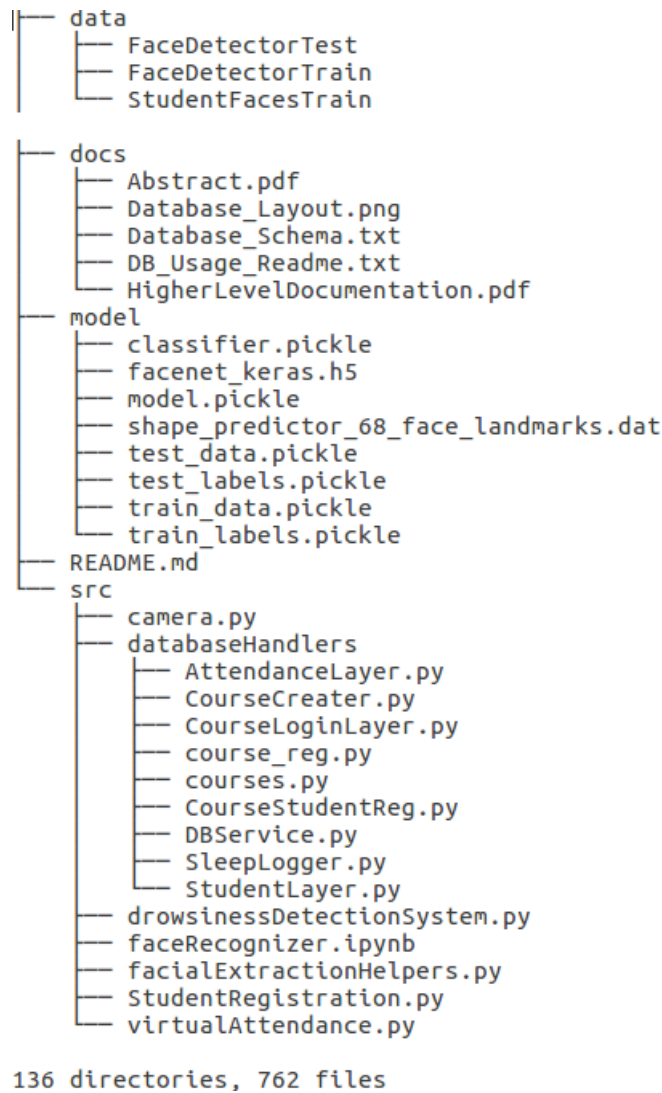


Figure 8: DirectoryStructure

7.1 AttendanceLayer.py

This file helps in writing the attendance information of the student to the data base.It has the following methods in it.

markAttendance() is used to mark the attendance of the student whose

fields are initialised in the keys of dictionary passed to constructor.

getAttendanceByDate() also take input in form of dictionary with following key fields, courseid, rollnumber, semestertype, semesteryear, classroom, dateofclass. This functions returns True if the student was present on a particular date in a particular subject class.

getAttendanceByMonth() expects dictionary with key fields of form courseid, rollnumber, semestertype, semesteryear, month. This function returns the total number of classes attended by a particular student in particular course in a particular month.

getGrossAttendance() expects a dictionary with key and values as courseid, rollnumber, semestertype, semesteryear, year. Then this function returns a value indicating how many classes till date student has attended for the subject for which course id was given.

getNetAttendance() expects parameters in same form as getGrossAttendance(), but this returns total attendance of student after deductions based on his Drowsyness score.

7.2 CourseCreator.py

This python code presents a menu based interface to add courses, get information about a particular offered course.

7.3 CourseLoginLayer.py

This python file helps in verifying the credentials entered by the user(TA or Professor) to start attendance of the Course.It has two methods named insertCourseAuthDetails() and verifyAuth(), where the first one is used to insert the course-id and the passwords to the data base,which will be used for authentication further,while second one is used to start the attendance.i.e It verifies whether the credentials passed to it are correct (or) not.

7.4 CourseStudentReg.py

This particular file can be used for making students register for the courses they wish at the beginning of their semesters.The file expects the roll number of the student and the courses he/she wishes to take in that semester and helps them in registering the course.

7.5 DBService.py

This class's method getConnection() provides a connection object to mysql database.It is used by all other files, which wanted to use database either, for updating or reading, to get a connection object.It is designed this way, to separate out the connection code, so that even if we change data base credentials, they need to get modified at a single place.

7.6 SleepLogger.py

This particular file can be used for marking students who are found sleepy in the class. It has the following methods:

markStudentSleepy() expects a set of rollnumbers found sleepy in the class (will be used by the drowsydetection module at the end of class, by noting all the roll numbers in a python set, so that there will be no issues of redundant data) along with the course-Id and inserts the sent information into the drowsydata for further actions.

getSleepCount() expects a roll number and courseId as parameter and returns a int value indicating in how many classes till date, student found sleeping.

calculatePenalty() expects a roll number and courseId as parameter and returns a value that must be deducted from student's total attendance as penalty.

7.7 StudentLayer.py

The constructor of this class expects dictionary with keys and values as, name(Full name of student), facialFeatures(optional)(facial embedding vector(1X128) of face of student passed as sequence of bytes).

createStudent() creates student with details contained in the dictionary passed to the constructor. Roll Number and Date of Creation are auto-assigned.

getFullName(rollNumber) expects an integer argument that corresponds to the roll number of student and returns the full name corresponding to that rollNumber.

getFacialFeatures(rollNumber) expects an integer argument that corresponds to the roll number of the student and returns the facial feature embedding vector as sequence of bytes. You need to convert it to numpy array for further processing.

getNextRollNumber() gives an integer output indicating what next roll number will be assigned to a new student. This function does not expect an argument.

7.8 courseReg.py

The constructor of this class expects dictionary with keys and values as courseId(Unique Id of the Course), rollNumber(Roll Number assigned to the Student), SemesterType(Should Specify whether its monsoon or spring Semester), SemesterYear(The current Semester year of the Student).

registerCourse() adds the courses entries registered by students into the courseReg table taking all the above mentioned fields as input.

getRollnumber() expects courseId, SemesterType and year as arguments and it returns the students roll numbers registered for that course Id.

getCourseId() expects roll number, SemesterType and year as arguments and it returns the courses the student with roll number passed has taken.

7.9 courses.py

The constructor of this class expects dictionary with keys and values as courseId(Unique Id of the Course), courseName(Name of the Course corresponding to the Course-ID).

addCourse() adds the course entry into the Courses table, which maintains the tuples with attributes CourseId and its Name.

getCourseName() expects course Id as an argument and it returns the corresponding course name assigned to that course Id.

getCourseId() expects course name as an argument and it returns the corresponding course Id assigned to that course name.

7.10 StudentRegistration.py

It has a class called StudentRegistration which deals with Registration of student at the time of admission. The following are the methods of the class:

takeStudentDetails() takes student details as input, if details are ok it calls to capture image of student.

takeStudentPhoto() captures the student present in front of camera, this module present in camera.py.

createAugmentationAndSave() replicates different version of images and save in the path which is provided.

7.11 camera.py

It has a Class called Webcam, which has the following methods:

takeStudentPhoto() captures the image of student standing in front of camera and returns the frame information to the function which called it.

7.12 virtualAttendance.py

This code is triggered when some one tries to start the camera for marking the attendance. The code initially prompts the user asking the course Id and the

password,making sure that only authorized people can start the camera once verified.

verifyCourseIDandPassKey() It expects the course-Id and pass key as the arguments,where these arguments are passed to the `verifyAuth()` method of `CourseAuthenticator` class,where the passed credentials are compared with true ones stored in database and returns 1 to the called function(if credentials matched) and 0(if there is a mismatch).

loadStudent() It takes the Course-Id as input and passes the received information along with semester year and type(whether it is Monsoon or Spring) to the method `getRollnumber()` of `CourseReg` Class to get all the roll numbers of Students under that Course to a python set. By reading the details of all Students to the Python data Structure(set here) helps us,in not visiting the database for every single Student to check whether he/she is registered for that Subject or not.Rather it can use the data structure to check if the roll number predicted is present in it or not.

markAttendance() This method is called by the `StartAttendance()` of the same class by passing the predicted roll numbers list and `courseId` as arguments.This method iterates over the roll number list passed and checks if the student is enrolled for the course(using `studentId` set) and calls the `markAttendance()` of `Attendance` Class,by passing all the relevant details required to mark attendance.

startAttendance() It takes video input, captures frames and calls `performTest()` of `functions.py` to mark attendance for student in each frame.

loadModule() This method helps in loading the pre-trained module, which will be used to get the facial features of Students,where the representations given by the module will be in a way that,similar persons get the features closer in the vector space,where as different persons get the features which are at a far distances from each other in vector space.These features will be used by the `performTest()` of `functions.py` file,inorder to get roll number of Student,by using random forest classifier.

7.13 functions.py

This file stores the core functions used by attendance model to mark attendance.

extractFace() This method is used to detect a face from an image and extract 160*160*3 sized face as a numpy array.

loadFaces() It is a helper function which uses `extractFace()` to extract faces for all the images in a particular directory.

loadDataset() It's a another helper function which uses `loadFaces()` to extract faces of all students present in dataset.

getEmbedding() It represents face as a 1*128 feature vector obtained using pre-trained FacNet model. So that these features can be used for training and prediction.

extractFaceWebcam() This method is used to detect a face from a webcam input and extract 160*160*3 sized face as a numpy array.

performTest() Given a face obtained using extractFaceWebcam() it will use getEmbeddings to get facial features and use pretrained random forest model to recognize the student.

7.14 drowsinessDetectionSystem.py

verifyCourseIDandPassKey() This method will verify the TA or the Instructor's login credentials before they can start the Drowsiness detection system

loadStudents() This function will load the roll numbers of Students belonging to that specific course after correctly verifying the credentials of the TA or the Instructor.

eyeAspectRatio() This function will calculate the eye aspect ratio of the student in the frame. The eye aspect ratio is an estimate of the eye opening state. This function is later on useful to check in the following checkDrowsiness() function.

checkDrowsiness() This function will keep taking in frames from the webcam and checks whether or the student in the frame is sleeping or not. It includes of a threshold of a certain amount of time. If the student sleeps for a time beyond that threshold then, they are marked using the following retrieveRollNumber() function.

retrieveRollNumber() This function takes in the frame and calculates a 1 x 128 numpy array of all facial features using facenet. These features stored are used to detect if there is anyone in the frame, and if there is, we run the predict function that runs the features across the trained model and returns the Roll Number of the student, who has been sleeping. If the face is predicted, we append the Roll Number to a unique array and this array is saved.

loadModule() This method helps in loading the pre-trained module, which will be used to get the facial features of Students, where the representations given by the module will be in a way that, similar persons get the features closer in the vector space, where as different persons get the features which are at a far distances from each other in vector space. These features will be used by the performTest() of functions.py file, in order to get roll number of Student, by using random forest classifier.