PAIN DETECTION USING FACIAL RECOGNITION

FINAL REVIEW REPORT

Submitted by

19BCB0125 - Harshvardhan Mishra
19BCB0109 - S S Sharat Chandra Gubbala
19BCB0116 - Rishin Pandit
19BCE0023 - Vishnu Vardhan Reddy

Prepared For:

SOFTWARE DESIGN AND DEVELOPMENT (CSE1005)

PROJECT COMPONENT

Submitted To:

Ragunthar.T

Assistant Professor (Senior)

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

TABLE OF CONTENTS:

1. INTRODUCTION:

Introduction to project domain – Healthcare:

Healthcare is one of the most crucial fields in humanity. Health care or healthcare is the maintenance or improvement of health via the prevention, diagnosis, and treatment of disease, illness, injury, and other physical and mental impairments in people. Health care is delivered by health professionals in allied health fields. Our project dealt with the use of facial recognition to solve a healthcare related problem: - Pain detection.

Problem Statement:

The current paradigm in all Indian colleges usually uses biometric systems as a means for Attendance and counting issues. We hope to replace this with a more effective, timesaving and less manual process: Facial Recognition.

We aim to detect the pain in a person's facial expressions by means of an algorithm which deduces the Pain Index Level (PIL) and hence, the degree of emotions a person is going through. This can also be extended to detect and display other emotions such as happiness, anger, and so on.

This can be applied by extending this system to hospitals, where there is a dire need for such innovations and applications. This can be used in hospitals as a tool to detect strong emotions like pain and stress for immobile patients, hence immediate medical attention can be brought in such instances.

Objectives – Functionalities:

Aims to detect pain level of a patient using facial recognition concepts.

- Mobile system for pain detection. Can be done in transit on vehicles.
- Easy UI where the application can be opened and used with under 3 clicks.
- Proper report generation with highly accurate results.

Existing system:

There is no such pain detection system to help the patients to detect the pain level using facial muscles.

Proposed System:

Pain is typically assessed by patient self-report. Self-reported pain, however, is difficult to interpret and may be impaired or in some circumstances (i.e., young children and the severely ill) not even possible. To circumvent these problems behavioral scientists have identified reliable and valid facial indicators of pain. Hitherto, these methods have required manual measurement by highly skilled human observers. In this paper we explore an approach for automatically recognizing acute pain without the need for human observers

Frame-level ground truth was calculated from presence/absence and intensity of facial actions previously associated with pain. Active appearance models (AAM) were used to decouple shape and appearance in the digitized face images. Support vector machines (SVM) were compared for several representations from the AAM and of ground truth of varying granularity. We explored two questions pertinent to the construction, design and development of automatic pain detection systems.

For clarification, FACS is an index of facial expressions, but does not actually provide any bio-mechanical information about the degree of muscle activation. Though muscle activation is not part of FACS, the main muscles involved in the facial expression have been added here for the benefit of the reader. Action units (AUs) are the fundamental actions of individual muscles or groups of muscles. Action descriptors (ADs) are unitary movements that may involve the actions of several muscle groups (e.g., a forward-thrusting movement of the jaw). The muscular basis for these actions hasn't been specified and specific behaviors haven't been distinguished as precisely as for the AUs. For most accurate annotation, FACS suggests agreement from at least two independents certified FACS encoders.

2. PROJECT SCOPE and PRINCIPLE:

Our main target audience are patients around the world especially those who require an emergency dose of medicine (painkillers) while outside their homes (on expeditions, treks). Using this software on a mobile/laptop works for their benefit.

Principle involved in the project:

Principle applied in the project are: Pain is typically assessed by patient self-report. Self-reported pain, however, is difficult to interpret and may be impaired or in some circumstances (i.e., young children and the severely ill) not even possible. To circumvent these problems behavioral scientists have identified reliable and valid facial indicators of pain. Hitherto, these methods have required manual measurement by highly skilled human observers. In this paper we explore an approach for automatically recognizing acute pain without the need for human observers. Specifically, our study was restricted to automatically detecting pain in adult patients with rotator cuff injuries. The system employed video input of the patients as they moved their affected and unaffected shoulder. Two types of ground truth were considered. Sequence-level ground truth consisted of Likert-type ratings by skilled observers. Frame-level ground truth was calculated from presence/absence and intensity of facial actions previously associated with pain. Active appearance models (AAM) were used to decouple shape and appearance in the digitized face images. Support vector machines (SVM) were compared for several representations from the AAM and of ground truth of varying granularity. We explored two questions pertinent to the construction, design, and development of automatic pain detection systems. Using FACS, human coders can manually code nearly any anatomically possible facial expression, deconstructing it into the specific action units (AU) and their temporal segments that produced the expression. As AUs are independent of any interpretation, they can be used for any higher order decision making process including recognition of basic emotions, or preprogrammed commands for an ambient intelligent environment. The FACS Manual is over 500 pages in length and provides the AUs, as well as Ekman's 6 interpretation of their meaning. FACS defines AUs, which are a contraction or relaxation of one or more muscles. It also defines a number of Action Descriptors, which differ from AUs in that the authors of FACS have not specified the muscular basis for the action and have not distinguished specific behaviors as precisely as they have for the AUs. For example, FACS can be used to distinguish two types of smiles as follows: Insincere and voluntary Pan-Am smile: contraction of zygomatic major alone Sincere and involuntary Duchenne smile: contraction of zygomatic major and inferior part of orbicularis oculi.

3. KEY CONTRACTS And STAKEHOLDERS:

NAME:	REGISTRATION NO:	Phone Number
Harshvardhan Mishra	**19BCB0125**	**8953429849**
S S Sharat Chandra Gubbala	**19BCB0109**	**7893618314**
Rishin Pandit	**19BCB0116**	**8968537644**
Vishnu Vardhan Reddy	**19BCB0023**	**8008783836**

4. PROJECT RESOURCE REQUIREMENTS:

4.1 SOFTWARE RESOURCE REQUIREMENTS:

User Interfaces:
There are three interfaces for logging into the attendance system. One is the interface for students, which has the details of the classes attended, classes missed and the percentage of attendance. The other interface for parents is of the same nature, where parents can log in too and access the attendance details. The other interface is for the teachers who can both log into and change the attendance credentials of the students.
User interface is required for accessing the database.

Hardware Interfaces:
The hardware interfaces include a HD camera, and a GPU.
The GPU will be used in processing the images for data analysis. The camera will take photos and store it in its memory, which will be later processed with the aid of the GPU to analyze the photos taken and given into the system for further analysis and to work upon them in a more organized manner. The camera and the GPU work together in a synchronized manner to properly work upon the images and achieve the desired results.

Software Interfaces:
We shall be using Python 3.7x for the ML and AI components and further analysis. When invalid inputs are given to the modules then the error messages will be popped up to inform the user that the input provided is not taken by the database. When incomplete information is provided by the user and the user try to submit the form in order to store the details in the database, they will pop up a message box asking the user to enter all the details required.
Communications Interfaces:
The machine will have to be part of the college Local area Network to access the central database. Users should have a web browser in order to access the facilities.

Functional Requirements: -
System Feature 1

Description and Priority:
Capturing the face of a student and marking him/her present on the record, which is the updated in the database. This has the highest priority among all the features.

Stimulus/Response Sequences:
The student who desires to give attendance, has to go and stand before the camera. The system then detects and captures the face in front of it. This is then checked for a match in the database, and correspondingly marked present.

Functional Requirements:
Use "TBD" as a placeholder to indicate when necessary, information is not yet available.
The student is required for marking the attendance. He/she should go and stand in front of the camera in order to be marked present.
A camera is required for capturing the face of student. A GPU is required for processing the image.
<Each requirement should be uniquely identified with a sequence number or a
meaningful tag of some kind.>
REQ-1:
REQ-2:

SystemFeature2:
Description and Priority:

The students and parents can view the attendance ie- all the classes attended by him/her.

The faculty can both view and alter the attendance in the database. This has the second priority.

Stimulus/Response

A web browser is required for accessing this feature. Login is required to goto the portal for viewing and/or changing attendance.

Functional Requirements:

A browser is required to view the attendance. A login id is required to log into the system.
Non-Functional Requirements: -

Performance Requirements:

A minimum RAM of 2 GB and a Nvidia GPU is required for the system to efficiently performall the processing and working.

Safety Requirements

Care should be taken that the GPU isn't overloaded, which might lead to its heating and possible loss of data.

Security Requirements:

The security requirements deal with the primary security. The software should be handled only by the administrator and authorized users. Only the administrator has right to assign permission like creating new accounts and generating password. Only authorized users can access the system with username and password. The server should be secure and care should be taken that the data is not leaked anywhere.

Software Quality Attributes:

The design of the portal is accessible. The database design helps to reduce redundancy and view all the data in an organized manner, thus avoiding confusion.

5. SRS REPORT:

+**Introduction to project domain**: Healthcare is one of the most crucial fields in humanity. Health care or healthcare is the maintenance or improvement of health via the prevention, diagnosis, and treatment of disease, illness, injury, and other physical and mental impairments in people. Health care is delivered by health professionals in allied health fields. Our project dealt with the use of facial recognition to solve a healthcare related problem:- Pain detection.

+ **Problem Statement** The current paradigm in all Indian colleges usually uses biometric systems as a means to attendance and counting issues. We hope to replace this with a more effective, timesaving and less manual process: Facial Recognition.

+ We aim to detect the pain in a person's facial expressions by means of an algorithm which deduces the Pain Index Level (PIL) and hence, the degree of emotions a person is going through. This can also be extended to detect and display other emotions such as happiness, anger, and so on.

+ This can be applied by extending this system to hospitals, where there is a dire need for such innovations and applications. This can be used in hospitals as a tool to detect strong emotions like pain and stress for immobile patients, hence immediate medical attention can be brought in such instances.

Objectives: Functionalities Aims to detect pain level of a patient using facial recognition concepts • Mobile system for pain detection. Can be done in transit on vehicles. • Easy UI where the application can be opened and used with under 3 clicks. • Proper report generation with highly accurate results.

+**Existing system**: There is no such pain detection system to help the patients to detect the pain level using facial muscles.

+ **Proposed System** Pain is typically assessed by patient self-report. Self-reported pain, however, is difficult to interpret and may be impaired or in some circumstances (i.e., young children and the severely ill) not even possible. To circumvent these problems behavioral scientists have identified reliable and valid facial indicators of pain. Hitherto, these methods have required manual measurement by highly skilled human observers. In this paper we explore an approach for automatically recognizing acute pain without the need for human observers Frame-level ground truth was calculated from presence/absence and intensity of facial actions previously associated with pain. Active appearance models (AAM) were used to decouple shape and appearance in the digitized face images. Support vector machines (SVM) were compared for several representations from the AAM and of ground truth of varying granularity. We explored two questions pertinent to the construction, design and development of automatic pain detection systems. For clarification, FACS is an index of facial expressions, but does not actually provide any biomechanical information about the degree of muscle activation. Though muscle activation is not part of FACS, the main muscles involved in the facial expression have been added here for the benefit of the reader. Action units (AUs) are the fundamental actions of individual muscles or groups of muscles. Action descriptors (ADs) are unitary movements that may involve the actions of several muscle groups (e.g., a forward-thrusting movement of the jaw). The muscular basis for these actions hasn't been specified and specific behaviors haven't been distinguished as precisely as for the AUs. For most accurate annotation, FACS suggests agreement from at least two independents certified FACS encoders.

External Interface Requirements: - User Interfaces There are three interfaces for logging into the attendance system. One is the interface for students, which has the details of the classes attended, classes missed and the percentage of attendance. The other interface for parents is of the same nature, where parents can log in too and access

the attendance details. The other interface is for the teachers who can both log into and change the attendance credentials of the students.

Hardware Interfaces The hardware interfaces include a HD camera, and a GPU. The GPU will be used in processing the images for data analysis. The camera will take photos and store it in its memory, which will be later processed with the aid of the GPU to analyze the photos taken and given into the system for further analysis and to work upon them in a more organized manner. The camera and the GPU work together in a synchronized manner to properly work upon the images and achieve the desired results.

+ **Software Interfaces** We shall be using Python 3.7x for the ML and AI components and further analysis. When invalid inputs are given to the modules then the error messages will be popped up in order to inform the user that the input provided is not taken by the database. When incomplete information is provided by the user and the user tries to submit the form in order to store the details in the database, they will pop up a message box asking the user to enter all the details required.

Communications Interfaces The machine will have to be part of the college Local area Network to access the central database. Users should have a web browser in order to access the facilities.

+**Functional Requirements**: - Description and Priority Capturing the face of a student and marking him/her present on the record, which is the updated in the database. This has the highest priority among all the features.

Stimulus/Response Sequences The student who desires to give attendance, has to go and stand before the camera. The system then detects and captures the face in front of it. This is then checked for a match in the database, and correspondingly marked present.

+**Functional Requirements**

• Use “TBD” as a placeholder to indicate when necessary, information is not yet available.

• The student is required for marking the attendance. He/she should go and stand in front of the camera in order to be marked present.

• A camera is required for capturing the face of student. A GPU is required for processing the image.

+**Description and Priority** The students and parents can view the attendance i.e., all the classes attended by him/her. The faculty can both view and alter the attendance in the database. This has the second priority.

+ **Performance Requirements** A minimum RAM of 2 GB Non-Functional Requirements and a Nvidia GPU is required for the system to efficiently perform all the processing and working.

+ **Safety Requirements** Care should be taken that the GPU isn’t overloaded, which might

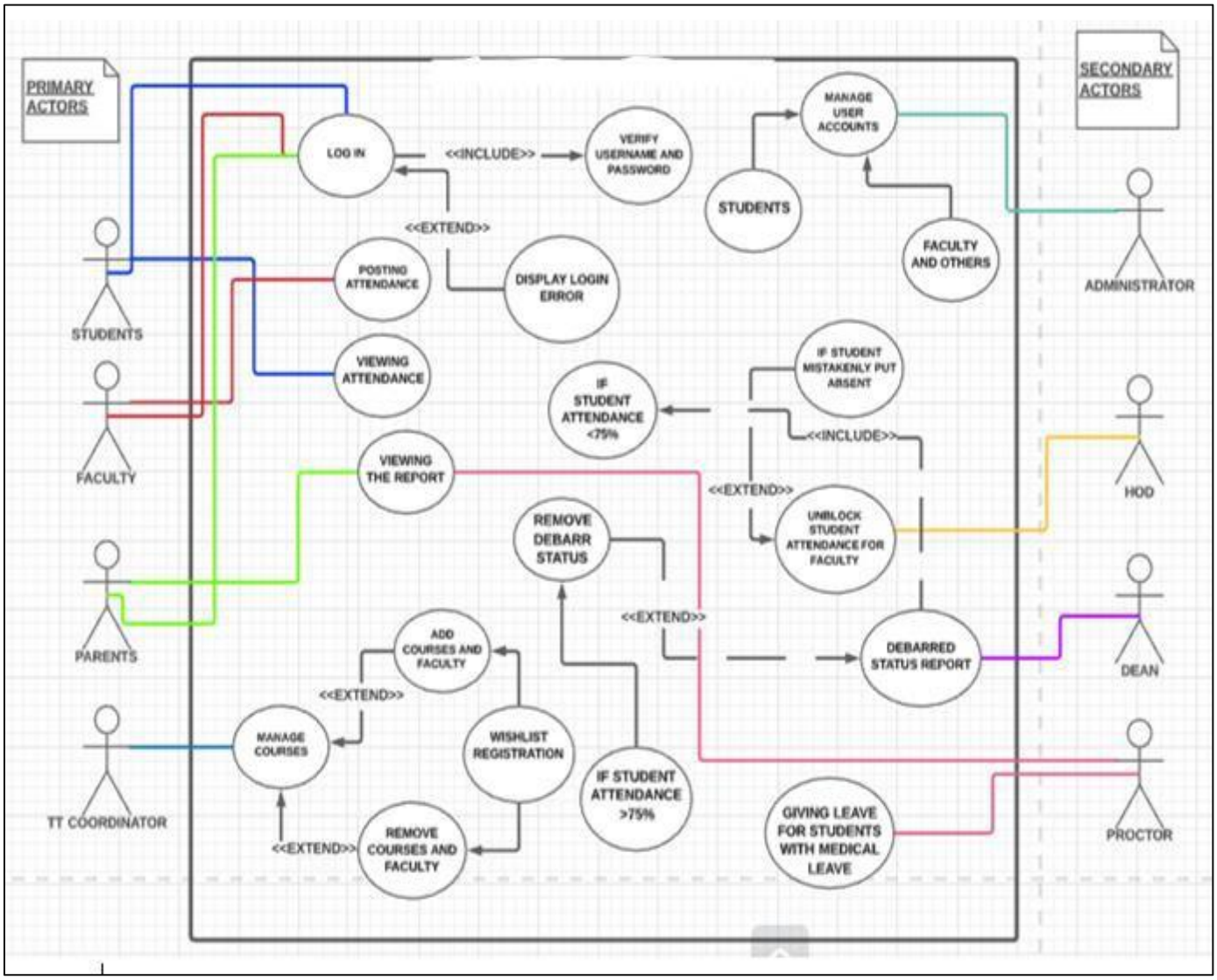
lead to its heating and possible loss of data.

+ **Security Requirements** The security requirements deal with the primary security. The software should be handled only by the administrator and authorized users. Only the administrator has right to assign permission like creating new accounts and generating password. Only authorized users can access the system with username and password. The server should be secure, and care should be taken that the data is not leaked anywhere.

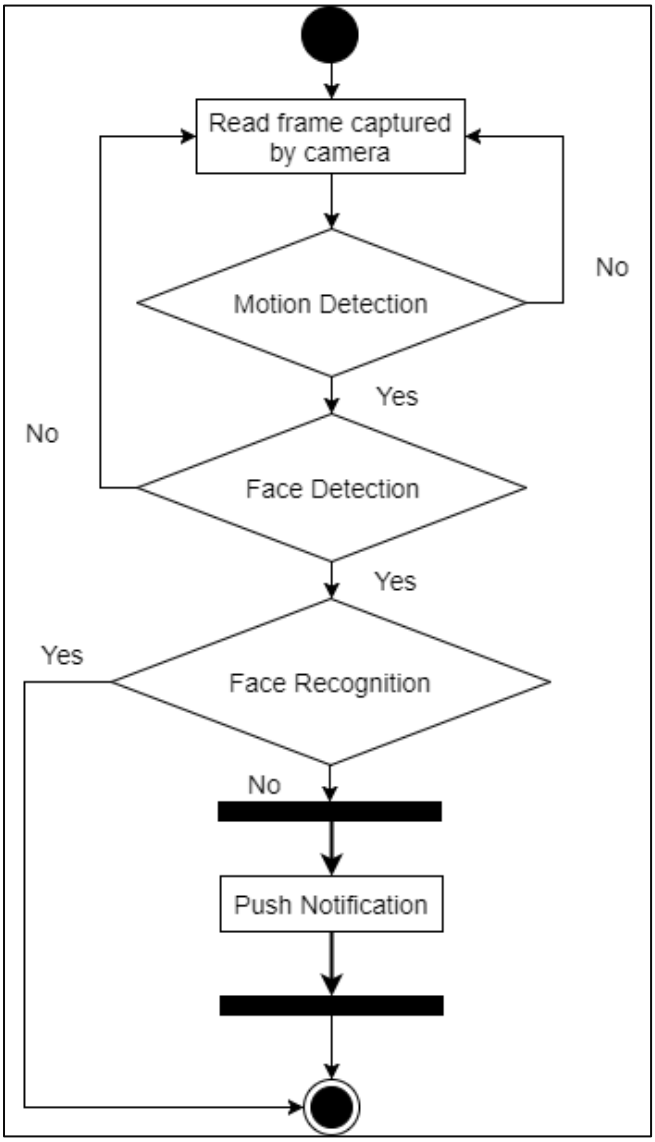
+ **Software Quality Attributes** The design of the portal is accessible. The database design helps to reduce redundancy and view all the data in an organized manner, thus avoiding confusion.

6. DESIGN CONTENTS:

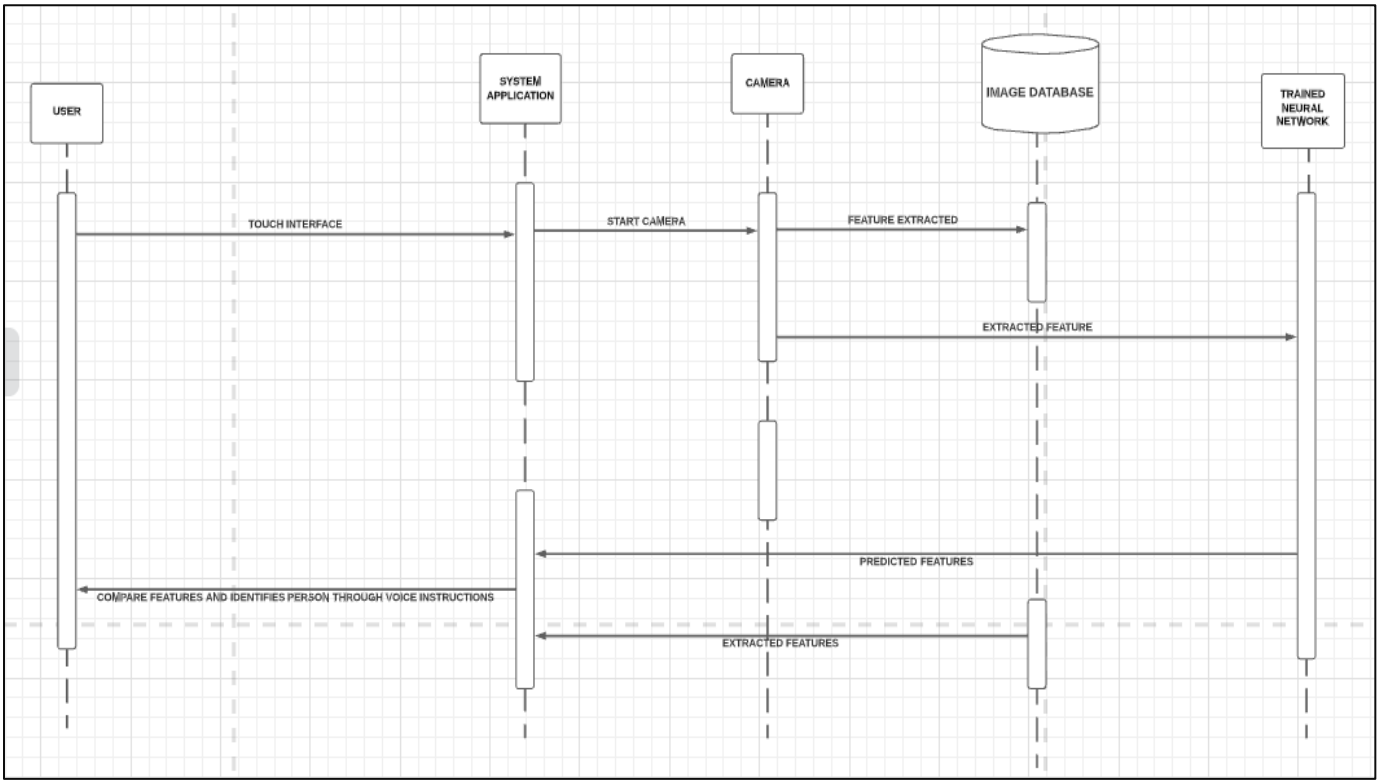
- **UML USECASE DIAGRAM**



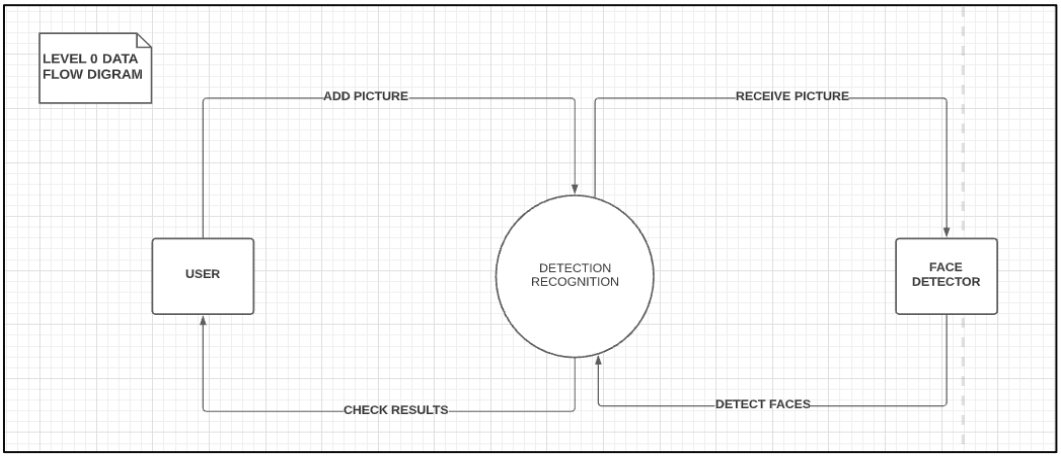
- **ACTIVITY DIAGRAM**

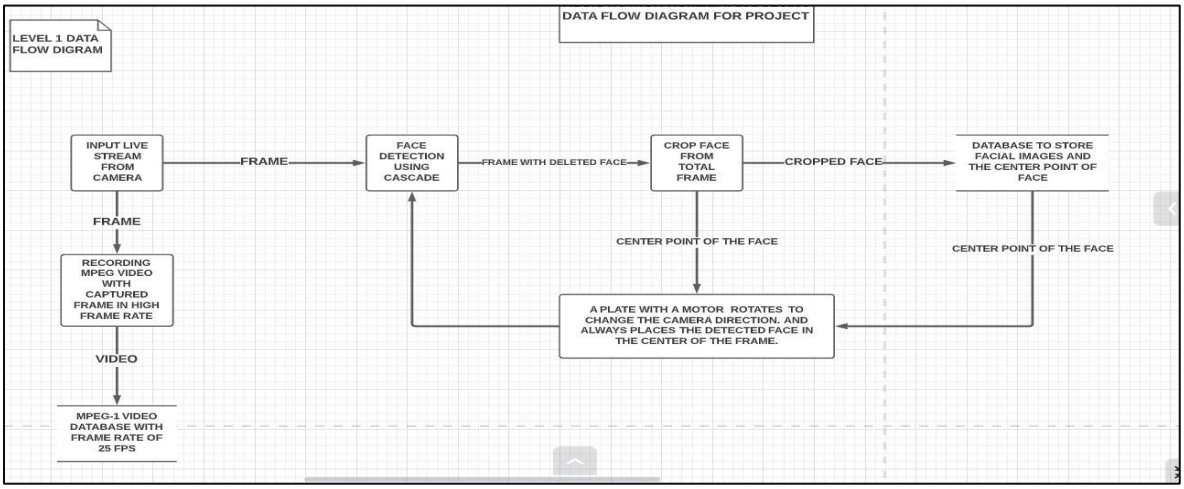


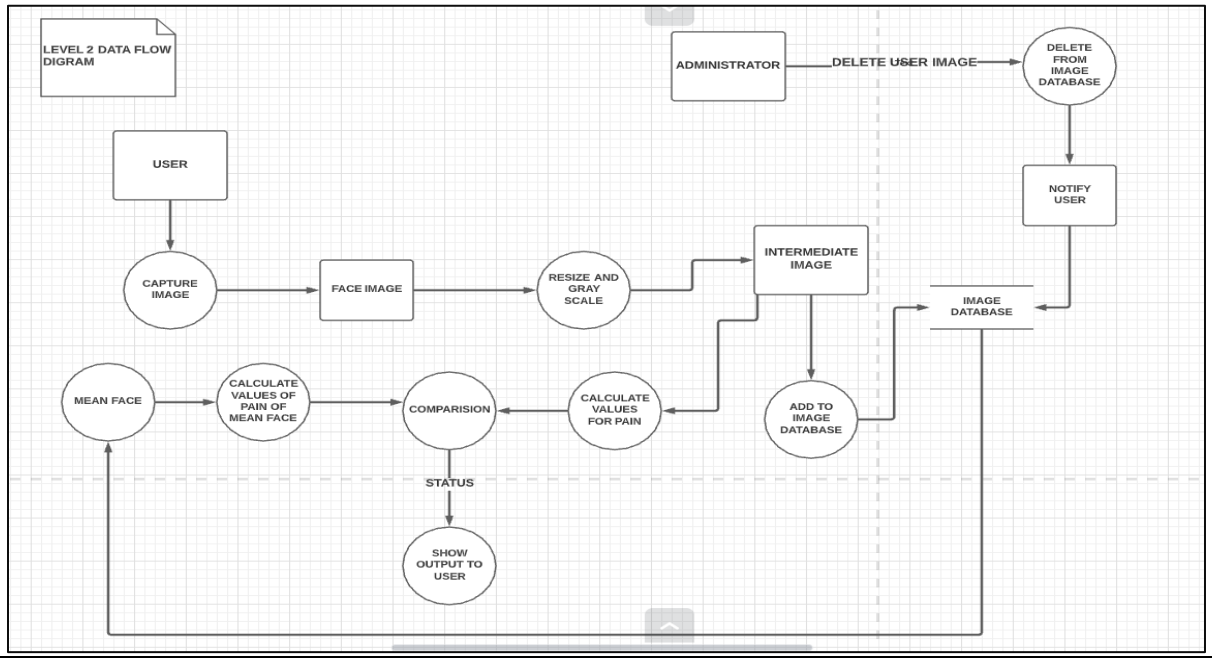
- **SEQUENCE DIAGRAM**



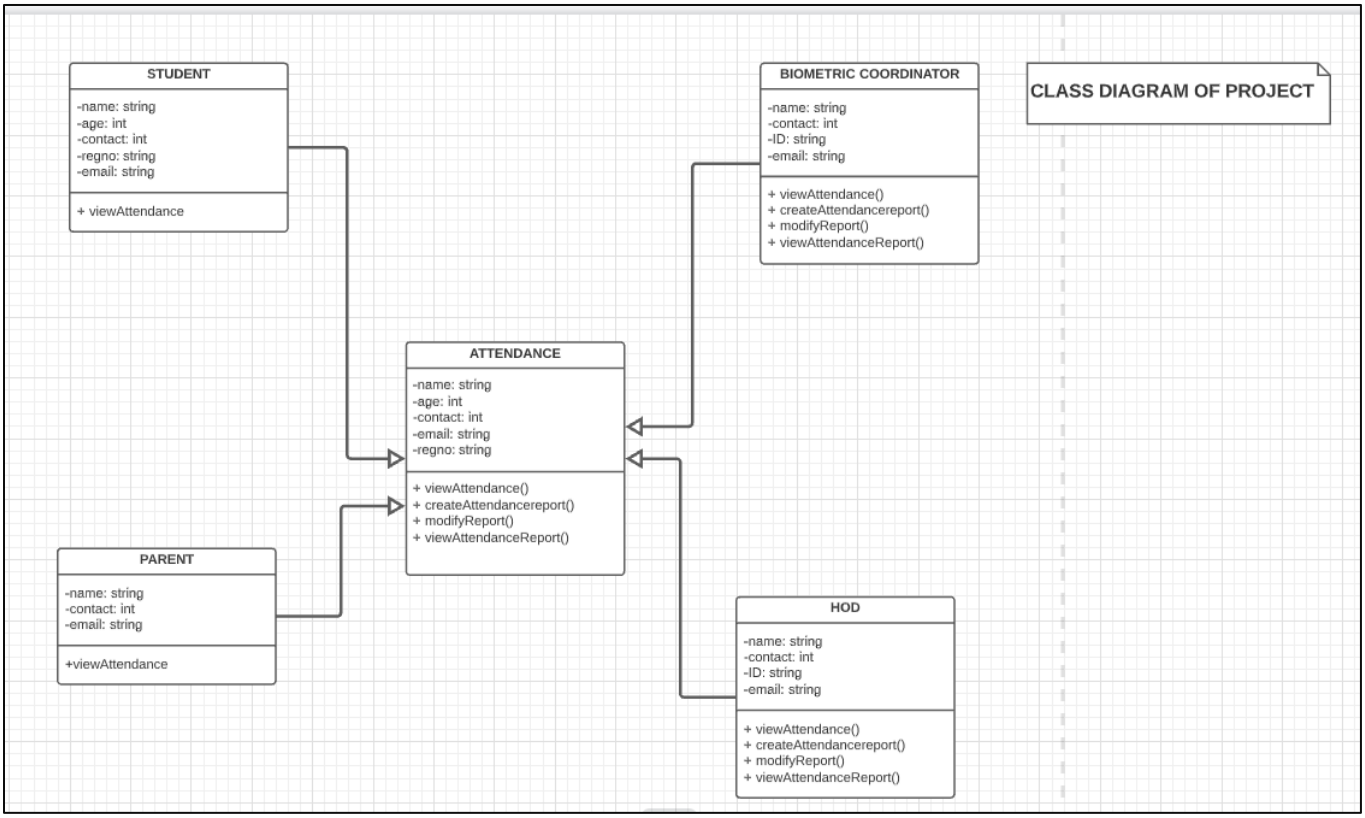
- **DATAFLOW DIAGRAM**



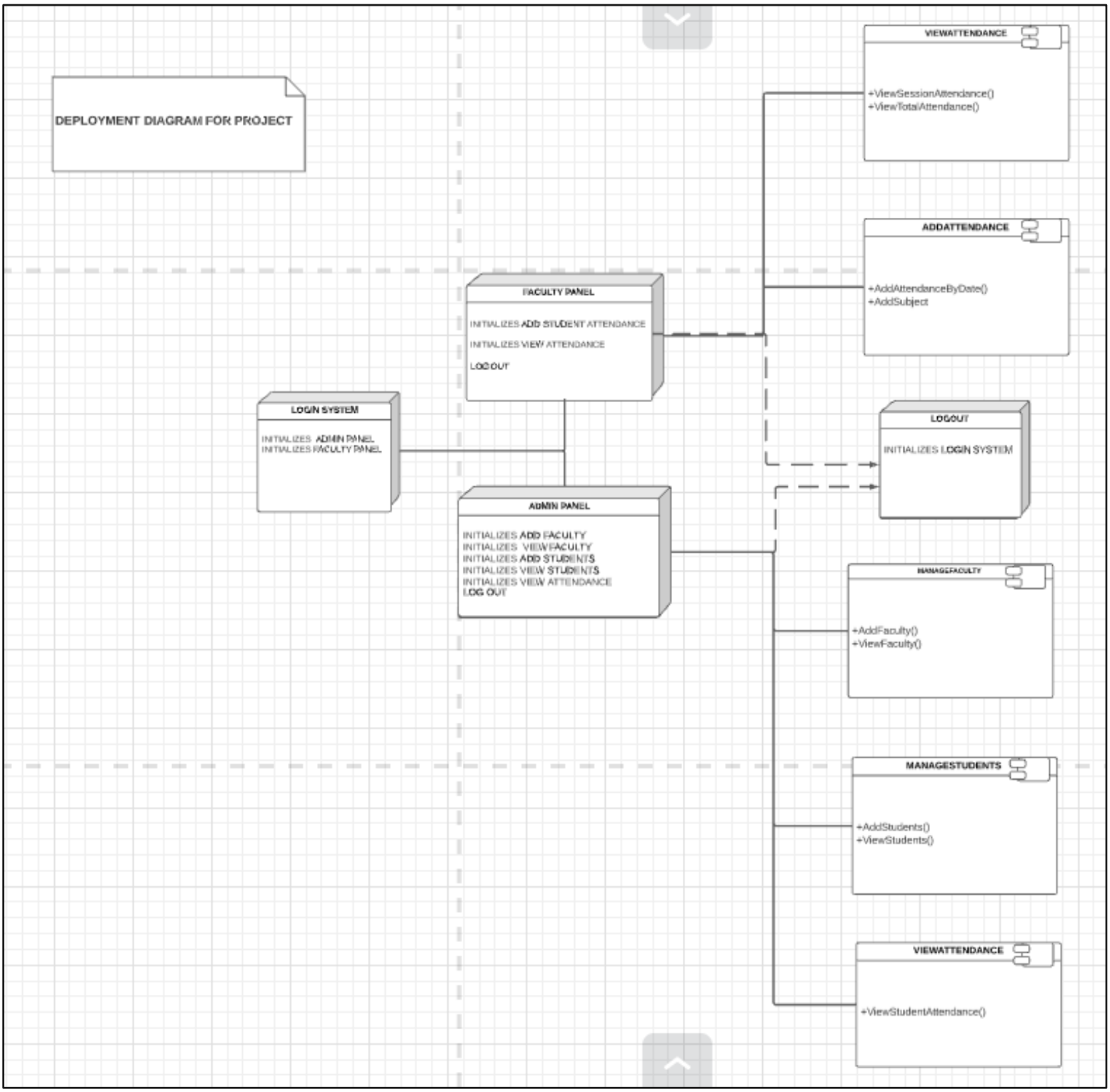




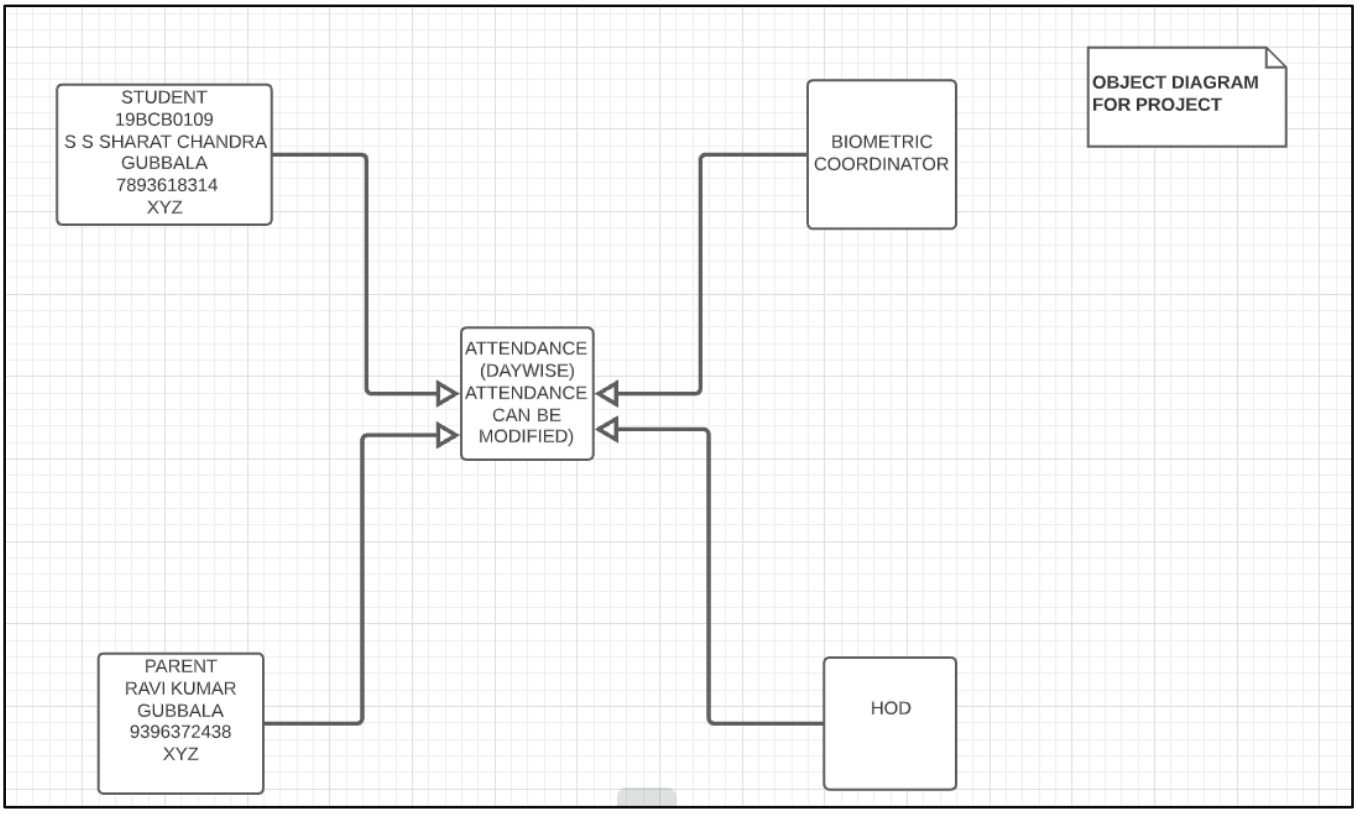
- **CLASS DIAGRAM**



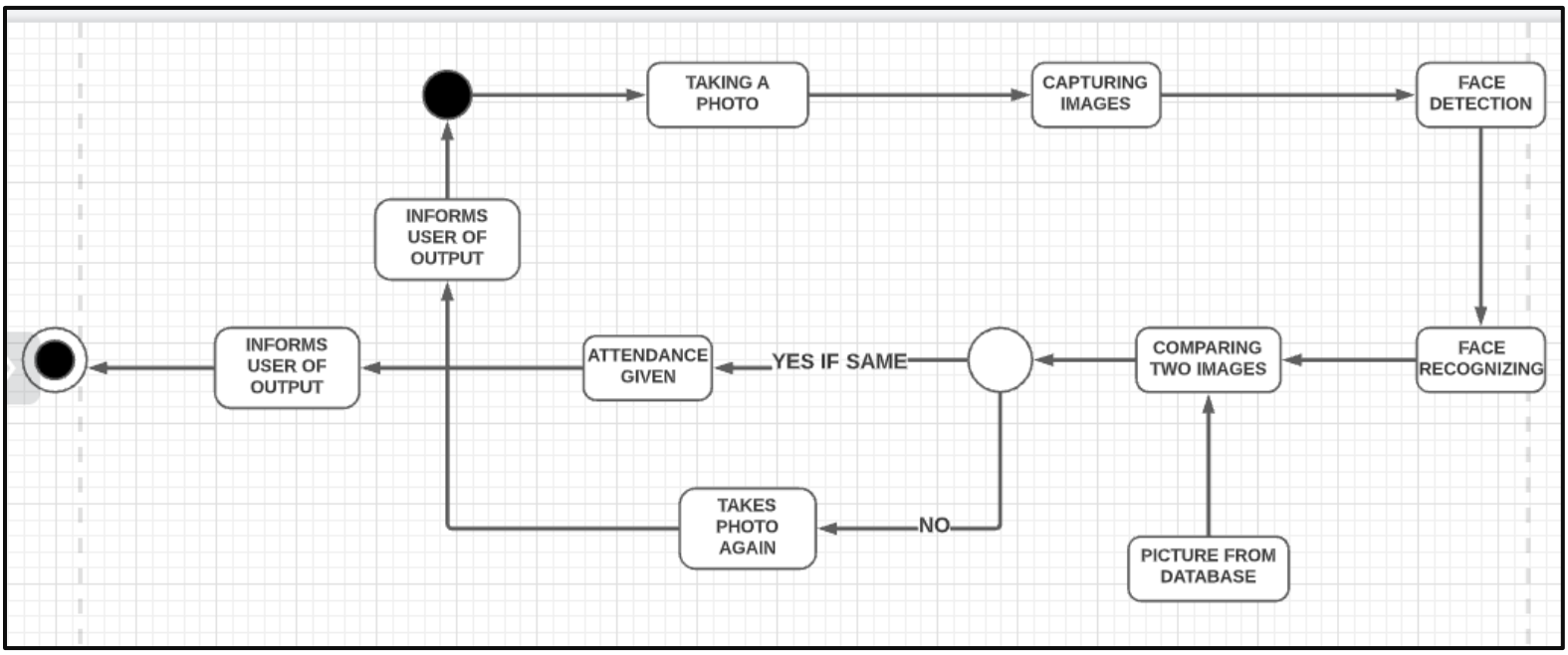
- **DEPLOYMENT DIAGRAM**



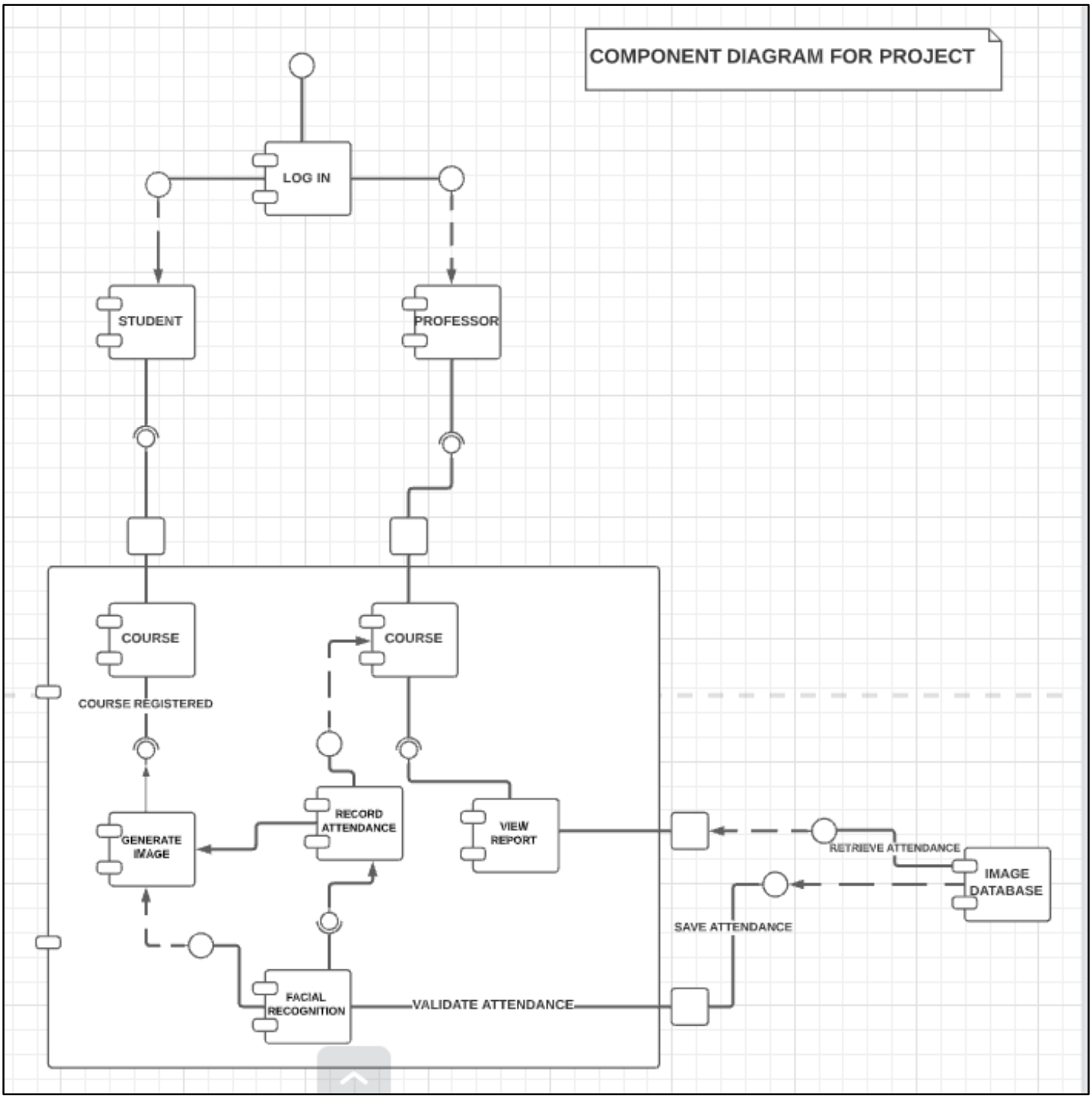
- **OBJECT DIAGRAM**



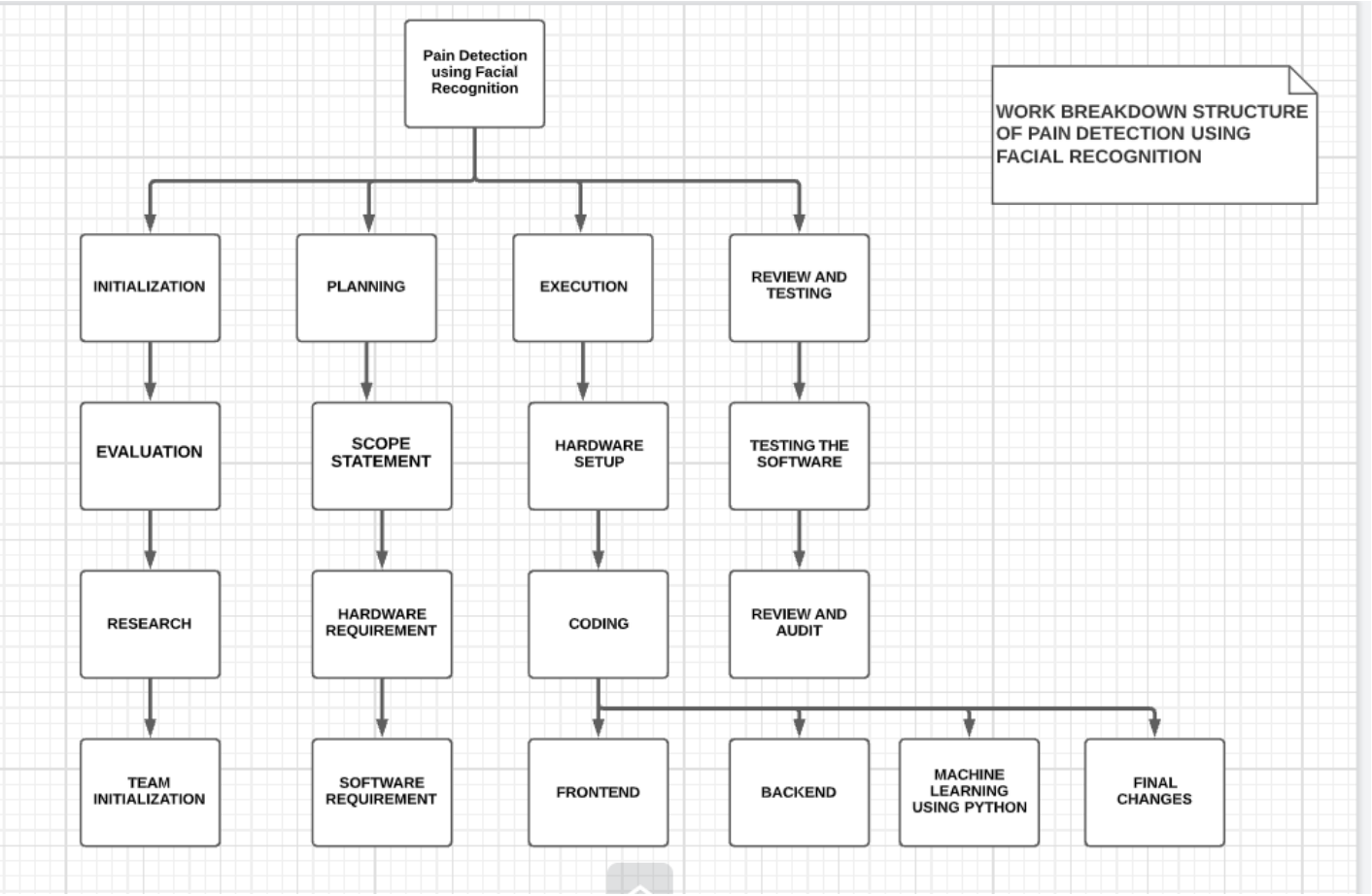
- **STATE CHART DIAGRAM**



- **COMPONENT DIAGRAM**



- **WORK BREAKDOWN STRUCTURE**



7. TEST CASES OF THE PROJECT (UNIT TEST REPORT AND USABILITY TEST REPORT) :

- **UNIT TEST REPORT:**

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
T001	Verify if the system is able to access the webcam	User 1	System is able to access the webcam.	System accesses the webcam	Pass
T002	Verify if the system is able to access the webcam	User 2	System is able to access the webcam.	System accesses the webcam	Pass
T003	Verify if the system is able to access the webcam	User 3	System is able to access the webcam.	System accesses the webcam	Pass
T004	Verify if the user's face is detected by the webcam	User 1	User's face gets detected.	User's face is detected	Pass
T005	Verify if the user's face is detected by the webcam	User 2	User's face gets detected.	User's face is detected	Pass
T006	Verify if the user's face is detected by the webcam	User 3	User's face gets detected.	User's face is detected	Pass
T007	Verify if the system is able to give the output	User 1	System is able to give output	System gives output	Pass
T008	Verify if the system is able to give	User 2	System is able to give output	System gives output	Pass

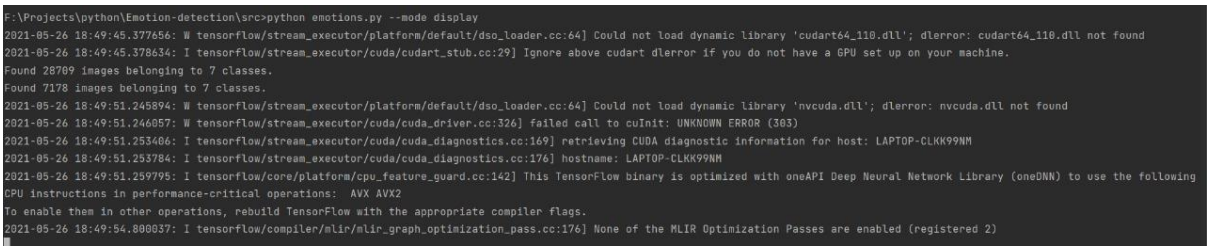
	the output				
T009	Verify if the system is able to give the output	User 3	System is able to give output	System gives output	Pass
T010	Verify if the system can detect the emotion of happiness	User 1	System can detect the emotion	Emotion is detected	Pass
T011	Verify if the system can detect the emotion of happiness	User 2	System can detect the emotion	Emotion is detected	Pass
T012	Verify if the system can detect the emotion of happiness	User 3	System can detect the emotion	Emotion is detected	Pass
T013	Verify if the system can detect the emotion of angry	User 1	System can detect the emotion	Emotion is detected	Pass
T014	Verify if the system can detect the emotion of angry	User 2	System can detect the emotion	Emotion is detected	Pass
T015	Verify if the system can detect the emotion of angry	User 3	System can detect the emotion	Emotion is detected	Pass
T016	Verify if the system can detect the emotion of disgusted	User 1	System can detect the emotion	Emotion is detected	Pass
T017	Verify if the system can detect the emotion of disgusted	User 2	System can detect the emotion	Emotion is detected	Pass

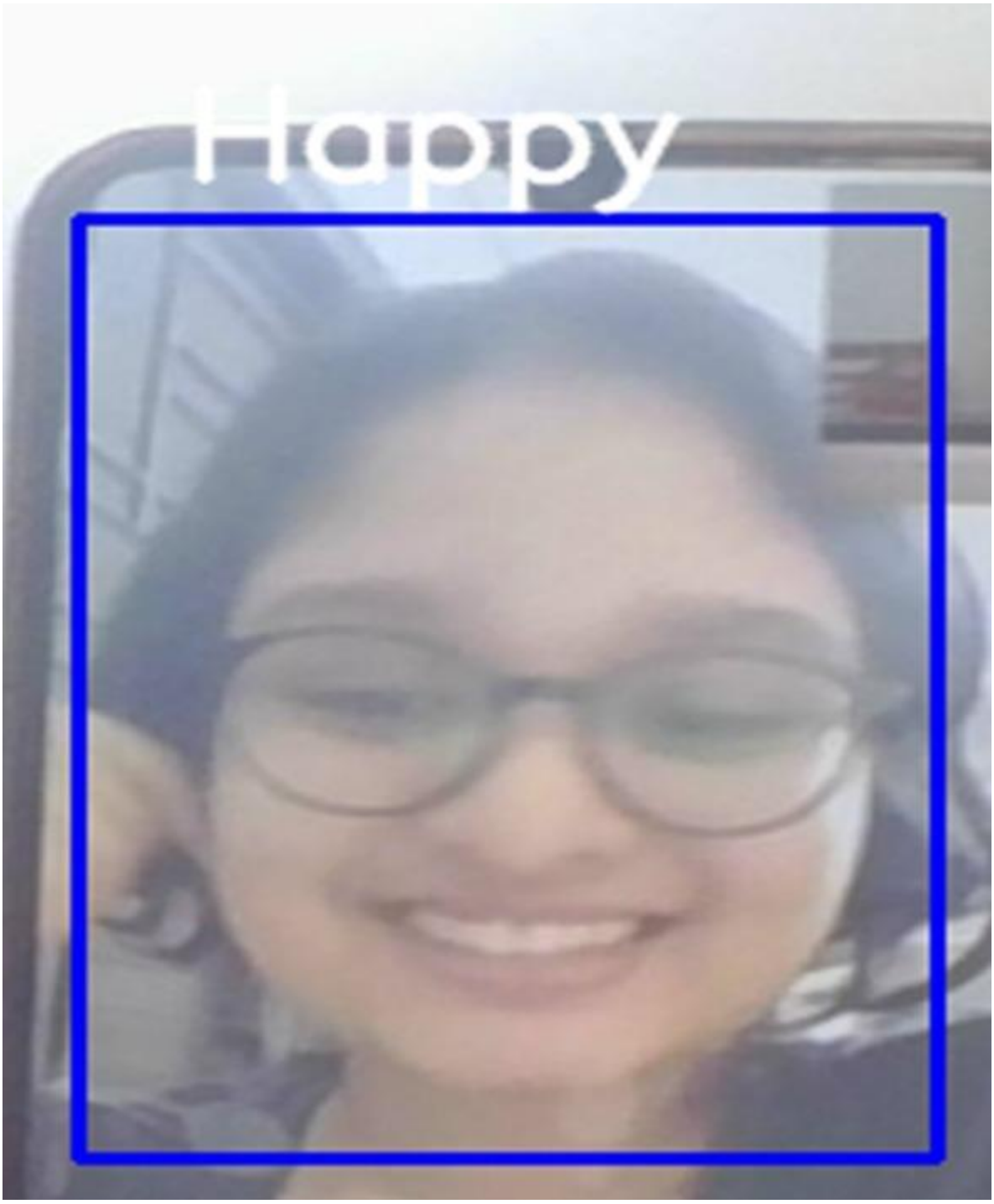
T018	Verify if the system can detect the emotion of disgusted	User 3	System can detect the emotion	Emotion is detected	Pass
T019	Verify if the system can detect the emotion of fearful	User 1	System can detect the emotion	Emotion is detected	Pass
T020	Verify if the system can detect the emotion of fearful	User 2	System can detect the emotion	Emotion is detected	Pass
T021	Verify if the system can detect the emotion of fearful	User 3	System can detect the emotion	Emotion is detected	Pass
T022	Verify if the system can detect the emotion of sad	User 1	System can detect the emotion	Emotion is detected	Pass
T023	Verify if the system can detect the emotion of sad	User 2	System can detect the emotion	Emotion is detected	Pass
T024	Verify if the system can detect the emotion of sad	User 3	System can detect the emotion	Emotion is detected	Pass
T025	Verify if the system can detect the emotion of neutral	User 1	System can detect the emotion	Emotion is detected	Pass
T026	Verify if the system can detect the emotion of neutral	User 2	System can detect the emotion	Emotion is detected	Pass

T027	Verify if the system can detect the emotion of neutral	User 3	System can detect the emotion	Emotion is detected	Pass
T028	Verify if the system can detect the emotion of surprise	User 1	System can detect the emotion	Emotion is detected	Pass
T029	Verify if the system can detect the emotion of surprise	User 2	System can detect the emotion	Emotion is detected	Pass
T030	Verify if the system can detect the emotion of surprise	User 3	System can detect the emotion	Emotion is detected	Pass
T031	Verify if the system can detect multiple users	User 1	Multiple Users can be detected	Multiple Users are detected	Pass
T032	Verify if the system can detect multiple users	User 2	Multiple Users can be detected	Multiple Users are detected	Pass
T033	Verify if the system can detect multiple users	User 3	Multiple Users can be detected	Multiple Users are detected	Pass
T034	Verify if the region of image containing the face is resized to 48x48 and is passed as input to the CNN	User 1	System can resize the image without blurring the picture	System resizes the image without blurring it.	Pass

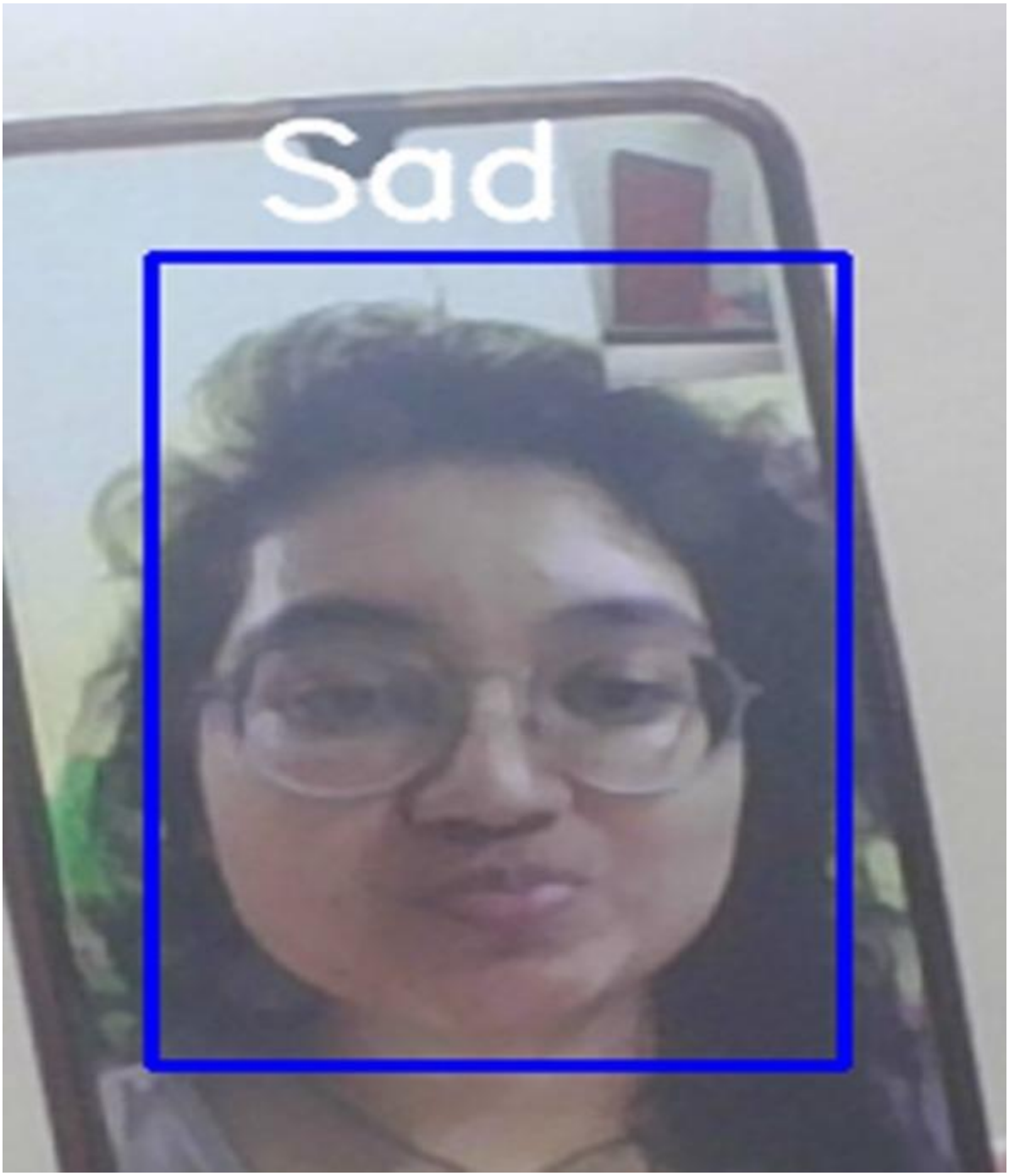
T035	Verify if the region of image containing the face is resized to 48x48 and is passed as input to the CNN	User 2	System can resize the image without blurring the picture	System resizes the image without blurring it.	Pass
T036	Verify if the region of image containing the face is resized to 48x48 and is passed as input to the CNN	User 3	System can resize the image without blurring the picture	System resizes the image without blurring it.	Pass
T037	Verify the functioning of opencv	User 1	Opencv will work efficiently	Opencv works efficiently	Pass
T038	Verify the functioning of opencv	User 2	Opencv will work efficiently	Opencv works efficiently	Pass
T039	Verify the functioning of opencv	User 3	Opencv will work efficiently	Opencv works efficiently	Pass
T040	Verify the compatibili ty with tensorflow	User 1	Tensorflow will be compatible	Tensorflow is compatible	Pass
T041	Verify the compatibili ty with tensorflow	User 2	Tensorflow will be compatible	Tensorflow is compatible	Pass
T042	Verify the compatibili ty with tensorflow	User 3	Tensorflow will be compatible	Tensorflow is compatible	Pass
T043	Verify the working of python emotions.p y --mode train	User 1	The module will work efficiently	Module works efficiently	Pass

T044	Verify the working of python emotions.py --mode train	User 2	The module will work efficiently	Module works efficiently	Pass
T045	Verify the working of python emotions.py --mode train	User 3	The module will work efficiently	Module works efficiently	Pass




Happy


Neutral


Sad


Neutral

- **USABILITY TEST REPORT:**

User Profile Characteristics:

Team Number	Name	Age	Gender
13	Divyansh Jain	19	Male
13	Aditya Anavekar	19	Male
13	Bhupesh Pandey	19	Male

Following is a summary of the participants' computing environment:

URL of tested website:	N/A
Computer platforms:	Windows, linux
Browser tested:	None
Screen resolution:	1536 x 864
Operating system:	windows
Connection speed:	Normal wifi speed

Table :1

# Questions for Testing the product	# Summary of the product		
Participants1(19BCI0240, Bhupesh) Observer (19BCB0116 ,Rishin)	Passed or failed	Time taken	Interface is good/bad/ok
1. Verify the compatibility with tensorflow	Pass	instantaneously	Good
2. Verify the compatibility with tensorflow-2.0	Pass	instantaneously	Good
3. Verify the working of python emotions.py --mode train	Pass	30 mins approx	Good
4. Verify the python emotions.py --mode display	Pass	instantaneously	Good
5. Verify the functioning of open cv	Pass	instantaneously	Good
6. Verify if the system is able to access the webcam	Pass	instantaneously	Good
7. Verify if the user's face is detected by the webcam	Pass	instantaneously	Good
8. Verify if the system is able to give the output	Pass	instantaneously	Good
9. Verify if the system can detect the emotion of happiness	Pass	Shows live result	Good
10. Verify if the system can detect the emotion of angry	Pass	Shows live result	Good
11. Verify if the system can detect the emotion of disgusted	Fail	Shows live result	Bad
12. Verify if the system can detect the emotion of fearful	Pass	Shows live result	Good
13. Verify if the system can detect the emotion of sad	Pass	Shows live result	Good
14. Verify if the system can detect the emotion of neutral	Pass	Shows live result	Good
15. Verify if the system can detect the emotion of surprise	Pass	Shows live result	Good
16. Verify if the system can detect multiple users	Pass	instantaneously	Good
17. Verify model accuracy which approximately reaches 63.2% in 50 epochs.	Pass	About 30 mins	Good
18. Verify if the region of image containing the face is resized to 48x48 and is passed as input to the CNN	Pass	Shows live result	Good
19. Verify if the model's ability to train with new dataset	Pass	Depends on the size of the data set (30 mins for current dataset)	Good

Table :2

# Questions for Testing the product	# Summary of the product		
Participants2(19BCI0236 ,Aditya) Observer (19BCB0116,Rishin)	Passed or failed	Time taken	Interface is good/bad/ok
1. Verify the compatibility with tensorflow	Pass	instantaneously	Good
2. Verify the compatibility with tensorflow-2.0	Pass	instantaneously	Good
3. Verify the working of python emotions.py --mode train	Pass	30 mins approx	Good
4. Verify the python emotions.py --mode display	Pass	instantaneously	Good
5. Verify the functioning of open cv	Pass	instantaneously	Good
6. Verify if the system is able to access the webcam	Pass	instantaneously	Good
7. Verify if the user's face is detected by the webcam	Pass	instantaneously	Good
8. Verify if the system is able to give the output	Pass	instantaneously	Good
9. Verify if the system can detect the emotion of happiness	Pass	Shows live result	Good
10. Verify if the system can detect the emotion of angry	Pass	Shows live result	Good
11. Verify if the system can detect the emotion of disgusted	Fail	Shows live result	Bad
12. Verify if the system can detect the emotion of fearful	Pass	Shows live result	Good
13. Verify if the system can detect the emotion of sad	Pass	Shows live result	Good
14. Verify if the system can detect the emotion of neutral	Pass	Shows live result	Good
15. Verify if the system can detect the emotion of surprise	Fail	Shows live result	Bad
16. Verify if the system can detect multiple users	Pass	instantaneously	Good
17. Verify model accuracy which approximately reaches 63.2% in 50 epochs.	Pass	About 30 mins	Good
18. Verify if the region of image containing the face is resized to 48x48 and is passed as input to the CNN	Pass	Shows live result	Good
19. Verify if the model's ability to train with new dataset	Pass	Depends on the size of the data set (30 mins for current dataset)	Good

Table :3

# Questions for Testing the product	# Summary of the product		
Participants3(19BCT0260 ,Divyansh) Observer (19BCB0125,Harsh)	Passed or failed	Time taken	Interface is good/bad/ok
1. Verify the compatibility with tensorflow	Pass	instantaneously	Good
2. Verify the compatibility with tensorflow-2.0	Pass	instantaneously	Good
3. Verify the working of python emotions.py --mode train	Pass	30 mins approx	Good
4. Verify the python emotions.py --mode display	Pass	instantaneously	Good
5. Verify the functioning of open cv	Pass	instantaneously	Good
6. Verify if the system is able to access the webcam	Pass	instantaneously	Good
7. Verify if the user's face is detected by the webcam	Pass	instantaneously	Good
8. Verify if the system is able to give the output	Pass	instantaneously	Good
9. Verify if the system can detect the emotion of happiness	Pass	Shows live result	Good
10. Verify if the system can detect the emotion of angry	Pass	Shows live result	Good
11. Verify if the system can detect the emotion of disgusted	Pass	Shows live result	Good
12. Verify if the system can detect the emotion of fearful	Pass	Shows live result	Good
13. Verify if the system can detect the emotion of sad	Pass	Shows live result	Good
14. Verify if the system can detect the emotion of neutral	Fail	Shows live result	Bad
15. Verify if the system can detect the emotion of surprise	Pass	Shows live result	Good
16. Verify if the system can detect multiple users	Pass	instantaneously	Good
17. Verify model accuracy which approximately reaches 63.2% in 50 epochs.	Pass	About 30 mins	Good
18. Verify if the region of image containing the face is resized to 48x48 and is passed as input to the CNN	Pass	Shows live result	Good
19. Verify if the model's ability to train with new dataset	Pass	Depends on the size of the data set (30 mins for current dataset)	Good

Exit Questions/User Impressions

At the end of each session, we asked participants questions:

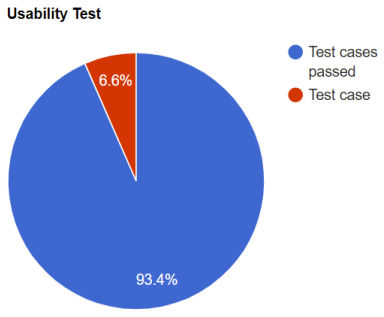
- What is your overall impression to the system?
- What is your impression of the detector capability?
- Do you feel this site is efficient?
- What did you like best about the system?
- What did you like least about the system?
- If you were the developer, what would be the first thing you would do to improve the system?
- Is there anything that you feel is missing on this site?
- If you were to describe this system to a colleague in a sentence or two, what would you say?
- Do you have any other final comments or questions?

Participate 1	Questions	Answer
	What is your overall impression to the system?	Quite an efficient system
	What is your impression of the detector capability?	Fairly accurate
	Do you feel this system is efficient?	Yes
	What did you like best about the system?	The degree of accuracy to which it senses the emotion
	What did you like least about the system?	Range of emotions could have been increased
	If you were the developer, what would be the first thing you would do to improve the system?	Improve the interface
	Is there anything that you feel is missing on this site?	The system could have displayed the percentage of various emotions
	If you were to describe this system to a colleague in a sentence or two, what would you say?	Touchless Attendance System
	Do you have any other final comments or questions?	Nice attempt

Participant 2	Questions	Answer
	What is your overall impression to the system?	Amazing system
	What is your impression of the detector capability?	Very accurate
	Do you feel this system is efficient?	Yes
	What did you like best about the system?	No need to touch anything
	What did you like least about the system?	Interface could have been better
	If you were the developer, what would be the first thing you would do to improve the system?	Improve the interface
	Is there anything that you feel is missing on this site?	Some kind of confirmation that data is successfully stored
	If you were to describe this system to a colleague in a sentence or two, what would you say?	A nice system to improve the way of recording attendance
	Do you have any other final comments or questions?	Execution of the idea is very nice

Participant 3	Questions	Answer
	What is your overall impression to the system?	Nice
	What is your impression of the detector capability?	100% perfection is not there but it still works
	Do you feel this system is efficient?	Yes
	What did you like best about the system?	The overall idea is very new to me
	What did you like least about the system?	The quality of the image
	If you were the developer, what would be the first thing you would do to improve the system?	Improve the quality of the camera
	Is there anything that you feel is missing on this site?	Confirmatory signal
	If you were to describe this system to a colleague in a sentence or two, what would you say?	This system accurately understands the range of emotions just by a glance
	Do you have any other final comments or questions?	Nice idea

Total Test Cases	Test Cases Executed	Test Cases Passed	Test Cases Failed
57	57	53	4



8. IMPLEMENTATION:

- **CODE:**

```
from future import division, absolute_import import
numpy as np
import tflearn
from tflearn.layers.core import input_data, dropout, fully_connected, flatten22
SoftwareRequirements Specificationfor<Project> Page 22
from tflearn.layers.conv import conv_2d, max_pool_2d, avg_pool_2d from
tflearn.layers.merge_ops import merge
from tflearn.layers.normalization import local_response_normalization from
tflearn.layers.estimator import regression
from os.path import isfile, join
import sys
import tensorflow as tf
import os
# prevents appearance of tensorflow warnings
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
tf.logging.set_verbosity(tf.logging.ERROR)
class EMR:
def init(self):
self.target_classes = ['angry', 'disgusted', 'fearful', 'happy', 'sad', 'surprised', 'neutral']
def build_network(self): """
Build the convnet.
Input is 48x48
3072 nodes in fully connected layer """
self.network = input_data(shape = [None, 48, 48, 1])
print("Input data ",self.network.shape[1:])
self.network = conv_2d(self.network, 64, 5, activation = 'relu')
print("Conv1 ",self.network.shape[1:])
self.network = max_pool_2d(self.network, 3, strides = 2)
print("Maxpool1 ",self.network.shape[1:])
self.network = conv_2d(self.network, 64, 5, activation = 'relu')
print("Conv2 ",self.network.shape[1:])
self.network = max_pool_2d(self.network, 3, strides = 2)
print("Maxpool2 ",self.network.shape[1:])
self.network = conv_2d(self.network, 128, 4, activation = 'relu')
print("Conv3 ",self.network.shape[1:])
self.network = dropout(self.network, 0.3)
print("Dropout ",self.network.shape[1:])
self.network = fully_connected(self.network, 3072, activation = 'relu')
print("Fully connected",self.network.shape[1:])
self.network = fully_connected(self.network, len(self.target_classes), activation
= 'softmax')
```

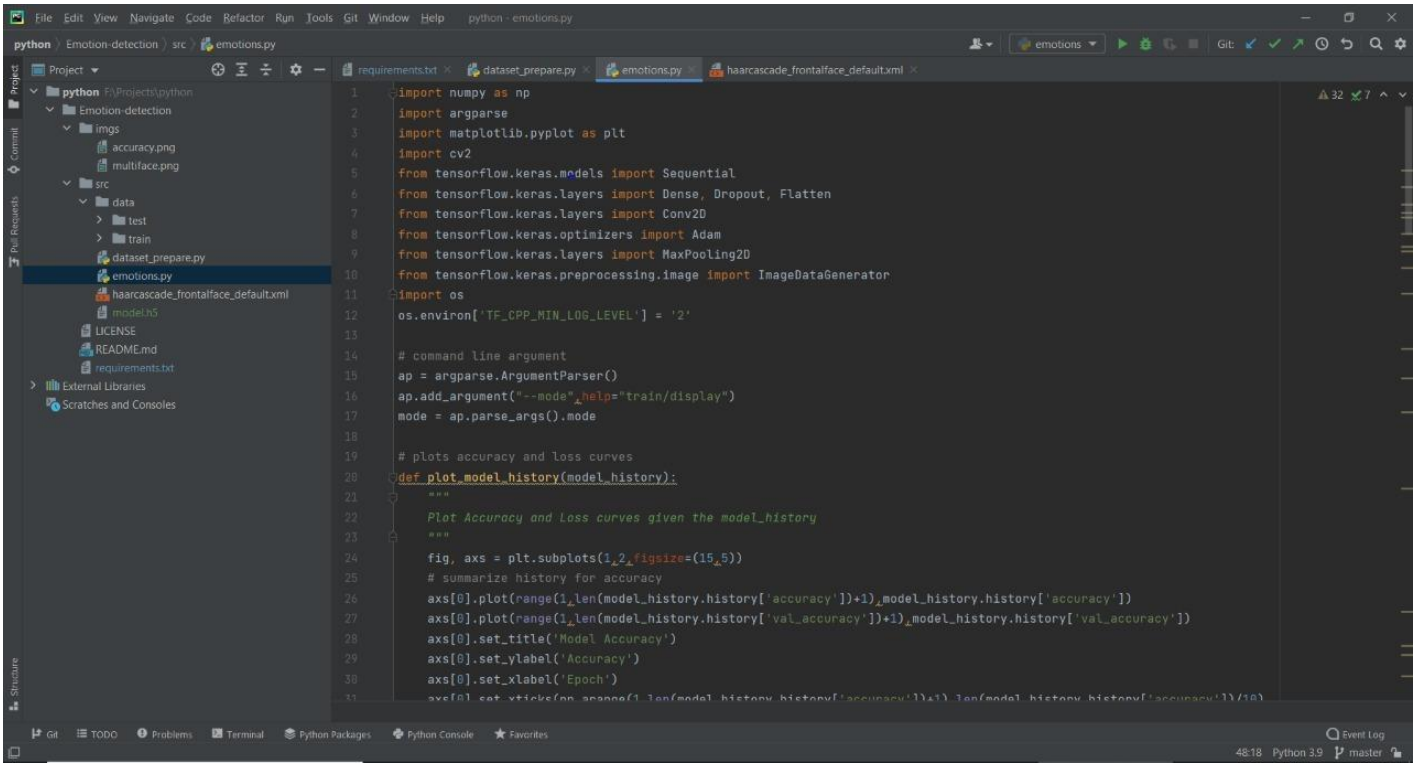
```
print("Output ",self.network.shape[1:])
print("\n")
# Generates a TrainOp which contains the information about optimization process -
optimizer, loss function, etc
self.network = regression(self.network,optimizer = 'momentum',metric =
'accuracy',loss = 'categorical_crossentropy')
# Creates a model instance.23
SoftwareRequirements Specificationfor<Project> Page 23
self.model = tflearn.DNN(self.network,checkpoint_path =
'model_1_atul',max_checkpoints = 1,tensorboard_verbose = 2)
# Loads the model weights from the checkpoint
self.load_model()
def predict(self, image):
"""
Image is resized to 48x48, and predictions are returned.
"""
if image is None:
return None
image = image.reshape([-1, 48, 48, 1]) return
self.model.predict(image)
def load_model(self):
"""
Loads pre-trained model. """
if isfile("model_1_atul.tflearn.meta"):
self.model.load("model_1_atul.tflearn") else:
print("---> Couldn't find model")
if name == "main":
print("\n------------Pain Detection Program ----------------------\n")
network = EMR()
if sys.argv[1] == 'singleface':
import singleface
if sys.argv[1] == 'multiface':
import multiface24
SoftwareRequirements Specificationfor<Project> Page 24
Part 2:
\import cv2
import sys
import numpy as np from
model import EMR
# prevents opencl usage and unnecessary logging messages
cv2.ocl.setUseOpenCL(False)
EMOTIONS = ['angry', 'disgusted', 'fearful', 'smile', 'sad', 'surprised', 'neutral']
def format_image(image): """
Function to format frame """
if len(image.shape) > 2 and image.shape[2] == 3: #
determine whether the image is color
```

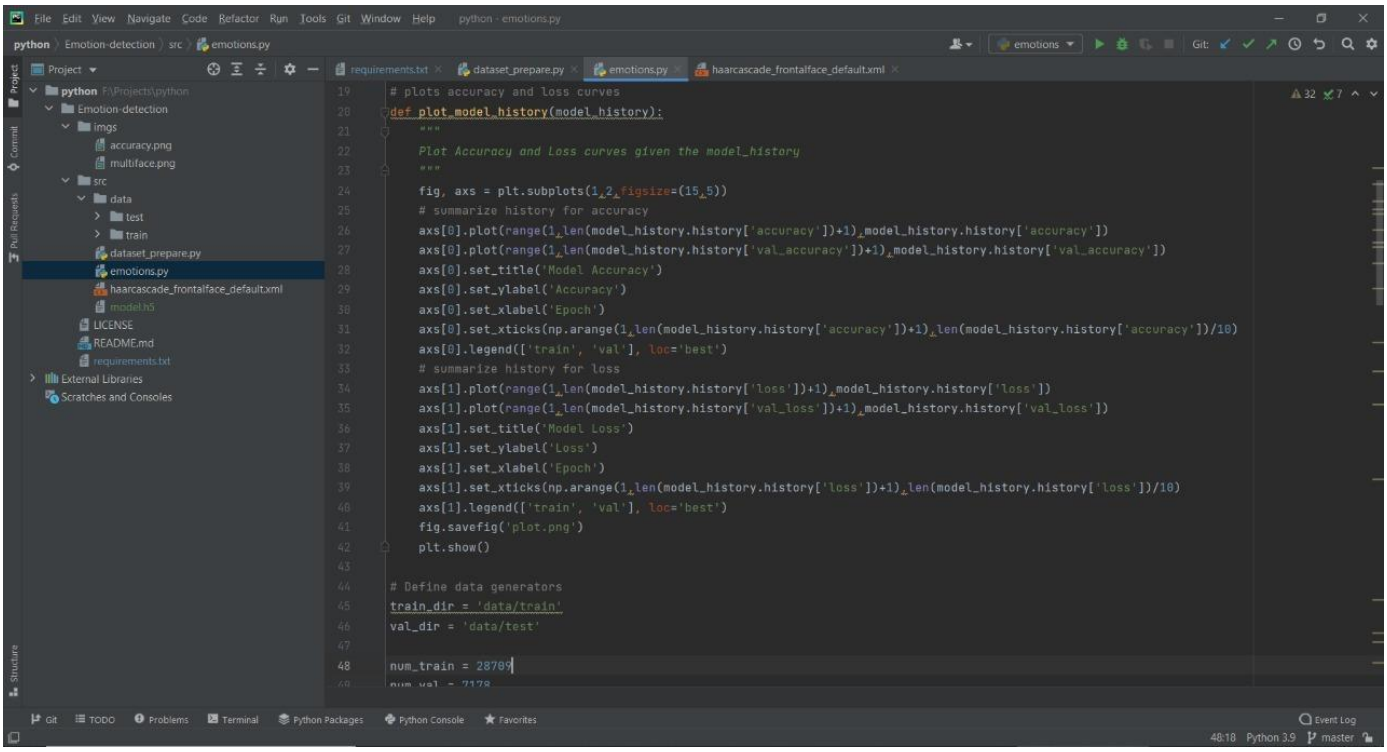
```
image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) else:
# Image read from buffer
image = cv2.imdecode(image, cv2.CV_LOAD_IMAGE_GRAYSCALE)
cascade_classifier =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
faces = cascade_classifier.detectMultiScale(image,scaleFactor = 1.3
,minNeighbors = 5)
if not len(faces) > 0:
return None
# initialize the first face as having maximum area, then find the one with max_area
max_area_face = faces[0] for
face in faces:
if face[2] * face[3] > max_area_face[2] * max_area_face[3]:
max_area_face = face
face = max_area_face
# extract ROI of face
image = image[face[1]:(face[1] + face[2]), face[0]:(face[0] + face[3])]
try:25
SoftwareRequirements Specificationfor<Project> Page 25
# resize the image so that it can be passed to the neural network
image = cv2.resize(image, (48,48), interpolation = cv2.INTER_CUBIC) /
255.
except Exception:
print(" ------ >Problem during resize")
return None
return image
# Initialize object of EMR class
network = EMR()
network.build_network()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
feelings_faces = []
# append the list with the emoji images
for index, emotion in enumerate(EMOTIONS):
feelings_faces.append(cv2.imread('./emojis/' + emotion + '.png', -1))
while True:
# Again find haar cascade to draw bounding box around face ret, frame
= cap.read()
if not ret:
break
facecasc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml') gray =
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = facecasc.detectMultiScale(gray,scaleFactor=1.3, minNeighbors=5)
# compute softmax probabilities
result = network.predict(format_image(frame)) if
result is not None:
```

```
expsum=result[0][0]+result[0][1]+result[0][2]+result[0][4]+result[0][5]+result[ 0][6]
# write the different emotions and have a bar to indicate probabilities for each class
if result[0][3]>0.7 and expsum<0.25: pain
= 0
else:
pain26
SoftwareRequirements Specificationfor<Project> Page 26
=3*result[0][4]+3*result[0][0]+1*result[0][1]+4*result[0][2]+1.5*result[0][3]+
0.5*result[0][5]-0.05*result[0][6]
for index, emotion in enumerate(EMOTIONS):
cv2.putText(frame, emotion, (10, index * 20 + 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1);
cv2.rectangle(frame, (130, index * 20 + 10), (130 +
int(result[0][index] * 100), (index + 1) * 20 + 4), (255, 0, 0), -1)
# find the emotion with maximum probability and display it
#maxindex = np.argmax(result[0])
#font = cv2.FONT_HERSHEY_SIMPLEX
#cv2.putText(frame,EMOTIONS[maxindex],(10,200), font,
2,(255,255,255),2,cv2.LINE_AA)
#face_image = feelings_faces[maxindex]
#painlevel of the person along with the bar
if(pain<1.35):
cv2.putText(frame,"Pain level:"+str(pain),(10,350), font,
1,(255,255,255),2,cv2.LINE_AA)
elif(pain>=1.35 and pain<2.1):
cv2.putText(frame,"Pain level:"+str(pain),(10,350), font,
1,(0,255,255),2,cv2.LINE_AA)
else:
cv2.putText(frame,"Pain level:"+str(pain),(10,350), font,
1,(0,0,255),2,cv2.LINE_AA)
cv2.rectangle(frame, (115, 400), (115+ int(pain*90), 380), (255, 0, 0),
-1)
#for c in range(0, 3):
# The shape of face_image is (x,y,4). The fourth channel is 0 or 1.
In most cases it is 0, so, we assign the roi to the emoji.
# You could also do: frame[200:320,10:130,c] = frame[200:320, 10:130, c]
(1.0 - face_image[:, :, 3] / 255.0)
#frame[200:320, 10:130, c] = face_image[:,:,c]*(face_image[:, :, 3]
/ 255.0) + frame[200:320, 10:130, c] * (1.0 - face_image[:, :, 3] / 255.0)
if len(faces) > 0:
# draw box around face with maximum area 27
SoftwareRequirements Specificationfor<Project> Page 27
max_area_face = faces[0]
for face in faces:
if face[2] * face[3] > max_area_face[2] * max_area_face[3]:
max_area_face = face
```

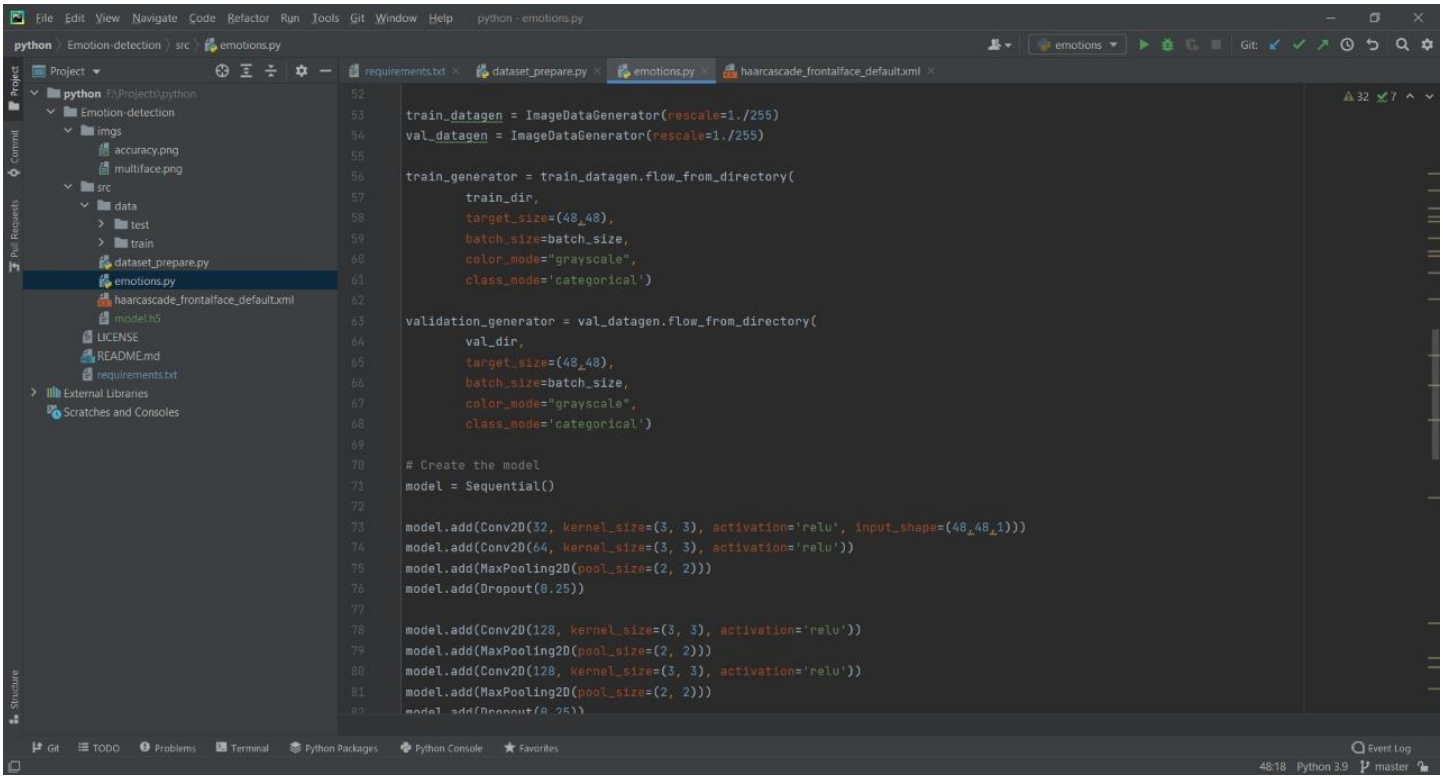
face = max_area_face (x,y,w,h) =
max_area_face
frame = cv2.rectangle(frame,(x,y-50),(x+w,y+h+10),(255,0,0),2)
cv2.imshow('Video', cv2.resize(frame,None,fx=2,fy=2,interpolation =
cv2.INTER_CUBIC))
if cv2.waitKey(1) & 0xFF == ord('q'): break
cap.release()
cv2.destroyAllWindows()
#pain = 6*result[0][4]+4*result[0][0]+2*result[0][1]+result[0][2]-
4*result[0][3]+2*result[0][5]

- **SCREENSHOTS OF OUTPUT:**

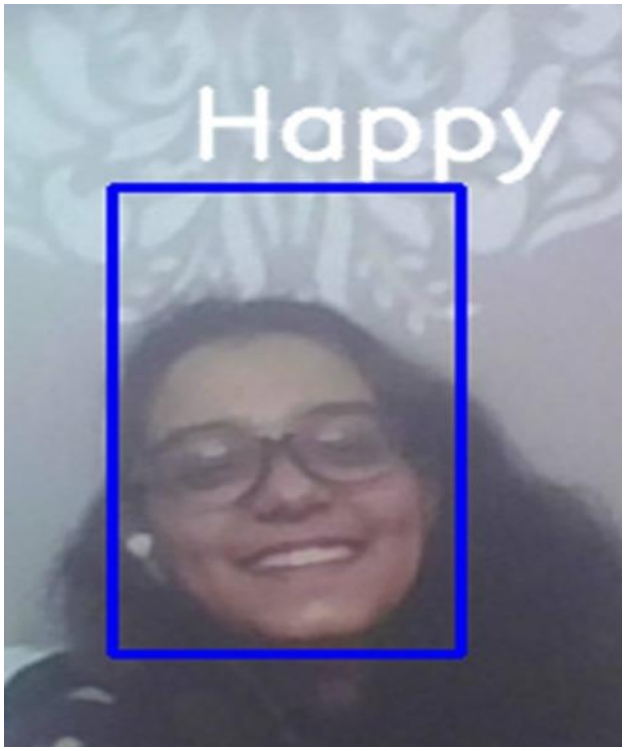


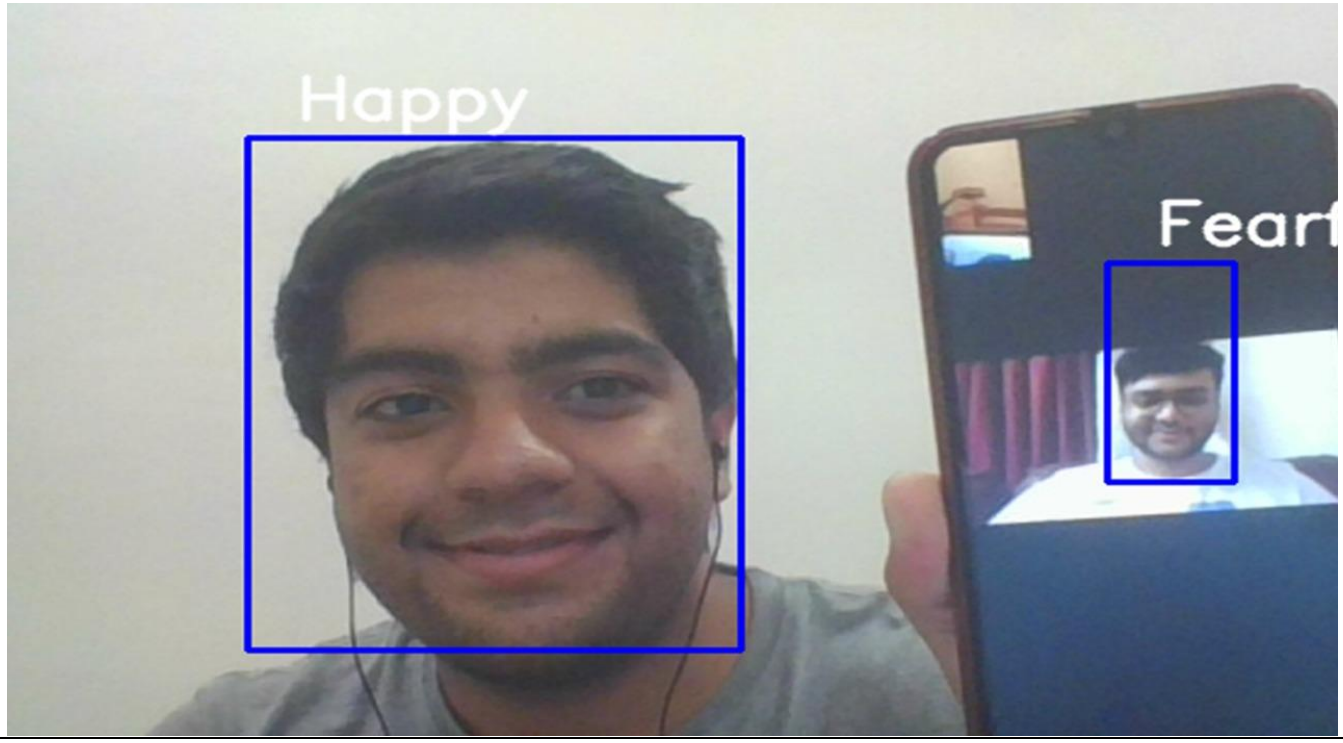


```
# plots accuracy and loss curves
def plot_model_history(model_history):
    """
    Plot Accuracy and Loss curves given the model_history
    """
    fig, axs = plt.subplots(1,2,figsize=(15,5))
    # summarize history for accuracy
    axs[0].plot(range(1,len(model_history.history['accuracy'])+1),model_history.history['accuracy'])
    axs[0].plot(range(1,len(model_history.history['val_accuracy'])+1),model_history.history['val_accuracy'])
    axs[0].set_title('Model Accuracy')
    axs[0].set_ylabel('Accuracy')
    axs[0].set_xlabel('Epoch')
    axs[0].set_xticks(np.arange(1,len(model_history.history['accuracy'])+1),len(model_history.history['accuracy'])/10)
    axs[0].legend(['train', 'val'], loc='best')
    # summarize history for loss
    axs[1].plot(range(1,len(model_history.history['loss'])+1),model_history.history['loss'])
    axs[1].plot(range(1,len(model_history.history['val_loss'])+1),model_history.history['val_loss'])
    axs[1].set_title('Model Loss')
    axs[1].set_ylabel('Loss')
    axs[1].set_xlabel('Epoch')
    axs[1].set_xticks(np.arange(1,len(model_history.history['loss'])+1),len(model_history.history['loss'])/10)
    axs[1].legend(['train', 'val'], loc='best')
    fig.savefig('plot.png')
    plt.show()

# Define data generators
train_dir = 'data/train'
val_dir = 'data/test'

num_train = 28709
```



```
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(48,48),
        batch_size=batch_size,
        color_mode="grayscale",
        class_mode='categorical')

validation_generator = val_datagen.flow_from_directory(
        val_dir,
        target_size=(48,48),
        batch_size=batch_size,
        color_mode="grayscale",
        class_mode='categorical')

# Create the model
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```


Happy


Happy
Fear

9. REFERENCES

1. https://www.researchgate.net/publication/270583005_A_Review_Paper_on_Face_Recognition_Techniques

2. http://dro.dur.ac.uk/20694/1/20694.pdf

3. https://www.ijcsi.org/papers/IJCSI-9-6-1-169-172.pdf