DEPRESSION DETECTION

USING ARTIFICIAL INTELLIGENCE

REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

Six Months Industrial Training (TR-104)

at

Sabudh Foundation, Mohali

(from July, 2021 to Dec, 2021)

SUBMITTED BY

Sunny Kumar

Information Technology

1805110

Information Technology Department

GURU NANAK DEV ENGINEERING COLLEGE

LUDHIANA, INDIA

CERTIFICATE OF TRAINING

Here I'll attach my 6th Months Industrial Training Certificate.

STUDENT'S DECLARATION

I hereby certify that the work which is being presented in this training report with the project entitled **"Depression Detection using Artificial Intelligence"** by Sunny Kumar, University Roll No. 1805110 in partial fulfillment of requirements for the award of degree of B.Tech. (Information Technology) submitted in the Department of Information Technology at **GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA under I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY** is an authentic record of my own work carried out under the supervision of Arya Bhattacharyya, Jr. Data Scientist of Sabudh Foundation. The matter presented has not been submitted by me in any other University / Institute for the award of B.Tech. Degree.

Student Name: Sunny Kumar

Univ. Roll No: 1805110

Signature of Student

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of Internal Examiner

The External Viva-Voce Examination of the student has been held on _____________.

Signature of External Examiner

Signature of HOD

ABSTRACT

The bursting scope of Artificial Intelligence has been a major source for building our project on depression detection. AI enables researchers to better define mental illness subtypes and understand patient symptoms. Addressing ethical concerns surrounding AI in psychiatry may encourage clinicians to adopt the technology. Depression is one of the serious conern in today's time. Our motive is to built an end to end multimodal system to detect this serious concern in order to save many lives. The AI techniques used in our project helps to enhance the speed, precision and effectiveness of human efforts by automating and detecting the causes at an initial stage and get cured with a medical treatment.

ACKNOWLEDGEMENT

I am highly grateful to the Dr. Sehijpal Singh , Principal, Guru Nanak Dev Engineering College, Ludhiana, for providing this opportunity to carry out Six Months industrial training at Sabudh Foundation, Mohali.

The constant guidance and encouragement received from Dr. K. S. Mann, HoD, GNDEC Ludhiana has been of great help during the training and project work and is acknowledged with reverential thanks.

I would like to express a deep sense of gratitude and thanks profusely to Dr. Sarabjot Singh Anand, Director of Sabudh Foundation. Without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

The help rendered by Mr. Arya Bhattacharyya, Jr. Data Scientist (Sabudh Foundation) for experimentation is greatly acknowledged.

I express gratitude to other faculty members of Information Technology department of GNDEC for their intellectual support throughout the course of this work.

Finally, I am indebted to all whosoever have contributed in this report work and friendly stay at Sabudh Foundation.

Sunny Kumar

List of Figures

List of Algorithms

List of Tables

List of Abbreviations

Abbreviation	Full Form
LSTM	Long-Short Term Memory
BVP	Blood Volume Pulse
EAR	Eye Aspect Ratio
ECG	Electrocardiogram
ANN	Artificial Neural Networks
CBEM	Content Based Ensemble Model
FERM	Facial Emotion Recognition Models
AAM	Active Appearance Models
MIL	Multiple Instance Learning
DCNN	Deep Convolutional Neutral Networks
TDA	Topological Data Analysis
DOCC	Damped Oscillator Cepstral Coffecients
SVM	Support Vector Machine
FER	Face Emotion Recognizer
CNN	Convolutional Neural Network
MTCNN	Multi-task Cascade Convolutional Neural Network

Contents

1 Introduction

1.1 Introduction to Organization

Sabudh Foundation - An NGO that applies data science for social good. Sabudh Foundation is formed by the leading data scientists in the industry with the objective to bring together data and young data scientists to work on focused, collaborative projects for social benefit. Sabudh foundation is working on solving the real and high impact problems in areas such as education, governance, healthcare, and agriculture using Artificial Intelligence and Machine Learning techniques. Data science can be used across a number of industries in order to be beneficial for the society. For example in agriculture, there are now Agrobots and drones being used to gauge the health of the harvest that can help farmers improve their crop yield and reduce costs. With the help of advanced technologies, we're able to save 90 The foundation has taken steps to involve Colleges, Universities, and Industry from the region for the social cause. Particularly, the foundation has signed academic and researchbased MoUs with Panjab University, Chandigarh, GNDEC, Ludhiana, BML Munjal University, Punjab Government (Punjab Police), Punjabi University, Patiala, and Punjab Engineering College, Chandigarh.

1.2 Introduction to Project

Depression being one of the most common mental disorders that is still not well understood in both research and clinical practice tilll date. Not all patients suffer with the same symptoms thus its a bit difficult to diagnose such an illness. More seriously, Depression can lead to self-mutilation and suicide behaviors. According to the World Health Organization in 2017, there are about 350 million depressive patient worldwide and depression will become the second leading cause of death by 2030. Depression and anxiety disorders are highly prevalent world wide. But fortunately it is treatable, people with depression and anxiety also have increased absenteeism and presenteeism rates as well as low productivity resulting in decreased work performance.Therefore, we are building up an effective AI system that helps to detect depression in patients at an early stage.

1.3 Project Category

- Web Application
- System Development
- Internet based

1.4 Objectives

- To use artificial intelligence for detection of depression in patients.
- To design an end to end encrypted model that helps to rectify mental disorder.
- To monitor the progress of a patient by observing their interactive intellectual and behavior in order to draw general conclusions.

1.5 Problem Formulation

This project is built in association with the organization known as "lifeback". This organization is headed by one of the doctor at RML hospital. This organization demanded to make an end-to-end AI based system so that they can rectify whether a person is depressed or not based on the data provided to us. The data comprises of varied audios and videos as a sample for both healthy and depressed people.

1.6 Identification/Reorganization of Need

Identify whether person is depressed or healthy. Organise data based on voice modulation, facial expression, blinking, eyeball detection, unusual gesture, expressions.

1.7 Proposed System

The system is proposed in a way that it deals with a general healthcare issue that is to be resolved with a positive outcome on a large scale to benefit the society. The system is built in such a way that it enables the patients to interact and coordinate well with their assisted doctors. This end to end system provides proper medication, track record of patients, regular mental health care checkups, text and audio visualizations and a monitored healthcare guidance for its patients.

1.8 Unique Features of the System

In order to built our model we extracted varied features like: Video visualization, Voice/Audio modulation, Expression fluctuation, Unsual gesture detection, Blink detection model, Eyeball detection and Transcription of Audios.

2 Requirement Analysis and System Specification

2.1 Feasibility study

Artificial intelligence has been the overpowering technology to be used to combact "Depression", the leading cause of death in today's time. With the increased depression rate, detecting this issue more accurately and conveniently has been a major concern. Keeping this in mind, healthcare practioners are in search for more feasible techniques and ways to detect depression at an early stage. By the usage of AI driven techniques and algorithms, many obstacles have been cut down in order to establish an end to end system in our project. There is a huge issue with Ethics in AI. Sometimes an algorithm learns that a person belonging from particular race, gender or religion has certain characteristics this is a kind of training error or bias that occurs in AI and this is an ethical issue. The basic platform used in it for coding was google collab as its more convenient and directly connected with drive. Data transfer and enormous storage was ethically managed by google drive. FER model with a simplified (MTCNN) architecture added more effect to our model. An efficient and collaborative environment with maximum resources and least expenditure defines the simplicity and feasiability of our model.

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new device and debugging old running device/system. All system is feasible if they are unlimited resources. There are aspects in the feasibility study portion of the preliminary investigation:

2.1.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to work properly?
- Are there technical guarantees of accuracy, reliability, case of access and control?

2.1.2 Operational Feasibility

Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there controlled efficiently?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undetermined the possible application benefits?

2.1.3 Economical Feasibility

In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. But if we consider the economic feasibility of our system was below the nominal expenses. Therefore economically our system was feasible enough.

2.2 System Requirement Specification

Minimum Software required

- Win 10 or Ubuntu 18.04
- Python Environment/Google Collab
- Python Editor (Visual Studio Code)
- Python Libraries

Minimum Hardware required

- Intel dual core i5
- 512 GB Hard disk
- 12 GB RAM

2.3 Expected hurdles

- While accessing the files from AWS S3 bucket we faced the issue regarding the read permissions.
- Run time disconnection in google collab while working with video data is quite difficult.
- After Audio slicing when we try to generate melspectrograms at some point it was a bit difficult to access the audio files mounted on google drive.
- It was diffcult to send the json responses from server to client due to flaws with json serialization and primary key (object id).
- Due to CORS (Cross-Origin Resource Sharing) error, we could not share the files over the network (server to client).

3 System Design

3.1 User Interface Design

Our system is designed in such a way that it enables the doctor to manage appointments with the patients through a web interface with respect to the data provided by the hospital. The doctor can even track the health status of the patient depending upon the level of his/her mental condition.

3.1.1 Login Profile

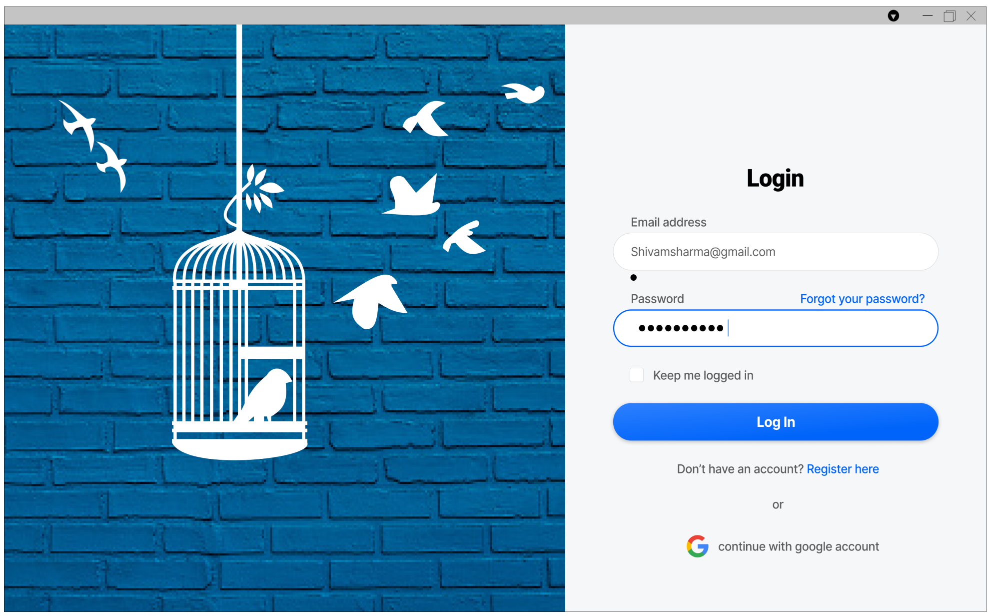


Figure 1: Login Page

3.1.2 Doctor's Profile

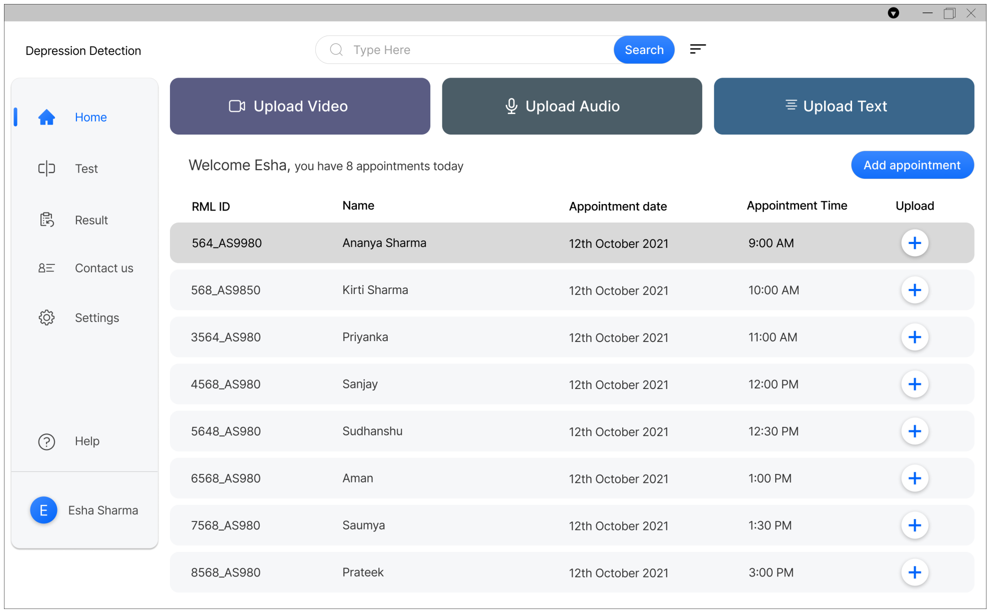


Figure 2: Doctor's Profile

3.1.3 Patient's Profile

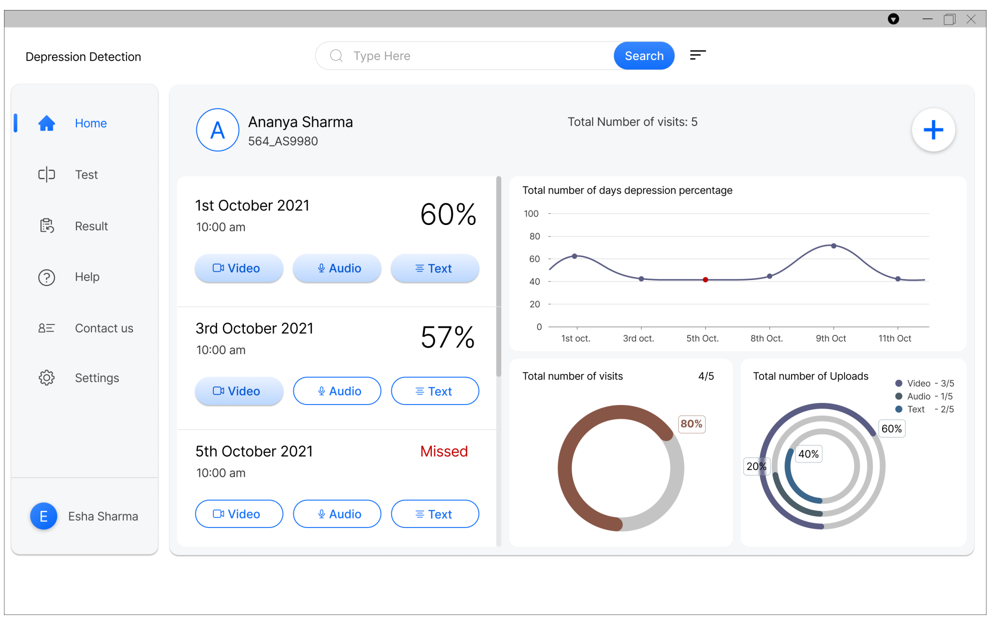


Figure 3: Patient's Profile

3.2 Database Design

PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible. For example, you can define your own data types, build out custom functions, even write code from different programming languages without recompiling your database. PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The data alongside is a collection of varied audios and mel spectrograms. Altogether this data is uploaded on postgreSQL and further on it is linked with the AWS server. The details like audio name, audio URLs, s3-keys, and so on are fetched from the buckets.

Data Output Explain Messages Notifications

	id [PK] integer	audio_name text	audio_s3_key text	audio_url text	image_name text
1	1	AI_101_result_slice_0.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_0.png
2	2	AI_101_result_slice_1050000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_1050000.png
3	3	AI_101_result_slice_1120000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_1120000.png
4	4	AI_101_result_slice_140000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_140000.png
5	5	AI_101_result_slice_210000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_210000.png
6	6	AI_101_result_slice_280000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_280000.png
7	7	AI_101_result_slice_350000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	M Mel_AI_101_result_slice_280000.png
8	8	AI_101_result_slice_420000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_420000.png
9	9	AI_101_result_slice_490000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_490000.png
10	10	AI_101_result_slice_560000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_560000.png
11	11	AI_101_result_slice_630000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_630000.png
12	12	AI_101_result_slice_70000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_70000.png
13	13	AI_101_result_slice_700000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_700000.png
14	14	AI_101_result_slice_770000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_770000.png
15	15	AI_101_result_slice_840000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_840000.png
16	16	AI_101_result_slice_910000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_910000.png
17	17	AI_101_result_slice_980000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_101_result_slice_980000.png
18	18	AI_102_result_slice_0.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_102_result_slice_0.png
19	19	AI_102_result_slice_1050000.mp3	s3://lifeback/depressed_...	https://lifeback.s3.ap-so...	Mel_AI_102_result_slice_1050000.png

Figure 4: Database Design

3.3 Database Connection and Backend Scripting

Flask is a web framework, it's a Python module that lets you develop web applications easily and has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine. As applications grow larger and more developers work on them, structuring and naming things like routes becomes something that needs to be standardized. While REST is much more than route standardization (it is a standard for building web applications), one of the ideas is centering applications around resources and naming the routes for those resources appropriately (we call that RESTful routing). Below is the CRUD operations performed for the database. The abbreviation CRUD expands to Create, Read, Update and Delete. These four are fundamental operations in a database. In the sample database, we will create it, and do some operations.

3.3.1 Getting started

Let's build our first CRUD app with Flask. To get started we first need a resource. Since we will be creating a file, we'll start by making a file called abc.py which will store a simple file class. In order to make sure that we can uniquely identify each file, we will add a property called id that increments by one anytime a file is created. Now let's create an app.py file to start our server with some sample data. When the user visits the route /add user, let's start by creating an index route where we will return a template that shows all of our data. Following are the peformed CRUD operations.

3.3.2 Add User Input

The create function allows users to create a new record in the database. In the SQL relational database application, the Create function is called INSERT. In Oracle HCM Cloud, it is called create. Remember that a record is a row and that columns are termed attributes. A user can create a new row and populate it with data that corresponds to each attribute, but only an administrator might be able to add new attributes to the table itself. Before we discuss what the modified route will look like, we should think about what we want to happen when we submit the form. Once we have finished creating a file, it would be a bit silly to render another HTML page telling us that we just created a file. Instead, it would make more sense to go back to the index page and see an updated list of all the file. So, how can we make another request to send us the index page? To do that we have to introduce a concept called "redirecting."

Redirect: A redirect is actually two separate requests:

- First, the server sends a response with a header called 'location' with a value that is a route
- The browser recieves the response and immidiately issues a new request to the route provided in the location header
- If the route exists on the server, the server responds accordingly (or returns a 404 status code (page not found)). To do this with Flask, we need to import redirect. We will also import url_for so that we do not have to "hard code" our routes, as well as request which we will use to collect data from a form.

Notice that we had to specify in our @app.route for file that our application should accept both GET and POST. Then, inside of index, we can handle each case separately.

Algorithm 1 Add User Input

```
@app.route("/add-users", methods=["POST"])
def add_user():
    db = get_database()
    data = request.get_json()
    db.get_collection("users").insert_one({
        "name": data["name"],
        "email": data["email"],
        "contact": data["contact"],
        "password": data["password"],
        "age": data["age"]
    })
    return jsonify({
        "response": "Added"
    })
```

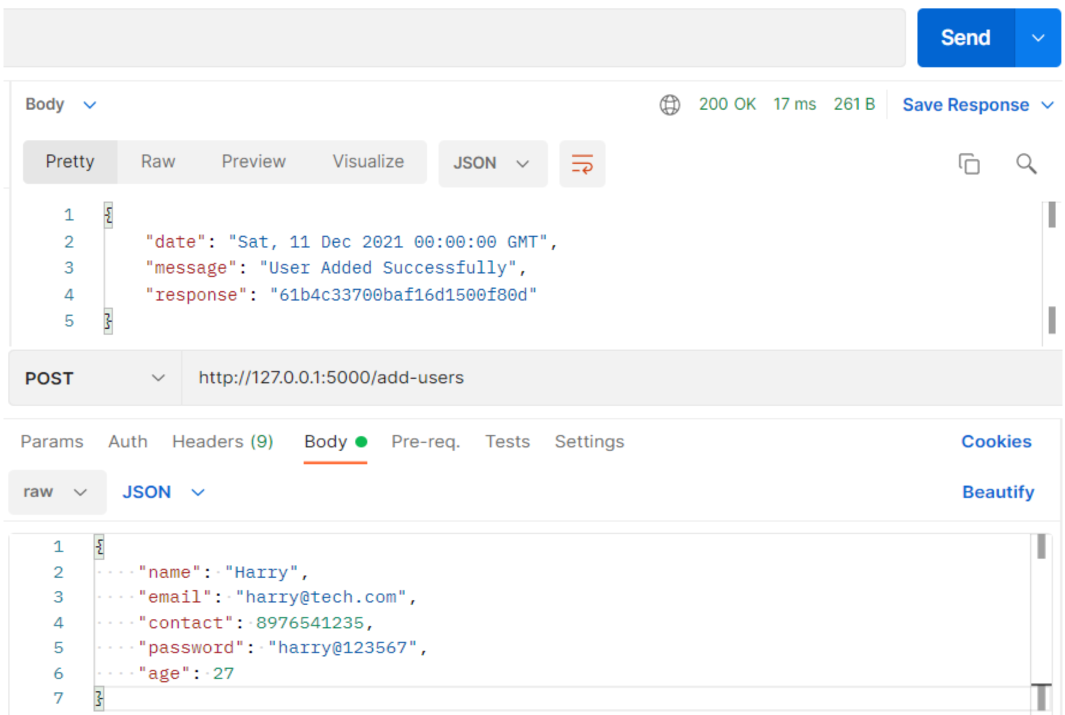


Figure 5: Add User Output

3.3.3 Get User Input

The read function is similar to a search function. It allows users to search and retrieve specific records in the table and read their values. Users may be able to find desired records using keywords, or by filtering the data based on customized criteria. For example, a database of cars might enable users to type in "1996 Toyota Corolla", or it might provide options to filter search results by make, model and year. Now that we are able to read new functions, let's make a route to show some additional information about the functions. In order for this to work, we are going to need a way to find individual functions by their id. Let's make a route that includes a dynamic parameter called id which is an integer. Let's call the function that this route triggers show. Now let's create a simple page that shows more information about the file. Let's also add a link for us to edit a file. We will call this function edit, and since we need to specify which toy to edit, we will pass an id of a file to this url.

Algorithm 2 Get User Input

```
@app.route('/get-users', methods=["GET"])
def get_users():
    db = get_database()
    items = db.get_collection("users").find()
    array = [item for item in items]
    print(array)
    return "hello"
```

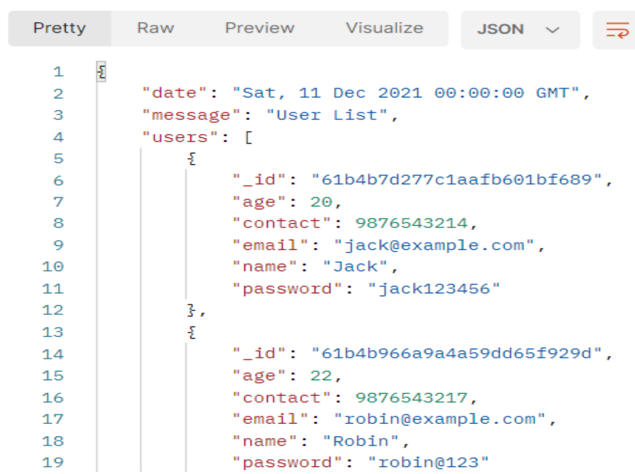


Figure 6: Get User Output

3.3.4 Delete User Input

The delete function allows users to remove records from a database that is no longer needed. Both SQL and Oracle HCM Cloud have a delete function that allows users to delete one or more records from the database. Some relational database applications may permit users to perform either a hard delete or a soft delete. A hard delete permanently removes records from the database, while a soft delete might simply update the status of a row to indicate that it has been deleted while leaving the data present and intact. By now you should be able to perform CRU completely within your browser. Next, let's put the D into CRUD by building out the delete functionality. In our edit.html we are going to add another form to delete a file. This is done with a form since we need to make a DELETE request.

Algorithm 3 Delete User Input

```
@app.route("/delete-users/<id>", methods=["DELETE"])
def delete_users(id):
    db = get_database()
    db.get_collection("users").delete_one({"_id": ObjectId(id)})
    return "User deleted successfully"
```

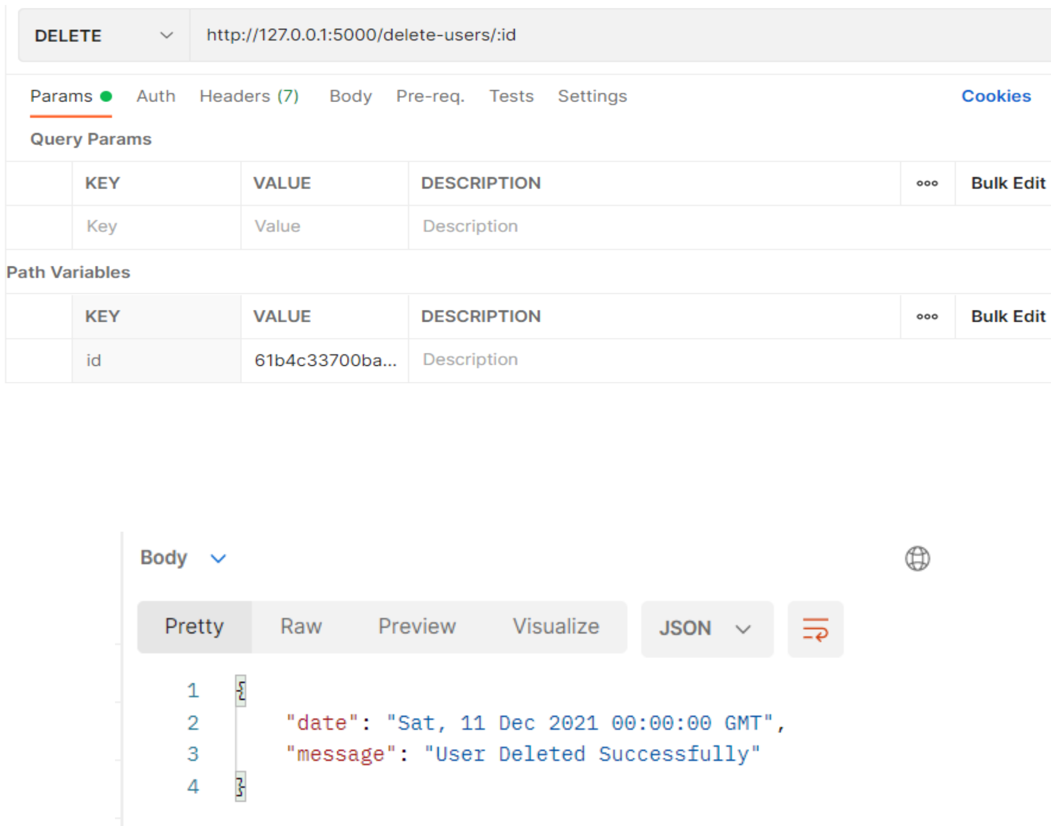


Figure 7: Delete User Output

3.3.5 Update User Input

The update function is used to modify existing records that exist in the database. To fully change a record, users may have to modify information in multiple fields. For example, a restaurant that stores recipes for menu items in a database might have a table whose attributes are "dish", "cooking time", "cost" and "price". One day, the chef decides to replace an ingredient in the dish with something different. As a result, the existing record in the database must be changed and all of the attribute values changed to reflect the characteristics of the new dish. In both SQL and Oracle HCM cloud, the update function is simply called "Update". In order to edit a file we need to first make sure we have a route that renders a form for editing. Before we can edit a file, though, we first need to render a page with a form to edit the file.

Algorithm 4 Update User Input

```
@app.route("/update-users/<id>", methods=["PUT"])
def update_user(id):
    db = get_database()
    data = request.get_json()
    db.get_collection("users").replace_one(
    { "_id": ObjectId(id) },
    {
        "name": data["name"],
        "email": data["email"],
        "contact": data["contact"],
        "password": data["password"],
        "age": data["age"]
    })
    return jsonify({
        "response": "updated"
    })
```

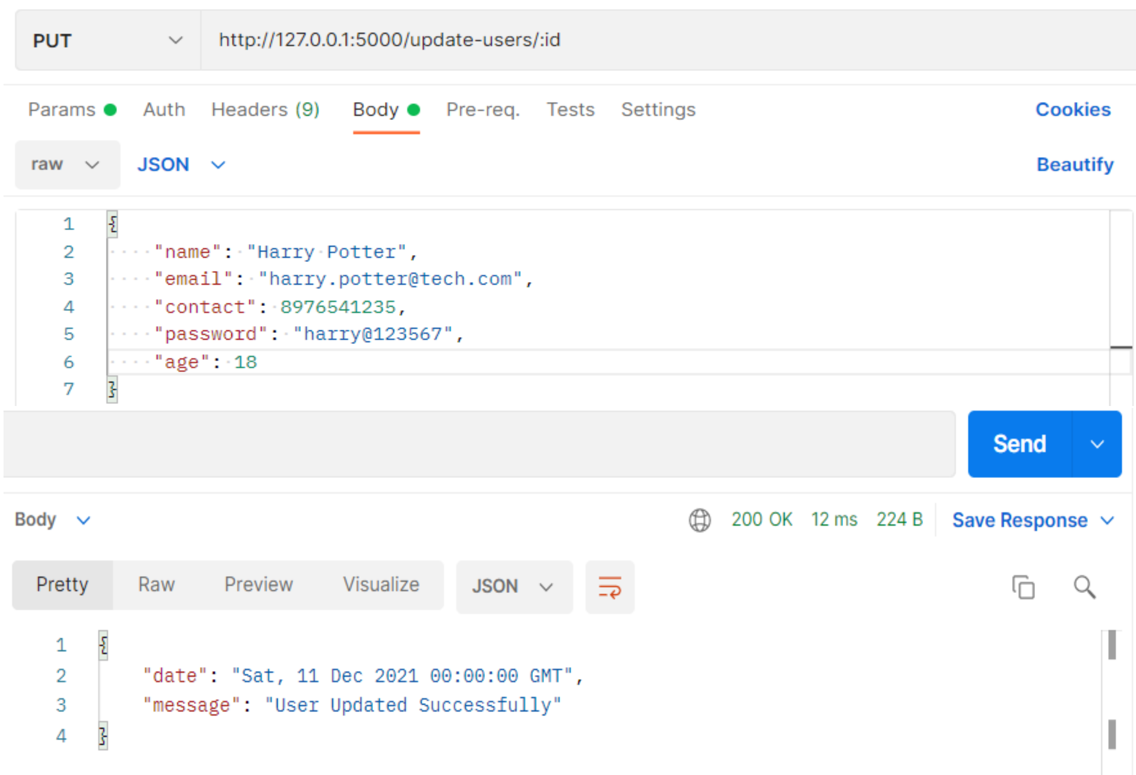


Figure 8: Update User Output

3.3.6 HTTP Requests Log

An HTTP request is made by a client, to a named host, which is located on a server. The aim of the request is to access a resource on the server. To make the request, the client uses components of a URL (Uniform Resource Locator), which includes the information needed to access the resource. HTTP Logging is a middleware that logs information about HTTP requests and HTTP responses. HTTP logging provides logs of: HTTP request information, Common properties and Headers.

```
 * Serving Flask app 'app.py' (lazy loading)
 * Environment: development
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [11/Dec/2021 20:56:47] "POST /add-users HTTP/1.1" 200 -
127.0.0.1 - - [11/Dec/2021 20:58:46] "PUT /update-users/61b4c33700baf16d1500f80d HTTP/1.1" 200 -
127.0.0.1 - - [11/Dec/2021 20:59:54] "PUT /delete-users/61b4c33700baf16d1500f80d HTTP/1.1" 405 -
127.0.0.1 - - [11/Dec/2021 21:00:00] "DELETE /delete-users/61b4c33700baf16d1500f80d HTTP/1.1" 200 -
```

Figure 9: HTTP Requests Log

3.4 Methodology

3.4.1 Literature Review

Literature review played a significant part in building up a basic end to end artificial intelligence based depression detection model. Going through various Literature Surveys, it enables us to create a groom and technically skillful Data Science live working model. After reviewing various research papers, we conclude the summary of those research papers below:-

From the paper "Topological Data Analysis to Engineer Features from Audio Signals for Depression Detection" we conclude that by Applying TDA to timeseries allows for the extraction of features relating to the shape of the data.we believe the DA features are more robust to the variance of audio expression between participants which is one of the main challenges in depression detection from audio [1]. From the paper "Identifying Depression in a Person Using Speech Signal" by Extracting Energy and Statistical Features. The deification of depressed people has successfully done with average 81.567% of test set audio with no over fitting [2].

From the paper "The Detection of Depression Using Multimodal Models Based on Text and Voice Quality Features."The Text analysis model & Voice quality model performs detection of depression on word-level using a transcript of the individual's interview with a virtual interviewer. They had created a voice quality analysis model that uses five glottal flow voice features [3]. From the paper "Multi-Modal Depression Detection and Estimation" This paper has used audio and/or visual data to address depression. Compared to audio and visual cues, few rescarch focus on the higher level feature - the language information. Depression Classification and Depression Estimation are considered at the same time, better performance could be obtained [4].

From the paper "Multimodal Spatiotemporal Representation for Automatic Depression Level Detection" LPQ feature was extracted from each frame of a video segment and used the mean of these features as the video segment-level feature in the competition of AVEC2013 [5]. From the paper "Hybrid CNN-SVM classifier for efficient depression detection system" they have used novel audio based approach to automatically detect depression using hybrid model. This model combines convolutional neural networks (CNN) and support vector machines (VM), where SVM takes the place of the fully connected layers in CNN [6].

From the paper "Extraction of Facial Features as Indicators of Stress and Anxiety" The result directs at initial study that aims to find an effective approach of stress and anxiety assessment using facial signs consisting of the mouth activity, head motion, heart rate, blink rate and eye movements. Several features have been analyzed, which had been calculated by a set of different algorithms, each targeting a specific facial region [7]. From the paper "Automatic Depression Level Detection through Visual Input" he key idea was to develop a video-based decision network system that can detect the depression of the user. The system provides the result to the user in the form of document consisting the detected depression level. The convolutional 3D model resulting in detection of salient features from an input image to provide an emotion vector categorized as: Angry, Sad, Happy, Surprise, Fear and Neutral. the classification of the user into one out of 4 levels: Minimal level, Mild level, Moderate level, or Severe level depression is achieved [8].

3.4.2 Comparison

This section should have table where you need to compare the results from different prominent research of area of interest. Below mentioned Research papers were compared in order to get the most efficient one.

Table 1: Literature Review

Sr. No.	Research Paper	Year of Publication	Dataset used	Model Used
1	TDA to Engineer Features from Audio Signals for Depression Detection	2020	DAIC-WOZ, Moodable/EMU	Betti curve & TDA
2	Automatic Depression Level Detection through Visual Input	2020	FER2013	CNN model
3	Identifying Depression in a Person Using Speech Signal	2020	RAVDESS	Feed Forward Neural Network
4	The Detection of Depression Using Multimodal Models Based on Text and Voice Quality Features.	2021	DAIC-WOZ	Text analysis model & Voice quality model
5	Multi-Modal Depression Detection and Estimation	2019	DCGAN and DAIC-WOZ	DCNN
6	Multimodal Spatiotemporal Representation for Automatic Depression Level Detection	2020	LLDs and MFCCs	CNN and LSTM
7	Automatic Depression Detection Via Facial Expressions Using MIL	2020	DAIC-WOZ	LSTM and MIL
8	Hybrid CNN-SVM classifier for efficient depression detection system	2020	DAIC-WOZ	Hybrid CNN-SVM model
9	A Novel Approach for Depression Detection using Audio Sentiment Analysis	2021	DAIC-WOZ	Audio Sentiment Recognition & CNN Architecture
10	Automated Depression Detection using Audio Features	2020	DAIC-WOZ	Voice quality model
11	Improved Classification Model for Depression Detection Using EEG and Eye Tracking Data	2021	Lanzhou University Second Hospital, the resting state EEG and eye movement data	CNN and LSTM architectures, CBEM, Dynamic & Static Model
12	Extraction of Facial Features as Indicators of Stress and Anxiety	2015	DAIC-WOZ	AAM, FERM
13	Hybrid CNN-SVM classifier for efficient depression detection system	2020	DAIC-WOZ dataset/AVEC 2016	Hybrid CNN-SVM model
14	Analysis of Physiological Responses from Multiple Subjects for Emotion Recognition	2012	3rd and 4th AVEC	BVP and the ECG sensors.
15	Noise and Reverberation effects on Depression Detection from Speech	2016	AVEC-2014	SVR model and ANN for each feature type and training condition

3.4.3 Dataset / Exploratory Data Analysis

We are using the Dataset provided by RML, which consists of depresed and healthy audio, vedio, of (90-95) patients, with embedded audios in vedios and the length of the video is 15-20 minutes each. The video consist of a doctor and a patient, when a doctor is intervewing the patient about their mental health. Lifestyle, choices, recent happening were asked in the interview which gives the narration or overview of their lives in the last few months.

In all total our dataset consists of positive skewed audios that is biased in comparison to depressed people audios. The table below represents the number of Audio sample provided as a part of the Dataset we used to train our model:-

Table 2: EDA of Audio Samples

AUDIO SAMPLE NAME	NUMBER OF SLICES	ACTUAL LABEL	PREDICTED LABEL
AIC 304 (HEALTHY)	20	100% HEALTHY	100% HEALTHY
AI 159 (DEPRESSED)	25	100% DEPRESSED	40% DEPRESSED

3.4.4 Multimodal Depression Detection using Deep Learning Algorithm

As multimodal approaches are found to be effective in depression analysis as depression is a multifactor disorder. So a unimodal approaches in depression detection will only give some information without considering the important features. So the future works should follow a multimodal approach and it has been found that deep learning based classification gave significant improvement in accuracy compared to traditional machine learning approaches. It is found that the multimodal performs much better in the detection of depression as it considers several factors for the classification of emotions. Single modality may give only one sided information or may miss out an inherent parameter.

Our model will be based on three basic factors using which we will diagnose depression are as follows:-

- Audio
- Video
- Text

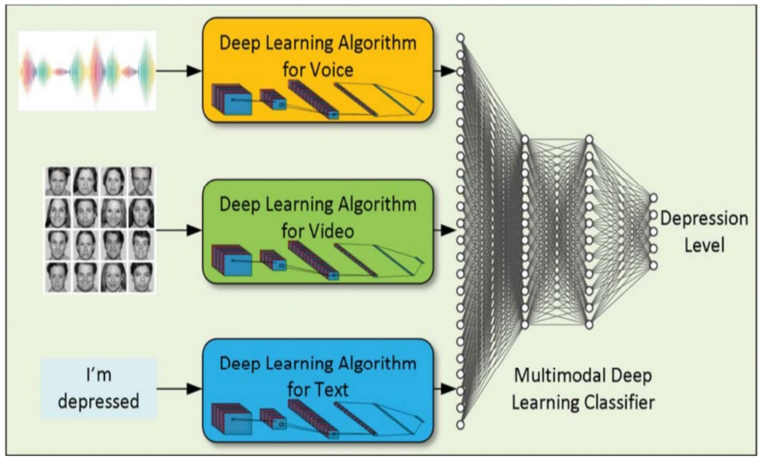


Figure 10: Multimodal Depression Detection using Deep Learning Algorithm

3.4.5 Audio based Model

Audio has been successfully used in the detection of sentiment for a long time. Via our literature survey, we came to the conclusion that audio has a rich history of application in the area of emotion detection with proven results. Hence, our first endeavour was targeted at audio-based AI models. Since emotion detection is a similar classification problem, we have used some characteristics of those models. After thorough experimentation with various approaches involving complete audio vs audio slices (with various intervals) and MEL features vs MFCC features among others, the following conditions were giving us the best results:

1. **Features Used:** MEL Spectrograms of audio slices. The before mentioned audio slices are of duration 80 seconds with an overlap of 10 seconds with the content of the previous audio sample. Other slicing durations were tried ranging from an overlap period between 5 seconds and 20 seconds with the total duration of the sample ranging from 20 seconds to 180 seconds.

2. **Model Used:** A simple CNN-based architecture was used with only 2 layers of convolution, 1 flatten layer, and 2 dense layers. The dropout was set at 20% and MaxPooling layers were added after the convolutional layers.

3. **Accuracy:** The accuracy that we got with the model was 85% over the validation set.

We've some plots of the MEL & MFCC features and also a waveform that illustrates the slicing procedure we have described earlier.

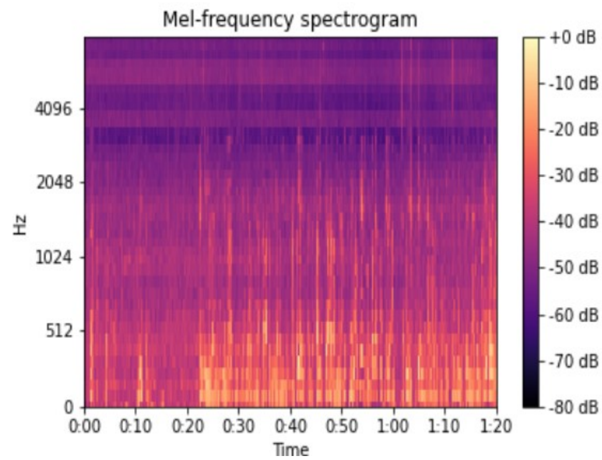


Figure 11: MEL spectrogram

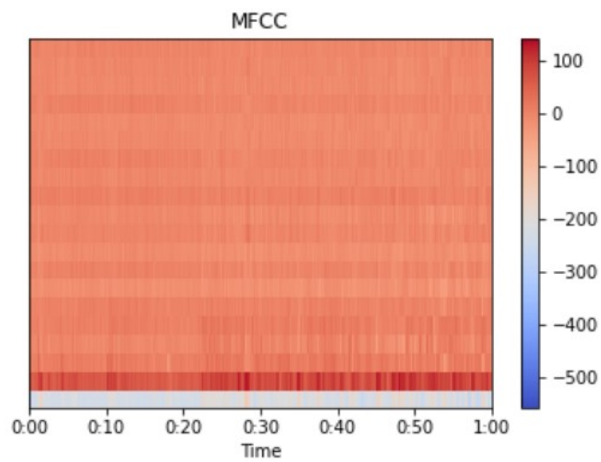


Figure 12: MFCC analysis of the Audio file

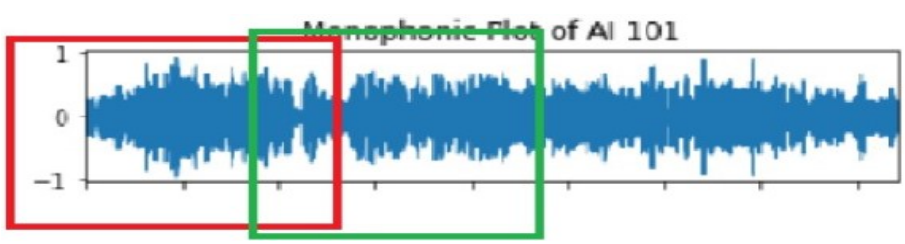


Figure 13: Slicing mechanism of Audio

3.4.6 Video based Model

Artificial Intelligence combines various strategies to distinguish depression in a person through visual or vocal narratives.

(i) Blink Detection Model

We have built a model that is capable of detecting and counting blinks in video streams using facial landmarks and OpenCV. The model is performing well for grayscale videos having dimensions of 640 X 360. We have also tried for RGB videos but the model is not performing as well which is why we converted RGB videos to grayscale. The following conditions were giving us the best results:

- **Features Used:** To build the blink detector we have computed a metric called the eye aspect ratio (EAR). Once the person blinks the eye aspect ratio rapidly drops close to zero, then increases again, indicating a single blink has taken place. The video slices are of duration 30 seconds with an overlap of 10 seconds. Other slicing durations were tried for example, 70 sec with an overlap of 10 seconds.
- **Model Used:** shape_predictor_68_face_landmarks.dat model creates the predictor object that takes in an image region containing some object and outputs a set of point locations that define the pose of the object. dlib.get_frontal_face_detector was used for detecting the faces in a frame.
- **Accuracy:** The accuracy was pretty good in each frame.

(ii) MTCNN Model

The architecture of MTCNN is mentioned below:

Stage 1: The Proposal Network (P-Net)

This first stage is a fully convolutional network (FCN). The proposal network is used to obtain candidate windows and their bounding box coordinates using regression.

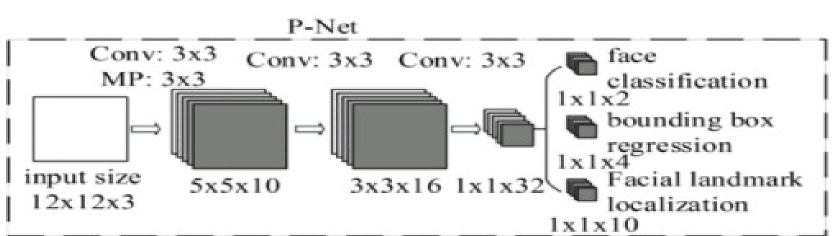


Figure 14: The Proposal Network (P-Net)

Stage 2: The Refine Network (R-Net)

All candidates from the P-Net are fed into the Refine Network, which is a CNN. The R-Net outputs wether the input is a face or not, a 4 element vector which is the bounding box for the face, and a 10 element vector for facial landmark localization.

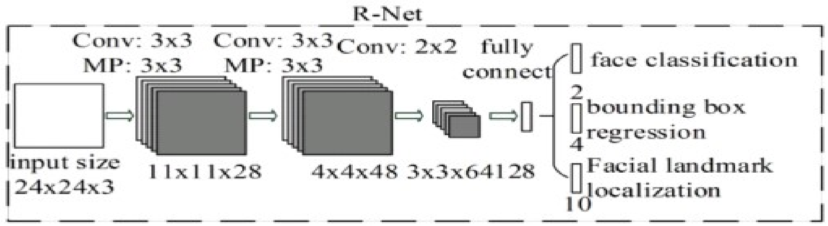


Figure 15: The Refine Network (R-Net)

Stage 3: The Output Network (O-Net)

This stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth.

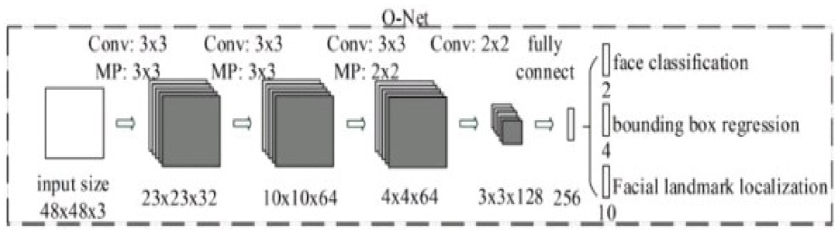


Figure 16: The Output Network (O-Net)

The Three Tasks of MTCNN:

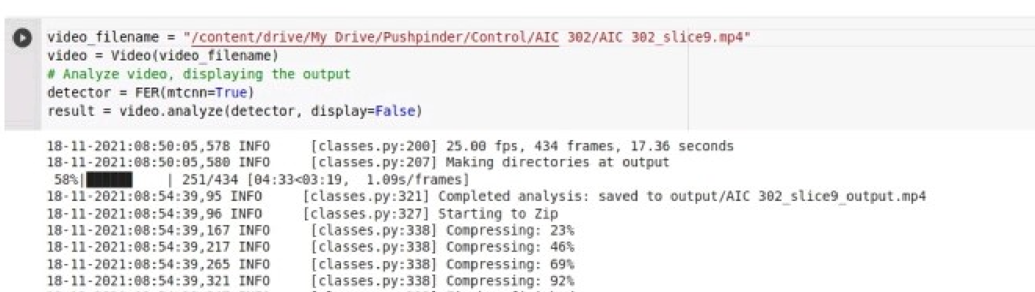
1. **Face Classification:** This is a binary classification problem that uses cross-entropy loss.

2. **Bounding Box Regression:** For each candidate window, the offset between the candidate and the nearest ground truth is calculated. Euclidean loss is employed for this task.

3. **Facial Landmark Localization:** The localization of facial landmarks is formulated as a regression problem, in which the loss function is Euclidean distance.

Accuracy: The accuracy that we got with the model was a promising 95% against the human perception.

The image below shows the execution of the model. It takes on an average 7-8 min to process a slice of around 400 frames:



```
video_filename = "/content/drive/My Drive/Pushpinder/Control/AIC 302/AIC 302_slice9.mp4"
video = Video(video_filename)
# Analyze video, displaying the output
detector = FER(mtcnn=True)
result = video.analyze(detector, display=False)
```

```
18-11-2021:08:50:05,578 INFO     [classes.py:200] 25.00 fps, 434 frames, 17.36 seconds
18-11-2021:08:50:05,580 INFO     [classes.py:207] Making directories at output
 58%|█████     | 251/434 [04:33<03:19,  1.09s/frames]
18-11-2021:08:54:39,95 INFO     [classes.py:321] Completed analysis: saved to output/AIC 302_slice9_output.mp4
18-11-2021:08:54:39,96 INFO     [classes.py:327] Starting to Zip
18-11-2021:08:54:39,167 INFO     [classes.py:338] Compressing: 23%
18-11-2021:08:54:39,217 INFO     [classes.py:338] Compressing: 46%
18-11-2021:08:54:39,265 INFO     [classes.py:338] Compressing: 69%
18-11-2021:08:54:39,321 INFO     [classes.py:338] Compressing: 92%
18-11-2021:08:54:39,347 INFO     [classes.py:339] Zip has finished
```

Figure 17: Execution of model

Every emotion is calculated and the output is put on a scale of 0 to 1 , with values close to 0 indicating complete absence of that emotion and values close to 1 indicating dominant presence of that emotion. These were directly converted to percentage to show what percentage of each emotion is present in that particular slice of video:

```
Emotions : ['angry0', 'fear0', 'happy0', 'neutral0', 'sad0']
Count : [13.82488479262673, 5.0691244239631335, 0.9216589861751152, 77.88018433179722, 2.3041474654377883]
```

Figure 18: Slice of video

The image below contains shows a sample of the final tabulated results, where we have human perception of sliced videos as ground truth:

1		Happy	Neutral	Sad	Angry	Fear	Surprise	Prediction	Ground Truth	Label
2	AI 117_slice1	0.65	20.45	78.89	0	0	0	Sad	Sad, Neutral	Depressed
3	AI 117_slice2	0.97	37.01	62.01	0	0	0	Sad	Sad	Depressed
4	AI 117_slice3	0.65	28.01	71.33	0	0	0	Sad	Sad	Depressed
5	AI 117_slice4	0.65	17.86	81.49	0	0	0	Sad	Sad	Depressed
6	AI 117_slice5	0.7	23.51	75.78	0	0	0	Sad	Neutral	Depressed
7	AI 117_slice6	0	23.78	73.93	2.28	0	0	Sad	Sad, Neutral	Depressed
8	AI 117_slice7	0.32	45.78	53.57	0.32	0	0	Sad	Sad, Neutral	Depressed
9	AI 117_slice8	0.7	29.06	68.51	0	1.73	0	Sad	Sad	Depressed
10	AI 117_slice9	0.32	64.7	34.97	0	0	0	Neutral	Sad	Depressed

Figure 19: Final tabulated results

3.4.7 Transcription of Audios

Since the audio is in Hindi most of the time, we need to have the text in the Devanagari script. For the same, we were initially considering Microsoft Azure Speech to Text. However, upon close inspection of their privacy policy, we came to the conclusion that we cannot use the service and would hence have to rely on manual transcription. Manually transcription was indeed a time consuming task for about 100 to 200 audios. Further, we are looking forward for some softwares that can generate results accurately with respect to the transcription work.

Herein we have a glimpse of the discussion between Doctor and Patient.

Doctor: अच्छा, कितने सालो से नौकरी कर रहे हैं इस विभाग में ?

Patient: उनको,उनको हो गया 15-16 साल हो गया |

Doctor: 15-16 साल हो गया अच्छा | अच्छा, सारा दिन जब पति चले जाते हैं बच्चे चले जाते हैं आज कल तो corona हो गया तो बच्चे घर ही में होंगे |

Patient: हाँ जी |

Doctor: तो सारा दिन करती क्या है आप ?

Patient:घर में बहुत काम है,घर का काम करते हैं थोड़ा सो जाते हैं दिन में |

Doctor: क्या काम करती हैं वैसे ?

Figure 20: Example of Transcript Audio

4 Implementation, Testing and Maintenance

4.1 Introduction to Languages, Tools & Technologies used for Implementation

Then languages that we used in our project is HTML, CSS and ANGULAR as our Frontend and Python as our Backend for API's and backend Scripting. Frontend languages that we used in our project helps build interactive and dynamic web application through its compelling features that include templating, two-way binding and Restful API's handling. Python is a natural choice for the API because of its simplicity and power. For the same reasons, Angular is a great choice on the client side. Angular's use of TypeScript makes it easy to get started with and still powerful enough to handle your most advanced scenarios.

4.1.1 Video based AI models

Artificial Intelligence (AI) combines various strategies to distinguish depression in a person through visual or vocal narratives. Researches show that disheartened people differ in their physiological as well as physical features. Depression shows through an assortment of visual indications. Sudden changes in the movement of facial muscles and electrodermal skin reactions regularly reflect the incessant and persevering negative considerations and sentiments of misery that portray sadness. Visual signs have extensively been investigated for depression detection.

(a) Blink Detection Model

We have built a model that is capable of detecting and counting blinks in video streams using facial landmarks and OpenCV. To build the blink detector we have computed a metric called the eye aspect ratio (EAR). Fig:- The eye is represented by a set of 6 labeled facial points with specific coordinates. Horizontal line is distance between points $p1$ and $p4$ (width of an eye), and vertical line is distance between middle of points $p2$ and $p3$ and middle of points $p6$ and $p5$ (height of an eye.) The length of the horizontal line will always be a constant, while the length of the vertical line will change depending on the opening and closing of the eye. We can detect blinking by calculating the length of these two lines and then finding the ratio between them. This ratio will be approximately constant while the eye is open, and it will quickly fall to zero when a blink occurs. We can calculate the aspect ratio with the following equation:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 21: Eye Aspect Ratio

(b) Eyeball Detection Model

The first thing we have we have done is to find eyes before we can move on to image processing and to find the eyes, we need to find a face. The facial KeyPoint detector takes a rectangular object of the dlib module as input which is simply the coordinates of a face. We have created a new black mask using NumPy of the same dimensions as frame. After doing this we have a black mask where the eye area is drawn in white. This white area is expanded a little using a morphological operation. Thresholding is used to create a binary mask. So, our task was to find an optimal threshold value against which we can segment out the eyeballs from the rest of the eye and then we need to find its center. But the threshold value will be different for different lighting conditions so we can make an adjustable trackbar for controlling the threshold value.so this was the methodology for eyeball detection model.

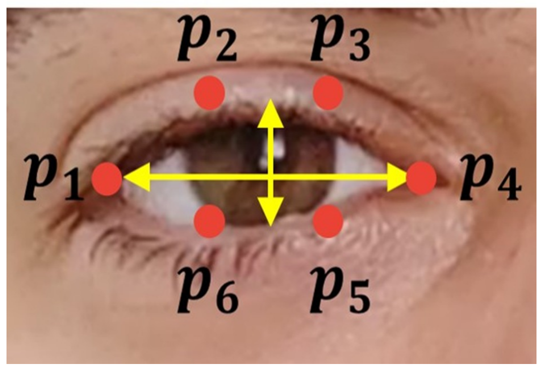


Figure 22: Six labeled facial points of eye with specific coordinates

(c) FER Model

For detecting emotions from videos, we used the Face Emotion Recognizer (FER) library, an opensource library and has dependencies on OpenCV and Tensorflow. It is a CNN based model with weights saved in a hdf5 file present in the source code . This can be overridden by using the FER constructor() when the model is called and initiated. The FER constructor() is initialized by giving it a face detection classifier (either OpenCV Haarcascade or MTCNN) . We used the MTCNN classifier. 80 slices of the videos from the RML data, of 10 seconds each and without overlap, were fed into the model. Out of them, 40 slices each, having a total duration of 200 sec, belonged to 2 healthy controls and 2 depressed subjects. It draws bounding boxes around the detected faces and then classifies the detection of emotion- namely into six categories, namely, ‘fear’, ‘neutral’, ‘happy’, ’sad’, ‘anger’, and ‘disgust’.

4.2 Testing Techniques and Test Plans

4.2.1 Blink Detection Model

Algorithm 5 Calculating Eye Aspect Ratio

```
from scipy.spatial import distance as dist
from imutils.video import FileVideoStream
from imutils.video import VideoStream
from google.colab.patches import cv2_imshow
from imutils import face_utils
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2

def eye_aspect_ratio(eye):
  A = dist.euclidean(eye[1], eye[5])
  B = dist.euclidean(eye[2], eye[4])
  C = dist.euclidean(eye[0], eye[3])
  #Compute the eye aspect ratio
  ear = (A + B)/(2.0 * C)
  return ear

p = '/content/shape_predictor_68_face_landmarks.dat'
v = '/content/drive/MvDrive/Ruhika/27AT_108.mp4'

EYE_AR_THRESH = 0.3
EYE_AR_THRESH_FRAMES = 3

#extract the left and right eye coordinates, then use the
#coordinates to compute the eye aspect ratio for both eyes

leftEye = shape[lStart: lEnd]
rightEye = shape[rStart: rEnd ]
leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)
ear = (leftEAR + rightEAR) / 2.0

#compute the convex hull for the left and right eye, then
#visualize each of the eyes
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
cv.drawContours(gray, [leftEyeHull], -1, (0, 255, 0), 1)
cv2.drawContours(gray, [rightEyeHull], -1, (0, 255, 0), 1)

if ear < EYE AR THRESH:
  COUNTER += 1
#otherwise, the eye aspect ratio is not below the blink threshold
else:
  if COUNTER >= EYE_AR_CONSEC_FRAMES:
    TOTAL += 1
```

Algorithm 6 Determine the facial Landmarks for the face region

```
TOTAL = 0

print ("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector ()
predictor = dlib.shape_predictor (p)
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye" ]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye" ]
print ("[INFO] starting video stream thread...")

vs =  FileVideoStream (v) .start ()
fileStream = True
time.sleep(1.0)

while True:

  #if this is a file video stream, then we need to check if there
  #any more frames left in the buffer to process

   if fileStream and not vs.more) :
     break

  #grab the frame from the threaded video file stream, resize it,
  #and convert it to grayscale channels

  frame = vs.read()
  frame = imutils.resize (frame, width = 450)
  gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

  #detect faces in the grayscale frame
  rects = detector (gray, 0)

  #loop over the face detections
  # reset the eye frame counter
COUNTER = 0
#draw the total number Of blinks on the frame along with the
#computed eye aspect ratio for the frame

cv2.putText (gray, "Blinks: {}".format(TOTAL), (10, 30),
               cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2

cv2.putText (gray, "EAR: {:. 2ft}".format(ear), (300, 30),
               cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2

#show the frame

cv2_imshow(gray)
key = cv2.waitKey(1) & 0xFF
#if the break ‘q‘ key was pressed, break from the loop
if key == ord("q"):
  break
# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

4.3 Test cases used in the Project

4.3.1 Testing of Eye Blink Detection Model

The model is performing well for grayscale videos having dimensions of 640 X 360. We have also tried for RGB videos but the model is not performing as well which is why we converted RGB videos to grayscale.

		Eye Blink Detection using OpenCv		
Dimensions	**Duration**	**Execution Time**	**Accuracy**	**Channel**
1920×1080	17s	2.22	Good	RGB
152×720	8s	1.12	Good	RGB
1280×720	34s	After 3 min runtime disconnected	Buffered data was truncated after reaching the output size limit.	RGB
640×360	1m 10s	After 3.40 sec runtime disconnected	_	RGB
1137×640	1m 10s	After 4.16 sec runtime disconnected	_	GRAY
1137×640	30s	After 3.35 sec runtime disconnected	-	GRAY
640×360	1m 10s	After 5 min runtime disconnected	_	GRAY
854x480	30s	After 3.23 sec runtime disconnected	_	GRAY
640x360	30s	5m	Good	GRAY

Figure 23: Testing of Eye Blink Detection Model on different modes.

4.3.2 Blink Detection Model Sample

The samples are as following:-

Figure 24: Eye blink detection for actual blink.

Figure 25: Non detection Blink.

5 Results and Discussions

5.1 User Interface Representation

Our system is designed in such a way that it enables the doctor to manage appointments with the patients through a web interface wrt the data provided by the hospital. The doctor can even track the health status of the patient depending upon the level of his/her mental condition.

5.1.1 Login Profile

The figure below gives a glimpse of the login page for our website hosted for Lifeback Organization run by RML Hospital.

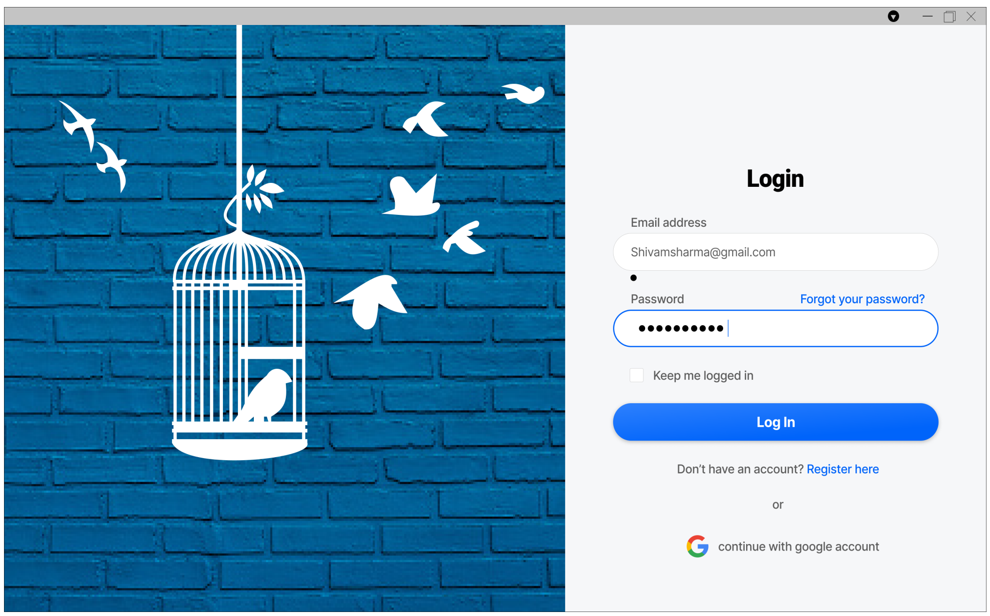


Figure 26: Login Page

5.1.2 Doctor's Profile

The snippet alongside represents the profile and appointment schedule of one of the doctors from RML Hospital. The schedule contains the unique RML IDs, Name of Patient, Appointment date, and time. This pannel even contains a place to upload videos, audio, and text for carrying out further consultation with the patients.

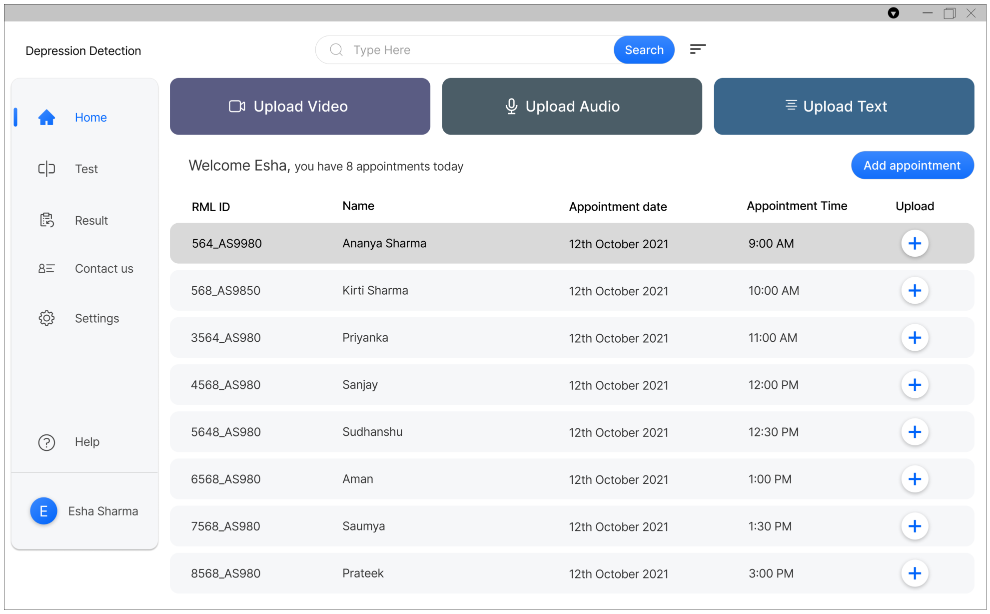


Figure 27: Doctor's Profile

5.1.3 Patient's Profile

The image alongside shows the Patient name with his/her profile depicting the percentage rise and fall in the depression rate of the person each day. It also shows the number of visits and number of uploads made with time.

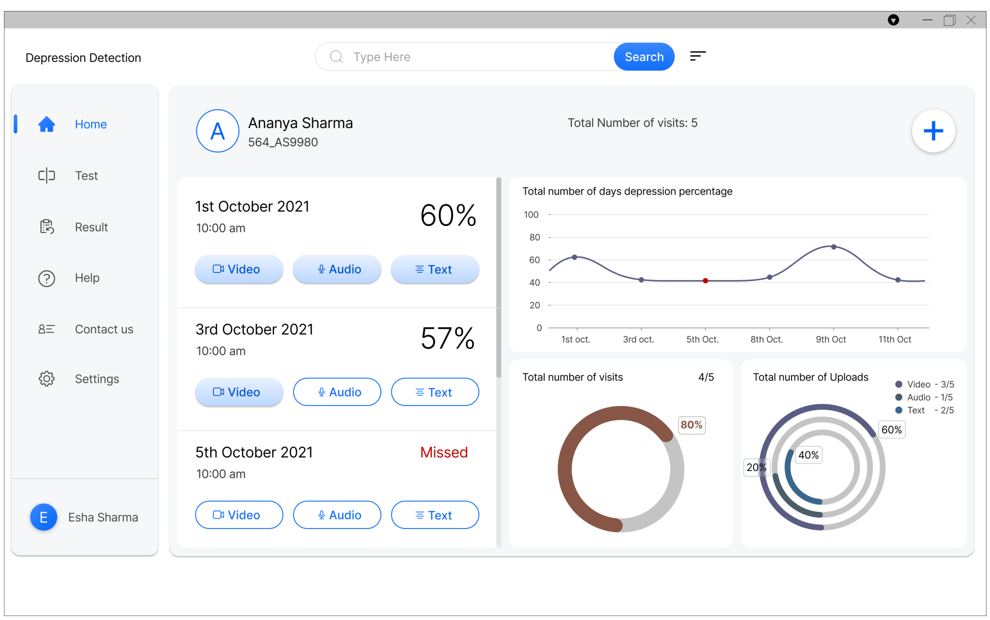


Figure 28: Patient's Profile

After thorough experimentation with various approaches involving complete audio vs audio slices (with various intervals) and MEL features vs MFCC features among others, the following conditions were giving us the best results:

1. **Features Used:** MEL Spectrograms of audio slices. The aforementioned audio slices are of duration 80 seconds with an overlap of 10 seconds with the content of the previous audio sample. Other slicing durations were tried ranging from an overlap period between 5 seconds and 20 seconds with the total duration of the sample ranging from 20 seconds to 180 seconds.

2. **Model Used:** A simple CNN-based architecture was used with only 2 layers of convolution, 1 flatten layer, and 2 dense layers. The dropout was set at 20% and MaxPooling layers were added after the convolutional layers.

3. **Accuracy**: The accuracy that we got with the model was 85% over the validation set.

6 Conclusion and Future Scope

6.1 Conclusion

For blink detection model we could detect and count the number of blinks in each frame and its accuracy was found to be pretty good. The accuracy we observed from FER model was promising 95% against the human perception. The UI/UX part assisted doctors in detecting the level of depression by providing a variety of outcomes based on machine learning algorithms. Doctors can even track and manage the appointments of patients using this system. Transcription of varied audios with reference to healthy and depressed people for detection of depression using ASR to text conversion. There are varied other features that are to be extracted from the depression detection model in the present case. For carrying out these further implementations the project is still under progress.

6.2 Future scope

We have created a feature list in the form of possible questions that we want to ask the data via the use of specialized AI-based models. They belong to the modalities of audio and video. They are as follows:

6.2.1 Audio

1. Vocal prosody, which includes fundamental frequency perceived as pitch, intensity perceived as loudness, and timing perceived as speech rate.

2. The low intensity of speech in depressed patients (Less Energetic).

3. Switching Pause Duration—Switching pause (SP), or latency to speak, is defined as the pause duration between the end of one speaker's utterance and the start of an utterance by the other

4. Very less to no presence of energetic laughter (or even just laughter) in patients.

6.2.2 Video

1. Is the Veraguth & Omega sign appearing?

2. Analysis of Eye Closure Duration Based on the Height of Iris.

3. Pauses while speaking.

4. Inner brow raiser (sadness), brow lowered (sadness), nasolabial furrow deepener (distress), lip corner depressor (distress)

5. Head Pose and Movement Analysis.

6. Eye stare - restricted, shorter eye to eye connection.
7. Is voice pitch energetic or not?
8. Duration of eye remaining close.
9. Extended activity of Corrugator Supercilii muscle.
10. Slumped or limp body posture.

These questions were still not answered while we have pretty good accuracy when it comes to visual emotion recognition and audio classification still, we wanted to ask questions via ML model so that it increases the explain ability of the result that we get and also kind of improve the credibility. for further investigation smaller models can be created wherein all these smaller model outputs will be assign weightages and give out a final prediction based on all the output we have for whether a person is depressed or not.

References

[1] ML Tlachac;Adam Sargent;Ermal Toto;Randy Paffenroth; Elke Rundensteiner 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) Year: 2020 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/9356319

[2] Dimple Muskan Shukla;Kushal Sharma;Sandeep Gupta 2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS) Year: 2020 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/9087223

[3] Hanadi Solieman;Evgenii A. Pustozerov 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) Year: 2021 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/9396540

[4] Le Yang 2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW) Year: 2019 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/8925288

[5] Mingyue Niu;Jianhua Tao;Bin Liu;Jian Huang;Zheng Lian IEEE Transactions on Affective Computing Year: 2020 | Early Access Article | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/9226102

[6] Afef Saidi;Slim Ben Othman;Slim Ben Saoud 2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET) Year: 2020 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/9318302

[7] M. Pediaditis;G. Giannakakis;F. Chiarugi;D. Manousos;A. Pampouchidou;E. Christinaki;G. Iatraki;E. Kazantzaki;P. G. Simos;K. Marias;M. Tsiknakis 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) Year: 2015 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/7319199

[8] Akshada Mulay;Anagha Dhekne;Rasi Wani;Shivani Kadam;Pranjali Deshpande;Pritish Deshpande 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) Year: 2020 | Conference Paper | Publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/9210301

[9] https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/

[10] https://towardsdatascience.com/real-time-eye-tracking-using-opencv-and-dlib-b504ca724ac6

[11] https://www.researchgate.net/publication/259931796_Eye_movement_analysis

[12] https://www.researchgate.net/publication/789931776_Eye_movement_analysis

[13] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4791067/

[14] https://www.nature.com/articles/s41599-020-0499-z

[15] Yifang Yuan;Qingxiang Wang 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA) Year: 2019 | Conference Paper | Publisher: IEEE. [Online]. Available:: https://ieeexplore.ieee.org/document/8964128

[16] https://www.researchgate.net/publication316451292_Analysis_of_Eye_Closure