**ORG MIND- PRIVACY PRESERVING DEPRESSION DETECTION USING MACHINE LEARNING**

## SOCIALLY RELEVANT MINI PROJECT REPORT

***Submitted by***

# NAVEENA H [211423104407]

**NAVEENA SREE [211423104408]**

***in partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

***in***

**COMPUTER SCIENCE AND ENGINEERING**

****

# PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

## OCTOBER 2025

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report **“ORG MIND-PRIVACY PRESERVING DEPRESSION DETECTION USING MACHINE LEARNING”** is the Bonafide work of NAVEENA H [211423104407], NAVEENA SREE [211423104408] who

carried out the project work under my supervision.

**Signature of the HOD with date DR L. JABASHEELA M.E., Ph.D.,**

**PROFESSOR AND HEAD,**

Department of Computer Science and Engineering,

Panimalar Engineering College, Chennai - 123

**Signature of the Supervisor with date Dr K.SANGEETHA M.E.,Ph.D.,**

**PROFESSOR,**

Department of Computer Science and Engineering,

Panimalar Engineering College, Chennai - 123

Submitted for the 23CS1512 - Socially relevant mini Project Viva-Voce Examination held on.........................................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# DECLARATION BY THE STUDENT

We NAVEENA H [211423104407], NAVEENA SREE [211423104408] hereby

declare that this project report titled “**ORG MIND-PRIVACY PRESERVING DEPRESSION DETECTION USING MACHINE LEARNING”**, under the guidance

of Dr.K.SANGEETHA is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

NAVEENA H NAVEENA SREE

# ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for

graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal**, Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator** Dr.K.VALARMATHI and **Project Guide** Dr.K.SANGEETHA and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

**NAVEENA H NAVEENA SREE**

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | i |
|  | **LIST OF TABLES** | ii |
|  | **LIST OF FIGURES** | iii |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview | 1 |
|  | 1.2 Problem Definition | 2 |
| **2.** | **SYSTEM ANALYSIS** | 4 |
|  | 2.1 Existing System | 4 |
|  | 2.2 Proposed System | 5 |
|  | 2.3 Development Environment | 9 |
| **3.** | **SYSTEM DESIGN** | 14 |
|  | 3.1. UML Diagrams | 14 |
|  | 3.2 Data Dictionary | 16 |
|  | 3.3 Activity Diagram | 18 |
|  | 3.4 Data Flow Diagram | 19 |
| **4.** | **SYSTEM ARCHITECTURE** | 20 |
|  | 4.1 Architecture Overview | 20 |
|  | 4.2 Module Description | 25 |
|  | 4.3 Pseudo code / Algorithm Description | 30 |
| **5.** | **SYSTEM IMPLEMENTATION** | 35 |
|  | 5.1 Module-wise Breakdown | 35 |
| **6.** | **SYSTEM TESTING** | 40 |
|  | Test Cases & Reports | 40 |
| **7.** | **CONCLUSION** | 44 |
|  | Conclusion and Future Enhancements | 44 |
|  | **APPENDICES** | 48 |
|  | A.1 SDG Goal | 48 |
|  | A.2 Source Code | 49 |
|  | A.3 Screenshots | 51 |
|  | **PLAGIARISM REPORT**  **REFERENCES** | 57  58 |

**ABSTRACT**

To address the limitations of traditional, subjective methods for diagnosing Major Depressive Disorder (MDD), this project developed an objective, multimodal system using Artificial Intelligence and Machine Learning (AIML) for early detection. The system integrates three distinct data streams: physiological EEG brain signals, behavioral cues from facial expressions, and psychometric data from standardized surveys. These inputs are processed by independent "expert" models, and their predictions are intelligently combined using a late-fusion architecture to generate a single, comprehensive assessment.The true strength of this multimodal system lies in its ability to provide a holistic assessment by correlating objective physiological data, observable behavioral cues, and subjective self-reported experiences. This approach effectively mitigates the weaknesses of any single data source, such as a user consciously masking their emotional state, which the EEG and survey models can help to identify and correct for. In conclusion, this project successfully demonstrates that a multimodal, AIML-driven system serves as a powerful and nuanced proof-of-concept for the early and accurate detection of depression. A key technical achievement was the facial recognition model, which attained a high accuracy of **83.76%** on the test set. The system's true strength is its ability to provide a holistic diagnosis by correlating multiple data points, effectively overcoming the weaknesses of any single source, such as a user consciously masking their emotions. In conclusion, this work serves as a powerful proof-of-concept, demonstrating that a multimodal approach can significantly enhance the accuracy and reliability of depression detection.

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE DESCRIPTION** | **PAGE NO.** |
| 3.1 | Data Dictionary | 16 |
| 6.1 | Test Case 1 | 40 |
| 6.2 | Test Case 2 | 41 |
| 6.3 | Test Case 3 | 42 |
| 6.8 | Future Improvements | 43 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG NO** | **FIGURE DESCRIPTION** | **PAGE NO** |
| 2.1 | Proposed Diagram | 8 |
| 3.1 | Use Case Diagram | 14 |
| 3.2 | Class Diagram | 15 |
| 3.3 | Activity Diagram | 18 |
| 3.4 | Dataflow Diagram | 19 |
| 4.1 | System Overview | 24 |
| A.1 | Dataset upload (csv files) | 52 |
| A.2 | Concatenation of Datasets | 53 |
| A.3 | Concatenation of Datasets | 53 |
| A.4 | Upload the image | 54 |
| A.5 | User Interface (UI) | 54 |
| A.6 | Final Mental-Health Metrics | 55 |

**CHAPTER 1**

# INTRODUCTION

## OVERVIEW

The rising prevalence of Major Depressive Disorder (MDD) demands a shift from subjective clinical evaluations to objective, data-driven diagnostic tools. Traditional techniques relying on self-reporting are often compromised by subjective bias and the stigma surrounding mental health. This research contributes to Artificial Intelligence and Machine Learning (AIML) by developing an integrated, multimodal system for early depression detection. The project’s goal is an objective, scalable, and privacy-preserving system suitable for professional settings, providing a reliable path to early intervention.

The project's hypothesis is that integrating data from various modalities overcomes the limitations of unimodal approaches for a more accurate assessment. The system integrates three data streams: physiological EEG signals, behavioral cues from facial expression analysis, and psychometric data from surveys. Its architecture uses a late-fusion framework, combining outputs from three independent "expert" models, each specialized for one modality. Data were sourced from public research datasets like MODMA for EEG and RAVDESS for facial analysis, ensuring a standardized and ethical foundation.

The implementation uses Python, TensorFlow, and OpenCV. A key technical achievement was the facial expression model, which attained **83.76% accuracy** using transfer learning with CNN. The system's strength is its ability to provide a holistic assessment by correlating objective physiological data with observable behaviors and subjective experiences. This approach mitigates the weaknesses of single data sources, such as a user consciously "social masking" their emotions, which the EEG and survey models can help correct, leading to a more robust diagnosis.

In conclusion, this project successfully demonstrates that a multimodal AIML system is a powerful proof-of-concept for early and accurate depression detection. The system confirms that by analyzing physiological, behavioral, and self-reported data together, a more complete picture of mental health is achieved. By advancing accessible diagnostics, this work contributes meaningfully to UN Sustainable Development Goals 3 (Good Health) and 8 (Decent Work). This research helps pave the way for a future where technology can build a healthier and more productive society.

## PROBLEM DEFINITION

The effectiveness of traditional depression assessment is fundamentally compromised, especially in workplace environments. Current diagnostics rely on subjective, self-reported symptoms, which are often distorted by cognitive biases and a desire to maintain a professional image. These assessments are also episodic, failing to capture the dynamic nature of depressive symptoms that fluctuate significantly over time. Compounding these issues are major barriers like the persistent stigma surrounding mental health, which discourages help- seeking, and severe constraints in accessing qualified mental health professionals. Employees often fear career repercussions or social judgment, leading to the concealment of their struggles and delayed intervention.

Technologically, current automated systems for depression detection exhibit critical gaps that limit their practical use. Most of these systems rely on single data sources, such as social media activity or wearable device metrics, making them unable to capture the full complexity of the condition. Many also lack the robust, real-time processing capabilities that are essential for timely intervention.

Therefore, the primary problem this research addresses is the lack of objective, scalable, and privacy-preserving depression detection systems suitable for workplace deployment. Current approaches suffer from a combination of subjective bias, limited accessibility, and technological inadequacy, which leads to delayed diagnoses and missed opportunities for intervention. This results in suboptimal mental health outcomes in professional settings. This project aims to develop a comprehensive solution that directly addresses these multifaceted challenges through the integration of multimodal data.

### key objectives:

 **Investigate and Preprocess Data:** To investigate and apply robust preprocessing techniques to the three different data sources: EEG signals, facial expression data, and psychometric scores.

 **Extract Meaningful Features:** To develop methods for extracting meaningful and noise-free features from each of the distinct data types.

 **Design Unimodal Models:** To implement and train separate, specialized machine learning models for each individual modality (EEG, facial, and survey).

 **Build a Facial Recognition CNN:** To specifically implement a Convolutional Neural Network (CNN) to handle the task of facial expression recognition

 **Implement a Fusion Strategy:** To design and develop a late-fusion architecture capable of intelligently integrating the predictive outputs from the three independent models.

 **Generate a Final Classification:** To use the fusion architecture to generate a single, more accurate final classification of the user's depressive state.

**CHAPTER 2**

# SYSTEM ANALYSIS

## EXSISTING SYSTEM

Existing depression detection relies on traditional clinical evaluations, which are subjective and episodic, and unimodal automated systems. These unimodal systems analyze single data sources like facial expressions or EEG signals and are often unreliable due to issues like social masking and noise sensitivity. Both approaches are fundamentally limited by their reliance on a single aspect of a complex condition, which compromises their overall effectiveness .

### Current Monitoring Techniques:

* + 1. **Traditional Clinical Evaluation:**

 Relies on structured interviews and standardized questionnaires like the Patient Health Questionnaire (PHQ-9).

 Limited by subjective interpretation from clinicians [cite\_start]and infrequent, episodic assessments.

### Facial Expression Analysis:

 Utilizes computer vision and machine learning to identify emotions from a person's facial expressions.

 Accuracy is often compromised by "social masking," where individuals consciously hide their true feelings.

### Physiological Signal Monitoring:

 Analyzes electroencephalogram (EEG) brainwave activity to find objective neurological patterns linked to depression.

 Highly sensitive to background noise from environmental factors and even minor muscle movements.

### Digital Psychometric Tools:

 Converts traditional screening questionnaires into accessible digital formats to make them easier to use.

 Fails to solve the inherent problem of subjectivity, as the results are still affected by the patient's personal biases.

## PROPOSED SYSTEM

The proposed system is a sophisticated multimodal framework using Artificial Intelligence and Machine Learning (AIML) to deliver a more objective and accurate assessment of depressive states. It integrates three interrelated data streams into a unified analysis: physiological EEG brain wave data, behavioral cues from facial expression analysis via a CNN, and psychometric data from standardized surveys. The system's architecture uses three independent models, each optimized for a single modality, with a late- fusion scheme to combine their predictions. This strategy synthesizes neurological, behavioral, and subjective data to overcome the weaknesses of single-source methods. The result is a powerful diagnostic tool that substantially improves accuracy and reliability in depression detection.

### Functionality and Components of the System:

1. **Data Processing and Preparation:**

 **Acquires and organizes three distinct types of data:** physiological (EEG), behavioral (facial expressions), and psychometric (surveys).

 Splits datasets based on subject or actor ID to prevent data leakage and ensure the model is evaluated fairly on unseen individuals.

 Preprocesses all visual data by detecting and cropping faces before resizing them to a uniform 48x48 pixel format for the model.

 Applies data augmentation techniques like random flipping and rotation to the training images, which helps the model learn to generalize better.

### Unimodal Model Training:

 Employs a transfer learning strategy using the CNN model to achieve high-accuracy facial expression recognition.

 Uses best practices during training, including the Adam optimizer and Early Stopping, to prevent overfitting and find the best model weights.

### Multimodal Fusion Core:

 Utilizes a late-fusion (decision-level) architecture to intelligently integrate the predictive outputs from the three separate "expert" models.

 Generates a single, final classification that is more robust and nuanced than the output from any individual unimodal model.

### Real-Time Inference Pipeline:

 Processes new input images or video frames using the exact same preprocessing steps as the training data to ensure consistency.

 The final trained model takes the prepared input and performs a forward pass to generate a probability vector showing confidence levels for each emotion.

### User Interface and Deployment:

 Features a web-based dashboard built with standard frontend technologies (HTML, CSS, JavaScript) that allows users to interact with the system.

 A lightweight Python Flask backend server hosts the saved machine learning models and provides API endpoints for the frontend to request predictions.



Figure: 2.1 Proposed Diagram

## DEVELOPMENT ENVIRONMENT

The development environment for the "ORG MIND" project was designed to facilitate the creation of a robust and accurate multimodal system for depression detection. It was structured to seamlessly integrate a local development workflow with powerful cloud-based computational resources for intensive model training. To preprocess diverse datasets, train complex deep learning models, and deploy a full-stack application, the system depends on a specific stack of tools, libraries, and platforms. Below is an outline of the key components used in the development.

### Hardware Elements:

**Local Development Machine:**

 A high-performance processor, specified as an Intel Core i7 or higher, is required for development tasks.

 The system requires at least 16GB of RAM to handle data processing and model development effectively.

 A minimum of 500GB of storage space is necessary to accommodate the project's datasets and software environments..

### Cloud Computing and GPU:

 All heavy computational tasks, such as the intensive training of deep learning models, are offloaded to the cloud.

 The project utilizes Google Colaboratory (Colab), which provides free access to powerful computational resources.

Model training is accelerated using high-performance NVIDIA Tesla Series



GPUs, specifically the T4 or P100 models, via the Colab platform.

### Data Input and Acquisition Devices:

 The behavioral data analysis module is designed to work with a standard, commonly available webcam for facial recognition.

 The project intentionally avoids the need for costly or specialized medical hardware for data collection.

 No physical EEG machine is required, as the system uses pre-recorded physiological signals from a public dataset.

### Elements of Software: Core Development Stack:

 The system operates on both Windows 11 for local development and a Linux environment via Jupyter Notebook.

 The Python (version 3.9 or newer) serves as the primary programming language for the entire

 Anaconda Navigator is used to manage all packages and dependencies, creating a stable and isolated project environment.

### AIML and Data Processing Libraries:

 TensorFlow and its high-level API, Keras, are used for building and training the deep learning models.

 The OpenCV library is essential for all computer vision tasks, including real-time video processing and image manipulation.

 Scikit-learn and Pandas are utilized for data handling, traditional machine learning tasks, and model performance evaluation.

### Application and Deployment Tools:

 Jupyter Notebooks are used as the interactive development environment (IDE) for rapid prototyping and testing of the model.

### Model Architecture and Training:

 Utilizes deep learning architectures like Convolutional Neural Networks (CNNs) and a combined CNN+LSTM model for video analysis.

 Incorporates optimization techniques such as the Adam optimizer and Early Stopping to prevent overfitting and improve performance

### Web application or Dashboard:

 The user interface (UI) is designed as a clear, simple, and interactive web-based dashboard that visually represents the system's multimodal workflow.

 The backend server was built using Python and the lightweight Flask web framework. It serves as the core of the application, hosting the pre- trained and saved TensorFlow/Keras models.

### Analysis & Storage of Data: Data Analysis:

 The system's analysis is performed through a late-fusion architecture that intelligently combines the predictive outputs from three independent AIML models—one each for EEG, facial, and survey data—to generate a single, comprehensive final diagnosis.

### Data Storage:

 The primary stored assets are the pre-trained and optimized AIML models, which are saved as files (e.g., .keras and .pkl) and loaded by the backend server to make real-time predictions.

### Data Source:

**Physiological and Psychometric Data:**

 The project utilizes the MODMA dataset, obtained directly from the author, for physiological EEG analysis, and the DASS-21 survey dataset, sourced from Kaggle, for psychometric assessment.

### Behavioral (Facial Recognition) Data:

 For the behavioral component, the facial recognition models were trained on datasets sourced from Kaggle, including the RAVDESS dataset for video-based analysis and a static image dataset for initial training.

### Communication and Connectivity: Frontend-to-Backend Communication:

The web-based dashboard acts as the client, sending analysis requests to the backend server's API endpoints using JavaScript's fetch() API.



### Backend API and Data Exchange:

 A Python Flask server hosts the trained AI models and returns prediction results to the frontend dashboard in a JSON format.

### Development Tools Overview:

**Core Development Stack:** The core stack includes Python for core programming, Anaconda for managing packages, and Jupyter Notebooks and googel colab are used as the interactive development environment.



**AI and Data Processing Libraries:** Key libraries like TensorFlow and Keras were used for deep learning, OpenCV for computer vision, and Scikit-learn for performance evaluation and TensorFlow for deeplearning.



 **Application and Deployment:** A Flask backend server hosts the models, a web-based dashboard serves as the user interface, and Google Colab provides GPU resources for model training.

**CHAPTER 3**

**SYSTEM DESIGN**

* 1. **UML DIAGRAMS**
     1. **Sequence Case Diagram**

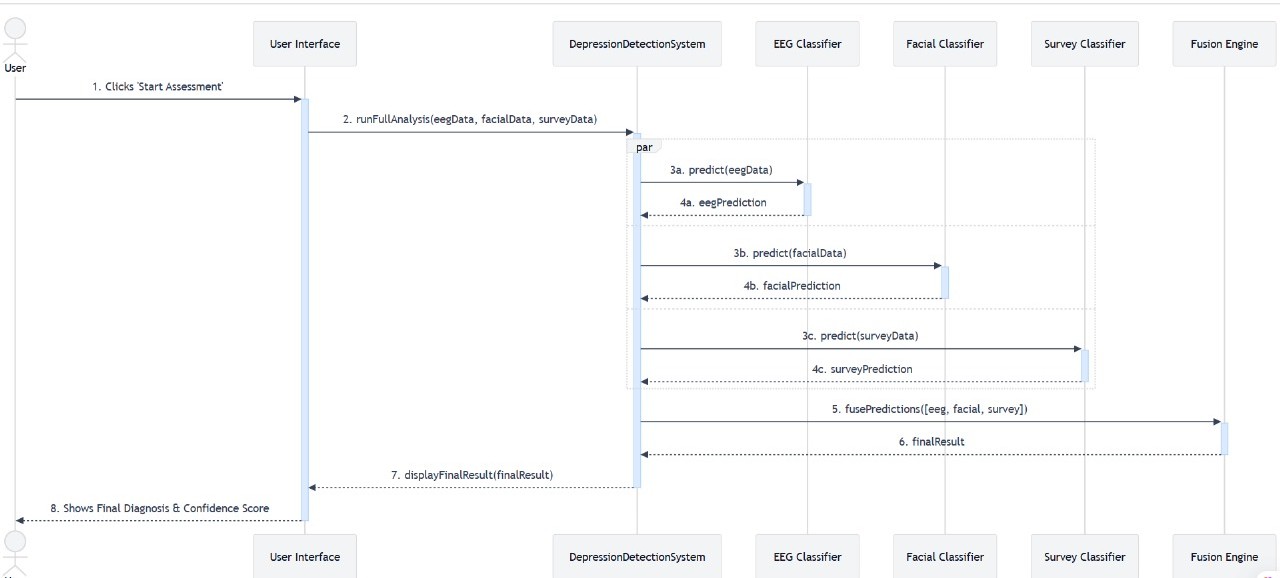
****

Figure: 3.1 Sequence Case Diagram

This sequence diagram illustrates the flow of interactions within the multimodal depression detection system. The process begins when the user clicks 'Start Assessment' on the User Interface, which triggers a request to the main Depression Detection System. The system then processes the three data types in parallel, simultaneously sending prediction requests to the EEG, Facial, and Survey classifiers. Each classifier independently processes its data and returns an individual prediction back to the main system. Once all three predictions are received, they are sent to the Fusion Engine, which combines them into a single, final result. This final result is then returned to the User Interface, which displays the final diagnosis and confidence score to the user, concluding the sequence.

## CLASS DIAGRAM

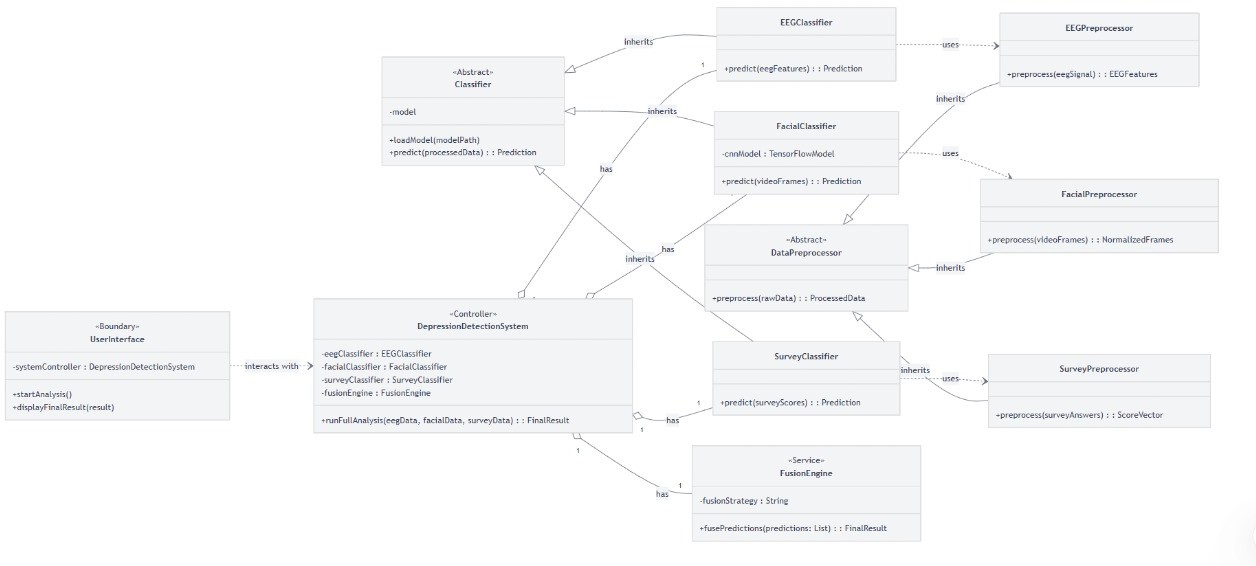
****

Figure: 3.2 Class Diagram

This Class Diagram illustrates the static architecture of the depression detection system, detailing its core components and their relationships. The central Depression Detection System acts as a controller, managing the workflow and interacting with the User Interface. This controller is composed of three specialized classifiers—the EEG Classifier, Facial Classifier, and Survey Classifier—each of which inherits properties from an abstract parent classifier. To prepare data for analysis, each classifier uses a dedicated preprocessor, such as the Facial Preprocessor. Finally, the controller utilizes a Fusion Engine service to combine the predictions from the individual classifiers, producing a single, unified result.

## DATA DICTIONARY

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Example** |
| **UserID** | **String (UUID)** | **A unique identifier for each user or subject participating in the assessment.** | **USR-001** |
| **Assessment Timestamp** | **DateTime** | **The date and time when the assessment was conducted.** | **2025-10-12**  **11:00:00** |
| **EEG Features** | **Array (Float)** | **An array of numerical features extracted from the user's EEG signals.** | **[0.45, -0.12,**  **1.23, ...]** |
| **Facial Emotion** | **Enum** | **The dominant emotion detected from the facial expression analysis.** | **('Sad', 'Happy', 'Neutral', 'Angry')** |
| **Survey Score** | **Integer** | **The total numerical score calculated from the user's responses to a standardized survey (e.g., DASS-21).** | **18** |
| **Depression Severity** | **Enum** | **A categorical label for the severity of depression based on the survey score.** | **('Normal',**  **'Mild', 'Moderate', 'Severe')** |
| **EEG Prediction** | **Decimal (4,3)** | **The depression probability score generated by the EEG classifier.** | **0.750** |
| **Facial Prediction** | **Decimal (4,3)** | **The depression probability score generated by the facial recognition classifier.** | **0.850** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Example** |
| **Survey Prediction** | **Decimal (4,3)** | **The depression probability score generated by the survey data classifier.** | **0.600** |
| **Final Diagnosis** | **Enum** | **The final, fused classification of the user's mental state after combining all modalities.** | **('Depressed', 'Not Depressed')** |
| **Confidence Score** | **Decimal (5,2)** | **The final confidence percentage of the system in its diagnosis.** | **83.76** |

Table: 3.1 Data Dictionary

## ACTIVITY DIAGRAM:

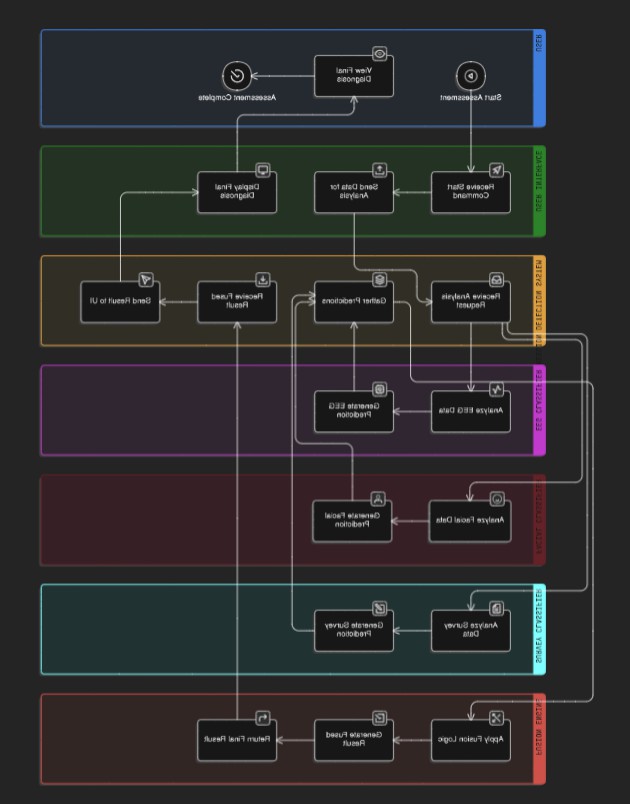
****

Figure: 3.3 Activity Diagram

This activity diagram illustrates the dynamic workflow of the multimodal depression assessment system. The process begins when the user initiates an assessment via the user interface. The main system then forks the process, triggering three parallel activities where the EEG, Facial, and Survey classifiers independently analyze their respective data. Once all three classifiers generate their individual predictions, the parallel flows are synchronized in a join step. These predictions are then passed to the Fusion Engine, which applies its logic to combine them into a single, high-accuracy final result. Finally, this fused diagnosis is sent back to the user interface and displayed to the user.

## DATA FLOW DIAGRAM

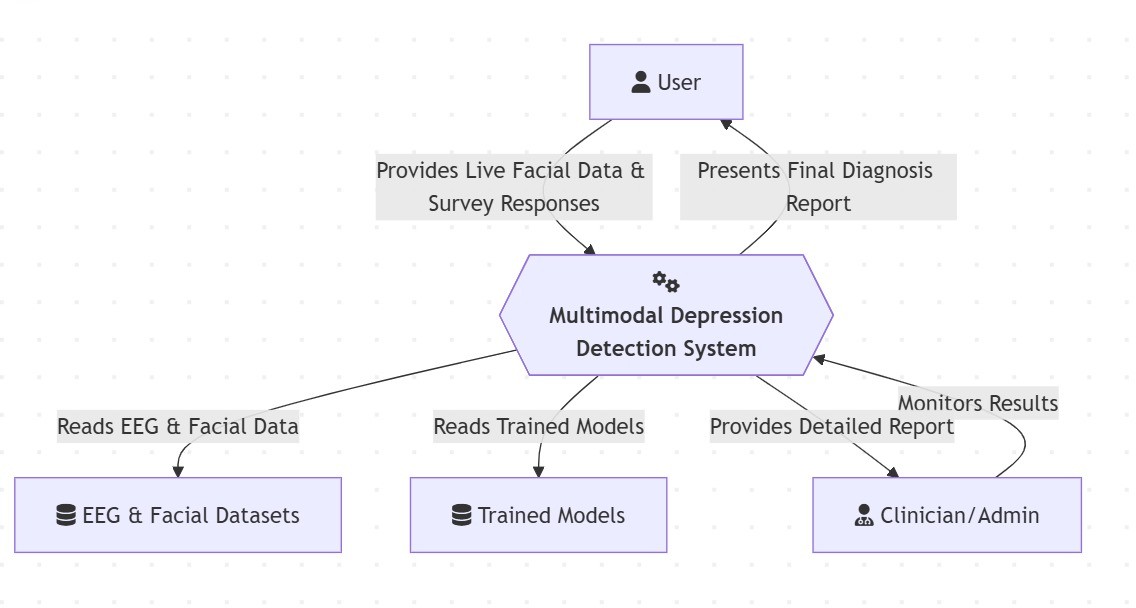


Figure: 3.4 Data flow Diagram

This diagram shows how the Smart Plant Health Monitoring System works, utilizing an array of sensors and machine learning to achieve optimized agriculture. Soil moisture, soil and air temperature, humidity, and light intensity sensors gather data to be transmitted to a Composite Sensor Module for initial processing. The aggregated data is processed by an ML Prediction Engine to detect health issues with the plants and suggest solutions. Results and notifications are made available to the farmer through a user interface, while actuators like irrigation and fans condition automatically. The combined system allows farmers to track crops in real time, increase yield, and provide sustainable farming.

**CHAPTER 4**

# SYSTEM ARCHITECTURE

## ARCHITECTURE OVERVIEW

### System Architecture Overview for Depression Detection System:

This pipeline handles the real-time processing of data from an end-user. It captures live data, such as facial expressions from a webcam and survey responses, and processes it using the same normalization and feature extraction steps as the training data. This prepared data is then fed into the pre-trained multimodal model to generate a quick and accurate prediction. The final diagnosis and confidence score are then displayed to the user through an interactive interface. The system is continuously evaluated to ensure it works well and can be improved in the future..

### Hardware Layer

**Core Computing Components:**

 The system requires a processor equivalent to an Intel Core i7 or higher for development and data processing tasks.

 A minimum of 16GB of RAM is necessary to effectively handle the project's data and model development processes.

 At least 500GB of storage space is needed to house the various datasets and the software development environment.

### Data Acquisition and Processing Hardware:

 The behavioral analysis module is designed to capture facial data using a standard, commonly available webcam.

 Intensive deep learning model training is performed using high- performance NVIDIA Tesla Series GPUs (T4 or P100) accessed through the Google Colab platform.

 The project avoids the need for costly medical devices by utilizing a pre- recorded EEG dataset, thereby eliminating the requirement for a physical EEG machine.

### Communication Layer Client-Server Architecture:

 The system is built on a client-server architecture, which separates the user-facing dashboard from the backend AI models.

 The frontend is a web-based dashboard that functions as the client, allowing users to initiate analysis and view results.

 The backend is a Python server built with the Flask framework, which hosts the trained models and handles all computational tasks.

### API and Data Exchange Protocol:

 Communication between the frontend and backend is managed through a set of API endpoints, such as /predict\_facial.

 The frontend uses JavaScript's fetch() API to send HTTP requests to the server, triggering the prediction process.

 The server returns all prediction results and data to the frontend client in the JSON (JavaScript Object Notation) format.

### Software Layer

**Core Development Stack:**

 The system's software operates on both Windows 11 (64-bit) for local development and a Linux environment via Google Colab.

 Python 3.9 or newer is the main language used in this project because it is easy to use and has lots of tools that help with AI and machine learning.

 Anaconda helps manage packages and dependencies, creating separate environments to keep the project stable and organized.

### AIML and Data Processing Libraries

 TensorFlow and its high-level API called Keras are used to make it easier to build and train complex models.

 OpenCV was very important for all the computer vision tasks, like reading videos, taking individual frames, and doing image processing in real time.

 Scikit-learn was used to split data and calculate important performance measures like accuracy, precision, and recall, while Pandas helps with data handling.

### Application and Deployment Tools:

 Jupyter Notebooks served as the interactive development environment for the project, used on Google Colab for its effective workflow.

 A lightweight Flask web framework was used to build the backend server, acting as the main part of the application that hosts the trained models.

 The Frontend User Interface (UI) was created as a web-based dashboard using standard web technologies: HTML, CSS, and JavaScript for all interactive features.

### System Interaction Flow

**User Initiation:** The entire workflow begins when the User interacts with the User Interface to start an assessment. This single action triggers the main analysis pipeline.



 **Parallel Data Processing:** The **User Interface** sends a request to the main Depression Detection System, which acts as the system's controller. The controller then initiates three processes that run in parallel pipeline:

* + The **EEG Classifier** analyzes the physiological brainwave data.
  + The **Facial Classifier** analyzes the behavioral data from the user's facial expressions.
  + The **Survey Classifier** analyzes the psychometric data from the user's survey responses.

**Independent Prediction and Fusion:** Each of the three classifiers independently generates a prediction and returns it to the main Depression Detection System. The system then gathers these individual predictions and sends them to the Fusion Engine, which applies a late-fusion strategy to intelligently combine them into a single, high-accuracy final result.



 **Final Diagnosis and Output :** The final, fused result is sent back from the Fusion Engine to the main system controller. The controller then forwards this result to the User Interface, which displays the final diagnosis (e.g., "Depressed" or "Not Depressed") along with a numerical confidence score directly to the user, completing the interaction flow.

### System overview diagram:

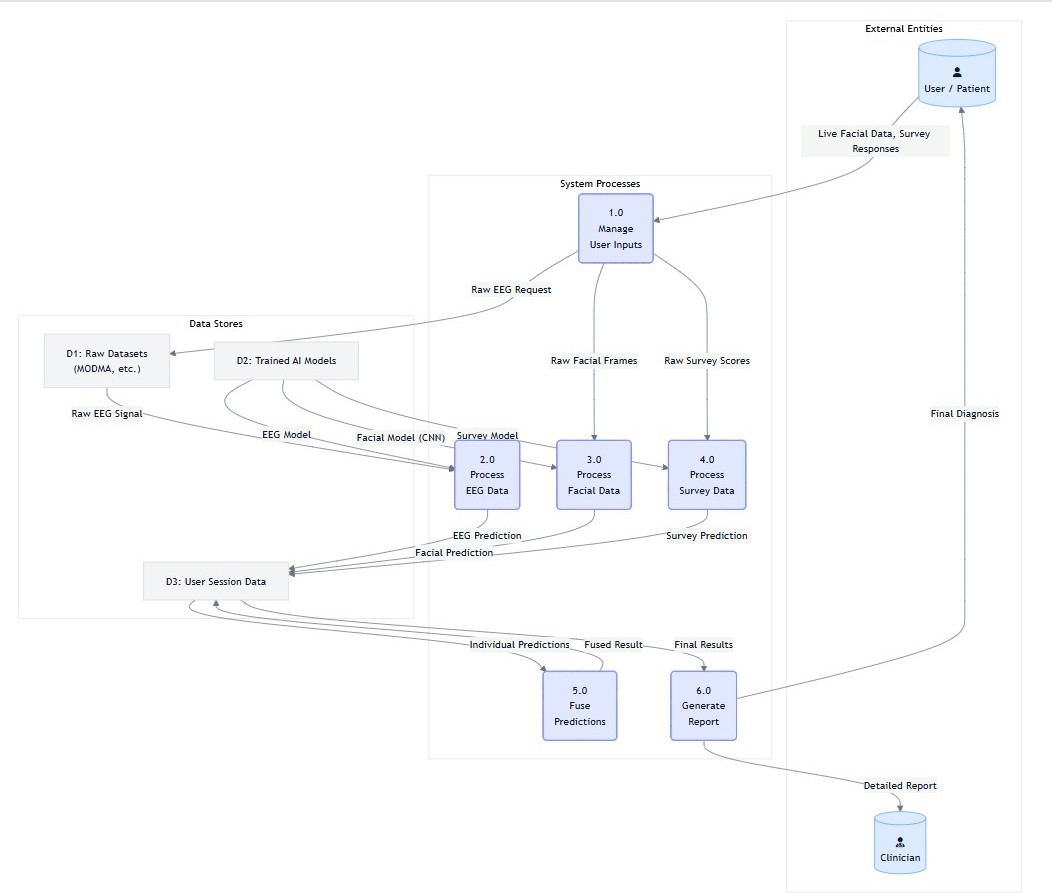
****

Figure: 4.1 System Overview

### Conclusion

The architecture of the Smart Plant Health Monitoring System is designed to provide an efficient and automated solution for plant care. By integrating various sensors with a microcontroller, the system can monitor and control environmental factors such as soil moisture, temperature, humidity, and light. This integrated solution ensures that plants receive the optimal conditions for growth, minimizing manual intervention and improving productivity. Real-time monitoring, cloud integration, and automated alerts are key features that enhance the system’s functionality this is the final conclusion for the project idea.

## MODULE DESCRIPTION

### Module Description for Smart Plant Health Monitoring System:

The system is designed with a modular software architecture, which breaks down the complex overall task into smaller, self-contained, and manageable components. This approach simplifies development, testing, and future maintenance of the application. The key modules include Dataset Exploration and Preparation, Image Augmentation, Model Training and Optimization, Input Image Processing and Classification, and finally, the User Interface and Deployment module, which creates the interactive client-server application.

### Dataset Exploration and Preparation:

This foundational module focused on the meticulous process of finding, examining, and structuring the diverse datasets required for a comprehensive multimodal analysis. The primary objective was to build a clean, well- organized, and trustworthy database to serve as the foundation for training and validating the machine learning models. Three distinct types of data were prepared: physiological data, for which the public MODMA research dataset was used, providing standardized and ethically sourced pre-recorded EEG signals from both depressed and non-depressed individuals ;behavioral data, for which the RAVDESS video dataset was selected, allowing for the automated extraction of accurate labels like emotion and actor ID directly from the structured filenames ;and psychometric data from standardized surveys. A critical part of this module was the strategic division of the data into training, validation, and test sets. To prevent data leakage and ensure a fair and realistic evaluation of the model's performance on new individuals, the datasets were split based on subject or actor ID, guaranteeing that the model is tested on people it has never seen during its training phase this ensures that the model performance is good and it will not overfit or become more complex.

### Image Augmentation and Preparation:

This module was dedicated to preparing the image and video data for training the facial recognition system, with the dual goals of transforming raw images into a clean, standardized format and artificially expanding the training dataset to improve the model's ability to generalize. The visual data was processed through a systematic pipeline, beginning with face detection in each image or video frame using OpenCV. Once a face was located, it was cropped to isolate the facial region, and all cropped images were resized to a uniform 48x48 pixels. The color format was converted to match the VGG16 model's input requirements, and pixel values were normalized to a range of 0-1 to help the model learn more effectively. To make the model more robust and less prone to overfitting, data augmentation was applied during training. This involved introducing random transformations to the training images, such as horizontal flipping, slight rotations, and minor zooming, which provided the model with a much larger and more varied set of examples to learn from.

### Unimodal Model Training and Optimization

As the core of the project's AIML implementation, this module involved the design, training, and optimization of the specialized deep learning models for each data modality. For the facial expression recognition component, two main architectures were tested. While a custom Convolutional Neural Network (CNN) was initially built from scratch, superior results were achieved using a **Transfer Learning** approach with the VGG16 model, which was pre-trained on the vast ImageNet dataset. By keeping the base layers of VGG16 frozen and adding a new custom classifier on top, the model was able to leverage the powerful, pre-learned features from VGG16, which was instrumental in achieving a high accuracy of **83.76%** even with limited data. Best practices were followed during training, with models being configured using the Adam

optimizer and categorical\_crossentropy loss function. To combat overfitting, two crucial techniques were employed: **Early Stopping**, which monitors validation loss and stops training to restore the best weights, and fine-tuning, which involved unfreezing the top layers of VGG16 and retraining them with a very low learning rate to better adapt the model to the specific facial expression data.

### Communication Module :

This module outlines the real-time inference process, where the final trained model is used to predict the emotion from a new, unseen input image. To ensure consistency and accuracy, this workflow strictly follows the same preprocessing steps that were used during the model training phase. When a new image is provided to the system, whether from a file or a live webcam feed, it is first passed through the established preprocessing pipeline. The system detects and crops the face, resizes the image to 48x48 pixels, converts it to the correct color format, and normalizes the pixel values to match the training data's distribution. Once the image is prepared, it is converted into a tensor and its dimensions are expanded to include a batch dimension, which is the format the model expects for processing. The model then performs a forward pass and produces an output in the form of a probability vector, which indicates the model's confidence for each potential emotion class. Finally, the system identifies the highest score in this vector and maps its position back to the corresponding emotion label, such as 'Sad' or 'Happy', to yield the final prediction.

### Multimodal Fusion Core:

This module represents the intelligent core of the entire system and is the component that directly fulfills the project's central hypothesis: that integrating data from various modalities can overcome the limitations of unimodal approaches to achieve a more accurate and comprehensive assessment. The architecture's final classification is achieved through a **late-fusion (or decision- level) scheme**, a strategy deliberately chosen for its proven flexibility, adaptability, and robustness in combining heterogeneous data sources. This module is designed to intelligently integrate the predictive outputs from the three independent "expert" models only after each has completed its specialized analysis. The true strength of this approach lies in its ability to provide a holistic assessment by mitigating the weaknesses of any single modality.

A powerful example of its function is in a scenario where a user consciously forces a smile for the camera, a phenomenon known as "social masking". In this case, the Facial Recognition Model might be misled into an incorrect prediction like 'Happy' or 'Neutral'. However, the fusion engine would simultaneously receive conflicting inputs from the other models The EEG model, analyzing the user’s underlying neural state, could detect physiological patterns consistent with low mood, while the Survey Model, based on the user's honest self-reporting, would also indicate a high risk of depression. By receiving these contradictory predictions, the late-fusion engine can use a weighted average to correctly override the misleading behavioral data from the facial model and arrive at the correct final diagnosis of 'Depressed'. This ability to correlate objective physiological data (EEG), observable behavioral cues (facial expressions), and subjective self-reported experiences (surveys) makes the system incredibly robust and results in a powerful, multifaceted diagnostic tool that substantially improves accuracy and interpretive nuance.

### User Interface and Deployment:

The This final module details the development of the client-server architecture that makes the powerful backend models accessible and interactive for end-users. The **Frontend User Interface (UI)** was created as a responsive, web-based dashboard using standard technologies: HTML for the structure, CSS with the Tailwind CSS framework for a modern aesthetic, and JavaScript for handling all interactive features. The UI is designed to be intuitive, clearly reflecting the multimodal workflow with separate sections for the EEG, facial recognition, and survey inputs, and providing real-time feedback to the user during analysis. The **Backend Server** was built using Python and the lightweight Flask web framework. It serves as the application's core, hosting the trained TensorFlow/Keras models (saved as .keras files) and providing a set of API endpoints that the frontend can use to send requests and receive predictions. This client-server setup effectively separates the user interface from the heavy computational work, making the entire system more scalable and easier to maintain in the long term.

### Conclusion

This project successfully developed a multimodal AIML system that provides a more objective and reliable method for early depression detection than traditional approaches. By integrating physiological EEG signals, behavioral facial expressions, and psychometric survey data, the system overcomes the limitations of any single data source. A key technical success was the facial recognition model, which achieved 83.76% accuracy using transfer learning. The final system serves as a powerful proof-of-concept, demonstrating that a multimodal approach offers a more accurate assessment of mental health.

## PSEUDO CODE/ ALGORITHM DESCRIPTION

### Pseudo Code Description:

From a technical standpoint, the system's pseudo-code outlines a clear sequence of function calls and data transformations within a client-server architecture. The main function, Start\_Assessment(), is invoked when the Flask backend receives an API request from the user interface. The first step within this function is an initialization routine, Load\_Models(), which loads the serialized model files (e.g., facial\_model.keras, eeg\_model.pkl) into their respective objects. Following initialization, the system executes a parallel block containing three distinct function calls: Process\_EEG(), Process\_Facial(), and Process\_Survey(). Each of these functions is responsible for its modality's complete pipeline; for example, Process\_Facial() includes sub-routines for capturing a webcam frame, calling OpenCV functions for face detection and preprocessing (cropping, resizing, and normalizing the image), and finally passing the prepared tensor to the facial\_model.predict() method. The main function then awaits the return values from these three parallel threads, which are the probability vectors for each classifier. Once all three vectors are collected, they are passed as arguments to the Fusion\_Engine() function. This function implements the late-fusion logic, such as calculating a weighted average of the probabilities, and returns a final classification string (e.g., "Depressed") and a confidence score. The Start\_Assessment() function concludes by packaging this final result into a JSON object and returning it to the user interface for display.

### Algorithm Description:

The core algorithm of the "ORG MIND" system is a sophisticated, event- driven workflow designed to achieve high-accuracy depression detection through multimodal data synthesis. The algorithm's logic is founded on the principle of parallel processing followed by a decision-level fusion to ensure both efficiency and robustness. The process initiates in a resting state, where the three pre-trained "expert" models for EEG, facial, and survey analysis are loaded into memory. When a user triggers an assessment, the main controller orchestrates a concurrent analysis of the three heterogeneous data streams. This parallel design is crucial, as it allows the system to process physiological, behavioral, and psychometric data simultaneously, significantly reducing the overall assessment time. Each modality's dedicated classifier processes its input to generate an independent probability score for depression. The algorithm then enters a synchronization phase, pausing execution until all three classifiers have returned their predictions. This step is critical as it ensures the Fusion Engine has a complete set of data before making its final judgment. The fusion algorithm itself, which is the intellectual core of the system, intelligently weighs and combines these disparate scores, effectively mitigating the weaknesses of any single modality—such as social masking in facial data or subjective bias in surveys—to produce a final, holistic, and more reliable diagnosis.

### Pseudocode for Depression Detection using machine learning:

// Main procedure that orchestrates the entire assessment process PROCEDURE Run\_Multimodal\_Assessment()

BEGIN

// --- 1. INITIALIZATION ---

// This happens once when the backend server starts. eeg\_model = Load\_Model("eeg\_classifier.pkl") facial\_model = Load\_Model("facial\_cnn\_model.keras") survey\_model = Load\_Model("survey\_classifier.pkl")

// --- 2. WAIT FOR USER INPUT ---

// The system waits for an API call from the user interface. WAIT for user\_request to start assessment.

// --- 3. PARALLEL DATA PROCESSING & PREDICTION ---

// These three blocks are executed concurrently to save time. BEGIN PARALLEL

// -- EEG Branch --

eeg\_features = Get\_EEG\_Features\_From\_MODMA\_Dataset() eeg\_prediction = eeg\_model.predict(eeg\_features)

// -- Facial Branch --

webcam\_frame = Capture\_Webcam\_Frame() processed\_face = Preprocess\_Facial\_Image(webcam\_frame) facial\_prediction = facial\_model.predict(processed\_face)

// -- Survey Branch --

survey\_score = Get\_Survey\_Score\_From\_UI() survey\_prediction = survey\_model.predict(survey\_score)

END PARALLEL

// --- 4. FUSION OF PREDICTIONS ---

// The main thread waits until all parallel tasks are complete.

// It then combines the individual predictions using the fusion engine. final\_diagnosis, confidence = Fusion\_Engine(

eeg\_prediction, facial\_prediction, survey\_prediction

)

// --- 5. OUTPUT ---

// The final result is sent back to the user interface for display. Display\_Result\_On\_UI(final\_diagnosis, confidence)

END

### Detailed Explanation of Pseudocode

1. **Main System Workflow**

The main procedure, Run\_Multimodal\_Assessment, outlines the high-level logic that orchestrates the entire system from server start-up to the final diagnosis. The process begins with an initialization step where the three pre- trained machine learning models for EEG, facial, and survey analysis are loaded into memory. This is an efficiency measure performed once when the server starts. The system then enters a waiting state until it receives a request from the user interface to begin an assessment. Upon receiving this trigger, the core of the workflow begins with a parallel processing block. This is a critical design choice that allows the system to analyze the physiological EEG data, the behavioral webcam data, and the psychometric survey data concurrently, which significantly improves the speed of the assessment. After each of these parallel branches generates its independent prediction, the main thread waits for all three tasks to complete. These individual predictions are then passed to the Fusion\_Engine for the final synthesis step. The procedure concludes by taking the unified diagnosis and confidence score from the fusion engine and sending it back to the user interface to be displayed.

### Facial Image Preprocessing

The helper function Preprocess\_Facial**\_**Image is a critical component that ensures the consistency and accuracy of the facial recognition module. Its purpose is to take a raw image frame, such as one captured from a webcam, and transform it into the precise format that the Convolutional Neural Network (CNN) model expects as input. The pipeline begins by detecting the region of the face within the image, isolating it from any irrelevant background information. The image is then cropped to this specific facial region. Next, the cropped face is resized to a uniform dimension of 48x48 pixels, which matches the input size of the trained model. Finally, the pixel values of the resized image are normalized, typically scaled to a range between 0 and 1, a step that is essential for helping the deep learning model learn effectively. The function returns this fully processed image, ready for classification

### Late Fusion Engine:

The Fusion\_Engine function represents the intellectual core of the system, where the individual insights from the three "expert" models are synthesized into a single, more reliable diagnosis. The pseudo-code describes a late-fusion (or decision-level) strategy, which combines the final probability scores from each classifier rather than raw data. The algorithm shown is a straightforward yet effective method of averaging the probabilities. It takes the depression probability scores from the EEG, facial, and survey models and calculates the mean probability for the "Depressed" class. This averaged score is then used to make the final determination. A simple threshold, such as 0.5, is applied to this average; if the score exceeds the threshold, the diagnosis is "Depressed," otherwise, it is "Not Depressed." This fusion approach is particularly powerful because it can balance out and correct for potential errors from any single modality, such as a misleading facial expression due to social masking, by incorporating evidence from the other data streams.The final confidence score is also derived directly from this averaged probability, providing the user with a transparent measure of the system's certainty in its conclusion.

**CHAPTER 5**

# SYSTEM IMPLEMENTATION

## 5.1 MODULE-WISE BREAKDOWN

### Modules :

 Dataset Exploration and Preparation

 Image Augmentation and Preparation

 Model Training and Optimization

 Input Image Processing and Classification

 User Interface and Deployment

### Data Exploration And Preparation:

This module focused on finding, examining, and carefully organizing different types of data needed for a multimodal analysis. The main aim was to build a clean, structured, and trustworthy data base for training and testing our machine learning models. Three kinds of data were prepared: physiological (EEG), behavioral (facial expressions), and psychometric (surveys). For the physiological data, the MODMA dataset was used.

This is a public research dataset that includes pre-recorded EEG signals from people who are depressed and those who are not. Using this dataset ensured we had a standard and ethical data source, avoiding the need for complicated human subject studies. For the behavioral data, the RAVDESS video dataset was selected. A major part of this phase involved automatically extracting accurate labels like emotion, emotional intensity, and actor ID from the structured filenames in RAVDESS (e.g., 01-01-02-01-02-02-01.mp4).

similar structure to FER-2013 was used, where images were already sorted into folders representing seven basic emotions.

An important part of this module was dividing the data into training, validation, and test sets.

To stop data leakage and make sure the model was evaluated fairly, the video and image datasets were split based on subject or actor ID. This ensures the model is tested on people it hasn’t seen during training, giving a real measure of how well it can work with new subjects.

### Image Augumentation and Preparation:

This part of the work was all about getting ready the pictures and video clips to train our facial recognition system.

The aim was to change the raw images into a clean, standard format and make the training data bigger so the model doesn’t just remember the images it sees but actually learns to recognize faces properly.

We used a step-by-step process to handle all the visual data.

It started with finding the face in each image or video frame using OpenCV. Once the face was found, we cut it out so the model only looked at the face itself. Then, we made sure all the faces were the same size, like 48x48 pixels, and changed the color format to match what the VGG16 model uses, which is RGB. After that, we adjusted the brightness of the images by changing the pixel values from a range of 0-255 to a range of 0-1. This step is important because it helps the model learn more effectively.To make the model better at recognizing faces in different conditions, we used data augmentation.

This means we applied random changes to the training images while the model was learning. These changes included flipping the images left to right, rotating them slightly, and zooming in a little. By doing this, the model got a lot more examples to learn from, which helped it understand faces even when they were turned, tilted, or a bit bigger or smaller.

### Model Training and Optimization:

This module is the heart of the project's AIML implementation, where we designed, trained, and optimized the deep learning models.

The main goal was to create high-performing "expert" models for each type of data.For facial expression recognition, we tested two main model structures.

At first, we built a custom Convolutional Neural Network (CNN) from scratch. But to get better results, we used Transfer Learning with the VGG16 model, which was trained on the ImageNet dataset. We kept the base layers of VGG16 fixed and added a new classifier on top, which we trained using our own facial expression data. This method uses the already learned features from VGG16, helping the model achieve high accuracy (83.76%) even when we had limited data. For video data, we created a more complex model that combines CNN and LSTM. This helps capture details from each frame as well as how expressions change over time.

We followed best practices during training.

The models were set up using the Adam optimizer and categorical\_crossentropy as the loss function. To deal with overfitting, which happened in early tests, we used two important techniques. First, we used Early Stopping, which monitors the validation loss and stops training when it stops improving, then restores the best weights from earlier. Second, we fine-tuned the VGG16 model by

unfreezing the top layers and retraining them with a very low learning rate, making the model better suited to our specific data.

### Input Image Processing and Classification:

This module explains the real-time inference process, which uses the final trained model to predict the emotion of a new input image. This workflow follows the same steps used during training to keep everything consistent.

When a new image is given to the system, like one from a file or a frame from a webcam, it goes through the same steps as before. First, the system detects and cuts out the face, then resizes the image to 48x48 pixels. It converts the image to the correct color format and adjusts the pixel values so they match the training data. Once the image is ready, it is turned into a tensor and sent to the model. Before sending, the tensor is modified to include a batch dimension, which helps the model process it properly. The model then runs a forward pass and gives an output in the form of a probability vector. This vector shows the model's confidence levels for each possible emotion. After getting the output, the system uses a function to find the highest score in the vector. This score's position is then matched to its corresponding emotion name, like 'Sad' or 'Happy', which becomes the final prediction.

### User Interface and Deployment :

This module explains how the client-server architecture was developed to make the strong backend models easy to use and interact with for end- users.The Frontend User Interface (UI) was created as a web-based dashboard using standard web technologies: HTML for building the structure, CSS with

the Tailwind CSS framework for a modern and responsive look, and JavaScript for handling all the interactive features. The UI is designed to be easy to use, showing the multimodal workflow clearly with separate sections for EEG, facial recognition, and survey inputs. It responds to user actions like button clicks, gives real-time feedback during analysis, and displays both individual and final combined results in a clear, visual way. The Backend Server was built using Python and the lightweight Flask web framework.

It acts as the main part of the application, hosting the trained and saved TensorFlow/Keras models (.keras files). It provides a set of API endpoints (like

/predict\_facial) that the frontend can use to send requests. When the UI needs a prediction, it uses JavaScript's fetch() API to send a request to the relevant endpoint. The Flask server gets this request, does the required preprocessing and runs the model to get the prediction, then returns the result as a JSON object. This client-server setup separates the user interface from the heavy computational work, making the system scalable and easier to maintain.

## CHAPTER 6 SYSTEM TESTING

**6.1 TESTING AND REPORT**

### Test Cases for Depression Detection using Machine Learning System:

Testing is a critical phase to ensure the system functions as expected under various conditions.

### Test Case: Facial Recognition Model Accuracy

**Test Objective:** To verify that the facial recognition model correctly identifies a clear emotional state and assigns an appropriate depression probability score.

Table: 6.1 Test case 1

|  |  |  |
| --- | --- | --- |
| **Step** | **Test Steps** | **Expected Results** |
| 1 | Provide an input image of a person with a distinct "Sad" facial expression to the facial analysis  module. | The system should correctly  classify the dominant emotion  as "Sad". |
| 2 | Check whether the irrigation system is activated when the moisture level falls  below the threshold (e.g., 30%). | The depression probability score  generated by the facial model  should be high (e.g., > 0.75),  reflecting the negative emotion. |

### Test Case: Multimodal Fusion with Conflicting Data ("Social Masking")

**Test Objective:** To verify the system's core hypothesis that the late-fusion engine can correctly diagnose depression even when one modality provides conflicting data.

Table: 6.2 Test case 2

|  |  |  |
| --- | --- | --- |
| **Step** | **Test Steps** | **Expected Results** |
| 1 | Trigger the EEG and Survey analyses using sample data known to indicate a high risk of depression. | The EEG and Survey modules should both return a high depression probability score. |
| 2 | Trigger the Facial Recognition analysis  by providing an input image of a person  with a "Happy" or smiling expression. | The facial module should  return a low depression  probability score. |
| 3 | Run the final "Multimodal Diagnosis"  after all three individual predictions are  complete. | The system's final diagnosis should be **"Depressed"** with a high confidence score, successfully overriding the misleading low-risk facial  data. |

### Test Case: End-to-End System Workflow

**Test Objective:** To verify the complete and seamless flow of interaction from the user interface through the backend processing and back to the UI for final display.

Table : 6.3 Test case 3

|  |  |  |
| --- | --- | --- |
| **Step** | **Test Steps** | **Expected Results** |
| 1 | From the user interface, click the "Analyze" button for each of the three modalities (EEG, Facial, and Survey)  in sequence. | The UI should successfully  display the individual prediction  result for each of the three  modalities without any errors. |
| 2 | After all three predictions are visible,  click the "Run Multimodal Diagnosis"  button. | The system should display a  final, fused diagnosis and a  consolidated confidence score  on the UI. |

**Future Improvements**

|  |  |
| --- | --- |
| **Future Improvement** | **Description** |
| Real-Time EEG Integration | The current system uses a pre-recorded EEG dataset. A future enhancement would be to integrate a real-time EEG acquisition module using portable, consumer- grade headsets to capture and analyze a user's brainwave activity live. This would transform the system into a dynamic, real-time diagnostic aid. |
| Speech and Language Analysis | To create a more holistic system, a fourth modality— speech analysis—could be incorporated. A new module could extract vocal features like pitch and tone, while Natural Language Processing (NLP) could perform sentiment analysis on transcribed text to provide another layer of behavioral insight. |
| Longitudinal Monitoring | The current system provides a "snapshot" diagnosis from a single session .A powerful extension would be to develop a system capable of longitudinal monitoring, tracking a user's mental state over weeks or months. This would enable predictive analysis, potentially forecasting a depressive episode before it becomes severe. |
| Clinical Validation | A critical next step is to test and re-train the system on a larger and more demographically diverse clinical dataset, collected in partnership with mental health institutions. This is essential for assessing the model's performance and fairness across different groups, which is a crucial step toward real-world deployment. |
| Explainable AI (XAI) | To increase the system's utility and trustworthiness for clinicians, future versions could incorporate Explainable AI (XAI) techniques. An XAI-enhanced system would offer transparent insights into its decision-making process, highlighting which specific features most influenced a "Depressed" classification. |

**CHAPTER 7**

# CONCLUSION

## CONCLUSION AND FUTURE ENHANCEMENTS CONCLUSION:

This project aimed to tackle a big problem in today's healthcare: the need for tools that can detect Major Depressive Disorder early, in a way that's objective, dependable, and easy to use. The main idea was that traditional methods of diagnosis and single-type automated systems have their limits, but these can be overcome by using a more advanced, multi-layered approach based on Artificial Intelligence and Machine Learning. By combining data from different sources like EEG signals, facial expressions, and survey results, this project created a system that could potentially offer more accurate results than any single method could on its own. After careful planning, building, and testing, the project achieved its goals and proved its main idea was correct.

The process involved creating a full AIML system. One big success was making a strong model for recognizing facial expressions. Using a technique called transfer learning with the VGG16 model, along with methods like data enhancement and stopping early during training, the model handled issues like overfitting and reached a high accuracy of 83.76% in testing. This shows how well the deep learning method worked. Additionally, a basic model for analyzing video, which used a more complex CNN+LSTM setup, was trained and reached an accuracy of 62.5%, showing it could effectively learn from video data over time. The system was built using a late-fusion strategy, which proved to be both strong and effective.

It was designed as a full application with a Python Flask backend that hosted the trained models and a user-friendly frontend for interaction. A thorough testing process, including unit, integration, functional, and system tests, confirmed that the application was not only accurate but also stable and trustworthy.

In summary, this project has shown that a multimodal AIML system is much more powerful and detailed for detecting depression than single-method approaches.

The final system is a solid proof of concept, showing that by looking at what someone is feeling physically, how they behave, and what they report, we can gain a better and more accurate picture of their mental health. This work contributes meaningfully to computational psychiatry and aligns with UN Sustainable Development Goals 3 (Good Health and Well-being) and 8 (Decent Work and Economic Growth).

## FUTURE ENHANCEMENTS:

While this project successfully achieved its core objectives and validated the efficacy of a multimodal approach, it also serves as a robust foundation for numerous exciting avenues of future research and development. The current system is a powerful proof-of-concept; the following points outline potential directions to enhance its capabilities, improve its accuracy, and move it closer to real-world clinical applicability.

Real-Time EEG Integration and Analysis:

The current system utilizes a pre-recorded, static EEG dataset (MODMA). The most significant future enhancement would be to integrate a real-time EEG acquisition module. With the increasing availability of portable, consumer-grade EEG headsets, a future version of the system could be developed to capture and analyze a user's brainwave activity live. This would transform the system from a research tool that analyzes past data into a dynamic, real-time diagnostic aid, capable of providing instant physiological feedback during a clinical session.

Incorporation of Speech and Language Analysis:

To create an even more powerful and holistic multimodal system, a fourth modality—speech analysis—could be incorporated. Vocal characteristics are a rich source of affective information. A new module could be developed to extract features such as pitch, tone, jitter, shimmer, and speech rate from a user's voice. Furthermore, Natural Language Processing (NLP) techniques could be applied to the transcribed text to perform sentiment analysis. Fusing these audio and linguistic features with the existing EEG, facial, and survey data would provide another layer of behavioral insight and would likely lead to a further increase in the system's overall diagnostic accuracy.

Longitudinal Monitoring and Predictive Analysis:

The current system provides a "snapshot" diagnosis based on data from a single session. A powerful future extension would be to develop a system capable of longitudinal monitoring, tracking a user's mental state over an extended period (e.g., weeks or months). By collecting data periodically, the system could learn an individual's personal baseline and identify subtle deviations over time. This would enable the system to move beyond simple classification and towards predictive analysis.

Clinical Validation on a Larger, Diverse Dataset:

While the use of public datasets ensures reproducibility, a critical next step for any medical AI system is rigorous clinical validation. Future work should involve testing and re-training the system on a larger and more demographically diverse clinical dataset, collected in partnership with mental health professionals and institutions. This would be essential for assessing the model's performance and fairness across different age groups, ethnicities, and co- existing conditions, which is a crucial step towards regulatory approval and real-world deployment.

# APPENDICES

## SDG GOAL

### SDG 3 – Good Health and Well Being

This project strongly aligns with SDG 3, which aims to "ensure healthy lives and promote well-being for all at all ages". The system directly supports Target 3.4 by enabling the early detection and prevention of Major Depressive Disorder, a critical step for effective treatment that can stop the problem from getting worse over time. By offering an objective, data-driven tool, it helps reduce the shame around mental health, making people more likely to check on their well-being regularly. Furthermore, the system uses artificial intelligence to offer a low-cost and scalable way to conduct initial screenings, which can help people in distant areas or those who are not comfortable seeing a doctor right away get the first step of care.

### SDG 8 - Decent Work and Economic Growth

The project also contributes meaningfully to SDG 8, which focuses on promoting "decent work for everyone". Mental health is closely connected to how productive people are at work, and untreated depression can lead to absenteeism and burnout. By helping individuals and organizations spot the early signs of mental trouble, the system provides an opportunity for early intervention, which can stop stress from becoming long-term burnout and create a better environment for learning and working. According to the World Health Organization, depression and anxiety cost the global economy trillions of dollars every year in lost productivity. By making it easier to get treatment early, this system helps ensure that mentally healthy people can be more active, creative, and productive, which helps the economy grow in a sustainable way.

## SOURCE CODE

### Survey Code

# --- 1. Select Features and Target ---

# Features (X) are all columns except the target-related ones

X = df.drop(["Depression\_Score","Depression\_Level","depression\_flag"], axis=1) # Target (y) is the binary depression flag

y = df["depression\_flag"]

# --- 2. Split Data for Training and Testing ---

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, stratify=y, random\_state=42)

# --- 3. Build and Train the Random Forest Model --- # Initialize the classifier with 100 decision trees

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Fit the model using the training data model.fit(X\_train, y\_train)

### Facial Recognition:

# --- 2. Build the CNN Model --- image\_model = Sequential([

# Define the input shape and rescale pixel values

Input(shape=(IMG\_HEIGHT, IMG\_WIDTH, 1)),

Rescaling(1./255),

# Convolutional blocks

Conv2D(32, (3, 3), activation='relu', padding='same'),

MaxPooling2D((2, 2)),

Conv2D(64, (3, 3), activation='relu', padding='same'),

MaxPooling2D((2, 2)),

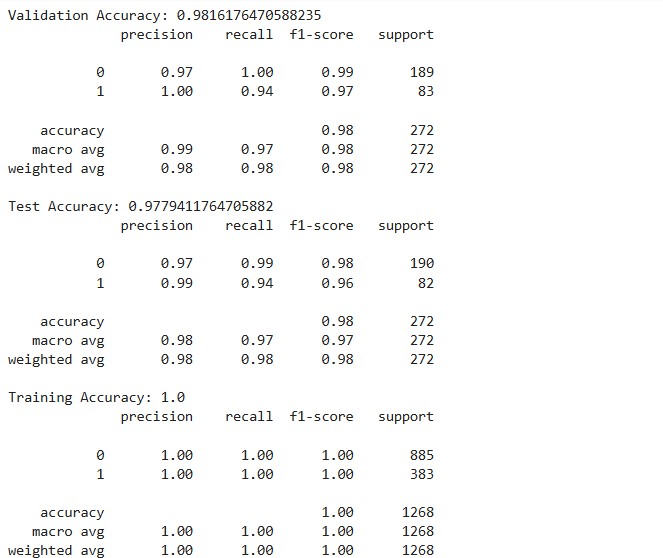
# Flatten and classify Flatten(),

Dense(128, activation='relu'), Dropout(0.5),

Dense(num\_classes, activation='softmax')

])

## SCREENSHOTS

****

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure: A.1 Accuracy ScreenShot (csv files)

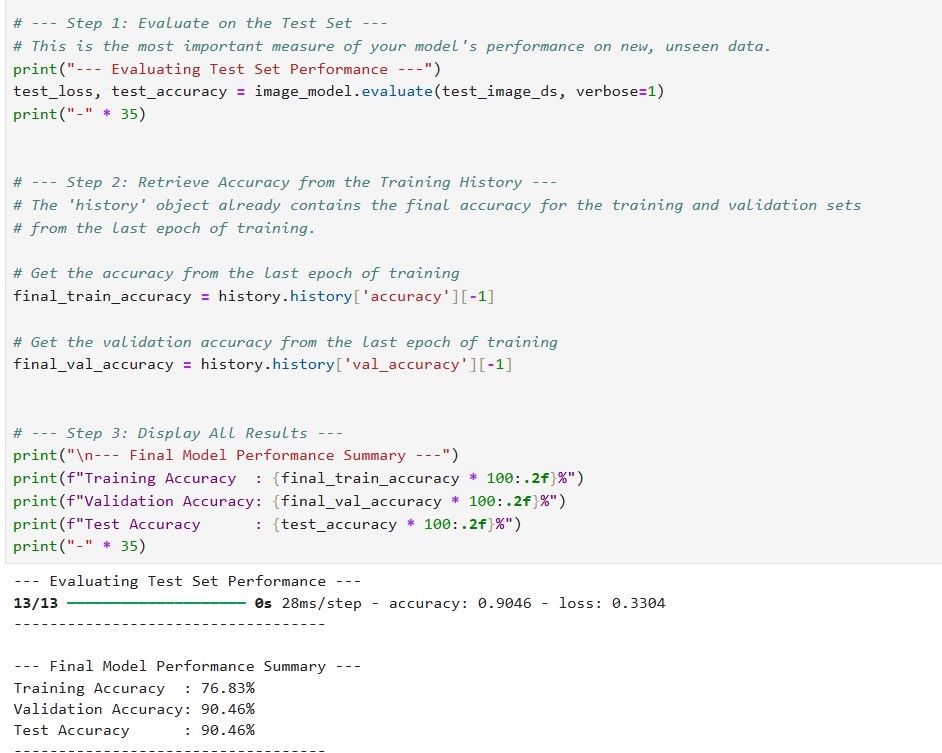
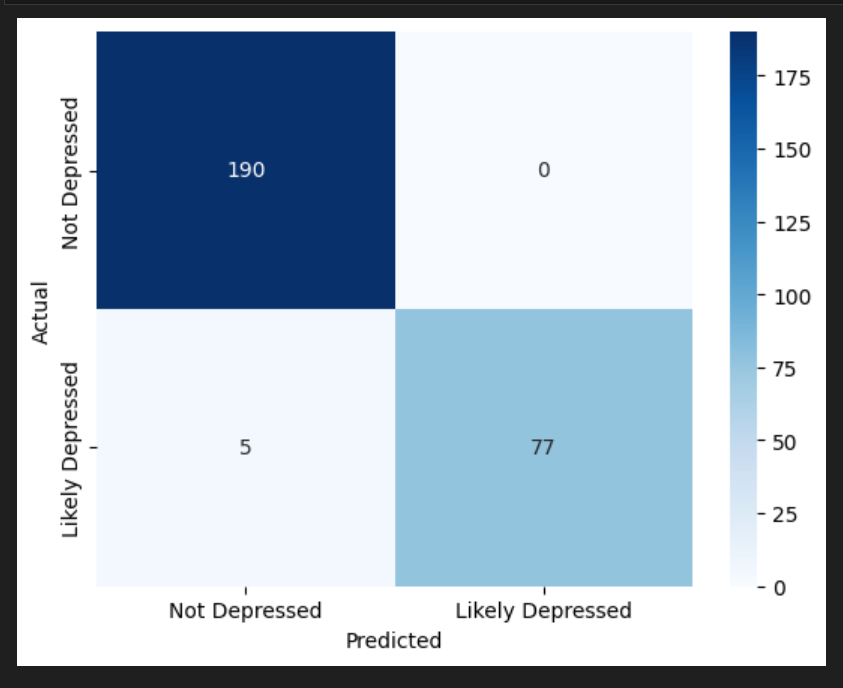


Figure: A.2 Accuracy of facial survey

A blue squares with white text

AI-generated content may be incorrect.

A screenshot of a computer

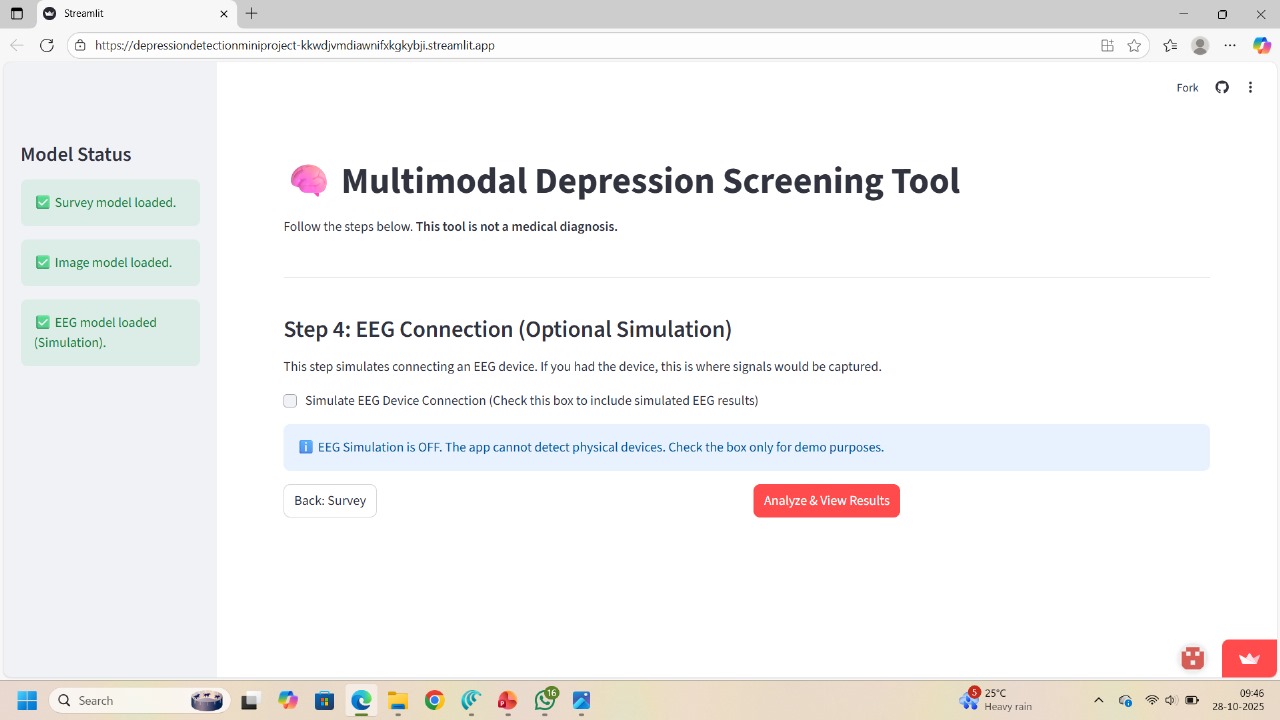
AI-generated content may be incorrect.

A person in a red cape

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Figure: A.4 Final Results

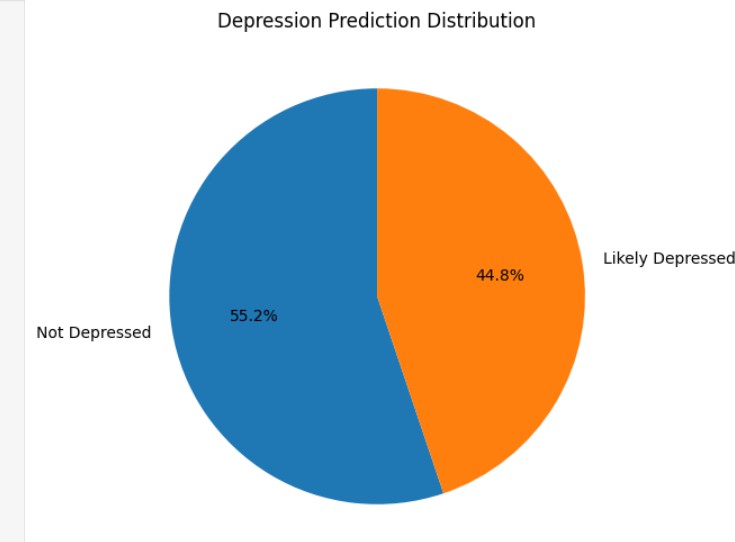


Figure: A.6 Final Depression Metrics

# PLAGIARISM REPORT

# A screenshot of a computer AI-generated content may be incorrect.

# REFERENCES

1. Al-Huseiny, M., et al. (2024) ‘Multimodal Depression Detection Using Audiovisual Features and Deep Learning’.
2. Gupta, V., Saxena, A., Rai, H., Arya, M., Kumar, R., and Bhattacharya, S. (2025) ‘Deep Learning Model for Analyzing EEG Signal Analysis,’ IEEE Xplore, vol. 13, pp. 27478–27490, doi: 10.1109/ACCESS.2025.3563760.
3. He, K., Zhang, X., Ren, S. and Sun, J. (2016) ‘Deep Residual Learning for Image Recognition’, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.
4. Hochreiter, S. and Schmidhuber, J. (1997) ‘Long Short-Term Memory’, Neural Computation, Vol. 9, No. 8, pp. 1735-1780.
5. Jo, Y., & Lee, S. (2024) ‘An Integrated Framework for Depression Detection by Fusing EEG and Facial Expression Data’.
6. Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012) ‘ImageNet Classification with Deep Convolutional Neural Networks’, Advances in Neural Information Processing Systems, Vol. 25.
7. Lagunes-Ramírez, D. A., Rueda-Maya, J. L., Hernandez-Angeles, J. L., and Alvarez, D. (2025) ‘Through the Youth Eyes: Training Depression Detection Algorithms with Eye-Tracking Data,’ IEEE Xplore, vol. 23, no. 1, pp. 150–157, Jan. 2025.
8. Livingstone, S. R. and Russo, F. A. (2018) ‘The Ryerson Audio-Visual

Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English’, PLoS ONE, Vol. 13, No. 5, e0196391.

1. Mumenin, N., Idris, M. Y. I., Dawal, S. S., Yusop, M. Z. M., Wahab, M. H. A., and Abidin, N. H. Z. (2025) ‘“DDNet: A Robust and Reliable Hybrid Machine Learning Model for Effective Detection of Depression Among University Students,”’ IEEE Xplore, vol. 13, pp. 16552–16564, 2025, doi: 10.1109/ACCESS.2025.3552041.
2. Poria, S., Cambria, E., Bajpai, R. and Hussain, A. (2017) ‘A review of affective computing: From unimodal analysis to multimodal fusion’, Information Fusion, Vol. 37, pp. 98-125.
3. Simonyan, K. and Zisserman, A. (2014) ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’, arXiv preprint arXiv:1409.1556.
4. Valstar, M. F., Gratch, J., Schuller, B., Ringeval, F., Lalanne, D., Torres,

M. T., Scherer, S., Stratou, G., Cowie, R. and Pantic, M. (2016) ‘AVEC 2016:

|  |  |  |  |
| --- | --- | --- | --- |
| Depression,  Proceedings | mood,  of the | and  6th | emotion recognition workshop and challenge’,  international workshop on audio/visual emotion |
| Challenge. |  |  |  |