



**Moodify** is a machine learning powered solution that transforms subtle facial expressions and vocal nuances into clear, actionable insights, enabling you to gauge and respond effectively to your audience sentiments during presentations or helping you analyse your pitch for improvements before your high-stakes meetings.

Please check out the latest application at <https://github.com/danieltw/moodify> and start gaining insights into your meetings.

## Moodify 😊 😐 😞

---

### Table of Contents

- [Moodify](#)
  - [Table of Contents](#)
  - [Requirements](#)
  - [Project Structure](#)
  - [Notes](#)
  - [Installation](#)
  - [Additional Setup Instructions](#)
  - [Usage](#)
  - [Docker](#)

### Requirements

1. Python 3.10.7
2. Docker installed on your system
3. Docker-compose installed on your system
4. FFmpeg
  - To check if FFmpeg is install, run the following in terminal

```
ffmpeg -version
```

If you see a version number, that means ffmpeg is installed. Otherwise, please follow the next step to install it.

- Download and install FFmpeg from [FFmpeg](#)
  - For windows, you may find a guide on [PhoenixNap](#)

## Project Structure

```

.
├── app                                <- Moodify web app (backend)
│   ├── Dockerfile                    <- for dockerising app
│   ├── main.py
│   └── src
│       ├── face_detection            <- for detecting faces in videos
│       └── Mood_detector             <- for detecting emotions from facial
expression
│   ├── speech_emotion               <- for detecting emotions in speech
│   ├── speech_to_text               <- for transcription service
│   └── sql                          <- contains sql scripts
├── data                              <- contains datasets for training and testing
│   └── Webcam Images                <- contains the latest facial data for
training and testing the model
│   ├── Webcam Images (Augmented)    <- contains augmented facial data
│   └── sample_videos                <- contains sample data for testing the
application
├── frontend                          <- Moodify web app (frontend)
│   ├── app                          <- main frontend app
│   └── components                   <- frontend components
├── notebooks
│   ├── Classical Learning           <- notebooks for training classical ML for
facial sentiment prediction
│   ├── Neural Networks              <- notebooks for training classical ML for
facial sentiment prediction
│   ├── speech_emotion               <- notebooks for testing speech emotion model
│   └── data_augmentation            <- notebooks for data augmentation of images
├── data collected
│   └── XXX
├── postgres
│   ├── create_tables.sql            <- SQL file for creating the PostgreSQL tables
│   └── insert_values.sql            <- SQL file for inserting the pre-loaded data
in PostgreSQL
├── requirments.txt                  <- required Python packages
├── xxx
└── docker-compose                   <- docker compose file for Moodify

```

## Notes

To use the Moodify application, please follow the instructions in [Installation](#), [Additional Setup Instructions](#) and [Usage](#) sections.

## Installation

### 1. Create virtual environment

```
conda create -n moodify python=3.10.7
```

## 2. Activate the virtual environment

```
conda activate moodify
```

## 3. Install required packages

- If you are using Windows, run

```
pip install -r requirements.txt
```

- If you are using Mac, please switch to Windows. We do not guarantee that the app will run without errors if you choose to run this on Mac due to Tensorflow incompatibilities on Apple silicon.

## Additional Setup Instructions

As the ML model files are too big to be committed to Github, please follow the instructions below to obtain the model files before running the application:

1. Go to [Moodify Google Drive](#) to access all the model files
2. Download the Face Detection weights:
  - In the [Moodify Google Drive](#), go to the [face\\_detection](#) folder and download the [yolov8n-face.pt](#) file into the [app/src/face\\_detection](#) directory
3. Download the Mood Detector model:
  - In the [Moodify Google Drive](#), go to the [Mood\\_detector](#) folder and download the [trained\\_emotion.keras](#) file into the [app/src/Mood\\_detector](#) directory

## Usage

1. Start the PostgreSQL and RabbitMQ docker. In the root directory of the repository, run the following in terminal:

```
docker-compose up -d
```

2. In another terminal, activate the virtual environment and start the Moodify backend application

```
cd app
```

```
conda activate moodify
```

```
python main.py
```

3. In another terminal, install and start the Moodify frontend application

```
cd frontend
```

```
npm i
```

```
npm run dev
```

4. In another terminal, activate the virtual environment and start the Moodify video\_analyser

```
cd app
```

```
conda activate moodify
```

```
python video_analyser.py
```

If the video\_analyser.py stops running / face issues, repeat this step to re-run the script in the terminal.

5. Access the Moodify application at <http://localhost:3000/>

- Open your webbrowser
- Enter the following url: <http://localhost:3000/>
- You may use the videos found in [data/sample\\_videos](#) to test the application

## Docker

Below are the details and instructions specific for the docker containers set-up. You need not run this to use the application.

1. Start the docker containers. In the root directory of the repository, run the following in terminal:

```
docker-compose up -d
```

## 2. Connect to Postgres on PGAdmin

- Host: localhost
- Port: 5455
- User: user
- Password: password Keep the other settings as default DB available at **localhost:5455**

## 3. RabbitMQ available at **localhost:5672**. RabbitMQ Web UI available at **localhost:15672**

## 4. To shutdown the application, run the following command.

```
docker-compose down -v
```