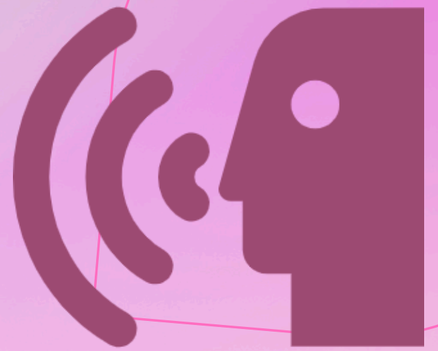
جامعة الإسكندرية
ALEXANDRIA UNIVERSITY
كلية الحاسبات وعلوم البيانات



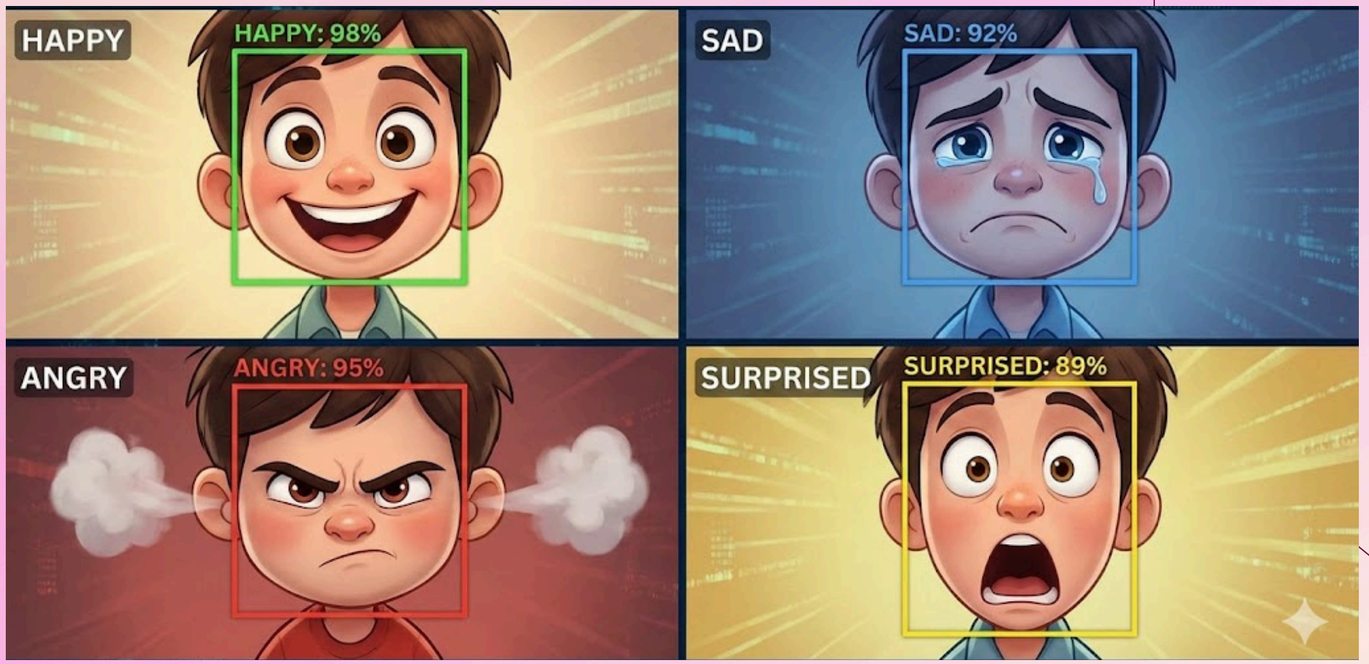
SPEECH RECOGNITION FINAL PROJECT

By:
Ranwa Khaled - 2203153
Nour Nader - 22030156
Nour Adel - 2203169

About

in In this project we create a neural network to classify emotions and we train a machine learning model to detect the speaker along with a friendly user interface

GitHub Repository:
https://github.com/RanwaKhaled/Emotion-Recognition-from-Speech



Dataset

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): https://zenodo.org/records/1188976#.XsAXemgzaUk

Description:

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains 7356 files (total size: 24.8 GB). The dataset contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent.

Expressions:

calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions.

Each expression is produced at two levels of emotional intensity (**normal, strong**), with an additional **neutral** expression.

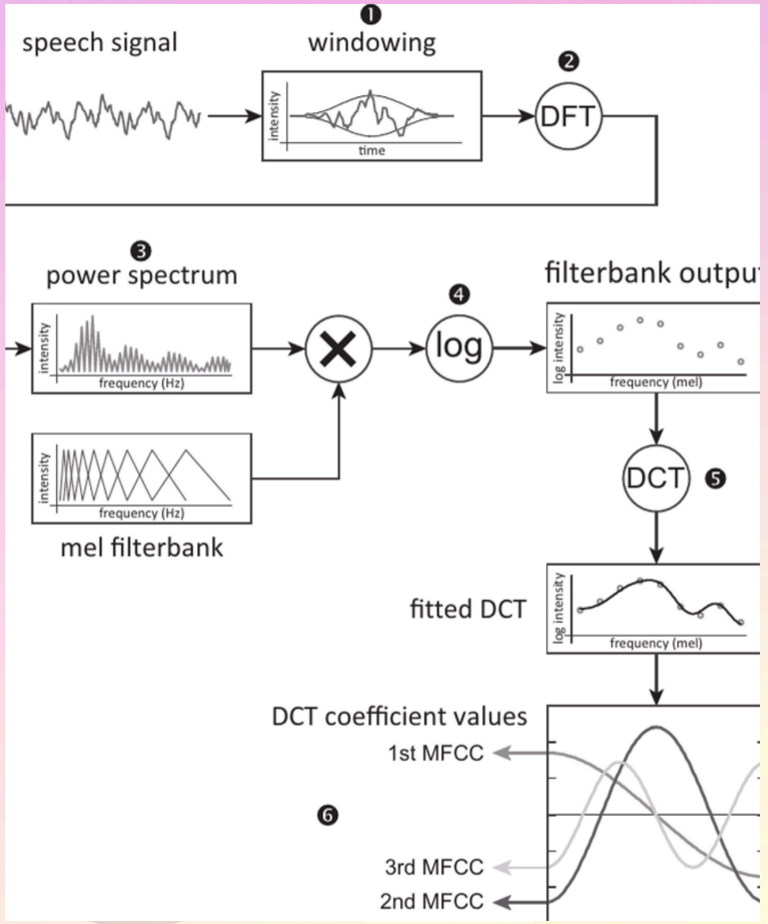
We used on the **normal** tone for each class

Why Feature Extraction?

RAW AUDIO IS JUST A WAVEFORM → TOO **COMPLEX** FOR ML

- We extract **meaningful information** about human speech such as:
  - Tone of voice
  - Pitch variation
  - Energy & loudness
  - Frequency distribution

This allows the model to learn emotional differences **more effectively**



Feature Extraction (Librosa)

WE EXTRACTED 74 ACOUSTIC FEATURES PER AUDIO SAMPLE

- **MFCCs:** Captures vocal timbre and emotion cues
- **Chroma Features:** Represents pitch class distribution
- **Spectral Features:** Measures sharpness & energy spread
- **ZCR:** Detects abrupt pitch changes
- **RMS Energy:** Measures loudness level
- **Pitch:** Emotion-driven frequency variation

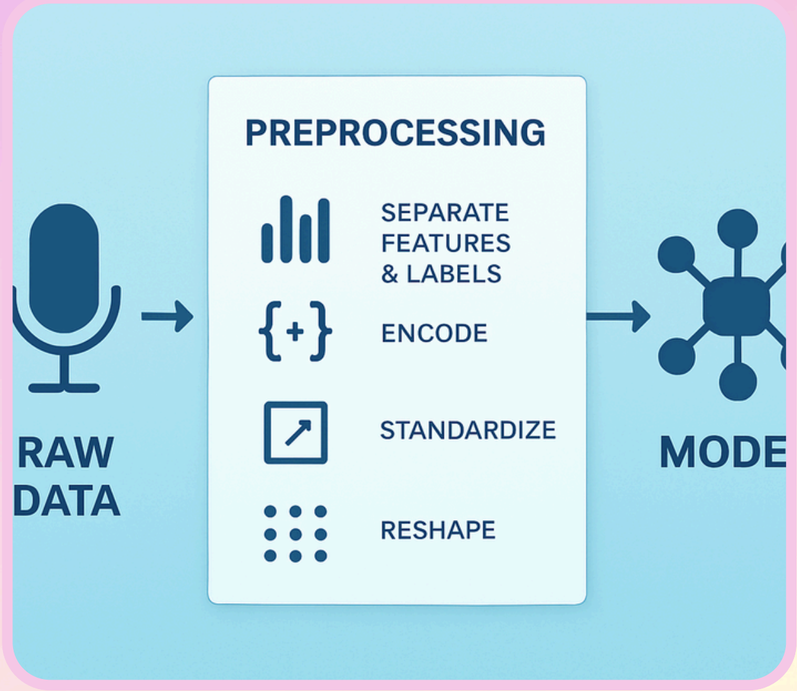
librosa

Data Preprocessing Overview

GOAL: PREPARE DATA SO THE MODEL CAN LEARN EMOTION PATTERNS

STEPS:

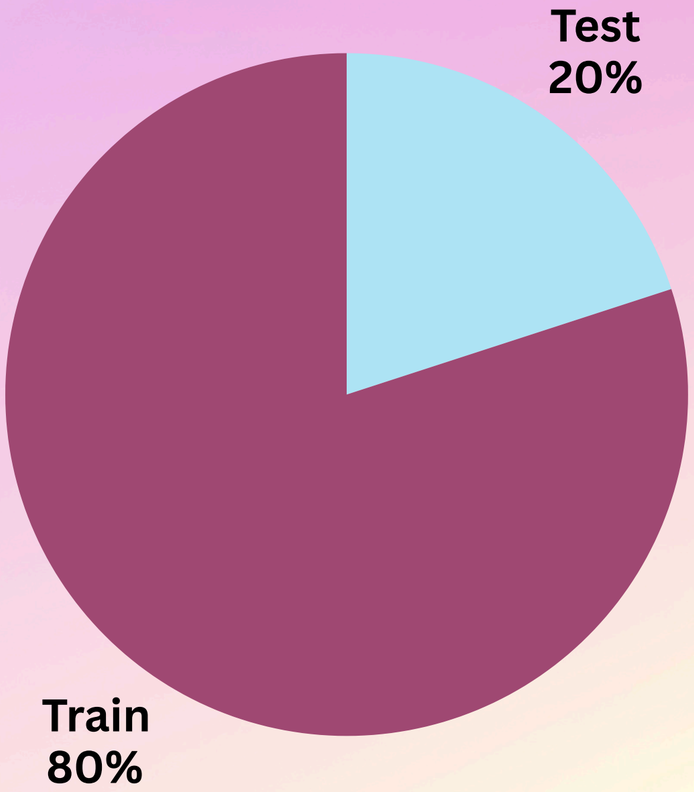
1. SEPARATE FEATURES & LABELS
2. ENCODE LABELS TO NUMERICAL FORMAT
3. STANDARDIZE FEATURE VALUES
4. RESHAPE DATA → LSTM-COMPATIBLE FORMAT



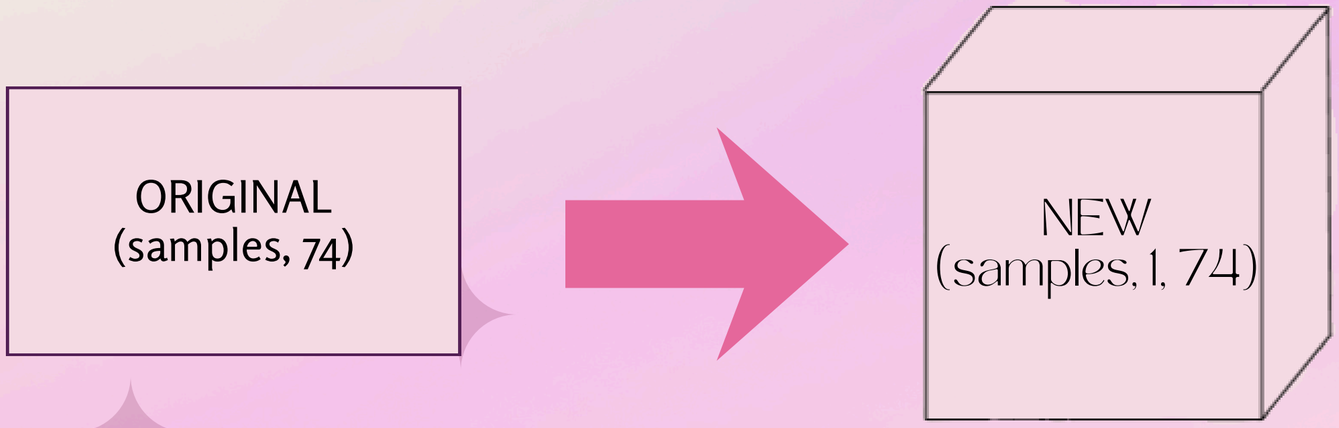
Train/Test Split + Scaling

Train/Test Split: 80% Training / 20% Testing
Stratified split → keeps emotion distribution balanced
Feature Scaling
StandardScaler → makes features similar range (mean 0, std 1)

- Important for better neural network learning
- Saved scaler parameters for future predictions



Shape Preparation for LSTM

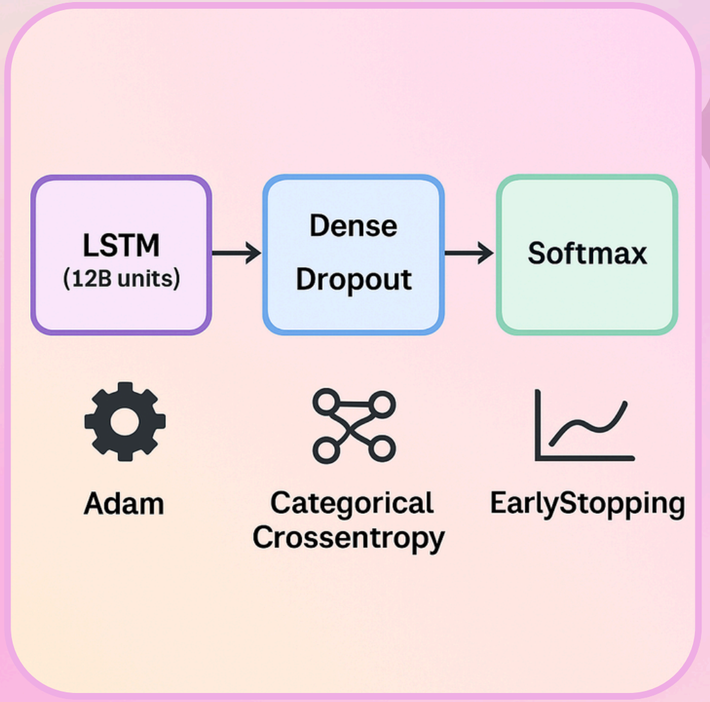


LSTM needs 3D input → *(samples, timesteps, features)*

Original: (samples, 74)

New shape: (samples, 1, 74) → each sample as one timestep

LSTM Model Structure & Training Architecture



LSTM (128 units):

- Dense layer + Dropout (reduce overfitting)
- Softmax → predicts emotion class

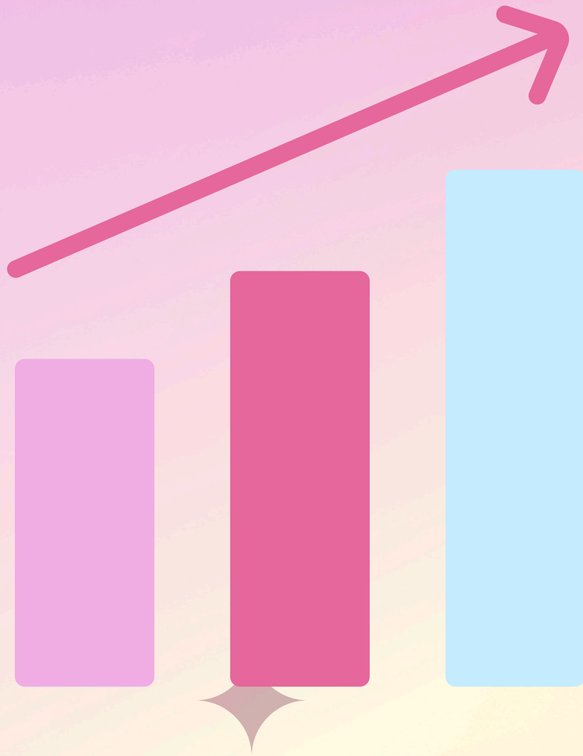
Training:

- Adam optimizer
- Categorical crossentropy
- EarlyStopping → stops when model stops improving

Model Evaluation Metrics

Test Performance Summary

- Overall Accuracy: **72.92%**
- Macro F1-Score: **0.72**
- Evaluated using:
  - Precision
  - Recall
  - F1-Score
  - Confusion Matrix



These metrics allow us to understand:

- Which emotions the model predicts well
- Common confusion between similar emotions

(e.g., Calm ↔ Neutral)

Per-Emotion Performance

Emotion	F1-Score	Interpretation
Calm	0.84	Best recognized emotion
Angry	0.78	Clear acoustic signals, very well detected
Fear	0.78	High recall: model rarely misses fear
Suprise	0.77	Strong prediction consistency
Happy	0.69	Often confused with similar tones
Sad	0.62	Soft tone emotions less distinct
Neutral	0.54	Most challenging due to similarity to Calm

Emotions with strong acoustic patterns performed better

Neutral emotional state is often ambiguous — true for humans too

Results Summary

- LSTM model achieved strong accuracy: **72.92%**
- Best recognized emotions**: Calm, Angry, Fear, Surprise**
- Most confused emotions: **Neutral and Sad**

Overall performance shows effective emotion understanding from speech

Speaker Identification Model

Dataset: A recorded set of audios from each of the project participants of 10 sentences with 3 tones:

- Loud - Normal - Soft

With 2 repetitions of each sample, producing 60 samples with total of 180 samples total.

Dataset Split: Since the dataset is fairly small, it was split to 90%-train, 10%-test

Preprocessing:

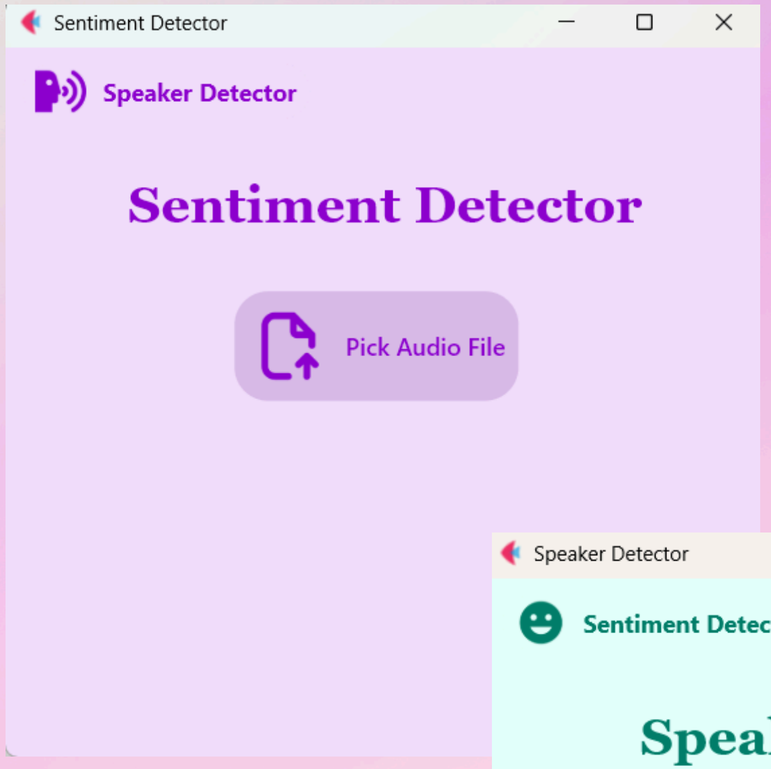
- We use this pre-trained speaker ID model: **speechbrain/spkrec-ecapa-voxceleb** to extract the embeddings of the audio, outputting **192** parameter
- We use **PCA** for dimensionality reduction of the embedding matrices to reduce overfitting bringing them down to **20 component**

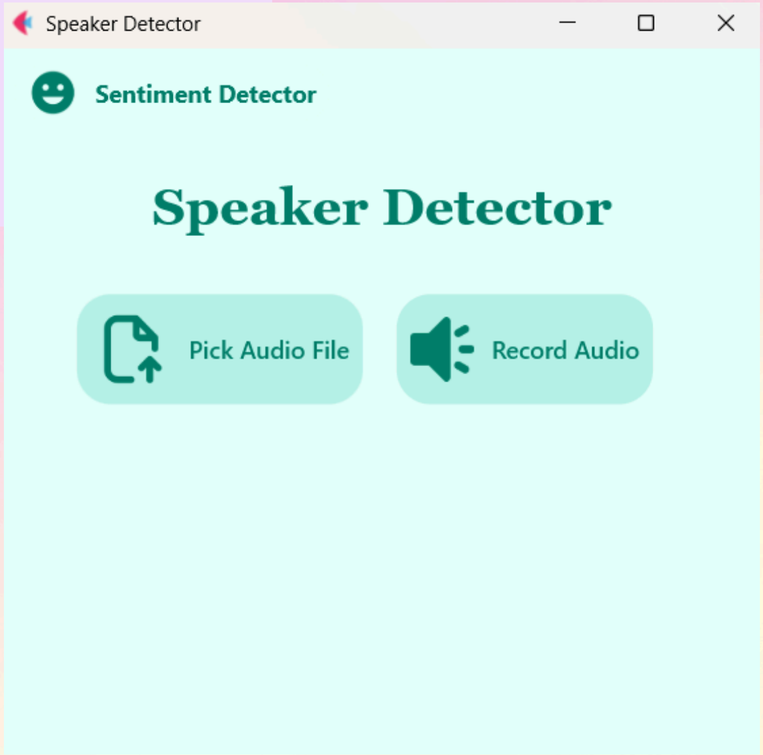
Model:

Logistic Regression → for multiclass classification

Results: The model shows considerable overfitting which was expected due to dataset size but with similar tones to the recorded dataset it is sometimes able to correctly identify the speaker in real time

User Interface





Type: Desktop application

Technology used: flet python library

Features:

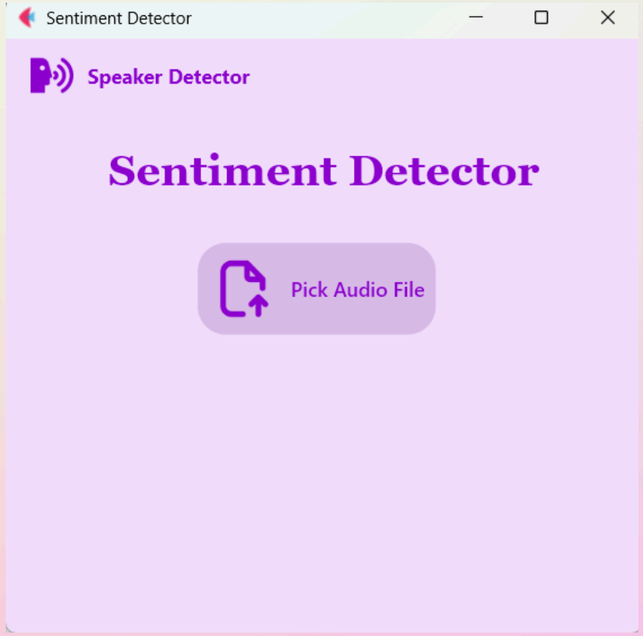
Screen 1: Emotion Detection

Screen 2: Speaker Detection

Figma Design:

https://www.figma.com/design/lZw1W59CN9vu8SUpJkx9j9/speech-project?node-id=0-1&t=xGCNL8Xv3DMeeSPJ-1

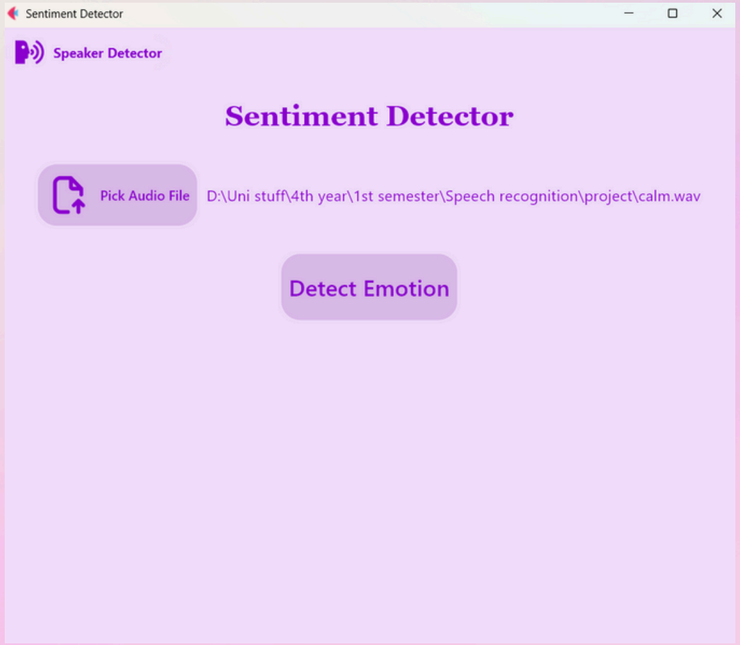
Emotion Detection



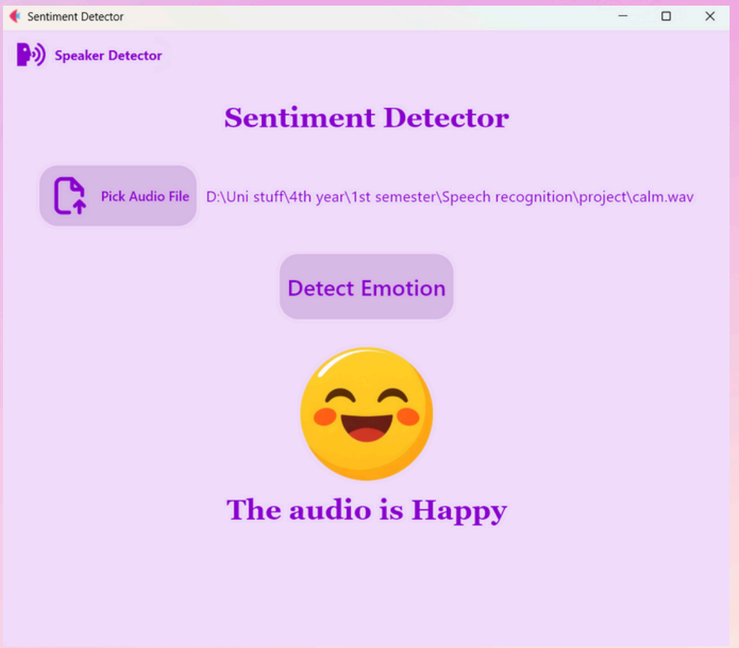
Sentiment Detector Screen:

This is the landing page of the application and it has:

1. A button to navigate to the speaker detector screen in the upper left
2. A button to pick an audio file to detect the emotion (only audio extensions like mp3, wav, m4a...etc. are allowed)

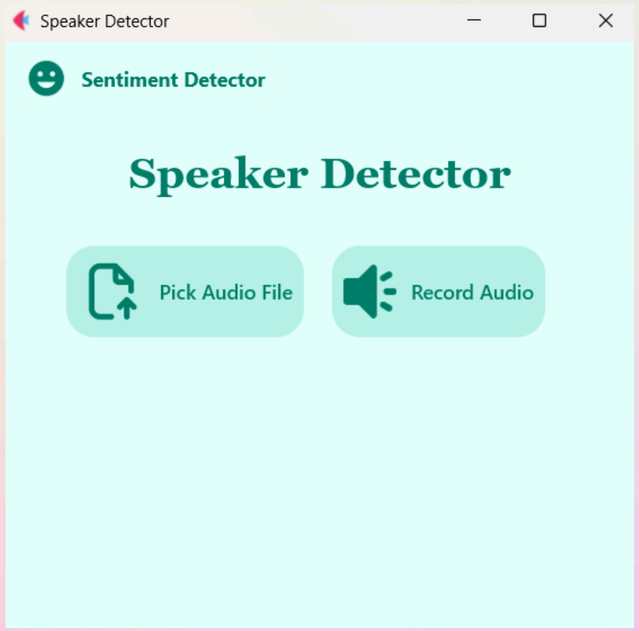


Choosing a file triggers the appearance of the **"Detect Emotion"** button which calls upon the feature extraction function and trained model to prep the new sample and make the prediction.



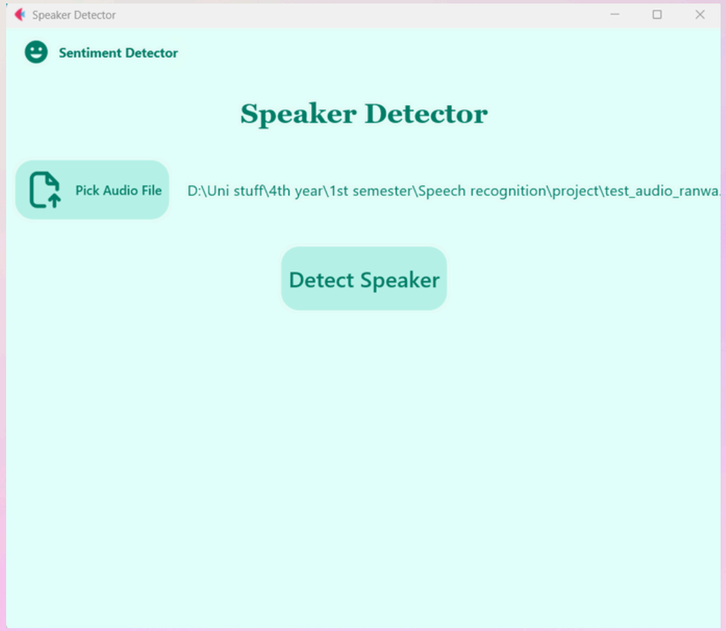
The predicted class is then mapped to its label and an emoji appears corresponding to the predicted emotion

Speaker Detection

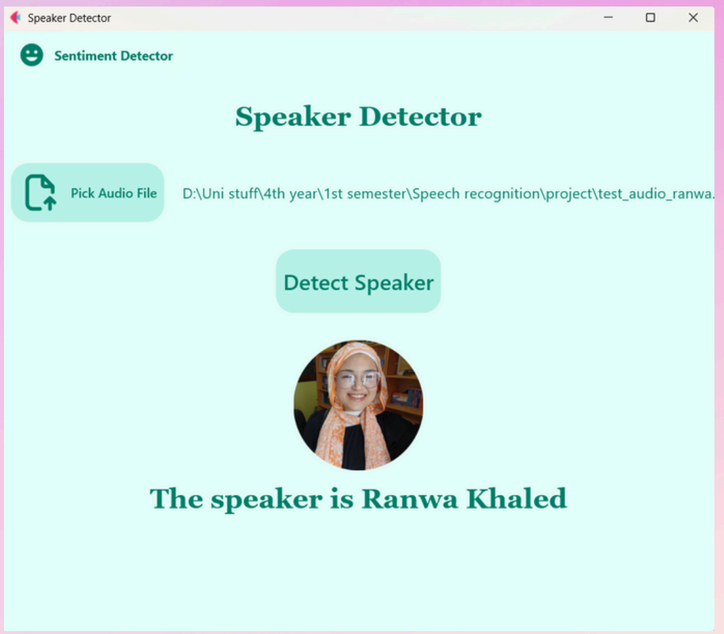


Speaker Detector Screen:
This is the landing page of the application and it has:

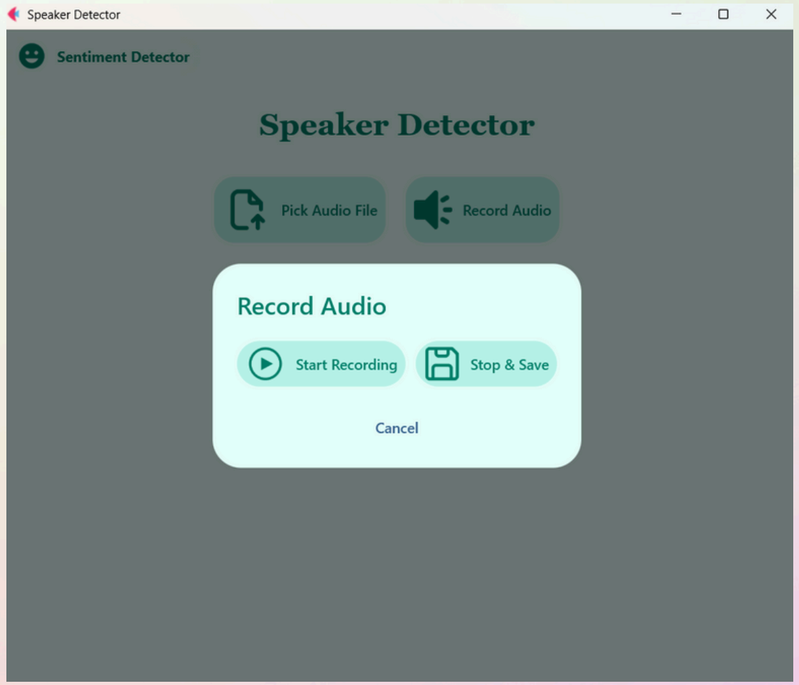
1. A button to navigate to the speaker detector screen in the upper left
2. A button to pick an audio file to detect the emotion (only wav files are allowed)
3. A button to record audio in real time



Choosing a file triggers the appearance of the **"Detect Emotion"** button which calls upon the feature extraction function and trained model to prep the new sample and make the prediction.

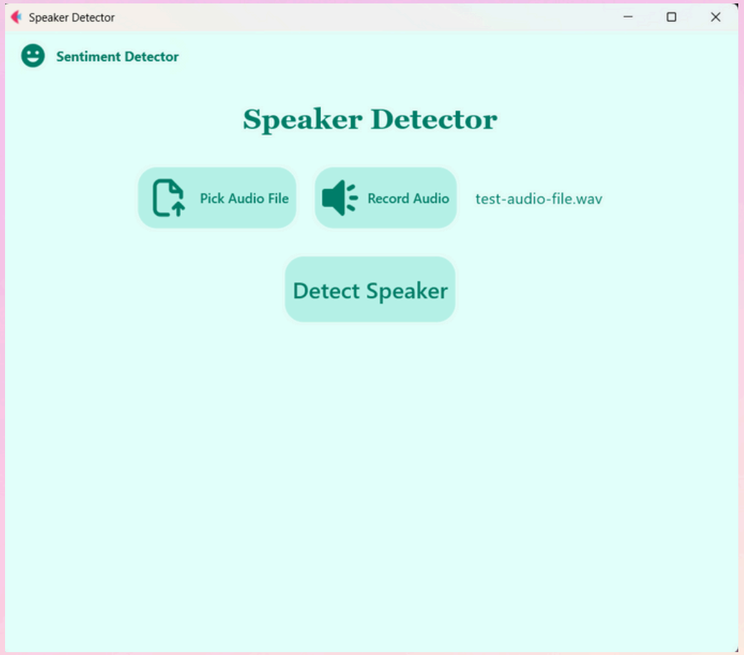


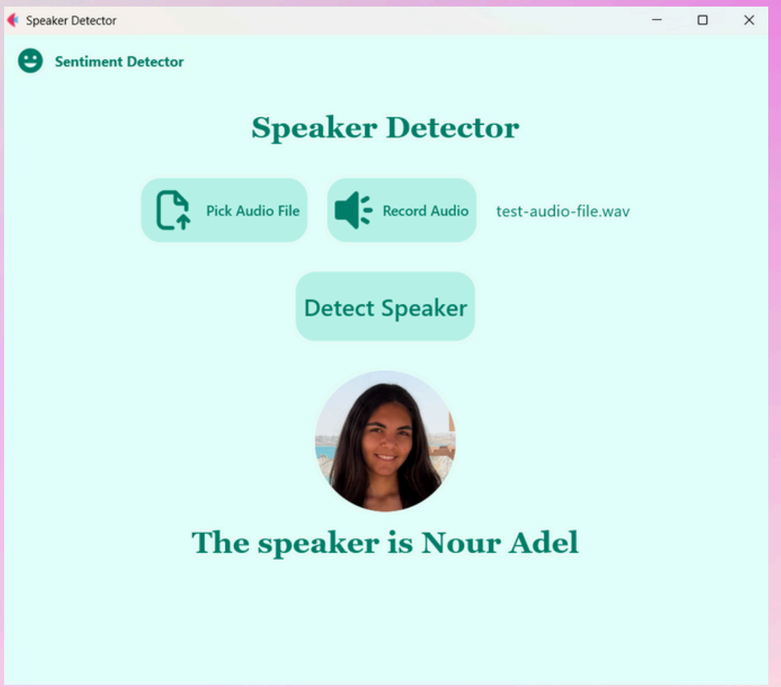
The predicted class is then mapped to the name of the speaker and her picture which then appears on the screen



Choosing to record audio shows a dialogue where a user can start recording and save it or cancel and choose something else

The recorded file is saved in the "*test-audio-file.wav*" path every time a new recording is made, and is then fetched so the user can press the button to identify the speaker





The predicted class is then mapped to the name of the speaker and her picture which then appears on the screen

Thank You!