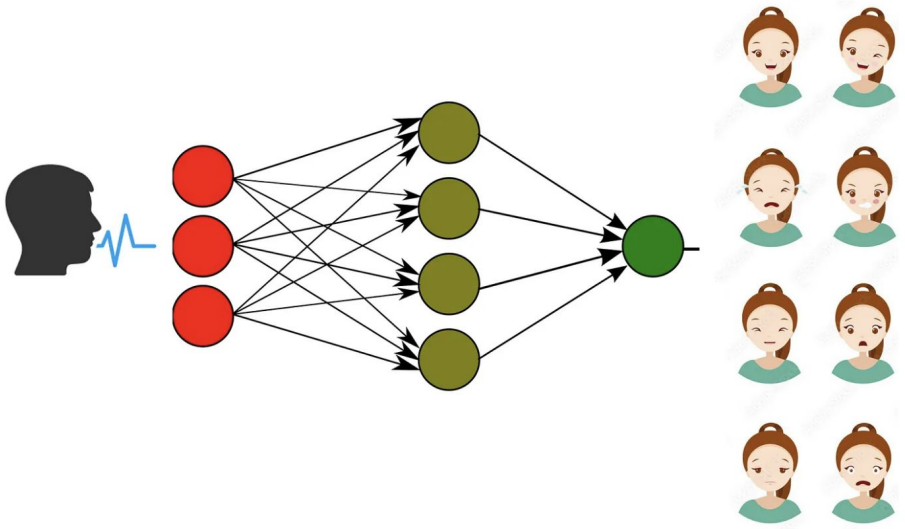
Speech Emotion Recog

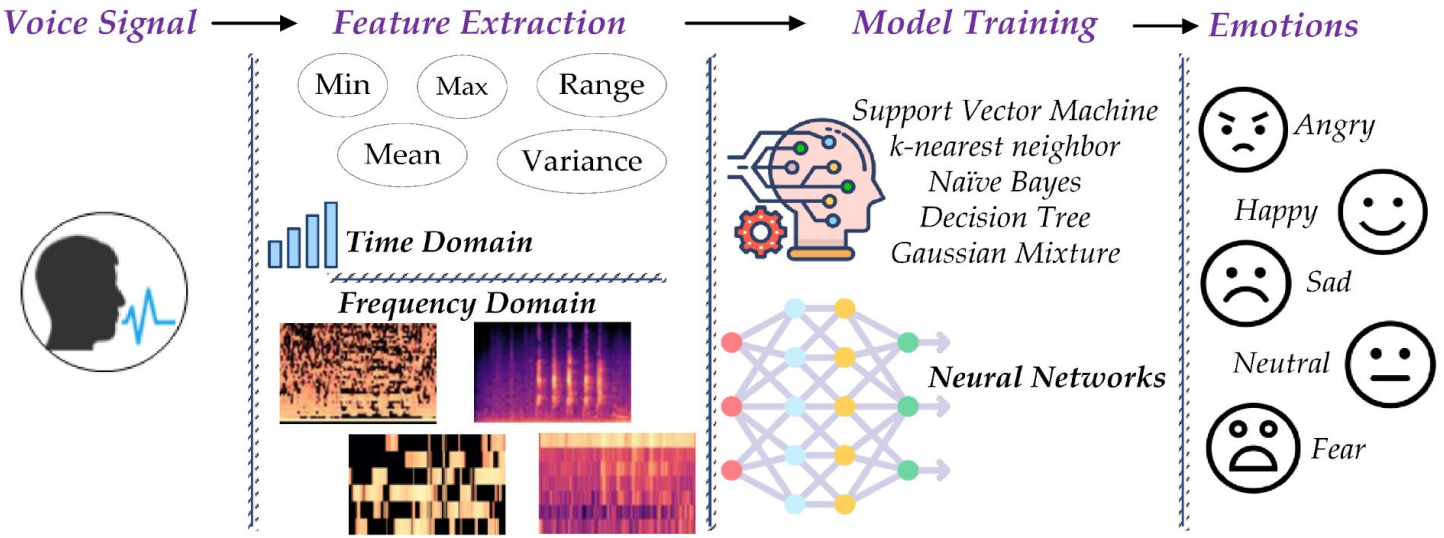


Presented By:

Sahil Kakadiya
Vaibhav Gajera
Prateeksha Ranjan

Introduction

Speech Emotion Recognition (SER) identifies emotions from speech using Machine Learning and Deep Learning Algorithms by analyzing tone, pitch, and frequency to detect happiness, sadness, anger, or neutrality.



Applications Of SER

Video Games 🎮
Adapts difficulty based on player emotions. Hellblade: Senua's Sacrifice

Safety 🚦
Detects driver fatigue in smart cars. Tesla's AI-driven driver monitoring

E-learning 📚
Adjusts lessons based on student emotions. AI tutors like Carnegie Learning

Security 🔒
Identifies stress in fraud detection. Banks analyzing call center voices for scams

Call Centers 📞
Escalates frustrated callers to human agents. NICE CXone AI-driven support

Review Systems ⭐
Analyzes sentiment in voice reviews. Amazon Alexa's voice feedback

RAVDESS – A High Quality SER Dataset

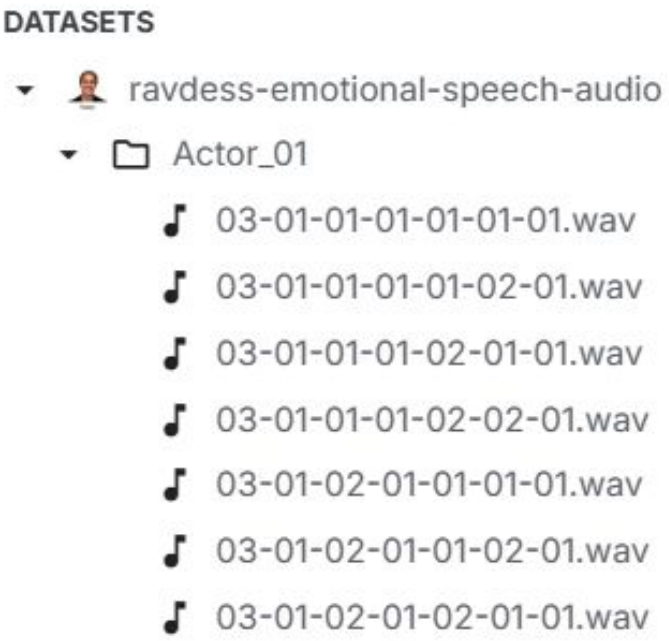
Dataset Overview

🎧 **Total Files:** 1,440 speech samples

🎭 **Actors:** 24 professionals (12M, 12F)

😃 **Emotions:** Neutral, Calm, Happy, Sad, Angry, Fearful, Disgust, Surprised

🎼 **Format:** 16-bit, 48kHz **.wav**



File Naming Breakdown

Modality (03): Audio-only

Emotion (06): Fearful

Statement (02): "Dogs are sitting by the door"

Actor ID (12): Female (even-numbered)

Preprocess Data for Better SER Performance

Preprocessing ensures **clean, consistent data** by removing noise and standardizing audio quality, which enhances the efficiency and accuracy of SER models.

Visualizing Raw Audio

- Plotted **Waveform & Spectrogram** to analyze audio characteristics
- **Issue Identified:** Silent segments in samples increased data size and processing time

Audio Preprocessing Steps

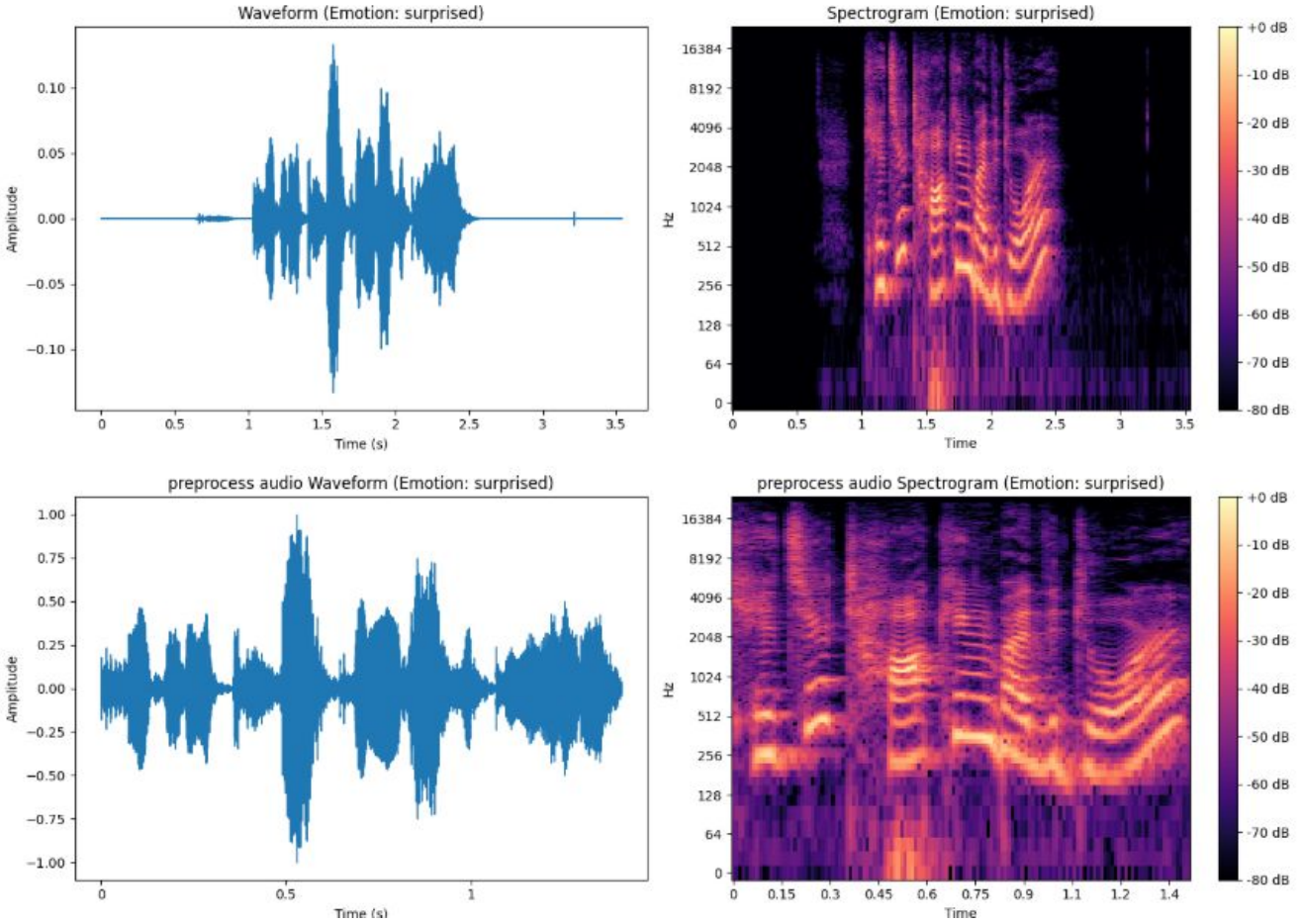
- **Method:** Used librosa.effects.trim() with a **20 dB threshold**
- **Benefit:** Eliminated low-energy segments, reducing file size and focusing on meaningful data

Normalization

- **Method:** Applied librosa.util.normalize() for consistent amplitude
- **Benefit:** Ensured uniform audio levels across samples

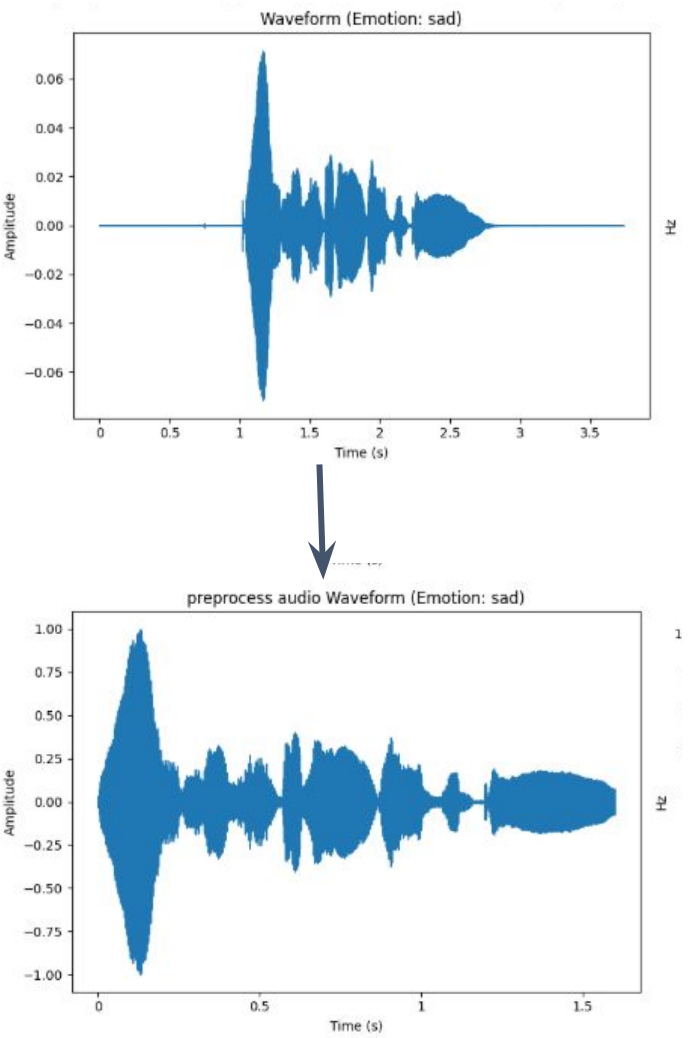
Sample Plot for Surprised Emotion

Modality: 03, Vocal Channel: 01, Intensity: 01, Statement: 01, Actor ID: 02, Actor_Gender: Female



Benefits of Audio Preprocessing for SER

- **Cleaner Waveforms**
  - Removes silent gaps, resulting in continuous speech signals that improve feature extraction accuracy.
- **More Compact Spectrograms**
  - Refines audio, highlighting key features for emotion classification, making the spectrograms more focused and relevant.
- **Reduced Data Size**
  - Minimizes data size while retaining essential speech features, boosting model efficiency without losing critical information.



Extracted Features

- **MFCC (Mel-Frequency Cepstral Coefficients):** Captures key speech patterns by mimicking human hearing, focusing on lower frequencies where emotional cues are stronger.
- **Delta & Delta-Delta MFCCs:** Capture how speech patterns evolve over time — Delta represents the rate of change, while Delta-Delta captures the acceleration, helping to identify emotional variations.
- **Mel Spectrogram:** Visualizes speech energy across Mel frequencies over time, highlighting frequency patterns linked to different emotions.

```
# Choose feature type ('MFCC' or 'WAV2VEC')
feature_type = "MFCC-48000-40"
MFCC_X, MFCC_y = data_prep.load_dataset(feature_type

print(f"\nMFCC_X shape: {MFCC_X.shape}")
print(f"MFCC_y shape: {MFCC_y.shape}")
```

```
Method: MFCC
Sample Rate: 48000
Number of Coefficients: 40

🔍 Scanning dataset directory: /kaggle/input/ravdess-emot

✅ Found 1440 valid audio files.

Extracting the MFCC type feature with sample_rate: 48000
```

```
✅ Dataset Extracted and Loaded Successfully!
Total audio files processed: 1440
Emotion distribution:
  surprised: 192 files
  neutral: 96 files
  disgust: 192 files
  fearful: 192 files
  sad: 192 files
  calm: 192 files
  happy: 192 files
  angry: 192 files

MFCC_X shape: (1440, 160)
MFCC_y shape: (1440,)
```

Model Training & Optimization

Training Configuration

- **Model Types:** LSTM, CNN, CNN+LSTM
- **Hyperparameters:**
  - **Epochs:** Max to 150
  - **Batch Size:** 36
  - **Learning Rate:** 0.001
  - **Loss Function:** Focal Loss
  - **Optimizer:** Adam

```
focal_loss(alpha=0.25, gamma=2.0):
    def loss_fn(y_true, y_pred):
        cce = SparseCategoricalCrossentropy(reduction=tf.keras.losses.Reduction.NONE)
        cross_entropy = cce(y_true, y_pred)
        pt = tf.exp(-cross_entropy)
        focal_weight = alpha * (1 - pt) ** gamma
        focal_loss = focal_weight * cross_entropy
        return tf.reduce_mean(focal_loss)
    return loss_fn
```

Model Training & Optimization

- **Optimization Strategies:**
  - **ReduceLROnPlateau:** Dynamically adjusts learning rate for convergence.
  - **Early Stopping:** Stops training when validation loss plateaus to prevent overfitting.
- **Training Process:**
  - Compiled with **accuracy** as the metric.
  - **Adaptive learning** using callbacks for better generalization.

Baseline Model: LSTM

Architecture: LSTM (256 units) + Dense (128, 64 units) + Softmax for 8 emotions

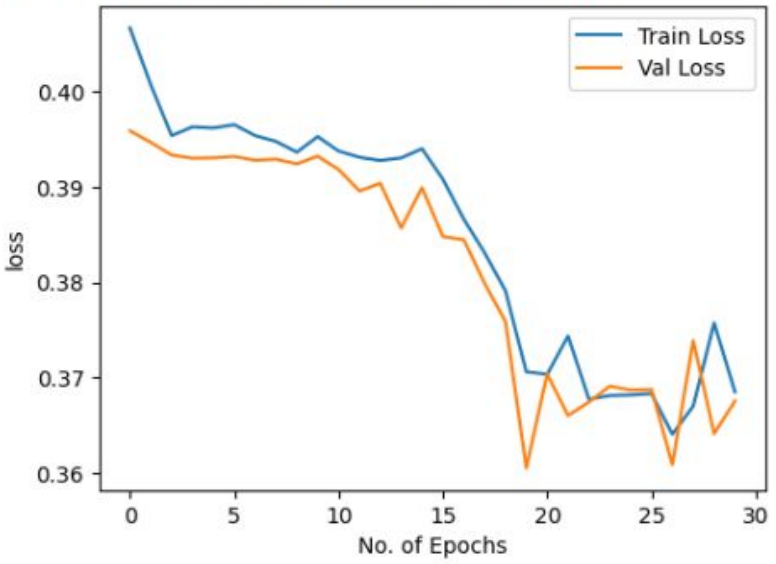
Parameters: 305,864 trainable

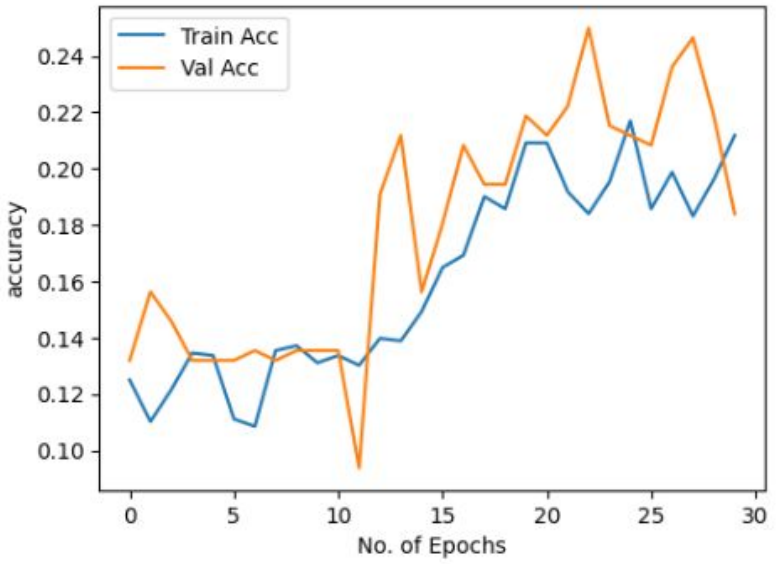
Performance:

- **Train Accuracy:** 21.61%
- **Test Accuracy:** 21.88%

Observations:

- LSTM struggles with meaningful feature extraction.
- "Angry" class: High recall (0.82), low precision (0.21).
- Other emotions: Zero recall/precision.





Baseline Model: CNN

Architecture: 3 Conv1D layers (64, 128, 256 filters), MaxPooling1D, Dropout, Dense layers, Softmax

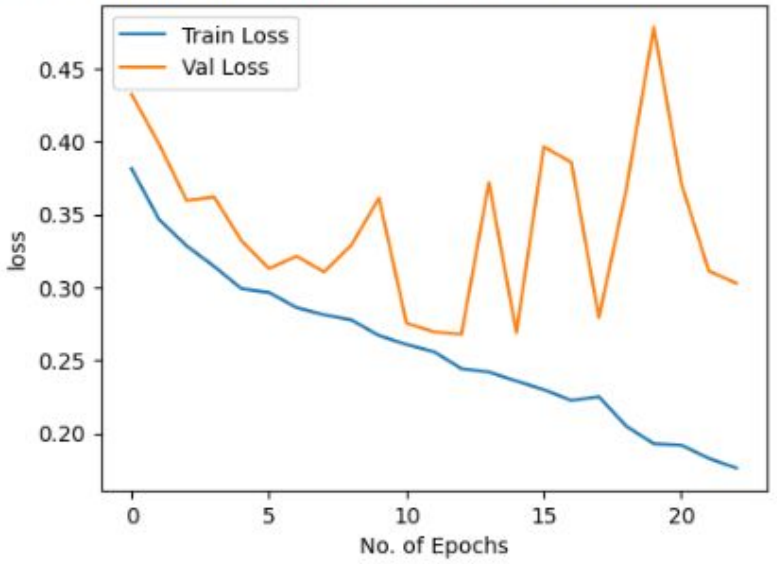
Parameters: 166,984 trainable, 50% Dropout

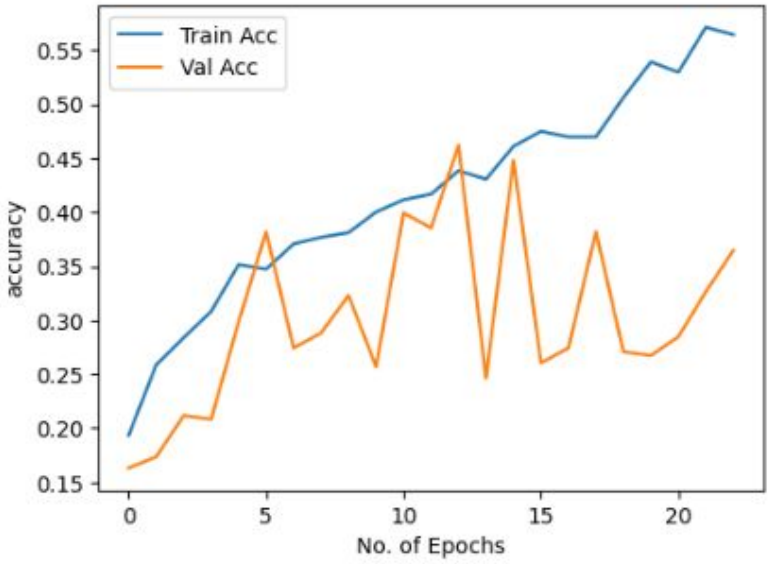
Performance

- **Train Accuracy:** 45.57%
- **Test Accuracy:** 46.18%

Observations:

- Outperforms LSTM in feature extraction.
- Best at recognizing "Calm" (82% recall).
- Struggles with "Fearful," "Disgust," and "Surprised."





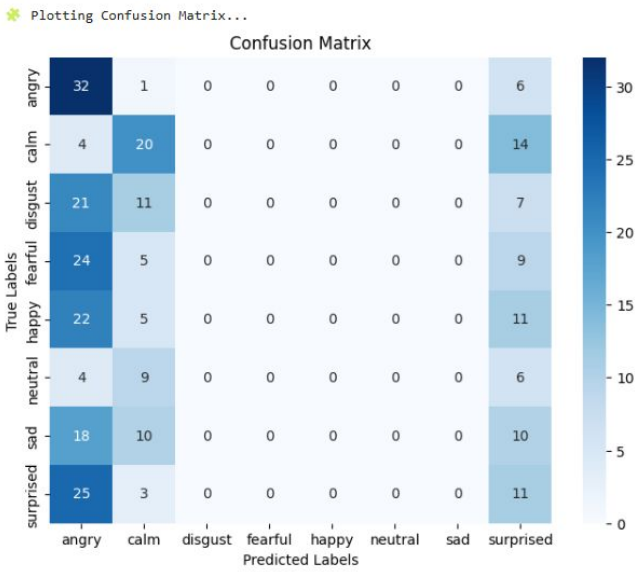
Performance Analysis (LSTM to CNN)

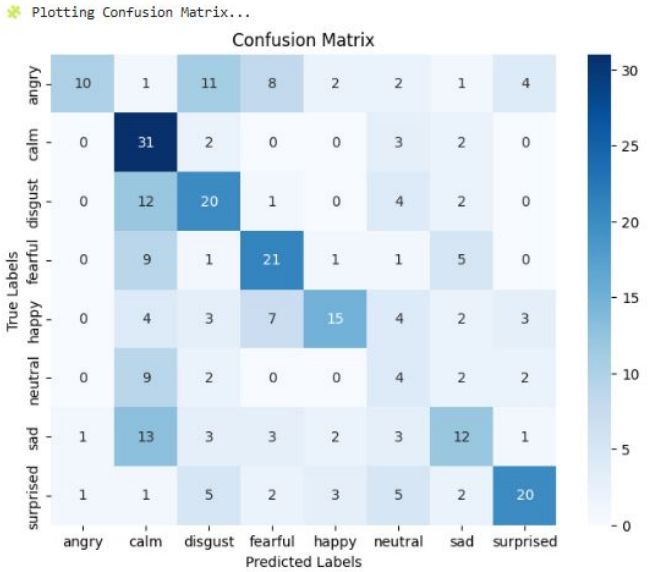
Accuracy Gain:

- **LSTM: 21.88% → 46.18%** (gradual learning)
- **CNN:** Better feature extraction from spectrograms

Class Differentiation:

- **LSTM:** Higher recall for "Angry" & "Calm"
- **CNN:** Stronger precision & recall, especially for **"Calm"** & **"Surprised"**





Baseline Model : CNN + LSTM

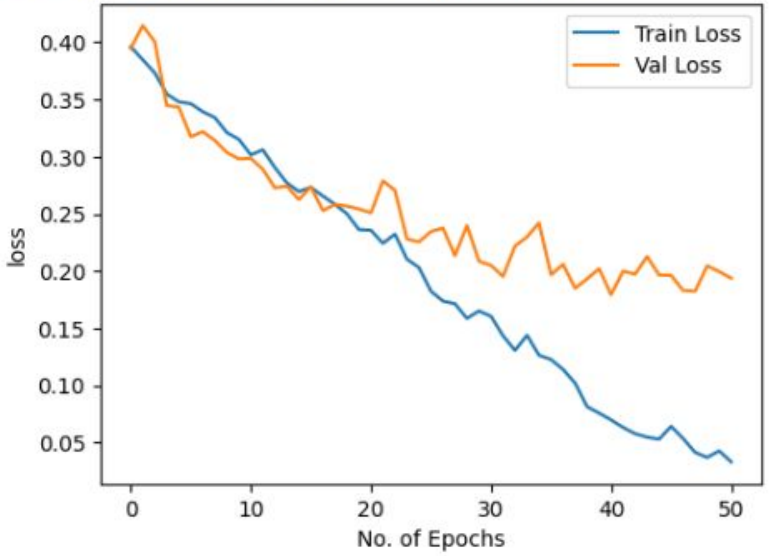
- **Architecture:** Conv1D layers for feature extraction + LSTM for sequential learning
- **Parameters:** 378,344 total

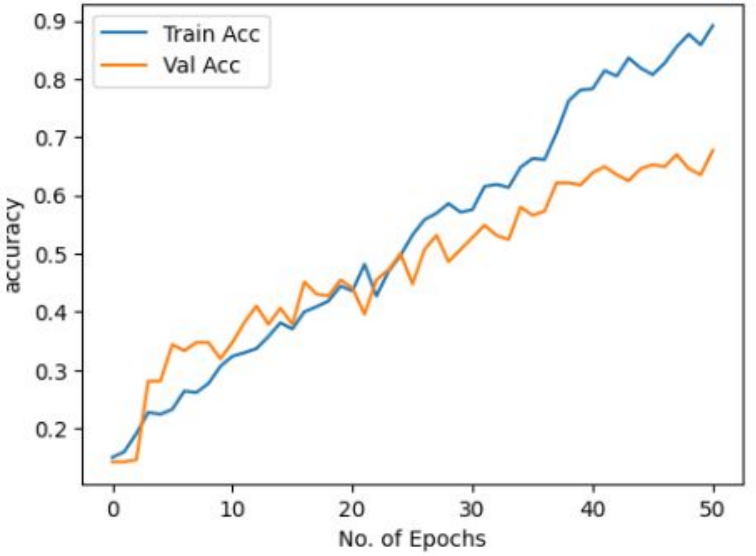
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 160, 64)	256
batch_normalization (BatchNormalization)	(None, 160, 64)	256
max_pooling1d (MaxPooling1D)	(None, 80, 64)	0
conv1d_1 (Conv1D)	(None, 80, 128)	24,704
batch_normalization_1 (BatchNormalization)	(None, 80, 128)	512
max_pooling1d_1 (MaxPooling1D)	(None, 40, 128)	0
dropout_4 (Dropout)	(None, 40, 128)	0
conv1d_2 (Conv1D)	(None, 40, 256)	98,560
batch_normalization_2 (BatchNormalization)	(None, 40, 256)	1,024
max_pooling1d_2 (MaxPooling1D)	(None, 20, 256)	0
lstm_2 (LSTM)	(None, 20, 128)	197,120
lstm_3 (LSTM)	(None, 64)	49,408
dense_6 (Dense)	(None, 64)	4,160
dropout_5 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 32)	2,080
dropout_6 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 8)	264

Total params: 378,344 (1.44 MB)
Trainable params: 377,448 (1.44 MB)
Non-trainable params: 896 (3.50 KB)

CNN + LSTM Model Evaluation

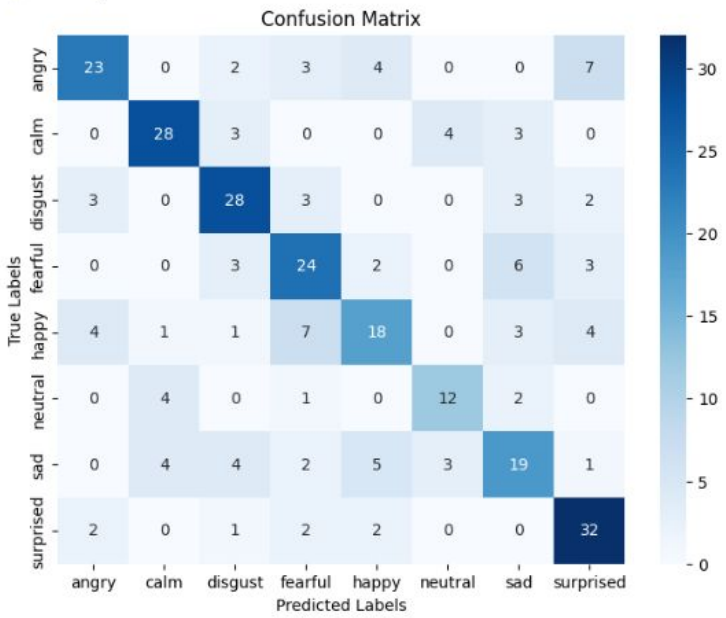
- **Train Accuracy: 88.28% | Test Accuracy: 63.89%**
- **Key Insights:**
  - Best performance among models, leveraging both spatial and temporal features
  - Balanced precision/recall (~60-75%) across most emotions
  - "Surprised" (82% recall) & "Calm" (74% recall) best detected
  - High training accuracy suggests slight overfitting but strong generalization





Baseline Model : CNN + LSTM

- **Accuracy Improvement: Test Accuracy: 63.89%** (Significant improvement with better generalization)
- **Class Performance:** Better precision & recall, especially for "Calm," "Surprised," and "Disgust"



Train Accuracy: 88.28%
Test Accuracy: 63.89%

Generating Classification Report...

Classification Report:

	precision	recall	f1-score	support
angry	0.72	0.59	0.65	39
calm	0.76	0.74	0.75	38
disgust	0.67	0.72	0.69	39
fearful	0.57	0.63	0.60	38
happy	0.58	0.47	0.52	38
neutral	0.63	0.63	0.63	19
sad	0.53	0.50	0.51	38
surprised	0.65	0.82	0.73	39
accuracy			0.64	288
macro avg	0.64	0.64	0.64	288
weighted avg	0.64	0.64	0.64	288

Evaluation: CNN v/s LSTM v/s LSTM+CNN

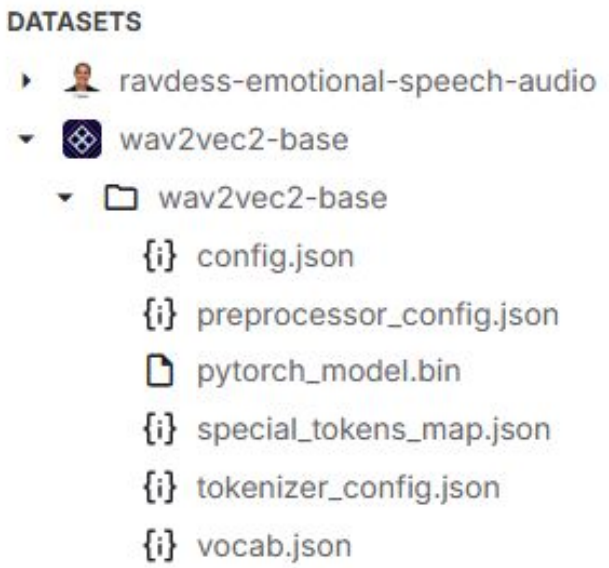
Model	Train Accuracy	Test Accuracy	Best Recognized Emotion	Weakest Recognized Emotion
LSTM	21.61%	21.88%	Angry (82% recall)	Disgust, Happy, Sad (0% recall)
CNN	45.57%	46.18%	Calm (82% recall)	Neutral (21% recall)
CNN+LSTM	88.28%	63.89%	Surprised (82% recall)	Happy (47% recall)

Key Takeaways:

- **CNN+LSTM** achieves the best performance by combining feature extraction with sequential learning.
- **CNN outperforms LSTM**, proving that speech spectrograms hold rich spatial features.
- **LSTM struggles alone**, showing that temporal modeling alone is insufficient for speech emotion recognition.

Advanced Model : Wav2Vec 2.0

- Transformer-based self-supervised model for raw audio (**Facebook AI**).
- Learns speech features without labeled data.
- **Benefits:**
  - Captures detailed audio features, boosting accuracy.
  - Reduces need for large labeled datasets.
  - Adapts across languages & speech variations.
- **Implementation Details:**
  - Used wav2vec2-base for feature extraction.
  - Fine-tuned the model on the emotion classification task.
  - Model files used:
    - config.json, preprocessor_config.json
    - pytorch_model.bin (Pre-trained weights)
    - tokenizer_config.json, vocab.json



Wav2Vec2 Feature Extraction & Random Forest Classification

- **Feature Extraction:**
  - Resamples audio (16kHz), extracts embeddings
  - Outputs feature matrix [time_steps, feature_dim]
- **Random Forest Classifier:**
  - Trains on extracted features
  - Predicts on train & test data

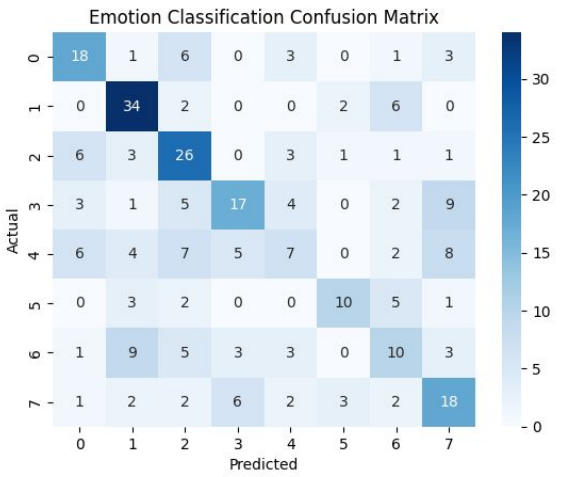
```
✅ Dataset Extracted and Loaded Successfully!
Total audio files processed: 1440
Emotion distribution:
  surprised: 193 files
  neutral: 96 files
  disgust: 192 files
  fearful: 192 files
  sad: 192 files
  calm: 192 files
  happy: 192 files
  angry: 192 files

WAV2VEC_X shape: (1440, 170, 768)
WAV2VEC_y shape: (1440,)
```

Model Performance Analysis

- **Overfitting:** Train Accuracy = 100%
  - Test Accuracy = 48.61% → Poor generalization.
- **Best Class:** Class 1 (Precision: 0.60, Recall: 77%).
- **Worst Class:** Class 4 (Precision: 0.32, Recall: 18%).
- **Class Imbalance:** Poor performance in smaller classes, especially Class 4.

```
Making predictions on the test set...
Predictions completed.

Train Accuracy: 100.00%
Test Accuracy: 48.61%

Generating Classification Report...

Classification Report:
              precision    recall  f1-score   support

           0       0.51      0.56      0.54        32
           1       0.60      0.77      0.67        44
           2       0.47      0.63      0.54        41
           3       0.55      0.41      0.47        41
           4       0.32      0.18      0.23        39
           5       0.62      0.48      0.54        21
           6       0.34      0.29      0.32        34
           7       0.42      0.50      0.46        36

    accuracy                           0.49       288
   macro avg       0.48      0.48      0.47       288
weighted avg       0.48      0.49      0.47       288
```

Plotting Confusion Matrix...



Conclusion & Future Work

Model Performance

- CNN+LSTM effectively extracts features (CNN) and learns sequences of sounds (LSTM).
- Achieves competitive accuracy for speech emotion classification.

Future Improvements

- Hyperparameter tuning & more training data.
- Train attention mechanisms & pre-trained models for better accuracy.
- Addressing class imbalance for improved generalization.

Next Steps

- Implement wav2vec2-large-xlsr-53 for enhanced feature extraction.
- Leverage its larger architecture for better accuracy in emotion classification.

Thank You

For Your Attention!

Any Questions