BA865 Project Proposal
Group 1
Jishnu Moorthy , Yashvardhan Singh Ranawat

Problem Statement & Motivation

Identifying human emotions from audio recordings has various applications in fields of customer service, patient health monitoring and emergency services.

However, there are significant challenges due to the complex nature of emotional expression. By developing a neural network model to classify emotions from audio data, we can enable better natural and empathetic understanding between humans using technology, leading to improved user experiences and business outcomes.

For example, such a model could be used to improve customer service experience for large organizations or be used by emergency service operators to gauge emotions of callers under extreme circumstances.

Dataset

For this project, we will be primarily focusing on using the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset. This dataset contains 7,356 audio files from 24 professional actors (12 female, 12 male) vocalizing statements in a neutral North American accent while expressing 8 different emotions: calm, happy, sad, angry, fearful, disgust, surprised, and neutral.

Link to datasets:

RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) :
https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio

SAVEE (Surrey Audio-Visual Expressed Emotion) :
https://www.kaggle.com/datasets/ejlok1/surrey-audiovisual-expressed-emotion-savee

EMODB (Emotional Speech Database for Classification Problem)
https://www.kaggle.com/datasets/piyushagni5/berlin-database-of-emotional-speech-emodb

BA865 Project Proposal
Group 1
Jishnu Moorthy , Yashvardhan Singh Ranawat

Complexity and Handling:

To dive deeper into emotional analysis through audio, we decided to pursue the following :

1. Cross-Cultural Emotion Recognition: The RAVDESS dataset is limited to North American speakers. To make the model more robust and generalizable, we can incorporate datasets like the SAVEE dataset, which includes speakers from the UK, and the EMODB dataset, which includes German speakers.
2. Temporal Dynamics of Emotion: Instead of treating each audio sample independently, we can explore recurrent neural network (RNN) architectures, such as LSTMs, to capture the temporal evolution of emotional expressions over the duration of an audio clip.

We will focus solely on the audio data from the provided datasets, and we will use the Kaggle API to efficiently import these datasets, as manual uploads of audio files can be computationally challenging. If we encounter any computing issues, we will employ stratified sampling techniques to ensure a balanced representation of data from all emotion classes in our sampled datasets.

Proposed Methodology

1. Baseline Model: We will start with a simple non-deep learning model, such as logistic regression, to establish a baseline performance on the emotion classification task.
2. Convolutional Neural Network (CNN): We will develop a CNN-based model that can effectively learn features from the raw audio waveforms or spectrograms, and classify the emotional content.
3. Transformer-based Model: To capture the temporal dynamics of emotional expressions, we will explore the use of transformer-based architectures, such as the Wav2Vec 2.0 model, which has shown promising results for audio-based tasks. We can also utilize a pre-trained model called Whisper, developed by OpenAI, which incorporates 680k hours of multilingual and multitask data from various websites.The Whisper model employs a Seq2seq architecture, utilizing deep learning techniques like transformer encoder and decoder.

BA865 Project Proposal
Group 1
Jishnu Moorthy , Yashvardhan Singh Ranawat

Speech Recognition System Implementation Plan

- Data Preprocessing:
  - Standardize the audio data by converting it to a consistent format and sample rate.
  - Extract relevant acoustic features like Mel-Frequency Cepstral Coefficients (MFCCs) from the audio files using the librosa library.
  - Augment the dataset with techniques like adding noise or shifting pitch to make the model more robust to variations.
- Model Architecture:
  - Use Convolutional Neural Networks (CNNs) to capture spatial patterns in the audio features.
  - Employ Long Short-Term Memory (LSTM) networks to model the temporal dynamics and sequences in the audio data.
- Model Training:
  - Split the dataset into training and validation sets.
  - Train the model on the training data, using techniques like early stopping to prevent overfitting.
  - Monitor the model's performance on the validation set during training.
- Hyperparameter Tuning and Regularization:
  - Tune hyperparameters like the learning rate and batch size to optimize the model's performance.
  - Apply regularization techniques such as dropout and L2 regularization to prevent overfitting and improve generalization.
- Evaluation:
  - Assess the model's performance using metrics like accuracy, precision, recall, and F1 score.
  - Test the model on a separate, unseen dataset to evaluate its ability to generalize to new data.
- Deployment:
  - Develop a RESTful API using frameworks like Flask or FastAPI to enable processing of live audio inputs.
  - Optionally, create a user-friendly frontend interface for interactive speech recognition.