DETERMINING EMOTIONS IN HUMAN SPEECH

Neel Patel & Seong Hu Kim
Department of Biomedical Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
{npatel664,skim3108}@gatech.edu

Kevin Le, Sahil Walke, Andre Hutagaol
Colleges of Computing, Business, and Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
{kevin32,sahil.walke,ahutagaol3}@gatech.edu

ABSTRACT

Understanding affective states is vital for gaining insights into human behavior across various fields. Advanced computing has facilitated emotion recognition, improving human-computer interactions. Multimodal approaches that combine audio and facial expression analysis enhance emotion detection accuracy and context-awareness, with applications in healthcare, education, and user experience, promoting more empathetic communication. This study aims to achieve high accuracy in detecting six emotions—neutral, disgust, fear, sadness, happiness, and anger—from speech. We hypothesize that machine learning models, by utilizing acoustic features and MelSpectrograms, can exceed 0.6 accuracy, thereby enhancing the emotional range and generalizability of speech emotion recognition systems. Combining four emotion recognition datasets—CREMA-D, RAVEDESS, TESS, and SAVEE—into one, we tested this hypothesis. The proposed model, integrating both acoustic parameters and Mel spectrograms with CNN and MLP methods, achieved 0.67 accuracy, while the VGGish-based model reached 0.63. In comparison, the YAMNet model, which used audio embeddings and Mel spectrograms, performed poorly with an accuracy of 0.40. These results emphasize the potential of combining acoustic features and Mel spectrograms with CNN architectures for speech emotion recognition. This project successfully detects six emotions with over 0.6 accuracy, potentially contributing to a standardized emotion detection set and enhancing our understanding of unconscious emotional expression. Further applications include human-computer interaction, mental health diagnostics, and sentiment analysis. Future work should focus on optimizing feature extraction and expanding data sources to improve accuracy.

1 INTRODUCTION

Understanding and analyzing affective states—emotions and moods—is crucial across disciplines, offering insights into human behavior. Psychiatrists use emotional cues to diagnose mood disorders and tailor treatments (Eyben et al., 2016), while psychologists study emotional nuances in vocal expressions to better understand communication. Linguists and phoneticians examine how emotions are conveyed through language. The advent of advanced computing has further advanced affective state analysis, with engineers developing systems that recognize and interpret emotional states. These technologies enhance human-computer interactions, enabling emotion-aware virtual assistants and applications in sentiment analysis to foster more intuitive and effective communication.

Innovative applications of affective state analysis increasingly rely on multimodal approaches to improve emotion detection and interpretation. By integrating audio analysis with facial expression recognition, these systems can achieve higher accuracy and better context-awareness when identifying emotions. This multimodal approach is particularly valuable in real-world settings such as healthcare, education, and user experience design. Additionally, incorporating emotion-detection capabilities into platforms like ChatGPT marks a significant advancement. By analyzing audio inputs to interpret user emotions, these systems could deliver empathetic, personalized responses, greatly enhancing user engagement and satisfaction. The objective of this study is to achieve state-of-the-art accuracy in detecting six emotions from speech. The hypothesis is that, by utilizing acoustic features and MelSpectrograms, machine learning models can surpass accuracy of 0.6 in recognizing six distinct emotions: neutral, disgust, fear, sadness, happiness, and anger. This approach aims to expand the emotional range detected by current speech emotion recognition systems, thereby improving both their performance and generalizability.

2 RELATED WORK

Current speech emotion recognition (SER) systems generally achieve accuracy rates ranging from 0.5 to 0.7, typically focusing on emotions such as anger, sadness, neutrality, and happiness. Various approaches have

been explored to improve these accuracy rates. For instance, a hierarchical binary decision tree applied to 384-dimensional features achieved an accuracy of 0.5846 (Lee et al., 2011). A convolutional neural network (CNN) with an attention mechanism on MelSpectrograms reached an accuracy of 0.6211 (Neumann & Vu, 2017). A combined convolutional-LSTM (Long Short-Term Memory) model achieved an accuracy of 0.5940 (Satt et al., 2017). A multi-input model using deep neural networks (DNN), CNN, and recurrent neural networks (RNN) on diverse features resulted in an accuracy of 0.5830 (Yao et al., 2020). A bidirectional LSTM (BLSTM) with attention applied to 64-dimensional features reached an accuracy of 0.5521 (Li et al., 2021). The integration of MelSpectrogram and Geneva Minimalistic Acoustic Parameter Set (GeMAPS) features with CNN and DNN models, along with the application of the FL function, achieved an accuracy of 0.6149, surpassing or matching existing methods (Toyoshima et al., 2023).

The limited emotional range in current SER systems can largely be traced back to the datasets used for training. One such dataset, the Interactive Emotional Dyadic Motion Capture (IEMOCAP) Database, developed by the University of Southern California, is frequently used in speaker-independent experiments that ensure model generalization by having different actors for the training and testing. The IEMOCAP dataset includes both scripted and improvised speech, focusing primarily on four emotions: anger, sadness, neutrality, and happiness. While this dataset offers valuable resources for training models, its narrow emotional scope limits the ability of SER systems to recognize a wide range of emotions. The lack of more diverse emotional categories means that existing models are often underperforming when tasked with detecting emotions beyond these basic four. This restricted emotional range in the dataset contributes to the limited capabilities of current SER systems and highlights the need for more comprehensive datasets to improve model accuracy and generalization across various emotional expressions. To overcome these limitations, expanding emotional categories and incorporating a greater variety of emotional contexts into datasets would be crucial for enhancing the performance of SER systems.

3 Data Collection and Preparation

3.1 Data Collection

To address the limitations in the emotional range of current SER systems, several alternative datasets have been developed, offering a broader set of emotional categories for training and testing. These datasets, including CREMA-D, RAVEDESS, TESS, and SAVEE, provide valuable resources for enhancing the accuracy and generalizability of emotion recognition models. Each of these datasets contains a diverse range of emotional expressions, contributing to more robust models capable of recognizing a wider variety of emotions in speech.

The CREMA-D, RAVEDESS, TESS, and SAVEE datasets are commonly used resources for training and testing SER models. CREMA-D consists of 91 actors (48 males, 43 females) who performed 12 scripted sentences expressing six emotions: neutral, disgust, fear, sadness, happiness, and anger. It offers a diverse range of emotional expressions across a large number of actors. RAVEDESS, containing 24 actors (12 males, 12 females), includes eight emotions: neutral, calm, disgust, fear, sadness, happiness, anger, and surprise. This dataset's balance of gender and emotional categories makes it a versatile resource for emotion recognition research.

TESS consists of 2 female actors who performed 200 scripted sentences across seven emotions: neutral, disgust, fear, sadness, happiness, anger, and surprise. Despite having fewer actors, it provides extensive samples for each emotion, making it useful for models requiring large datasets. The SAVEE dataset, with 4 male actors performing 15 scripted sentences, covers eight emotions: neutral, common, disgust, fear, sadness, happiness, anger, and surprise. Although smaller, SAVEE is valuable for research focused on male speech and specific emotional categories. All four datasets contain six or more emotions, offering a rich variety of emotional expressions crucial for developing more accurate and generalized emotion recognition systems. All datasets are available on Kaggle.

3.2 Exploratory Data Analysis

The analysis of four emotional speech datasets—CREMA-D, TESS, SAVEE, and RAVDESS—reveals notable variations in emotion category distribution as shown in Figure 1. CREMA-D demonstrates a balanced representation across six emotions (angry, disgust, fear, happy, neutral, and sad), with sample counts ranging between 1087 and 1271 per category. Similarly, TESS exhibits uniformity, with 400 samples allocated to each emotion. Conversely, SAVEE presents an imbalance, with the "neutral" emotion dominating (120 samples) and all other categories equally underrepresented at 60 samples each. RAVDESS shows moderate variation, with "neutral" having the highest count (288 samples), while the remaining six emotions are evenly distributed at 192 samples each. These differences in dataset composition highlight potential implications for emotion recognition models, particularly in ensuring robust performance across varied distributions.

Figure 1: Speech Emotion Recognition Datasets

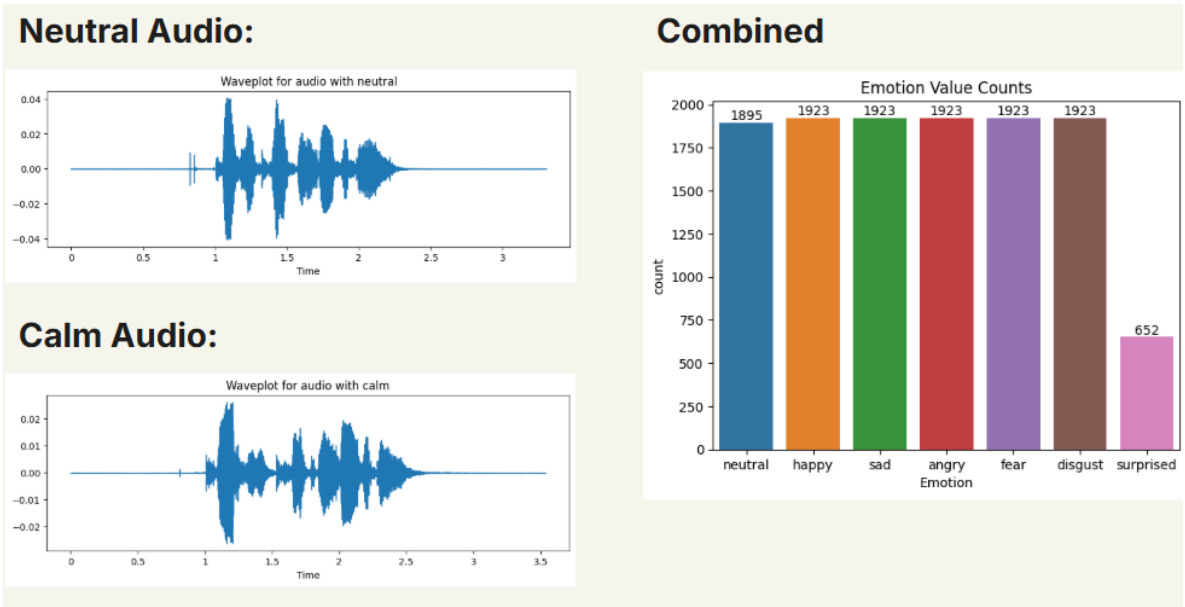

Figure 2: Similarity of Neutral and Calm Audio and the Combined Dataset

Initially, the ”surprised” category was included in the analysis, augmented twice to yield 1956 samples from an original 652. Of these, 1304 augmented samples were used for training and validation, while 652 unaugmented samples were reserved for testing. However, this approach led to data leakage, compromising validation and test accuracy. To resolve this, the “surprised” category was removed, leaving six emotions for classification. Furthermore, to address overlap in emotional categories, “neutral” and “calm” labels from RAVDESS, as well as “neutral” and “common” labels from SAVEE, were consolidated into a single “neutral” category. This decision

was based on the significant similarity between these labels, which posed challenges for differentiation as shown in Figure 2. Combining them under a unified label ensured a more coherent dataset structure and minimized ambiguity during classification.

3.3 PREPARATION

For the custom model, all audio files were resampled to 24414 Hz using the `torchaudio.transforms.Resample` library. The low-pass filter width was 16 and the resampling method used was the `sinc_interp_hann` method. After resampling, audio files underwent data augmentations.

3.4 AUGMENTATIONS

We used data augmentation for improving model performance and generalization. Augmentations such as adding white noise, time-stretching, pitch shifting, polarity inversion, and gain adjustments help simulate diverse real-world conditions, like noise, speaking style variations, and recording inconsistencies. These techniques introduce variability into the training data, allowing models to learn robust features and handle unseen scenarios effectively. Audio augmentations were applied only to the training data set to avoid data leakage.

3.4.1 WHITE NOISE

We added white noise by mixing a low-amplitude random signal into the audio samples. This technique can simulate environmental noise, improving model robustness to noisy conditions. It's subtle enough to preserve the original speech's content while diversifying the dataset.

3.4.2 STRETCH

We added stretch to our audio samples by speeding up or slowing down speech, mimicking variations in speaking rate across individuals. This augmentation is especially useful for enhancing a model's ability to handle speech rate differences.

3.4.3 PITCH

Pitch augmentation was used by adjusting the fundamental frequency of the audio without affecting its duration. Raising or lowering the pitch can simulate variations in speaker tone or emotional expression, making the model more resilient to pitch diversity.

3.4.4 POLARITY

We used Polarity inversion as augmentation by flipping the audio waveform vertically, reversing its phase. While it doesn't affect the perceived sound for humans, it can introduce diversity in how a model interprets the signal, improving generalization.

3.4.5 GAIN

Lastly, we used gain as the augmentation where we modified the signal's amplitude, effectively making the audio louder or quieter. This augmentation helps simulate recording conditions with varying microphone sensitivities or distances, improving the model's robustness to volume changes.

3.5 FEATURE EXTRACTION

Following augmentations, feature extraction was performed. These features were Mel Spectrogram, harmonic-noise-ratio (HNR), zero crossing rate (ZCR), root mean square energy (RMS), and Mel-Frequency Cepstral Coefficients (MFCC).

3.5.1 MEL SPECTROGRAM

Mel Spectrograms are well known for their use in audio classification as they are essentially images for audio data, capturing frequency relationships over time. Using the `torchaudio.transforms.MelSpectrogram` library, Mel Spectrograms were extracted using 1024 frames with a hop length of 128. The number of mel bands used was 128 leading to a Mel Spectrogram of size $(128, 573)$. Following the Mel Spectrogram, a decibel transformation was applied using the `torchaudio.functional.amplitude_to_DB` library, with a multiplier of 10 and lower clamp of 10^{-6}. These Mel Spectrograms would later be passed into the CNN sub model as described in the Methodology section

3.5.2 HARMONIC-NOISE-RATIO

HNR is a single positive number used as a measure of vocal roughness for audio signals. The libraries used were the `librosa.effects.harmonic` and `librosa.effects.percussive`. The ratio of the harmonic component to the percussive component is the HNR.

3.5.3 ZERO CROSSING RATE

Zero Crossing Rate (ZCR) is a widely used feature in signal processing, especially in speech and audio analysis. It represents the rate at which the signal changes its sign (crosses the zero amplitude level). For Speech Emotion Recognition (SER), ZCR is often used as a feature to analyze the speech signal's energy and frequency components.

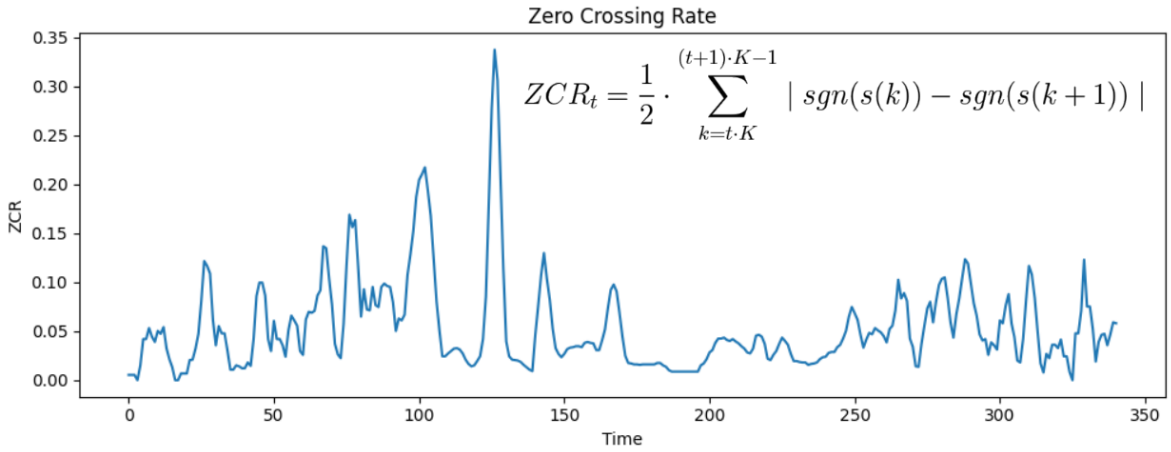

Figure 3: Zero Crossing Rate

High ZCR usually Indicates more frequent changes in the signal, often associated with higher-pitched or fricative sounds (e.g., anger, excitement) whereas Low ZCR indicates smoother transitions, often found in calm or sad emotional states. Moreover, voiced speech (vowels, with vocal cord vibration) typically has lower ZCR while unvoiced speech (consonants like /s/ or /f/, with no vocal cord vibration) has higher ZCR. For our project we use a total of 144 ZCR features.(one from each time frame)

3.5.4 ROOT MEAN SQUARE ENERGY

Root Mean Square (RMS) is another fundamental feature in speech and audio processing. It quantifies the energy or amplitude of a signal and can provide insights into the emotional intensity of speech. For our project we use a total of 144 RMS features.(one from each time frame)

High RMS is often associated with loud speech, often found in emotions like anger, happiness, or excitement while low RMS is associated with softer speech, often found in emotions like sadness or calmness. RMS reflects the average power of the signal in a specific frame, making it a direct indicator of energy, which varies significantly between different emotional states.

3.5.5 MEL-FREQUENCY CEPSTRAL COEFFICIENTS

Mel-Frequency Cepstral Coefficients (MFCCs) are one of the most widely used features in speech processing tasks. They provide a compact representation of the speech spectrum, capturing information related to the timbre and frequency distribution of the speech signal. MFCCs are especially valuable because they model how humans perceive sound, emphasizing frequencies in the mel scale.

MFCCs emphasize lower frequencies, capturing the speech signal's harmonic and formant structure, which are important for identifying emotions. MFCCs reduce dimensionality while retaining critical information, making them suitable for machine learning models. Variations in pitch, intonation, and articulation across different emotions are reflected in MFCC features. MFCCs are also less sensitive to noise compared to raw spectral features. For our case, we are using a total of 13 MFCCs for each audio sample.

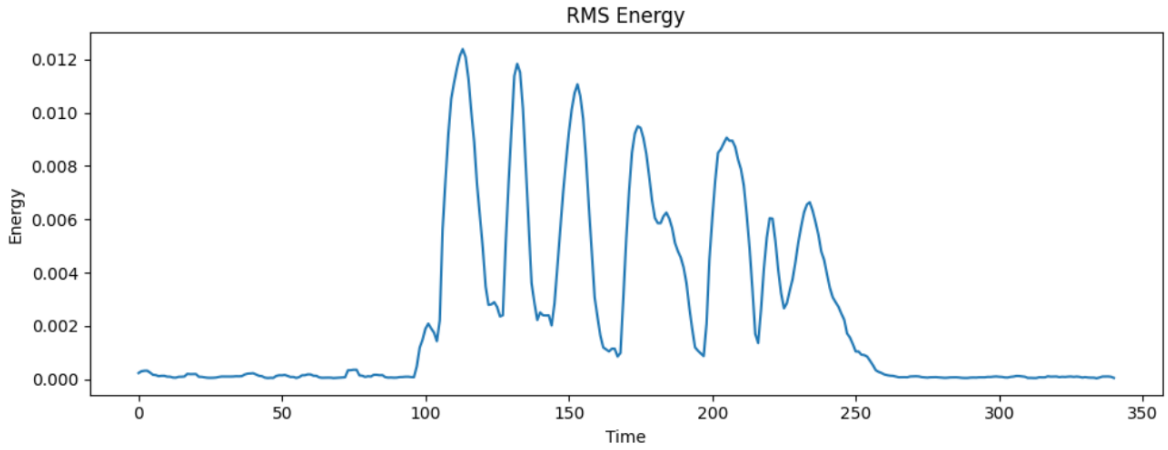

Figure 4: Root Mean Square Energy

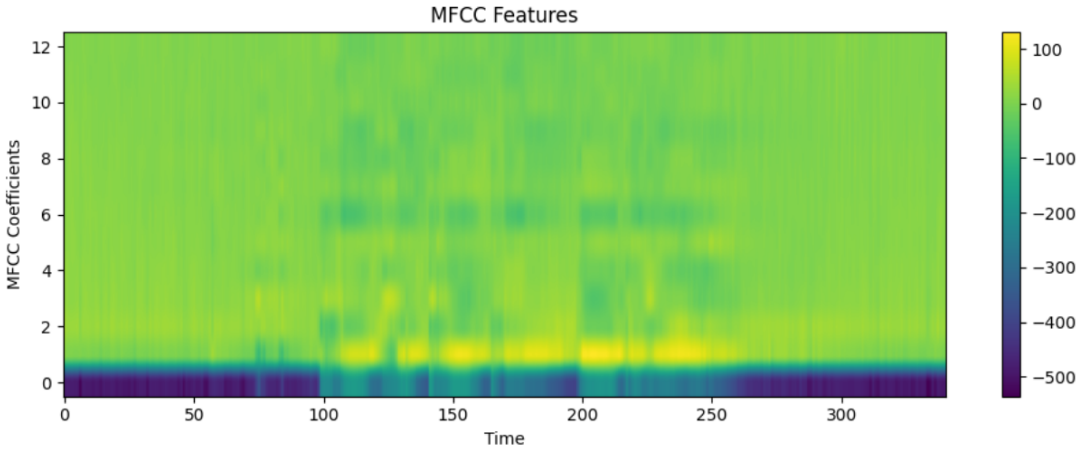

Figure 5: Mel-Frequency Cepstral Coefficients

3.6 Standardization

Prior to standardization, the features MFCC, ZCR, HNR, and RMS were concatenated to create a 302 long feature vector per audio file. This concatenated input will be referred to as the feature vector. The Mel Spectrograms of size $(128, 573)$ were kept as a separate input. Mel Spectrogram standardization consisted of flattening all Mel Spectrograms in the training set and computing the global mean and standard deviation. The mean and standard deviation of the training set were then applied to the validation and testing set.

For the feature vector, the mean and standard deviation were computed dimension-wise from the training set and applied to the validation and testing set.

4 Methodology

4.1 Custom Model

4.1.1 Overview

We propose a hybrid deep learning architecture combining convolutional neural networks (CNN) and multi-layer perceptrons (MLP) for audio emotion classification. The model processes two complementary input types: Mel spectrograms and feature vectors derived from audio characteristics such as ZCR, HNR, MFCC, and RMS. The CNN captures spatial and temporal patterns from the Mel spectrograms, while the MLP extracts detailed insights from the supplementary audio features. Outputs from the two networks are fused together to create a

richer and more comprehensive representation in an attempt for more accurate classification of emotional states in the audio rather than using Mel Spectrograms alone.

4.1.2 CONVOLUTIONAL NEURAL NETWORK (CNN)

The CNN submodel processes the Mel Spectrograms (input dimension: 128×573) through a series of eight convolutional blocks. Each block has:

- Two convolutional layers with 3×3 kernels and ReLU activation
- Batch normalization for stable training
- Max Pooling layer of kernel size 2 for down-sampling and efficient training
- A dropout layer ($p = 0.3$) for regularization

The convolutional layers progressively increase in the number of filters ($64 \rightarrow 128 \rightarrow 256 \rightarrow 256 \rightarrow 512 \rightarrow 512 \rightarrow 512$). The output is then flattened into a vector of size 64 for fusion with the MLP output.

4.1.3 MULTI-LAYER PERCEPTRON (MLP)

The MLP processes an input feature vector of size 302 with an output of 64. It consists of four fully connected layers with ReLU activations and batch normalization:

- Layer sizes: $302 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 64$
- Dropout ($p = 0.3$) is applied to each layer to reduce overfitting

4.1.4 FUSION AND CLASSIFICATION

The outputs of the CNN (64) and MLP (64) are concatenated into a 128-dimensional vector. This combined representation passes through two fully connected layers ($128 \rightarrow 64 \rightarrow 6$) with batch normalization and ReLU activation. The final layer outputs logits for the 6 class classification task. The figure below shows the model architecture.

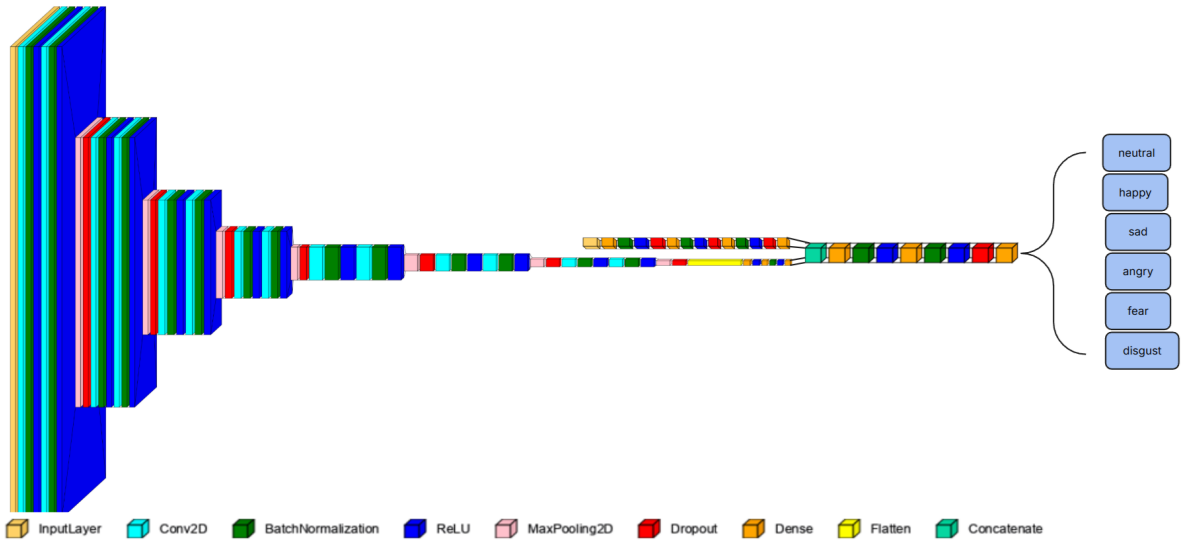

Figure 6: Emotion Classification Hybrid Model Architecture

4.1.5 TRAINING AND EVALUATION

The model is trained using cross-entropy loss. We utilized the Adam Optimizer ($\text{lr} = 10^{-3}$) with a ReduceLROnPlateau learning rate scheduler to adjust learning rate based on the validation loss. Evaluation metrics include accuracy, precision, recall, and F1-score calculated after each epoch. Additionally, these metrics were calculated for each emotion to assess model performance across each emotion.

4.2 TRANSFER LEARNING

In the previous section, a hybrid model, a concatenation of a convolutional neural network and a multilayer perceptron, were used as one of the methodologies. Another method proposed to predict emotions in audio is transfer learning.

Transfer learning is a technique that takes a pre-trained model for one task and retrains it to perform another task. Fine-tuning takes a pre-trained model and retrains it on a new dataset. In contrast, transfer learning takes the features extracted from the pre-trained model and uses them as input for another model. Several advantages transfer learning offers compared to creating a custom model or using fine-tuning, including saving training and computational time; not needing a lot of data; and increasing performance on the new task. While several advantages come with transfer learning, some disadvantages come with it. One disadvantage of transfer learning is that it is dependent on the quality of data used for the pre-trained model. If the pre-trained model was trained on bad data, the output of the pre-trained model will not be good inputs for the new model. Another disadvantage with transfer learning is the performance is also dependent on how similar the new and old tasks are. Transfer learning will be perform poorly if the new task is significantly different from the pre-trained model's task.

The models that transfer learning will be applied on are VGGish and YAMNet, both pre-train models that offer features that are relevant for audio emotion classification.

4.2.1 VGGISH

VGGish is an audio classifier developed by Google, trained on a large YouTube dataset using the VGG architecture. It converts audio waveforms into 128-dimensional embeddings representing audio segments.

4.2.2 VGGISH - TRANSFER PIPELINE

VGGish was trained using audio features computed through the following steps:

- All audio is resampled to 16 kHz mono.
- A spectrogram is generated using the magnitudes of the Short-Time Fourier Transform (STFT) with a window size of 25 ms, a window hop of 10 ms, and a periodic Hann window.
- The spectrogram is mapped to 64 mel bins covering the frequency range of 125–7500 Hz to create a mel spectrogram.
- A stabilized log mel spectrogram is obtained by applying the transformation $\log(\text{mel-spectrum}+0.01)$, where the offset prevents taking the logarithm of zero.
- These features are framed into non-overlapping examples, each lasting 0.96 seconds and encompassing 64 mel bands and 96 frames, with each frame representing 10 ms of audio.

After obtaining the 128-dimensional embeddings from the VGGish model, the output is further trained using a CNN model to predict the probability of an audio sample belonging to a specific class. The flowchart of the transfer learning process using VGGish is shown in Figure 7.

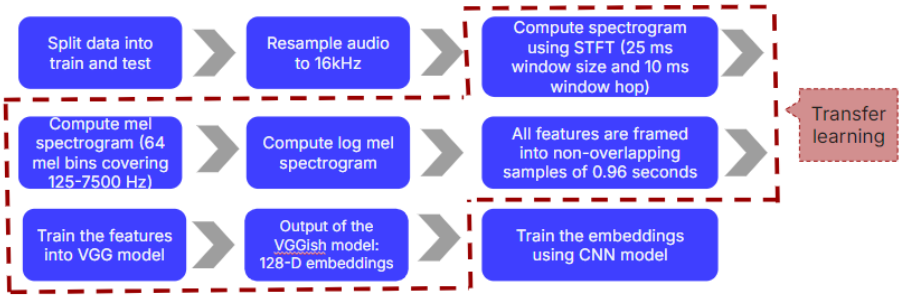

Figure 7: Flowchart of Transfer Learning Using VGGish

4.2.3 VGGISH - CONVOLUTIONAL NEURAL NETWORK (CNN)

We proposed a CNN model to train embeddings derived from the VGGish model. This CNN processes 1D sequential data with an input shape of (28, 128). It consists of three convolutional blocks, each featuring

a Conv1D layer (with 512, 256, and 128 filters, respectively), ReLU activation, batch normalization, max-pooling, and dropout for regularization. A GlobalAveragePooling1D layer reduces the feature maps to a fixed-size vector, which is then passed through two fully connected layers with 128 and 64 units, ReLU activation, L2 regularization, batch normalization, and dropout. Finally, the output layer applies a softmax activation to classify the inputs into the specified number of classes. The model is trained using the Adam optimizer and categorical cross-entropy loss for multi-class classification. The details of the layers of the CNN model can be shown in Table 1 and Figure 8.

Layer (Type)	**Parameters**	**Output Shape**
Conv1D	Filters: 32, Kernel Size: 3, ReLu Activation	(28, 512)
BatchNormalization		(28, 512)
MaxPooling1D	Pool Size: 2, Stride: 2	(14, 512)
Conv1D	Filters: 256, Kernel Size: 3, Stride: 1, ReLu Activation	(14, 256)
BatchNormalization		(14, 256)
MaxPooling1D	Pool Size: 2, Stride: 2	(7, 256)
Dropout	Rate: 0.3	(7, 256)
Conv1D	Filters: 128, Kernel Size: 3, Stride: 1, ReLu Activation	(7, 128)
BatchNormalization		(7, 128)
MaxPooling1D	Pool Size: 2, Stride: 2	(4, 128)
Dropout	Rate: 0.3	(4, 128)
Global Average Pooling		(128)
Dense	Units: 128, ReLu Activation, L2 regularizer (1e-4)	(128)
BatchNormalization		(128)
Dropout	Rate: 0.4	(128)
Dense	Units: 64, ReLu Activation	(64)
BatchNormalization		(64)
Dropout	Rate: 0.3	(64)
Output Layer		(6)

Table 1: Architecture sumary using VGGish embeddings.

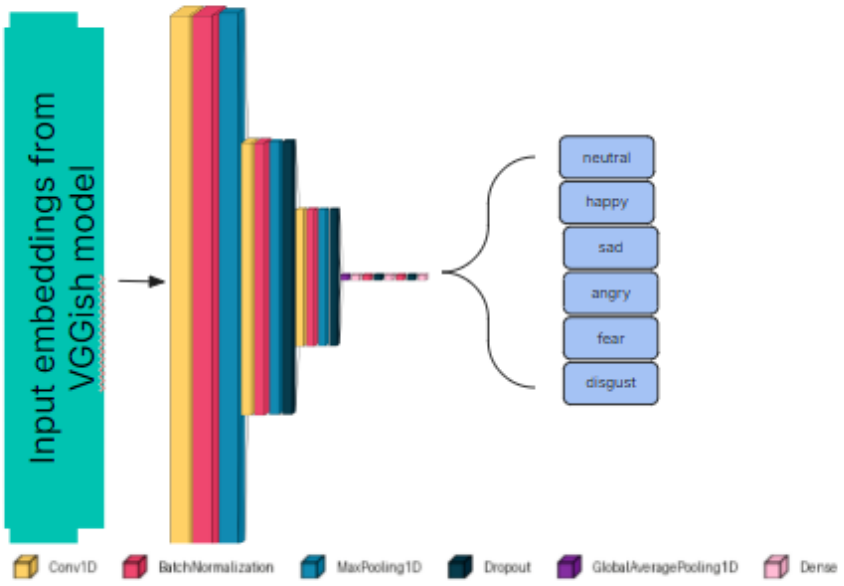

Figure 8: VGGish Transfer Learning - Emotion Classification Model Architecture

4.2.4 VGGISH - TRAINING AND EVALUATION

This model was trained using crosss-entropy loss with the Adam Optimizer (lr $= 10^{-6}$). This model has the same evaluation metrics as the previous two models: accuracy, precision, recall, and F1-score. It also calculated the metrics for each emotion to assess model performance across each emotion.

4.2.5 YAMNET

YAMNet is an audio event classifier created by Google. This model was trained on the AudioSet dataset, which contains 632 classes and 2,084,320 10-second sound clips (˜5,800 hours) from Youtube videos.

4.2.6 YAMNET - TRANSFER PIPELINE

The training pipeline for transfer learning with YAMNet is identical to that of VGGish. We split the un-augmented data into training and testing sets. The samples from training and testing set were resampled to 16,000 Hz to match the input requirement of YAMNet. It goes through the same process as the VGGish with some key differences. The embedding size from YAMNet is 1,024 dimensions rather than 128. In addition to embeddings as a feature outputted from YAMNet, the model also offers log Mel Spectrograms and predicted scores. The predicted scores is just a probability of an audio belonging to a class. The predicted score will not be used in transfer learning since the speech was the highest scoring for all files in the dataset.

4.2.7 YAMNET - CONVOLUTIONAL NEURAL NETWORKS (CNN)

Our proposed model using transfer learning with YAMNet, will process embeddings and log Mel spectrograms with separate CNNs. Two different sub-models are necessary because embeddings are a 1D feature while log Mel spectrograms are a 2D feature. The dimensions of the embeddings and log Mel spectrograms extracted from the YAMNet respectively are (6, 1,024) and (336, 64). For embeddings, there are 6 audio segments (0.96 seconds for each segment) and 1,024 features. For log Mel spectrograms, there are 336 frames and 64 Mel filter banks. To make the log Mel spectrograms two-dimensional, another dimension is added to represent the number of channels. Since every audio is mono audio, one would be added to make the new log Mel spectrogram dimension (336, 64, 1). The dimensions of the embeddings and log Mel spectrograms will be the input for their respective model. The embedding submodel will go through two rounds of 1D convolutional, batch normalization, and dropout. Log Mel spectrograms, on the other hand, will go through two rounds of 2D convolutional, batch normalization, dropout, ReLU, and MaxPooling 2D. Both outputs are flattened afterwards. The tables below show the layers used for each sub-model.

Layer (type)	**Parameters**	**Output Shape**
1D Convolutional	Filters: 64, Kernel Size: 3, ReLU Activation	(4, 64)
Batch Normalization		(4, 64)
Dropout	p = 0.2	(4, 64)
1D Convolutional	Filters: 32, Kernel Size: 3, ReLU Activation	(2, 32)
Batch Normalization		(2, 32)
Dropout	p = 0.2	(2, 32)
Flatten		(64)

Table 2: Embedding Submodel Architecture Summary

Layer (type)	**Parameters**	**Output Shape**
2D Convolutional	Filters: 32, Kernel: (3,3), Padding: Same	(336, 64, 32)
Batch Normalization		(336, 64, 32)
Dropout	p = 0.2	(336, 64, 32)
Rectified Linear Unit (ReLU)		(336, 64, 32)
MaxPooling2D	Kernel: (2,2)	(168, 32, 32)
2D Convolutional	Filters: 64, Kernel: (3,3), Padding: Same	(168, 32, 64)
Batch Normalization		(168, 32, 64)
Dropout	p = 0.2	(168, 32, 64)
Rectified Linear Unit (ReLU)		(168, 32, 64)
MaxPooling2D	Kernel: (2,2)	(84, 16, 64)
Flatten	0	(86016)

Table 3: Log Melspectrogram Submodel Architecture Summary

4.2.8 YAMNET - FUSION AND CLASSIFICATION

The outputs from both CNN sub-models are concatenated into an 86,080 dimensional vector, and the vector goes through two rounds of dense, batch normalization, and dropout. Like the two previous models, the final layer will output logits for the 6 class classification tasks. The table and figure below show the layers used in the final part of the model and its architecture.

Layer (type)	Parameters	Output Shape
Concatenate		(86080)
Dense	Neurons: 256, ReLU Activation	(256)
Batch Normalization		(256)
Dropout	p = 0.6	(256)
Dense	Neurons: 128, ReLU Activation	(128)
Batch Normalization		(128)
Dropout	p = 0.6	(128)
Dense	Neurons: 6, Softmax Activation	(6)

Table 4: Hybrid Model Architecture Summary

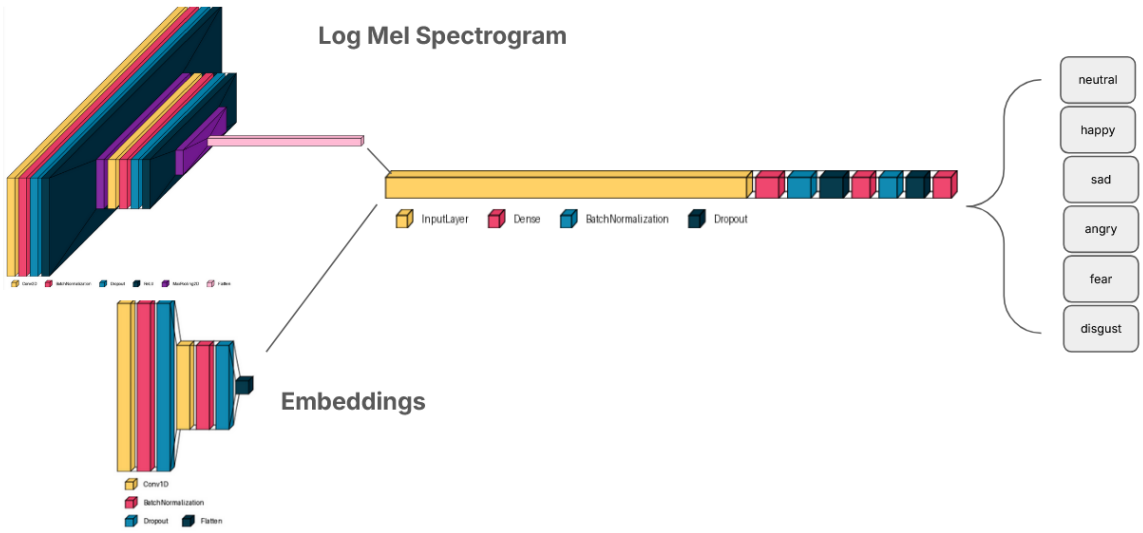

Figure 9: YAMNet Transfer Learning - Emotion Classification Hybrid Model Architecture

4.2.9 YAMNET - TRAINING AND EVALUATION

This model was trained using crosss-entropy loss with the Adam Optimizer (lr $= 10^{-6}$). This model has the same evaluation metrics as the previous two models: accuracy, precision, recall, and F1-score. It also calculated the metrics for each emotion to assess model performance across each emotion.

5 EXPERIMENTAL RESULTS

5.1 CUSTOM MODEL RESULTS

5.1.1 WITH AUGMENTATIONS NO STANDARDIZATION

We first evaluated the performance our model with just the augmentations and without standardization. The model performs poorly achieving only 51% accuracy (random chance of guessing an emotion is 16.67%). After careful inspection, the combination of augmentations were drastically changing the audio files when compared to the original leading to difficult to classify audio emotions. The tables below summarize the metrics from this experimental group.

Evaluation Metric	Result
Test Loss	1.4467
Test Accuracy	0.5123
Precision	0.5294
Recall	0.5123
F1-Score	0.5105

Table 5: Custom Model with augmentations no standardization - Overall Evaluation Metric

	Precision	Recall	F1-Score	Support
Neutral	0.66	0.57	0.61	652
Happy	0.43	0.39	0.41	652
Sad	0.63	0.36	0.46	652
Angry	0.43	0.60	0.50	652
Fear	0.48	0.58	0.52	652
Disgust	0.54	0.57	0.56	652
Accuracy			0.51	3912
Macro Avg	0.53	0.51	0.51	3912
Weighted Avg	0.53	0.51	0.51	3912

Table 6: Custom Model with augmentations no standardization - Classification Report

The confusion matrix below shows that the model is having a hard time, in general, with the disgust category. Furthermore, the model confuses disgust with neutral and angry with happy. The model best performs with the neutral category.

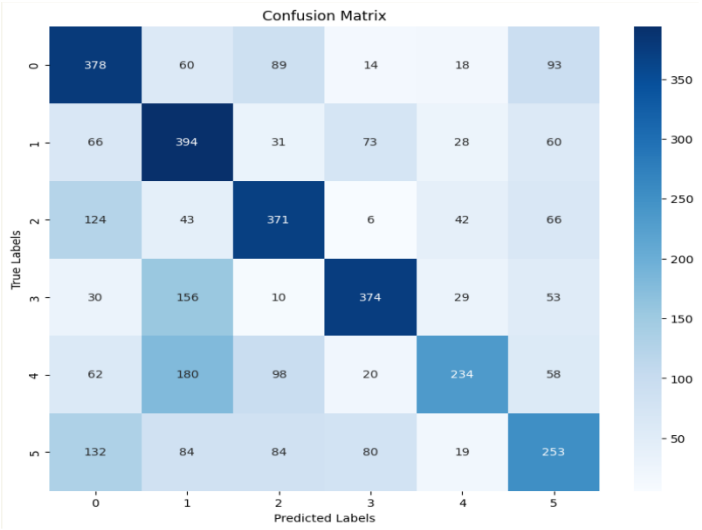

Figure 10: Confusion matrix for model with augmentations without standardization

The figure below shows the training and validation loss captured by tensorboard. Because of dropout and batchnorm regularization, the model is not overfitting.

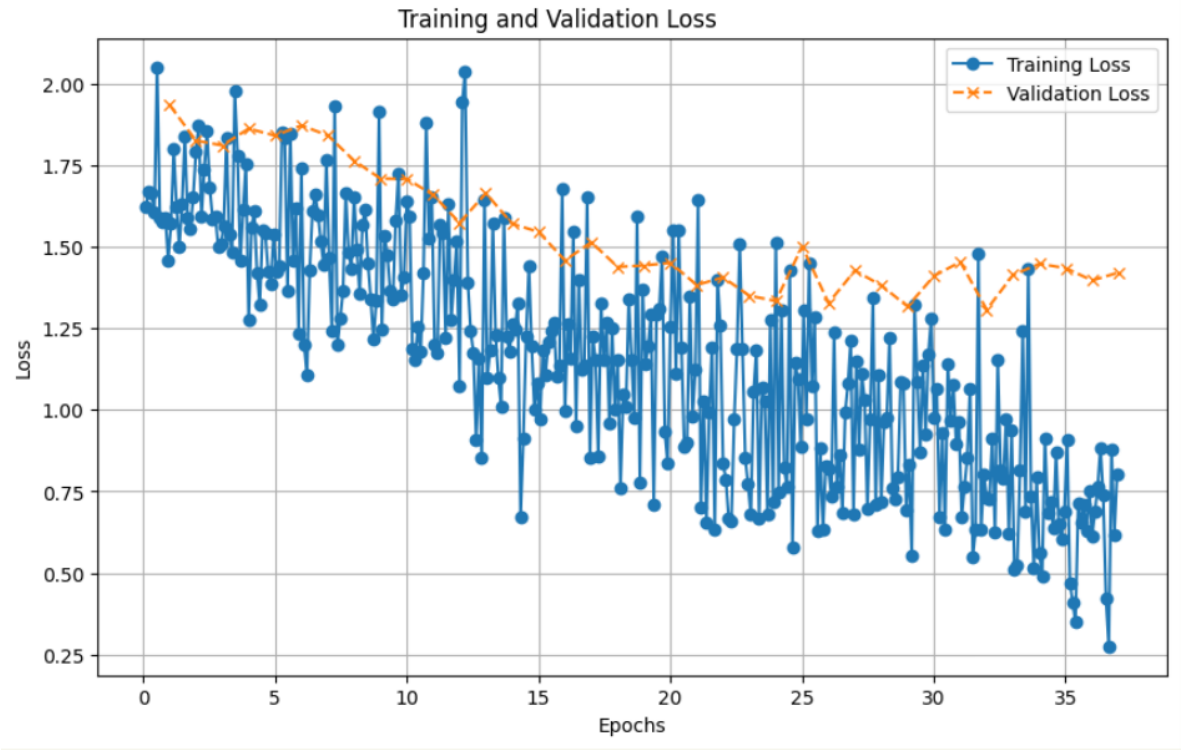

Figure 11: Training and Validation loss for model with augmentations without standardization

5.1.2 No Augmentations and No Standardization

From the previous section's results, we investigated training the model on un-augmented training data. The model substantially improved across all metrics as shown in the bottom two tables.

Evaluation Metric	Result
Test Loss	0.9259
Test Accuracy	0.6659
Precision	0.7011
Recall	0.6661
F1-Score	0.6678

Table 7: Custom Model no augmentations and no standardization - Overall Evaluation Metric

	Precision	Recall	F1-Score	Support
Neutral	0.77	0.77	0.77	289
Happy	0.48	0.84	0.61	289
Sad	0.71	0.56	0.63	289
Angry	0.81	0.52	0.63	289
Fear	0.77	0.71	0.74	284
Disgust	0.67	0.60	0.63	288
Accuracy			0.67	3912
Macro Avg	0.70	0.67	0.67	1727
Weighted Avg	0.70	0.67	0.67	1727

Table 8: Custom Model no augmentations and no standardization - Classification Report

Additionally, the confusion matrix shows that model still does well with the neutral emotion. However, the disgust category is still hard for the model to distinguish.

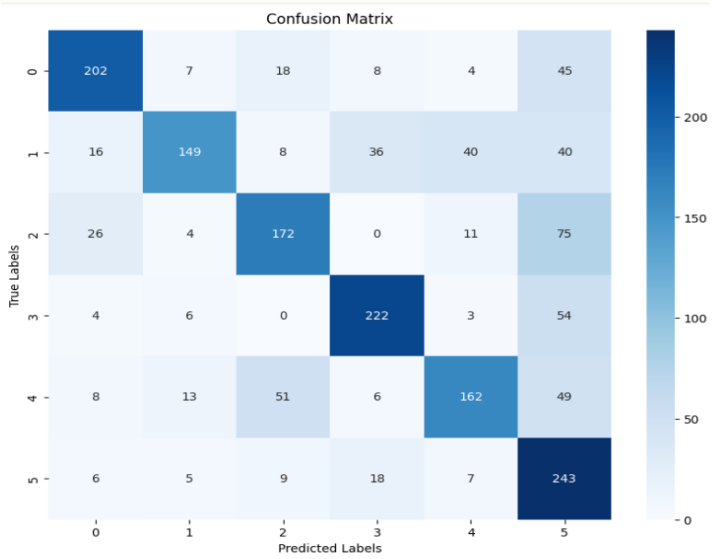

Figure 12: Confusion matrix for model no augmentations and no standardization

Furthermore, the training and validation loss curves show little to no overfitting indicating good regularization.

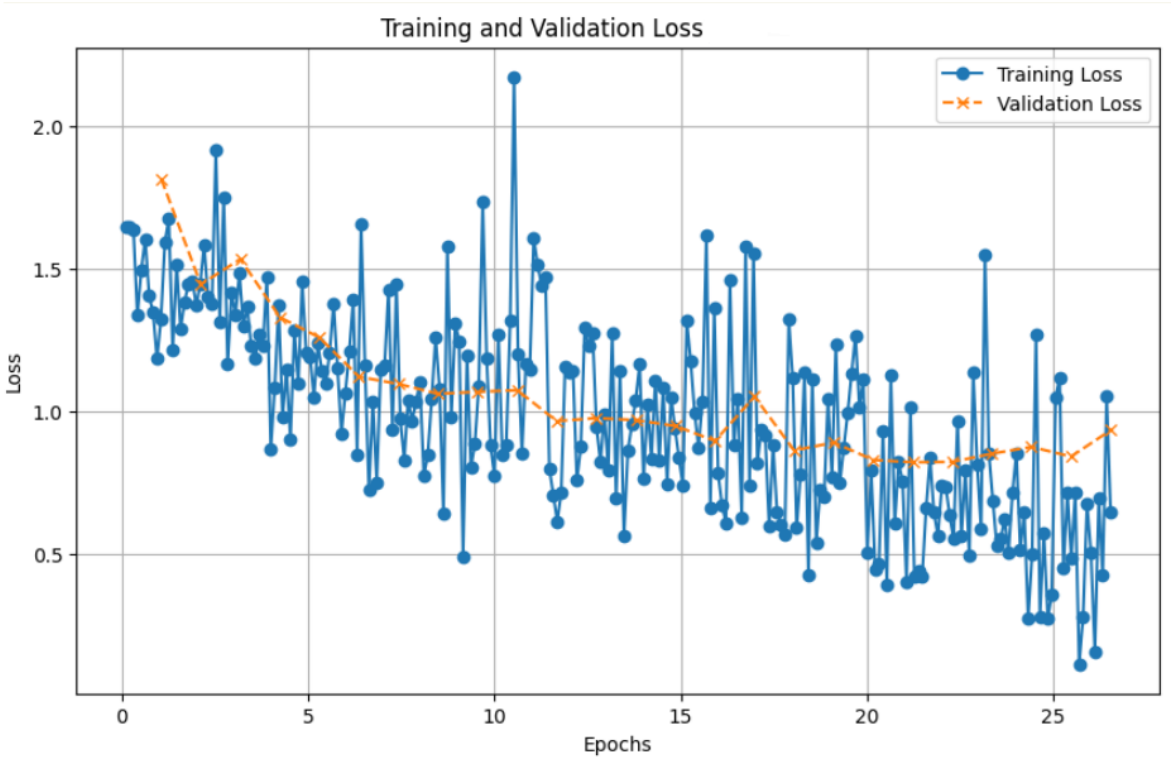

Figure 13: Training and Validation loss for model with no augmentations and no standardization

5.1.3 No Augmentations with Standardization

Finally, we investigated the effects of standardization on the un-augmented data. From the tables below, we do not see a significant difference compared to the no augmentations no standardization experimental group.

Evaluation Metric	Result
Test Loss	0.9129
Test Accuracy	0.6676
Precision	0.6876
Recall	0.6680
F1-Score	0.6638

Table 9: Custom Model no augmentations with standardization - Overall Evaluation Metric

	Precision	Recall	F1-Score	Support
Neutral	0.75	0.82	0.78	289
Happy	0.79	0.49	0.60	289
Sad	0.75	0.52	0.61	289
Angry	0.69	0.69	0.69	289
Fear	0.58	0.81	0.68	284
Disgust	0.57	0.69	0.62	288
Accuracy			0.67	3912
Macro Avg	0.69	0.67	0.66	1727
Weighted Avg	0.69	0.67	0.66	1727

Table 10: Custom Model no augmentations with standardization - Classification Report

With standardization of the Mel Spectrograms and feature vector, the model simply shifts its focus on different emotions. For example, the precision of the happy emotion significantly increases to 0.79 but the recall drops significantly to 0.49 compared to the no augmentation no standardization group. The confusion matrix provides more insight into this precision/recall balance.

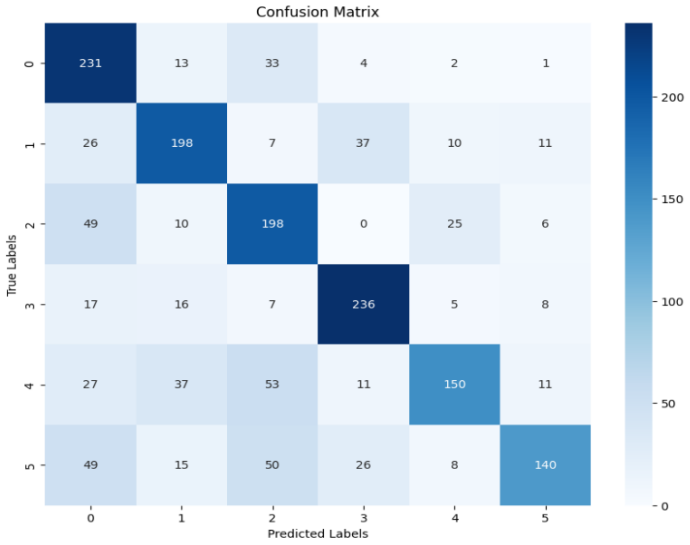

Figure 14: Confusion matrix for model no augmentations with standardization

Likewise, the training and validation loss curves show little to no overfitting indicating good regularization.

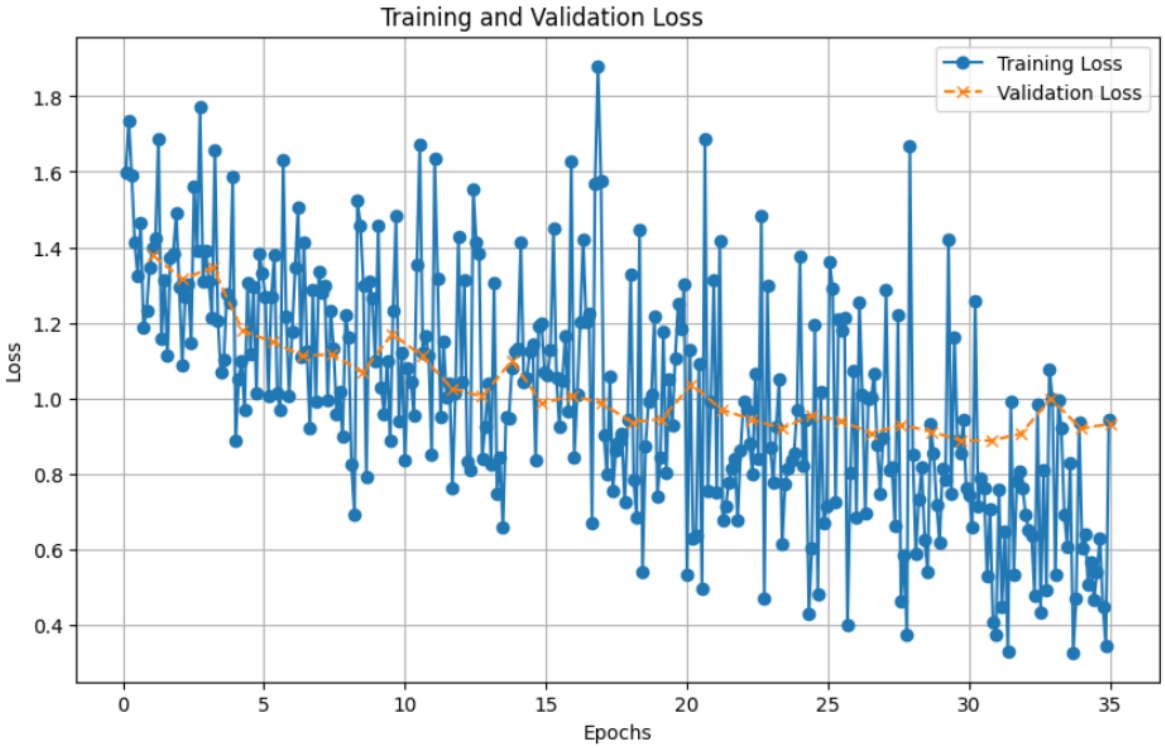

Figure 15: Training and Validation loss for model with no augmentations with standardization

5.2 VGGish - Transfer Learning Results

Now let's take a look at the evaluation using the VGGish model. The overall accuracy is 63%, which is lower than the accuracy of the model discussed in 5.1.2. Table 11 shows the overall evaluation metrics of the model, and Table 12 provides the evaluation metrics for each class.

Evaluation Metric	**Result**
Test Loss	1.1723
Test Accuracy	0.6269
Precision	0.6283
Recall	0.6269
F1-Score	0.6266

Table 11: VGGish Transfer Learning with no augmentations - Overall Evaluation Metric

	Precision	**Recall**	**F1-Score**	**Support**
Neutral	0.65	0.71	0.68	374
Happy	0.54	0.53	0.54	378
Sad	0.59	0.66	0.62	381
Angry	0.77	0.73	0.75	384
Fear	0.58	0.55	0.57	387
Disgust	0.65	0.58	0.61	401
Accuracy			0.63	2305
Macro Avg	0.63	0.63	0.63	2305
Weighted Avg	0.63	0.63	0.63	2305

Table 12: VGG Transfer Learning with no augmentations - Classification Report

From Table 12 we can see that the model performs very well in predicting angry and neutral emotions, since the recall and precision of these emotions are the highest (above 65%). However, the model does not perform well in predicting other emotions, as it misclassifies them with other emotions as well. The details of the model's prediction for each emotion can be shown in the Figure 16.

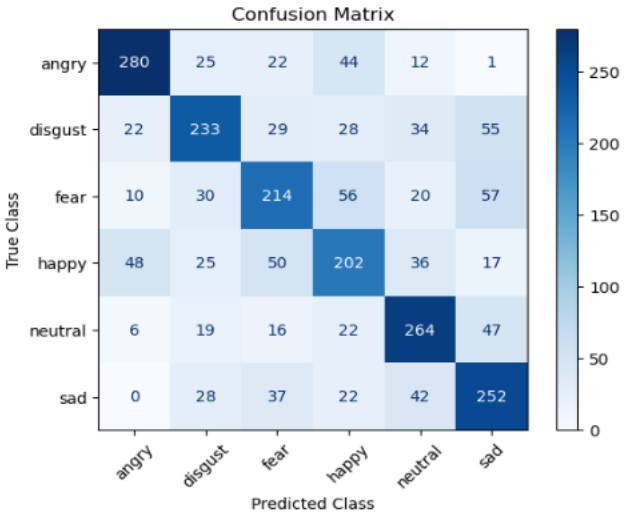

Figure 16: VGGish Transfer Learning with no augmentations - Confusion Matrix

Figure 17 shows the training and testing loss chart of the model. Note that the training loss keeps decreasing towards 0 as the epoch increases, while the testing loss plateaus after epoch 20. This indicates that the model is not overfitting.

Figure 17: VGGish Transfer Learning with no augmentations - Training and Validation Loss Results

5.3 YAMNET - TRANSFER LEARNING RESULTS

The model performs the worst out of all the models with an accuracy of 40%. Although this model performed worse compared to the other models, it still still did better than random choice. The tables below show the evaluation metrics for the model overall and for each emotion.

Evaluation Metric	Result
Test Loss	1.5208
Test Accuracy	0.4018
Precision	0.5632
Recall	0.4018
F1-Score	0.3840

Table 13: YAMNet Transfer Learning with no augmentations - Overall Evaluation Metric

The table below shows the evaluation metrics for each of the emotions. For each emotion:

	Precision	Recall	F1-Score	Support
Neutral	0.71	0.38	0.49	378
Happy	0.49	0.31	0.38	375
Sad	0.30	0.51	0.38	357
Angry	0.59	0.23	0.34	400
Fear	0.94	0.15	0.26	396
Disgust	0.32	0.84	0.46	396
Accuracy			0.40	2302
Macro Avg	0.56	0.40	0.38	2302
Weighted Avg	0.56	0.40	0.38	2302

Table 14: YAMNet Transfer Learning with no augmentations - Classification Report

The figure below shows the training and validation loss decreasing as the number of epochs increases. The training loss continues to decrease while the validation loss plateaus at around the third epoch. The model is not overfitting since the both losses are decreasing.

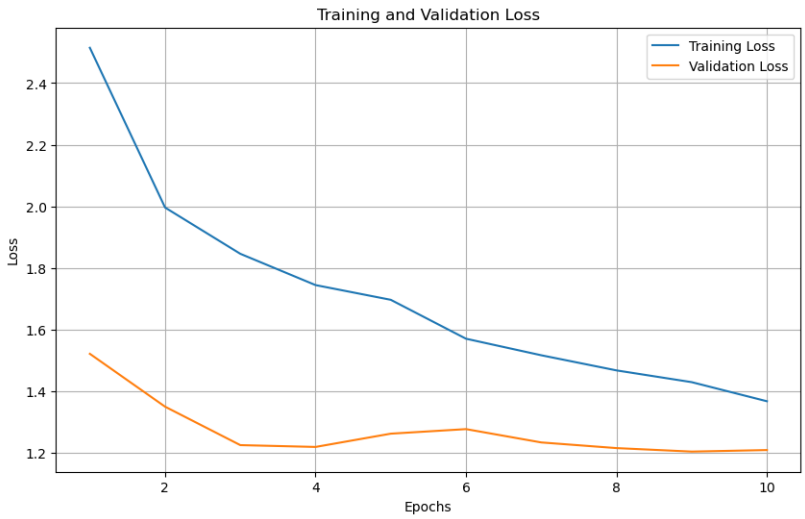

Figure 18: YAMNet Transfer Learning with no augmentations - Training and Validation Loss Results

In the confusion matrix, shown below, the model commonly misclassifies neutral, happy, and angry as sad and disgust. For sad, the model commonly misclassifies it as disgust. Fear is also often misclassified as disgust, while disgust is commonly misclassified as sad.

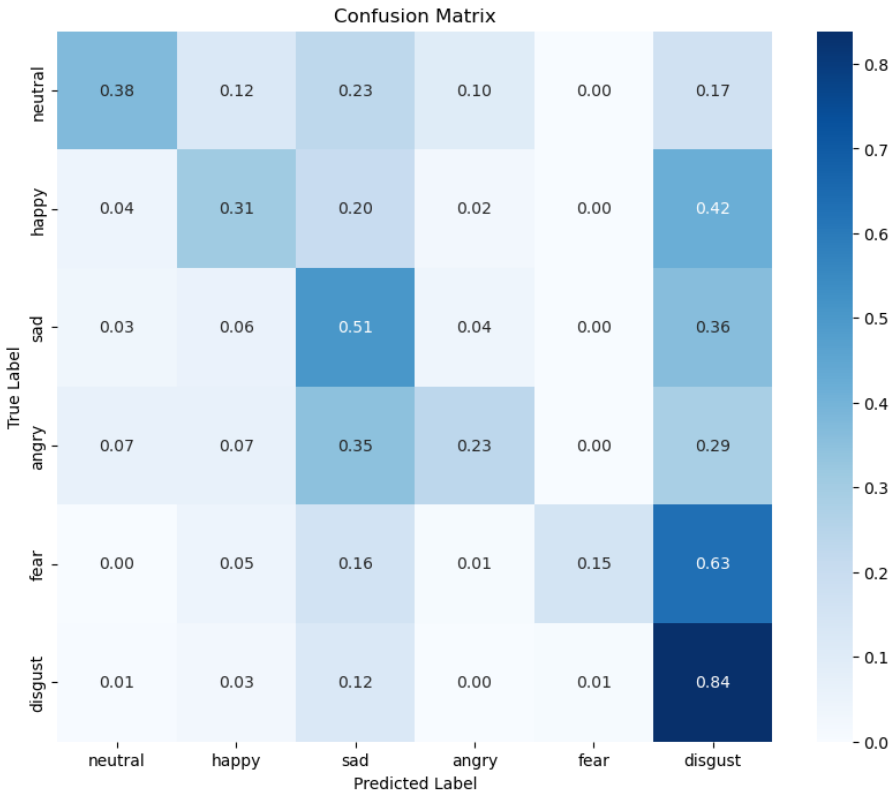

Figure 19: YAMNet Transfer Learning with no augmentations - Confusion Matrix

6 Discussion and Interpretation

6.1 Custom Model

For the group with augmentations and without stanadardization, the model predicts sad and neutral emotions with above 60% precision. However, the recall for neutral is 57% while sad is 36%. This means that the model is conservative when classifying the sad emotion. When it classifies sad it is right 63% of the time. However out of all the sad instances, it classifies them correctly 36% of time, meaning it's missing the sad emotion among all the other emotions. This indicates that the model is having a hard time to distinguish the sad emotion from other emotions as some of the features may be overlapping. Therefore, this highlights the importance of effective and optimized feature extraction, in order to create better separation from other emotions. However, the augmentations were applied for this group so the model could be having a hard time distinguishing the emotions due to the augmentations. In this group, the model confuses disgust with the neutral emotion and angry with the happy emotion.

For the groups without augmentations and no standardization, the model's performance improves drastically, indicating that the augmentations were too much and clouding the model's ability to predict efficiently. Similar to the other group, the model does really well with the neutral category, but still struggles with the disgust category. The model captures most happy samples (high recall) but often misclassified other emotions as happy, leading to the low precision. This indicates that the model is biased to the happy emotion. Additionally, the model struggles to identify the sad emotion (low recall); many of the sad samples are confused with neutral or the disgust emotion. This similar behavior is exhibited in the group without augmentations but with the standardization. Standardization can influence the ability of the model to predict certain emotions better than its non-standardization counterpart. For example, without augmentations but with standardization, the model has high precision and low recall for the happy emotion compared to the without standardization group. For the with-standardization group, the precision is lower than the recall for the happy emotion, even though both groups have around the same accuracy of 67%. This indicates that the standardization component isn't improving the model but rather shifting its focus on different components for audio classification. This could be exploited in future work by feeding standardized and non-standardized Mel Spectrograms and feature vectors into the model to allow the model to learn the optimal tradeoff of importance between standardized and non-standardized audio features for predicting emotions.

6.2 VGGISH

The VGGish model's results in predicting each emotion revealed varied performance across the categories. For Angry, the model demonstrated high precision (0.77) and recall (0.73), indicating strong accuracy in identifying this emotion. In contrast, for Disgust, precision and recall were relatively low at 0.65 and 0.58, suggesting frequent confusion with other emotions such as sadness and fear, as observed in the confusion matrix. Fear presented significant challenges, with a precision of 0.58 and a recall of 0.55, reflecting a high rate of misclassification, particularly as sad or happy emotions. Happy was identified as the most difficult emotion for the model, with precision (0.54) and recall (0.53), indicating frequent misclassification as fear and angry. The model performed well for Neutral, with precision (0.65) and recall (0.71), but there were notable misclassifications primarily with sadness and disgust. For sad, the model showed moderate performance with precision (0.59) and recall (0.66), often misclassifying this emotion as fear or neutral.

6.3 YAMNET

The test accuracy and recall are relatively low, showing that the model struggles to correctly classify instances. The precision is higher than recall, meaning the model avoids many false positives but at the cost of missing true positives. The f1-score highlights the imbalance between precision and recall, indicating the model's difficulty in achieving strong generalization. Finally, the test loss suggests that the model is making predictions that deviate significantly from the target values.

The model is very specific when predicting angry, but it fails most of the time. For disgust, the model has a bias when predicting fear, which results in many false positives but fewer false negatives. The model shows a strong bias when predicting fear. It had the most difficulty when predicting happy. For neutral, the model is conservative, leading to many false negatives. Finally, the model tends to overpredict sad.

7 CONCLUSION

In this study, we utilized four well-established emotion recognition datasets—CREMA-D, RAVEDESS, TESS, and SAVEE—combined into a single dataset to test the hypothesis that machine learning models, by leveraging acoustic features and Mel spectrograms, can achieve an accuracy exceeding 0.6 in recognizing six emotions: neutral, disgust, fear, sadness, happiness, and anger. Both the proposed model, which combined acoustic parameters and Mel spectrograms with CNN and MLP methods, and the transfer learning model using VGGish with acoustic parameters and CNN, achieved strong results, with accuracies of 0.67 and 0.63, respectively, based on this combined dataset. In contrast, the transfer learning model using YAMNet, which incorporated audio embeddings and Mel spectrograms, showed a lower accuracy of 0.40. These findings highlight that both the proposed model and the VGGish-based model performed effectively across the combined datasets, demonstrating the potential of using acoustic features and Mel spectrograms, along with CNN-based architectures, for advancing speech emotion recognition.

Our project demonstrates several strengths, particularly in its ability to detect six emotions—neutral, disgust, fear, sadness, happiness, and anger—achieving above 0.6 accuracy using the combined four datasets. This is a notable improvement over traditional models that typically focus on only four emotions. However, the project also has some weaknesses that need to be addressed. The use of only scripted acting from actors limits the naturalness of the speech data, which may impact the generalization of the model in real-world scenarios. Additionally, the surprised emotion was excluded from our model, and there was no testing conducted using standard evaluation datasets to compare our results with existing state-of-the-art models, which could have provided more comprehensive benchmarking.

Looking ahead, there are several promising future directions for improving the model. Optimizing feature extraction techniques and selecting acoustic features that can better distinguish emotions, especially those like disgust and fear, would improve accuracy. Furthermore, addressing scalability is important, as our current approach is constrained by the audio files recorded by actors. A broader range of naturalistic speech data could enhance the model's performance. The potential impact of this work lies in its ability to help build a standardized emotion detection set and provide insights into how people unconsciously express emotions, which could be valuable in applications such as human-computer interaction, mental health diagnostics, and sentiment analysis.

REFERENCES

Florian Eyben, Klaus R. Scherer, Bjorn W. Schuller, Johan Sundberg, Elisabeth Andre, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, 7:190–202, 2016.

Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53:1162–1171, 2011.

Dongdong Li, Jinlin Liu, Zhuo Yang, Linyu Sun, and Zhe Wang. Speech emotion recognition using recurrent neural networks with directional self-attention. *Expert Systems with Applications*, 173:114683, 2021.

Michael Neumann and Ngoc Thang Vu. Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech. *In Proceedings of the 18th Annual Conference of the international Speech Communication Association*, 2017.

Aharon Satt, Shai Rozenberg, and Ron Hoory. Efficient emotion recognition from speech using deep learning on spectrograms. *In Proceedings of the 18th Annual Conference of the international Speech Communication Association*, 2017.

Itsuki Toyoshima, Yoshifumi Okada, Momoko Ishimaru, Ryunosuke Uchiyama, and Mayu Tada. Multi-input speech emotion recognition model using mel spectrogram and gemaps. *Sensors*, 23:1743, 2023.

Zengwei Yao, Zihao Wang, Weihuang Liu, Yaqian Liu, and Jiahui Pan. Speech emotion recognition using fusion of three multi-task learning-based classifiers: Hsf-dnn, ms-cnn and lld-rnn. *Speech Communication*, 120:11–19, 2020.