Bilingual Speech Emotion Recognition: Transfer Learning and Real-Time Applications

Syed Mominul Islam[a], Md. Moazzam Hossain[a], Md. Khabir Uddin Ahamed*[b], Redwanul Haque Ifty[c], Yaser Samin[c], Md. Shanjidul Islam Sadhin[c]

[a]*Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh*
[b]*Department of Computer Science and Engineering, Bangamata Sheikh Fojilatunnesa Mujib Science and Technology University, Jamalpur, Bangladesh*
[c]*Department of Computer Science and Engineering, American International University Bangladesh, Dhaka, Bangladesh.*

Abstract

Background: Speech emotion recognition (SER) is a fundamental component of human-machine interaction, with applications in healthcare, customer service, and education. Real-time SER can enhance the effectiveness and naturalness of these interactions.

Objective: This study investigates the use of transfer learning to fine-tune pre-trained deep learning models from image identification, audio embedding creation, and automatic voice recognition domains to improve SER performance.

Methods: We utilized four English datasets (TESS, RAVDESS, SAVEE, CREMA-D) and two Bengali datasets (SUBESCO, BSER) to fine-tune the pre-trained models. Furthermore, we also created a bilingual dataset by combining all the datasets.

Results: Our proposed transfer learning approach outperformed previous benchmarks, achieving accuracies of 99.64% (TESS), 88.54% (RAVDESS), 78.12% (SAVEE), 72.53% (CREMA-D), 95.0% (SUBESCO), and 87.41% (BSER). The combined dataset accuracy was 86.51%. In real-time evaluation, the system achieved a weighted F1 score of 58.77%.

Conclusion: The proposed study demonstrates that transfer learning is an effective strategy for enhancing the performance of speech emotion detection models across multiple languages. The real-time system shows promise for real-world applications. The study's cost-effective and flexible approach highlights the potential of transfer learning to advance the field of speech emotion recognition.

Keywords: Speech Emotion Recognition, Transfer Learning, Real-Time Systems, Bilingual Datasets, Human-Machine Interaction

1. Introduction

Communication through speech is fundamental to human interaction and cultural development. From early settlements to modern society, sharing information verbally has been essential. Often, speech is accompanied by emotion, enhancing comprehension and connectionSajjad et al. (2020). Human cognition naturally integrates various

*Md. Khabir Uddin Ahamed

Email addresses: mominulbu56@gmail.com (Syed Mominul Islam), mdmoazzamhossain135@gmail.com (Md. Moazzam Hossain), khabir.cse@bsfmstu.ac.bd (Md. Khabir Uddin Ahamed*), ifty8555@gmail.com (Redwanul Haque Ifty), saminyaserwork@gmail.com (Yaser Samin), dipuahamed321@gmail.com (Md. Shanjidul Islam Sadhin)

cues, such as tone and body language, to interpret emotions in conversationsSonmez & Varol (2019). Recognizing emotions in speech not only facilitates understanding but also improves dialogue qualityWilliams & Stevens (1972). Emotion recognition technology has diverse applications, from enhancing service in call centers through mood detectionBurkhardt et al. (2006) to monitoring pilots' stress levels for aviation safety. In gaming, emotion-aware systems can enrich user experience by adapting to players' feelingsHossain et al. (2015). In mental health, chatbots equipped with emotion recognition can assist in therapy and diagnosis, similarly, conversational chatbots can benefit from understanding users' emotions to improve interactions Oh et al. (2017); Yenigalla et al. (2018). The field of speech emotion recognition (SER) has expanded rapidly, offering solutions in areas like call center dialogues, automatic response systems, and moreSingh & Goel (2022). As technology advances, integrating emotional context into human-computer interaction (HCI) becomes increasingly vitalMishra et al. (2022). This involves real-time data analysis to make interactions more intuitive. Speech emotion recognition, therefore, is crucial for developing natural and effective communication interfaces between humans and machines

Over the past two decades, various machine learning algorithms have been refined for automatic emotion recognition and these systems use acoustic, prosodic, and linguistic features to identify emotionsLieskovská et al. (2021). Challenges include the subjective nature of emotions and the difficulty of accurate data annotation. Traditional classifiers like Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) have been employed in SERDileep & Sekhar (2013). Recent advancements in deep learning offer advantages such as automatic feature extraction and robustness to noise Deng et al. (2014).

Transfer learning utilizes knowledge from one domain to improve performance in anotherPires de Lima & Marfurt (2019). In SER, this can reduce the need for extensive labeled data and accelerate training. Methods include fine-tuning pre-trained models and using them as feature extractors. Feature extraction is critical in SER, focusing on identifying unique speech characteristicsChoudhary et al. (2022). Spectral features, prosodic features, and others like the Mel Frequency Cepstral Coefficient (MFCC) are commonly used. MFCCs provide valuable insights into the auditory features relevant for emotion classification.

Again, in the subject of Real-time and batach prediction, Real-time prediction involves immediate data processing, while batch prediction analyzes large datasets simultaneously. Each approach has distinct applications and resource requirements. Real-time SER systems are crucial in fields like customer support and telemedicine, where quick emotion detection enhances user experience.

This study explored the effectiveness of transfer learning in recognizing emotions in speech for both English and Bengali. We employed pre-trained deep learning models from different domains with fine-tuning using a limited dataset for speech emotion identification in both languages. Our primary contributions are as follows:

- Examining the current state of speech emotion recognition and transfer learning, identifying key challenges and opportunities within these domains.
- We developed a methodology for fine-tuning pre-trained deep learning models on a limited dataset for speech emotion recognition in both English and Bengali.
- We conducted experiments to evaluate the effectiveness of the proposed approach and compared the results with existing literature.

Table 1: List of Abbreviations and Symbols.

Abbreviations	Description
HCI	Human-Computer Interaction
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
MEDC	Mel Energy-spectrum Dynamic Coefficients
PLP	Perceptual Linear Prediction
PCA	Principal Component Analysis
MFCC	Mel-frequency cepstral coefficients
ANN	Artificial Neural Network
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Network
Adam	Adaptive Moment Estimation
VGG	Visual Geometry Group
YAMNet	Yet Another Mobile Net
SSL	Self-Supervised Learning
BERT	Bidirectional Encoder Representations from Transformers
MIR	Music Information Retrieval
HuBERT	Hidden Unit Bidirectional Encoder Representations from Transformers

- We created a real-time system and utilized it to test the performance of our models in a simulated real-world environment.
- Finally, we analyzed and interpreted the results of our research, discussing their implications for speech emotion recognition and affective computing.

The rest of the article is organized as follows: Section 2 includes a wide range of overviews in this field. The methodology of the suggested work is demonstrated in Section 3. Experimental performance results, including a discussion and the dataset description, are presented in Section 4. Section 5 provides our conclusions and proposals for future development work.

2. Literature Review

Research in artificial intelligence and human-computer interaction is increasingly focused on detecting emotions from speech, with applications ranging from customer service to mental health support. These systems typically utilize machine learning algorithms to analyze various acoustic features like pitch and rhythm, often employing advanced deep learning techniques such as CNNs and RNNs for improved accuracy. Despite recent advancements, challenges remain, including the need for more annotated data, the diversity of emotional expressions, and the

systems' sensitivity to context and culture. This section explores the current state of speech emotion recognition. It covers various methods and techniques used, challenges and limitations of existing systems, and potential applications and implications of this technology.

Youdha Beer Singh and Shivani Goel Singh & Goel (2022) reviewed speech emotion recognition (SER) using machine learning (ML) and deep learning (DL) from 152 papers between 2000 and 2021. They highlighted three main challenges: selecting the right dataset, speech features, and classifiers. They found that DL techniques, especially those converting speech to spectrograms, often outperform traditional ML methods in emotion recognition tasks. Trinh Van, L., et al.Trinh Van et al. (2022)explored speech emotion recognition using CNN, CRNN, and GRU models, finding GRU to be the most effective with a 97.47% accuracy. They demonstrated that data augmentation and altering voice inputs significantly enhance model performance, despite increased memory use and training time. Their approach surpasses existing methods, highlighting the importance of model and feature selection alongside data augmentation for improved results. Koduru et al.Koduru et al. (2020) improved voice emotion recognition by using feature extraction methods like pitch, energy, and MFCC, focusing on noise reduction. Their system demonstrated better accuracy with classifiers: SVM at 70%, Decision Tree at 85%, and LDA at 65%, outperforming previous methods by effectively distinguishing emotions through comprehensive signal analysis.

Ragheb et al.Ragheb et al. (2022) introduce a novel method for detecting emotional speech by leveraging visual deep neural network models. They use transfer learning with pre-trained VGG-16 models, transforming auditory data into visual forms for emotion recognition. Their approach, tested on the Berlin EMO-DB dataset, shows significant advancements in recognizing seven distinct emotions. Alnuaim et al.Alnuaim et al. (2022) explore improving speech emotion detection by using a 1D CNN model, which surpasses traditional machine learning methods. Their approach leverages discriminative features and data augmentation across multiple language datasets, achieving high accuracy rates: 97.09% on BAVED, 96.44% on ANAD, and 83.33% on SAVEE.Islam et al.Islam et al. (2022) introduce a method for recognizing emotions with intensity from speech, utilizing 3D voice signals and deep learning frameworks. Their REIS approach employs a 3D CNN and Bi-LSTM architecture, achieving an 87.71% accuracy with limited training data. This model demonstrates significant improvements over existing techniques by effectively integrating 3D signal transformations. Nasim et al.Nasim et al. developed a Speech Emotion Recognition model using popular machine learning classifiers and combined the TESS and RAVDESS datasets for greater diversity. They used feature extraction techniques like MFCC, Chroma, and Mel Spectrogram, achieving 99.64% accuracy on TESS and 54.40% on RAVDESS with MLP. Gradient Boosting on the pooled dataset resulted in an 84.69% accuracy, with TESS yielding better outcomes due to its focus on female voices. S. Akinpelu and S. ViririAkinpelu & Viriri (2022) introduced a deep transfer learning approach for speech emotion classification using emotionally rich feature selection. By preprocessing audio to create Mel-spectrograms and applying NCA for feature selection, they improved accuracy on the TESS, EMO-DB, and combined datasets. Their method achieved accuracies of 97.20% on TESS, 94.82% on EMO-DB, and 95.77% on the combined dataset, proving effective for emotion classification.

Stolar et al.Stolar et al. reformulate real-time Speech Emotion Recognition (SER) as an image classification task using AlexNet, leveraging pre-trained networks to avoid extensive training. They developed two systems, AlexNet-SVM and FTAlexNet, both achieving state-of-the-art results on the EMO-DB database by converting voice spectrograms into RGB images. While FTAlexNet offers slightly higher accuracy, AlexNet-SVM is more computationally

Table 2: Analysis of Selected Works in Speech Emotion Recognition

Study	**Methods**	**Datasets**	**Key Findings**	**Limitaions**
Krishnan et al. (2021)	IMF-SVM, KNN	TESS	Attained 93.30% accuracy	Limited to Toronto Emotional Speech dataset.Focus on native English speakers
Akinpelu & Viriri (2022)	DCNN-NCA-MLP Aug.	TESS	Effective in classifying speech emotions	Potential generalization issues with different datasets or languages.
Guizzo et al. (2020)	AlexNet w/ MTS Aug.	RAVDESS	Effective on smaller datasets	Effectiveness dependent on dataset size and diversity.
Aggarwal et al. (2022)	DNN	RAVDESS	When analyzed with a DNN, mel-spectrogram images outperformed numeric features in terms of accuracy	Challenge remains in achieving consistently high performance across varying datasets and models.
Farooq et al. (2020)	DCNN-CFS	SAVEE	High accuracy in speaker-dependent tests	Lower accuracy in speaker-independent tests compared to speaker-dependent ones.
Alnuaim et al. (2022)	1D CNN	SAVEE	83.33% accuracy achieved on SAVEE	Results may vary across different languages and datasets.
Mocanu et al. (2021)	SE-ResNet	CREMA-D	Achieved identification rates of 83% on RAVDESS and 64% on CREMA-D	Performance varies between datasets, with lower accuracy on CREMA-D.
Dolka et al. (2021)	MFCC+ANN	CREMA-D	Outperformed previous methods on RAVDESS	Lower effectiveness on the CREMA dataset compared to others.
Zielonka et al. (2022)	CNN	Consolidated Dataset (English)	Mel-spectrograms are more effective for training CNNs in SER	Overall accuracy is relatively low, indicating room for improvement in emotion classification.
Sultana et al. (2021a)	CNN+TDF+ BiLSTM	Consolidated Dataset (Bengali)	Provides a new direction for Bengali SER research	Limited by its focus on low-resource language datasets, which might affect generalizability.

efficient. Chatterjee et al.Chatterjee et al. (2021a) developed an AI-based smart home assistant using MFCC features and a 1D CNN for emotion recognition in speech. Their method achieved classification accuracies of 90.48% on RAVDESS and 95.79% on TESS. This approach enhances the assistant's ability to detect user emotions for improved support and feedback. Rehman A. et al.Rehman et al. (2022) developed a real-time system for emotion recognition using syllable-level features and a simple neural network. Despite achieving only average results in cross-corpus testing, the system excels in low latency due to its simplicity, operating as a browser-based application with Tensorflow JSSmilkov et al. (2019). However, performance issues on lower-end devices and JavaScript's speed limitations may hinder its practical use. Some recent state-of-art works on speech emotions recognition are presented in Table2.

An innovative SER architecture combining a time-distributed flattened layer with a DCNN and BLSTM network is proposed by Sultana et al. (2021a,b), achieving 86.86% accuracy on the Bengali SUBESCO dataset and 82.7% on the English RAVDESS dataset. This approach is expected to advance research in Bengali, a low-resource language. Chakraborty et al.Chakraborty et al. (2022) developed a Speech Emotion Recognition model for low-resource Indian languages using phase information-based cepstral coefficients and gradient boosting. Their approach achieved an average accuracy of 97% on Bengali datasets SUBESCO and BSER, demonstrating robust performance across multiple standard datasets. The transformer model by Al-onazi et al.Al-onazi et al. (2022) integrates seven acoustic features for emotion recognition, using data augmentation prior to feature extraction. It achieved impressive accuracy rates on multiple datasets: 95.2% on BAVED, 93.4% on EMO-DB, 85.1% on SAVEE, and 91.7% on EMOVO, demonstrating its effectiveness, particularly for Arabic vocal emotions.

3. Methodology

3.1. Datasets

3.1.1. TESS

The Toronto Emotional Speech Set (TESS) datasetPichora-Fuller & Dupuis (2020) featuring 200 target words spoken by two English-speaking actresses aged between 26 and 64. Comprising 2800 audio samples, the collection represents seven emotions: anger, disgust, fear, happiness, surprise, sadness, and neutrality. As depicted in Figure **??**, the actresses hail from the Greater Toronto Area, are native English speakers, and possess a university education along with musical training. Audiometric assessments confirmed that both have normal hearing abilities.

3.1.2. RAVDESS

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) datasetLivingstone & Russo (2018) includes emotional speech recordings from 24 professional actors with a North American accent, featuring eight emotions across 1440 audio files. These emotions include happy, surprised, sad, disgusted, angry, and a combined calm/neutral class. Each emotion is represented by 192 samples, except for neutral, which has 96 presented in Figure **??**. The dataset focuses on audio files with spoken content, excluding songs, and uses ".wav" files named with a specific coding system that indicates various attributes like presentation type, emotion, intensity, and actor number.

3.1.3. SAVEE

The Surrey Audio-Visual Expressed Emotion (SAVEE)Jackson & Haq (2014) database contains recordings from four male speakers of British English, showcasing seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. It includes 480 utterances as shown in Figure **??**, with each speaker contributing 120. The dataset features 15 sentences per emotion, comprising three common, two emotion-specific, and ten phonetically-balanced sentences. Additionally, 30 neutral utterances were recorded using three common and twelve emotion-specific statements.

3.1.4. CREMA-D

The CREMA-D datasetCao et al. (2014) is designed for multimodal emotion identification, featuring 7442 samples from 91 ethnically diverse actors. It covers six emotions: anger, happiness, sadness, fear, disgust, and neutral. Labels in Figure **??** were created with input from 2443 raters on a crowdsourcing platform, resulting in 223,260 evaluations that averaged emotion type and intensity. The dataset aims for an equal distribution of samples across each emotion category.

3.1.5. SUBESCO

The SUST Bangla Emotional Speech Corpus (SUBESCO) Sultana et al. (2021b) includes recordings from 20 speakers (10 men and 10 women) aged 21-35. Each participant recorded 10 sentences in seven emotional states: anger, contempt, fear, happiness, neutral, sadness, and surprise, with five repetitions per emotion, totaling 7000 utterances is depicted in Figure**??**. Each sentence lasts 4 seconds with consistent structure and includes diverse phonetic elements. The dataset is in ".wav" format with a 48KHz sample rate, and its total duration is 7 hours and 40 minutes, making it the largest Bengali SER dataset available.

3.1.6. BSER

BanglaSER (BSER) is a recent databaseDas et al. (2022) for Bengali speech emotion recognition, featuring recordings from 34 nonprofessional speakers (17 men and 18 women) aged 19 to 47. It includes 1,467 voice samples capturing five emotions: anger, happiness, neutrality, sadness, and surprise as presented in Figure **??**. Each emotion is represented by three trials, with 1,224 recordings for the main emotions and 243 recordings focusing on neutrality. The audio files are in ".wav" format with a 44.1 kHz sample rate, totaling 1 hour and 29 minutes.

3.1.7. Consolidated Datasets

To further assess our models, we combined all 6 of the previously mentioned speech emotion recognition datasets into 3 consolidated datasets. These consolidated datasets allowed us to evaluate our models' ability to generalize and perform well on complex, diverse datasets that incorporate a wide range of speech samples and emotional representations.

We merged the four English datasets (TESS, RAVDESS, SAVEE, CREMA-D) to create a unified English dataset. Similarly, we combined the two Bengali datasets (SUBESCO and BSER) to form a unified Bengali dataset. Finally, we integrated these to develop a bilingual dataset. Refer to figures **??**, **??**, and **??** for the class distribution of these consolidated datasets.

3.2. Data Preprocessing and Splitting

3.2.1. Preprocessing

Data preprocessing involves modifying data before it is used for model training to enhance performance. Given the diversity of datasets, the characteristics of audio samples vary significantly. A notable difference is in sample rates; for instance, the TESS dataset operates at a sample rate of 22,500 Hz, while the RAVDESS dataset uses 44,100 Hz. To ensure uniformity across all datasets, we down-sampled all audio samples to 16,000 Hz. This sample rate is widely accepted in human speech-related machine learning tasks, as the human hearing range extends up to 20 kHz Rosen & Howell (2011). Alongside standardizing the sample rate, we also trimmed each audio clip to remove any silence at the start or end. A threshold of 25 decibels was set, meaning any signal below this level was excluded from the final sample. A visual example of a trimmed audio sample is shown in Figure **??**.

3.2.2. Splitting

In deep learning, data splitting involves dividing a dataset into multiple subsets. This process is similar to other machine learning approaches, aiming to create a training set, a validation set, and a test set. In this study, we partitioned the dataset in a 70:10:20 ratio for training, validation, and testing for each model. These subsets were then used in both training and evaluation phases. During training, the model is fine-tuned using the training data, which needs to be plentiful and representative of the broader population the system will serve.

To ensure balanced distribution, we stratified the splits based on classifications or labels (such as emotions like angry, happy, sad, etc.), so each subset contained an equal number of samples for each class. Only the training set underwent data augmentation, while the validation and test sets remained unchanged. A validation step was incorporated during training to optimize the model parameters, focusing on improving metrics like classification accuracy. Validation performance measures how well the model classifies data during training, but its true effectiveness is determined by performance on the test set, which assesses generalization to new data. In the evaluation phase, the retrained model is tested on unseen samples, and the accuracy obtained is known as testing performance. It's essential to prevent any overlap between training and testing samples to maintain the integrity of the results.

3.3. Proposed Models

3.3.1. Image Classification Models

- VGG16: The ImageNet dataset, consisting of over 14 million images organized into more than 22,000 categories, was used to train a convolutional neural network developed by K. Simonyan and A. ZissermanSimonyan & Zisserman (2014).

  In VGG16, the architecture starts by processing 224x224x3 pixel images through two convolutional layers, followed by a max-pooling layer as shown in Figure 1. This is succeeded by two more convolutional layers and another max-pooling layer. The design then incorporates a max-pooling layer, three convolutional layers, another max-pooling layer, three additional convolutional layers, and concludes with a max-pooling layer. After these layers, fully connected and ReLU layers are introduced. Both convolution and max-pooling layers utilize a stride of 2, with convolutional filters sized at 3x3. In a similar alternative design, VGG-19 consists of

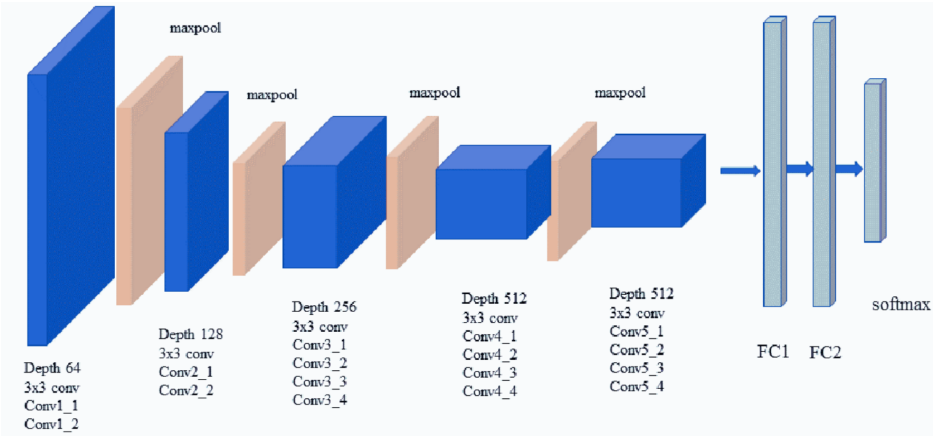


Figure 1: VGG16 Architecture

three fully connected layers, sixteen convolutional layers, a SoftMax layer, and five max-pooling layers, using filters with sizes of 64, 128, and 256 in the convolutional layers.

- ResNet50V2: He et al. He et al. (2016) proposed a novel deep residual network, ResNet50V2. ResNets, or deep residual networks, are composed of numerous stacked "Residual Units" that can be represented in a generic form, as demonstrated in the Equation 1.

$$y_l = h(x_l) + F(x_l, W_l), x_l + 1 = f(y_l) \tag{1}$$

The input and output of the l_{th} unit are x_l and x_{l+1}, respectively, while F is a residual function. Since $h(x_l) = x_l$ is the identity mapping, we can call f a ReLU functionNair & Hinton (2010). ResNets work by attaching a "identify skip" connection or "shortcut," allowing a network to learn the additive residual function F with respect to $h(x_l)$. Figure 2 depicts ResNet50's underlying architecture. ResNet50V2 now includes a new residual unit to facilitate training and improve generalization, making it an upgraded version of ResNet50. ResNet50V2 was pre-trained on the popular dataset ImageNet, which comprises over 1000 classes of single-label data.

- Hyper-Parameters for Fine-Tuning

  i Loss Function: The loss function measures how well a neural network models a dataset by producing lower values for accurate predictions and higher values for inaccurate ones. Different loss functions are used for tasks like regression and classification, with categorical cross-entropy being effective for multiclass classification problemsIslam et al. (2021), such as speech emotion recognition. This loss function, also known as "logarithmic loss," penalizes predictions based on their deviation from actual values. In this context, categorical cross-entropy was found suitable due to the diverse types of SE image categories involved.

  ii Optimization Function: Optimization involves minimizing loss or improving a neural network's accuracy. Adam optimizationKingma & Ba (2014), used for re-training models like VGG16 and ResNet, adjusts

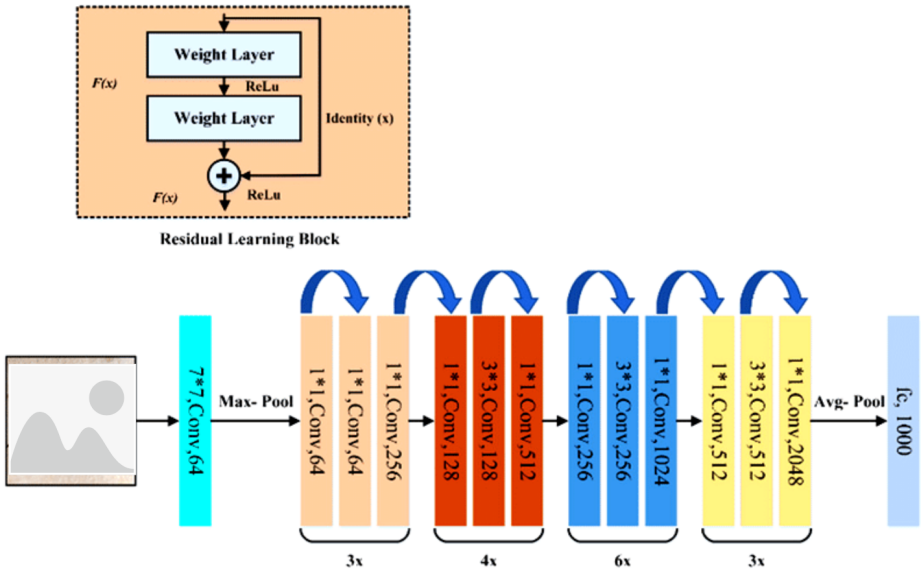


Figure 2: ResNet Architecture

learning rates for each network weight based on the network's progress. It combines the benefits of RMSProp and AdaGrad, where RMSProp averages recent gradient magnitudes and AdaGrad tracks individual learning rates for sparse gradients. This approach is robust against noise, making it suitable for complex scenarios.

iii Learning Rate: Transfer learning addresses the cross-domain issue in speech emotion recognition (SER) when training and test datasets differ. Song et al.SONG et al. (2014) tackled this using dimensionality reduction and Maximum Likelihood methods. The learning rate (LR) plays a crucial role, with values ranging from 0.01 to 0.000001, but a lower rate, like 0.0001, is preferred to stabilize training and enhance deep learning model performance. This approach helps achieve optimal results in cross-corpus scenarios. Overfitting in deep learning can be reduced using dropout and early stopping. Dropout involves randomly omitting neurons during training, typically using rates between 0.2 and 0.5, with 0.5Stivaktakis et al. (2019) often being effective. Consequently, we used 0.5 (i.e., 50%) to lessen over-fitting during training Early stopping halts training when a model's performance levels off, acting as a regularization technique in deep CNNs.

iv Activation Function: In this research, activation functions like ReLUNair & Hinton (2010) and SoftmaxMahdianpari et al. (2018) are applied to convolutional layers to enhance neural network performance. These functions are favored over Tanh and Sigmoid because they are more efficient, activating more neurons with simpler mathematical operations. When x is greater than zero, the ReLU function outputs the value of x. Conversely, when x is less than zero, the output is zero (as shown in equation 2). This indicates that only neurons with positive inputs are activated. The gradient of ReLU is constant, with a slope of either 1 $\forall x, x \geq 0$ or 0 $\forall x, x < 0$.

$$f(x) = \max(0, x) = \begin{cases} x_i & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2}$$

In multi-class classification tasks, the softmax function is used to predict the class with the highest probability from the input labels. Since we were working with a multi-class problem using our speech emotion imagery data, this activation function was appropriate. Softmax outputs values between 0 and 1, ensuring that the total probability across all classes sums to 1. If the input vector has $N > 1$ or $N = 0$,the softmax output will fall within the range $(0, 1)$. The function $f(zi, j)$ for N classes is computed using the equation referenced in 3.

$$s = f(z_{i,j}) = \frac{exp(z_i)}{\Sigma_{j=1}^{N} exp(z_j)} = \frac{e^{zi}}{\Sigma_{j=1}^{N} e^{zj}} \tag{3}$$

Early layers in pre-trained CNN models capture basic information such as colors and edges. In contrast, deeper layers extract more complex features specific to class details. Therefore, the initial layers typically require little to no modificationYosinski et al. (2014). In this study, we concentrated on fine-tuning the last three layers of the VGG-16 model, which are responsible for defining the 1000 classes. These modifications are crucial for adapting to a new classification taskSonawane & Shelke (2018); Lu et al. (2019). Hyper-parameters taken for VGG16 and ResNetV2 are shown in Table 3. The transfer learning pipeline of VGG16 and ResNetV2 is depicted in Figure 3.

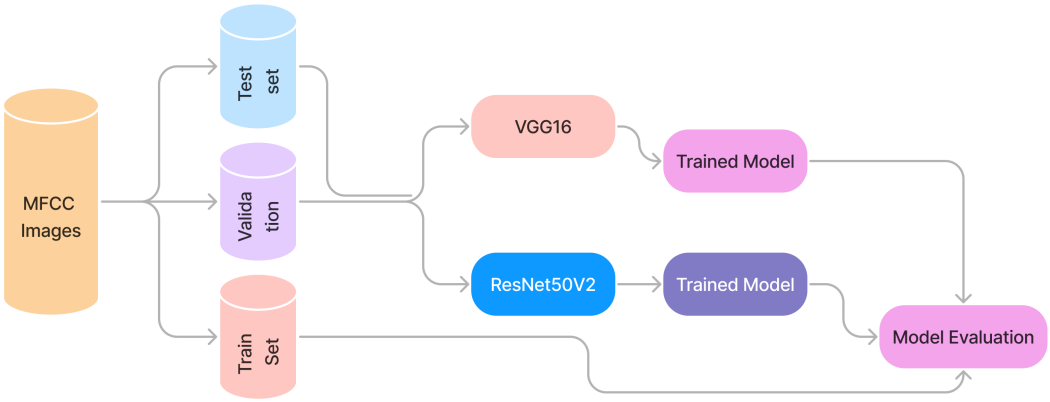


Figure 3: Transfer Learning Pipeline for VGG16 and ResNetV2 models.

To optimize fine-tuning, remove all network layers except the last three. Replace these with a fully connected (FC) layer, a SoftMax layer, and a classification output layer. The FC layer size should match the number of classes in the new dataset Sonawane & Shelke (2018); Lu et al. (2019).

After extracting features, audio MFCCs are processed using the Pillow Python module and resized from 288x288 to 224x224 pixels. The data is categorized into seven emotions: happy, sad, angry, neutral, disgust, fear, and surprise. It is split into a training set, test set, and validation set with a 70:20:10 ratio. These inputs are fed into the VGG-16 and ResNet50V2 models after passing through a 2D convolutional layer, converting all images to a resolution of 224×224 x 3 pixels.

Table 3: Hyper-parameters for VGG16 and ResNetV2

Parameter_Name	VGG16	ResNet50V2
Learning Rate	1e-5	1e-5
Batch Size	64	64
Optimizers	Adam	Adam
Loss Function	Categorical _Cross Entropy	Categorical_Cross Entropy
Width, Height	224*224	224*224

In the VGG-16 model, the last three layers are frozen, and the output layer is flattened. A fully connected layer with SoftMax activation is added to tailor the model to the dataset. For ResNet50V2, all layers are set to non-trainable. It's incorporated into a sequential model with a GlobalAveragePooling2D layer, which reduces the feature vectors from 7x7x2048 to 2048. A dense layer with 1024 units and ReLU activation follows, along with dense and dropout layers set at 0.4. Finally, a fully connected layer with SoftMax activation is added for emotion classification.

3.3.2. Audio Embedding Generation Models

Audio classification models take audio inputs and output labels, categorizing new audio samples based on their training. Embeddings, on the other hand, are numerical vectors that represent data in a lower-dimensional space, enabling meaningful predictions. Combining these concepts, audio embedding generation involves converting audio samples into fixed-length embeddings that capture essential properties for tasks like classification, speaker identification, and audio retrieval Cramer et al. (2019); Lukic et al. (2016); Surís et al. (2018). Traditional methods use hand-crafted features, while modern deep learning approaches train neural networks to learn embeddings directly from raw audio. The goal is to create compact, useful representations for various applications.

In this section, we explore two deep learning models for generating audio embeddings: VGGish and YAMNet. We then explain how these models can be applied to infer emotions from speech.

- VGGish: VGGish Hershey et al. (2017) is an audio event embedding model trained on the YouTube-8M dataset Abu-El-Haija et al. (2016), previously known as AudioSet. Inspired by VGG networks for image classification, VGGish consists of convolutional and activation layers, followed by max-pooling, totaling 17 layers. It takes audio waveforms sampled at 16000 Hz and produces $(N, 128)$ embeddings that capture the semantic information of the audio, where N represents the number of frames, each with 15600 samples or 0.96 seconds. Transfer Learning Pipeline of model VGGish is depicted in Figure 4.

- YAMNet: YAMNet Ellis & Plakal is a pre-trained deep neural network that classifies audio into 521 classes, using the YouTube-8M dataset. It utilizes the Mobilenet v1 architecture, known for its depth-wise separable convolutions. This lightweight network is designed for low latency, making it suitable for embedded devices. YAMNet outputs both class labels and embeddings with a size of $(N, 1024)$, where N is the frame count. The frame size and input sample rate match those of the VGGish model. Transfer Learning Pipeline of model YAMNet is presented in Figure 4.

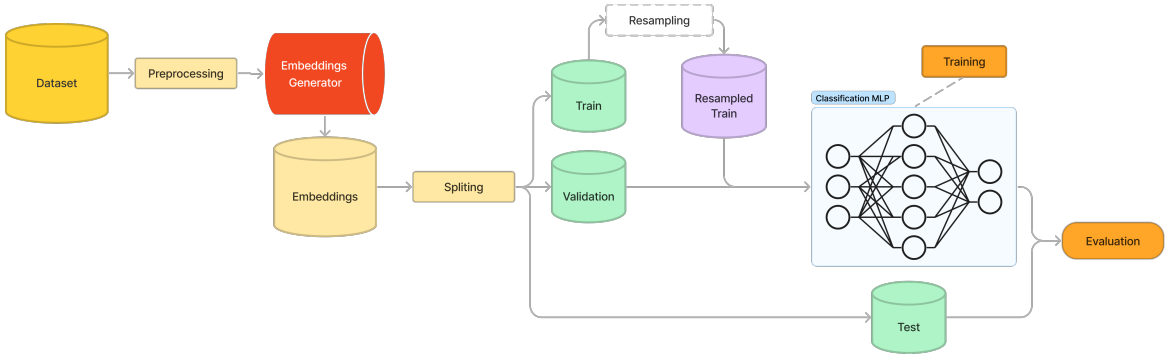


Figure 4: VGGish and YAMNet Transfer Learning Pipeline

- High-Level Feature Extractor: Both VGGish and YAMNet serve as high-level feature extractors. Audio samples are processed through these models to generate embeddings for each frame. These embeddings are then used to train a simple multi-layer perceptron (MLP) to predict emotion classes. The MLP is trained for up to 5000 epochs with a batch size of 256, using an early stopping mechanism that halts training if validation loss doesn't improve for 20 epochs. Additionally, the learning rate is reduced by half if no improvement is seen in the validation loss for 3 consecutive epochs.

  Hyperparameter tuning for the MLP, including the number and size of hidden layers, dropout rate, optimizer (Adam and AdaMax), and learning rate, was performed using random grid search due to the high dimensionality of the search space. This approach randomly samples combinations from a predefined set of possible values (see Table 4). Adam and AdaMaxKingma & Ba (2014), both adaptive learning rate optimizers, were chosen for their efficiency and robustness in neural network training, particularly AdaMax's suitability for time-varying data like speech. Adam adjusts learning rates based on past gradients, while AdaMax uses the infinity norm instead of the L2 norm used in Adam.

Table 4: Grid Search Parameters

Parameter Name	**Options**
Hidden Layer	[4096], [2048], [4096, 2048], [4096, 2048, 1024], [2048, 1024], [1024, 512], [512, 256], [1024, 512, 256], [256], [512], [1024]
Dropout Rate	0, 0.2, 0.5
Optimizers	Adam, Adamax
Initial Learning Rate	1e-3, 1e-4, 1e-5

Audio sample durations vary significantly across datasets due to factors like speech rate and phrase length, leading to an imbalanced dataset with some classes having far more samples than others (Figure **??**). Since VGGish and YAMNet generate embeddings per frame, longer samples produce more embeddings. To address this class imbalance caused by varying durations (Figure 5), a random oversampling and undersampling technique was employed to balance the number of samples in each class. To determine the balance, we

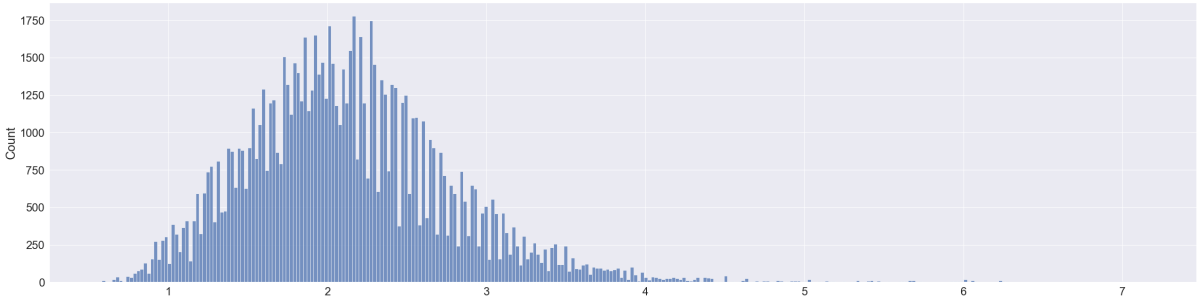


Figure 5: Histogram of audio duration (in seconds) of samples of all 6 datasets (After preprocessing)

computed the harmonic means of the total samples across all classes using the following Equation 4.

$$H = (\frac{\sum_{i=1}^{n} x_i^{-1}}{n})^{-1} \tag{4}$$

where, n is the number of classes and x_i is the i-th sample. The harmonic mean is calculated by taking the reciprocal of the average of the reciprocals of a set of numbers. This involves summing the reciprocals of the numbers and dividing by the count of the numbers. It is always less than or equal to the arithmetic mean and becomes closer when the numbers are similar. However, if the numbers vary greatly, the harmonic mean can be much smaller. For example, a set with both very large and very small numbers will have a harmonic mean significantly lower than its arithmetic mean. This makes the harmonic mean preferable in scenarios where you want to minimize the influence of classes with a large number of samples. To balance the dataset, we reduced samples in classes exceeding a threshold H and increased them in classes below H as shown in Figure**??**. This was achieved by randomly removing or duplicating samples. Only the training set was adjusted, while the validation and test sets remained unchanged.

- HuBERT: HuBERTHsu et al. (2021) is a BERT-like model for speech processing using self-supervised learning, similar to wav2vec 2.0. It distinguishes itself by employing a simpler loss function, a different clustering method, and embeddings from BERT's encoderVaswani et al. (2017) layers. Unlike wav2vec 2.0, which quantizes the entire convolutional network, HuBERT only quantizes the network's output, enhancing target quality. The authors found that embeddings from BERT's intermediate layers provide better-quality targets than CNN outputs alone. HuBERT processes audio by dividing it into 25ms segments and applying K-Means clustering on extracted MFCC features. Each cluster forms an invisible unit, labeling associated audio frames. In the next phase, these units are mapped to embedding vectors for predictions. This stage mirrors BERT's masked language modeling, where about 50% of input features are masked. The model predicts these areas using the cosine similarity between the transformer's outputs and the embeddings of hidden units. Incorrect projections are penalized with cross-entropy loss.

  We fine-tuned the HuBERT model on six individual datasets and three combined datasets using the base version pre-trained on Librispeech 960 Panayotov et al. (2015). There is also a larger model called 'hubert-large-1160k' trained on Libri-light Kahn et al. (2020), with 300M parameters, compared to the base model's 90M. Additionally, an X-Large version exists with 1B parameters. Due to high GPU memory requirements

and extended training time, we opted for the Base model. However, brief tests with the Large model showed a 2-4% improvement in F1 score on some datasets. The transfer Learning Pipeline of HuBERT is depicted in Figure**??**

We split the datasets into 70:10:20 for training, validation, and testing, respectively. Unlike previous experiments, we incorporated data augmentation, detailed in Section 3.4, to enhance only the training set. This adjustment was made because BERT-based models, with their large number of parameters, were prone to overfitting. To address this, data augmentation was used to increase the sample size, aiming for better model generalization. We compared results from both unaugmented and augmented datasets. Before training, we froze the feature extractor (CNN block, see Figure **??**) and fine-tuned the remaining architecture.

3.4. Augmentation

Data augmentation involves creating synthetic training samples by slightly altering the original data, enhancing the model's robustness and generalization. For training HuBERT models, five audio augmentation techniques were used. The first set includes traditional methods: adding random noise, pitch-shifting, and time-shifting. The second set employs SpecAugment Park et al. (2019). These augmentations expanded the training dataset to five times (5-fold) its original size.

3.4.1. Classic Augmentations

- Random Noise: Noise refers to random fluctuations that disrupt a signal and typically contain no useful information. In audio augmentation, noise involves adding random signals to clean audio, intentionally degrading its quality. We augmented our training set with these noisy samples. The noise level is set at 3.5% of the maximum value in an audio sample. This balance is crucial: too much noise can distort the signal, while too little may be ineffective. The detailed algorithm is outlined in Algorithm 1. Figure 6 shows the random noise augmentation on a phrase from the TESS dataset.

Algorithm 1 Random Noise Augmentation

function AddRandomNoise($data$)
  $NoiseRatio \leftarrow 0.035$
  $Max \leftarrow \max(data)$
  $RandomUniform \leftarrow$ Random number from a uniform distribution $[0.0, 1.0]$
  $NoiseAmp \leftarrow NoiseRatio \times Max \times RandomUniform$
  $Noise \leftarrow np.random.normal(size = data.shape[0])$
  $AugmentedData \leftarrow (NoiseAmp \times Noise) + data$
  return $AugmentedData$
end function

- Pitch Shifting: Pitch refers to the attribute of sound related to the frequency of air vibrations. Higher frequencies produce higher pitches, often associated with lighter voices, while lower frequencies result in deeper voices. Pitch shifting alters a sound's pitch and is useful for reducing overfitting in models trained

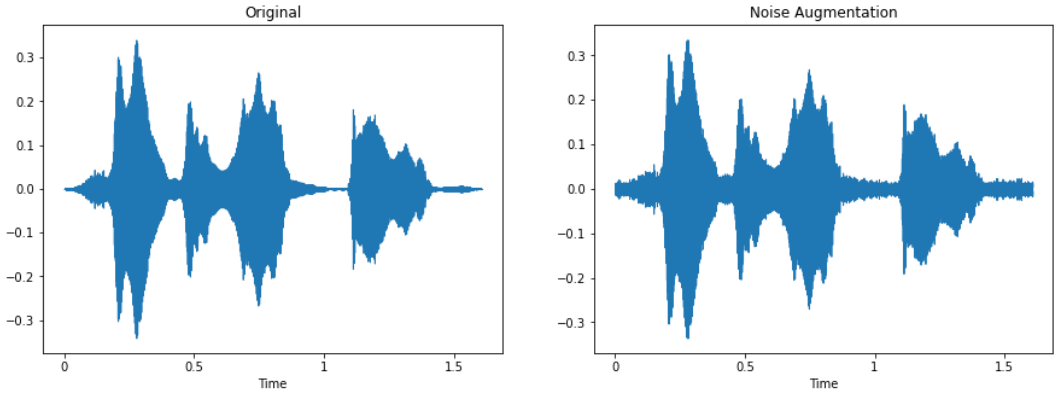


Figure 6: Random noise augmentation was performed on the phrase "Say the word bath" from the TESS dataset.

on datasets with audio samples from both genders. By applying a pitch shift factor of 0.7 to each training sample, we enhance the model's ability to detect emotions across different speakers.The detailed algorithm is outlined in Algorithm 2. Figure 7 demonstrates this technique.

Algorithm 2 Pitch Shifting

function PitchShift(data)
    $n\_steps \leftarrow 0.7$
    $fft\_data \leftarrow \text{FFT}(data)$
    $bins\_per\_octave \leftarrow 12$
    $semitones \leftarrow n\_steps \times (bins\_per\_octave/\text{len}(fft\_data))$
    $AugmentedData \leftarrow \text{Roll}(fft\_data, \text{int}(semitones \times \text{len}(fft\_data)/bins\_per\_octave))$
    $AugmentedData \leftarrow \text{Inverse\_FFT}(AugmentedData)$
    return *AugmentedData*
end function

- Time Shifting: Time-shifting involves moving a signal in time by adjusting the time variable. For a continuous-time signal $x(t)$, this is expressed as $y(t) = x(t \pm t_0)$. In datasets like TESS, where audio samples often begin with "Say the word...", time-shifting is used to disrupt such patterns, helping models to generalize better. We apply random time-shifting by rolling each audio sample by a factor $S_t = R \times 1000$, where R is chosen from a uniform distribution between -5 and 5. The detailed algorithm is outlined in Algorithm 3. Rolled elements reappear at the start to maintain the audio's duration. Figure 8 shows this process.

Algorithm 3 Random Time Shifting

function AddRandomTimeShifting(data)
    $S_t \leftarrow$ Random number from uniform distribution $[-5, 5] \times 1000$
    $AugmentedData \leftarrow \text{Roll}(data, S_t)$
    return *AugmentedData*
end function

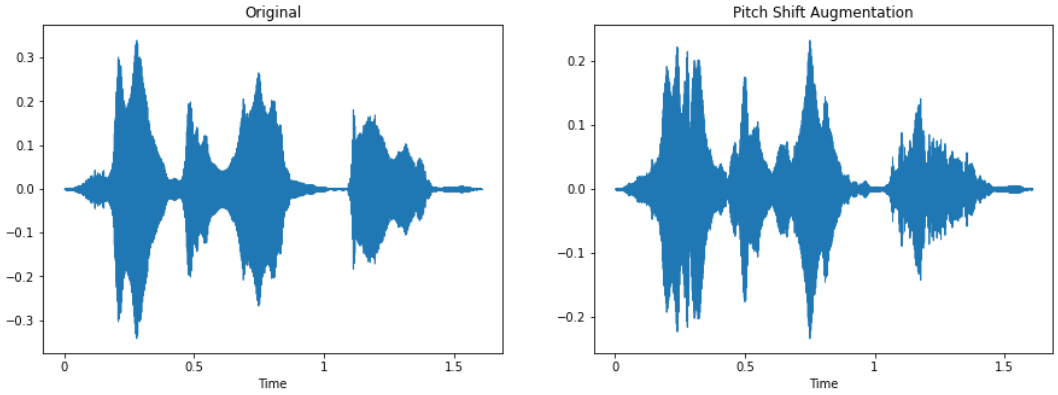


Figure 7: Pitch shifting augmentation was performed on the phrase "Say the word bath" from the TESS dataset.

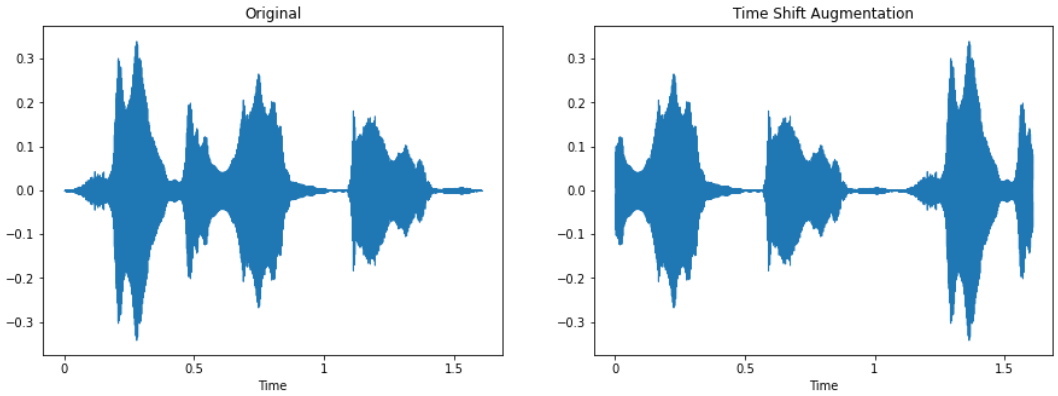


Figure 8: Time Shifting Augmentation done on the phrase "Say the word bath" from TESS dataset

3.4.2. SpecAugment

SpecAugment, as introduced by Park et al. (2019), is an augmentation technique applied directly to the spectrogram of an input utterance. This method has shown to enhance the performance of ASR networks on the LibriSpeech 960h and Switchboard 300h datasets. It includes three augmentation strategies: time warping, frequency masking, and time masking. The latter two are particularly effective in improving models that tend to overfit, and thus, we've integrated them into our training process. Both techniques operate similarly but along different axes. In frequency masking, f consecutive Mel frequency channels $[f_0, f_0 + f)$ are obscured, with f selected from a uniform distribution between 0 and the frequency mask parameter F, and f_0 chosen from $[0, v - f]$, where v is the number of Mel frequency channels. Time masking, conversely, affects the time axis, masking t consecutive time steps $[t_0, t_0 + t)$, with t chosen similarly to frequency masking. Figure 9 shows this process.

3.5. Real-time Speech Emotion Recognition (RTSER)

Real-time applications operate by detecting, processing, and acting on data streams instantaneously, unlike database-centric applications that store data for later analysis. In the context of speech emotion recognition, this involves continuously analyzing speech signals for immediate classification and response, known as online prediction. This contrasts with batch processing, where speed is less critical. A robust system must connect the user to the model,

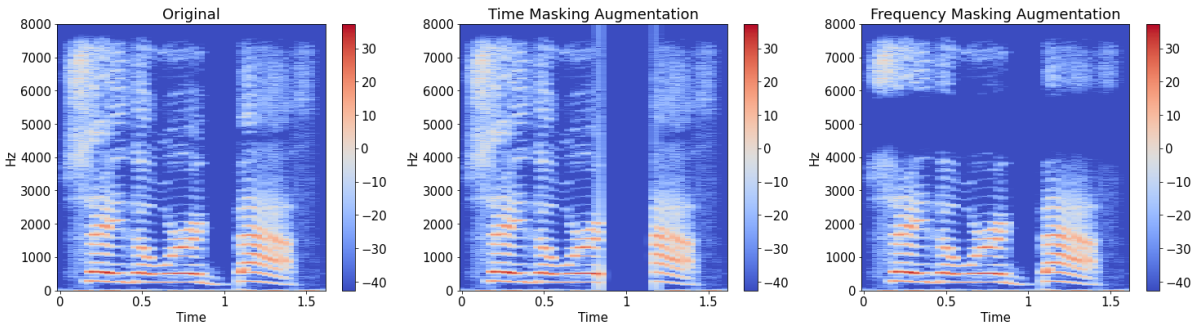


Figure 9: SpecAugment performed on the phrase "Say the word bath" from TESS dataset

allowing fast data transmission and result display, necessitating a user-friendly interface. Additionally, the models should generalize well and perform under suboptimal conditions. Thus, a real-time speech recognition system must balance speed, performance, and flexibility.

When running inference on new data, challenges such as mismatched accents, unfamiliar words, and poor sound quality can arise. To address these, it's crucial to train models on diverse datasets that include various accents, words, genders, and languages. Using techniques like noise augmentation can improve model accuracy under suboptimal conditions. These strategies were central to our model training approach.

Building an effective real-time speech emotion recognition system is challenging because emotions often overlap, such as fear with surprise or anger with disgust. Acted datasets don't always reflect these nuances, causing misclassification. To address this, we merged similar emotions. According toJack et al. (2014), four primary emotions are expressed by humans: Anger/Disgust, Fear/Surprise, Happiness, and Sadness. We combined Anger with Disgust and Fear with Surprise, omitting the Neutral class, to focus on these four categories for predictions.

3.5.1. Real-Time System Design

The real-time speech emotion recognition (RTSER) system is split into two main components: the front end and the back end. These parts work together to enable online predictions from speech input. The front end serves as the user interface for interacting with the emotion recognition models, while the back end processes requests and provides responses. This division addresses the issue highlighted byRehman et al. (2022) by eliminating the need for high computation power on the client side. The back end handles model hosting and inference, allowing the system to run on any device with a web browser, regardless of its resources. These components are detailed further and shown in a diagram 10. The code of the system is openly available on GitHub[1] under GPLV3 License (2007) license. The whole system is also available as a Docker Merkel et al. (2014) image, ready to be deployed to any server[2].

- Front-end: The front end is built with ReactAggarwal (2018), a widely used open-source JavaScript framework. It utilizes the Web Audio API to record audio from the user's microphoneSmus (2013). Periodically, it sends

[1] `https://github.com/SaminYaser-work/Real-Time-Speech-Emotion-Recognition`

[2] `https://hub.docker.com/repository/docker/fo0d/real_time_speech_emotion_recognition`

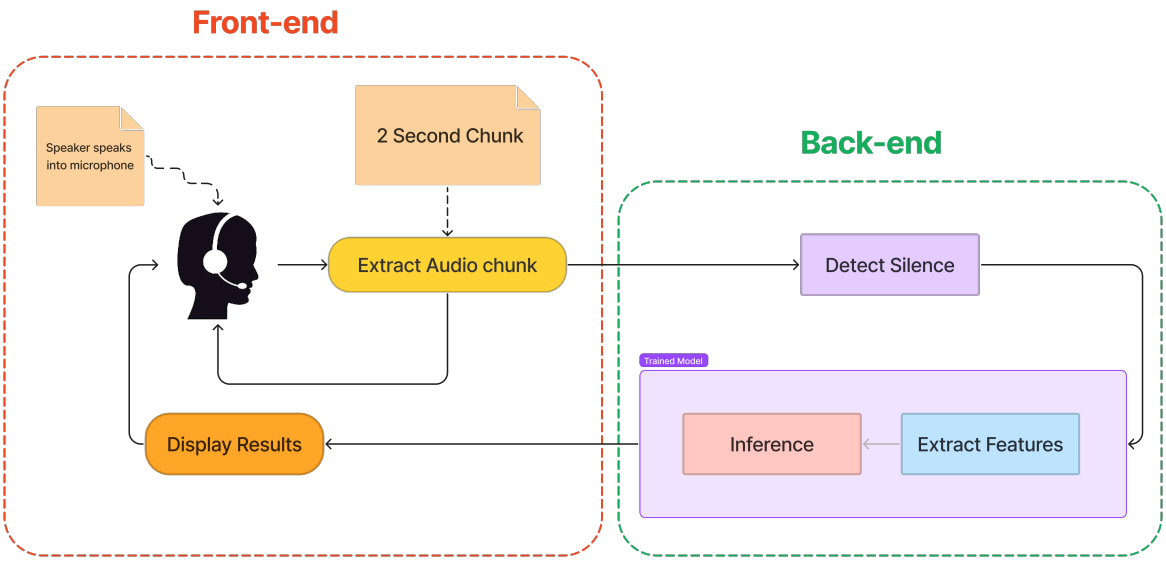


Figure 10: Proposed RTSER System

audio segments to the back end, allowing clients to receive immediate resultsChatterjee et al. (2021b), unlike previous approaches. Once the speech emotion recognition model on the back end makes a prediction, the front end processes and displays the results. During data transmission, the front end continues recording the next audio segment, ensuring a seamless and low-latency experience. Users can view the results in two formats.

1. Class Label: The single class label inferred by an individual from the input speech signal (Figure 11).

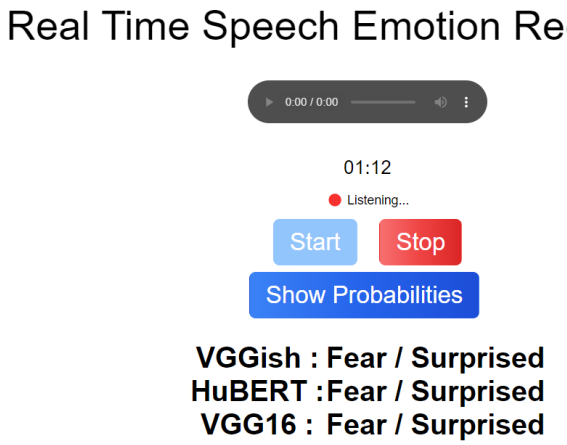


Figure 11: RTSER Front-end (Single Class Label Output)

2. Class Probability: Figure 12 shows the probability of each possible class for the previous utterance. The

chart changes dynamically with each request, without reactivating the page. It is implemented using Chart JS Da Rocha (2019).

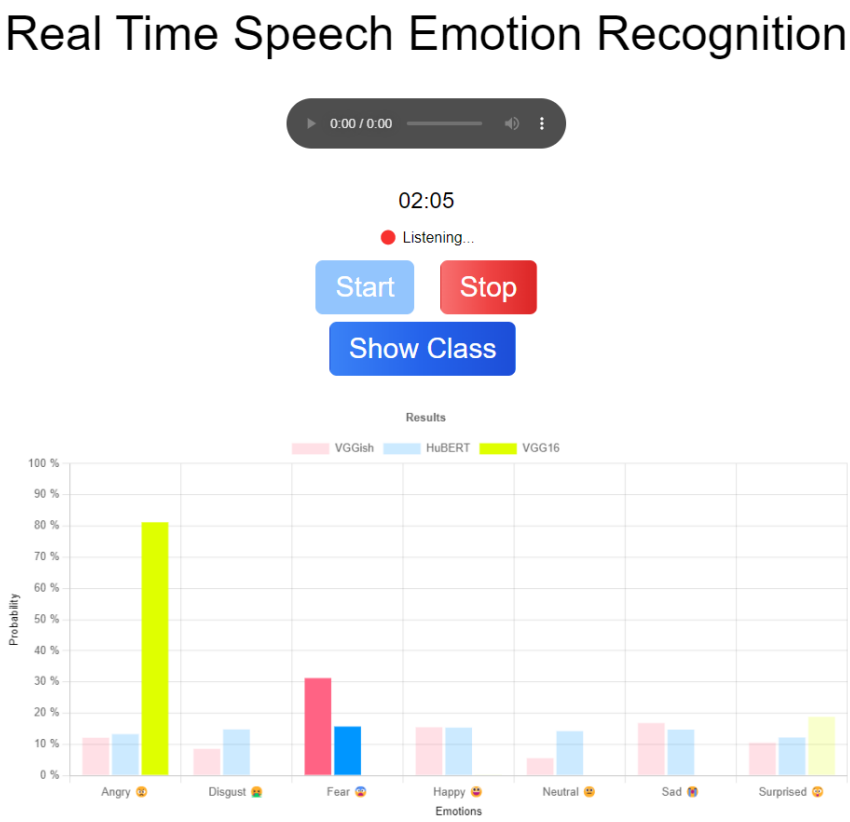


Figure 12: RTSER Front-end (Class Probability Output)

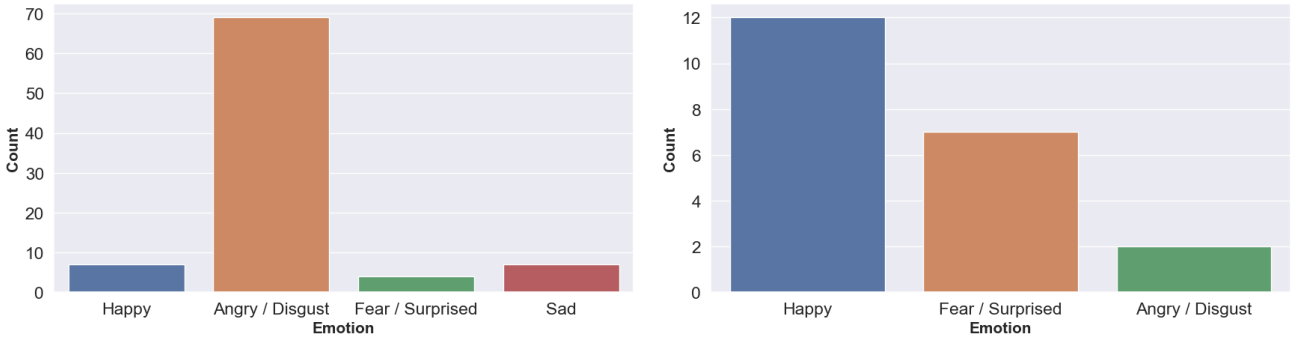
Users can easily switch between the two result display formats. After ending a session, they can listen to and save the complete recorded audio directly from the browser-based interface. This browser-based front end ensures portability and compatibility across various devices and operating systems.

- Back-end: The backend utilizes FlaskGrinberg (2018), a lightweight web framework in Python, known for its simplicity and ease of setup. Flask integrates seamlessly with our Python-based trained models. The backend accepts audio files from the frontend and checks for noise by calculating the audio's RMS energy. If the energy is below 0.01, the server classifies the audio as 'Neutral,' assuming silence. Otherwise, the audio is processed by the speech emotion recognition models. The results are formatted into JSON and sent back to the frontend for display.

3.5.2. Real-Time System Evaluation Criteria

We tested the RTSER system using two lengthy,unheard audio clip (see section 3.5.1). Each 2-second segment was manually assigned an emotional category. These clips were then input through the frontend's microphone3.5.1, and the resulting classifications were compared with the pre-assigned labels for evaluation.

The first footage [3] is from the movie Joker (2019), when the main protagonist executes a host of a talk show on live television. The clip is about 5 minutes long, and just the first 2 minutes and 22 seconds of audio are used for review. This extracted segment has approximately 500 words spoken in English. The video begins with light negative emotional tones, including melancholy, disgust, astonishment, and awkward humor. As the situation progresses, the discussion becomes angrier, culminating in violence and dread. Overall, this video conveys primarily unpleasant sentiments. Figure 13a shows the class distribution of this clip, which we manually rate.



(a) Class Distribution of First Clip

(b) Class Distribution of Second Clip

Figure 13: Class Distribution produced by evaluating two clips.

The following second clip[4] is taken from the Bengali movie 'Monpura' (2009), where two characters chat in a romantic way. In contrast to the first clip, this one contains positive emotional tones such as delight, amusement (positive surprise), and a minor degree of teasing (anger). The clip is about 2 minutes long and entirely in Bengali. Figure 13b shows the class distribution of this clip, which we manually rate.

4. Results Analysis and Discussions

4.1. Experimental Setup

4.1.1. Libraries Used

- Tensorflow: TensorFlow, developed by Google, is a powerful open-source framework for machine learning and AI, widely used in both business and academia. It's ideal for tasks like image classification, language translation, and time series prediction. As a comprehensive framework, TensorFlow facilitates building and training machine learning models with its efficient numerical computing engine for operations on multidimensional arrays. It is platform-independent, running on CPUs, GPUs, and TPUs, which makes it highly adaptable and performant, appealing to both developers and researchers.

  In our project, we utilized TensorFlow to create an MLP classification model. The framework also provides numerous pre-trained models that can be used directly or customized for specific datasets and tasks, significantly

[3] https://www.youtube.com/watch?v=WbliHNs4q14

[4] https://youtu.be/b9tQhSmQjVM?t=3247

reducing development time and cost. These models are particularly useful for image classification, object detection, and natural language processing. TensorFlow supports transfer learning, allowing the fine-tuning of models like VGG16, VGGish, and YAMNet to optimize performance on new tasks.

- PyTorch and Hugging Face: PyTorch, developed by Facebook's AI research team, is an open-source deep learning framework known for its flexibility, efficiency, and ease of use. It's particularly effective for building and training deep learning models. In contrast, Hugging Face specializes in tools and services for natural language processing (NLP) tasks such as language translation, text generation, and question-answering. Their open-source library provides numerous pre-trained models, simplifying their integration into various applications. In our project, we utilized Hugging Face's implementation of HuBERT and trained it using the PyTorch library.

- Librosa: Librosa is a Python toolkit designed for analyzing and transforming audio signals. It's lightweight and user-friendly, making it ideal for handling audio data. Commonly used in Music Information Retrieval (MIR), Librosa supports tasks like music classification, genre recognition, and artist identification. It is also widely applied in general audio signal processing, including voice recognition and audio categorization. We have utilized Librosa for the following tasks:

  i Reading and composing audio samples.
  ii Converting samples to the appropriate "wav" format.
  iii Resampling audio files to a sample rate of 16000kHz.
  iv Extracting audio features like MFCC.
  v Trimming audio samples and detecting silence.
  vi Use augmentation on the samples.
  vii Using spectrographs to visualize the audio samples.

4.1.2. Hardware and Software Information

Table 5 details the hardware and software setups used for training and evaluating all models, including the development and testing environments for the RTSER system.

Table 5: Hardware and Software Configuration

	Training	**RTSER System**
CPU	Intel Xeon (2 core)	Intel i5-8265U 1.60 GHz
GPU	Tesla P100-PCIE-16GB	×
RAM	16GB	8GB
OS	Linux-5.15.65	Windows 11 (Build 22621)
Python	3.7.12	3.10.6
TensorFlow	2.8.0	2.8.0
Hugging Face	4.25.1	4.25.1
PyTorch	19.04	19.04
Node	×	16.18.0

4.2. Results on Image Classification Models

In this study, we utilized six speech datasets: TESS Pichora-Fuller & Dupuis (2020), CREMA Cao et al. (2014), RAVDESS Livingstone & Russo (2018), SAVEE Jackson & Haq (2014), SUBESCO Sultana et al. (2021b), and BSER Das et al. (2022) alongside combinations of English, Bengali, and a mix of both languages, to train and evaluate our models. Our proposed model's performance was assessed using the F1 score and other accuracy metrics. For multi-class classification, accuracy alone is insufficient as an effective model should handle all classes with equal proficiency. Thus, the F1 score is employedAtsavasirilert et al. (2019), as it accounts for the recognition rate of each class. Unweighted Average Recall (UAR) is favored over Weighted Average Recall (WAR) for imbalanced datasets because it better represents the recognition rate for each class. The F1 score serves as a measure of the balance between precision and recall.

In our image transfer learning experiment, we leveraged two advanced models: VGG16 and ResNet50V2. We transformed the MFCC features of our speech data into compelling MFCC images, which were then input into both models. The results of our tests, highlighted in Tables 6 and 17, demonstrate the impressive capabilities of these models. Detailed parameters and outcomes for the VGG16 and ResNet50V2 models are presented in Tables 7 and 18.

The predictive performance of the VGG16 and ResNet50V2 models on the six datasets is detailed in Tables 6 and 17. VGG16 has been shown to outperform ResNet50V2 in emotion classification. Specifically, ResNet50V2's lowest accuracy was 17.69% on the BSER dataset, with its highest at 86.16% on TESS. In contrast, VGG16 achieved 61.09% on BSER and 89.99% on TESS. Additionally, VGG16 reached 59.38% accuracy on the English dataset, 66.65% on the Bangla dataset, and 65.71% on the combined English and Bangla datasets. ResNet50V2, on the other hand, recorded 57.33% on the combined English and Bangla datasets and 55.5% on the Bangla dataset.

4.3. Results on Audio Embedding Models

We utilized two models, VGGish and YAMNet, to generate audio embeddings for predicting emotions from speech signals. Initially, embeddings were created for all samples. These embeddings were then fed into a shallow MLP classification model to produce class labels. For hyperparameter tuning, we employed random grid search. The

Table 6: VGG16 Results

Dataset	Accuracy	F1 Weighted	Macro F1
TESS	89.89	89.94	90
RAVDESS	52.43	49.92	49
SAVEE	45.83	44.11	42
CREMA	53.33	53.59	45
SUBESCO	66.07	65.81	66
BSER	61.09	61.69	61
ALL EN	59.38	59.41	60
ALL BN	66.65	66.44	66
ALL	65.71	65.68	65

Table 7: VGG16 Classification Best Parameters

Dataset	Dropout Rate	Optimizer	Init. Learning Rate
TESS	None	Adam	0.0000100
RAVDESS	None	Adam	0.0000100
SAVEE	None	Adam	0.0000100
CREMA	None	Adam	0.0000100
SUBESCO	None	Adam	0.0000100
BSER	None	Adam	0.0000100
ALL EN	None	Adam	0.0000100
ALL BN	None	Adam	0.0000100
ALL	None	Adam	0.0000100

Table 8: VGG16 Classification Report on TESS Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	90.48	91.30	91.57	91
Disgust	81.96	92.23	86.62	90
Fear	92.84	91.17	92.29	89
Happy	95.69	92.97	94.88	91
Neutral	88.96	87.97	88.56	102
Sad	90.21	90.59	90.55	90
Surprise	95.50	86.61	90.99	90

Table 9: VGG16 Classification Report on RAVDESS Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	55.28	68.09	61.35	38
Disgust	65.53	58.63	61.75	38
Fear	63.49	58.90	60.05	38
Happy	36.37	38.40	37.36	39
Neutral	61.12	74.66	67.41	58
Sad	08.0	03.0	04.0	39
Surprise	46.57	58.09	51.37	38

test results for both VGGish and YAMNet across six individual datasets and three combined datasets are shown in Tables 28 and 30. The optimal parameters identified through random grid search are presented in Tables 29 and 31.

Table 10: VGG16 Classification Report on SAVEE Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	57.75	67.80	62.27	12
Disgust	67.10	33.13	44.74	12
Fear	40.98	17.60	24.58	12
Happy	38.22	25.58	30.97	12
Neutral	50.60	75.32	60.08	24
Sad	57.71	33.54	42.10	12
Surprise	25.69	42.10	31.46	12

Table 11: VGG16 Classification Report on CREMA Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	68.4	77.2	72.41	254
Disgust	56.88	44.61	49.23	254
Fear	45.33	34.2	39.19	254
Happy	49.62	47.0	48.37	255
Neutral	54.81	59.47	57.22	218
Sad	51.12	65.0	57.0	254

Table 12: VGG16 Classification Report on SUBESCO Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	73.20	78.56	75.0	200
Disgust	57.89	46.88	51.0	200
Fear	58.55	79.44	67.61	200
Happy	60.0	67.0	63.22	200
Neutral	82.35	76.0	79.0	200
Sad	66.0	54.44	59.32	200
Surprise	68.0	64.0	66.67	200

Table 13: VGG16 Classification Report on BSER Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	84.33	84.0	84	61
Happy	55.75	49.0	52.14	61
Neutral	69.0	55.21	61.0	49
Sad	60.12	68.0	64.48	62
Surprise	42.2	48.52	45.9	61

Table 14: VGG16 Classification Report on ALL EN Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	77.54	71.33	74	436
Disgust	47.6	67.43	55.75	385
Fear	56.12	43.22	48.77	385
Happy	54.62	50.88	52.69	436
Neutral	73.25	55.73	62.34	422
Sad	56.77	65.17	60.48	434
Surprise	63.0	70.0	67.0	188

Table 15: VGG16 Classification Report on ALL BN Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	83.56	72.0	77.0	261
Disgust	50.79	68.85	58.32	200
Fear	67.96	75.0	71.63	200
Happy	65.39	56.63	60.0	261
Neutral	68.64	85.33	75.0	249
Sad	75.86	48.22	58.0	261
Surprise	64.73	66.92	65.0	262

In Tables 28 and 30, it's evident that VGGish and YAMNet perform significantly worse at emotion detection compared to previous image classification models and the upcoming HuBERT model. There are two primary reasons for this: (1) The models' ability to generate embeddings suitable for emotion categorization is lacking, and (2) the classification MLP might be overfitting the generated embeddings. The initial analysis suggests that overfitting is a more likely issue, particularly with YAMNet embeddings, which perform worse than VGGish due to their longer vector length. Anticipating this, we used grid search to experiment with different layer numbers and sizes. However, the core issue seems to be that these models are not ideal for emotion recognition. Although one

Table 16: VGG15 Classification Report on ALL Dataset

Emotion	**Pre.**	**Re.**	**F1**	**Supp.**
Angry	73.2	78.55	75.3	646
Disgust	62.53	51.0	56.4	585
Fear	96.56	60.0	64.29	584
Happy	68.23	51.6	58.77	646
Neutral	69.43	76.45	72.34	628
Sad	56.28	74.26	64.35	646
Surprise	67.88	69.43	68.61	391

Table 17: ResNet50V2 Results

Dataset	**Accuracy**	**F1 Weighted**	**Macro F1**
TESS	86.16	86.14	86
RAVDESS	54.51	53.26	55
SAVEE	33.54	33.96	30
CREMA	46.07	45.33	45
SUBESCO	59.86	59.52	60
BSER	17.69	18.61	59
ALL EN	57.33	56.70	57
ALL BN	55.55	55.32	55
ALL	56.74	56.72	57

Table 18: ResNet50V2 Classification Best Parameters

Dataset	**Dropout Rate**	**Optimizer**	**Init. Learning Rate**
TESS	40	Adam	0.0000100
RAVDESS	40	Adam	0.0000100
SAVEE	40	Adam	0.0000100
CREMA	40	Adam	0.0000100
SUBESCO	40	Adam	0.0000100
BSER	40	Adam	0.0000100
ALL EN	40	Adam	0.0000100
ALL BN	40	Adam	0.0000100
ALL	40	Adam	0.0000100

might expect better performance since these models are designed to classify audio, they are primarily trained to identify environmental sounds Hershey et al. (2017), not nuances in human speech. Therefore, it's understandable that these models struggle to identify audio features crucial for predicting emotions.

Table 19: ResNet50V2 Classification Report on TESS Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	92.52	85.28	87.12	91
Disgust	90.83	81.72	85.57	90
Fear	90.52	85.61	88.12	89
Happy	86.37	93.19	89.77	91
Neutral	79.82	90.32	84.29	102
Sad	87.66	79.28	83.70	90
Surprise	83.24	89.47	86.36	90

Table 20: ResNet50V2 Classification Report on RAVDESS Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	67.57	63.32	65.85	38
Disgust	51.28	50.0	51.50	38
Fear	55.77	58.43	56.54	38
Happy	51.94	49.0	50.34	39
Neutral	59.43	75.83	66.53	58
Sad	44.31	21.5	28.43	39
Surprise	47.68	55.77	51.23	38

Table 21: ResNet50V2 Classification Report on SAVEE Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	50.43	17.4	25.0	12
Disgust	0	0	0	12
Fear	22.53	17.0	19.42	12
Happy	57.10	33.87	42.56	12
Neutral	46.31	88.0	60.0	24
Sad	36.0	33.5	35.0	12
Surprise	29.40	33.8	31.0	12

Table 22: VGG16 Classification Report on CREMA Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	67.37	68.63	67.48	254
Disgust	42.15	32.53	36.44	254
Fear	34.57	27.09	30.46	254
Happy	42.52	37.22	39.86	255
Neutral	41.73	53.0	46.0	218
Sad	47.80	61.48	53.58	254

Table 23: ResNet50V2 Classification Report on SUBESCO Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	64.76	67.19	66.97	200
Disgust	45.49	45.77	45.58	200
Fear	67.42	67.76	67.0	200
Happy	58.0	59.4	59.0	200
Neutral	61.5	79.63	69.33	200
Sad	68.60	47.61	55.43	200
Surprise	57.13	56.49	56.86	200

Table 24: VGG16 Classification Report on BSER Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	77.91	67.38	72.26	61
Happy	53.0	43.99	47.71	61
Neutral	62.85	73.33	67.10	49
Sad	62.47	53.48	57.38	53
Surprise	47.65	62.67	54.14	61

4.4. Results on HuBERT

We assessed the fine-tuned Base HuBERT model using both standard and augmented training sets across six individual datasets over five epochs. Table 32 displays the test outcomes. Due to slight imbalances in some datasets, we included a weighted F1 score alongside accuracy and macro F1 score for a comprehensive evaluation. The initial results were satisfactory, aligning with the state-of-the-art. However, significant improvements were observed with the inclusion of data augmentation, as detailed in Section 3.4. This process led to an average increase of 7.14% in the weighted F1 score across all datasets. Notably, the SAVEE and RAVDESS datasets showed an

Table 25: ResNet50V2 Classification Report on ALL EN Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	61.24	77.5	68	436
Disgust	46.18	52.48	49.53	385
Fear	57.73	35.23	43.33	385
Happy	54.78	49.65	52.34	436
Neutral	62.80	66.48	64.22	422
Sad	58.82	61.82	59.89	434
Surprise	68.65	59.39	63.11	188

Table 26: ReNet50V2 Classification Report on ALL BN Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	72.53	64.66	68.0	261
Disgust	44.0	56.40	49.16	200
Fear	69.85	52.17	59.0	200
Happy	47.43	64.0	54.36	261
Neutral	59.2	73.77	66.4	249
Sad	48.32	38.0	43.0	261
Surprise	58.66	42.12	48.12	262

Table 27: ResNet50V2 Classification Report on ALL Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	68.2	63.29	65.21	646
Disgust	46.33	49.84	47.21	585
Fear	61.69	46.23	53.63	584
Happy	54.0	56.46	55.85	646
Neutral	59.34	65.81	62.96	628
Sad	55.75	58.91	57.65	646
Surprise	55.12	58.3	56.23	391

18.60% and 11.88% boost in weighted F1 scores, respectively. The classification reports for individual classes (Tables 37 and 38) and the confusion matrix (Figure 28) for SAVEE indicate that augmentation improved the classification of Angry, Disgust, Happy, and Sad emotions. However, some misclassification persists, particularly between Fear and Surprise, which are somewhat similar to Happy. A similar pattern is observed in RAVDESS (Tables 35 and 36, Figure 27), where Fear, Happy, and Sad emotions continue to challenge perfect accuracy.

After noting the minimal improvements from augmentation in individual datasets, we opted to use augmented training only for the consolidated datasets, which have larger sample sizes. Training both unaugmented and

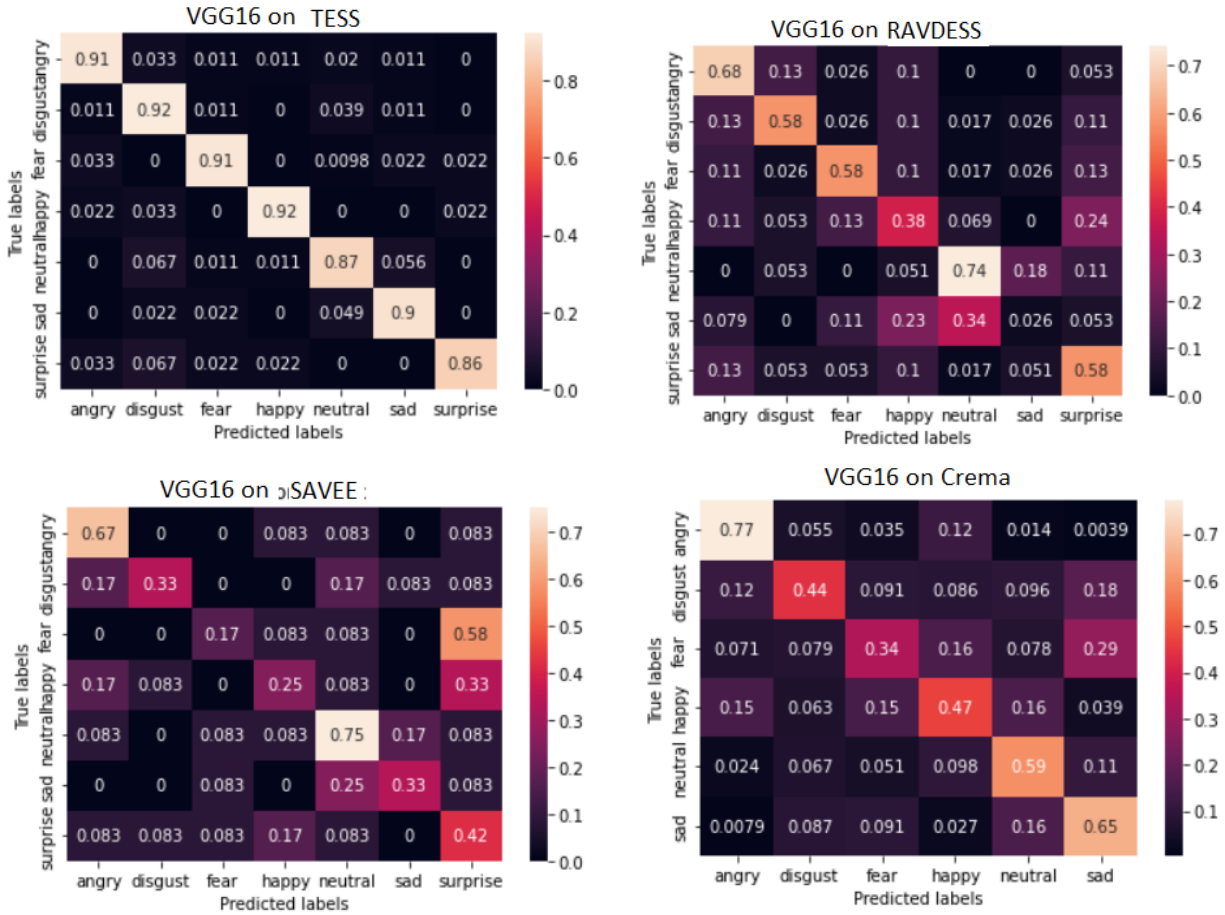


Figure 14: VGG16 Confusion Matrix (CM) on English Datasets

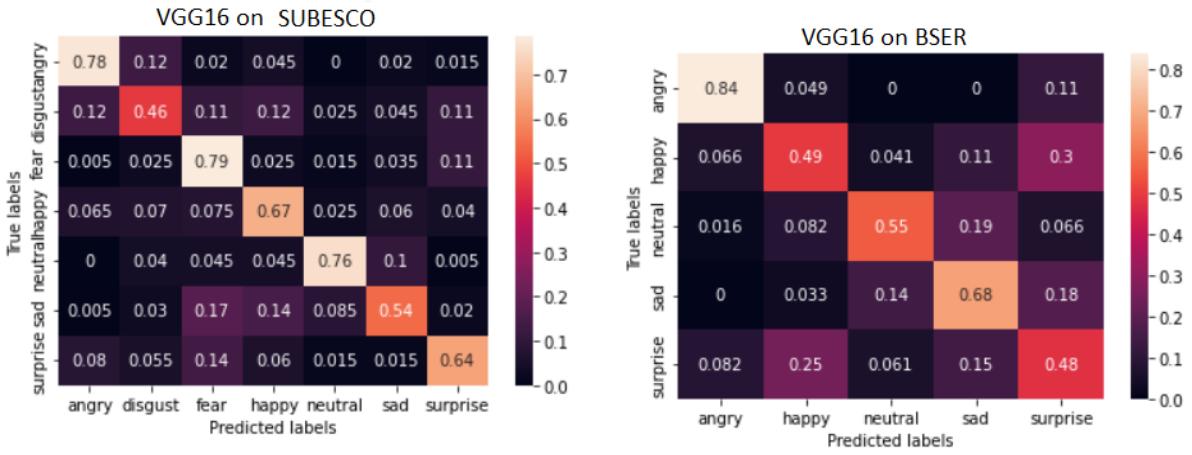


Figure 15: VGG16 Confusion Matrix (CM) on Bengali Datasets

augmented data would be too time-consuming. On the combined English and Bengali datasets, we achieved weighted F1 scores of 81.06% and 94.09%, respectively. Despite language differences and varying accents, we obtained a weighted F1 score of 86.52% on the combined dataset of all six English and Bengali datasets.

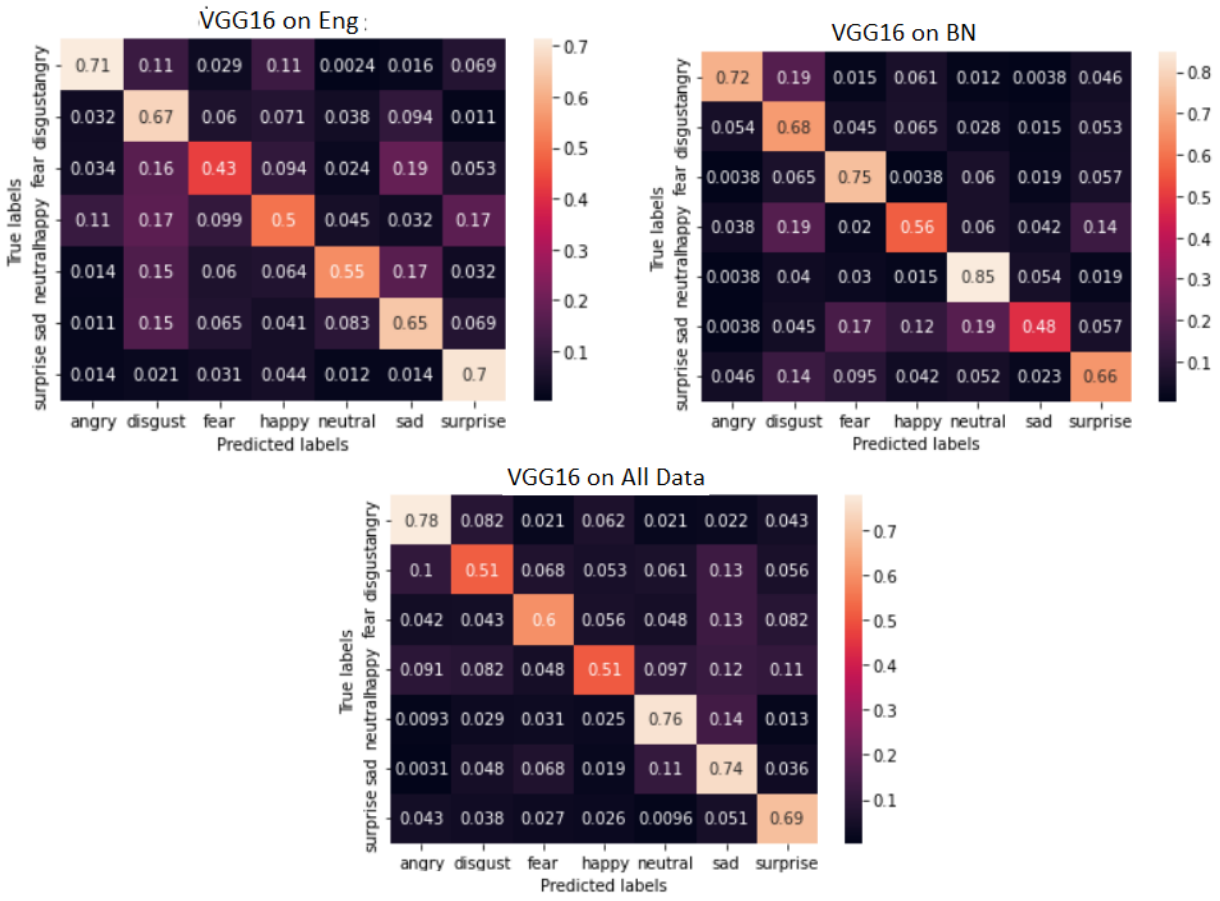


Figure 16: VGG16 Confusion Matrix (CM) on Consolidated Datasets

Table 28: VGGish+MLP Results

Dataset	Accuracy	F1 Weighted	Macro F1
TESS	94.18	94.16	94.04
RAVDESS	50.55	49.8	47.45
SAVEE	46.03	46.35	41.01
CREMA	40.05	40.01	40.71
SUBESCO	54.18	54.13	53.91
BSER	65.35	65.37	66.9
ALL EN	50.24	50.28	51.97
ALL BN	52.78	52.73	52.5
ALL	50.87	50.61	50.88

4.5. Comparative Analysis with Discussion

4.5.1. Comparison of Fine-Tuned Models

In this study, we examined models from three distinct domains for transfer learning, totaling five models. After fine-tuning these models, we subjected them to a rigorous testing process. The detailed results of their performance in

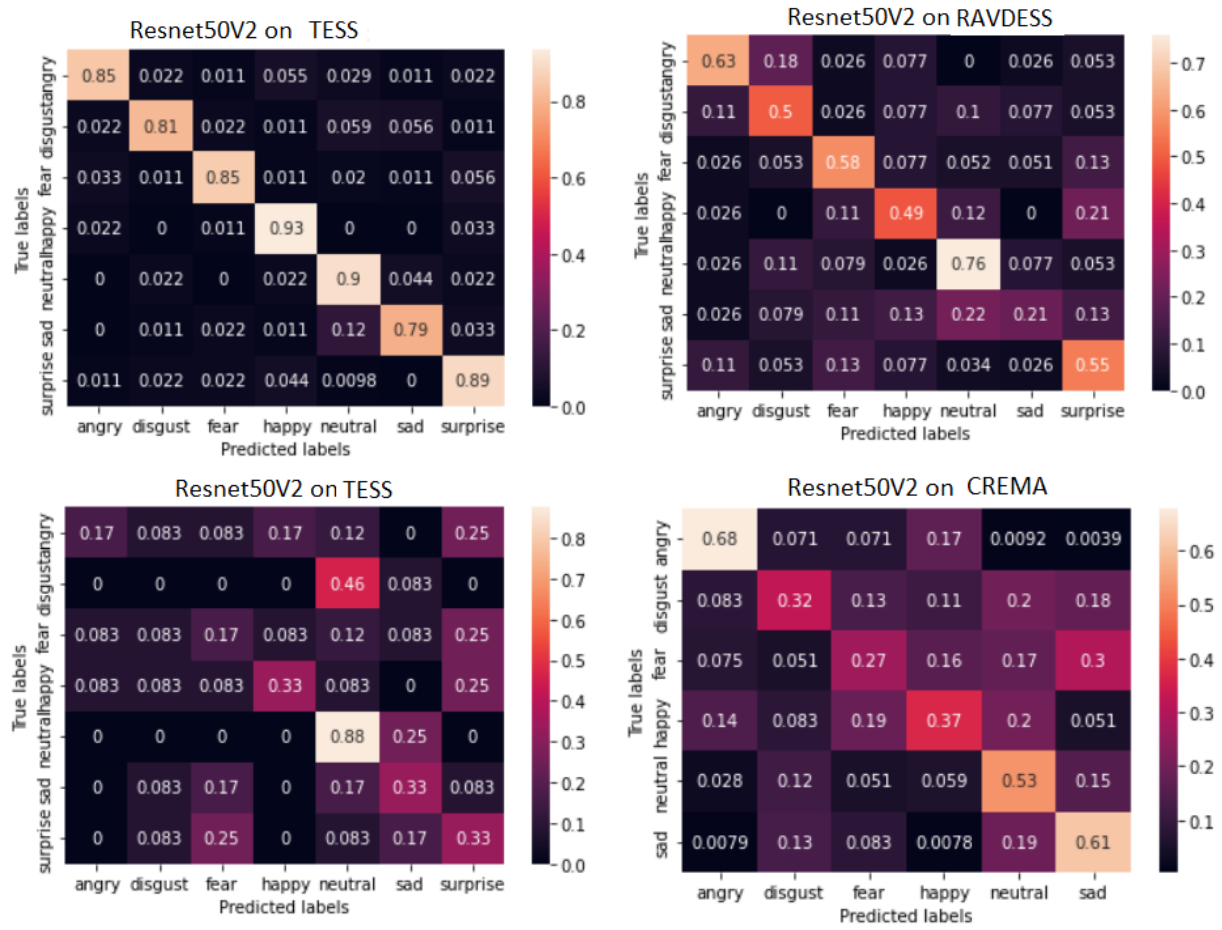


Figure 17: ResNet50V2 Confusion Matrix (CM) on English Datasets

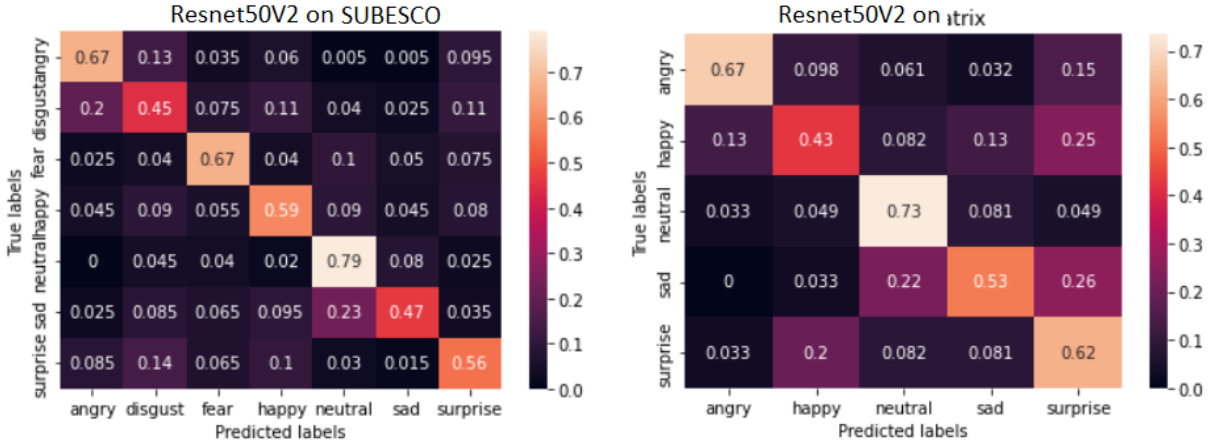


Figure 18: ResNet50V2 Confusion Matrix (CM) on Bengali Datasets

classifying individual samples are already discussed. Here, we compare and analyze the performance of these models. Figure 33 shows the classification accuracy of all five models across six datasets and their combinations. Notably, HuBERT outperformed the other four models. Except on SAVEE and CREMA-D, the fine-tuned HuBERT achieved over 80% accuracy on four individual datasets and three combined datasets. Conversely, the image classification

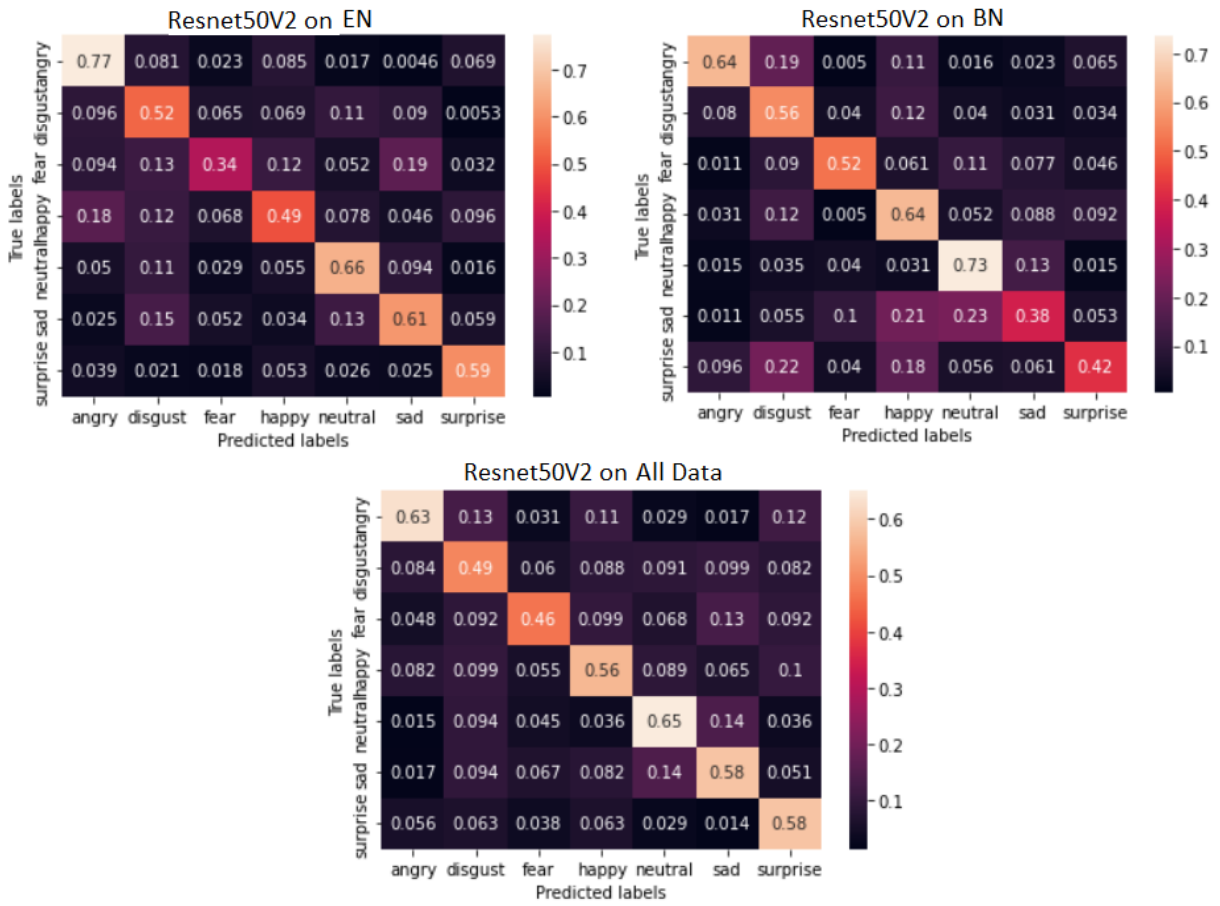


Figure 19: ResNet50V2 Confusion Matrix (CM) on Consolidated Datasets

Table 29: Classification MLP's Best Parameters with VGGish Embeddings

Dataset	Hidden Layer Size	Dropout Rate	Optimizer	Init. Learning Rate
TESS	1024, 512	20.0	Adam	0.000100
RAVDESS	512, 256	20.0	Adamax	0.000100
SAVEE	2048, 1024	0.0	Adamax	0.000100
CREMA	1024	0.0	Adam	0.000100
SUBESCO	512	0.0	Adam	0.001000
BSER	2048, 1024	20.0	Adam	0.000100
ALL EN	512	50.0	Adamax	0.000100
ALL BN	1024, 512	20.0	Adamax	0.000100
ALL	1024, 512	50.0	Adam	0.000100

models, VGG16 and ResNetV2, struggled to reach 60% accuracy. The audio embedding models, VGGish and YAMNet, performed slightly better than the image models but still lagged behind HuBERT.

HuBERT's superior performance can be attributed to its pre-training on extensive human speech data, specifically Librispeech 960 Panayotov et al. (2015), for ASR tasks. In contrast, the other models were pre-trained on different

Table 30: YAMNet+MLP Results

Dataset	Accuracy	F1 Weighted	Macro F1
TESS	78.72	78.81	78.36
RAVDESS	40.47	39.74	37.6
SAVEE	36.03	36.32	31.12
CREMA	34.94	33.59	33.93
SUBESCO	40.02	39.08	39.05
BSER	48.86	48.48	49.78
ALL EN	41.2	40.83	42.04
ALL BN	39.92	38.96	39.04
ALL	40.27	39.63	39.68

Table 31: Classification MLP's Best Parameters with YAMNet Embeddings

Dataset	Hidden Layer Size	Dropout Rate	Optimizer	Init. Learning Rate
TESS	2048	50.0	Adamax	0.001000
RAVDESS	2048	20.0	Adamax	0.000010
SAVEE	1024	50.0	Adam	0.001000
CREMA	4096	0.0	Adamax	0.000010
SUBESCO	4096, 2048	50.0	Adamax	0.000010
BSER	4096	0.0	Adamax	0.000010
ALL EN	4096	0.0	Adamax	0.000100
ALL BN	4096, 2048	20.0	Adamax	0.000100
ALL	4096, 2048, 1024	20.0	Adamax	0.001000

types of data. The image recognition models used the ImageNet Deng et al. (2009) database, which is significantly different from audio data, making transfer learning less effective. Although VGGish and YAMNet are audio-based models, they still performed similarly to the image models. This raises the question of why they didn't achieve similar success to HuBERT. To understand this, we need to consider the nature of their pre-training data. VGGish and YAMNet were trained on the Youtube-8M Abu-El-Haija et al. (2016) database, which includes not only human speech but also a variety of environmental sounds, musical instruments, and animal noises. In contrast, Librispeech 960 focuses solely on human speech. Therefore, HuBERT's superior performance is likely due to its pre-training on data directly relevant to speech emotion recognition.

4.5.2. Comparision of the models on RTSER system

To evaluate our RTSER system, we selected the top model from each category: HuBERT, VGGish, and VGG16. We used versions of these models that were trained on a combination of all six datasets. These models were tested on the two clips. We then compared the models' predictions with the true labels, excluding any 'Neutral' classifications or silent segments. These tests were crafted to simulate real-world conditions and evaluate the models' effectiveness

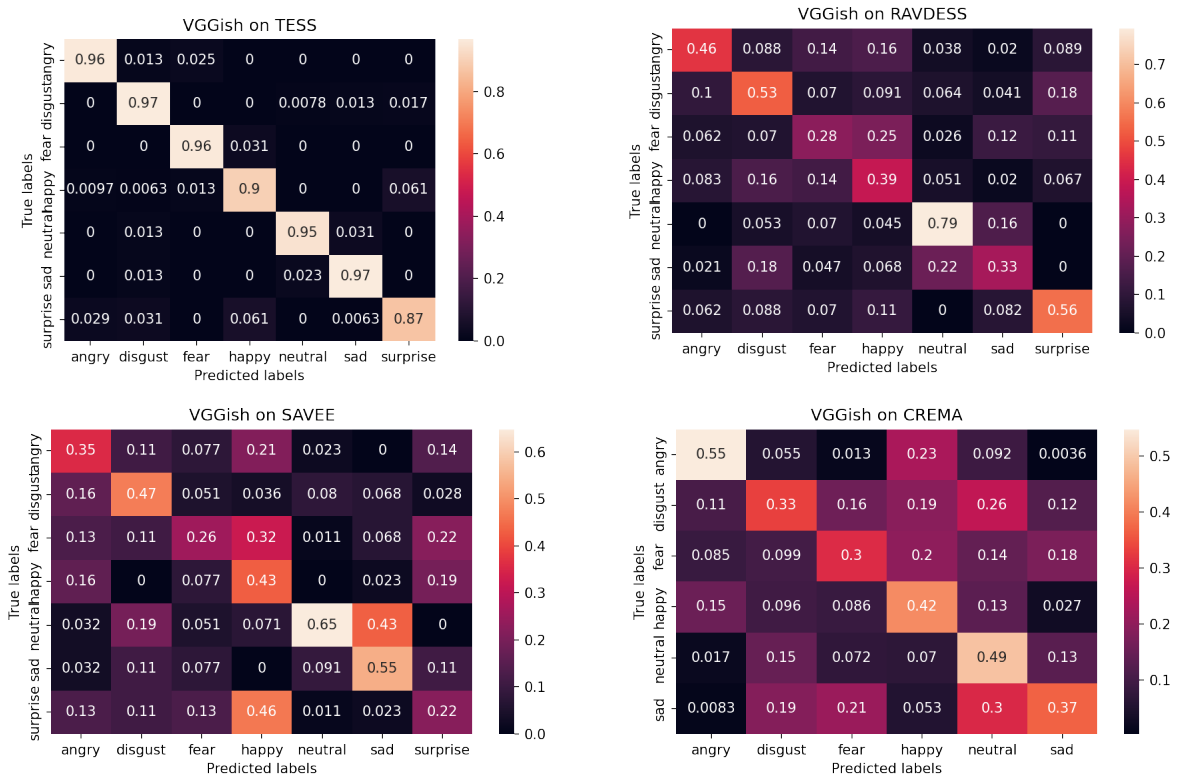


Figure 20: VGGish Confusion Matrix (CM) on English Datasets

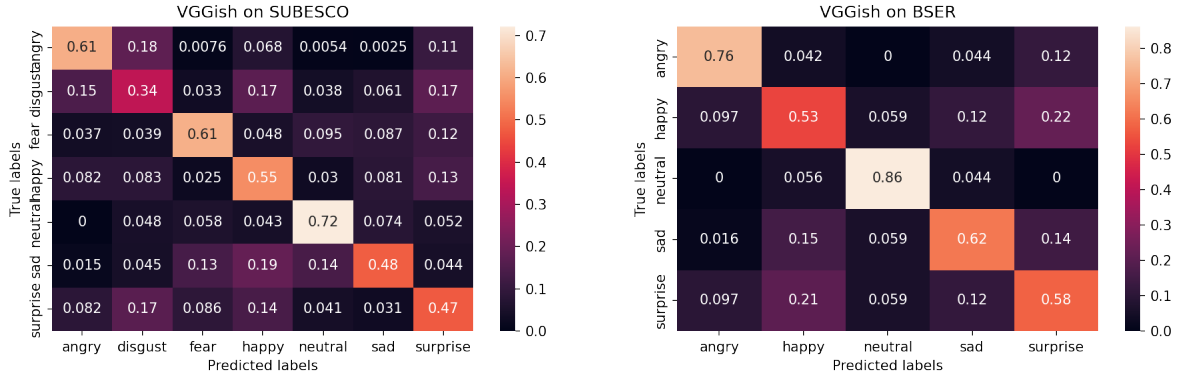


Figure 21: VGGish Confusion Matrix (CM) on Bengali Datasets

in recognizing emotions in real-time scenarios.

Table 48 clearly demonstrates that our real-time testing yielded positive results, particularly for HuBERT in conversational contexts. The model achieved an overall accuracy of 52.87% for the English clip and 47.62% for the Bengali clip. This indicates that our fine-tuned Bilingual HuBERT can consistently and accurately identify emotions in real-time. However, the same success was not observed for the VGG16 and VGGish models, suggesting that they are not yet ready for real-time deployment and require further improvement. Despite this, HuBERT's performance confirms that real-time emotion recognition in conversations is feasible, even with training on small, isolated datasets.

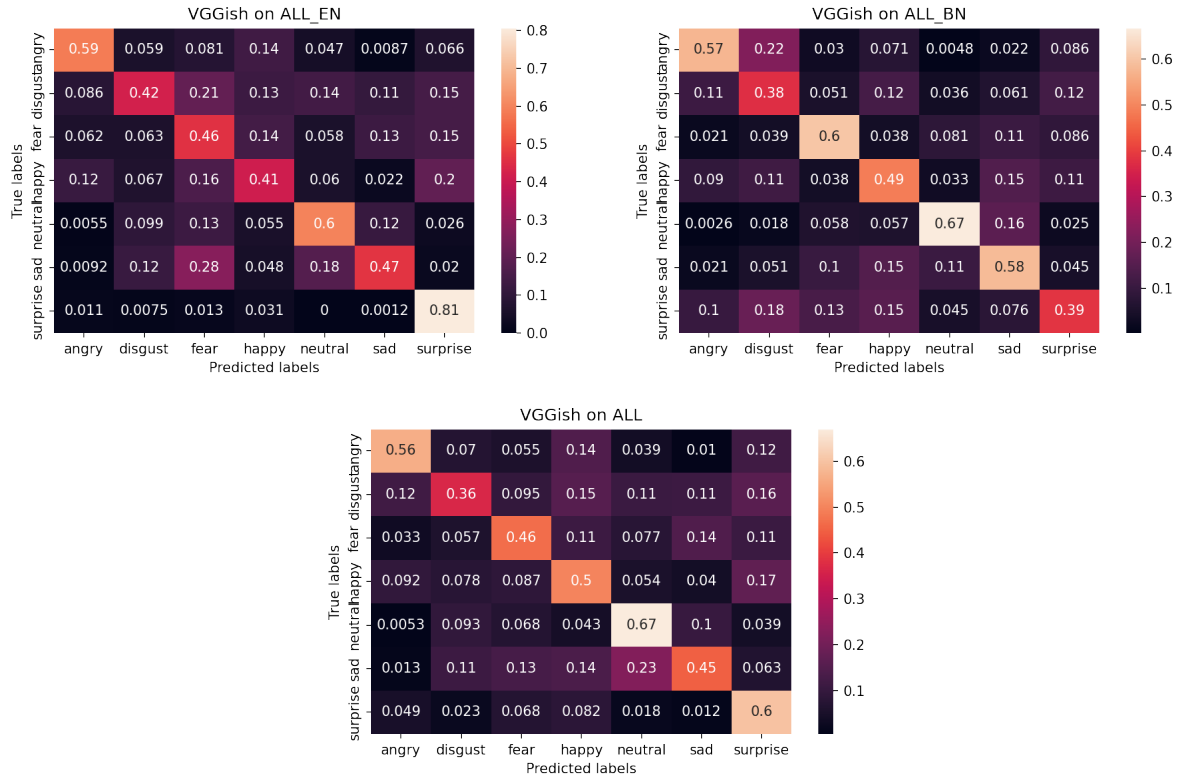


Figure 22: VGGish Confusion Matrix (CM) on Consolidated Datasets

In Figure 34, the line chart shows the response rate of the RTSER system. The average response time was 490.76ms while hosting and serving all three models simultaneously (see Table 48). The initial latency spike was due to library initialization. Once loaded into memory, the latency stabilized. The smaller spikes observed later are likely due to garbage collection Ismail & Suh (2018), which can be mitigated by allocating more threads and memory to the Flask server. Despite these occasional issues, the system's latency remains below the threshold that would affect real-time user experience. On a more powerful machine, inference times could be significantly lower, and latency can be further reduced by hosting a single model instead of three.

4.5.3. Comparison with existing state-of-art Works

Research on detecting emotions from speech has explored various techniques and methodologies. Here, we will compare the performance of our models with previous studies discussed in the Literature Review section2.

In comparison to other recent studies, our model, particularly the fine-tuned HuBERT, has achieved remarkable results in voice emotion recognition. To enhance our model's performance, we employed advanced techniques, including a transformer-based architecture and effective data augmentations. As a result, our fine-tuned HuBERT reached accuracies of 99.64%, 88.54%, and 72.53% on the TESS, RAVDESS, and CREMA-D datasets, respectively. These surpass the existing accuracies of 96.10%, 86.90%, and 71.69% (see Tables 49, 50, and 52). However, our other four fine-tuned models (VGG16, ResNet, VGGish, YAMNet) did not perform as expected. They only achieved satisfactory results on the TESS dataset and underperformed on the others. Although we hoped our approach would significantly advance speech emotion recognition, further improvements are needed for these models.

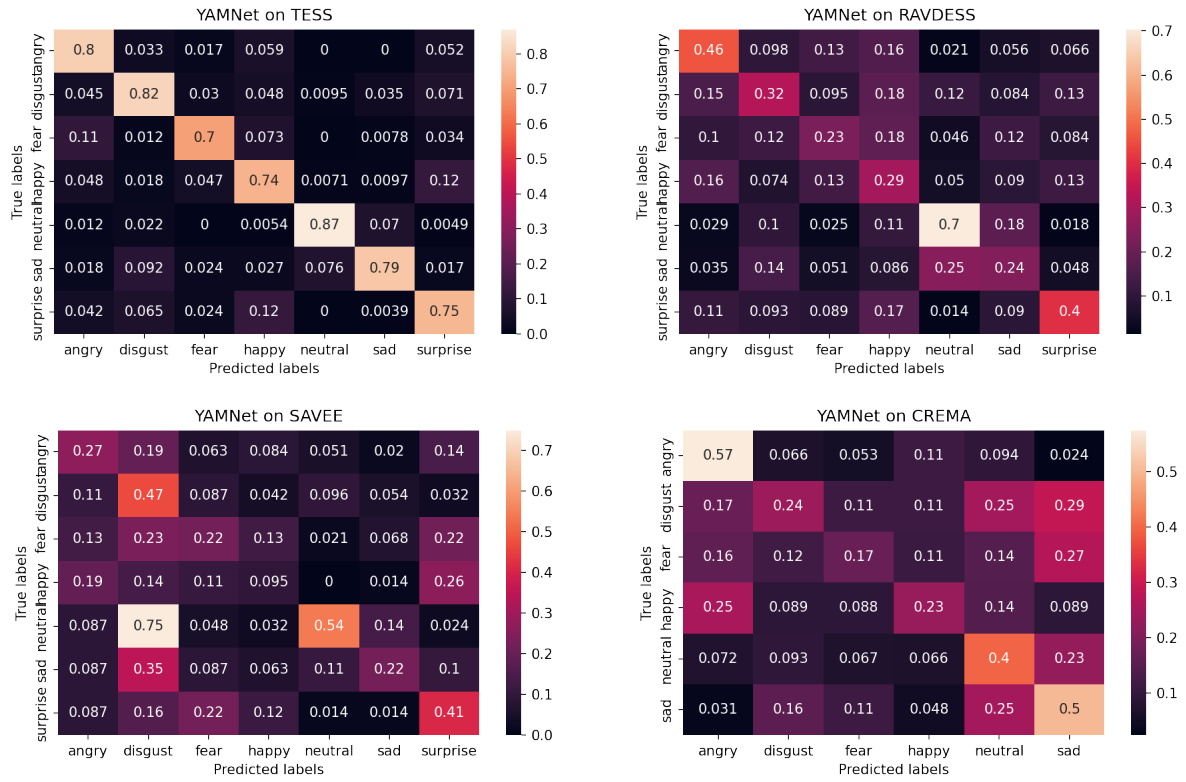


Figure 23: YAMNet Confusion Matrix (CM) on English Datasets

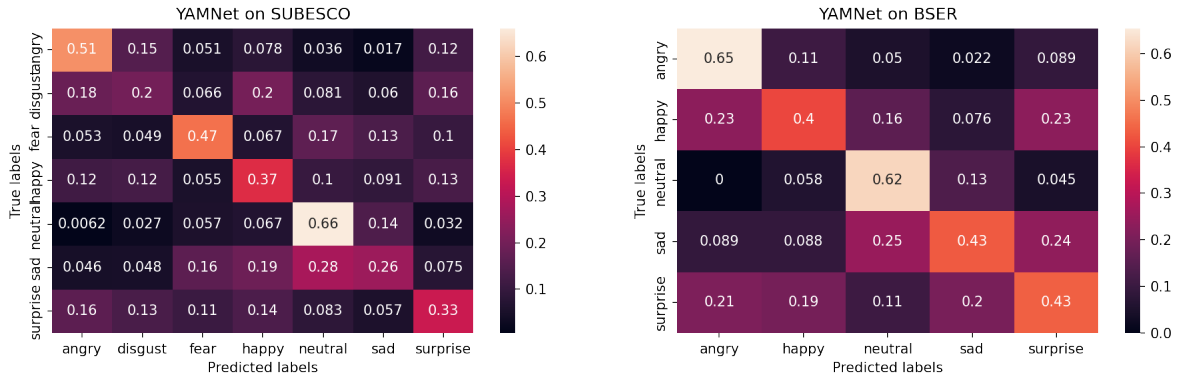


Figure 24: YAMNet Confusion Matrix (CM) on Bengali Datasets

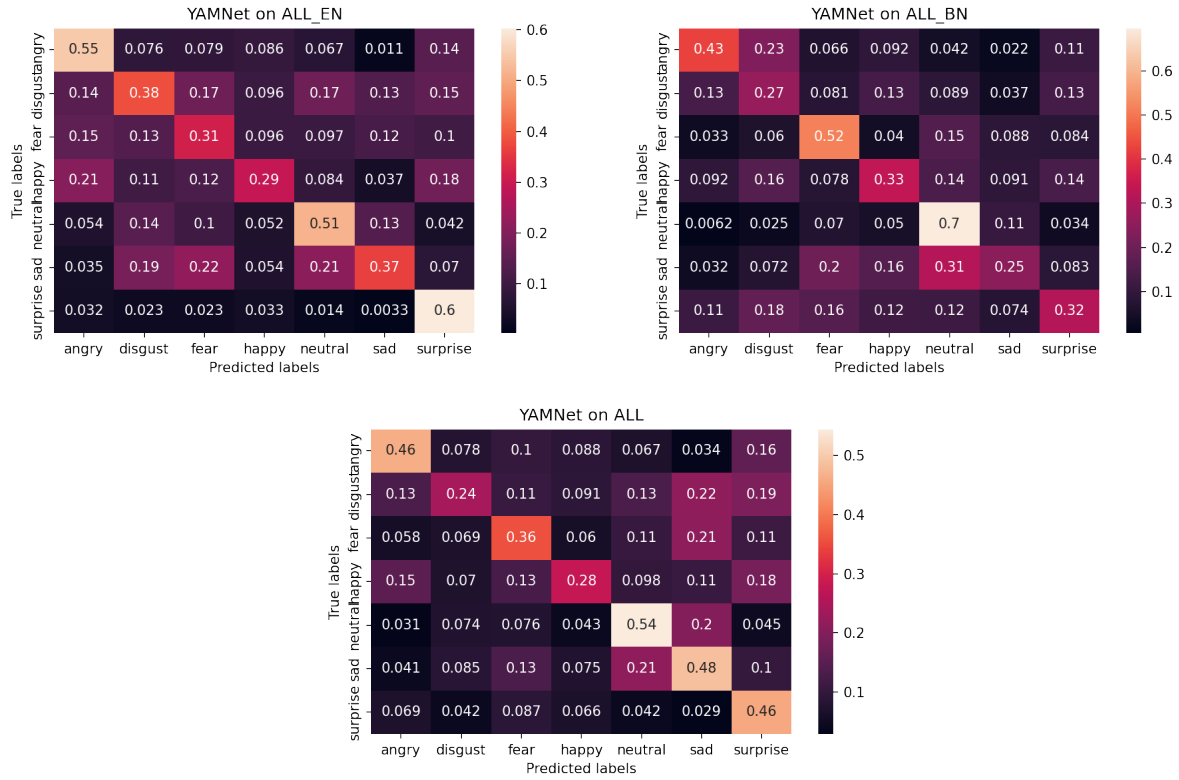


Figure 25: YAMNet Confusion Matrix (CM) on Consolidated Datasets

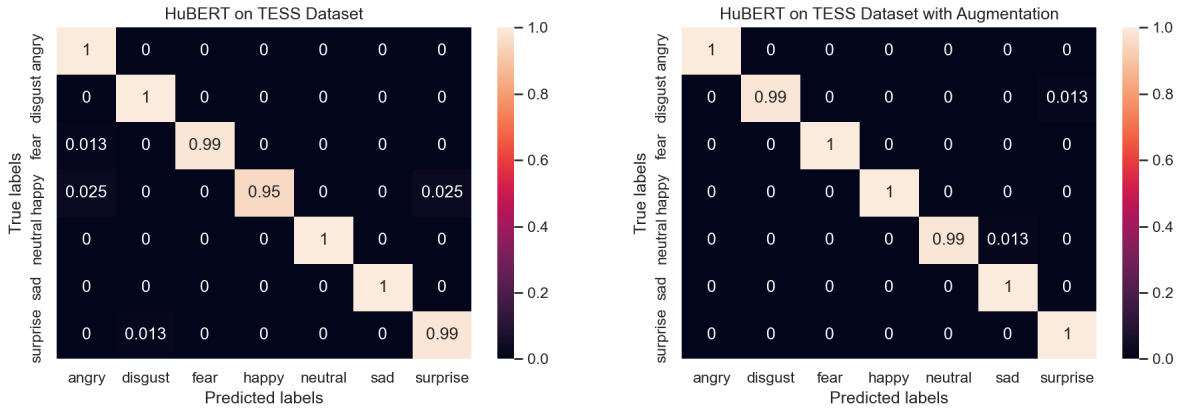


Figure 26: HuBERT Confusion Matrix (CM) on TESS

In Table 51, our HuBERT model slightly underperformed compared to existing work, showing an accuracy 5.49% lower than Qayyum et al.'s 1D CNN Qayyum et al.. This may be due to overfitting on the SAVEE dataset, which is the smallest in our study with only 480 samples. The base HuBERT model, with its 90 million parameters, struggles to generalize as effectively as simpler CNN models. Although we attempted to mitigate this with data augmentation, further research is needed to address overfitting on this dataset. Conversely, when evaluated on four combined English datasets, HuBERT achieved an accuracy of 81.18%, significantly surpassing the previous high of 57.42% (Table 53). Transformer-based models often perform better with larger datasets than CNNs, as shown in prior research.

Table 32: HuBERT Accuracy, Weighted F1, Macro F1 with and without Augmentation

Dataset	Acc.	Acc. with Aug.	WF1	WF1 with Aug.	MF1	MF1 with Aug.
TESS	98.93	99.64	98.93	99.64	98.93	99.64
RAVDESS	77.08	88.54	76.65	88.53	75.27	89.2
SAVEE	60.42	78.12	59.67	78.27	55.31	75.07
CREMA-D	69.91	72.53	69.74	72.51	69.86	72.66
SUBESCO	91.14	95.0	91.22	94.99	91.22	94.99
BSER	82.31	87.41	82.33	87.41	82.26	87.47
ALL EN	×	81.18	×	81.06	×	82.38
ALL BN	×	94.1	×	94.09	×	94.1
ALL	×	86.51	×	86.52	×	86.42

Table 33: HuBERT Classification Report on TESS Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	96.39	100.0	98.16	80
Disgust	98.77	100.0	99.38	80
Fear	100.0	98.75	99.37	80
Happy	100.0	95.0	97.44	80
Neutral	100.0	100.0	100.0	80
Sad	100.0	100.0	100.0	80
Surprise	97.53	98.75	98.14	80

Table 34: HuBERT Classification Report on TESS Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	100.0	100.0	100.0	80
Disgust	100.0	98.75	99.37	80
Fear	100.0	100.0	100.0	80
Happy	100.0	100.0	100.0	80
Neutral	100.0	98.75	99.37	80
Sad	98.77	100.0	99.38	80
Surprise	98.77	100.0	99.38	80

Table 35: HuBERT Classification Report on RAVDESS Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	88.57	81.58	84.93	38
Disgust	85.0	89.47	87.18	38
Fear	68.75	57.89	62.86	38
Happy	72.73	61.54	66.67	39
Neutral	84.13	91.38	87.6	58
Sad	46.43	68.42	55.32	38
Surprise	96.55	71.79	82.35	39

Table 36: HuBERT Classification Report on RAVDESS Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	94.87	97.37	96.1	38
Disgust	94.87	97.37	96.1	38
Fear	71.11	84.21	77.11	38
Happy	100.0	74.36	85.29	39
Neutral	90.48	98.28	94.21	58
Sad	82.86	76.32	79.45	38
Surprise	97.37	94.87	96.1	39

Table 54 compares our transfer-learned models with others on the SUBESCO, BSER, and Consolidated Bengali datasets. As these datasets are relatively new, there is limited research available. Our fine-tuned HuBERT model performs comparably to existing models, except on the BSER dataset. Analyzing the misclassified samples, we found that low-intensity emotional expressions are often difficult to interpret, even for native Bengali speakers.

Table 37: HuBERT Classification Report on SAVEE Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	71.43	41.67	52.63	12
Disgust	50.0	41.67	45.45	12
Fear	42.86	50.0	46.15	12
Happy	45.45	41.67	43.48	12
Neutral	85.19	95.83	90.2	24
Sad	64.29	75.0	69.23	12
Surprise	38.46	41.67	40.0	12

Table 38: HuBERT Classification Report on SAVEE Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	91.67	91.67	91.67	12
Disgust	76.92	83.33	80.0	12
Fear	75.0	50.0	60.0	12
Happy	52.63	83.33	64.52	12
Neutral	92.31	100.0	96.0	24
Sad	100.0	75.0	85.71	12
Surprise	55.56	41.67	47.62	12

Table 39: HuBERT Classification Report on CREMA Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	82.3	73.23	77.5	254
Disgust	67.98	67.72	67.85	254
Fear	66.4	66.14	66.27	254
Happy	68.18	76.47	72.09	255
Neutral	68.7	82.57	75.0	218
Sad	66.99	55.12	60.48	254

Table 40: HuBERT Classification Report on CREMA Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	76.34	83.86	79.92	254
Disgust	71.01	66.54	68.7	254
Fear	68.24	62.6	65.3	254
Happy	81.2	74.51	77.71	255
Neutral	78.64	79.36	79.0	218
Sad	61.75	69.29	65.31	254

Table 41: HuBERT Classification Report on SUBESCO Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	91.87	96.0	93.89	200
Disgust	90.82	89.0	89.9	200
Fear	96.46	95.5	95.98	200
Happy	97.45	95.5	96.46	200
Neutral	98.51	99.5	99.0	200
Sad	95.59	97.5	96.53	200
Surprise	94.36	92.0	93.16	200

Table 42: HuBERT Classification Report on SUBESCO Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	91.87	96.0	93.89	200
Disgust	90.82	89.0	89.9	200
Fear	96.46	95.5	95.98	200
Happy	97.45	95.5	96.46	200
Neutral	98.51	99.5	99.0	200
Sad	95.59	97.5	96.53	200
Surprise	94.36	92.0	93.16	200

Additionally, some samples have poor audio quality due to substandard microphones, contributing to lower performance compared to PBGL Chakraborty et al. (2022) on the BSER dataset. It's important to note that the authors used a different data splitting strategy and feature engineering process, incorporating more than just MFCCs, as detailed in the Literature Review section2. Ultimately, on the combined Bengali dataset, our fine-tuned HuBERT model's accuracy is very close to that of PBLG.

Table 43: HuBERT Classification Report on BSER Dataset

Emotion	Pre.	Re.	F1	Supp.
Angry	86.89	85.48	86.18	62
Happy	78.79	85.25	81.89	61
Neutral	84.44	77.55	80.85	49
Sad	75.81	77.05	76.42	61
Surprise	86.67	85.25	85.95	61

Table 44: HuBERT Classification Report on BSER Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	87.3	88.71	88.0	62
Happy	85.25	85.25	85.25	61
Neutral	88.0	89.8	88.89	49
Sad	88.33	86.89	87.6	61
Surprise	88.33	86.89	87.6	61

Table 45: HuBERT Classification Report on ALL EN Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	82.52	92.19	87.08	384
Disgust	86.22	76.56	81.1	384
Fear	71.47	71.09	71.28	384
Happy	82.15	81.09	81.62	386
Neutral	85.93	91.58	88.66	380
Sad	73.44	70.57	71.98	384
Surprise	96.83	93.13	94.94	131

Table 46: HuBERT Classification Report on ALL BN Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	94.49	91.6	93.02	262
Disgust	90.2	92.0	91.09	200
Fear	97.94	95.0	96.45	200
Happy	92.4	93.1	92.75	261
Neutral	97.24	99.2	98.21	249
Sad	93.68	96.55	95.09	261
Surprise	92.97	91.19	92.07	261

Table 47: HuBERT Classification Report on ALL Dataset with Augmentation

Emotion	Pre.	Re.	F1	Supp.
Angry	83.0	87.0	85.0	646
Disgust	80.0	83.0	82.0	584
Fear	83.0	75.0	79.0	584
Happy	91.0	79.0	85.0	647
Neutral	83.0	95.0	89.0	629
Sad	80.0	80.0	80.0	645
Surprise	88.0	91.0	89.0	392

Figure 27: HuBERT Confusion Matrix (CM) on RAVDESS

Figure 28: HuBERT Confusion Matrix (CM) on SAVEE

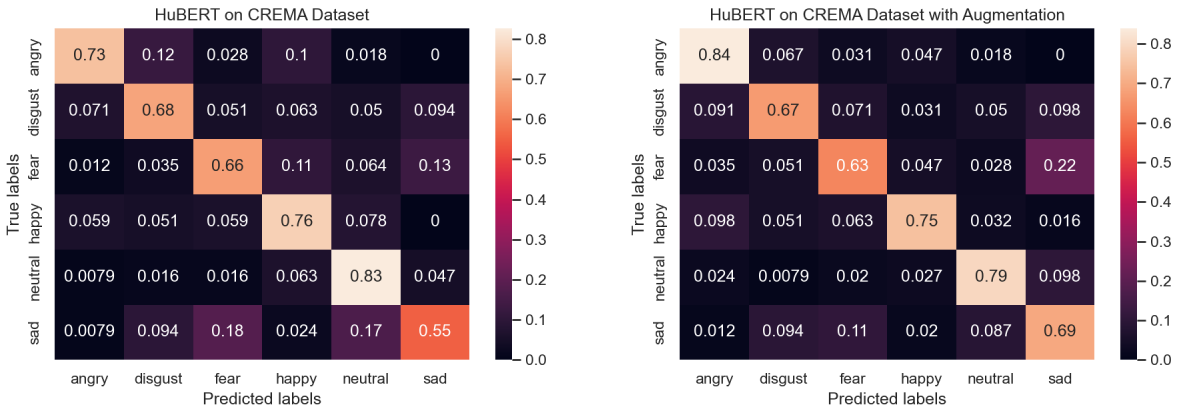


Figure 29: HuBERT Confusion Matrix (CM) on CREMA-D

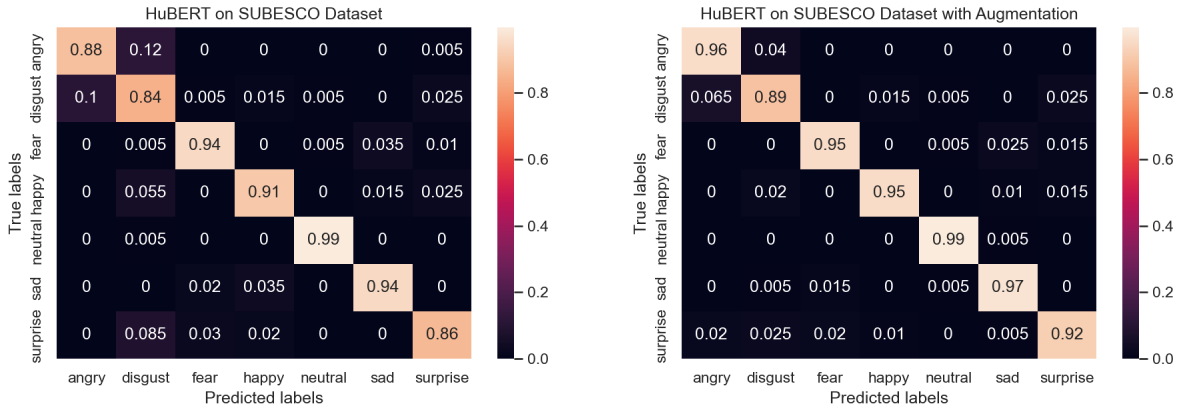


Figure 30: HuBERT Confusion Matrix (CM) on SUBESCO

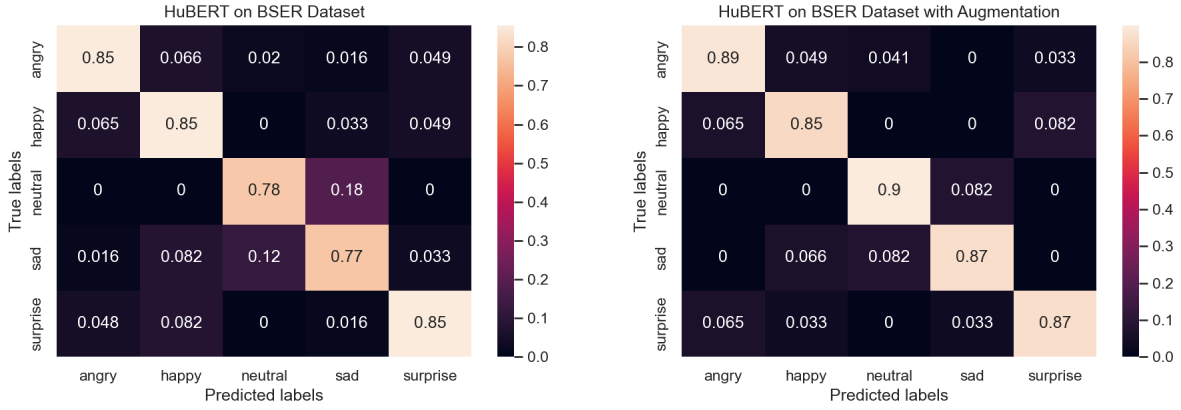


Figure 31: HuBERT Confusion Matrix (CM) on BSER

Table 48: RTSER Results

Model	**Clip No.**	**Acc**	**WF1**	**MF1**
HuBERT	1	52.87	58.77	34.22
	2	47.62	46.47	26.74
VGGish	1	21.84	25.90	15.60
	2	19.05	16.88	24.56
VGG16	1	9.09	6.36	7.62
	2	39.47	21.27	28.36

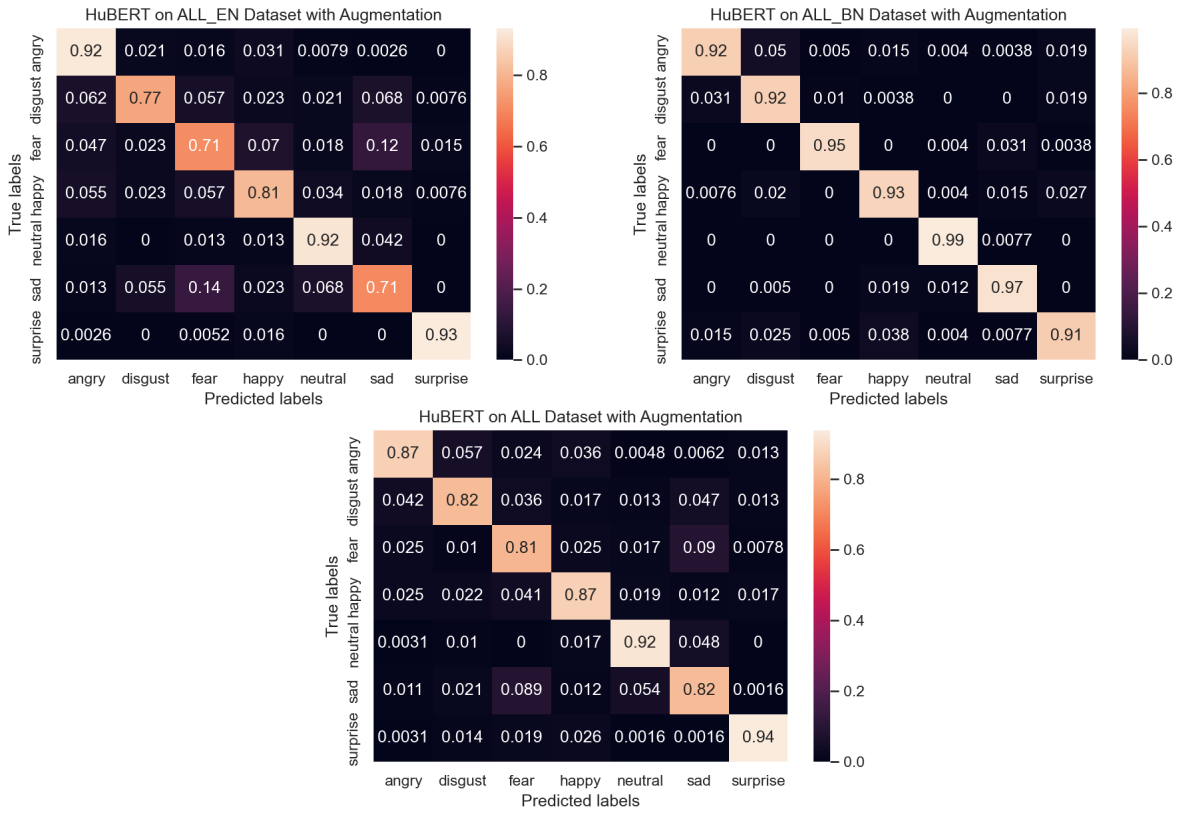


Figure 32: HuBERT Confusion Matrix (CM) on Consolidated Datasets

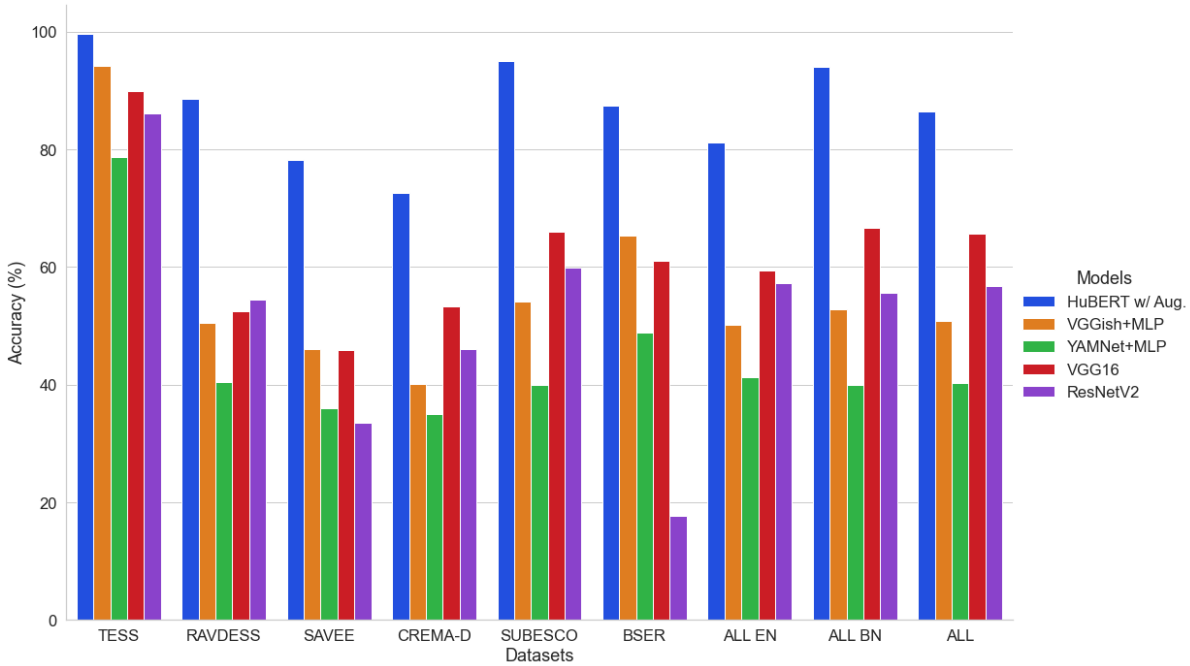


Figure 33: Comparison of classification accuracy among fine-tuned Models

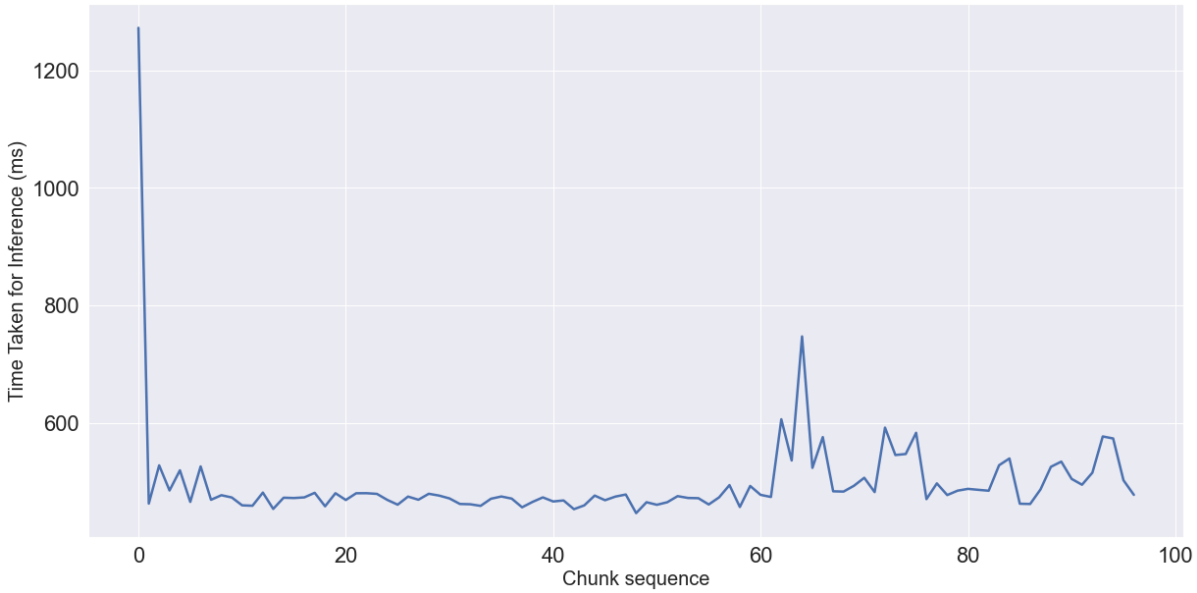


Figure 34: Response rate results of RTSER system on the first clip

Table 49: Comparison among the existing state-of-art works on TESS

Work Ref.	**Classification Model**	**Accuracy**
Guizzo et al. (2020)	CNN w/ MTS Aug.	53.05
Krishnan et al. (2021)	IMF-SVM, KNN	93.30
Blumentals & Salimbajevs (2022)	LSTM-FCNN	86.02
Patel et al. (2022)	Autoencoder + 1D CNN	96.00
Akinpelu & Viriri (2022)	DCNN-NCA-MLP Aug.	96.10
VGG16 (Ours)	VGG16	89.89
ResNetV2 (Ours)	ResNetV2	86.61
VGGish (Ours)	VGGish+MLP	94.18
YAMNet (Ours)	YAMNet+MLP	78.72
HuBERT (Ours)	HuBERT Base 960h	**99.64**

Table 50: Comparison among the existing state-of-art works on RAVDESS

Work Ref.	**Classification Model**	**Accuracy**
Sajjad et al. (2020)	Deep BiLSTM	82.02
Farooq et al. (2020)	DCNN-CFS	81.30
Guizzo et al. (2020)	AlexNet w/ MTS Aug.	55.85
Xu et al. (2021)	Head Fusion	77.80
Sultana et al. (2021a)	CNN+TDF+BiLSTM	86.90
Andayani (2022)	LSTM+Transformer Hybrid	77.33
Aggarwal et al. (2022)	DNN	81.94
VGG16 (Ours)	VGG16	52.43
ResNetV2 (Ours)	ResNetV2	54.51
VGGish (Ours)	VGGish+MLP	50.55
YAMNet (Ours)	YAMNet+MLP	40.47
HuBERT (Ours)	HuBERT Base 960h	**88.54**

Table 51: Comparison among the existing state-of-art works on SAVEE

Work Ref.	**Classification Model**	**Accuracy**
Qayyum et al.	1D CNN	**83.61**
Haider et al. (2021)	AFS-SVM	42.40
Farooq et al. (2020)	DCNN-CFS	82.10
	DGA+PCA	69.88
Alnuaim et al. (2022)	1D CNN	83.33
VGG16 (Ours)	VGG16	45.83
ResNetV2 (Ours)	ResNetV2	38.54
VGGish (Ours)	VGGish+MLP	46.03
YAMNet (Ours)	YAMNet+MLP	36.03
HuBERT (Ours)	HuBERT Base 960h	78.12

Table 52: Comparison among the existing state-of-art works on CREMA-D

Work Ref.	**Classification Model**	**Accuracy**
Mocanu et al. (2021)	SE-ResNet	64.92
Mocanu & Tapu (2021)	GhostVLAD+CNN	64.85
Dolka et al. (2021)	MFCC+ANN	71.69
VGG16 (Ours)	VGG16	54.33
ResNetV2 (Ours)	ResNetV2	46.07
VGGish (Ours)	VGGish+MLP	40.05
YAMNet (Ours)	YAMNet+MLP	34.94
HuBERT (Ours)	HuBERT Base 960h	**72.53**

Table 53: Comparison among the existing state-of-art works on Consolidated English Dataset (TESS, RAVDESS, SAVEE, CREMA-D)

Work Ref.	**Classification Model**	**Accuracy**
Zielonka et al. (2022) (4 Emotions)	CNN	55.89
Zielonka et al. (2022) (All Emotions)	CNN	57.42
VGG16 (Ours)	VGG16	59.38
ResNetV2 (Ours)	ResNetV2	57.33
VGGish (Ours)	VGGish+MLP	50.24
YAMNet (Ours)	YAMNet+MLP	41.20
HuBERT (Ours) (All Emotions)	HuBERT Base 960h	**81.18**

Table 54: Comparison among the existing state-of-art works on SUBESCO, BSER, and Consolidated Bengali Datasets

Work Ref.	**Classification Model**	**SUBESCO**	**BSER**	**Combined**
Sultana et al. (2021a)	CNN+TDF+BiLSTM	82.70	×	×
Chakraborty et al. (2022)	PBLG	**95.3**	**97.5**	**96.0**
VGG16 (Ours)	VGG16	66.07	21.08	66.64
ResNetV2 (Ours)	ResNetV2	59.85	17.68	55.54
VGGish (Ours)	VGGish+MLP	54.18	65.35	52.78
YAMNet (Ours)	YAMNet+MLP	40.02	48.86	39.92
HuBERT (Ours)	HuBERT Base 960h	95.00	87.41	94.09

5. Conclusion with Limitations and Future Directions

Recognition of human emotions through voice involves using machine learning and AI to identify and categorize a speaker's emotional state based on vocal patterns. Although this area has gained some attention recently, it hasn't been as popular as other audio-related deep learning tasks like voice recognition, NLP, and music synthesis. Our study explores the effectiveness of transfer learning in speech emotion recognition (SER). One key benefit of transfer learning is the ability to reuse models from different areas, significantly reducing development costs and time. We leveraged pre-trained models from fields such as image recognition, audio classification, and automatic speech recognition (ASR) to retrain high-performing SER systems with minimal resources. Our experiments demonstrate that fine-tuning these models on small emotional speech datasets can yield excellent results in identifying emotional states. We used four English datasets (TESS, RAVDESS, SAVEE, and CREMA-D) and two Bengali datasets (SUBESCO and BSER), achieving state-of-the-art accuracy in most cases.

Furthermore, we developed a robust real-time speech emotion identification system using these fine-tuned models. This system can analyze and categorize emotional speech in real-time, making it suitable for various practical applications.

One of the unique aspects of our study was the inclusion of both English and Bengali datasets. This allowed us to examine how models trained in one language could transfer to another and explore the potential of transfer learning to enhance emotion detection in bilingual contexts. We combined these datasets to create a bilingual dataset, enabling us to assess the effectiveness of transfer learning across languages. Using the fine-tuned HuBERT model, we achieved an accuracy of 86.51% on this combined dataset. These models were then implemented in a real-time speech emotion detection system, demonstrating their ability to accurately categorize emotional speech in both languages with satisfactory performance.

However, we recognize that transfer learning has its limitations. If the source and target tasks differ significantly, transfer learning might not be effective. Additionally, there's a need for more diverse and representative datasets, as the success of transfer learning systems heavily depends on the quality and variety of the training data.

There are several areas we believe warrant further exploration. For instance, it would be intriguing to continue studying the use of transfer learning for speech emotion recognition across multiple languages. Additionally, examining other models like openSMILE Eyben et al. (2010), Whisper Radford et al. (2022), or Inception Szegedy et al. (2016) for their effectiveness and feasibility in emotion identification systems would be beneficial.

To sum up, Transfer learning presents a promising approach to improve the effectiveness and efficiency of emotion recognition systems across various applications, such as personalized virtual assistants, emotion-sensitive human-machine interfaces, and advanced customer service platforms. By leveraging transfer learning, we can accelerate the development of high-performing emotion detection systems, reducing both time and cost. Implementing a real-time emotion recognition system for speech and utilizing multilingual datasets further demonstrate the versatility and applicability of this method. To fully harness its potential, it is crucial to assess the limitations of transfer learning and seek innovative solutions to address these challenges.

References

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., & Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, .

Aggarwal, A., Srivastava, A., Agarwal, A., Chahal, N., Singh, D., Alnuaim, A. A., Alhadlaq, A., & Lee, H.-N. (2022). Two-way feature extraction for speech emotion recognition using deep learning. *Sensors*, *22*, 2378.

Aggarwal, S. (2018). Modern web-development using reactjs. *International Journal of Recent Research Aspects*, *5*, 133–137.

Akinpelu, S., & Viriri, S. (2022). Robust feature selection-based speech emotion classification using deep transfer learning. *Applied Sciences*, *12*, 8265.

Al-onazi, B. B., Nauman, M. A., Jahangir, R., Malik, M. M., Alkhammash, E. H., & Elshewey, A. M. (2022). Transformer-based multilingual speech emotion recognition using data augmentation and feature fusion. *Applied Sciences*, *12*, 9188.

Alnuaim, A. A., Zakariah, M., Alhadlaq, A., Shashidhar, C., Hatamleh, W. A., Tarazi, H., Shukla, P. K., & Ratna, R. (2022). Human-computer interaction with detection of speaker emotions using convolution neural networks. *Computational Intelligence and Neuroscience*, *2022*.

Andayani, F. (2022). Investigating the impacts of lstm-transformer on classification performance of speech emotion recognition, .

Atsavasirilert, K., Theeramunkong, T., Usanavasin, S., Rugchatjaroen, A., Boonkla, S., Karnjana, J., Keerativittayanun, S., & Okumura, M. (2019). A light-weight deep convolutional neural network for speech emotion recognition using mel-spectrograms. In *2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)* (pp. 1–4). IEEE.

Blumentals, E., & Salimbajevs, A. (2022). Emotion recognition in real-world support call center data for latvian language. *SOCIALIZE 2022*, .

Burkhardt, F., Ajmera, J., Englert, R., Stegmann, J., & Burleson, W. (2006). Detecting anger in automated voice portal dialogs. In *INTERSPEECH*.

Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., & Verma, R. (2014). Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, *5*, 377–390. doi:10.1109/TAFFC.2014.2336244.

Chakraborty, C., Dash*, T. K., Panda, G., & Solanki, S. S. (2022). Phase-based cepstral features for automatic speech emotion recognition of low resource indian languages. *Transactions on Asian and Low-Resource Language Information Processing*, .

Chatterjee, R., Mazumdar, S., Sherratt, R. S., Halder, R., Maitra, T., & Giri, D. (2021a). Real-time speech emotion analysis for smart home assistants. *IEEE Transactions on Consumer Electronics*, *67*, 68–76.

Chatterjee, R., Mazumdar, S., Sherratt, R. S., Halder, R., Maitra, T., & Giri, D. (2021b). Real-time speech emotion analysis for smart home assistants. *IEEE Transactions on Consumer Electronics*, *67*, 68–76.

Choudhary, R. R., Meena, G., & Mohbey, K. K. (2022). Speech emotion based sentiment recognition using deep neural networks. *Journal of Physics: Conference Series*, *2236*, 012003. URL: https://dx.doi.org/10.1088/1742-6596/2236/1/012003. doi:10.1088/1742-6596/2236/1/012003.

Cramer, J., Wu, H.-H., Salamon, J., & Bello, J. P. (2019). Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3852–3856). IEEE.

Da Rocha, H. (2019). *Learn Chart. js: Create interactive visualizations for the web with chart. js 2*. Packt Publishing Ltd.

Das, R. K., Islam, N., Ahmed, M. R., Islam, S., Shatabda, S., & Islam, A. M. (2022). Banglaser: A speech emotion recognition dataset for the bangla language. *Data in Brief*, *42*, 108091. URL: https://www.sciencedirect.com/science/article/pii/S235234092200302X. doi:https://doi.org/10.1016/j.dib.2022.108091.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). Ieee.

Deng, L., Yu, D. et al. (2014). Deep learning: methods and applications. *Foundations and trends textregistered in signal processing*, *7*, 197–387.

Dileep, A. D., & Sekhar, C. C. (2013). Gmm-based intermediate matching kernel for classification of varying length patterns of long duration speech using support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, *25*, 1421–1432.

Dolka, H., VM, A. X., & Juliet, S. (2021). Speech emotion recognition using ann on mfcc features. In *2021 3rd International Conference on Signal Processing and Communication (ICPSC)* (pp. 431–435). IEEE.

Ellis, D., & Plakal, M. (). Yamnet. https://github.com/tensorflow/models/tree/master/research/audioset/yamnet. Accessed: 2022-12-20.

Eyben, F., Wöllmer, M., & Schuller, B. (2010). Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia* (pp. 1459–1462).

Farooq, M., Hussain, F., Baloch, N. K., Raja, F. R., Yu, H., & Zikria, Y. B. (2020). Impact of feature selection algorithm on speech emotion recognition using deep convolutional neural network. *Sensors*, *20*, 6008.

Grinberg, M. (2018). *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

Guizzo, E., Weyde, T., & Leveson, J. B. (2020). Multi-time-scale convolution for emotion recognition from speech audio signals. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6489–6493). IEEE.

Haider, F., Pollak, S., Albert, P., & Luz, S. (2021). Emotion recognition in low-resource settings: An evaluation of automatic feature selection methods. *Computer Speech & Language*, *65*, 101119.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630–645). Springer.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R., & Wilson, K. (2017). Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. URL: `https://arxiv.org/abs/1609.09430`.

Hossain, M. S., Muhammad, G., Song, B., Hassan, M. M., Alelaiwi, A., & Alamri, A. (2015). Audio–visual emotion-aware cloud gaming framework. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*, 2105–2118.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3451–3460.

Islam, M., Anik, M., Hoque, S., Islam, A. et al. (2021). Towards achieving a delicate blending between rule-based translator and neural machine translator. *Neural Computing and Applications*, *33*, 12141–12167.

Islam, M. R., Akhand, M. A. H., Kamal, M. A. S., & Yamada, K. (2022). Recognition of emotion with intensity from speech signal using 3d transformed feature and deep learning. *Electronics*, *11*. URL: `https://www.mdpi.com/2079-9292/11/15/2362`. doi:`10.3390/electronics11152362`.

Ismail, M., & Suh, G. E. (2018). Quantitative overhead analysis for python. In *2018 IEEE International Symposium on Workload Characterization (IISWC)* (pp. 36–47). doi:`10.1109/IISWC.2018.8573512`.

Jack, R. E., Garrod, O. G., & Schyns, P. G. (2014). Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. *Current biology*, *24*, 187–192.

Jackson, P., & Haq, S. (2014). Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*, .

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C. et al. (2020). Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7669–7673). IEEE.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, .

Koduru, A., Valiveti, H. B., & Budati, A. K. (2020). Feature extraction algorithms to improve the speech emotion recognition rate. *International Journal of Speech Technology*, *23*, 45–55.

Krishnan, P. T., Joseph Raj, A. N., & Rajangam, V. (2021). Emotion classification from speech signal based on empirical mode decomposition and non-linear features. *Complex & Intelligent Systems*, *7*, 1919–1934.

License, G. G. P. (2007). Version 3. *Free Software Foundation. URL: http://www. gnu. org/licenses/gpl. html*, .

Lieskovská, E., Jakubec, M., Jarina, R., & Chmulík, M. (2021). A review on speech emotion recognition using deep learning and attention mechanism. *Electronics*, *10*, 1163.

Pires de Lima, R., & Marfurt, K. (2019). Convolutional neural network for remote-sensing scene classification: Transfer learning analysis. *Remote Sensing*, *12*, 86.

Livingstone, S. R., & Russo, F. A. (2018). The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, *13*, e0196391.

Lu, S., Lu, Z., & Zhang, Y.-D. (2019). Pathological brain detection based on alexnet and transfer learning. *Journal of Computational Science*, *30*, 41–47. URL: `https://www.sciencedirect.com/science/article/pii/S1877750318309116`. doi:`https://doi.org/10.1016/j.jocs.2018.11.008`.

Lukic, Y., Vogt, C., Dürr, O., & Stadelmann, T. (2016). Speaker identification and clustering using convolutional neural networks. In *2016 IEEE 26th international workshop on machine learning for signal processing (MLSP)* (pp. 1–6). IEEE.

Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sensing*, *10*, 1119.

Merkel, D. et al. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux j*, *239*, 2.

Mishra, E., Sharma, A. K., Bhalotia, M., & Katiyar, S. (2022). A novel approach to analyse speech emotion using cnn and multilayer perceptron. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 1157–1161). IEEE.

Mocanu, B., & Tapu, R. (2021). Speech emotion recognition using ghostvlad and sentiment metric learning. In *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp. 126–130). IEEE.

Mocanu, B., Tapu, R., & Zaharia, T. (2021). Utterance level feature aggregation with deep metric learning for speech emotion recognition. *Sensors*, *21*, 4233.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.

Nasim, A. S., Chowdory, R. H., Dey, A., & Das, A. (). Recognizing speech emotion based on acoustic features using machine learning. In *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 1–7). IEEE.

Oh, K.-J., Lee, D., Ko, B., & Choi, H.-J. (2017). A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. In *2017 18th IEEE international conference on mobile data management (MDM)* (pp. 371–375). IEEE.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5206–5210). IEEE.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, .

Patel, N., Patel, S., & Mankad, S. H. (2022). Impact of autoencoder based compact representation on emotion detection from audio. *Journal of Ambient Intelligence and Humanized Computing*, *13*, 867–885.

Pichora-Fuller, M. K., & Dupuis, K. (2020). Toronto emotional speech set (tess). *Scholars Portal Dataverse*, *1*, 2020.

Qayyum, A. B. A., Arefeen, A., & Shahnaz, C. (). Convolutional neural network (cnn) based speech-emotion recognition. In *2019 IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)* (pp. 122–125). IEEE.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, .

Ragheb, W., Mirzapour, M., Delfardi, A., Jacquenet, H., & Carbon, L. (2022). Emotional speech recognition with pre-trained deep visual models. *arXiv preprint arXiv:2204.03561*, .

Rehman, A., Liu, Z.-T., Wu, M., Cao, W.-H., & Jia, C.-S. (2022). Real-time speech emotion recognition based on syllable-level feature extraction. *arXiv preprint arXiv:2204.11382*, .

Rosen, S., & Howell, P. (2011). *Signals and systems for speech and hearing* volume 29. Brill.

Sajjad, M., Kwon, S. et al. (2020). Clustering-based speech emotion recognition by incorporating learned features and deep bilstm. *IEEE Access*, *8*, 79861–79875.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, .

Singh, Y. B., & Goel, S. (2022). A systematic literature review of speech emotion recognition approaches. *Neurocomputing*, .

Smilkov, D., Thorat, N., Assogba, Y., Nicholson, C., Kreeger, N., Yu, P., Cai, S., Nielsen, E., Soegel, D., Bileschi, S. et al. (2019). Tensorflow. js: Machine learning for the web and beyond. *Proceedings of Machine Learning and Systems*, *1*, 309–321.

Smus, B. (2013). *Web Audio API: Advanced Sound for Games and Interactive Apps*. " O'Reilly Media, Inc.".

Sonawane, P. K., & Shelke, S. (2018). Handwritten devanagari character classification using deep learning. In *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)* (pp. 1–4). doi:10.1109/ICICET.2018.8533703.

SONG, P., JIN, Y., ZHAO, L., & XIN, M. (2014). Speech emotion recognition using transfer learning. *IEICE Transactions on Information and Systems*, *E97.D*, 2530–2532. doi:10.1587/transinf.2014EDL8038.

Sonmez, Y. , & Varol, A. (2019). New trends in speech emotion recognition. In *2019 7th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1–7). doi:10.1109/ISDFS.2019.8757528.

Stivaktakis, R., Tsagkatakis, G., & Tsakalides, P. (2019). Deep learning for multilabel land cover scene categorization using data augmentation. *IEEE Geoscience and Remote Sensing Letters*, *16*, 1031–1035. doi:10.1109/LGRS.2019.2893306.

Stolar, M. N., Lech, M., Bolia, R. S., & Skinner, M. (). Real time speech emotion recognition using rgb image classification and transfer learning. In *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)* (pp. 1–8). IEEE.

Sultana, S., Iqbal, M. Z., Selim, M. R., Rashid, M. M., & Rahman, M. S. (2021a). Bangla speech emotion recognition and cross-lingual study using deep cnn and blstm networks. *IEEE Access*, *10*, 564–578.

Sultana, S., Rahman, M. S., Selim, M. R., & Iqbal, M. Z. (2021b). Sust bangla emotional speech corpus (subesco): An audio-only emotional speech corpus for bangla. *Plos one*, *16*, e0250173.

Surís, D., Duarte, A., Salvador, A., Torres, J., & Giró-i Nieto, X. (2018). Cross-modal embeddings for video and audio retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (pp. 0–0).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Trinh Van, L., Dao Thi Le, T., Le Xuan, T., & Castelli, E. (2022). Emotional speech recognition using deep neural networks. *Sensors*, *22*, 1414.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Williams, C. E., & Stevens, K. N. (1972). Emotions and speech: Some acoustical correlates. *The journal of the acoustical society of America*, *52*, 1238–1250.

Xu, M., Zhang, F., & Zhang, W. (2021). Head fusion: Improving the accuracy and robustness of speech emotion recognition on the iemocap and ravdess dataset. *IEEE Access*, *9*, 74539–74549.

Yenigalla, P., Kumar, A., Tripathi, S., Singh, C., Kar, S., & Vepa, J. (2018). Speech emotion recognition using spectrogram & phoneme embedding. In *Interspeech* (pp. 3688–3692). volume 2018.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, *27*.

Zielonka, M., Piastowski, A., Czyżewski, A., Nadachowski, P., Operlejn, M., & Kaczor, K. (2022). Recognition of emotions in speech using convolutional neural networks on different datasets. *Electronics*, *11*, 3831.