Project Final Report

Dec 3, 2024

1. Introduction and Background

Speech Emotion Recognition (SER) has been a growing area of research, and various machine learning models have been explored in this domain, each contributing to advancements in recognition accuracy and robustness.

1.A. Literature Review

- Issa et al. [1] proposed a deep learning approach using CNNs, paired with feature extraction techniques such as MFCCs.
- Mashhadi et al. [2] proposed a Random Forest-based feature selection method, where a subset of features is selected based on the importance scores derived from a RF model.
- Striletchi et al. [3] presented TBDM-Net and achieved an impressive F1 score of 91.02 on the RAVDESS dataset.
- Chen et al.[4] employed SVMs and showed their effectiveness in speech emotion classification.
- Al Zoubi et al.[5] demonstrated that SVMs outperformed other classical machine learning models, achieving an accuracy of 85% on the RAVDESS dataset.

1.B. Dataset Description

We are using **RAVDESS** dataset[6], which has the following characteristics:

- **Number of records**: 1440 audio files.
- **Features**:
-
    - **Emotion**: Seven emotions (neutral, calm, happy, sad, angry, fearful, disgust) plus surprise.
-
    - **Emotional intensity**: Two levels (normal and strong) for most emotions, except for neutral which only has normal intensity.
-
    - **Statement**: Two lexically-matched statements.
-
    - **Repetition**: Each statement is repeated twice.

- - **Actor**: 24 actors, with odd-numbered actors being male and even-numbered actors female.
- - **Temporal data**: Since the dataset includes audio recordings, it is temporal in nature as it captures time-series data.

2. Problem Definition

2.A. Problem

The main problem addressed in this project is to develop a machine learning model capable of recognizing and classifying human emotions based on speech audio signals.

2.B. Motivation

Emotion recognition is crucial for enhancing human-computer interaction. It can lead to virtual assistants that respond empathetically, mental health monitoring tools, and improved customer service bots. By understanding emotions, technology can offer more natural interactions, personalized experiences, and even help detect emotional distress for timely support.

3. Methods

3.A. Data Processing Methods

1. **Feature extraction**
   - We've implemented feature extraction using the **'LibROSA'** python library[7]. The current database that we use is the RAVDESS dataset. This dataset has been used in Mahdi's research who employed the LibROSA library for feature extraction[2]. We followed the same method and extracted ZCR, MFCCS, Roll off, Spectral contrast, chroma CQT(Constant-Q Transformation), Tonnetz, Rms, Mel_spectrogram as the features. These features are numerical which allows machine learning models to deal with.
2. **Dimensionality Reduction**
   - After we concatenated every feature type, there are more than 100 columns in our dataset. For the **Random Forest Model**, we employed random forest based feature selection which selects features based on their computed significant level for reducing the dimension.
   - We also applied **PCA** for dimensionality reduction. Our implementation takes the feature-extracted dataset and runs PCA via the **'sklearn'** library[8] in Python. It resulted

in reducing an original number of 108 features to 37 with 95% of variance retained, drastically improving efficiency for our models.

3. **Data Augmentation**
   - We implemented data augmentation by applying noise reduction using the **'noisereduce'** library[9] in Python. It works by computing a spectrogram of a signal (and optionally a noise signal) and estimating a noise threshold (or gate) for each frequency band of that signal/noise. That threshold is used to compute a mask, which gates noise below the frequency-varying threshold.
   - Our noise reduction implementation also reduces multi-channel files to mono-channel, which reduces dimensionality of the data files and creates uniformity among the format of our data files.

3.B. ML Algorithms/Models

1. **Random Forest/Decision Tree**:
   - We implemented the **Random Forest Model** for training the features we extracted. It is beneficial to use the Random Forest Model because it handles data with high dimensions and non-linear relationships well. It also provides insight on the importance of each feature, allowing us to reduce the features by Random Forest based feature selection.
   - In this initial training, we trained the **Random Forest Model** with 100 numbers of estimators. After training the model, we selected the features with the top 20 importance levels and trained again with the selected features.
2. **Support Vector Machine**:
   - We implemented **Support Vector Machine (SVM)** for training the extracted features. SVM is particularly useful in this case because it performs well with high dimensional data and it could handle non-linear relationships effectively with kernel functions. Furthermore, SVM is also resistant to overfitting. In this case, we split the dataset into training size of 0.8 and testing size of 0.2.
   - We train the model with **Radial Basis Function (RBF)** as our kernel function. Then we fine tune the regularization, and selected 25 as our final value. We chose such a high value as we prefer a high level of accuracy of the model, given that there are 8 different emotions to recognize.
3. **Convolutional Neural Network**:
   - We implemented a **CNN** model based on **ResNet101**, using spectrogram images as input with each labelled with its corresponding emotion. The biggest strength of CNNs is its flexibility in capturing most relevant characteristics using convolution layers by itself. Thus, it is capable of finding relevant features in complex tasks which previously seemed impossible to extract. Therefore, it is a strong candidate in SER, as it may find the feature that is linked between voice and emotion.

- Papers using CNNs in our literature review used CNNs only as a feature extraction method and classified data with a combination of manually extracted features. Our proposed solution uses only the CNN extracted features for classification in order to check the capabilities of a completely CNN-based solution.
- The CNN operates in the following order:
  - Create a separate dataset by applying noise-reduction to the original dataset.
  - Both datasets go through preprocessing by time shifting and padding/trimming.
  - Spectrogram images are created for both datasets and converted to tensors for separate inputs of two ResNet101 models. Each model have **70%** of their dataset split for **training data** and **30%** for **testing data**.
  - Train both models until they reach a loss value smaller than **0.001**, with both models evaluated using their corresponding test data.
- The ResNet was trained until it had a loss value smaller than **0.001**; although this may result in overfitting, this was done due to:
  - (1) Memory constraints of creating checkpoint(pth) files while running the model(approximately 500MB per file)
  - (2) Test the performance of overfitted CNN models.

4. Results and Discussion

4.1 Random Forest:

4.1.A. Visualization:

We chose to use **Confusion Matrices** and **Receiver Operating Characteristic Curve (ROC)** to visualize our results via **Random Forest**.

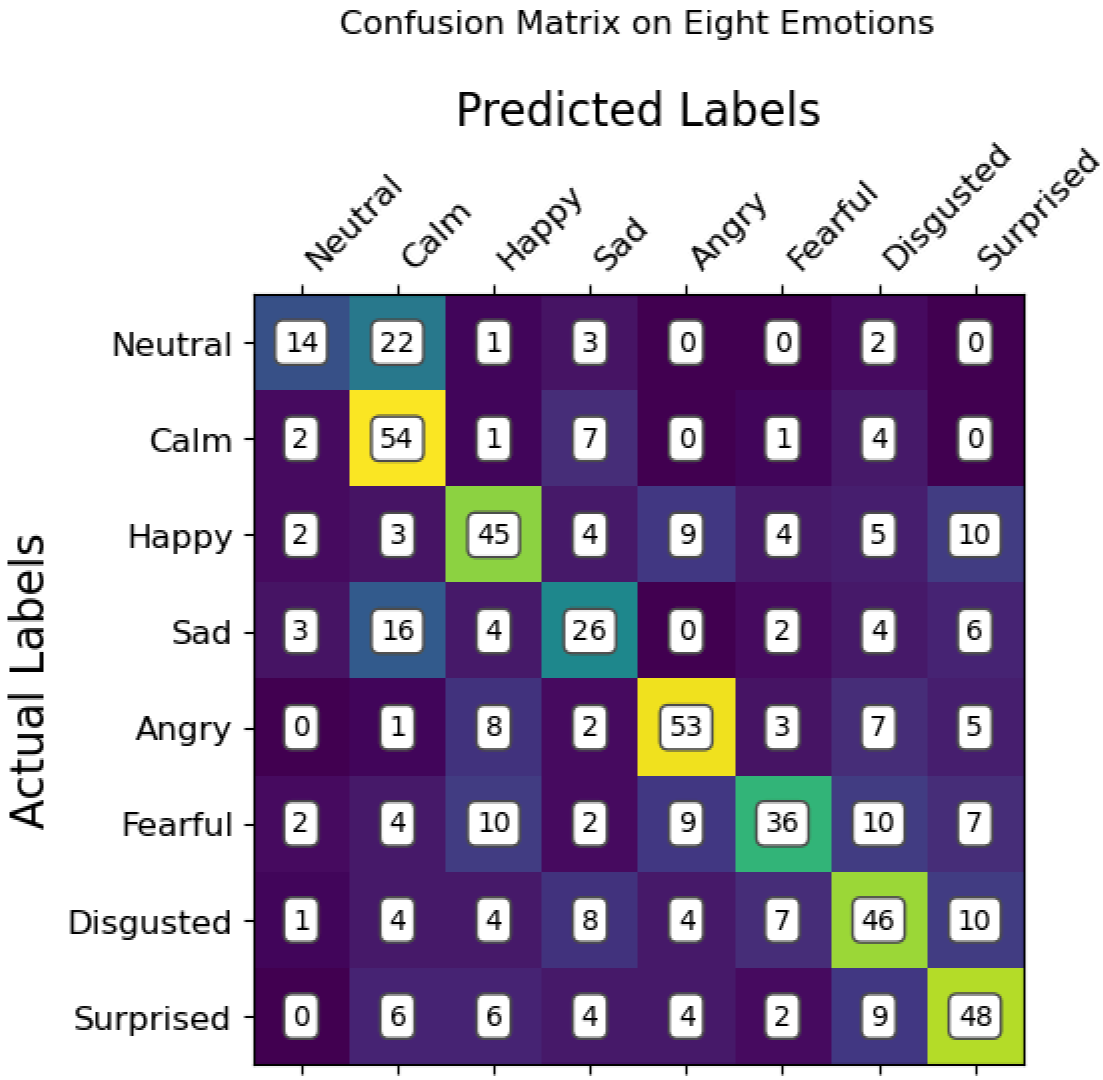

Figure 1. Confusion matrix with all features.

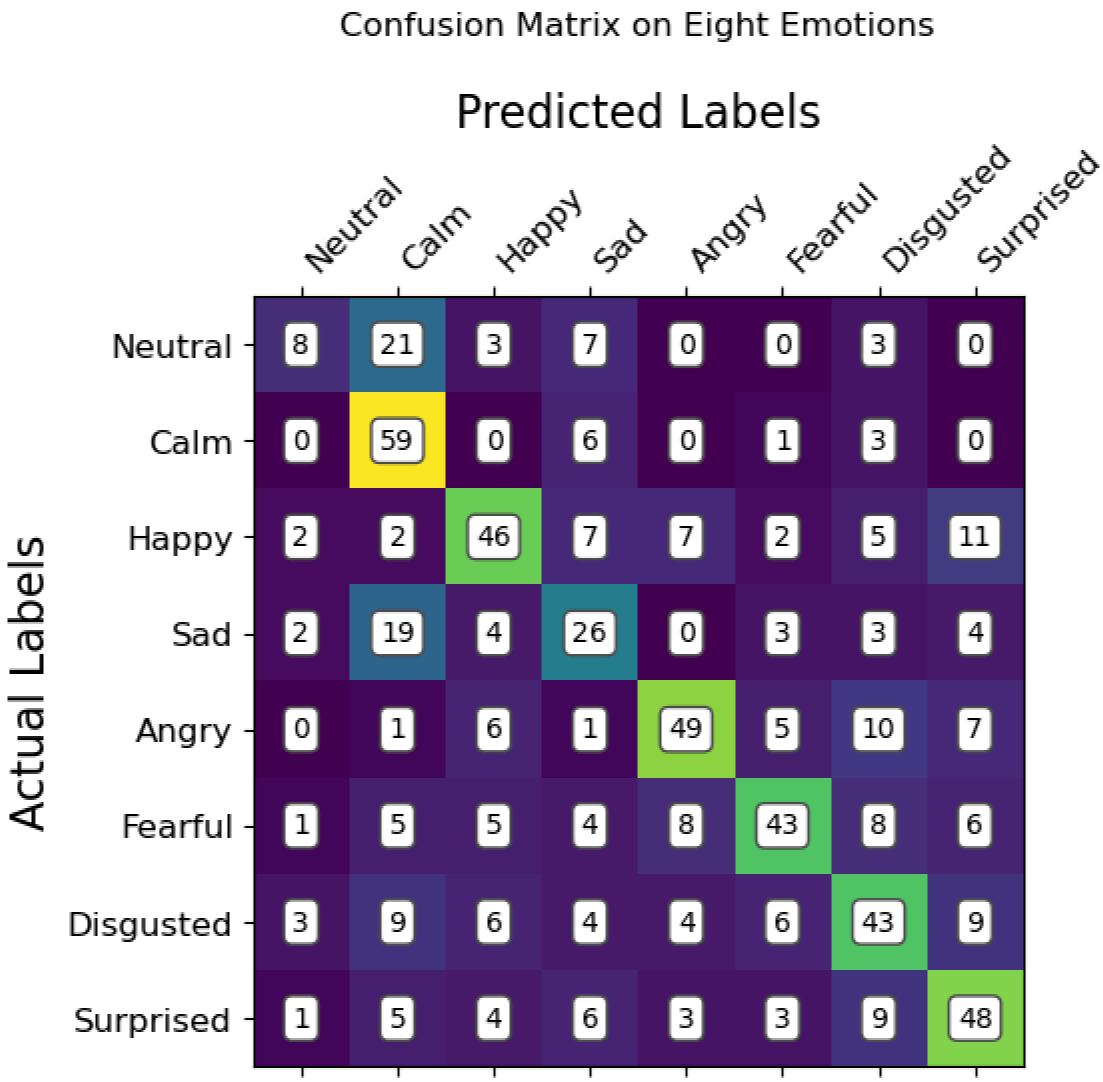

Figure 2. Confusion matrix with feature selection of 20.

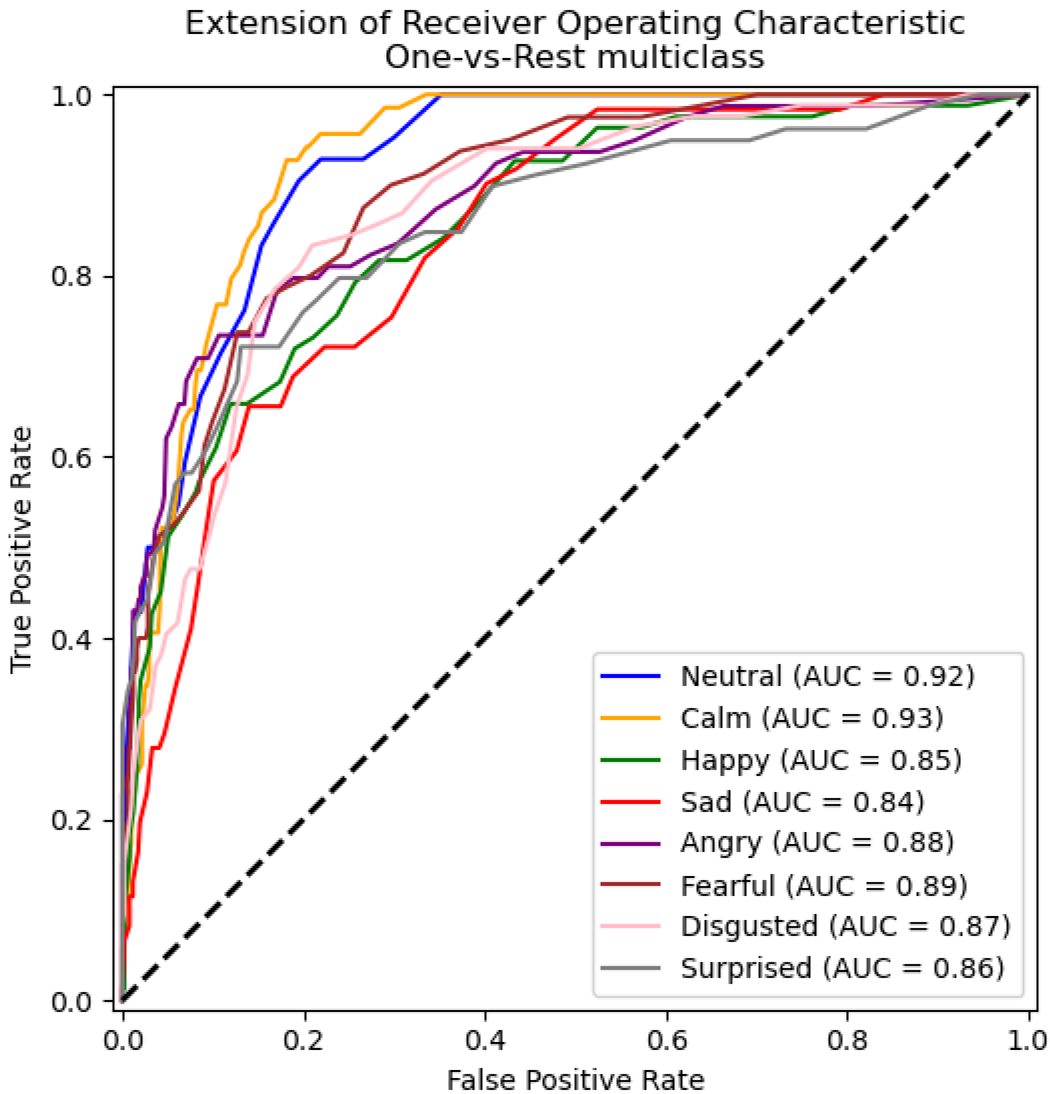

Figure 3. Receiver Operating Characteristic curve with all features.

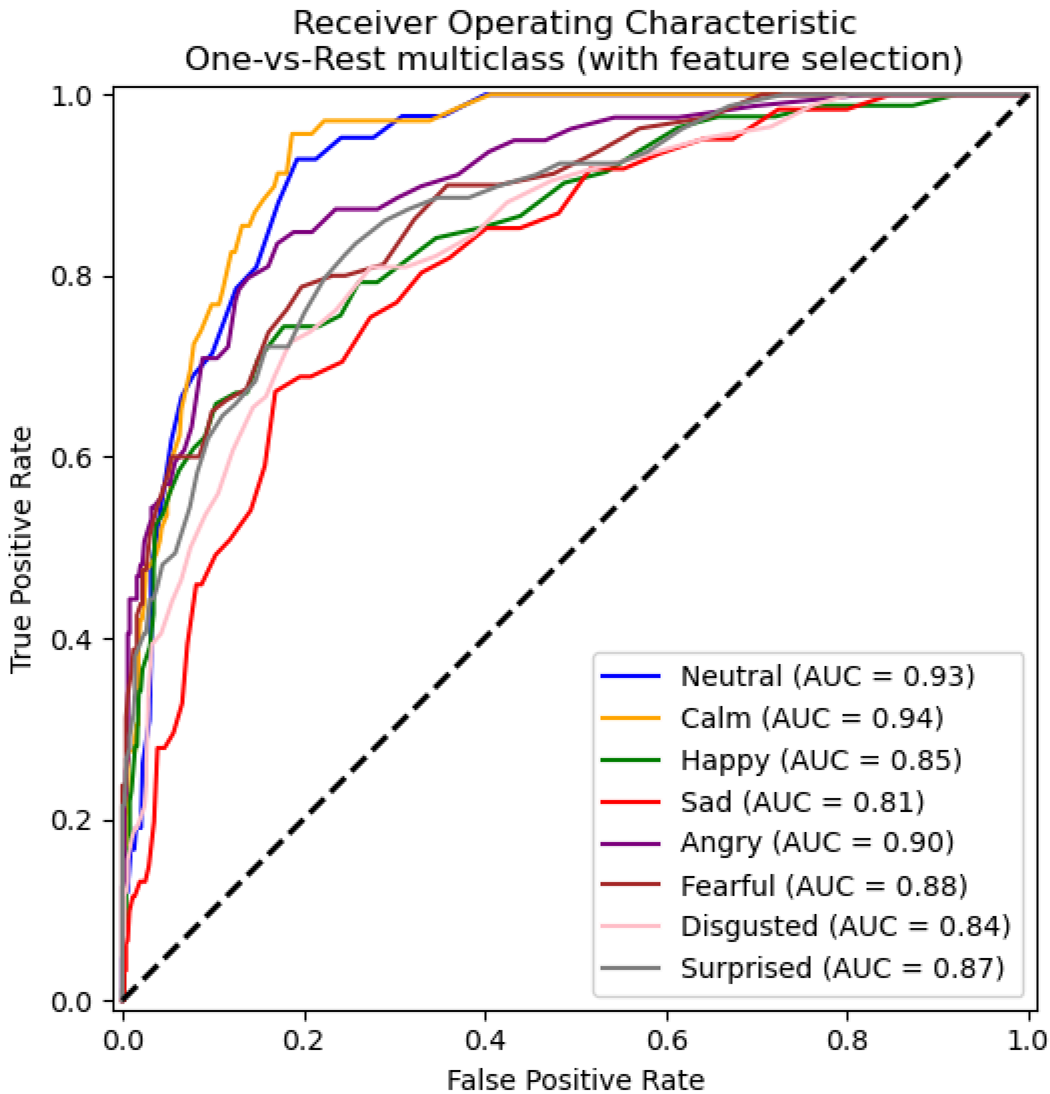

Figure 4. Receiver Operating Characteristic curve with feature selection of 20.

4.1.B. Quantitative Metrics

1. **Accuracy via Confusion Matrices**: with/without the feature selection implemented
   - **Figure 1** shows the result of our Random Forest model when predicting 8 emotions with all the features in this dataset. The diagonal elements indicate correct predictions, and off-diagonal values represent misclassifications. The model accurately predicted 54 instances of "Calm" but confused 16 instances of "Sad" as "Calm". Similarly, the model performed well for "Angry" (53 correct predictions) but misclassified a few instances as "Fearful" or "Sad." Actually, the model struggles more with "Neutral," with only 14 correct predictions and 22 instances misclassified as "Calm."While the model demonstrates strong performances for certain emotions like "Calm" and "Angry," it faces difficulties in accurately differentiating more nuanced emotions like "Neutral" and "Sad."
   - **Figure 2** would be the result with feature selection. The model shows improved accuracy for certain categories, such as "Calm," where 59 instances were correctly predicted, and "Angry," with 49 correct classifications. However, "Neutral" remains

challenging, with only 8 accurate predictions and 21 instances misclassified as “Calm.” The confusion between “Sad” and “Calm” also remains, with 19 instances of “Sad” misclassified as “Calm.”
- **Comparison**: Compared to the earlier model without feature selection, the model demonstrates better specificity for “Calm” and “Angry” but still struggles with emotions like “Neutral” and “Sad.” Feature selection. This suggests that feature selection can improve classification for certain emotions, but not overall accuracy.

2. **Accuracy via classification report**: with/without the feature selection implemented

	precision	recall	f1-score	support
Neutral	0.58	0.33	0.42	42
Calm	0.49	0.78	0.60	69
Happy	0.57	0.55	0.56	82
Sad	0.46	0.43	0.44	61
Angry	0.67	0.67	0.67	79
Fearful	0.65	0.45	0.53	80
Disgusted	0.53	0.55	0.54	84
Surprised	0.56	0.61	0.58	79
accuracy			0.56	576
macro avg	0.57	0.55	0.54	576
weighted avg	0.57	0.56	0.55	576

Figure 5. Precision, Recall, f1-score, support for each class with all features.

- The current accuracy score is approximately **0.56**, still below the expected accuracy. The table displays the F1-score for all eight classes of emotions, exhibiting the performance variations on predicting different classes. Specifically, the model was at the best balance when predicting angry emotions; meanwhile, the model has the lowest accuracy when predicting neutral and sad emotions.

	precision	recall	f1-score	support
Neutral	0.47	0.19	0.27	42
Calm	0.49	0.86	0.62	69
Happy	0.62	0.56	0.59	82
Sad	0.43	0.43	0.43	61
Angry	0.69	0.62	0.65	79
Fearful	0.68	0.54	0.60	80
Disgusted	0.51	0.51	0.51	84
Surprised	0.56	0.61	0.59	79
accuracy			0.56	576
macro avg	0.56	0.54	0.53	576
weighted avg	0.57	0.56	0.55	576

Figure 6. Precision, Recall, f1-score, support for each class with 20 features.

- The accuracy score is **0.56**. Surprisingly, the accuracy remains the same (sometimes it even went down). The table exhibits a similar pattern as the all-featured Random Forest Model.

3. **Accuracy via Receiver Operating Characteristic curve**:
   - AUC stands for Area Under Curve. The higher AUC each class presents, the more capable of the model in distinguishing it from other classes.
   - The dashed diagonal line represents a random classifier (AUC = 0.5). Any curve above this line indicates performance better than random guessing.
   - - With all features: From **Figure 3**, it can be observed that all the curves in the plot are significantly above the diagonal line, showing that the classifier is effective. Specifically, the random forest model performs great when predicting audios with Neutral, Calm sentiments.
   - - With 20 features selected: According to **Figure 4**, we can observe that the same pattern has been exhibited.

4.1.C. Analysis of Random Forest

While we did try on training with efficient data augmentation methods other than noise reduction, the result doesn’t have huge variation, so it is not documented. For further improvements, we can try with augmenting more data that will balance the number of audios for each emotional class, and thus balance precision and recall.

4.1.D Next Steps:

- To keep improving the result, in the data preprocessing aspect, we would need to fine tune the parameters of feature extraction, ensuring that the data fully represent this audio signal, and search on the parameters to augment data. Furthermore, scaling each feature to approximately the same scale would probably improve the result.
- For the model, we could also perform hyperparameter search through modifying the number of estimators, max_depth, sample split size, or class_weight to search for a better result.
- To evaluate the correctness of models in a comprehensive perspective, 1-2 more data visualization methods should be leveraged, and the remaining quantitative metrics, including the cross entropy, need to be evaluated.

4.2 Support Vector Machine:

4.2.A. Visualization:

We chose to use **Confusion Matrices** and **Receiver Operating Characteristic Curve (ROC)** to visualize our results via **Support Vector Machine(SVM)**.

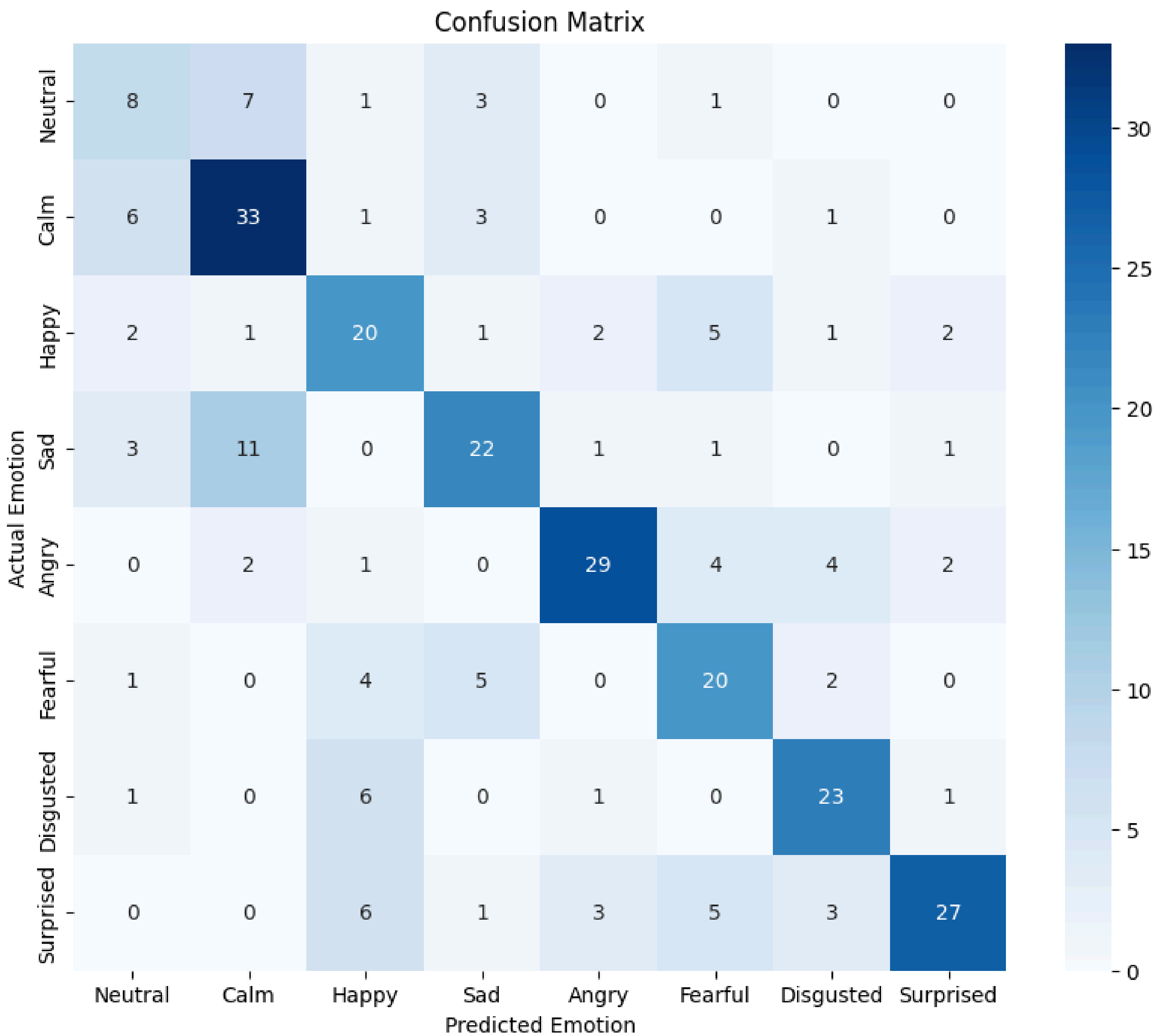

Figure 7. Confusion Matrix for SVM model.

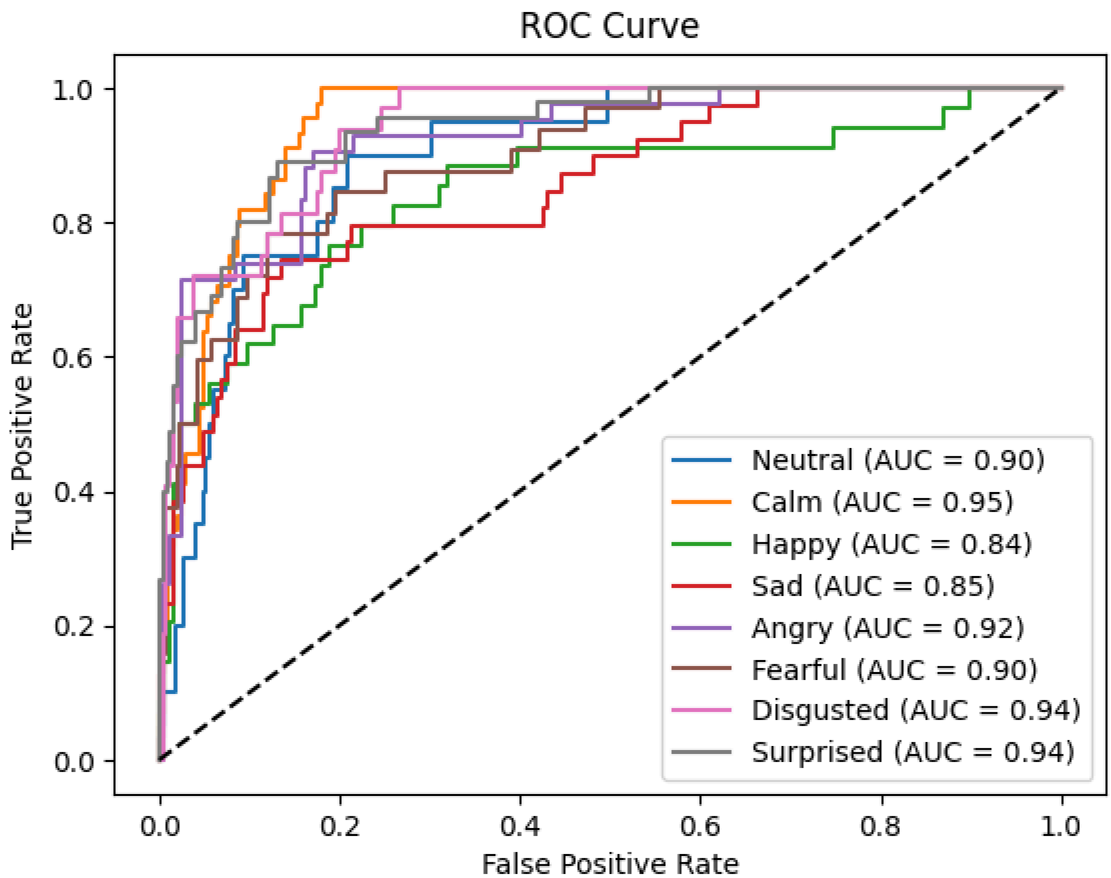

Figure 8. Receiver Operating Characteristic curve for SVM model.

4.2.B. Quantitative Metrics

1. **Accuracy via Confusion Matrix**:
   - The confusion matrix shows the result of our SVM model when predicting 8 emotions with all the features in this dataset. SVM model exhibits strong performances in predicting Calm, Angry audios, and it struggled to distinguish Neutral, Sad, and Calm. As for those with emotions “Happy”, “Surprised”, there are misclassifications across all classes.
2. **Accuracy via classification report**:

```
Classification Report:
              precision    recall  f1-score   support

     Neutral       0.38      0.40      0.39        20
        Calm       0.61      0.75      0.67        44
       Happy       0.51      0.59      0.55        34
         Sad       0.63      0.56      0.59        39
       Angry       0.81      0.69      0.74        42
     Fearful       0.56      0.62      0.59        32
   Disgusted       0.68      0.72      0.70        32
   Surprised       0.82      0.60      0.69        45

    accuracy                           0.63       288
   macro avg       0.62      0.62      0.62       288
weighted avg       0.65      0.63      0.63       288

Overall Metrics:
Accuracy: 0.63
Precision (weighted): 0.65
Recall (weighted): 0.63
F1-Score (weighted): 0.63
```

Figure 9. Precision, Recall, f1-score, support for each class.

- The current accuracy is approximately 0.63, still below the expected accuracy. The table displays the F1-score for all eight classes of emotions, exhibiting the performance variations on predicting different classes' precision and recall. Specifically, the model was at the best balance when predicting angry, disgusted, and surprised emotions; meanwhile, the model has the lowest accuracy when predicting neutral emotions.

3. **Accuracy via Receiver Operating Characteristic curve**:
    - From **Figure 8**, it can be observed that all the curves in the plot are significantly above the diagonal line, showing that the classifier is effective. Specifically, the SVM model performs great when distinguishing audios with Calm, Disgusted, and Surprised sentiments, and it is somehow confusing when predicting sentiments including “Happy” and “Sad”.

4.2.C. Analysis for Support Vector Machine Model

The SVM model provides more accurate predictions compared to the Random Forest model in general. To be specific, SVM is very capable of accurately distinguishing audios with emotions including “Calm”, “Disgusted”, and “Surprised” from other sentiments, although the model may find it struggling to tell the difference from “Sad” and “Neutral”.

4.2.D. Next Steps

- Comprehensive and effective data-preprocessing: we would need to fine tune the parameters of feature extraction and dimensionality reduction, ensuring that the data fully represent this audio signal, and search on the parameters to augment data. Furthermore, scaling each feature to approximately the same scale would probably improve the result.

- For the model, we could also perform hyperparameter optimization by experimenting with different kernels to find the best-fit one. In addition, we can use random search to test kernel-specific hyperparameters.
- To evaluate the correctness of models in a comprehensive perspective, 1-2 more data evaluation methods should be leveraged (including cross entropy).

4.3 Convolutional Neural Network:

4.3.A. Visualization:

We chose to use **Confusion Matrices** and **Receiver Operating Characteristic Curve (ROC)** to visualize our results via **CNN(ResNet101)**.

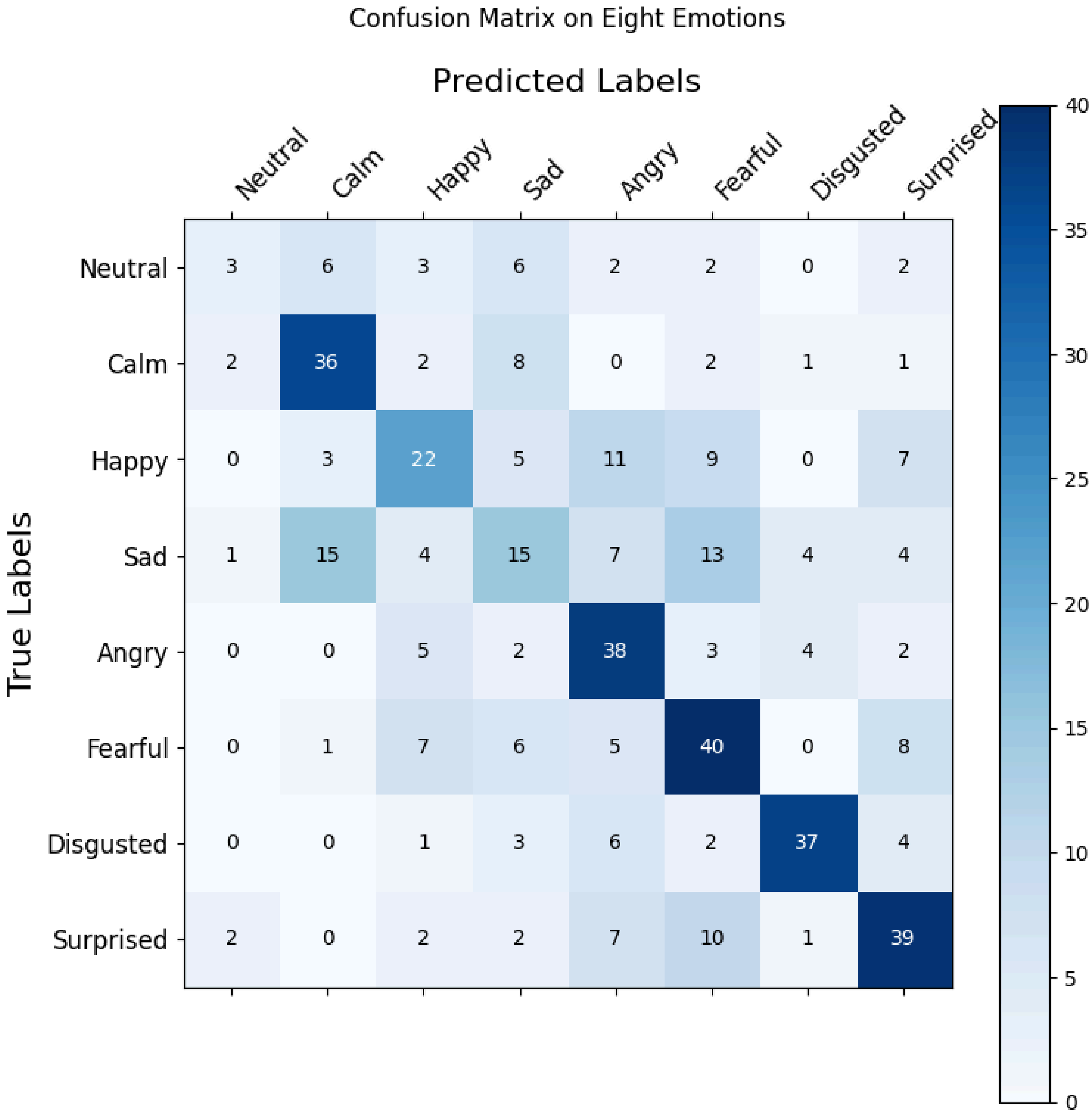

Figure 10. Confusion matrix of original dataset CNN.

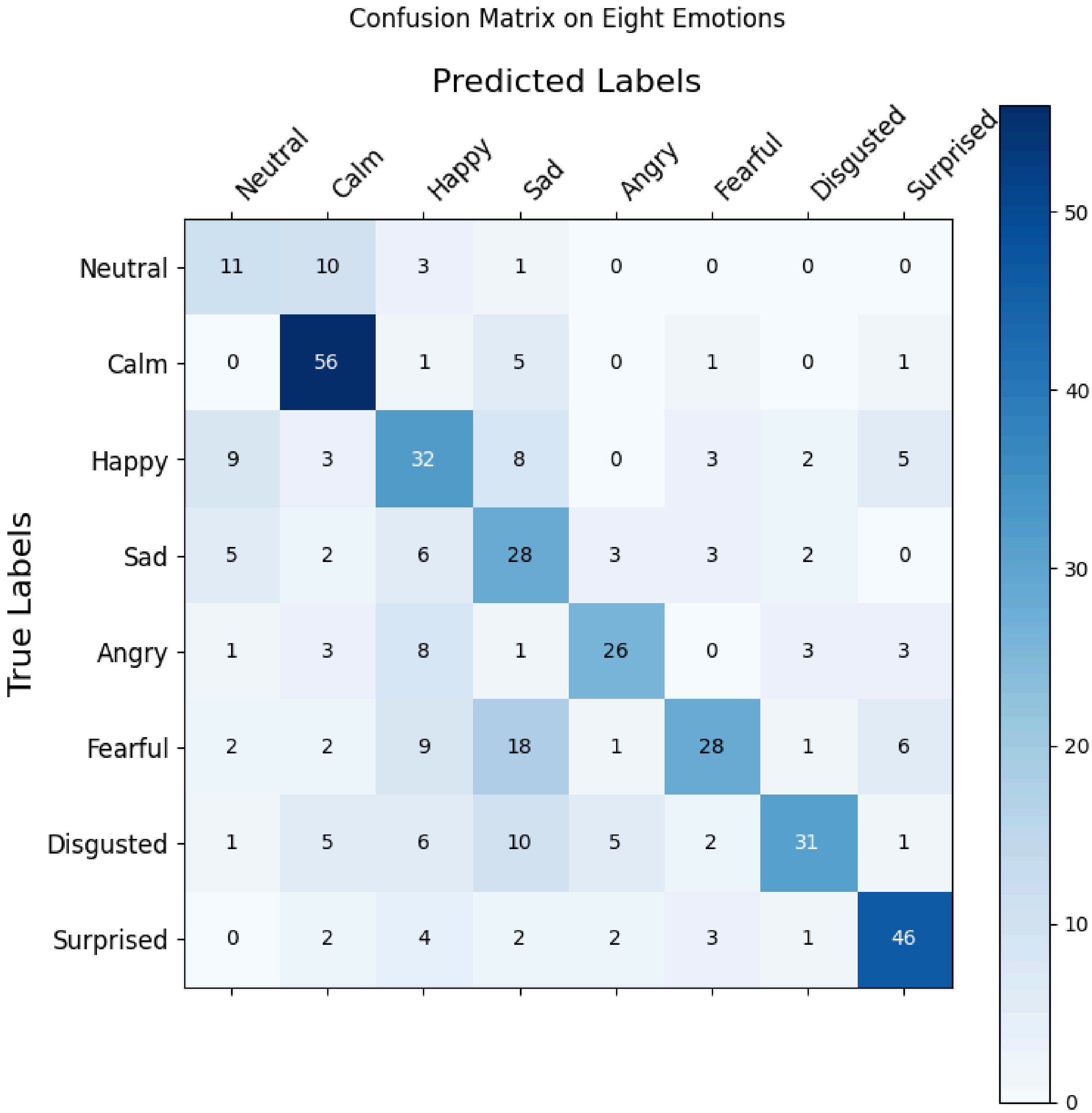

Figure 11. Confusion matrix with noise-reduced dataset CNN.

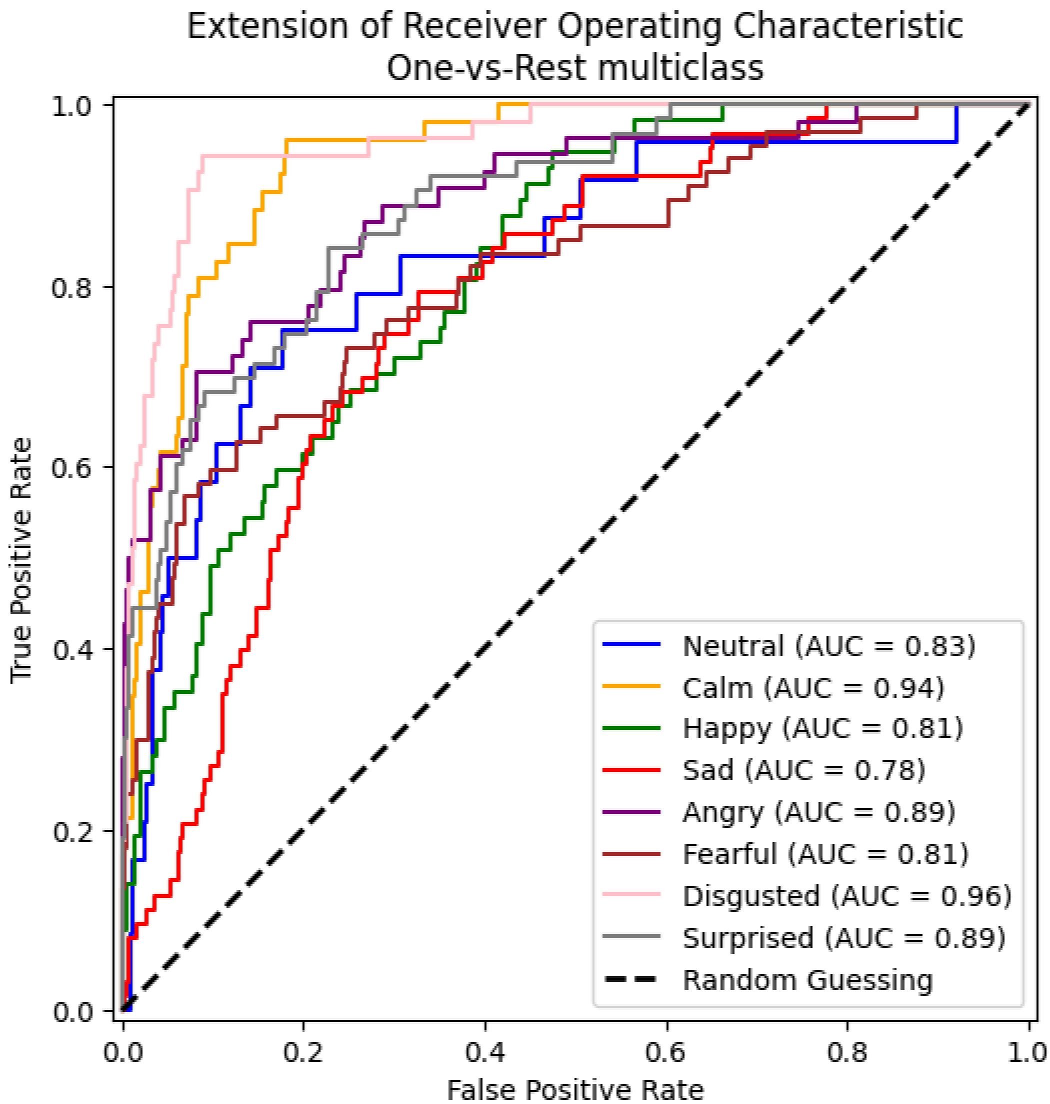

Figure 12. Receiver Operating Characteristic curve with original dataset CNN.

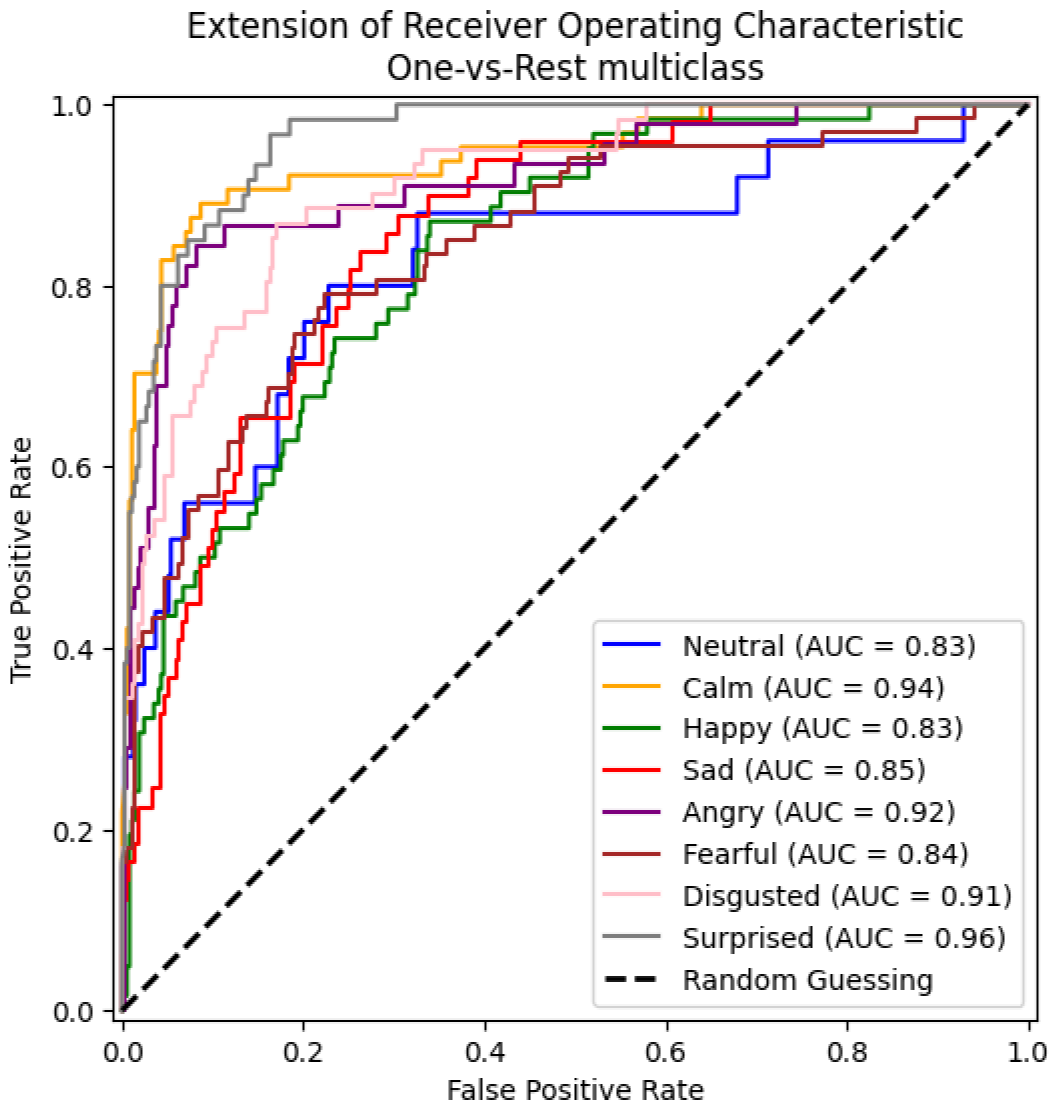

Figure 13. Receiver Operating Characteristic curve with noise-reduced dataset CNN.

```
Test Accuracy: 0.5312

Classification Report:
              precision    recall  f1-score   support

           0       0.38      0.12      0.19        24
           1       0.59      0.69      0.64        52
           2       0.48      0.39      0.43        57
           3       0.32      0.24      0.27        63
           4       0.50      0.70      0.58        54
           5       0.49      0.60      0.54        67
           6       0.79      0.70      0.74        53
           7       0.58      0.62      0.60        63

    accuracy                           0.53       433
   macro avg       0.52      0.51      0.50       433
weighted avg       0.52      0.53      0.52       433


Confusion Matrix:
[[ 3  6  3  6  2  2  0  2]
 [ 2 36  2  8  0  2  1  1]
 [ 0  3 22  5 11  9  0  7]
 [ 1 15  4 15  7 13  4  4]
 [ 0  0  5  2 38  3  4  2]
 [ 0  1  7  6  5 40  0  8]
 [ 0  0  1  3  6  2 37  4]
 [ 2  0  2  2  7 10  1 39]]
```

Figure 14. Precision, Recall, f1-score, support for each class for original dataset CNN .

```
Test Accuracy: 0.5958

Classification Report:
              precision    recall  f1-score   support

           0       0.38      0.44      0.41        25
           1       0.67      0.88      0.76        64
           2       0.46      0.52      0.49        62
           3       0.38      0.57      0.46        49
           4       0.70      0.58      0.63        45
           5       0.70      0.42      0.52        67
           6       0.78      0.51      0.61        61
           7       0.74      0.77      0.75        60

    accuracy                           0.60       433
   macro avg       0.60      0.58      0.58       433
weighted avg       0.62      0.60      0.60       433


Confusion Matrix:
[[11 10  3  1  0  0  0  0]
 [ 0 56  1  5  0  1  0  1]
 [ 9  3 32  8  0  3  2  5]
 [ 5  2  6 28  3  3  2  0]
 [ 1  3  8  1 26  0  3  3]
 [ 2  2  9 18  1 28  1  6]
 [ 1  5  6 10  5  2 31  1]
 [ 0  2  4  2  2  3  1 46]]
```

Figure 15. Precision, Recall, f1-score, support for each class for noise-reduced dataset CNN.

4.3.B. Quantitative Metrics

1. **Accuracy via Confusion Matrix**:
   - The confusion matrix shows the result of our CNN model when predicting 8 emotions with all the features in this dataset.
   - The CNN model trained and tested with the original dataset shows strong results in correctly classifying **Fearful, Surprised, Angry, Disgusted and Calm** emotions(Figure 10). The model was weakest in predicting a **Neural** emotion.

- The CNN model trained and tested with the noise-reduced dataset shows strong results in correctly classifying **Calm and Surprised** emotions(Figure 11). Again, the model was weakest in predicting **Neural** emotions.

2. **Accuracy via classification report**:
    - The classification report shows the result of our CNN model when predicting 8 emotions with all the features in this dataset. The table displays the **F1-score** for all eight classes of emotions, exhibiting the performance variations on predicting different classes' precision and recall. Note that the **labels are reduced by 1 due to tensor classification**; thus, one should be added to get the original RAVDESS emotion number.
    - The CNN model trained and tested with the original dataset resulted in an accuracy of approximately **0.53(Figure 14)**, which is much lower than accuracies achieved when combined with extracted features. The model showed strengths in predicting **Disgusted, Calm and Surprised** emotions; on the other hand, showed lowest accuracy when predicting **Neutral** emotions.
    - The CNN model trained and tested with the noise-reduced dataset resulted in an accuracy of approximately 0.6(Figure 15), which is much lower than accuracies achieved when combined with extracted features but a slight improvement over a CNN trained on non noise-reduced data. The model showed strengths in predicting **Calm, Disgusted, Surprised, Angry and Fearful** emotions; on the other hand, showed lowest accuracy when predicting **Neutral** emotions.
3. **Accuracy via Receiver Operating Characteristic curve**:
    - The curve of the CNN model trained and tested with the original dataset shows all curves above the "random guessing" line, showing it performing better than simple guessing and thus its effectiveness. The model is best in correctly classifying **Disgusted, Calm and Surprised** emotions and weak in classifying **Neutral and Happy** emotions.
    - The curve of the CNN model trained and tested with the noise-reduced dataset shows all curves above the "random guessing" line, showing it performing better than simple guessing and thus its effectiveness. The model is best in correctly classifying **Surprised, Calm and Angry** emotions and weak in classifying **Neutral and Happy** emotions.

4.3.C. Analysis of the CNN model:

As expected, the model performed poorly compared to solutions that combine the features extracted by a CNN and manually extracted features. However, the accuracy obtained by only using a deep CNN model(ResNet101) with spectrogram images as input is promising; mainly considering that our model is likely to be currently overfitted with the training data, which may lead to relatively low accuracies.

One key takeaway from our results is the difference between CNN models trained with **original** and **noise-reduced datasets**. The CNN trained by original data performed better in predicting **Angry, Fearful and Disgusted** emotions, while the model trained by noise-reduced data performed significantly better in other emotions.

4.3.D. Next Steps

There are several potential ways for improvement:

- Both models stopped training when a loss value of **0.001** or lower was achieved. This was to intentionally test how overfitted CNN models perform in motion prediction; testing additional loss values to find the optimal target value has high potential in increasing the accuracy of the model.
- It is evident that noise-reduction is effective in improving the accuracy of the model, with exception to three emotions; Angry, Fearful and Disgusted. Thus, **selective noise reduction** by excluding these three emotions from noise-reduction is likely to improve the overall performance of the model.
- The spectrogram obtained is relatively low-resolution with minimal data preprocessing. Using **higher-resolution spectrograms** with more **fine-grained preprocessing** would help the CNNs identify the corresponding features more effectively.

4.4 Model Comparisons:

We will be evaluating our models with regard to mainly 3 aspects:

- **Accuracy, Weighed Precision, Recall, F1-score**
- **Best & Worst Performing Sentiment Class**
- **Training time, Feature Scalability, Noise Sensitivity**

Models	Accuracy	Precision	Recall	F1-score	Best Performing class	Worst Performing class	Training cost	Feature Scalability	Noise sensitivity
Random Forest with all features	56%	0.57	0.56	0.55	Calm, Angry	Sad, Neutral	Low (parallelizable, quick)	High (handles many features)	Low (robust to noise due to averaging across trees)
Random Forest with 20 features selected	56%	0.57	0.56	0.55	Calm, Angry	Sad, Neutral	Low (reduced dimensionality)	High	Low
Support Vector Machine (SVM)	63%	0.65	0.65	0.63	Calm, Angry	Neutral	Medium(linear kernel is reasonable)	Medium (scales with dimensionality)	Medium (sensitive to noise in soft-margin SVMs)
CNN	53%	0.52	0.53	0.52	Fearful, Surprised	Neutral, Sad	High (requires large data and GPU for training)	Low (does not perform well with high-dimensional tabular data)	High (sensitive to noisy inputs)
CNN with noise-reduction	60%	0.62	0.6	0.6	Calm, Surprised	Neutral, Sad	High (includes preprocessing time)	Low	Medium (noise-reduction preprocessing mitigates sensitivity)

Figure 16. Accuracy Precision, Recall, f1-score, best performing class, worst performing class, training cost, feature scalability, noise sensitivity for all models.

The table above summarizes key findings retrieved from our previous visualizations & analysis. It is evident that each model has their benefits and drawbacks in multiple aspects:

- **SVM** is the optimal choice in terms of accuracy & overall performance.
- **RF** is more robust to noise and takes less cost to train.
- **CNN** are known to be effective for image-like data, but showed lower accuracy (53%) and high sensitivity to noise, with high training costs due to the computational demands of deep learning. However, it has relatively higher potential for improving its performance; CNN can greatly benefit from preprocessing steps, making them more insensitive to noisy inputs.

Therefore, for classifying the sentiment of audios in real-life scenarios, we can conclude that **the selection of machine-learning models should be based on reality constraints and different focus on the results**.

4.5 Quantitative Metrics References

1. **Precision**: the proportion of ground truth in all predicted positive results
2. **Recall**: the proportion of ground truth in all actual positive results
3. **F1-Score**: the balance of precision and recall; demonstrate overall model performance in terms of accuracy.
4. **Confusion Matrix**: Visualize model performance across different emotions.
5. **Receiver Operating Characteristic curve**:
    - Plotting the **True Positive Rate** against the **False Positive Rate** at various decision thresholds. It helps visualize the trade-off between **sensitivity (recall)** and **specificity** and identifies the **optimal threshold for classification**.
    - The **Area Under the Curve (AUC)** quantifies the model's overall performance, with a **value closer to 1** indicating **better discrimination ability**.

5. References

[1] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomedical Signal Processing and Control*, vol. 59, p. 101894, 2020. doi: 10.1016/j.bspc.2020.101894.

[2] R. Mashhadi, M. Mahdi, and K. Osei-Bonsu, "Speech emotion recognition using machine learning techniques: Feature extraction and comparison of convolutional neural network and Random Forest," *PLOS ONE*, vol. 18, no. 11, Nov. 2023. doi: 10.1371/journal.pone.0291500.

[3] V. Striletchi, C. Striletchi, and A. Stan, "TBDM-Net: Bidirectional dense networks with gender information for speech emotion recognition," *arXiv.org*, 2024. doi: 10.48550/arXiv.2409.10056.

[4] L. Chen, X. Mao, Y. Xue, and L. Cheng, "Speech emotion recognition: Features and classification models," *Digital Signal Processing*, vol. 22, no. 6, pp. 1154-1160, 2012. doi: 10.1016/j.dsp.2012.05.007.

[5] R. Al Zoubi, A. Turky, and S. Foufou, "Speech emotion recognition using support vector machine," in *Proc. 9th Int. Congr. Inf. Commun. Technol. (ICICT 2024)*, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., Lecture Notes in Networks and Systems, vol. 1003, Singapore: Springer, 2024, pp. 519-531. doi: 10.1007/978-981-97-3302-6_42.

[6] UWRF-Kaggler, "RAVDESS Emotional Speech Audio," *Kaggle*, 2018. [Online]. Available: https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio.

[7] librosa, "libROSA." [Online]. Available: https://librosa.org/

[8] scikit-learn, "sklearn." [Online]. Available: https://scikit-learn.org/stable/

[9] PyPI, "noisereduce." [Online]. Available: https://pypi.org/project/noisereduce/

Gantt Chart(Click to Download!)

Contribution Table

Name	Contribution
Adar Avsian	- Set up meeting with TA - Helped define problem description - Found dataset - Implemented SVM
Mei Li	- Literature Review - Group meetings setup - Data Visualization - Quantitative Metrics analysis - Presentation Slides
Feiyang Xie	- Literature Review - Feature Extraction - Data Augmentation - Implemented Random Forest Model
Eric Shao	- Literature Review - Presentation Video Recording
Calvin(Hyunsoo) Yang	- Literature Review - Feature Extraction - Data Augmentation(Noise Reduction) - Data Preprocessing(PCA) - Implemented CNN - Created Gantt Chart - Created Github Repo & Pages

CS 4641 ML Group 37 Project Webpage