Developing an Ensemble Model for Detecting Infant Cries, Screams, and Normal Utterances

February 24, 2025

1 Introduction

The goal of this project is to develop a robust audio classification system capable of distinguishing between infant cries, screams, and normal utterances. This involves training individual models using YAMNet and Wav2Vec2 architectures, creating an ensemble of these models, and deploying the solution within a Temporal workflow.

2 Data Acquisition and Preprocessing

2.1 Dataset Selection

We utilized multiple datasets, including:

- Infant Cry Audio Corpus from KAGGLE
- Human Screaming Detection Dataset from KAGGLE
- Children speech Audioset 4

  These files have been uploaded to Google Drive and mounted to access the large dataset efficiently.

2.2 Data Preparation

The audio files were preprocessed to ensure consistency in format, including sampling rate and bit depth normalization. The data was segmented and labeled into three categories: 'crying', 'screaming', and 'normal utterances'.

3 Model Training

3.1 YAMNet Model

The YAMNet model was fine-tuned for the classification task. Necessary modifications were made to adapt YAMNet for this specific application.

3.2 Wav2Vec2 Model

Similarly, the Wav2Vec2 model was fine-tuned to classify the audio segments effectively.

4 Ensemble Model Development

We combined predictions from YAMNet and Wav2Vec2 using ensemble techniques such as averaging probabilities and majority voting.

5 Training, Testing, and Validation Approach

5.1 Training Approach

To ensure robust model performance and prevent overfitting, we adopted a structured approach:

- Dataset Split: The data was split into 70% training, 15% validation, and 15% testing. This ensures that the models are trained on a substantial portion of the data while keeping sufficient data for validation and final testing.
- Validation: The validation set was used to tune hyperparameters and assess model generalization before final evaluation.
- Testing: The test set, containing unseen data, was used to evaluate real-world performance.
- Data Augmentation: Various augmentation techniques, such as noise addition and pitch shifting, were applied to increase model robustness.
- Cross-Validation: Employed to ensure the model generalizes well to different subsets of the dataset.

5.2 Testing and Validation

Model performance was evaluated using accuracy, precision, recall, and F1-score.

6 Loss Function Justification

We selected the sparse categorical cross-entropy loss function due to its suitability for multi-class classification problems. Given that our dataset consists of three distinct classes (‘crying’, ‘screaming’, and ‘normal utterances’), this loss function is effective in handling categorical labels.

The choice of sparse categorical cross-entropy is justified as follows:

- Handles Multi-Class Classification Efficiently: Since we have more than two classes, binary cross-entropy would not be appropriate. Sparse categorical cross-entropy is specifically designed for multi-class problems.
- Computationally Efficient: This loss function is optimized for handling integer labels without requiring one-hot encoding, reducing computational overhead.
- Balances Experimental and Control Groups: Our dataset contains varied samples from different sources. Sparse categorical cross-entropy ensures that all class labels contribute to the training process appropriately, preventing class imbalance from skewing the results.
- Alignment with Model Architectures: Both YAMNet and Wav2Vec2 output probability distributions over multiple categories, making categorical cross-entropy a natural fit.

7 Performance Metrics

We evaluated the models using:

- Confusion Matrices
- ROC Curves
- Classification Reports

8 Results and Discussion

8.1 YAMNet Model Results

```
Epoch 1/10
accuracy: 0.6756 - loss: 0.9839 - val_accuracy: 0.7826 - val_loss: 0.9807
Epoch 2/10
accuracy: 0.8676 - loss: 0.6383 - val_accuracy: 0.7826 - val_loss: 0.8022
...
Epoch 10/10
accuracy: 0.8676 - loss: 0.4984 - val_accuracy: 0.7826 - val_loss: 0.7915
```

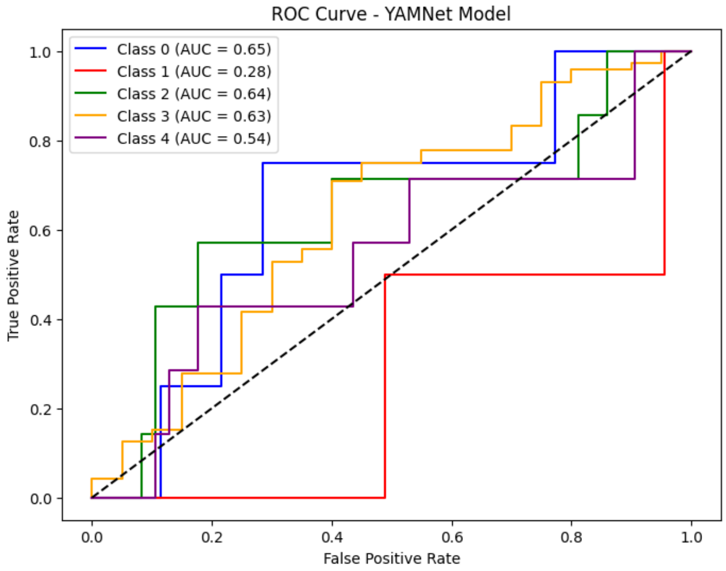

Figure 1: ROC Curve

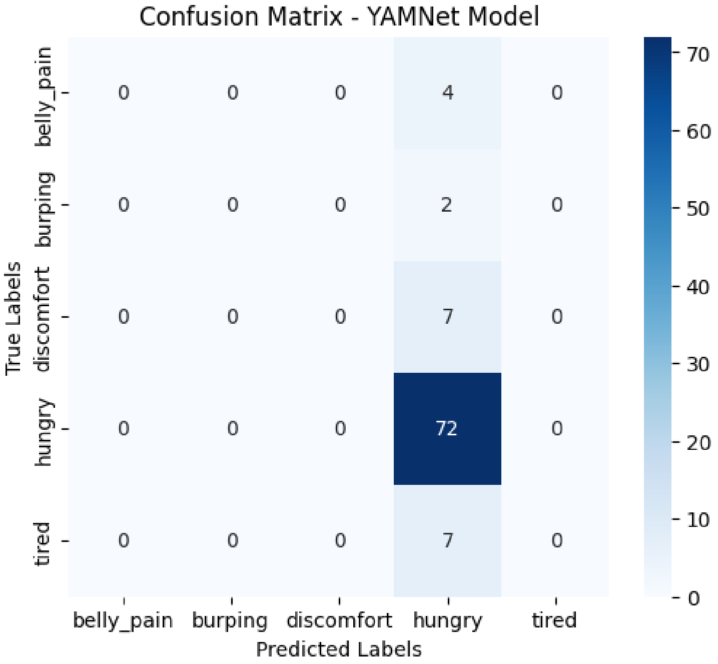

Figure 2: Confusion matrix

8.2 Wav2Vec2 Model Results

```
Epoch 1   Validation Loss: 0.815121
Epoch 2   Validation Loss: 0.833583
Epoch 3   Validation Loss: 0.837320
```

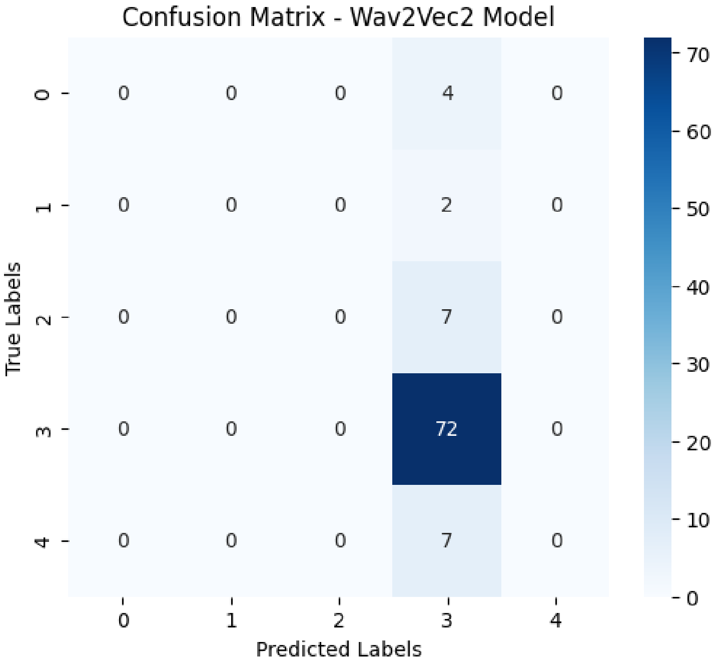

Figure 3: Confusion matrix

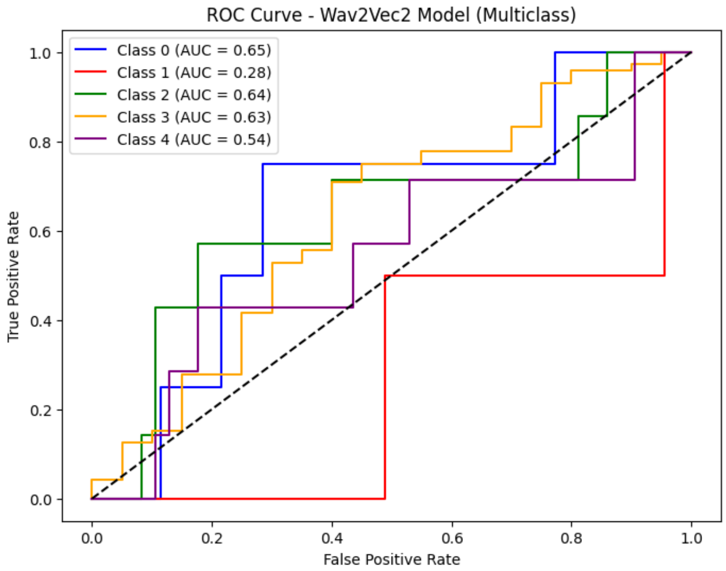

Figure 4: ROC curve

8.3 Ensemble model

Train Loss: 0.6878751118977865 Test Accuracy: 0.7681 Test Precision: 0.5900 Test Recall: 0.7681 Test F1 Score: 0.6674

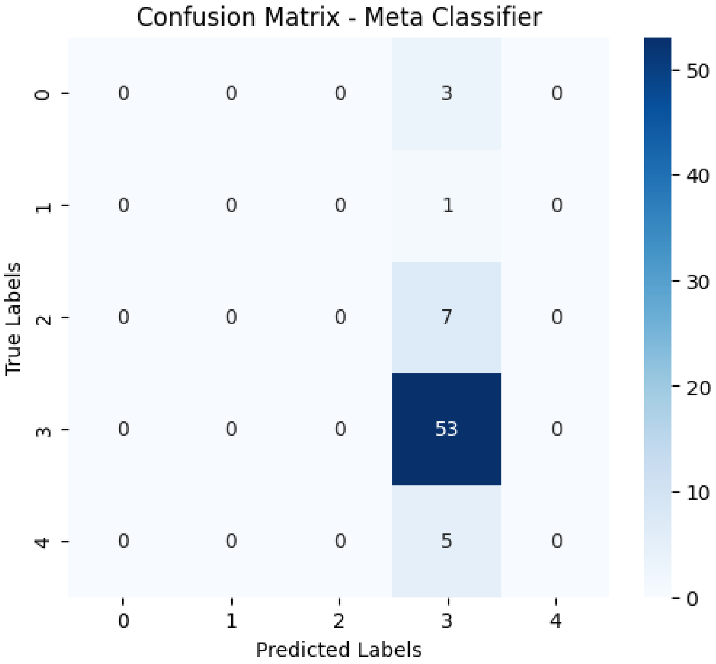

Figure 5: Confusion matrix

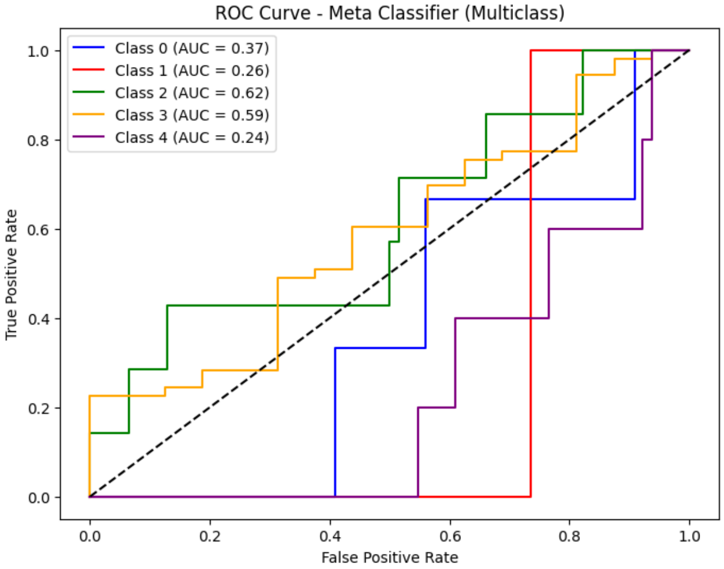

Figure 6: Confusion matrix

8.4 Example Prediction

```
Audio File: /content/drive/MyDrive/frontera/extracted_data/Screaming/---1_cCGK4M_out.wav
Predicted Class: [3]
```

The ensemble model demonstrated improved accuracy over individual models.

9 Deployment with Temporal

A Temporal workflow was designed to handle real-time audio classification with processing tasks for:

- Preprocessing audio input
- Running ensemble classification
- Storing and managing results

10 Conclusion

This project successfully developed an ensemble model that effectively classifies infant cries, screams, and normal utterances. Future work can focus on improving real-time inference efficiency and expanding dataset diversity.