Intelligent Urban Sound Detection

Jiahui Meng
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, United States
jiahuime@usc.edu

Yuming Li
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, United States
liyuming@usc.edu

***Abstract*—Sound event detection plays a critical role in intelligent audio surveillance and urban noise analysis. Traditional machine learning approaches often face challenges in handling complex, unstructured sound data, limiting their generalization capabilities. In this study, we propose a transformer-based model enhanced with MFCC (Mel-Frequency Cepstral Coefficients) features to address these limitations. The model utilizes a weighted Binary Cross-Entropy loss function to counter class imbalance in the dataset. Our results demonstrate that the model trained with MFCC features outperforms those using original sound features, achieving a test accuracy of 84.07% and an F1 score of 0.6066. These findings underscore the effectiveness of MFCC features and transformer architectures for accurate and robust sound classification in dynamic urban environments.**

***Index Terms*—Sound Event Detection, Transformer Model, MFCC Features, Weighted Binary Cross-Entropy, Audio Classification, Machine Learning**

I. Introduction

Sound event detection is an essential task with applications in environmental monitoring, urban planning, and intelligent audio surveillance. By identifying common urban sounds, such as traffic noises or emergency sirens, systems can improve real-time awareness and provide critical data insights for various applications [1]. However, traditional machine learning approaches for sound classification, such as support vector machines (SVM), decision trees, k-nearest neighbors (k-NN), and random forests, often face limitations in handling complex, unstructured sound data due to their reliance on hand-crafted features or limited context awareness.

To address these challenges, we aim to leverage transformer-based models, which have demonstrated success in sequential data tasks by capturing long-range dependencies and contextual relationships within data [2]. Transformers' self-attention mechanisms enable more nuanced feature extraction from audio sequences, potentially outperforming traditional methods in complex sound environments. We seek to enhance model accuracy and robustness in dynamic, real-world audio settings by applying this advanced approach to sound event detection. This proposal aims to develop a model that can detect and localize sound events by mixing sound files from the dataset into a one-minute audio clip and determining the occurrence and timing of each sound.

II. Related works

Mesaros, A., Heittola, T., and Virtanen, T. present the recording and annotation procedure, the database content, a recommended cross-validation setup, and the performance of a supervised acoustic scene classification system and event detection baseline system using Mel frequency cepstral coefficients and Gaussian mixture models [3]. Parascandolo, G., Huttunen, H., and Virtanen, T. present an approach to polyphonic sound event detection in real-life recordings based on bi-directional long short-term memory (BLSTM) recurrent neural networks (RNNs) [4]. Xu, Y., Kong, Q., Wang, W., and Plumbley, M. D. present a gated convolutional neural network and a temporal attention-based localization method for audio classification, which won the first place in the large-scale weakly supervised sound event detection task of Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 challenge [5]. Unlike the above work, our project will create a new dataset for audio semantic segmentation, like determining the occurrence and timing of each sound.

Salamon et al. address significant gaps in automatic urban sound classification by introducing a structured taxonomy and the UrbanSound dataset, which comprises 27 hours of annotated urban field recordings designed to facilitate standardized research [1]. They establish a detailed taxonomy to categorize urban sounds, enhancing comparability and reproducibility across studies. For empirical validation, they conduct baseline classification experiments using the UrbanSound dataset. The features employed are Mel-Frequency Cepstral Coefficients (MFCCs), a standard in audio analysis. Various classification algorithms are tested, including decision trees, k-NN, random forests, and support vector machines, using a 10-fold cross-validation method to assess the performance of the dataset. This foundational effort sets a benchmark for future urban acoustic research by illustrating the utility and challenges of the dataset.

III. Methods

This section provides an overview of the project's implementation. It begins with the creation of a custom dataset, followed by training using an enhanced Transformer encoder model, and concludes with the generation of prediction results. Figure 1 illustrates the project's general workflow. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score, providing a comprehensive understanding of its predictive capabilities.

A. Data Preparation

The UrbanSound8K dataset contains 8732 labeled sound excerpts of urban sounds from 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music [1]. Each class aligns with an urban sound taxonomy, facilitating sound classification tasks. The maximum number of sound excerpts in each category is 1000 and the minimum is 374. All data are stored in audio (WAV file) and the label is stored in a CSV file.

Based on the UrbanSound8K dataset, we created a dataset containing 5000 one-minute audio clips. Each audio clip consists of 30 small sound excerpts. Small sound excerpts belong to 10 different classes and all of them are from the dataset. For each audio clip, we saved detailed temporal labels, including the start and end times for each occurrence. Especially, if a single sound type excerpt appeared in non-continuous segments within an excerpt, we labeled each segment independently.

The project utilized Mel Frequency Cepstral Coefficient (MFCC) to extract sound features. MFCC better represents the perceptually relevant aspects of the short-term speech spectrum [6]. The feature extraction process includes Pre-emphasis, Framing, Window, Fourier-Transform and Power Spectrum, Filter Banks (or Mel spectrum), Mel-frequency Cepstral Coefficients (MFCCs) or Discrete Cosine Transform(DCT), and Mean Normalize. After doing feature extraction, each audio file from the UrbanSound8K dataset is first converted into a matrix. Mel-spectrograms effectively capture the temporal and spectral characteristics of audio, making them an ideal input for the Transformer model. Thus, there are two datasets in this project, one is the original audio waveform dataset and the other one is the MFCC matrix dataset. For the waveform dataset, the input size is 5000*1323000. For the MFCC dataset, the input size is 5000*256*2584. At the same time, we use the same output label for both of them and the size is 5000*600*10. 5000 is the number of files. 10 frames for each second, 600 frames for 1 minute, and 10 belong to 10 different categories.

B. Network Implementation

The model is based on the Transformer encoder architecture. Two one-dimensional convolutions are initially appended to the input of the Transformer encoder. Following the first convolution, the last two dimensions of the data are transposed before applying the second convolution, effectively compressing information across the feature and frame dimensions. This processed data is then fed into the Transformer encoder. In this configuration, unnecessary padding is removed and token embeddings are omitted, with each frame of data directly input as a token. At the output stage of the Transformer encoder, the data is flattened and processed through a multilayer perceptron (MLP) layer to perform multi-class classification across ten categories for each frame, ultimately yielding the classification results for all frames.

In this model, given the high prevalence of zero labels in the dataset, we employ a weighted Binary Cross-Entropy (BCE) loss to mitigate the effects of data imbalance. Specifically, we assign a weight of 6.45 to the label '1' based on the ratio of zeros to ones. This approach aims to equalize the influence of each class during the training process. The weighted BCE loss is mathematically formulated as follows:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} w_i \left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right] \quad (1)$$

where y denotes the true labels, $\hat{y}$ denotes the predicted probabilities, w_i is the weight associated with each label, and N is the number of samples. This formulation helps in emphasizing the minority class, thereby enhancing model performance.

C. Evaluation Metrics

For a thorough assessment of model performance in a multi-class problem with binary outcomes like [0,1,1,0], where each vector element represents the occurrence of different sounds within a time slice, it is critical to evaluate accuracy, precision, recall, and the F1 score. Each metric not only quantifies distinct aspects of model performance but also provides insights into potential areas of improvement.

- **Accuracy** measures the proportion of true results (both true positives and true negatives) in the total dataset, reflecting the overall effectiveness of the model across all classes. It is defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2)$$

  Although a useful general indicator, accuracy alone can be misleading in the presence of class imbalance, potentially overestimating model performance by favoring the majority class.
- **Precision** assesses the correctness of positive predictions within a predicted class, emphasizing the cost of false positives. It is particularly crucial in applications where the consequence of a false positive is significant. Precision for a class is:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (3)$$

  High precision indicates a low rate of false positive classification, which enhances trust in model predictions.
- **Recall** (or Sensitivity) evaluates the model's ability to correctly identify actual positives from each class, crucial for applications where missing a positive instance is costly. Calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (4)$$

  High recall ensures that the model detects most positive samples, which is vital in critical applications such as medical diagnostics or fault detection in machinery.
- **F1 Score** provides a balanced measure of precision and recall, useful when you seek a model with both high precision and high recall. The F1 score is particularly

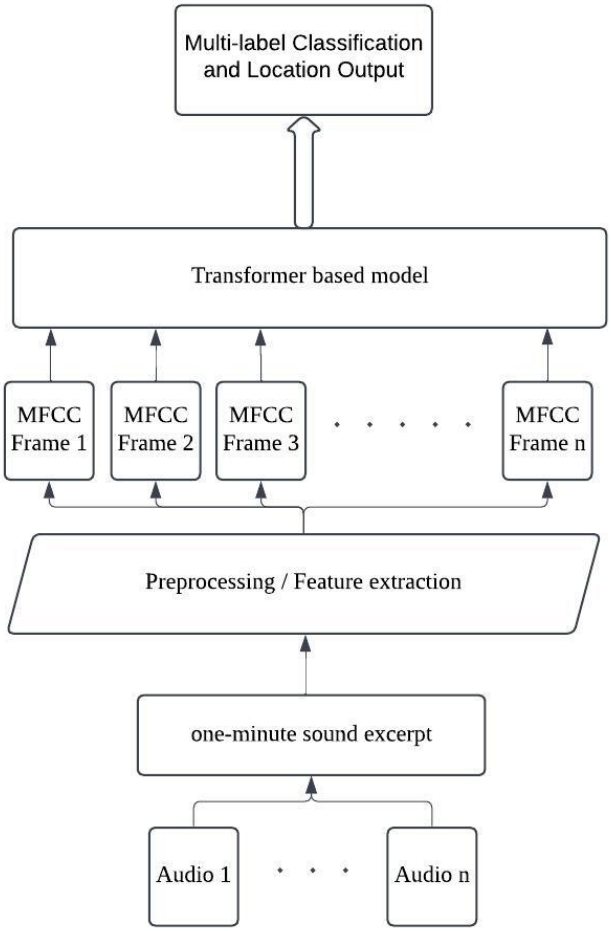

Fig. 1: General Workflow

useful for comparing model performance across various thresholds and when dealing with class imbalances:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

It harmonizes the contributions of precision and recall, offering a single metric to assess model accuracy while considering both the false positives and false negatives.

These metrics are individually important but gain additional significance when considered together. They provide a holistic view of model performance, highlighting strengths and pinpointing specific weaknesses that might not be apparent through a singular focus on accuracy.

IV. Results

Using MFCC features and original features, we trained the model for 200 epochs. Figures 2 and 3 illustrate the training and testing loss curves, respectively. It can be observed that the model trained with MFCC features exhibits a consistent and stable decrease in both training and testing loss, indicating strong generalization capabilities and the ability to effectively distinguish between different sounds. In contrast, the model trained with original features demonstrates significantly poorer performance. From the figures, although the training loss continues to decrease, the testing loss reaches a turning point early in the training process and progressively increases, suggesting the model's limited capacity to learn from original features. This phenomenon is largely attributed to the high noise levels and complex representation inherent in original features.

Figures 4 and 5 further depict the relationship between various evaluation metrics and the number of training epochs. Consistently, the model trained with MFCC features outperforms its counterpart on both the training and testing datasets, underscoring the advantages of MFCC features in enhancing model performance. Table 1 summarizes the final results of the models on the testing set, clearly highlighting the superior efficacy of MFCC features.

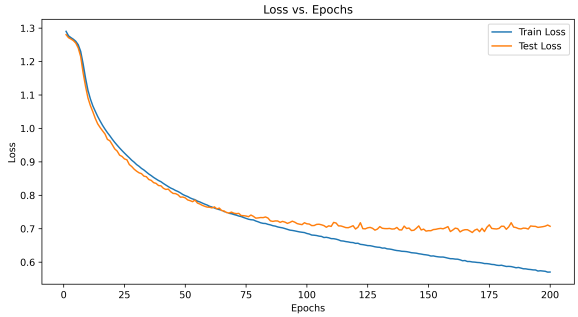

Fig. 2: loss for MFCC features over epochs

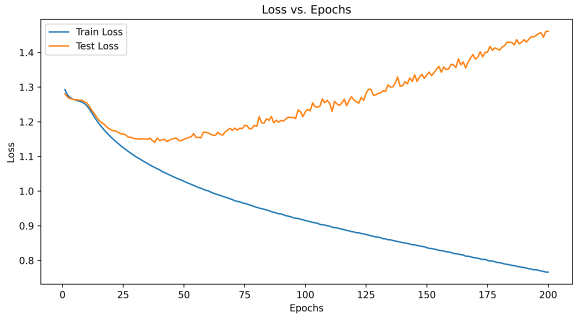

Fig. 3: loss for original features over epochs

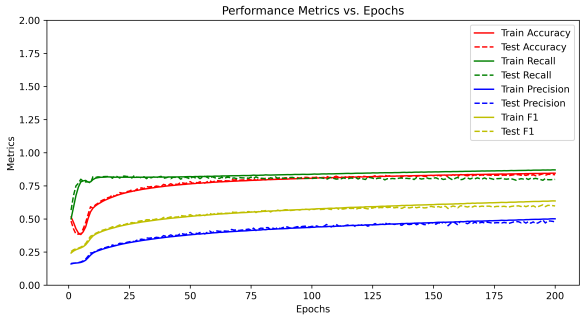

Fig. 4: performance for MFCC features over epochs

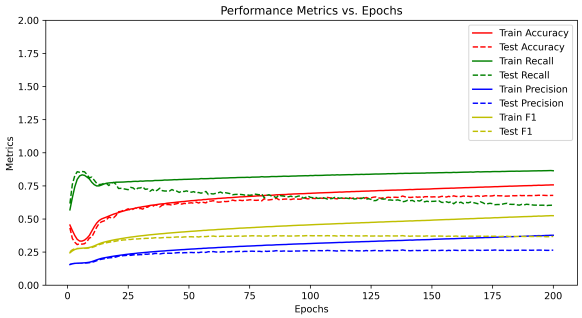

Fig. 5: performance for original features over epochs

DISSCUSSION

The results of this study underscore the significant advantages of leveraging MFCC features and transformer-based architectures for urban sound event detection. Our experiments revealed that models trained with MFCC features achieved superior performance compared to those trained on raw audio data. Specifically, the MFCC-based model attained a test accuracy of 84.07% and an F1 score of 0.6066, while the model using original features achieved only 67.78% accuracy and an F1 score of 0.3678. This substantial improvement highlights the critical role of feature engineering in enabling better generalization and learning efficiency.

The weighted Binary Cross-Entropy (BCE) loss function effectively addressed the class imbalance inherent in the dataset. By assigning higher weights to the minority class, the model was encouraged to learn from underrepresented sound events, thereby mitigating bias toward the majority class. This approach contributed to improved recall values, ensuring that positive instances were more accurately detected.

Our findings emphasize the suitability of MFCC features for urban sound detection tasks. MFCCs provide a compact and structured representation of audio signals, enabling the model to capture key spectral information while filtering out irrelevant noise. In contrast, models trained on original features suffered from overfitting, as indicated by their increasing test loss during training. This discrepancy can be attributed to the high dimensionality and noise within raw audio data, which complicates the learning process.

The transformer architecture played a pivotal role in this study. Its self-attention mechanism enabled the model to capture long-range dependencies within audio sequences, offering a more holistic understanding of sound events. Unlike traditional machine learning approaches that rely heavily on handcrafted features, transformers dynamically extract contextual relationships, making them highly effective for sequential audio data.

While the proposed model delivered promising results, several challenges remain. First, the imbalance in the dataset, despite mitigation efforts, still influenced the overall performance. Future work could explore advanced data augmentation techniques to generate synthetic minority-class samples, further improving model robustness. Additionally, hyperparameter tuning, including adjustments to learning rates and layer configurations, may yield further gains in accuracy and generalization.

In conclusion, this study demonstrates that combining MFCC features with transformer-based architectures significantly enhances urban sound detection. These findings have practical implications for real-world applications, such as smart city monitoring systems, emergency sound recognition, and environmental noise analysis. Future improvements could focus on refining the model's robustness and scalability to ensure reliable performance in diverse, dynamic audio environments.

Feature	Epoch	Test Loss	Test Accuracy	Test Precision	Test Recall	Test F1
MFCC	**200**	**0.7039**	**0.8407**	**0.4915**	**0.7921**	**0.6066**
Original	200	1.4607	0.6778	0.2643	0.6048	0.3678

TABLE I: Test Results Summary

REFERENCES

[1] J. Salamon, C. Jacoby, and J. P. Bello, 'A dataset and taxonomy for urban sound research', in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 1041–1044.

[2] A. Vaswani, 'Attention is all you need', Advances in Neural Information Processing Systems, 2017.

[3] A. Mesaros, T. Heittola, and T. Virtanen, 'TUT database for acoustic scene classification and sound event detection', in 2016 24th European Signal Processing Conference (EUSIPCO), 2016, pp. 1128–1132.

[4] G. Parascandolo, H. Huttunen, and T. Virtanen, 'Recurrent neural networks for polyphonic sound event detection in real life recordings', in 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2016, pp. 6440–6444.

[5] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, 'Large-scale weakly supervised audio classification using gated convolutional neural network', in 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2018, pp. 121–125.

[6] S. Davis and P. Mermelstein, 'Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences', IEEE transactions on acoustics, speech, and signal processing, vol. 28, no. 4, pp. 357–366, 1980.

APPENDIX

SAMPLE OUTPUT

MFCC:

seconds\class	0	1	2	3	4	5	6	7	8	9
10	0	1	0	1	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	1	0	1	0
17	0	0	0	1	0	1	0	0	1	1
18	0	0	0	1	0	1	0	0	1	1
19	0	0	0	1	0	1	0	0	1	1

Original:

seconds\class	0	1	2	3	4	5	6	7	8	9
10	0	1	0	1	1	0	0	0	1	0
11	0	1	0	1	1	0	0	0	1	0
12	1	1	0	1	1	0	0	1	0	1
13	0	1	0	1	1	0	0	0	1	0
14	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	1	0
16	1	0	1	1	1	0	1	0	1	1
17	1	0	1	1	1	0	1	0	1	1
18	1	0	1	1	1	1	0	1	1	1
19	1	1	1	1	1	1	0	1	1	1

Labels:

seconds\class	0	1	2	3	4	5	6	7	8	9
10	0	1	0	1	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0
13	0	1	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	0	1	0
17	0	0	0	1	0	1	0	0	1	1
18	0	0	0	1	0	1	0	0	1	1
19	0	0	0	1	0	1	0	0	1	1