Referring Audio Segmentation

Eesha Shetty[*]
eshetty@andrew.cmu.edu

Marlies Goes[*]
mgoes@andrew.cmu.edu

Noel D Souza[*]
ndsouza@andrew.cmu.edu

Sharang Pai[*]
sharangp@andrew.cmu.edu

Abstract

Audio segmentation is a critical task in several domains, including speech recognition, audio retrieval, and natural language processing. This project aims to address the audio segmentation challenge by utilizing 2D mel-spectrograms, which visually represent waveform audio as images, to improve the accuracy of detecting and localizing specific sound events within an audio clip. By employing advanced image segmentation methods, our objective is to accurately detect and localize specific sound events within an audio clip. To facilitate this, we curated a new dataset derived from the comprehensive BBC Sound Effects Dataset and used it to validate the effectiveness of the "You only hear once" (YOHO) implementation proposed by [13]. Furthermore, we investigated improvements by incorporating advanced embedding models such as VGGish and EfficientNet, as well as experimenting with an attention-inspired masking block. Our experimentation revealed that the EfficientNet architecture outperformed the YOHO approach, demonstrating its potential for enhancing audio segmentation accuracy.

1 Introduction

In recent years, the field of artificial intelligence (AI) and deep learning (DL) has seen significant advancements and has been applied to several domains, including speech recognition, natural language processing, computer vision, and more. One of the critical challenges in this field is to develop models that can understand and interpret multimodal information, such as text, audio, and visual data.

Audio segmentation is a critical task in several domains, including speech recognition, audio retrieval, and natural language processing. In audio retrieval systems, locating specific sound events in audio data is essential for tasks such as audio captioning, summarization, and indexing. Moreover, in scenarios where the audio data is unstructured, such as in user-generated content or live recordings, the task of audio segmentation becomes even more challenging. Audio segmentation also plays a crucial role in speech recognition. Accurate segmentation of audio data is necessary for speaker diarization, which involves identifying who spoke when in an audio recording. Therefore, developing accurate and efficient methods for audio segmentation is crucial for several applications in the field of AI and DL.

The aim of this project is to address the problem of audio segmentation, where given an audio clip and a class of sound events, the model predicts the exact timestamps of the occurrences of the given class in the audio file. Among various methods, we decided to look into the "segmentation by classification" approach. This approach involves dividing audio frames into segments of approximately 2 seconds

[*] Everyone Contributed Equally – Alphabetical order

and classifying each segment based on the characteristics of the audio content. YOLO is a popular object detection framework that is mainly used in image data, but it has gained popularity in audio spectrograms too, where we essentially employ the same bounding box strategy but for visualizations of audio spectrograms. For our project, we try to implement an existing architecture [13] called YOHO (You Only Hear Once) which builds upon YOLO and adapts it to work with audio spectrograms.

In this paper, we first present a comprehensive literature study in the field of audio segmentation and sound event detection (Section 2). Secondly, we describe three model architectures that have been evaluated within this work (Section 3). In Section 4, the conducted experiments and their results are discussed before we conclude our work in Section 5 and elaborate on possible future work in Section 6.

2 Literature Review

We looked into existing implementations of text-based audio retrieval and found a few implementations which we will discuss further.

2.1 Audio-Text Retrieval in Context

The study titled "Audio-Text Retrieval in Context" [6] provides an insightful investigation into audio features and proposes the utilization of sequence aggregation methods to enhance audio-text alignment in the context of retrieval tasks. Through their research, the authors discovered that semantic mapping outperforms temporal relations in achieving effective contextual retrieval.

The methodology employed in the study consists of three key steps. In the first step, audio word embeddings are extracted from the input audio signal and corresponding tokens. Moving to the second step, these embeddings are aggregated using a pooling approach. Finally, in the third step, cosine similarity is computed to quantify the similarity between the audio and text features. To generate the embeddings, the authors leverage two pre-trained models: Word2Vec for extracting text features and Pretrained Audio Neural Networks (PANNs) for audio features. Notably, PANNs are trained on AudioSet, and the feature extraction is performed before the pooling operation.

By integrating these techniques and pre-trained models, the study demonstrates notable advancements in audio-text retrieval. The use of sequence aggregation methods and the combination of Word2Vec and PANNs contribute to enhanced alignment and retrieval accuracy, outperforming the baseline model for AudioCaps and CLOTHO.

2.2 Audio Retrieval with Natural Language Queries: A Benchmark Study

In "Audio Retrieval with Natural Language Queries: A Benchmark Study" [5] the authors present a dataset, SoundDescs, specifically designed for cross-modal audio-text retrieval. This dataset comprises paired sound and natural language descriptions sourced from the BBC Sound Effects DB. The authors also establish benchmarks to evaluate the performance of different approaches in this task.

The research primarily focuses on content-based user-generated data, which typically lacks structured metadata. The objective is to retrieve audio data that aligns with the temporal sequence of events described in the query, rather than a single class tag. To accomplish this, the authors adapt existing frameworks that were initially designed for video retrieval to suit the context of audio retrieval.

Through their evaluation of different embedding frameworks, the authors provide valuable insights into the effectiveness of these frameworks for cross-modal audio-text retrieval. The benchmarks and findings presented in this study serve as a reference for future research and advancements in this domain.

2.3 Wnet: Audio-Guided Video Object Segmentation via Wavelet-Based Cross-Modal Denoising Networks

The research paper titled "Wnet: Audio-Guided Video Object Segmentation via Wavelet-Based Cross-Modal Denoising Networks" [8] addresses the challenging task of audio-guided video object segmentation, which involves automatically separating foreground objects from the background in a video sequence based on audio cues. This paper was selected due to its focus on audio processing techniques and their application in video segmentation.

While most existing methods for referring segmentation primarily rely on text-based expressions, this study explores the use of wavelet-based techniques to generate clean audio-based expressions. The authors argue that audio-guided analysis provides a more precise simulation of human cognition of the world compared to text-guided approaches [10].

The proposed approach in the paper involves representing the audio expressions using a 39-dimensional Mel-frequency cepstral coefficients (MFCC) representation. Visual features extracted from the videos using a ResNet-50 backbone are combined with the audio features in a model that incorporates a 2-layer, 8-head multi-head cross-attention module with a width of 3. This fusion of visual and audio features aims to enhance the segmentation performance.

This paper establishes a baseline for state-of-the-art pre-processing techniques in audio and provides insights into potential techniques for feature extraction in audio segmentation. The findings presented in this study contribute to advancing the understanding and development of audio-guided video object segmentation techniques.

2.4 Automated Audio Captioning with Recurrent Neural Networks

One of the early approaches to automated audio captioning is presented in "Automated audio captioning with recurrent neural networks" [2]. The model proposed in this paper adopts an encoder-decoder architecture with an alignment model in between.

In this approach, the encoder takes a sequence of log mel-band energies computed from an audio file as input, while the output is a sequence of words, representing a caption. The encoder is a multi-layered, bi-directional gated recurrent unit (GRU), which captures the temporal dependencies in the audio features. The decoder, also a multi-layered GRU, generates the caption based on the encoded audio representation. The decoder is equipped with a classification layer that is connected to the last GRU layer, enabling the generation of text expressions. The alignment model, along with the classification layer, consists of fully connected layers with shared weights across timesteps, enhancing the alignment between the audio and the generated captions.

Although the focus of this paper is not specifically on audio segmentation, it establishes a foundational approach for captioning or generating textual expressions using recurrent neural networks. This paper serves as a benchmark for future research in the field of audio captioning and has contributed to the development of datasets such as CLOTHO, which are crucial for advancing audio-related tasks.

2.5 You Only Hear Once: A YOLO-like Algorithm for Audio Segmentation and Sound Event Detection

The research paper titled "You Only Hear Once: A YOLO-like Algorithm for Audio Segmentation and Sound Event Detection" [13] presents an innovative approach to audio segmentation and sound event detection. While the majority of recent work in this field follows a segmentation-by-classification paradigm, the authors draw inspiration from the YOLO (You Only Look Once) algorithm [9] used in computer vision and propose a similar framework for audio analysis.

In their approach, the authors transform the audio segmentation problem from classification to regression. The model architecture includes separate output neurons for classifying audio events and predicting the start and end points of each event. By adopting this regression-based approach, the authors achieve notable improvements of 1-6% on various evaluation metrics compared to existing segmentation and sound event detection baselines. Additionally, the proposed algorithm significantly reduces inference time by approximately 6 times compared to traditional classification-based methods.

This paper serves as an interesting example of how simplifying the audio segmentation process using regression-based approaches can yield fast and efficient results. In this project, we aim to adapt and improve the YOHO algorithm proposed in the paper.

3 Method

3.1 Explored Models

We explored three different models for our task of audio segmentation. The baseline model utilized a MobileNet backbone with YOHO blocks. In our experiments, we tested a VGGish backbone and an EfficientNet backbone with a masking block. The implementation of these three architectures is described in the following section.

3.1.1 MobileNet Backbone with YOHO Blocks

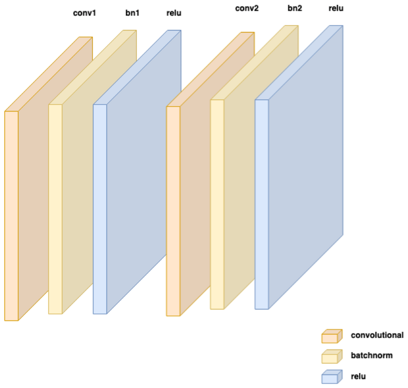

Figure 1: Baseline Architecture: YOHO Block

The model employed in the baseline architecture is a modified MobileNet [4] convolutional neural network, specifically adapted to generate both regression and classification outputs. The so-called *YOHO blocks* within the model employ a combination of depth-wise and pointwise convolution layers (Figure 1). Following each convolutional block, a Batch Normalization layer is applied to normalize the activations, followed by a Rectified Linear Unit (ReLU) activation function to introduce non-linearity. The model expects input data of shape (257x40), where the first dimension represents the temporal axis, and the second dimension represents the frequency bins. The model produces an output of shape (9x18), where the first dimension denotes the temporal axis, and the subsequent 18 values represent class, start, and end predictions for each of the six classes. An overview of this baseline architecture is shown in Figure 2.

3.1.2 EfficientNet Backbone with Masking Block

As part of our research, we chose to replace the MobileNet backbone with the EfficientNet architecture [12] (Figure 3). EfficientNet is specifically built for a lower amount of computing power and hence might perform better specifically for the type of dataset we gave it as compared to the YOHO model. In addition, we introduced a masking block to the backbone's output. This block consists of a series of convolutional layers with a kernel size of 1. The purpose of this masking block is to incorporate a masking mechanism into the model. By feeding the backbone's output through this module, the model learns to emphasize certain features. The output is then multiplied by the backbone output before being passed to the network's head. This approach allows the model to focus on relevant features when making class predictions and performing the start and end regressions. The integration

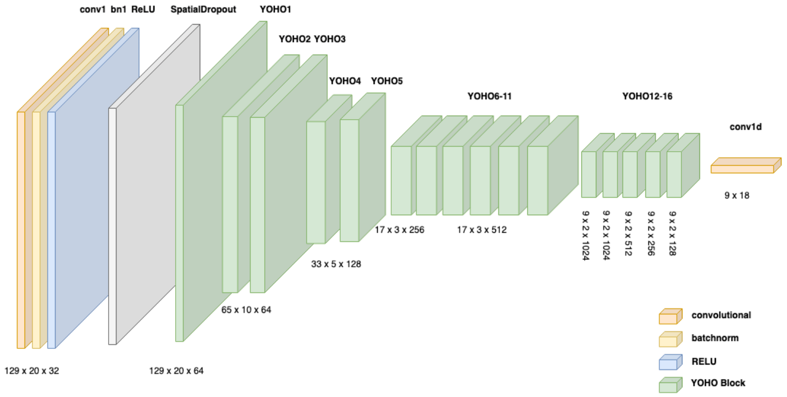

Figure 2: Baseline Architecture

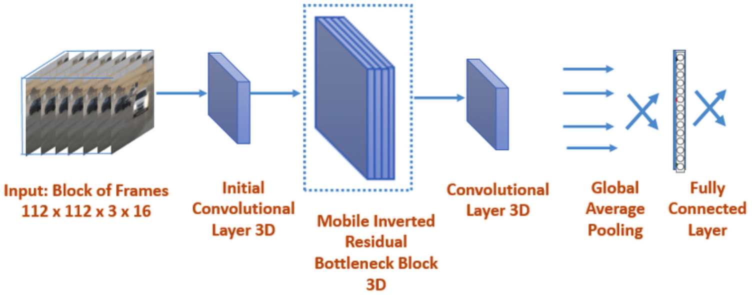

Figure 3: Modified Architecture: EfficientNet Block

of this masking mechanism aims to enhance the model's performance by directing its attention to the most important information.

3.1.3 VGGish Backbone

VGG [11], a convolutional neural network widely recognized for its effectiveness in object recognition systems, was a compelling candidate for our audio feature extraction needs. Leveraging the TensorFlow implementation of VGGish [3], an audio adaptation of VGG, we attempted to incorporate this architecture into our dataset. Our adaptation of VGGish deviated from the standard configuration, as we replaced the final fully connected layers with our custom reshaping and Conv1D layers, aligning the output format with the requirements of our model. While this modified VGGish model demonstrated satisfactory performance, it did not surpass the performance of our established baseline model, see the experiment results in Section 4.4.1.

3.2 Loss Function

Instead of employing a cross-entropy loss commonly used in classification tasks, the model is trained using a *sum squared distance* loss function. The loss function, depicted in Figure 1, quantifies the discrepancy between the predicted outputs and the ground truth values of all three learning objectives:

- The classification y_1: Is this class present in the current sound file?
- The regression of the start y_2: If this class is present, when does it start?
- The regression of the end y_3: If this class is present, when does it end?

Whenever the classification $y_1 = 0$ (the sound event is not present), the other outputs of the model are not representative and therefore not used in the loss calculation, as shown in Equation 1. y_i represents the ground truth, and $\hat{y}_i$ represents the prediction for each model output i.

$$\mathcal{L}_c = (\hat{y}_1 - y_1)^2 + y_1(\hat{y}_2 - y_2)^2 + y_1(\hat{y}_3 - y_3)^2 \tag{1}$$

This loss is then summed over all classes c (Equation 2).

$$\mathcal{L} = \sum_c \mathcal{L}_c \tag{2}$$

3.3 Evaluation Metrics

The evaluation of the model is conducted using the F1 score, a widely adopted metric that offers a reliable assessment of the model's performance relative to baselines. The F1 score strikes a balance between precision and recall, providing valuable insights into the model's ability to correctly identify positive instances while minimizing both false positives and false negatives. The F1 score is calculated as shown in Equation 3.

$$F_1 = \frac{precision \cdot recall}{precision + recall} \tag{3}$$

where precision is determined by the ratio of true positives (TP) to the sum of true positives and false positives (TP + FP), and recall is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (TP + FN). By utilizing these components, the F1 score effectively captures the model's performance by considering both the precision and recall aspects, thus enabling a comprehensive evaluation of its efficacy.

4 Experiments and Results

4.1 Building the Dataset

The primary data source for our project was the BBC Sound Effects Dataset [1], which served as the foundation for our research. However, the dataset required preprocessing to align with our specific objectives since it initially comprised audio clips containing only a single class of sound events. To enable the segmentation of audio by class, it was necessary to transform the dataset by breaking down the original audio clips into shorter segments of 10 seconds or less. Subsequently, we combined these segments in a pseudo-random manner, resulting in the creation of mixed audio files. This process yielded audio files ranging from 55 seconds to 127 seconds in duration, enabling us to focus on the segmentation of ambient sound. From the collection of available classes, we identified and selected the six most relevant classes (rainforest, cars, crowds, footsteps, clocks, and aircraft) to be modeled in our segmentation framework.

4.2 Data Processing

In order to process the audio data, we utilized the *librosa* library [7] to convert the audio signals into log mel-spectrograms. A range of parameter configurations was explored, ultimately leading us to segment the audio into windows of 2.56 seconds in length, which were then transformed into 40 mel-bands. This transformation yielded a representation of each mel-spectrogram as a vector of size 257x40. These transformed vectors served as the input data for our model, enabling predictions to be generated effectively.

To enhance the granularity of our model, we also applied preprocessing steps to the input labels. In the initial implementation by the authors of the YOHO-paper [13], the labels were provided as plain text files, with each file containing the occurrence of the ambient sound along with its corresponding start and end times. To facilitate further analysis, we devised a processing methodology to transform each 2.56-second window into a processed vector with dimensions of d x (num classes x 3). Here, d represents the number of divisions within the clip, which was set empirically to 9. Given the presence of six classes within our dataset, and each class has its binary label occurrence (0 or 1), start time, and end time, the resulting output vector was of size 9x18. During the inference phase, these vectors were combined to reconstruct the predicted text file, consolidating the information for each audio file.

By incorporating these data processing techniques, our model was equipped with more comprehensive information and finer-grained details, enabling it to make more nuanced predictions and enhance the overall performance of our system.

4.3 Experiments and Results

4.3.1 Overall Performance

After finetuning the MobileNet/YOHO baseline, VGGish, and EfficientNet models, we trained the three models for 100 epochs.

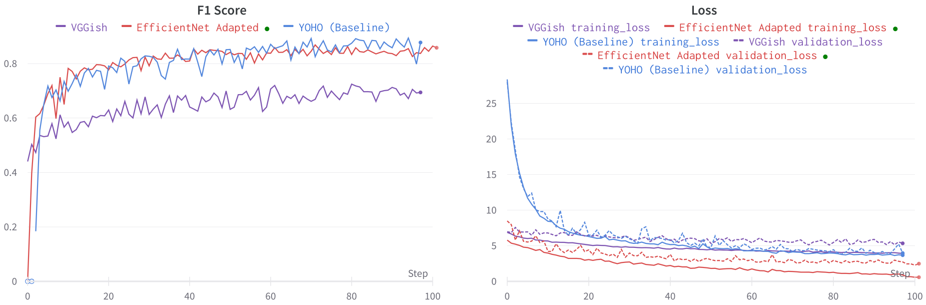

Figure 4: Training Metrics for the three Models over 100 Epochs

Figure 4 presents the training metrics over 100 epochs. The left plot illustrates the F1 score for each of the evaluated models, while the right plot shows the respective curves for training and validation loss. It is evident that the VGGish implementation exhibits notably inferior performance compared to the other two models. The EfficientNet adaptation demonstrates comparable F1 scores to the YOHO baseline implementation while exhibiting a lower loss. The F1 score solely considers the classification aspect of the predictions, whereas the loss incorporates both the classification and regression components (see Section 3.2). This suggests that the two models perform similarly in terms of classification, but the EfficientNet model excels in predicting the start and end timestamps.

4.3.2 Classwise Evaluation

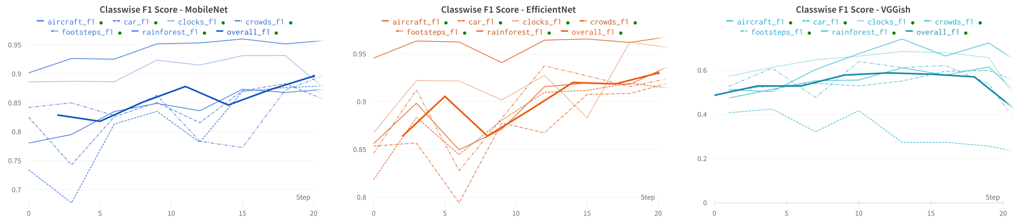

Figure 5: Classwise Metrics for the three Models over 20 Epochs

To gain deeper insights into the model's predictions, we recorded and analyzed class-wise metrics during the initial 20 epochs of training, displayed in Figure 5. The results demonstrate a generally well-balanced performance across the six classes. Specifically, the predictions for rainforest sounds exhibit slightly higher accuracy compared to the other classes, whereas the classification of car sounds shows relatively lower precision. These observed discrepancies in accuracy could potentially be further investigated and addressed to improve the overall performance of the model.

4.3.3 Masking Layer Evaluation

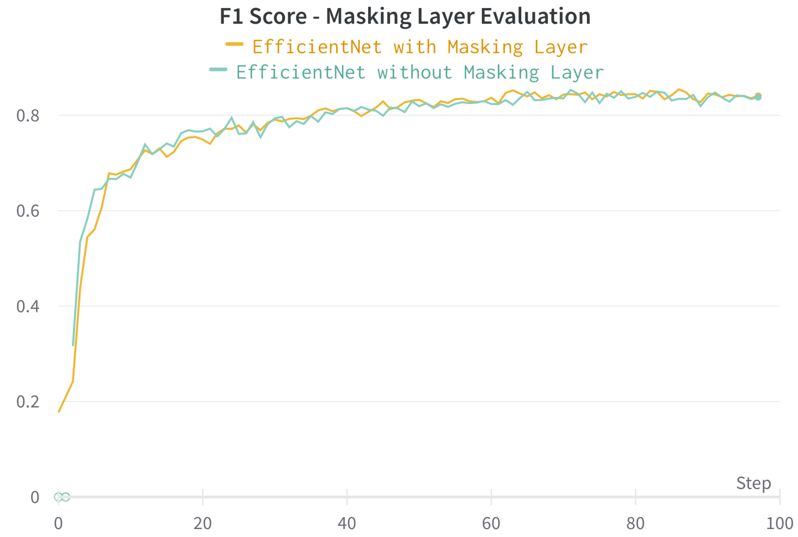

Figure 6: Comparison of the F1 score with and without the Masking Layer

In order to validate the effectiveness of the masking layer, we ran several ablations with our masking approach and compared it to no mask baselines. This included varying learning rates between 1e-4 and 1e-3. We also experimented with a step scheduler. Finally, we tried moving the masking layer further and closer to the head of the network. In none of these experiments did we see any statistically significant results, either in terms of F1 score, loss, or convergence time. Figure 6 shows the F1 score over a training run of 100 epochs with a learning rate of 1e-4.

4.4 Discussion

4.4.1 VGGish

Several factors may have contributed to the relatively lower performance of the VGGish model. Firstly, the absence of the Mobile Inverted Bottleneck Convolution, a crucial component found in the EfficientNet and YOHO baseline architectures, might have limited the VGGish model's ability to capture intricate audio representations effectively. Additionally, as VGGish represents an older architecture, its performance could be influenced by the specific characteristics of the input audio data, the size and diversity of the training dataset, and the targeted classification task.

Furthermore, our experiment employed a relatively smaller training dataset, consisting of approximately 1000 audio clips, each with an average duration of two minutes. This constrained dataset size may have impacted the VGGish model's performance, as it might have benefited from a larger and more diverse training dataset. These observations and hypotheses highlight the need to consider the interplay between model architectures and dataset characteristics when striving to achieve optimal performance in audio classification tasks.

4.4.2 EfficientNet+

As described in Section 3.1, we replaced MobileNet with EfficientNet and added a masking block. This architectural modification yielded superior results compared to our baseline model. After conducting a comparison between the models with and without the proposed masking layer, it became apparent that the observed improvement in overall performance can be attributed primarily to the utilization of the EfficientNet architecture. The addition of the masking layer did not exhibit a significant impact on the model's performance.

The masking block comprises a series of convolutions, each characterized by a kernel size of 1. By multiplying the output of this block with the embeddings derived from the EfficientNet backbone, we aimed to emphasize the importance of specific embeddings that significantly contributed to the tasks of regression and classification. This mechanism was supposed to "highlight" the most salient features, leading to improved performance in the regression task, which our experiments, unfortunately, didn't confirm.

Our choice to adopt the EfficientNet architecture seems to have improved the overall performance of the model. Our choice was motivated by the demonstrated effectiveness of the EfficientNet in achieving high performance with a reduced parameter count. Given the relatively smaller size of our training dataset, the prospect of attaining comparable or superior results while employing fewer parameters was particularly appealing. By incorporating EfficientNet as our backbone, we aimed to leverage its capabilities for our specific task, ultimately yielding improved model performance.

5 Conclusion

In our experiments, we evaluated the performance of three models (VGGish, EfficientNet, and YOHO baseline) for audio segmentation and sound event detection using the BBC Sound Effects Dataset. Preprocessing was applied to align the data with our objectives, and log mel-spectrograms were used as input. The VGGish model showed inferior performance compared to EfficientNet and YOHO, likely due to missing components in its architecture. EfficientNet achieved comparable F1 scores to YOHO but with lower loss, indicating its proficiency in predicting start and end timestamps. Class-wise evaluation demonstrated generally balanced performance across the six classes, with slight variations. The masking layer evaluation did not yield significant improvements, but the adoption of the EfficientNet architecture significantly enhanced overall model performance compared to the baseline, showcasing its effectiveness in achieving high performance with fewer parameters.

In conclusion, the experiments and results highlighted the importance of model architecture and dataset characteristics in audio segmentation and sound event detection tasks. The EfficientNet model, along with appropriate preprocessing techniques, demonstrated superior performance in capturing audio representations and making accurate predictions. These findings provide valuable insights for future research in audio analysis and encourage further exploration of architectural modifications and dataset augmentation to enhance performance in this field.

6 Future Work

Future work in this field could explore alternative representations of audio data beyond Mel spectrograms. Different feature extraction techniques, such as wavelet transforms or time-frequency representations, could be investigated to capture more intricate details and improve the performance of audio segmentation and sound event detection models.

Additionally, incorporating natural language queries as input instead of being limited to a predefined set of sound event classes could be an interesting avenue for future research. This would involve developing techniques to parse and interpret natural language expressions related to audio, allowing users to describe the desired sound events or characteristics they are looking for. Integrating natural language processing and audio analysis could enable more user-friendly and flexible systems for audio segmentation and sound event detection.

Moreover, while classification-based approaches have shown promising results in the context of ambient sound segmentation, further exploration of combining these methods with traditional audio captioning approaches could be beneficial. By leveraging the strengths of both approaches, it might be possible to achieve more accurate and descriptive segmentation of audio, providing richer information about the content and context of sound events. This could have applications in areas such as multimedia content analysis, automatic video editing, and interactive sound design.

A Appendix

All the code written for this project is available in our GitHub repository: `https://github.com/Guzzler/IDL_Project_Audio_Segmentation`

Demo code to test the model: `https://github.com/Guzzler/IDL_Project_Audio_Segmentation/tree/main/demo/`

Given an audio waveform as input, the model outputs the classes identified along with the time segments. A sample output file would look like this: `https://github.com/Guzzler/IDL_Project_Audio_Segmentation/blob/main/demo/sample-mixed-audio-label.txt`

References

[1] Bbc sound effects library. Audio dataset, 2023. URL `https://sound-effects.bbcrewind.co.uk/search`. Accessed 04/26/2023.

[2] Konstantinos Drossos, Sharath Adavanne, and Tuomas Virtanen. Automated audio captioning with recurrent neural networks. *CoRR*, abs/1706.10006, 2017. URL `http://arxiv.org/abs/1706.10006`.

[3] Shawn Hershey, Sourish Chaudhuri, and Daniel PW Ellis. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017. URL `https://arxiv.org/abs/1609.09430`.

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL `http://arxiv.org/abs/1704.04861`.

[5] A. Sophia Koepke, Andreea-Maria Oncescu, Joao Henriques, Zeynep Akata, and Samuel Albanie. Audio retrieval with natural language queries: A benchmark study. *IEEE Transactions on Multimedia*, pages 1–1, 2022. doi: 10.1109/tmm.2022.3149712. URL `https://doi.org/10.1109%2Ftmm.2022.3149712`.

[6] Siyu Lou, Xuenan Xu, Mengyue Wu, and Kai Yu. Audio-text retrieval in context. pages 4793–4797, 2022. doi: 10.1109/ICASSP43922.2022.9746786.

[7] Brian McFee, Colin Raffel, Dawen Liang, and Daniel PW Ellis. Librosa: Audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*, 8:18–25, 2015. URL `https://librosa.org/`.

[8] Wenwen Pan, Haonan Shi, Zhou Zhao, Jieming Zhu, Xiuqiang He, Zhigeng Pan, Lianli Gao, Jun Yu, Fei Wu, and Qi Tian. Wnet: Audio-guided video object segmentation via wavelet-based cross- modal denoising networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1310–1321, 2022. doi: 10.1109/CVPR52688.2022.00138.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. URL `http://arxiv.org/abs/1506.02640`. cite arxiv:1506.02640.

[10] Ramon Sanabria, Austin Waters, and Jason Baldridge. Talk, don't write: A study of direct speech-based image retrieval. In *Interspeech*, 2021.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. URL `https://arxiv.org/abs/1409.1556`.

[12] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/tan19a.html`.

[13] Satvik Venkatesh, David Moffat, and Eduardo Reck Miranda. You only hear once: A yolo-like algorithm for audio segmentation and sound event detection. *Applied Sciences*, 12(7), 2022.