Audio Segmentation with U-Net architecture: Ablation study of Depth Variations in 1-D and 2-D Audio Representations

Andrew Romero
Stanford University; EE367 Computational Imaging

Abstract—In this project, I explore using the U-Net convolutional neural network architecture for Music Information Retrieval (MIR), focusing specifically on the task of audio source separation. The ability to separate audio into sources allows for applications like noise removal, music re-mixing, and large dataset creation for controllable audio generation. Leveraging the U-Net model, used for many image segmentation tasks in the medical and self-driving technology field, should succeed at the similar task of audio segmentation. This investigation compares the performance of U-Net architectures trained on spectrogram images against those trained on raw audio data. Through implementing, training, testing, and evaluating two distinct U-Net model types on the same dataset, I seek to determine the optimal approach for audio source separation in MIR. This paper presents efforts to find and tune the best U-Net architecture for the task, considering model accuracy, training time, and compute resources.

Index Terms—Source Separation, U-Net Architecture

1 Introduction

In the field of Music Information Retrieval (MIR), the challenge of effectively separating audio sources from complex mixtures still has no obvious solution. In this project, I explore two possible techniques based on the U-Net architecture. The first uses a 1-dimensional U-Net that works directly on audio files in the time domain. The second uses a 2-dimensional U-Net, originally used with standard images, to segment audio using spectrogram data. There are two main qualities of the U-Net I explore and compare in this paper: number of layers (depth) and dimension of the U-Net model. Section 2 discusses the background of these models. Section 3 covers the theory behind the tools used to implement this project. I use the musdb18 dataset to train and test the network, as described in section 4 of this paper Results and discussion are presented in sections 5 & 6.

2 Related Work

The U-Net architecture, presented by Ronneberger et. al. [1], revolutionizes biomedical image segmentation through its unique design that combines an encoder and decoder type of convolutional neural network (CNN) for pixel classification. By using data augmentation, the network achieves high performance with very few annotated images, outperforming prior methods on tasks like segmentation of neuronal structures in electron microscopy and cell segmentation in light microscopy. The U-Net is fully convolutional and utilizes a tiling strategy to handle large images, avoiding the limitations imposed by GPU memory. This architecture's effectiveness is further enhanced by a novel weighted loss function for separating touching objects and an initialization strategy that ensures uniform variance across the network's layers. With its ability to be trained end-to-end from very few images and its fast operation, U-Net sets a new standard for biomedical image segmentation, as demonstrated by its success in multiple segmentation challenges.

Oh et. al. [2] introduces the Spectrogram-Channels U-Net, a novel adaptation of the U-Net architecture designed for the task of sound source separation, leveraging spectrogram-based modeling to enable direct separation of multiple audio sources simultaneously. Unlike traditional approaches that produce a mask to be applied to the original audio for source separation, this model outputs channels that correspond to the spectrograms of the individual sources, facilitating both singing voice and multi-instrument separation by merely adjusting the output channel count. To address the challenge of volume discrepancies between sources, a weighted loss function is proposed, promoting balanced separation across differently loud sources. The model demonstrates state-of-the-art performance in source separation tasks without relying on extensive data augmentation or post-processing techniques, showcasing its efficacy with the musdb18 dataset. This dataset is used in this project and is discussed in more detail below.

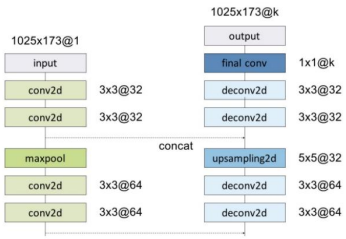

Fig. 1: 2D U-Net model [2]

The Wave-U-Net [3], which I also refer to as the 1D U-Net model, is a novel multi-scale neural network designed for end-to-end audio source separation, directly operating on time-domain audio signals to model both magnitude and phase information. It adapts the U-Net architecture to the one-dimensional time domain, utilizing a series of downsampling and upsampling layers to capture and combine features across different time scales, addressing long-range temporal dependencies inherent in audio data. Architectural improvements, including a difference output layer for enforcing source additivity, linear interpolation for upsampling, and a context-aware prediction framework, are introduced to enhance performance and reduce artifacts. Experiments demonstrate that the Wave-U-Net achieves comparable or superior performance to state-of-the-art spectrogram-based models for singing voice separation, with further applications to multi-instrument separation. The research highlights the importance of providing proper input context and addresses issues with traditional SDR evaluation metrics by proposing rank-based statistics as an alternative. This contributes to my hypothesis that 1D U-Net will perform better, subjectively and objectively, than the 2D U-Net model.

3 Theory

In this section, I explain the tools used to develop this audio segmentation pipeline.

3.1 Short-Time Fourier Transform

$$STFTx(n)(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)e^{-j\omega n} \quad (1)$$

The Short-Time Fourier Transform (STFT) is a technique used to analyze the frequency content of signals over time. It divides a longer time signal into shorter segments of equal length and applies the Fourier Transform to each segment, capturing the frequency spectrum at different time intervals. This process allows for the observation of how the frequency components of a signal vary over time, making STFT useful for time-frequency analysis in various applications, including audio processing and speech recognition. The result of an STFT is often visualized in a spectrogram, which displays the intensity of frequencies present in the signal at each time segment. The resolution of the analysis in both time and frequency domains can be adjusted by changing the window size and the overlap between consecutive segments.

3.2 U-Net architecture

The U-Net architecture is a convolutional neural network initially designed for biomedical image segmentation, featuring a symmetric structure with a contracting path to capture context and a symmetric expanding path that enables precise localization. Its design uses skip connections between layers of equal resolution in the contracting and expanding paths, which helps to recover spatial information lost during downsampling. When adapted for audio source separation, the U-Net operates on spectrograms of audio signals, utilizing its capacity for fine-grained detail preservation to regressively predict the constituent sources of a mixed signal. By treating each channel of the network's output as the spectrogram of a separate source, the U-Net can effectively disentangle and reconstruct individual audio components from a complex mixture. However, the performance of U-Net in audio source separation can significantly depend on the choice of loss function, with Mean Squared Error (MSE) being common but not necessarily optimal, as it may not fully capture perceptual aspects of audio quality.

3.3 Loss function

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \quad (2)$$

The Mean Squared Error (MSE) measures the average squared difference between the estimated values and the actual value, making it a widely used metric for regression tasks. Its utilization as a loss function in training models like the U-Net architecture, especially on spectrogram data, stems from its ability to quantify the discrepancy between the predicted output and the ground truth across the dataset. MSE's emphasis on squaring the errors prioritizes larger discrepancies, encouraging the model to minimize these errors during training, which is crucial for improving the accuracy of source separation in spectrograms. However, MSE might not always be the best choice for loss function in audio processing tasks because it treats all errors equally without considering the perceptual aspects of sound, potentially leading to a model that optimizes for numerical accuracy rather than perceptual quality. Therefore, exploring alternative loss functions that better capture human auditory perception could lead to models that perform more effectively in real-world audio separation tasks. A score based on subjective opinion, perhaps stored in a neural network, would be optimal for a loss function parameter that allows the model to maintain high audio quality.

4 Analysis and Evaluation

In this section, the dataset and evaluation methods are explained, as well as a discussion on the hardware configurations for each model.

4.1 MUSDB18 Dataset

The MUSDB18 dataset contains 150 full-length music tracks, with 10 hours of audio across various genres, designed for the task of music source separation. It is organized into two main subsets: a training set with 100 songs and a test set with 50 songs, to facilitate the development and evaluation of supervised music separation algorithms. All tracks are provided in stereo format with a standard sampling rate of 44.1kHz, and include isolated stems for drums, bass, vocals, and other instruments, designed for separation tasks. MUSDB18 is available in the Native Instruments stems format, comprising five stereo streams encoded as AAC @256kbps, including a mix and individual stems for each component, though the separate encoding of the mixture and stems may lead to slight discrepancies due to compression. [4] [5]

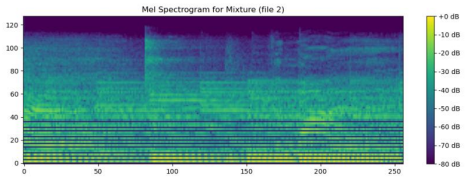

Fig. 2: 2D U-Net input: Mixture of Audio

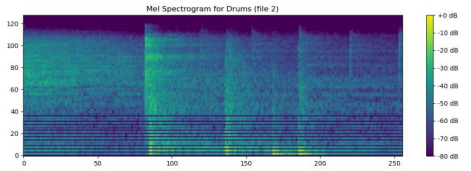

Fig. 3: 2D U-Net Label: True Drums

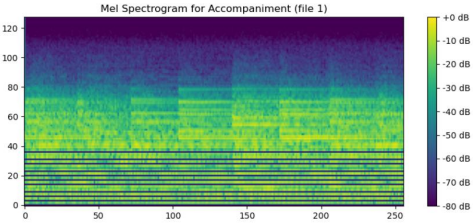

Fig. 4: 2D U-Net Label: True Accompaniment

The above three figures 2, 3 and 4 are some examples of what an input and some of its labels look like. For the sake of the larger 2D U-Net and my limited GPU size, I had to limit my audio to 3 second clips. This is where the data changes between the 1D and 2D U-Net, making the comparison somewhat inaccurate. This flaw is discussed in the future work section of this paper.

A significant portion of this project was learning about and implementing a custom dataset class for processing these spectrograms, which had to come from the lower-quality musdb18 dataset.

4.2 1D Wave U-Net with 3 layers: Experiment

For this first experiment, ran on Google Colab using a V100 GPU, I run the Wave-Net modelwith three layers using the high quality audio. This is possible because the model size is small enough where better data can be used and stored in the same GPU memory. To evaluate this test, I calculate the MSE with respect to the label and compare the subjective audio quality to that of the other experiments. I also fix the training time to three hours to set a fixed timeframe for all models to run.

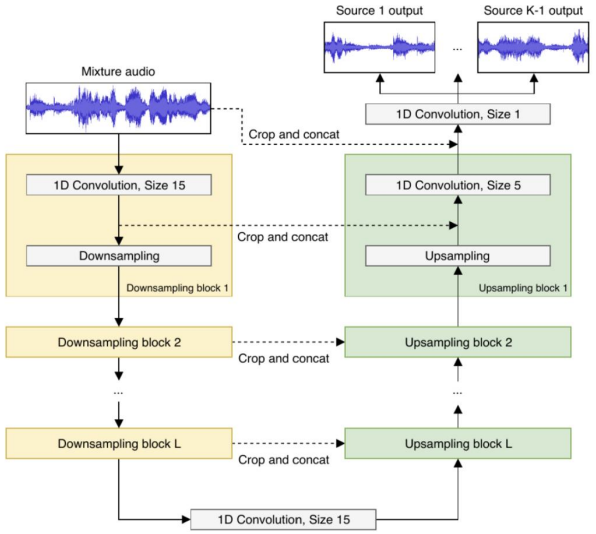

Fig. 5: 1D U-Net model [3]

4.3 1D Wave U-Net with 6 layers: Experiment

This test is the same as above, but 6 layers are used instead of 3. This made the model much larger and complete less epochs in the same amount of training time.

4.4 2D Wave U-Net with 2 layers: Experiment

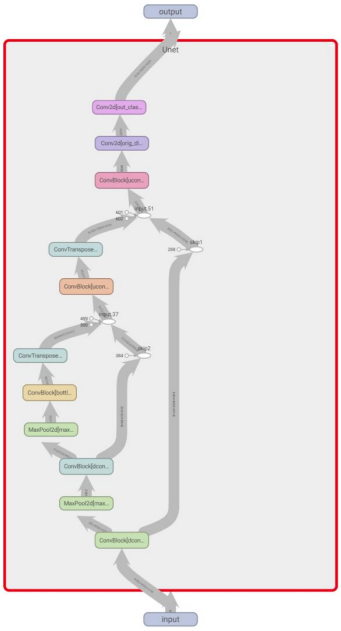

Fig. 6: My implementation of the 2D U-Net model

The bulk of the time spent on this project is here on the 2D U-Net model. I tried to normalize the audio and storing the mean and std of each set of instruments in the batch. I train this model, with two layers, on my laptop using

an RTX 3060 GPU. I also train this model for three hours to see, at some point, how the results compare to three hours with a 1D U-Net. Since the results of this experiment are incomplete as far as generating audio, I present the loss curve and generated spectrograms to show successful progress of implementation of this pipeline. I also show some preliminary but incorrect audio examples.

5 Results

Here I show the results of the three main trainings completed for this project. The results are subjective and quantitative.

5.1 Comparing 1D U-Net depth quality

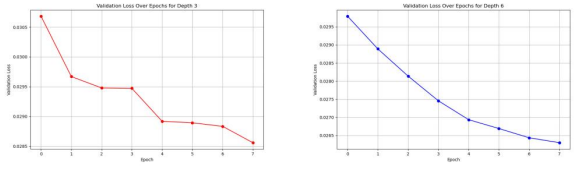

Fig. 7: Comparing loss functions - layers of 1D U-Net

MSE for 3-Layer U-Net	0.00166
MSE for 6-Layer U-Net	0.00149

TABLE 1: Comparing MSE of two model depths

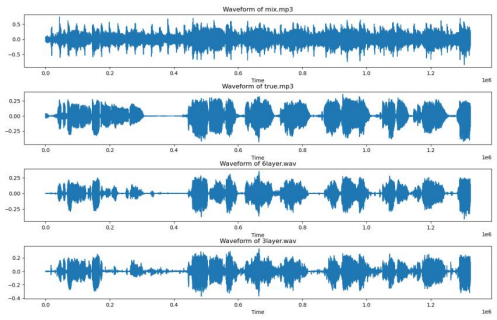

Fig. 8: Audio generations and comparison to mix and truth

5.2 Comparing 2D U-Net and 1D U-Net

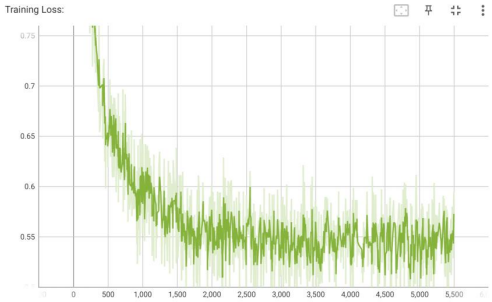

Fig. 9: Loss over iteration for training my 2D U-Net model

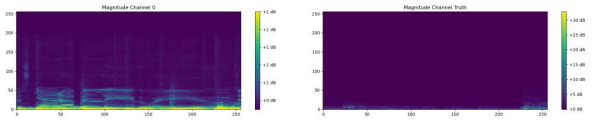

Fig. 10: Generated vs truth spectrogram

Unlike the 1D U-Net, the 2D Unet is not capable of generating recognizable audio yet. Therefore, it is not possible to compare the two networks quantitatively. Even if the audio was decent, there are changes in the way audio is normalized and pre-processed in each case so a true comparison would require many ablation-like exams to see why quality differences exist.

6 Discussion

In the first experiment, my hypothesis is confirmed both by a subjective listen and a quantitative MSE comparison. This impressive result for a model that's much more simple than the 2D U-Net should be taken into account by those trying to find an optimal network for their audio segmentation needs.

It was bold to generate spectrograms regressively with minimal control and regularization in the later layers of the network. I initially observed that the predicted values were extremely large, which is a sign of exploding activations or fault in normalization. Another issue is the quality of the audio generated, which I show with audio files generated from my decoding pipeline. The good news is that the generated spectrogram looks a lot like correct audio. Comparing 10 to 4, the wave forms appear quite similar.

7 Conclusion and Future Work

Although the 2D U-Net was able to train on and generate realistic spectrograms, I have yet to provide a side-by-side subjective comparison due to the way I processed data for each model type. This is a project I will attempt again from scratch, beginning with a strict and standard audio pre-processing step. Although there exist many powerful libraries for processing audio, there is no agreed upon standard for preparing and normalizing audio for general audio ML applications. I would also change the way I calculate the loss function, considering each individual channel instead of the entire output block. As I mention above, implementing a subjective regularizer into the loss function would keep the audio quality high, which is worth exploring in future work.

Another experiment I'd like to run on the 2D U-Net is using phase vs no phase in the dataset. How significant is the phase information? The 1D U-Net inherently keeps phase by just working with the audio in time domain. This may have to do with why it performs so well, and it's possible that a compressed version of the STFT is created within the convolutional layers of the 1D U-Net.

8 Acknowledgements

Special thanks to Ashwin Alinkil for the U-Net guidance and compute resources.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[2] J. Oh, D. Kim, and S. Yun, "Spectrogram-channels u-net: a source separation model viewing each channel as the spectrogram of each source," *CoRR*, vol. abs/1810.11520, 2018. [Online]. Available: http://arxiv.org/abs/1810.11520

[3] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *CoRR*, vol. abs/1806.03185, 2018. [Online]. Available: http://arxiv.org/abs/1806.03185

[4] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[5] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-hq - an uncompressed version of musdb18," Aug. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373