Tampere University

Petteri Nuotiomaa

VISUALLY ATTENDED SOUND SOURCE SEMANTIC SEGMENTATION

Forming a semantic mask using visual and audio cues

Bachelor's Thesis
Faculty of Information Technology and Communication Sciences
March 2024

ABSTRACT

Petteri Nuotiomaa: Visually attended sound source semantic segmentation
Bachelor's Thesis
Tampere University
Information technology
March 2024

Sound Source Separation from single input audio mixture is a problem that has had many different proposed solutions. Many of the modern solutions involve using machine learning and especially deep learning to achieve good separation results. The goal of this thesis is to develop a solution to the sound source semantic segmentation problem, evaluate the performance of the proposed method, and compare it with similar solutions

The thesis goes over the theory behind the problem and explores some of the previous solutions used for this problem. During the background research, it became clear that there aren't many solutions trying to solve the exact problem of sound source semantic segmentation. This is why the thesis examines solutions for different problems, that have some relation to the sound source semantic segmentation.

This thesis proposes a new solution for sound source semantic segmentation. The solution is based on some of the groundbreaking research done in the sound source separation field. The proposed deep learning system has two networks, one for audio and one for visual information. The audio network is a U-Net type network and the visual network is based on the ResNet model of networks. The addition of visual information to the final separation is done by weighing the audio instrument masks with the probabilities of predicted instruments. The network uses common hyperparameters that are used in different works in the field. The loss function is slightly different than any of the previous works as it uses probabilities instead of hard labels for ground truth values.

The results from the proposed system indicate that the system can separate simple audio sources from an audio mixture. The solution didn't achieve wanted results in all test cases and specially for duet audios the method had problems separating the instruments. The system showed some promise in the multimodality approach, but the better results for the multimodality approach could be accounted for by the different initial randomization of the network. The proposed solution achieved results that are better than the baseline and the most simple approaches to the same problem but failed to meet the standard separating quality that can be expected from a modern deep learning approach.

Keywords: machine learning, deep learning, sound source separation, multimodality, sound source semantic segmentation

TIIVISTELMÄ

Petteri Nuotiomaa: Visuaalisen tiedon avulla avitettu eri äänien semanttinen erottelu
Kandidaatin tutkielma
Tampereen yliopisto
Tietotekniikka
March 2024

Eri äänilähteiden automaattinen erottelu yhdestä ääniraidasta on ongelma, jonka ratkaisemiseen on käytetty monia eri menetelmiä. Monet uusimmista menetelmistä käyttävät koneoppia, varsinkin syväoppimista, saavuttaakseen äänilähteiden laadukkaan erottelun. Työn tavoitteena on kehittää menetelmä äänilähteiden semanttiseen erotteluun yhdestä ääniraidasta ja evaluoida, miten menetelmän saavuttamat tulokset vertautuvat muihin äänenerottelumenetelmiin.

Työssä käydään läpi relevantti teoria äänen erottelusta ja esitellään mitä lähestymistapoja on aikaisemmin käytetty ongelman ratkaisemiseksi. Työn taustatutkimuksen aikana huomattiin, että tutkimuskohteena äänilähteiden semanttinen erottelu on harvinainen. Tämän takia teoriaosiossa käydään läpi myös kirjallisuutta, joka ei ratkaise äänilähteiden semanttista erottelua.

Työ esittää äänilähteiden semanttiseen erotteluun uuden menetelmän, joka pohjautuu vanhoihin äänilähteiden erottelu ratkaisuihin. Työn ehdotettu syväoppimisverkko on yhdistelmä kahdesta syväoppimisverkosta. Toinen verkoista käsittelee äänidataa U-Net-syväoppimisverkossa ja toinen taas käsittelee visuaalista informaatiota ResNet-syväoppimisverkon avulla. Visuaalisen informaation ja äänidatan yhdistäminen on tehty lisäämällä ResNet-syväoppimisverkon antamat soitintodennäköisyydet painoarvoiksi U-Net-syväoppimisverkon tuottamien yksittäisten soittimien maskeille. Työssä ehdotetun syväoppisverkon hyperparametrit ovat pääosin samoja kuin aikaisemmissa äänilähteiden erotteluun käytetyissä verkoissa. Tappiofunktio kuitenkin eroaa aikaisemmista menetelmistä käyttämällä todennäköisyyksiä totuusarvona kategorioiden sijaan.

Työssä esitetyn menetelmän tulokset osoittavat, että menetelmä pystyy erottelemaan musikaalisia äänilähteitä toisistaan yksinkertaisissa tilanteissa. Menetelmä ei kuitenkaan toimi kaikissa tilanteissa tavoitellulla tavalla. Erityisesti duettoääniraitojen erottelussa menetelmä harvoin onnistuu saavuttamaan laadukkaan erottelun. Näyttöä sille, että visuaalinen informaatio olisi auttanut äänilähteiden erottelussa, saatiin vain rajallinen määrä. Menetelmää käyttämällä saavutettiin kuitenkin tulokset, jotka ohittavat erottelukyvyssään yksinkertaisimmat äänenerotteluratkaisut. Modernit syväverkkoratkaisut kuitenkin saavuttavat parempia tuloksia kuin työssä kehitetty verkko.

Avainsanat: koneoppi, syväoppi, äänilähteiden erottelu, multimodaalisuus, äänilähteiden semanttinen segmentointi

PREFACE

This thesis was conducted as part of my bachelor's studies. During its development, machine learning tools, namely ChatGPT 3.5, were used to format sentences to improve the reading experience. No external tools were used in creating the abstracts.

At Tampere, 7th March 2024

Petteri Nuotiomaa

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

AMnet	Appearance and Motion network (Proposed by Zhu & Rahtu [1])
ASA	Auditory Scene Analysis
CASA	Computational Auditory Scene Analysis
CNN	Convolutional Neural Network
DNN	Deep Neural Network
IoU	Intersection over Union
ISTFT	Inverse Short Time Fourier Transform
ML	Machine Learning
NMF	non-Negative Matrix Factorization
NN	Neural Network
ReLu	Rectified Linear Unit
SAR	Signal to Artifact Ratio
SDR	Signal to Distortion Ratio
SIR	Signal to Interference Ratio
SoP	Sound of Pixels network (Proposed by Zhao at al. [2])
STFT	Short Time Fourier Transform
T-F	Time-Frequency

1. INTRODUCTION

Separating sound sources is a task that has its uses in a multitude of different fields. One of the best-known sound source separation problems is the cocktail party problem, where the objective is to separate different speakers in an audio recorded at a cocktail party. For this study, the goal is not to separate speakers, instead the goal is to separate musical instruments from a video with multiple different musical instruments present. The neural network (NN) structure proposed in this study is designed to achieve this objective.

Over the years there have been multiple different types of solutions for this problem ranging from strictly algorithmic approaches [3] to deep neural network (DNN) approaches [2], that rely on training the network with real data. Lately, there has been a lot of development in approaches that use not only the audio but also the visual cues given in a video. These approaches have been used for speech separation [4], music separation [2], and environmental sound source localization [1].

In this study, we propose a new approach for sound source separation. The proposed method generates a semantic mask, which can be used with the original audio to isolate distinct audio sources. For the training, evaluation, and testing of the network, the solo performance videos from the MUSIC-21 [5] dataset are used.

The thesis is structured as follows: Chapter 2 presents the background information about the sound source separation task and a brief introduction to some popular methods. Chapter 3 defines the network structure studied in this paper and chapter 4 presents the experiment and evaluation result of the defined network. The final chapter summarizes the results and draws conclusions.

2. BACKGROUND AND RELATED WORKS

Machine learning (ML) is a rapidly growing field of research that affects many different branches of technology. While the breadth of ML subfields is extensive, this chapter only focuses on background information and relevant papers closely tied to the specific problem addressed in this thesis.

2.1 Background

Humans possess an ability to analyze and interpret the sounds around us, even when there are multiple different audio sources present [6]. This capacity to separate the intended sound in a mixture of auditory inputs is a fundamental aspect of Auditory Scene Analysis (ASA), a field studying human audio perception, first conceptualized by Bergman in his influential work [7]. Subsequently, the approach of Computational Auditory Scene Analysis (CASA) emerged, aiming to replicate human-like perceptual abilities through computation. As said by D. Wang et al. in the book [8], CASA can be defined as "the field of computational study that aims to achieve human performance in ASA by using one or two microphone recordings of the acoustic scene". This thesis focuses not on developing a comprehensive CASA system, but rather on developing a semi-supervised sound source separation system that solves the separation problem by training on artificial sound mixtures.

The goal of sound source separation is to obtain clear separated audio, which can then be utilized in applications such as robust automatic speech speaker recognition, hearing prostheses, and auditory scene reconstruction [8]. As mentioned in the introduction chapter, the need to separate certain instruments from musical sound sources is a good example of a sound source separation problem. For musical instruments, one possible application of sound source separation is that when an instrument is separated from a sound mixture, you can then use or analyze that instrument's audio independently.

Many of the sound source separation methods use the principle that periodic signals, such as music, can be represented as a combination of pure sine wave frequencies [1][2][5]. A process that is similar to what the human auditory system does in the cochlea [8]. The method on how to deconstruct a signal to its sine wave components is called a Fourier transform. This process moves the signal to the frequency domain. To obtain

a time-frequency (T-F) representation of the signal, we can apply the Fourier transform to consecutive frames. In the T-F domain, we can then filter the signal, so that only the target signal remains. After the filtering, we can use the inverse Fourier transform to get back to the time domain with not wanted frequencies removed from the signal. Figure 2.1 provides an example of instrument spectrograms, which are a type of T-F representation.

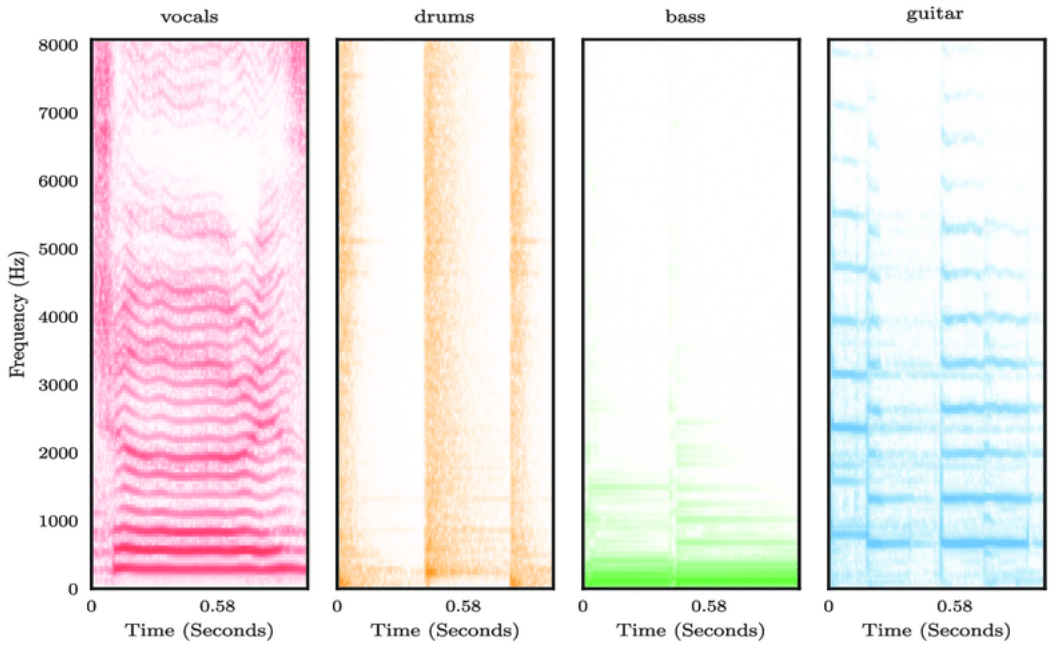

Figure 2.1. *Four examples of a spectrograms for different instruments. [9]*

In musical signals, each instrument produces a unique combination of different frequencies and these different characteristic frequencies can be used to separate different instruments from each other. These characteristics can be seen in Figure 2.1, as each instrument has a different structure in the spectrogram. What makes musical sound source separation even more feasible, is the sparsity of musical signals in the T-F domain. At most T-F points, the sources have very little energy [9], as seen in the figure 2.1. Another contributing factor to musical sound source separation is the presence of repeating structures in musical signals [9]. For instance, a drummer may adhere to a 4/4 time beat, where the bass drum is struck on every beat. These repeating structures are evident in Figure 2.1, particularly for the drums, where stronger energy concentrations occur at regular intervals.".

In sound mixtures, sound sources have different temporal properties, including variations in onset and offset times, which helps in separating different sources from each other [10]. However, in musical contexts, these temporal properties often correlate [10], for example, when orchestral instruments commence playing simultaneously upon the conductor's cue.

There are many challenges involved in separating musical sound sources which makes it

a non-trivial problem. As mentioned earlier one problem is the correlation of sources, which makes separation harder for methods that depend on independent component analysis [10]. Additionally, B. Pardo et al.[10] mention that unrealistic mixing scenarios are common in modern music. Instruments may be recorded as single tracks and then combined with different equalization, panning, and reverberation to form the final track. Another notable challenge, as highlighted by D. Wang et al. in the book [8], arises from the polyphonic nature of musical signals, compared to monophonic signals usually present in cases like noise removal or some speech processing applications. The last challenge, as noted by Bryan Pardo et al.[10], is the evaluation of musical signals. Unlike some applications where evaluation metrics are standardized, such as speech intelligibility, assessing music often requires more subjective and artistic considerations.

2.2 Related works

The network proposed in this study uses both visual and sound cues to form a semantic mask for sound source separation. While studies addressing this specific problem are limited, we can represent the problem as a collection of subproblems and then examine the relevant literature associated with each subproblem.

2.2.1 Sound source separation

Historically, sound source separation required a human to manually manipulate sound mixture frequencies to isolate a specific source. This process can be laborious and time-consuming. To automate it, one of the pioneering computational methods employed was non-negative matrix factorization (NMF), developed in the 1990s by D. Lee and Seung [11]. Although NMF was not originally devised for sound source separation, its subsequent utilization by Virtanen [12] demonstrated its effectiveness in isolating sound sources within audio signals. The results obtained through this approach serve as a baseline for evaluating the quality of sound source separation in the later chapters.

Most notable modern methods use machine learning, especially deep learning, to achieve sound separation [1][2][5][13]. A common approach is to perform the Short Time Fourier Transform (STFT) on the original audio signal to represent the audio in the T-F domain [1][2][5]. After the STFT, there are two different components of the signal: the amplitude of the signal and the phase of the signal. Phase information can be excluded as input [2], and only amplitude information is considered when generating a spectrogram. This spectrogram represents the presence of each frequency over time frames and is used as an input to the network. An alternative approach involves utilizing the time domain waveform signal directly as the input to the network [14][13].

Many modern methods first encode the audio signal to a smaller feature space and then

try to decode those features back to a binary mask [1][2][5]. The mask can then be used to form the output spectrogram by multiplying the input spectrogram with the mask created by the network and combining the result with the phase information. Binary masks are predicted, consistent with the observation by D. Wang et al. in the book [8]: "...different lines of computational consideration have converged on the use of binary masks...". Predicting binary masks allows for a more simplified representation, as it only involves determining the presence or absence of specific frequencies at each T-F bin, rather than trying to directly predict full spectrogram details. Numerous modern methods employ the spectrogram approach [1][2][5], but some methods use the time domain waveform signal as an input for the system [14][13]. The differences in spectrogram approach network architectures manifest in the distinct methods by which the analysis and synthesis stages encode and decode signals. Some approaches also add additional information by analyzing the audio signal in different ways, as example, Tzinis et al.[15] used an independent audio source classifier to derive more information about the sources present in the signals and then combined the information from the encoder and the classifier before the decoder stage.

2.2.2 Visual-audio correspondence

For human beings, it's natural to associate movement or appearance with a sound source. For example, when a person sees someone playing a guitar, the person expects to hear a sound that resembles previously heard guitar sounds. In addition, the motion information of the guitar player's hands can indicate what type of sound can be heard, as the sound depends on the motion of the hand. To use these observations within ML networks, some solutions take the approach of using an image classification network and take either the prediction output or the output of one of the last layers and feed that information to the sound network [1][2]. To get the motion information, one approach is to calculate the trajectory of areas of interest or individual pixels, in multiple consecutive frames, to estimate the motion of an object i.e. hand or a mouth. Zhao et al.[2] utilized appearance information for sound source localization, while Zhu & Rahtu [1] and Zhao et al.[5] incorporated both appearance and motion information to enhance the system's sound source separation and localization capabilities.

2.2.3 Semantic segmentation

The final goal of the suggested system is to form a semantic mask that separates each of the instruments present in the signal. Semantic segmentation forms a mask that holds information about an image associated with the mask. Usually, each pixel value has a corresponding value in the mask, which represents the class information of that pixel. In figure 2.2, there is an example of a semantic mask and a picture associated with

it. Semantic segmentation is a task usually formed for pictures, but in this case, the segmentation is done to the spectrogram of the input audio. While not as common as image segmentation, this type of segmentation has been done before. Sudo et al.[16] used a pre-trained event detection convolutional neural network (CNN) in addition to a mask U-Net structure to achieve a semantic mask for the environmental sound source. Kong et al.[17] also used a CNN with weakly labeled data to achieve a segmentation mask for sound events. Both of the mentioned segmentation methods use mostly monophonic signals, with Sudo et al.[16] using a maximum of 0.5 seconds of overlap between sound sources and Kong et al.[17] not having any overlap between predicted events, but adding background noise to the mixture.

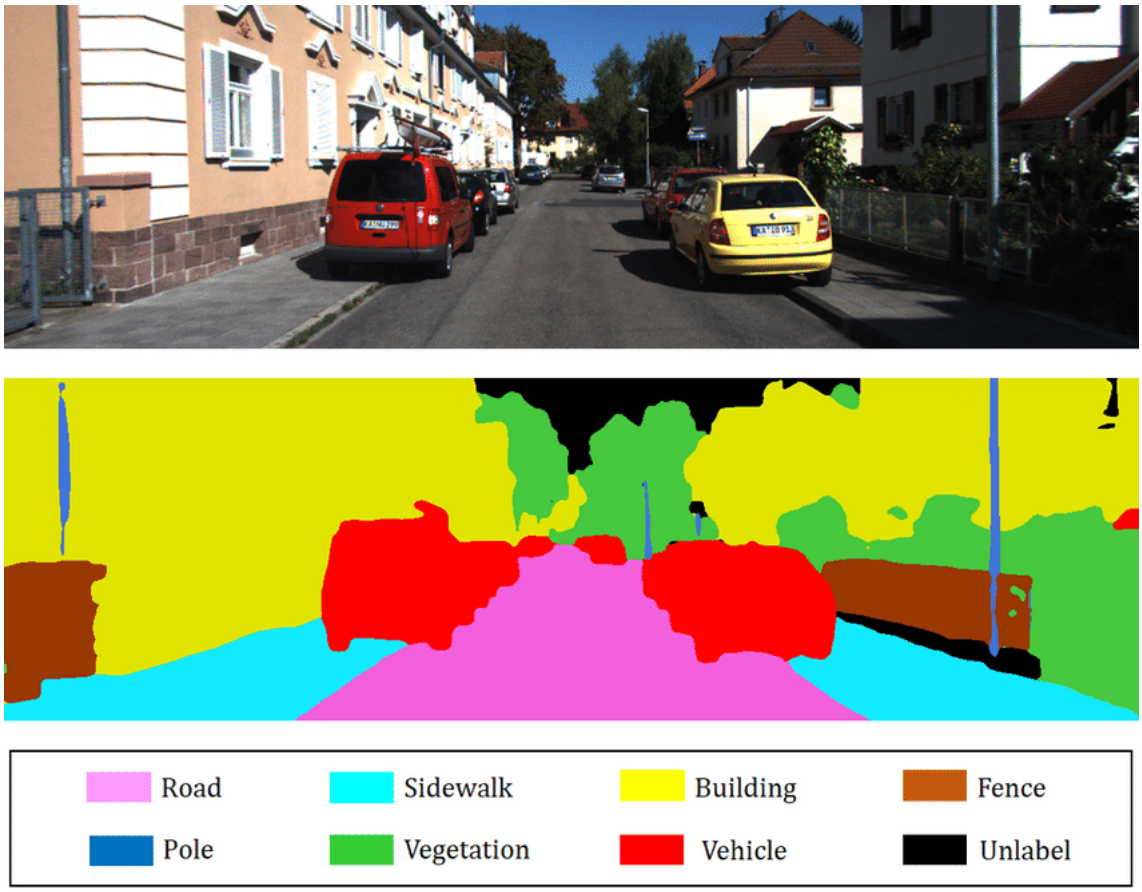

Figure 2.2. *An example of a picture and the associated semantic mask. Different colors represent different classes. [18]*

3. METHODOLOGY

The goal of the study was to find an ML method capable of separating sound sources by forming a mask that can then be multiplied with the original audio spectrogram to get each separate sound source from an audio mixture. The final architecture of the study uses a similar approach to the approach used by Zhao et al.[2], which showed promising results in separating one musical instrument from a mixture of many instruments. There are changes done to the architecture specified in the paper, as the problem at hand is slightly different from that in the paper. The most important change was made to the way the visual features were incorporated into the architecture and how many binary masks were formed by the decoder structure.

This chapter initially outlines the entire system pipeline before elaborating on the functionality of each component.

3.1 Overview

The proposed system has two different pipelines, one for training and evaluation and one for separating sound sources from duet videos. Duet videos are not used in training, because it would require a lot of processing to define ground truth values for the instrument audio tracks and that is not the goal of this study. An illustration of the full final pipeline can be found in figure 3.1.

3.1.1 Training and Evaluation Pipeline

The training pipeline first takes two different videos from different instruments and extracts a 6-second audio clip, sampled at 11025 Hz, from a random part of each video. Then it combines the two audio clips to form the input audio mixture. The audio mixture is transformed into the time-frequency domain using an STFT with a frame size of 1024 and a hop size of 256. Only the magnitude information is fed into the system. Phase information is used in the reconstruction of the audio, but not by the trained part of the network. To establish the ground truth probabilities, the STFT of each instrument is computed, followed by the application of a softmax activation along the instrument axis with instruments that are not present having a magnitude spectrogram of zeros.

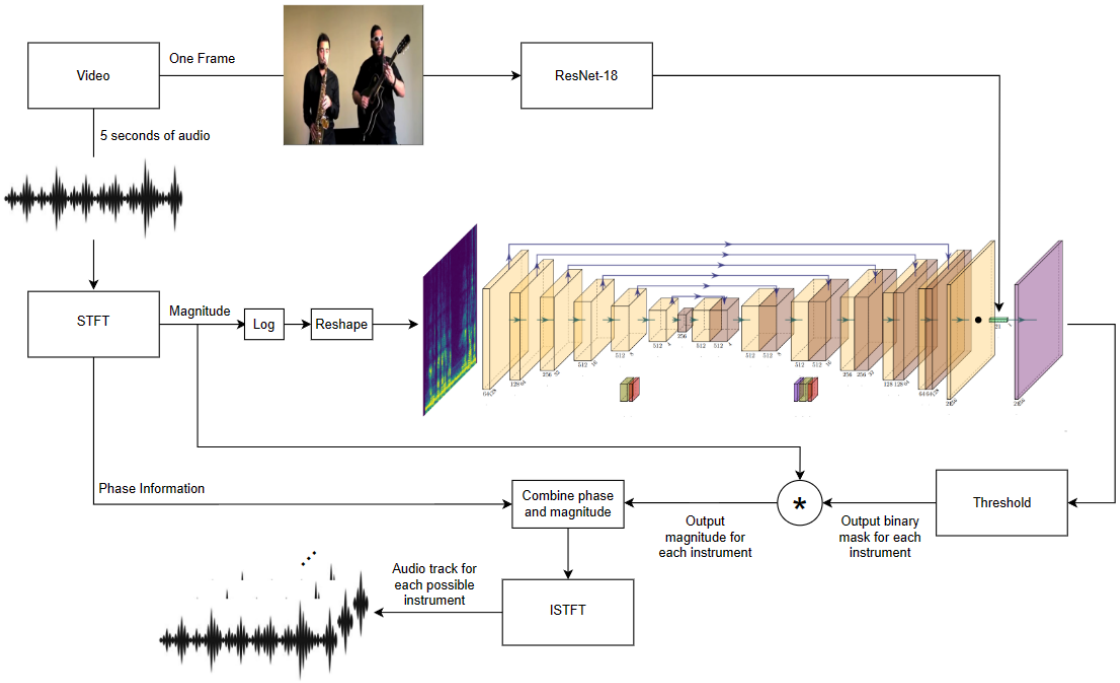

Figure 3.1. *Illustration of the full pipeline.*

The image input is a frame from the middle of the six-second audio clip, for both instruments. Both of these frames are passed through the ResNet-18 network individually and the prediction results are then combined by taking a maximum value from each frame prediction output. This is done to achieve a similar output during the training as the final product pipeline which has two instruments present in one frame.

In the feed-forward stage, the audio mixture magnitude information is first warped to match the input size of the network and then passed to the system along with the frame prediction information. The frame information is combined with the audio information after the audio network outputs the probabilities mask for each instrument. The system output is a tensor that defines a mask for all of the possible instruments in the dataset. The loss value is then calculated from the output tensor and the ground truth using the loss defined later in this chapter. Following the loss calculation, the system employs backward propagation to determine the gradient and then updates the weights using the defined optimizer.

The evaluation pipeline mirrors the training pipeline until the loss calculation stage. However, instead of updating weights, the calculated loss is used to assess the system's performance. To achieve the output audios, the most likely instrument is predicted for all points in the spectrogram using the output instrument probabilities. The result is then thresholded so that if none of the instruments are predicted with high enough confidence, a background class is predicted for that part of the spectrogram. From this result, we can then form the binary mask for each instrument so that if the instrument was the most likely,

that point in the spectrogram is 1, and if not then it a 0. This mask is then multiplied with the input mixture magnitude information and combined with the input phase information. Then by taking the Inverse Short Time Fourier Transformation (ISTFT) from the spectrogram, we can reconstruct the audio to time domain and evaluate the separation result by comparing the output audios with the original instrument audios. The testing pipeline is the same as the evaluation pipeline, but it uses testing data instead of evaluation data.

3.1.2 Final Product Pipeline

The final product pipeline is similar to training and evaluation, but the main difference is that the audio input to the system is a duet clip containing two audio sources rather than an audio mixture artificially created by combining two different videos. The Duet clip is the same length as the audio mixture used for the training and evaluation. The input for the video network is a frame taken from the middle of the duet. The desired output of the system is two audio tracks which each contain audio parts from only one instrument of the duet. All presented metrics in the later chapters are calculated using the evaluation/test pipeline because duet videos don't have ground truth values available.

3.2 Audio separating

The audio-separating network is based on the U-Net structure proposed by Ronneberger et al.[19]. The original U-Net structure is modified to be better suited for the problem at hand. U-net structure is based on the approach of first analyzing the input in the encoder stage and then synthesizing an output in the decoder stage. What sets U-Net apart from other autoencoder networks is its way of utilizing skip connections to merge encoder features with decoder features during the decoding stage. Figure 3.2 shows the full network architecture and it is split up by the dashed line to the encoder and decoder stages.

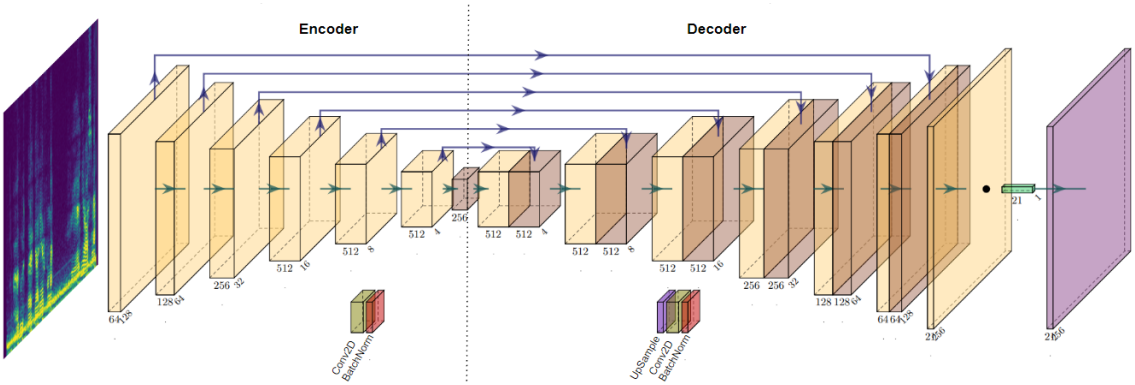

Figure 3.2. *Full audio separation network. Image predictions are represented by the green colored block. The dashed line separates the encoder and decoder stages.*

3.2.1 Encoder

In Figure 3.2 encoder architecture is represented on the left side of the architecture. The arrows leaving each feature map are the skip connections that are connected to the feature maps of the decoder part. Each one of the six green arrows represents one layer. There is also one more layer from the input to the first feature map not present in the figure. In total there are seven layers and each layer, except the first and last layer, consists of three independent components:

1. Convolutional layer
2. Batch normalization layer
3. Rectified Linear Unit activation layer (ReLu)

Convolution layers are the main building blocks for CNNs, which is a category of networks that the proposed U-Net falls under. The convolutional layer convolves multiple filters with the input using different filter kernels. Each kernel responds to different structures in the input and from the kernel-filtered outputs, the network can derive information about the input. Adding multiple consecutive convolutional layers has been proven to make the network learn more complex patterns [20]. The filtering kernels are learned through the training process using a backpropagation algorithm, which aims to minimize the output loss. Further details on the optimization method used will be discussed later in this chapter.

The used convolutional layer has a filter size of 4x4, stride of 2, and padding of 1. This means that every input for a convolutional layer is first padded with zeros so that the width and length increase with 2 (e.g., a 128x128 input becomes 130x130). After the padding, each input is convolved with several filters of size 4x4. How much the filter kernel moves after a singular convolution operation is defined with the stride parameter. The number of filters increases with the depth of the network so that the first convolutional layer has 64 filters and the maximum amount of filters is 512.

After the convolutional layer, there is a batch normalization layer, which is a popular component used to fasten the training procedure by normalizing the intermediate outputs of convolution blocks. It was first introduced by Ioffe & Szegedy in paper [21] and it is still widely used in many different DNN structures. The idea behind batch normalization is to normalize the outputs of a layer before the activation function. Normalization is not done with the mean and variances of the whole training dataset, rather it's done with the mean and variance of each mini-batch using per-dimension variances. The batch normalization layer also learns scale and shift components which are used to move and shift the normalized outputs for better results. During training a moving average and moving variance are calculated from all mini-batches and these are then used in the inference stages to normalize the outputs.

The final component of each layer is the ReLu activation, which is a common activation used in modern deep-learning architectures. It is defined as follows:

$$f(x) = max(0, x)$$

ReLu is often used in deep learning architectures because of the two advantages it has over the Sigmoid activation function. The first advantage is the smaller likelihood of vanishing gradient as the gradient of a ReLu function is either 1 or 0, while in Sigmoid functions the gradient is between 0 and 0.25. When using gradients smaller than 1 in a large neural network, the multiplication of these gradients can result in extremely small gradients. This occurs as each successive multiplication further reduces the overall gradient, which can lead to ineffective learning. The other advantage is better computational efficiency, as the function only chooses the max between zero and the already known x.

The input for the encoder is the spectrogram of the audio that needs to be separated. The input is warped to the size 256x256 needed by the network architecture. In the first layer, there is no batch normalization so the input is first convolved and then a ReLu activation is applied. In the last layer, there is no batch normalization for the convolution output.

3.2.2 Decoder

In Figure 3.2 decoder architecture is represented on the right side of the architecture. The arrows coming to each feature map represent the skip connection information coming from the encoder part of the network. Each one of the eight green arrows represents one layer. The green box is the predictions for each class calculated by the image network. In total there are eight layers and the first six layers consist of four independent components:

1. Upsample layer
2. Convolutional layer
3. Batch normalization layer
4. ReLu layer

Convolutional, batch normalization, and ReLu layers work the same way as in the encoder stages so they are not gone through again. For the convolutional layer, the parameters in the decoder stages are as follows: filter size 3x3, stride 1, and padding 1.

The upsample component is the only component that is not present in the encoder stage. The upsample layer doubles the width and height of the input feature map using bilinear upsampling (e.g., a 4x4x512 input becomes 8x8x512). Bilinear upsampling uses all neighboring values with linear interpolation to calculate new values. An example is presented in the figure 3.3 below.

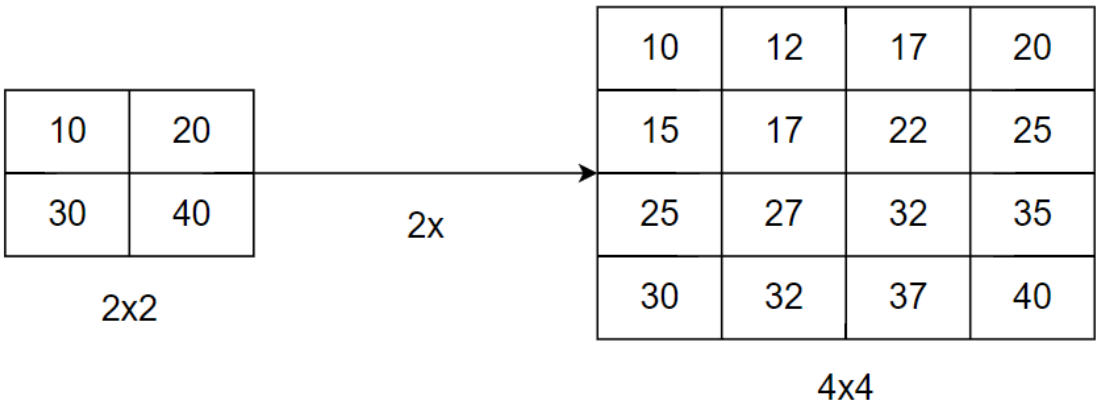

Figure 3.3. *Example of bilinear upsampling.*

The second to last layer doesn't use batch normalization and after the last convolution, a sigmoid activation is used to get the probabilities for each instrument mask. The architecture's final layer multiplies the instrument audio probabilities with the output of the visual feature extraction. This multiplication of different modality features is only done during evaluation and testing as the audio and frame networks are trained independently.

3.3 Visual feature extraction

The visual feature extraction is done by using a ResNet structure which was first proposed by He et al.[22]. The specific ResNet used in this study is a version of the ResNet-18 architecture. The ResNet-18 architecture is designed with 18 layers of operations and some of the operations have a residual connection to the output of an earlier operation. Residual connection is visualized in the figure 3.4. The ResNet-18 structure used in this study is modified so that the output layer outputs the probability for each one of the 21 possible instruments.

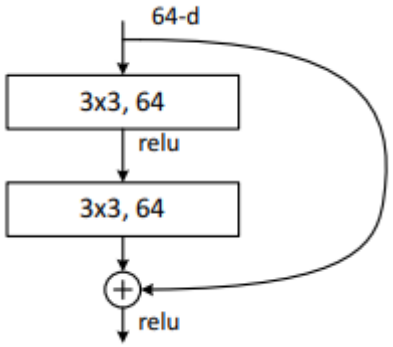

Figure 3.4. *Example of a singular ResNet block [22]. The arrows indicate input direction and the arrow skipping the two layers is the residual connection.*

3.4 Other Details

3.4.1 Optimiser

The optimization method used in this study was mini-batch gradient descent. It calculates the gradients that are used for updating the weights, for a small batch from the training set, rather than the whole training set at once. Mini-batch gradient descent was chosen because it achieved faster computation than batch gradient descent and less fluctuation of the loss than stochastic gradient descent.

3.4.2 Loss function

The final pipeline uses cross-entropy loss as the loss function. The specific cross-entropy loss used in this study uses probabilities instead of hard labels as the ground truth. The decision to use probabilities was made because the spectrogram of the mixture contained a significant presence of nearly inaudible frequencies that made the system learn non-relevant patterns when hard labels were used as the ground truth. In figure 3.5 we can see the results between both of the methods. The probabilities are computed through a softmax operation applied to the magnitude spectrograms of all instruments along the instrument axis. In instances where there is no audio for a particular instrument, its representation is a spectrogram composed entirely of zeros. The probabilistic loss function is defined in the equation 3.1.

$$l(x, y) = L = \{l_1, ..., l_N\}, l_n = -\sum_{c=1}^{21} log \frac{\exp(x_{n,c})}{\sum_{i=1}^{C} \exp(x_{n,i})} y_{n,c} \tag{3.1}$$

In equation 3.1 N is the batch size, x is the input and y is the target value given as a probability of appearing in the audio mixture. The loss is then reduced to a single number within the batch by taking the mean of the loss values.

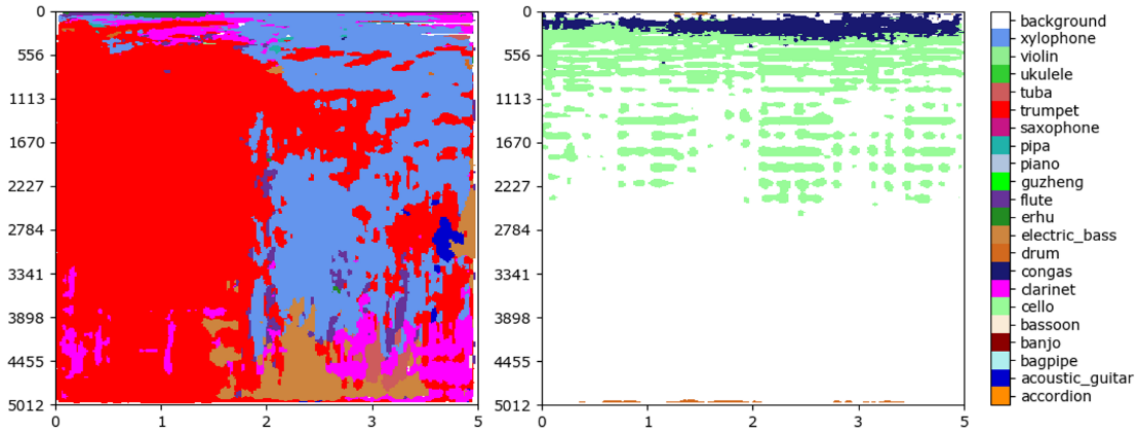

Figure 3.5. *Two spectrogram semantic masks with the left one being the prediction with a network trained with hard labels and the right one being the prediction for a network trained with probabilities.*

3.4.3 Training, Evaluation and Test Data

Training, Evaluation, and Test data consist of solo instrument performances downloaded from YouTube. The dataset is the extended version of the MUSIC dataset used by Zhao et al.[2]. The expanded dataset includes videos featuring 21 distinct instruments, ranging from a minimum of 23 videos for the saxophone to a maximum of 85 videos for the acoustic guitar. Instrument distribution is shown in figure 3.6. The dataset is divided into training, evaluation, and test datasets with the training dataset containing 80% of the videos and evaluation and test each containing 10% of the videos.

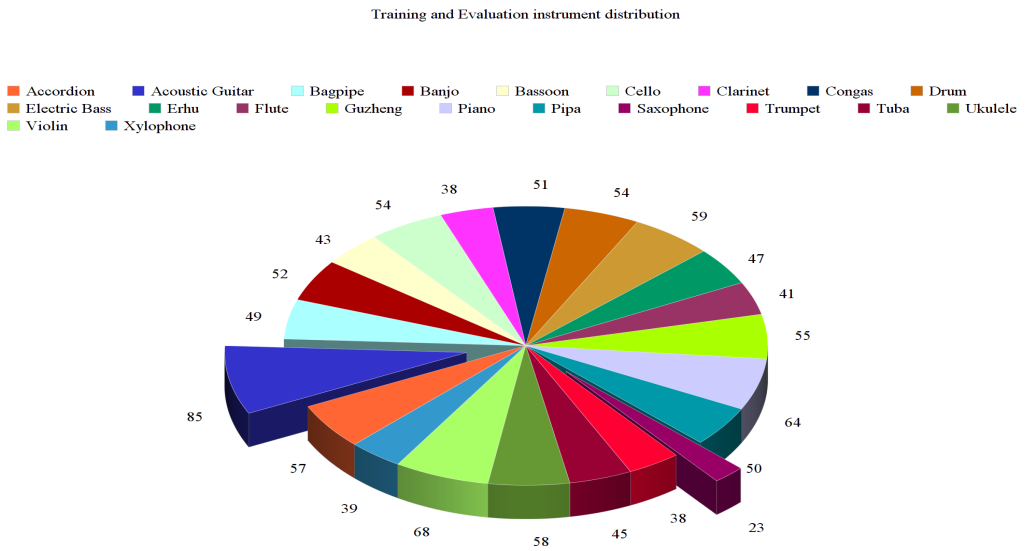

Figure 3.6. *Instrument breakdown of the whole dataset.*

4. RESULTS

To measure the quality of audio separation, 3 different aspects of the output were measured: Audio quality, classification accuracy, and semantic mask separation. Each aspect is measured using standard metrics commonly used in sound source separation literature and they are explained in their respective subsection.

4.1 Audio quality

One of the study's objectives was to obtain clean audio tracks for individual instruments. As part of this goal, one evaluated aspect of the output is the audio quality of the resulting tracks. To quantify the audio quality of separated audio tracks, three commonly used metrics are measured from the output: Signal to Distortion Ratio (SDR), Signal to Interference Ratio (SIR), and Signal to Artifact Ratio (SAR). Each of these metrics measures different aspects of the output audio. The following equation is a simplification of what components the output audio is made of:

$$\overline{s} = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}$$

To get perfect output audio we would minimize the errors caused by the different errors (e). SAR measures the amount of unwanted artifacts (e_{artif}) in the signal, SIR measures how much leakage (e_{interf}) there is in the signal from other sound sources and SDR measures how good the sound source sounds by comparing the target sound with the errors ($e_{artif}, e_{interf}, e_{noise}$). For each metric, a higher number indicates a better performance.

From table 4.1, we can see that the implementation outperforms the baseline and NMF on every measured audio quality metric, except the SAR for the baseline. The baseline is calculated by using the mixture signal as the system output, which is why there are no artifacts detected in the signal. NMF results are the ones reported by Zhao et al.[5]. On the other hand, both of the DNN approaches, Sound of Pixels (SoP) [2] and Appearance and Motion network (AMnet) [1], outperform this study's network. An important note is that SoP reports values for the MUSIC dataset containing only 11 instruments compared to the 21 instruments used in both this study and the AMnet study.

The audio-only and combined network results shown in table 4.1 indicate that the sys-

tem did achieve better results when using the multimodality approach. The difference in sound quality is minor, which means that the quality difference may be caused by different random initializations or by the random selection of 6-second clips used while training.

Table 4.1. *Sound quality metrics for the separated output audios*

Audio Quality Metrics			
Metric	SDR	SIR	SAR
Baseline	0.653	0.575	71.406
Study Audio Only	4.741	13.580	8.372
Study Combined	5.146	13.361	9.679
NMF	2.78	6.70	9.21
SoP	7.52	13.01	11.53
AMnet	11.08	18.00	13.22

4.2 Classification

To automate the separation of instruments, it is important to see if there are any false positives in the output semantic masks. To assess false positives, a confusion matrix is constructed based on the semantic mask, utilizing the top 3 predicted classes. Additionally, to evaluate the frame network, both accuracy and a confusion matrix are computed from the frame network's output.

Figure 4.1 is the confusion matrix for the combined network. The confusion matrix is created by taking the top three most predicted instruments in the output semantic mask and checking if the top three contain the ground truth labels. Typically, the most frequently predicted label corresponds to the background label. The background label is associated with confusion only when there are no erroneous predictions or ground truth labels within the top 3. Otherwise, confusions are attributed to incorrectly predicted labels.

Analyzing the combined confusion matrix reveals that, in the majority of instances the correct instruments ranked within the top 3 predicted instruments. The matrix also reveals that the system sometimes misclassified similar instruments for each other. For example, the acoustic guitar was often mistaken for the banjo, congas, and ukulele. This confusion is notable because the banjo and ukulele, like the guitar, fall under the string instrument category, sharing similar traits in both appearance and sound.

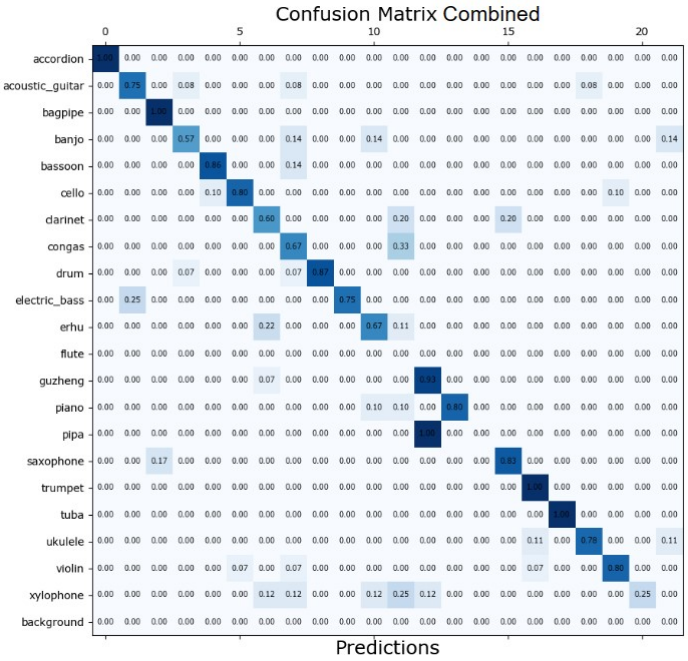

Figure 4.1. *Confusion matrix for the combined network*

Figure 4.2 is the confusion matrices calculated for the frame network and audio network separately. Examining the confusion matrices reveals comparable overall performances, yet distinct misclassifications between them. This is anticipated, as certain instruments might share visual similarities while possessing entirely different sound characteristics, and vice versa. The frame network achieved an average accuracy of 0.813.

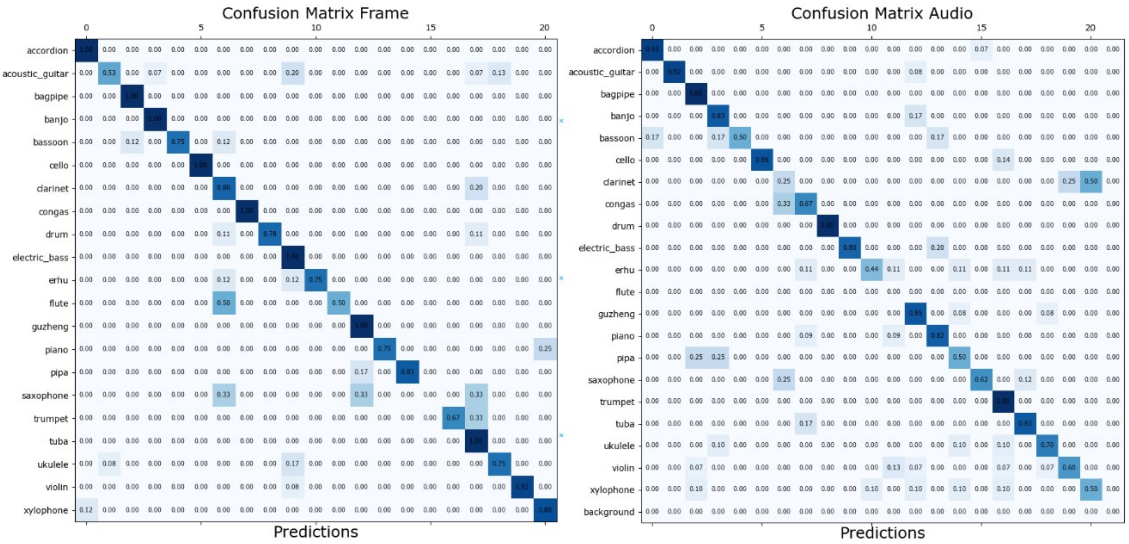

Figure 4.2. *Frame network and audio network confusion matrices. In the randomly selected test set, there were no flute samples.*

4.3 Mask metrics

One way to measure the system's ability to separate audio is to measure how well the predicted mask of an instrument overlaps with the ground truth mask of the same instrument. For this purpose two similar, but slightly different metrics were used: Intersection over Union (IoU) and Dice Coefficient, the latter also serving as the F1 score for binary mask predictions. IoU is calculated as follows:

$$IoU = \frac{TP}{(TP + FP + FN)} \tag{4.1}$$

which is similar to the dice coefficient calculation:

$$Dice = \frac{2TP}{(2TP + FP + FN)} \tag{4.2}$$

As mentioned earlier, while IoU and the Dice Coefficient are similar metrics, IoU tends to penalize over- and under-segmentation more than the Dice Coefficient.

In table 4.2 we can see the average IoU and Dice scores for different thresholds of ground truth mask. The threshold is for the probability calculated for the ground truth, which is explained in section 3.3.2. The Dice and IoU results indicate an overlap between the prediction and ground truth, but the scores fall below typical IoU standards for image semantic segmentation. In the realm of audio semantic segmentation, there is a lack of prior works measuring IoU, making comparisons with previous studies unfeasible. Overall, the masks were consistently under-predicted, particularly in capturing complex high-frequency information.

Table 4.2. *Table containing chosen mask metrics for comparing ground truth values with the output*

Mask Metrics				
Threshold	0.05	0.10	0.15	0.20
Dice	0.132	0.205	0.198	0.180
IoU	0.086	0.146	0.138	0.121

4.4 Overall results

Figure 4.3 displays visualizations of the results from the test data, and figure 4.4 displays visualizations for the duet data. The figures show the audio and frame inputs of the system and the final predicted masks. In figure 4.3 there is also the ground truth mask associated with each of the inputs.

From the 4.3 figure we can see that in many cases the system can distinguish different instruments from each other. While the audio quality in these cases may not match the quality of the ground truth audio, it still captures the characteristic sounds associated with the instruments. In specific instances, such as in the rightmost collection, the system encounters challenges in predicting the correct instruments, resulting in nearly silent or entirely silent output. Many of these cases can be accounted for by how the mixture is formed: Some of the audio clips had higher overall amplitude levels, than others, which made the system more likely to predict that instrument. Usually, in such instances, the audio track for the other instrument contains the audio corresponding to the failed prediction.

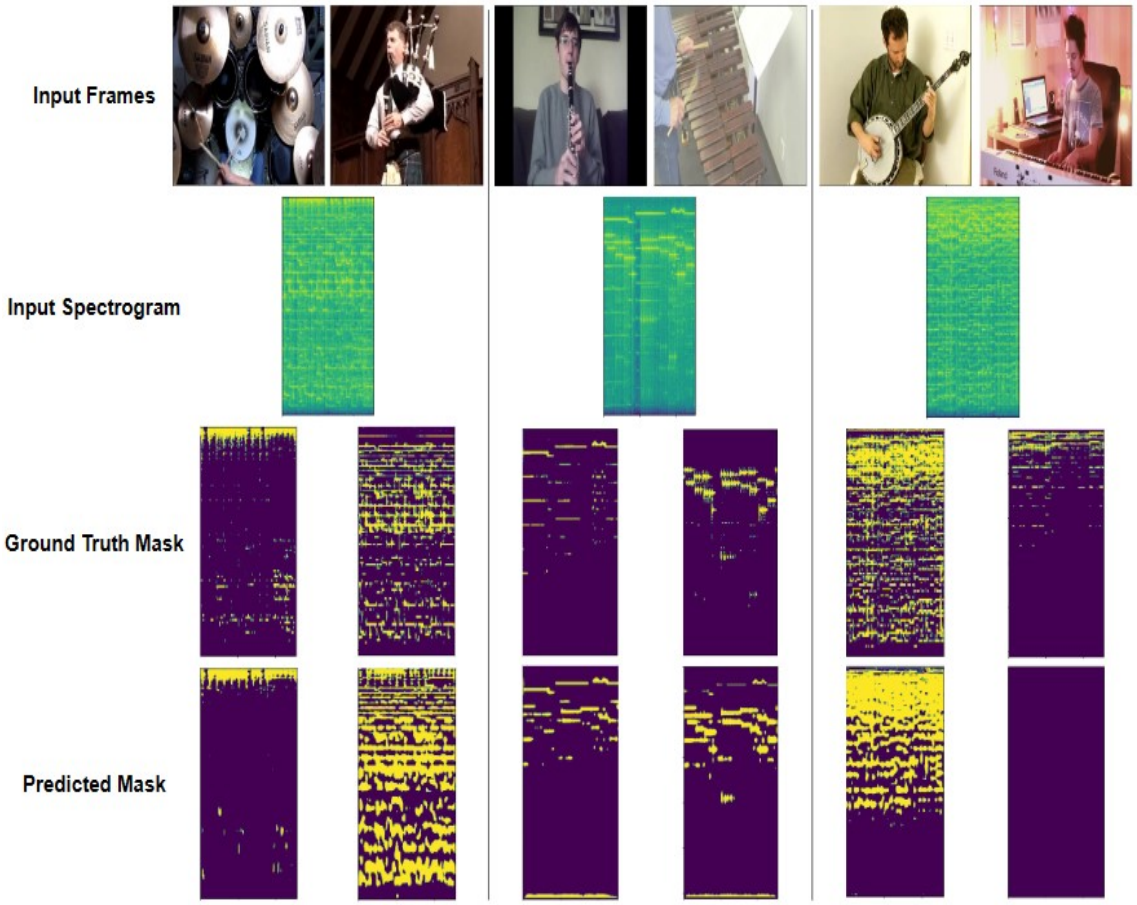

Figure 4.3. *Collection of overall results from the evaluation pipeline.*

For the final duet pipeline, where the input is duet audio, the majority of separated outputs were unsuccessful. Typically, the system predicts both instruments on one track and nothing on the other. The same behavior was present with the mixture clips. However, in the case of duet audios, where the overall amplitude level of the instruments is similar, this suggests that factors other than amplitude level differences may contribute to this behavior. Another type of failure occurs when there are no predictions for either audio track. In Figure 4.4 are some of the better results. In these results, one track may initially

exhibit both instruments after applying the threshold operation. However, after assigning individual pixels in the mask to the instrument with the highest probability, the resulting sound quality improved.

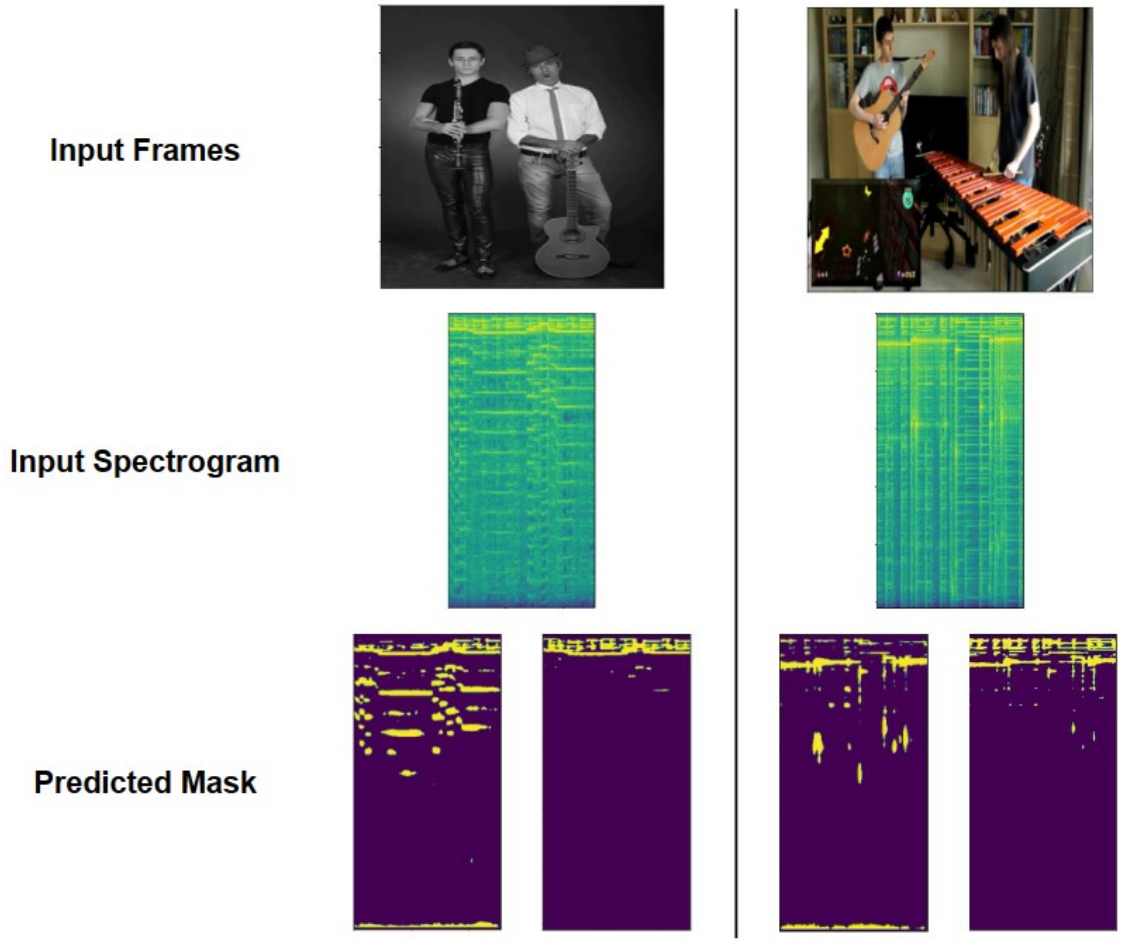

Figure 4.4. *Collection of overall results from the final duet pipeline.*

5. CONCLUSIONS

The goal of this study was to create a solution for a modified cocktail party problem where the object is to separate musical instruments. The chosen solution was to use two different DNNs and combine the results from each one using a simple weighing. The code used to produce the results is available at Github[1].

From the results shown in Chapter 4, we can say that the system was not able to achieve results good enough to classify the study as a success. Many of the artificial sound mixture cases were separated well, but there were also some failure cases. For the duet audios, there were even more significant issues, with only a handful of the duet audios being effectively separated.

The study showed some promise in the multimodality approach for sound source separation as the results of the multimodality approach had audio quality that was slightly better than the pure audio approach. There is also a possibility that the differences in the results are attributed to the distinct random initialization and the random selection of clips from the training samples.

Some potential improvements for the system involve refining the process of combining audio features with the frame features. In this study, the combination was performed after training both networks independently; a more sophisticated approach could involve merging audio and frame features and then training the entire system. Additionally, exploring alternative methods for combining, beyond simply weighing the audio output with the frame output, could be considered.

[1] https://github.com/Notsk1/SoundSeparationImplementation

REFERENCES

[1] Zhu, L. and Rahtu, E. Visually Guided Sound Source Separation and Localization using Self-Supervised Motion Representations. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 2171–2181. DOI: 10.1109/WACV51458.2022.00223.

[2] Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J. and Torralba, A. The Sound of Pixels. *The European Conference on Computer Vision (ECCV)*. Sept. 2018.

[3] Ono, N., Miyamoto, K., Le Roux, J., Kameoka, H. and Sagayama, S. Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. *2008 16th European Signal Processing Conference*. 2008, pp. 1–4.

[4] Gao, R. and Grauman, K. VisualVoice: Audio-Visual Speech Separation with Cross-Modal Consistency. *CVPR*. 2021.

[5] Zhao, H., Gan, C., Ma, W. and Torralba, A. The Sound of Motions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 1735–1744. DOI: 10.1109/ICCV.2019.00182. URL: https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00182.

[6] Miller, G. A. The masking of speech. en. *Psychol. Bull.* 44.2 (1947), pp. 105–129.

[7] Bregman, A. S. *Auditory Scene Analysis: The Perceptual Organization of Sound*. The MIT Press, May 1990. ISBN: 9780262269209. DOI: 10.7551/mitpress/1486.001.0001. URL: https://doi.org/10.7551/mitpress/1486.001.0001.

[8] Wang, D. and Brown, G. J. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[9] Cano, E., FitzGerald, D., Liutkus, A., Plumbley, M. D. and Stoter, F.-R. Musical Source Separation: An Introduction. *IEEE Signal Process. Mag.* 36.1 (Jan. 2019), pp. 31–40.

[10] Vincent, E., Virtanen, T. and Gannot, S. Audio Source Separation and Speech Enhancement. (Aug. 2018). DOI: 10.1002/9781119279860.

[11] Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* 401 (1999). URL: https://doi.org/10.1038/44565.

[12] Virtanen, T. Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria. *IEEE Transactions on*

Audio, Speech, and Language Processing 15.3 (2007), pp. 1066–1074. DOI: 10.1109/TASL.2006.885253.

[13] Stoller, D., Ewert, S. and Dixon, S. Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation. *CoRR* abs/1806.03185 (2018). arXiv: 1806.03185. URL: http://arxiv.org/abs/1806.03185.

[14] Défossez, A., Usunier, N., Bottou, L. and Bach, F. R. Music Source Separation in the Waveform Domain. *CoRR* abs/1911.13254 (2019). arXiv: 1911.13254. URL: http://arxiv.org/abs/1911.13254.

[15] Tzinis, E., Wisdom, S., Hershey, J. R., Jansen, A. and Ellis, D. P. W. Improving Universal Sound Separation Using Sound Classification. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020. DOI: 10.1109/icassp40776.2020.9053921. URL: http://dx.doi.org/10.1109/ICASSP40776.2020.9053921.

[16] Sudo, Y., Itoyama, K., Nishida, K. and Nakadai, K. Environmental sound segmentation utilizing Mask U-Net. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 5340–5345. DOI: 10.1109/IROS40897.2019.8967954.

[17] Kong, Q., Xu, Y., Sobieraj, I. and Plumbley, M. Sound Event Detection and Time-Frequency Segmentation from Weakly Labelled Data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* PP (Apr. 2018). DOI: 10.1109/TASLP.2019.2895254.

[18] Jeong, J., Yoon, T. and Park, J. Towards a Meaningful 3D Map Using a 3D Lidar and a Camera. *Sensors* 18 (Aug. 2018), p. 2571. DOI: 10.3390/s18082571.

[19] Ronneberger, O., Fischer, P. and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells and A. F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[20] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE* 86 (Dec. 1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[21] Ioffe, S. and Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 448–456.

[22] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.