Tampere University of Applied Sciences

Machine learning model for signal isolation in drum recordings

Nikke Karaksela

BACHELOR'S THESIS
November 2024

Degree Programme in Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Software Engineering

NIKKE KARAKSELA:
Machine learning model for signal isolation in drum recordings

Bachelor's thesis 21 pages
November 2024

The aim of this thesis is to train an audio denoising machine learning model that can remove bleed from snare drum recordings while still attaining all the characteristics of the sound of the drum. Microphone bleed – a common problem in music production – happens when a microphone unintentionally captures unwanted secondary sound sources. This issue especially occurs when multiple sound sources and microphones are being used like when recording a drum set.

To develop a snare drum denoising model, a comprehensive dataset of snare drum recordings was gathered and used. The model was trained using machine learning techniques. The results indicate that the proposed model succeeds in reducing microphone bleed but not eliminating it entirely. The model faces challenges in accurately denoising recordings with ghost notes and differentiating between snare drums and other percussive elements such as tom and bass drums.

These findings suggest that further improvements could be achieved by incorporating a higher sample rate, more diverse training data, and exploring different model architectures. Additionally, more computational resources could enhance the training process and the overall performance of the model. This research contributes to the field by demonstrating the potential of machine learning to solve practical audio engineering problems such as the issue of microphone bleed is.

Key words: machine learning, drum recording, microphone bleed, audio source separation, audio denoising

CONTENTS

ABBREVIATIONS AND TERMS

bleed	unwanted secondary sound source picked up by a microphone, e.g. cymbals being captured by a microphone intended for a snare drum
DAW	digital audio workstation
ghost notes	drum strokes that are played very softly
compression	audio processing method for controlling the dynamic range by decreasing the amplitude by a desired ratio when the amplitude exceeds a defined threshold, e.g. audio exceeding the 4dB threshold is attenuated by 2 dB if the ratio is 1:2
equalization	audio processing method for increasing or decreasing a specific frequency range by a desired amount of decibels
MIDI	musical instrument digital interface is a way to communicate between devices like computers and electronic musical instruments
drum trigger	a device that converts a hit of a drum into an electric signal that can be used to trigger drum samples or control a noise gate
decay	time it takes for a sound of a drum to completely disappear after it has been struck
SNR	signal-to-noise ratio
MSE	mean squared error

1 INTRODUCTION

In recent years, machine learning has become increasingly more utilized in every field, including the music industry. Music streaming services like Spotify and Apple music have trained machine learning models for their specific needs like playlist creation and song recommendations based on users' listening histories (Robinson & Schnapp 2023). Helpful tools for the creative process of making music have been developed like Moises, AIMI, MuseNet and Magneta Studio which are leveraging the power of machine learning whether it is for isolating vocal and instrumental tracks from a song or creating new compositions.

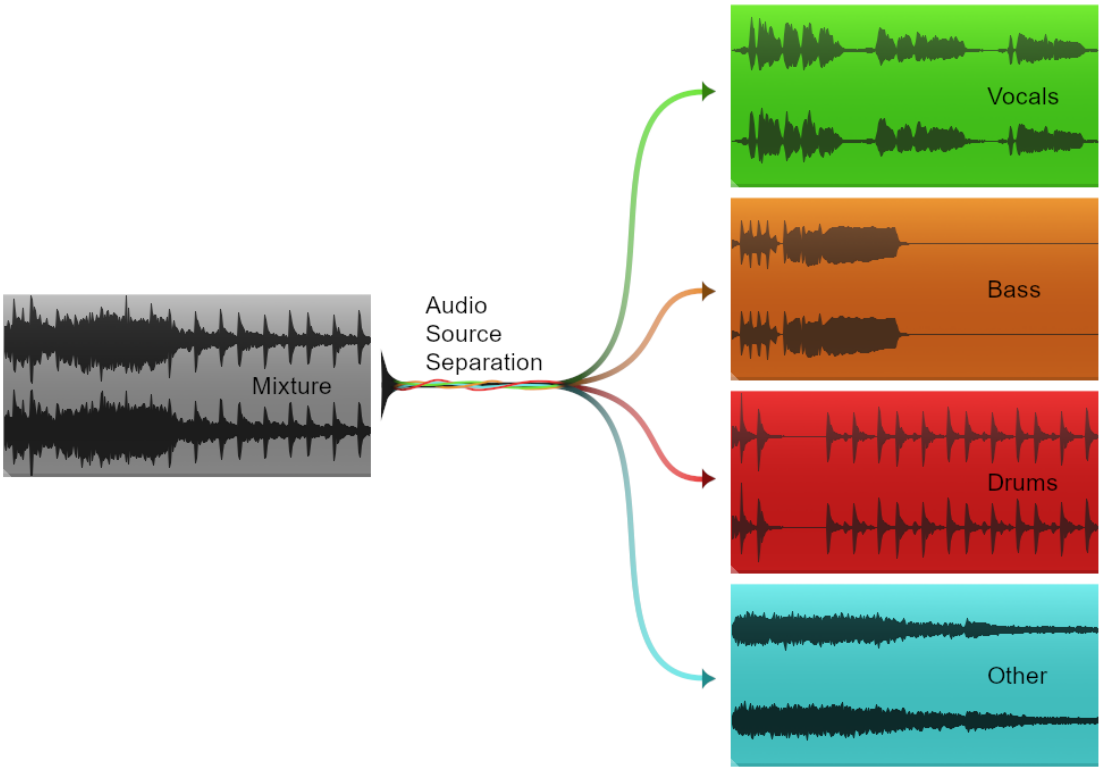
One specific challenge in music production where machine learning could have a significant impact is the issue of microphone bleed. Microphone bleed occurs when a microphone captures not only the intended sound source but also unwanted sounds from other sources, which can complicate the mixing process. In a recording studio, bleed can complicate the mixing process, making it difficult to achieve a clean and balanced mix. In live performances, bleed can interfere with sound clarity and make it challenging to produce high-quality audio for both the audience and recording purposes. By reducing or eliminating microphone bleed, engineers can gain greater control over the final sound, leading to more precise and high-quality audio production.

This thesis addresses the issue of microphone bleed by exploring how it can be mitigated using machine learning. Specifically, the focus is on developing a denoising model tailored for snare drum recordings. The aim of the thesis is to train a model that can isolate the desired snare drum signal from unwanted sound sources, thereby reducing the impact of bleed and improving the overall quality of drum recordings.

2 BACKGROUND

2.1 Audio source separation

Extracting information from music such as tempo and key or predicting proper tags and genre of a song is part of a research area called Music Information Retrieval (MIR). Many MIR problems have been solved by using both conventional machine learning techniques as well as deep learning techniques. (Choi, Fazekas, Cho & Sandler 2018.)



PICTURE 1. Illustration of audio source separation where one mixture of different audio sources is separated into their own tracks.

Among other MIR problems, there is audio source separation which is a complex task where the aim is to isolate individual audio sources from one mixture of audio (Picture 1). This kind of source separation is useful in many cases, especially in the world of music production and education, where having access for example to the vocals, bass or drums of a song is helpful whether it is when creating a

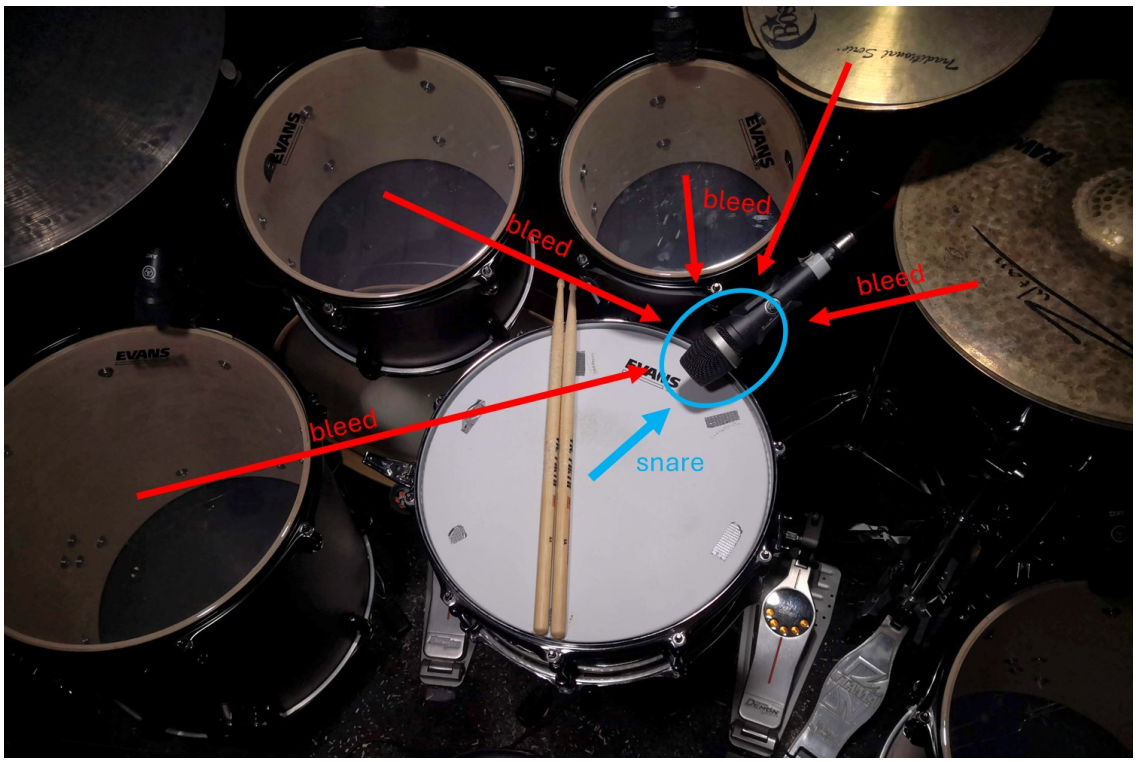
play-along tracks for students or remixing a song. (Pereira, Araújo, Korzeniowski & Vogl 2023.)

Earlier research on audio source separation for drum recordings showed that it is possible to break down a stereo mixture of a drum recording into individual stem tracks i.e. a drum recording captured by two overhead microphones separated into individual tracks of each instrument of the drum set such as bass drum, snare drum, tom drums, hi-hat, ride cymbal, and other cymbals. This source separation process enables remixing the balance of the instruments of the two-channel recording in post-production. (Cai, Su & Su 2021.)

Taking a closer look at recordings of individual instruments of a drum set, a similar source separation approach could work when it comes to mitigating bleed. For a snare drum microphone, the wanted signal is just the sound of the snare drum and everything else is considered bleed. In this case, there would be only two classes of audio sources: bleed and snare drum. Given that bleed is often unwanted, there is no need to separate it into its own track; instead, the snare drum recording could be denoised to minimize the bleed.

2.2 The challenges of bleed in drum recordings

In music production, capturing the sound of an acoustic drum set is a complex task. There are multiple approaches when it comes to microphone placement. These approaches include decisions on what type of and how many microphones to use. A common way to record drums is to have two overhead microphones capturing the drum set as a whole and close microphones on the drum shells – possibly on the cymbals too – for capturing isolated sounds of the individual pieces of the drum set. Close miking drums allow the individual instruments to be more heavily processed in post-production.



PICTURE 2. Illustration of a snare drum microphone (blue circle) capturing the snare drum signal (blue arrow) and the unwanted bleed (red arrows).

During the mixing-stage of post-production, audio processing usually means applying effects such as equalization and compression inside a Digital Audio Workstation (DAW) like Pro Tools, Cubase, and Reaper. However, if close microphones capture bleed (Picture 2), meaning secondary sound sources that are unwanted, the processing may become challenging due to the bleed being in the way of getting a desired sound. Applying compression to a snare drum track shapes the sound of the snare drum but also affects the bleed by bringing it up in volume which can result in an unbalanced mix. Mixing drum recordings with bleed can be difficult and time consuming which in many cases results in relying on using drum samples instead of the original drum recording. The ideal situation would be to have isolated tracks with minimal amount of bleed present to enable the mixing engineer the possibility to apply more processing (Oltheten 2019).

Since every drummer sets up their drums different ways, there can be various amounts of bleed present on the close microphones. Adding distance between the individual pieces of the drum set reduces the amount of bleed but is not always viable depending on the drummer and their playing style. If the drummer

strikes their cymbals hard, more bleed from them will end up audible in the microphones on the drum shells.

2.3 Ways to deal with microphone bleed in post-production

There are multiple different techniques for mitigating bleed. Often, a combination of these techniques is used for optimal results. Depending on the recorded instrument and the music genre, bleed is not always a problem that needs to be addressed. For some styles, bleed in a multitrack drum recording can serve the song by gluing together the sound of the drum set in a natural manner. (Sound On Sound 2009.)

The simplest way to get rid of bleed is to manually mute sections of audio when a specific instrument, like a snare drum, is not being played. However, this can be time-consuming depending on how long the recordings are and if it is applied over multiple tracks and takes. Yet sometimes editing manually can be the quickest way to remove bleed if there are only a few sections that need to be muted like a tom drum track.

An automated solution for removing bleed is to use a noise gate which will only let the audio pass through when the amplitude of the signal exceeds a predefined threshold. A noise gate has parameters like attack and release for controlling how fast the gate opens and closes.

However, applying a noise gate for a source like a snare drum often requires the mixing engineer to manually set the threshold for parts that are significantly quieter. Problems may occur if the recorded instrument is as quiet or even quieter than the bleed itself since the gate will start to get opened by the bleed exceeding the threshold. A possible way to solve this problem is to use a side-chain signal for triggering the gate. This side-chain signal could be a drum trigger that produces a short audio signal every time the snare drum is hit.

Earlier research on the topic of noise gates presented an algorithm that could adjust all the settings of a noise gate based on the input signal which would be

more sophisticated approach (Terrell, Reiss & Sandler 2011). Other more complex bleed removal solutions exist like Tominator by Joey Sturgis Tones and Silencer by Black Salt Audio which are being used by professional mixing engineers.

2.4 How can machine learning solve the issue of microphone bleed

Machine learning can be a powerful way to enhance audio recordings. There are multiple solutions for speech enhancing available like Krisp which is integrated in Discord and Nvidia's RTX Voice. Many of these solutions offer more than just speech denoising, like echo cancellation or audio quality enhancements, and they work in real time.

In the case of solving the issue of microphone bleed in the context of music production, a similar machine learning based approach would be a viable option. While many speech denoising models do work in real time, a model that will be used as part of music production would benefit from a model that provides better quality audio in expense of having the denoising happening real time. Denoising audio tracks before the mixing process would save up on important computational resources since the denoising would be only done once. If the audio is not denoised in real time, there would not be any added delay during playback.

Training such a model begins by gathering large enough dataset containing snare drum recordings with variety so that the model would generalize well and perform effectively on unseen data. The model would be trained to distinguish the desired audio signal, known as the ground truth, from the unwanted bleed.

When a model like this is finetuned to work well, it could be integrated into mixing engineers' workflow which would eliminate the time used to deal with bleed and allow the mixing process to be focused more on the creative decision making. Additionally, if the model would be implemented to work in real time, it could be a more reliable solution in a live music setting compared to conventional drum gating techniques.

3 IMPLEMENTATION

The implementation was conducted using audio recording equipment such as microphones, interfaces, and preamps as well as Python programming language and various libraries accompanied with Spyder integrated development environment for developing the machine learning model. The model development was done on a local computer with a dedicated GPU with 6GB of VRAM. The work had two main parts: gathering the dataset and training and evaluating the model.

3.1 Dataset

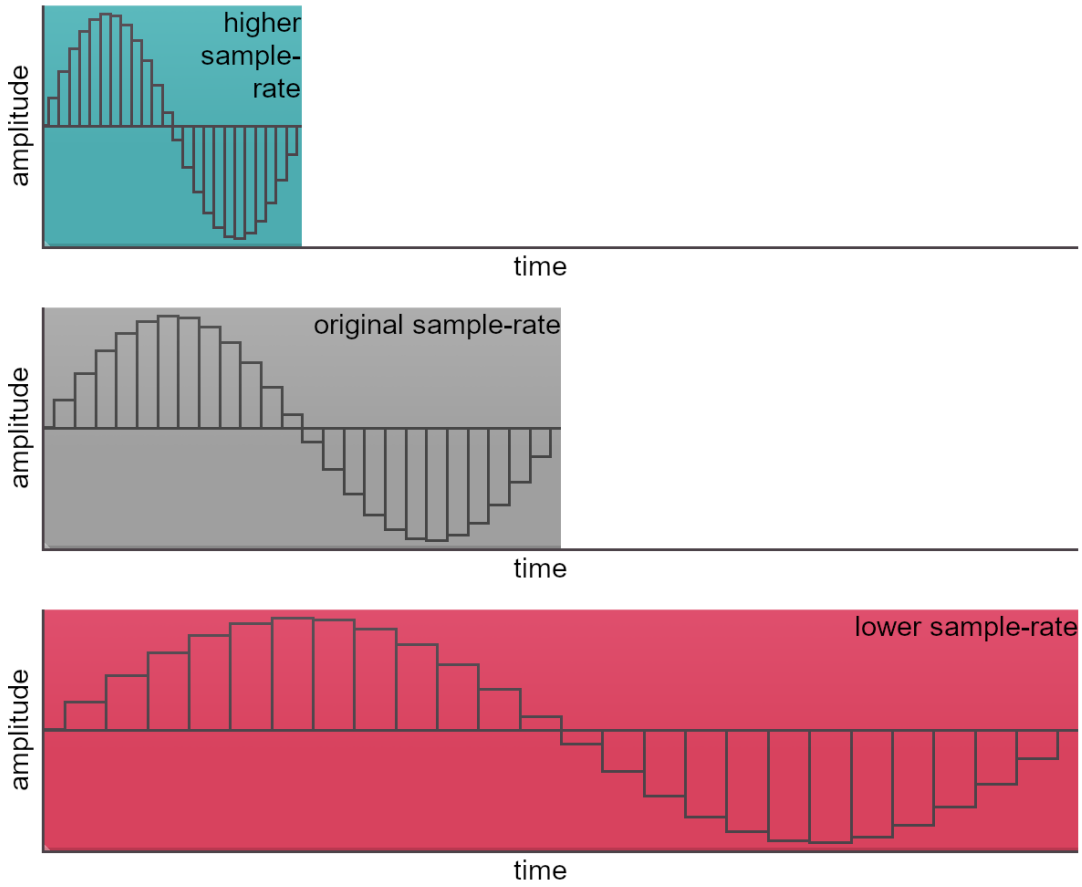
The implementation started by gathering a dataset of snare drum samples, that were already recorded over the years, and expanding the dataset by recording new samples. Most of the audio samples are around 10 min long with some exceptions that are a lot shorter or longer than 10 minutes. The total length of the dataset is 66.1 hours of audio. All the samples are single channel mono audio.

The samples were recorded by using different kinds of snare drums with different tunings and different amounts of dampening. Other variables that affect sound characteristics were also considered such as microphone and its placement, possible effects on the way in like compression and equalization, and sticks used by the drummer along with dynamics of their performance. To introduce more variety, an electronic drum kit accompanied by Superior Drummer software was utilized. This involved recording MIDI data to trigger drum samples to create realistic recordings. Other openly available drum sample libraries such as Unruly Drums and Big Rusty Drums were also utilized in the creation of the dataset.

Each snare drum sample represents the ground truth and is accompanied by its noisy counterpart. These noisy samples were created by capturing bleed from the other pieces of the drum set with the snare drum close microphone, and then combining the clean samples with the bleed. This process simulates a real-life drum recording which almost always has some amount of audible bleed.

3.2 Audio augmentation

An audio augmentation method was utilized to introduce new data and variety to the dataset. By altering both the pitch and duration of the audio recordings, it was possible to simulate snare drums tuned to different pitches and with varying decay. Stretching the audio duration resulted in a lower pitch and longer decay, mimicking a lower-tuned snare drum. Conversely, shrinking the audio duration resulted in a higher pitch and shorter decay, simulating a higher-tuned snare drum.



PICTURE 3. Illustration of sample-rate conversion where gray is the original audio recording, red is a version converted to a lower sample-rate, and blue is a version converted to a higher sample-rate.

A Python script was developed to handle the sample-rate conversion for all audio recordings. By adjusting the sample rate, both the pitch and duration of each re-

cording could be controlled. Instead of using frequency directly, the term semitone was used to describe changes in pitch. For example, raising the pitch three semitones and shortening the duration can be achieved with

$$New\ sample\ rate = Old\ sample\ rate \times 2^{\frac{semitones}{12}} \quad (1)$$

where *semitones* equals to 3 and *Old sample rate* equals to 48 kHz. *New sample rate* will be approximately 57 kHz therefore resulting in a shorter audio duration and a higher pitch, while the total number of audio samples remains the same as shown in picture 3.

The script was run with semitone value set to -3 and 3 on every audio recording. This expanded the dataset to approximately three times its size since each original recording now had two additional versions. This method was repeated with different semitone values to further increase the variety of the dataset.

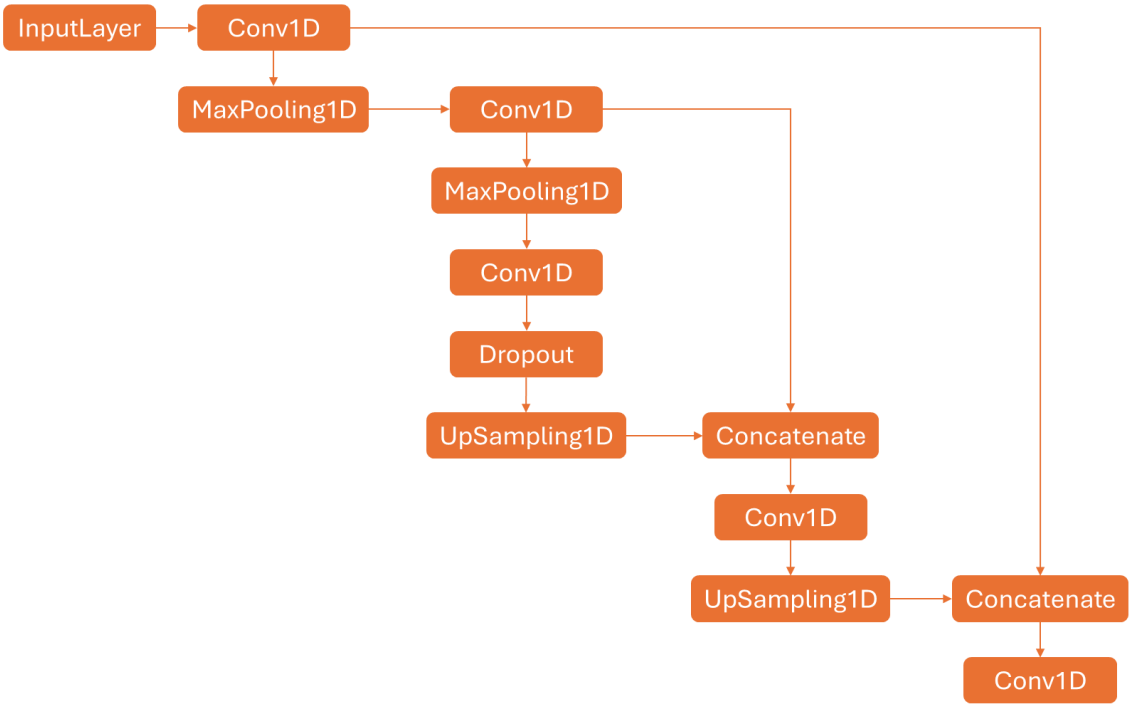
3.3 Data preparation

To be able to train the model, the audio data should be in a proper format based on what kind of model architecture is being used. In this thesis, two approaches were explored. The first approach was to train the model using Mel-scaled spectrogram images that were converted from the audio samples. However, since the approach did not consider the phase of the signal, which is an important feature in music production, it was replaced with the second approach of training the model with the audio data of the samples.

Preparing the dataset for training included splitting the samples into 10-second-long chunks and resampling them down to 20 kHz. Then the amplitude of the samples was normalized to be within the range of -1 and 1. The whole dataset was split into test and train sets with 20% of data in the test set and 80% of data in the train set.

3.4 Model training

Python was chosen for the development of the model due to its capabilities in machine learning. The model training was done utilizing an open-source library, Keras, which was used to define the neural network architecture, manage layers, and apply regularization techniques. TensorFlow served as the backend for handling computational tasks during model training.



PICTURE 4. Model architecture.

The model architecture chosen for this task was a convolutional autoencoder (Picture 4) which is a type of neural network that has an encoder and a decoder. The encoder compresses the input data into a simpler form allowing the model to learn the most important features. Then the decoder reconstructs the data back to its original form. This architecture is suitable for denoising tasks as it can isolate and preserve the key features of the snare drum sound while minimizing unwanted bleed.

A data generator function was implemented which yields batches of data pairs, that were loaded as float32 NumPy arrays, to be used as the inputs and the corresponding target outputs for training the model. These data pairs consist of noisy

and clean audio samples, sampled at a rate of 20 kHz. Each audio sample contains float32 values representing the amplitude of the audio signal at each time point. For example, a 10-second audio clip would contain 200,000 float32 values. The train generator was used to train the model while the test generator was used to validate the model's performance during training.

The layer structure of the neural network was determined through a process of experimentation and iteration. Various configurations of layers and parameters were tested to optimize the model's performance for the task of denoising audio data.

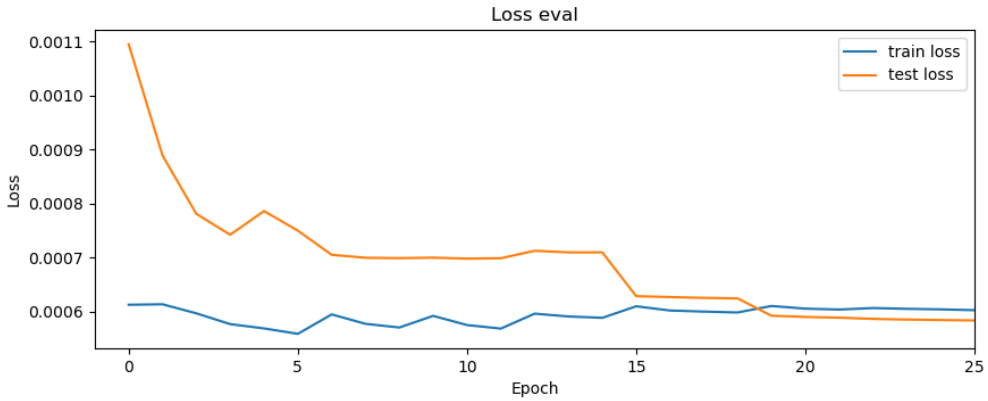
The input layer was defined to accept audio data with any length and a single channel (mono audio). The encoder part consisted of two convolutional layers followed by max-pooling layers to downsample the input data and extract relevant features. The bottleneck layer further processed the data with additional convolutional layers and a dropout layer for regularization. The decoder part included upsampling layers followed by concatenation with corresponding layers from the encoder, allowing the model to reconstruct the input data from the compressed representation. The output layer used a linear activation function to produce the final reconstructed audio signal.

The model was compiled with the Adam optimizer for its adaptive learning rate capabilities and efficiency. The loss function used was Mean Squared Error (MSE), and a custom Signal-to-Noise Ratio (SNR) metric was implemented to evaluate the model's performance during training. Additionally, several callbacks were used to enhance the training process: EarlyStopping to prevent overfitting, ModelCheckpoint to save the best model, and ReduceLROnPlateau to adjust the learning rate when the validation loss plateaued.

3.5 Evaluation

The evaluation of the model's performance was done based on three factors: visual and auditorial features and SNR values. The model was tweaked accordingly to improve the denoising results. A small validation dataset was created to

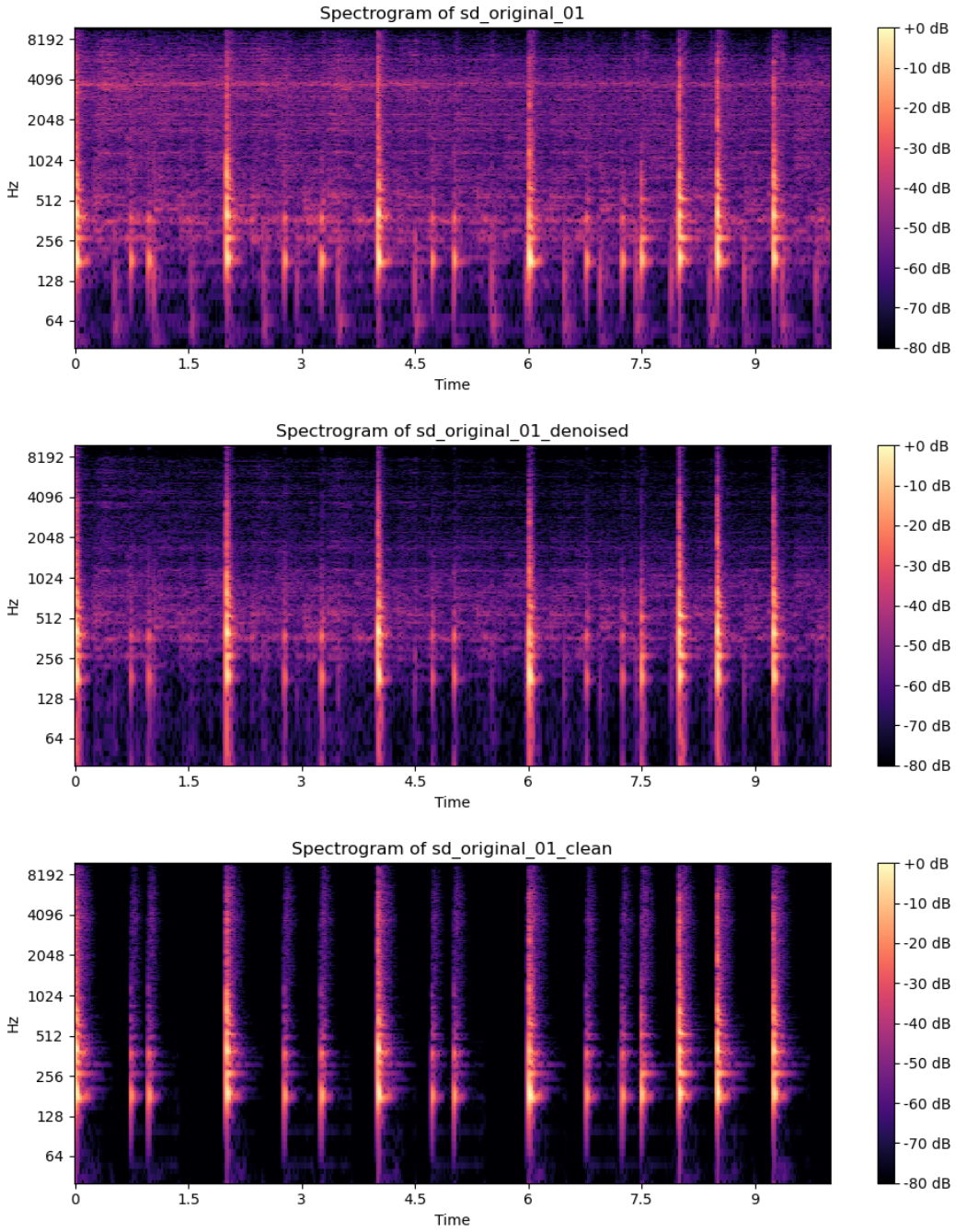
evaluate the model's performance in a real-life setting. The validation dataset did not include data from the train and test datasets.



PICTURE 5. Example of training and validation loss plots which were used to quickly determine if the latest training was generalizing well by examining if both the train (blue) and validation (yellow) losses were decreasing.

The training history was saved for further analysis and visualization. Each time after training a model, training and validation loss plots (Picture 5) were created. The plot was used to quickly determine the overall performance of the latest model training. If both the train and validation losses were decreasing after every epoch – a complete pass through of the entire training dataset – it indicated that the model was learning and generalizing well. However, if the plots were something else, it suggested that more fine tuning of the model parameters needed to be done.

Signal-to-Noise Ratio values were calculated which showed that generally, the denoised data had higher SNR compared to the original noisy data. Comparing which model had the highest SNR was considered when evaluating the performance. By having a higher SNR the model generally performed better at denoising.



PICTURE 6. Mel spectrograms of an original snare drum recording that has bleed (top), a denoised version of it (middle), and a clean version without bleed (bottom) were used to evaluate the model's performance.

Qualitative listening tests and visual analysis were conducted. The trained model was used to denoise the validation dataset. This denoised data was then compared to the original noisy and clean signals visually by creating and inspecting Mel spectrograms (Picture 6) as well as auditorily by listening to each version of the samples.

When comparing the Mel spectrograms of the original, denoised, and clean versions (Picture 6), the visual analysis focused on identifying the presence and reduction of unwanted frequencies such as high frequencies from the cymbals. The spectrograms provided a detailed visual representation of the frequency content over time, allowing for a clear comparison of how effectively the model reduced bleed while preserving the essential characteristics of the snare drum sound.

The original, denoised, and clean versions of the validation dataset were examined within the DAW, Reaper, which created helpful visual waveforms of the samples. The assessment focused on how well the model denoised elements such as cymbals and identified any artifacts, distortion, or other anomalies introduced during the denoising process.

4 DISCUSSION

4.1 Comparison between the denoising model and a noise gate

The trained machine learning model was compared against a conventional drum gate plugin hosted within the DAW, Reaper. The evaluation dataset was denoised using the model and gated using Reaper's ReaGate noise gate.

Calculating the SNR of gated and denoised versions of the evaluation dataset revealed that the gated versions have higher values than the denoised versions which indicates that overall, a gated version performs better. However, when conducting the qualitative listening tests, the gated versions let more high frequency information through, like cymbals, during each hit on the snare drum while the denoised versions did not, as seen in picture 6. The denoised versions suppressed the cymbals while letting the attack of the snare drum through, although cutting the high frequencies too aggressively in some of the cases compared to the ground truth versions.

Combining both the noise gate and the denoising model could lead to interesting results. Adding a noise gate to the tracks before denoising could potentially clear out more bleed. Also, during the training process, having a custom gating layer among other layers could have upsides which should be explored further.

4.2 Further development

The model was trained using data that was downsampled to 20 kHz to save on computational resources and cut the time that it takes to train the model. Such a low sample rate is not ideal in a music production setting. To further develop the model, the training would need to be done using data that has a sample rate of at least 44.1 kHz or 48 kHz. A higher sample rate would serve the needs of a mixing engineer better without compromising on quality of the end product such as a fully mixed song.

The input data for the model was chosen to be 10 seconds long which seemed like a good starting point. However, testing out different durations, or even variable durations, should be something to take into consideration.

The resulting model has difficulties denoising snare drum recordings that include ghost notes. During testing of the denoising capabilities, the softer strokes get lowered in amplitude as does the bleed. Including more samples with ghost notes present could lead to a better performing model in this regard.

Also, recordings that include other drums than just a snare drum proved to be difficult to denoise by the model. The model has a hard time differentiating the snare drum from tom and bass drums which leads them to be audible even after the denoising the snare drum recordings. However, if a snare drum recording does not include bleed from percussive sources like tom and bass drums, the best results with this model are achieved.

A better performance overall could be achieved by exploring different model architectures accompanied by a larger dataset that would have more variety and consider all the little nuances that are part of recording a drum set. Additionally, more computational resources would greatly benefit the training process by reducing the training times and allowing the model to be developed further.

Exploring different approaches to solve the issue of microphone bleed – whether it is cymbals bleeding into a snare drum microphone during a studio recording session or a drum set bleeding into a singer’s microphone on a live concert – is an interesting challenge that should be researched further for the sake of better music in studio productions and live performances.

REFERENCES

Cai, C. -Y., Su, Y. -H. & Su, L. 2021. Dual-channel Drum Separation for Low-cost Drum Recording Using Non-negative Matrix Factorization. Read on 25.8.2023.
https://ieeexplore.ieee.org/document/9689545

Choi, K., Fazekas, G., Cho, K. & Sandler, M. 2018. A Tutorial on Deep Learning for Music Information Retrieval. Read on 20.5.2024.
https://doi.org/10.48550/arXiv.1709.04396

Terrell, M., Reiss, J. D. & Sandler, M. 2011. Automatic Noise Gate Settings for Drum Recordings Containing Bleed from Secondary Sources. Read on 11.8.2023.
https://doi.org/10.1155/2010/465417

Pereira, I., Araújo, F., Korzeniowski, F. & Vogl, R. 2023. Moisesdb: A dataset for source separation beyond 4-stems. Read on 20.9.2023.
https://doi.org/10.48550/arXiv.2307.15913

Oltheten, W. 2019. Working With Mic Bleed. Sound On Sound 5/2019. Read on 30.04.2024.
https://www.soundonsound.com/techniques/working-mic-bleed

Robinson, A. & Schnapp, D. 2023. Rise of the Machines: How AI is Shaking Up the Music Industry. JD Supra 06.04.2023. Read on 04.09.2023.
https://www.jdsupra.com/legalnews/rise-of-the-machines-how-ai-is-shaking-5696778/

Sound On Sound. 2009. Q. What is the best way to reduce bleed on a drum recording?. Read on 29.09.2023.
https://www.soundonsound.com/sound-advice/q-what-best-way-reduce-bleed-drum-recording