Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Singing Voice Separation Using Interpretable Deep Learning

Christos Filippidis

Supervisors:

Pablo Zinemanas and Marius Miron

August 2022

Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Singing Voice Separation Using Interpretable Deep Learning

Christos Filippidis

Supervisors:

Pablo Zinemanas and Marius Miron

August 2022

Table of Contents

Dedication

I dedicate this thesis to my parents that have always supported me on every decision I took.

Acknowledgments

First of all, I would like to thank my supervisors Pablo and Marius, for their immense help and useful insights, as well as for the time they dedicated to discuss our findings and decide the appropriate steps to follow.

Pablo's previous work, which has been used as our baseline, has introduced me to new and fascinating computational concepts and methods that I was able to explore in detail through the experiments carried out.

And Marius' advice, especially on source separation related topics, has enhanced my knowledge in this field which is of great interest for me.

I would also like to thank all the people I've met in Barcelona during these two years of studying here, for the amazing time we spent together and for all the memories.

Lastly, I would like to thank my Dana, who supported me since the beginning, and moved with me to Spain.

Abstract

Audio Source Separation concerns the field of study, where the general aim is to isolate the sources from an auditory mixture. Deep learning models, which are frequently used for audio source separation, have contributed towards significant improvements in recent years. However, their black-box nature might lead to unintended effects, such as reinforcement of biases, because of the difficulty of understanding their inner workings. Thus there has recently been an increasing interest in the development of models that provide explanations for their decisions. Given that there is a lack of research in interpretability in the audio domain, in this thesis we carry out a series of experiments to leverage an existing interpretable model designed for sound classification, to singing voice separation. By using the mechanisms provided by the interpretable nature of the model as well as by analyzing the model's predictions through additional interpretable methods, we facilitate the masking process for the singing voice separation task.

The masks generated by the current intrinsic interpretable explanations of the model are not suitable for carrying out source separation tasks because of the low resolution of the computed saliency masks. However, after analyzing the model with other visual and auditory post-hoc explanations, we achieved sharper results in saliency maps, which have been used as our masks and have resulted in a significantly improved separation of the vocals. We compute the SDR and the SDR improvement (NSDR) for the reconstructed vocals estimations, in order to evaluate the separation and discuss the results. Even though the results of the evaluation are not in line with the performance of other state-of-the-art source separation algorithms, our method offers a novel approach in the field of interpretability in the source separation field and the audio domain.

Keywords: singing voice source separation; interpretable deep learning; music source separation; explainable deep learning;

1. Introduction

Deep learning has contributed towards significant advances in many research fields in recent years, but a critical concern is the difficulty to understand how or why deep learning models determine their output. This means that in some situations it is hard to know whether the model learns what is intended. Because of this, there has been a growing interest in designing models that are interpretable in some level of detail, a field also known as Interpretable Deep Learning.

Two recent interpretable models, AttProtos [1] and APNet [2], have been designed for audio classification and sound event detection respectively. They incorporate interpretable mechanisms which give explanations for their decisions. Using AttProtos as our base model, we are examining whether its intrinsic interpretable nature can be used to leverage it from audio classification to singing voice source separation. After training the model for a classification task, we explore visually and audibly the points of the input where the model focuses to calculate its predictions. The saliency maps computed through the model's attention mechanisms, are very localized and the resolution is not ideal for source separation tasks. For this reason, we explore different additional post-hoc interpretable methods, which could be used for explaining the model's decision process as well as for the separation.

The visual explanations of the post-hoc and intrinsic interpretable analyses, in the form of saliency maps, are used to calculate masks for the vocal estimations which we later apply on the input spectrograms of the mixtures, in order to extract the vocal estimation signals. The audible results of the source separation experiments for both the intrinsic and post-hoc analyses, have shown that the post-hoc results isolate more of the vocal elements than the intrinsic results, but in both cases the separation does not seem satisfactory enough. After re-constructing all the vocal estimations for all the mixtures of the test set, we evaluate the separation using Signal to Distortion Ratio (SDR) and Normalized Signal to Distortion Ratio (NSDR) as our evaluation metrics, and we compare and discuss the results.

2. State of the Art

In this section we will review the main fields of study and their subtopics. Firstly, we present the general concept of Audio Source Separation and the most up-to-date techniques, terms and approaches related to Music Source Separation (MSS). In the second part we describe the conceptual basis of Interpretable Deep Learning, and we introduce the interpretability methods which we utilise in our experiments. Finally, we explain in detail the related work of Interpretability in the audio domain, specifically the architecture of the model which provided the basis for our project.

2.1 Audio Source Separation

Over the past few decades there has been an increasing interest in the research field of audio source separation. The term is used to describe the computational process of isolating individual sounds in an auditory mixture of multiple sounds. Instead of dealing with mixtures, it is often preferred by researchers to process the isolated source signals of a mixture in order to carry out other signal processing tasks. Recovering the individual sources can facilitate the enhancement of several tasks within the Music Information Retrieval (MIR) field, such as automatic music transcription, musical instrument detection, vocal activity detection and many others [3].
For source separation tasks, the training requires large datasets where apart from the music mixtures, the isolated source targets are available in order to be used as ground truths [4]. One of the most common uses of source separation is Speech Separation where the aim is to separate two or more people talking at the same time. Another related research field is Speech Enhancement, which aims to isolate the speech from background noise. Advancement in these fields can also contribute towards music source separation.

2.1.1 Music Source Separation Algorithms

The current state-of-the-art MSS algorithms can be divided in three categories, based on the input representation used. They can be used either directly on the waveform of the mixture [5], on the magnitude or power spectrogram of the mixture [6, 7, 8] or they can be used on both waveform and spectrogram, also referred to as hybrid [9].

2.1.1.1 Input Representation

Most approaches in MSS are using spectrograms generated by the short-time Fourier Transform (STFT), instead of operating directly on the waveform. Eventually, all source separation models should be able to convert the processed audio back to a waveform. In the research carried out in [5], apart from achieving higher results than the previous state-of-the-art waveform approaches, it has also been observed that spectrogram domain models have an advantage when the content is mostly harmonic and fast changing, while for sources without pitch or with strong and emphasized attacks, waveform domain models will preserve better the structure of the source signal.

Spectrogram as Input

Time-frequency representations are the most common types of input in source separation approaches. They are calculated by the STFT, and can be converted back to a waveform by computing the inverse short time Fourier Transform (ISTFT). The parameters to consider when computing the STFT are:

• Window Type: the shape of which affects which frequencies get emphasized

• Window Size: which affects the number of samples included in each window, and also the frequency resolution.

• Hop Size: which determines the distance between two windows, usually set as half of the window size.

The phase of a signal is not explicitly represented in generated spectrograms as it is difficult to model it. In source separation the spectrograms usually used are Magnitude spectrograms, Power Spectrograms, Log Spectrograms, Mel Spectrograms etc. When re-constructing the waveform from the output mask, the phase is needed. Thus one approach is to use the phase of the original mixture. Another approach is to estimate the phase at the stage of applying the estimated mask to the mixture spectrogram. A common way to reconstruct the phase is by using the Griffin-Lin algorithm, which attempts to do the task by iterating the computation of STFT and ISTFT [10].

Waveform as Input

Demucs [5] is a waveform-to-waveform model, which had outperformed all existing state-of-the-art architectures before it in terms of Signal-to-Distortion Ratio (SDR). It operates directly on the raw input waveform and for each source it generates a

waveform as output. Instead of following masking approaches, it has been inspired by models for music synthesis.

Another recent work in the waveform category, is Wave-U-Net model [13]. It follows an approach that aims to tackle limitations that arise in the source separation using spectrogram as input, such as the inability of these models to learn to estimate signals including phase and having to estimate it or use the phase of the original mixture instead. It has achieved state-of-the-art results for waveform domain models.

Hybrid models

The hybrid version of Demucs model [9] currently performs the best out of any other model in the MSS field. Its architecture is an extended and improved version of the original architecture of Demucs. It consists of two parallel branches, one of which operates in the time domain and the other on the frequency domain. However, the processes behind the hybrid approach are over-engineered and black-boxed.

2.1.1.2 Evaluation metrics

The most reported evaluation metric in the field of MSS is the Source-to-Distortion-Ratio (SDR). It is usually shown as one number which represents the mean of SDR values calculated on a whole dataset. For this reason, it's considered an objective summary statistic that could sometimes obscure the whole distribution of calculated SDR values [3]. Other frequently reported metrics are Source-to-Artifact-Ratio (SAR), which represents the amount of unwanted artifacts in the estimated source and Source-to Interference-Ratio (SIR), which could be considered as the amount of other sources that can be heard in the estimated source. A metric suggested in [14] is Normalised SDR (NSDR) which is commonly reported when evaluating the quality of the separation of individual sources [15]. It measures the improvement of the SDR in decibels, between the input mixture and the estimated source.

Another measure is the subjective evaluation of the quality of the separation by the human ear, however it is expensive and time-consuming [3], thus it is rarely used. In some occasions it is used in order to confirm the objective evaluation results [9].

2.2 Deep Learning and Interpretability

Most common music source separation approaches use Neural Network-based methods, which are also known as Deep Learning or Deep Net methods. In the following sections we will explain the fundamentals behind Deep Learning approaches, and the methods which are used for interpreting the processes behind the decision making of Deep Neural Networks.

2.2.1 Deep Learning

Deep Learning is a subfield of machine learning that is concerned with artificial neural networks, which are algorithms inspired from the function of the biological learning of the brain [8, 16]. In recent years, deep learning has set new standards in many fields of computer science, including audio processing. There are many reasons behind this growing popularity, such as that it enables computer systems to improve with experience and data, and it also enables the computer to build complex concepts out of simpler concepts [16]. Furthermore, the existence of more powerful computers, larger datasets and techniques to train deeper networks have undoubtedly contributed towards the growth of Deep Learning. However, one drawback is the difficulty to interpret the internal representations and to explain the outputs. Another challenge is that it requires large datasets of labelled data for supervised learning.

2.2.2 Interpretable models

Even though Deep Learning has contributed towards significant advances in many scientific fields, the black-box nature of Deep Neural Networks makes it hard to interpret the process behind the decision-making. Furthermore, it may produce unintended effects, such as reinforcement of inequality and bias. The term Interpretable Deep Learning concerns a relatively recent research field which aims to create models that provide explanations of their decisions in some level of detail, which can be interpreted by users.

There is a lack of research in Interpretable Deep Learning in the audio domain. Thus, the approaches towards interpretability applied to AttProtos [1] and APNet [2] have been inspired by interpretable image classification tasks. The interpretable nature of the models is based on explanations through prototypes and concepts, previously used for the classification of images. Interpretability also relies on how the researcher explains the reasoning behind complicated classification tasks in the visual domain [2].

The methods for interpretability can be divided in categories according to many criteria [17]. In this section we will talk about two categories of interpretability methods and the relevant architectures and methods we use in our experiments.

2.2.2.1 Intrinsic Interpretability

In general, intrinsic interpretability can be achieved by restricting the complexity of the machine learning model. It can refer to machine learning models that are simplified in structure, thus they are considered easier to interpret [17]. The challenge in intrinsically interpretable machine learning is to create models that fit the data accurately while uncovering the types of patterns that the user would find interpretable [2]. Intrinsic interpretability can also be achieved by incorporating specific components in the architecture to produce explanations or to facilitate the interpretation of the network behaviour.

APNet

APNet is an intrinsic interpretable model for audio classification. The architecture of APNet allows designers to refine, debug, and improve the model, through prototype or channel redundancy, and also through manual editing [2]. APNet is designed to be an interpretable deep neural network which allows further analysis to provide insight into the inner workings of the network. The explanations produced are in the form of sound prototypes. The results have supported the claim that interpretability may help for the design of better models, and the performance contradicts the prominent assumption that there is an unavoidable trade-off between interpretability and accuracy. However, it is not suitable for classification tasks that include multi-label inputs.

AttProtos

AttProtos is an interpretable model designed for polyphonic Sound Event Detection. It has overcome the limitation of the APNet classifier which was mentioned earlier. It is able to detecting various simultaneous sound events. The model incorporates Attention Mechanisms which learn to select the part of the input that the model should focus on. It generates visual explanations of its predictions, which are faithful to its computations instead of being post hoc explanations [1]. The explanations of the network are in the form of local prototypes and attention maps and its predictions rely solely on the

attention maps.

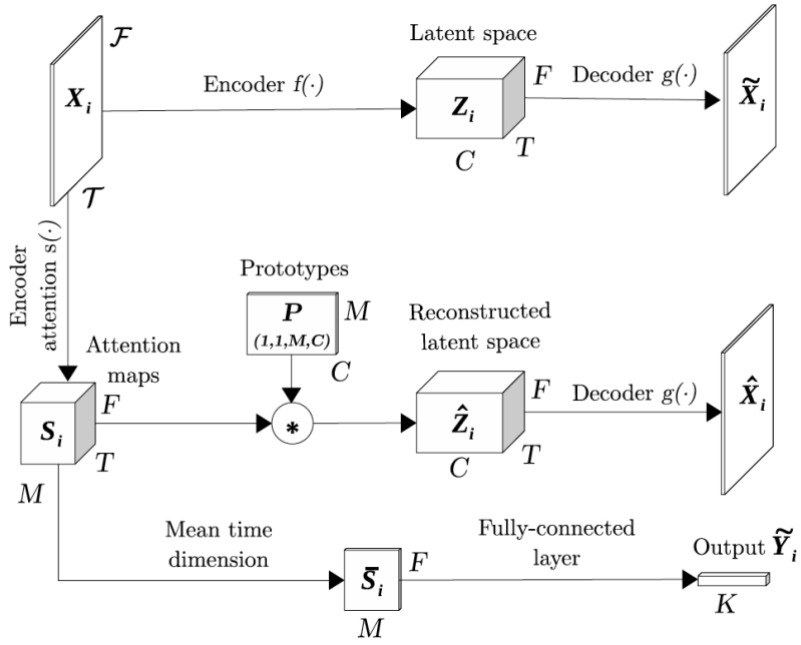

Figure 1.1: AttProtos model diagram. Figure by Zinemanas et. Al [1]

The model's architecture can be observed in figure 1.1. The top brunch of the diagram represents the auto-encoder, which is the same as the one used in APNet [11]. $Xi \in \mathbb{R}^{T \times F}$ is the input mel-spectrogram where *T* is the number of time frames and *F* is the number of frequency bins. The encoder *f(·)* extracts useful information from the input, and Zi is a tensor which represents the input in the latent space ($Zi = f(Xi)$). The shape of this tensor is (*T*, *F*, *C*) where *C* is the number of channels of the encoder's last convolutional layer. Finally, the decoder reconstructs the mel spectrogram.

Both the encoder and the decoder of the auto-encoder, are formed by three convolutional layers with leaky ReLu activations. Additionally, the encoder has two max-pooling layers between the convolutions, while the decoder has the corresponding unpooling layers.

The other two branches represent the model's classifier. $s(\cdot)$ is another encoder which extracts *M* attention maps in the latent space $Si = s(Xi)$, where *Si* is a tensor of shape (*T*, *F*, *M*). This encoder is similar to the one in the autoencoder, but with Relu activations instead of Leaky Relu, in order to return non-negative output. Each of the attention maps is related to one prototype, so the network learns a set of *M* prototypes. The model then tries to reconstruct the latent representation Zi, by multiplying each of the attention maps with the corresponding prototype, and then summing all the maps,

resulting in $\hat{Z}i$. Then, by using the decoder $g(\cdot)$ of the auto-encoder, the reconstructed tensor is projected in the input space and it can be visualised and inspected.
The bottom branch represents the detection operation which relies only on the attention maps for its predictions.
The model has three defined losses [1]. There is a binary crossentropy loss for learning the detection task, a mean squared error loss for preserving the quality in the reconstruction in the autoencoder. and a loss for the correct process of reconstruction, using the attention maps and prototypes. Additionally, there are two $l1$ regularisations, one for preventing mixing many prototypes during reconstruction, and one for keeping the explanations easy to interpret.

2.2.2.2 Post-Hoc Interpretability

Post hoc interpretability is based on interpretation methods that are applied after model training. Post hoc methods can also be applied to intrinsically interpretable models, and they can also be model-agnostic. These methods usually analyse the inner workings of models, in order to bring some insight into the reasons behind the predictions. The main challenges in post hoc interpretations of large neural networks are the high number of parameters and the stacked non-linear operations involved.

Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) is a post-hoc method for interpreting the prediction results of convolutional neural networks (CNN). It has been used for many applications such as to detect biases in models [18] and datasets [19], but also in the audio domain it has been used for speech recognition [20, 21]. The aim of the LRP technique is to find the relevance of the input image pixels to the output. To achieve that, it carries out a backward propagation from the output layer to the lower layers and at each of the layers it computes relevance values according to the selected output class and to the input image [22]. There are different rules and parameters that can define the way that relevance values are computed and distributed.
A limitation of most LRP techniques is that they are not class discriminatory, thus they return saliency maps for different classes that look almost identical, but differ only in the intensity of salience. This means that they recognize the same foreground objects of an instance for any class.

3 Methodology

In this thesis we use AttProtos as our base model, an intrinsically interpretable model which we initially train for sound classification, aiming to retrieve useful information and visual explanations of its decision making process. These explanations are provided by both Intrinsic and Post-Hoc analyses of the model, which we later transform to masks for the voice separation. Our methodology consists of four stages: a) the classification stage where AttProtos is trained on MUSDB18Mixtures and the evaluation of its classification predictions, b) the Post Hoc and Intrinsic analyses through which we compute the estimation masks, c) the reconstruction of the estimated vocals, and d) the evaluation of the separation. The classification experiments are carried out using the DCASE-models library [23]. The code for the experiments and the annotated data can be found at this link[1] with an open source license.

3.1 Dataset

For our experiments, we use is the MUSDB18 dataset [24] as it is designed to facilitate source separation tasks. It is made of 150 songs, that sum up to approximately 10 hours of total duration. It consists of a train set of 100 songs and a test set of 50 songs.

Furthermore, for each mixture there are isolated instrument stems for vocals, drums, bass and others, as well as a stem of the entire accompaniment without vocals.

A limitation of MUSDB18 is that the vocal stems have moments of silence, which could lead to false labeling during the classification task. In order to prevent this, we detect the silences the silence detector of the pydub library [25] and the time stamps are saved in files named “silences.json” and “nonsilent.json” for each song. However, to avoid the detected time stamps being slightly misaligned with the exact points of silences, we edit the times by removing 200ms from the start and end of the silent segments, with the

[1] The code and material used in this thesis can be found at this github repository: https://github.com/chrisfil/ThesisProject

exception of the first time stamp (t = 0s) and the final time stamp which marks the end of the whole song. Using the edited time stamps for silent and non-silent segments in the vocal tracks, we split the mixture audios of the whole dataset into chunks of mixtures with vocals and without vocals.

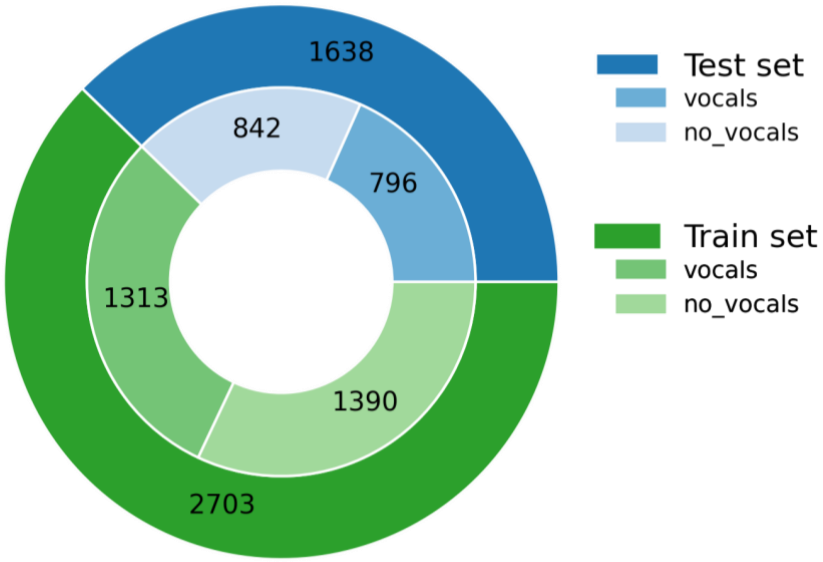

Figure 3.1: MUSDB18Mixtures dataset

This resulted in the dataset MUSDB18Mixtures as it is shown in Figure 3.1, with a train set consisting of 2703 excerpts, and a test set of 1638 excerpts, each with different durations. The label “vocals” is used for mixtures that include vocals, and “no_vocals” for mixtures that consist solely of the accompaniment.
The excerpt names and their corresponding time-stamps are also annotated on csv tables for each song of the original dataset, and are provided in this link.[2]

Additional segmentation for the isolated vocals stems, is carried out using the same time stamps we use for the mixtures. These excerpts are needed as they will be the groundtruth signals for the evaluation of the separation task.

[2] https://zenodo.org/record/7039146#.Yw_ZXexByu4

3.2 Training

Input representation

For the generation of our masks through the model's visual explanations, the model has to receive a time-frequency input representation such as STFT or Mel Spectrogram. Using the Feature Extractor of DCASE-models, we export the Mel Spectrograms in overlapping segments for each excerpt. DCASE-models utilizes librosa [26] to calculate the Short Time Fourier Transform (STFT), and then transforms it to Mel Spectrogram. Before extracting the features, it pre-processes the dataset, converting the data to mono and changing the sampling rate if necessary.

After carrying out various training and evaluation operations with different parameters to observe and improve the classification results, we set the window size to 4096 and the hop size to 1024, aiming for high frequency resolution. The overlapping windows smoothen the STFT along the time axis. We also use constant padding in order to include the whole song in the extracted features, and the sample rate is set to 44100 Hz. Each segmented frame of the Mel Spectrogram has a duration equal to a sequence time of 2.99s, and the sequence hop time has been set to 1.495s, which is half of the duration of the sequence. This would later assist for easier and more accurate calculation of the overlapped area in samples, to be used for the reconstruction of the signal.

Training parameters

The model is trained for 50 epochs at a learning rate of 0,001, using Adam as optimizer. The chosen loss weights are [10, 5, 0, 5] and the batch size is set to 96. We use softmax activation on our last layer. Because of the post-hoc analysis method requiring minor alterations to the original AttProtos model, we named the altered model as AttProtos2 and we trained it with the same parameters. More details about the alterations and the reasons behind it will be explained in the next section that concerns the analyses.

3.3 Analysis of trained model with Interpretable Methods

In this section we will present the details of our analysis methods, our observations, and how we improved the model's prediction results before proceeding to the reconstruction.

3.3.1 Intrinsic Analysis

Attention Maps

The attention maps on which AttProtos bases its predictions, can be extracted and plotted with the process defined in [1], in order to inspect them. We represent the test set of our dataset as $\{(X_i, Y_i)\}_{i=1}^{N}$, where X_i is the ith mel-spectrogram, Y_i are the one-hot encoded labels and N is the number of instances. For each X_i the model returns a prediction $\tilde{Y}_i$. For a given class k, the prediction $\tilde{Y}_i \in \mathbb{R}^{1\times k}$ is masked by a unit vector of the same shape, which has the same value in the index for the given class k. Then by calculating the gradient we retrieve the points of the previous layer that are more connected to the output k. To mask the attention maps by the most important connections to the output k, we reshape the gradient, we apply a half-wave rectifier to keep only the positive connections and we multiply it by the time averaged attention maps. Then we find the most connected prototype by maximizing the energy of the masked attention map and we extract the frequency-dependent attention function. Finally, we convert the attention map to the input space. To visualize them we multiply them with their corresponding input mel-spectrogram.

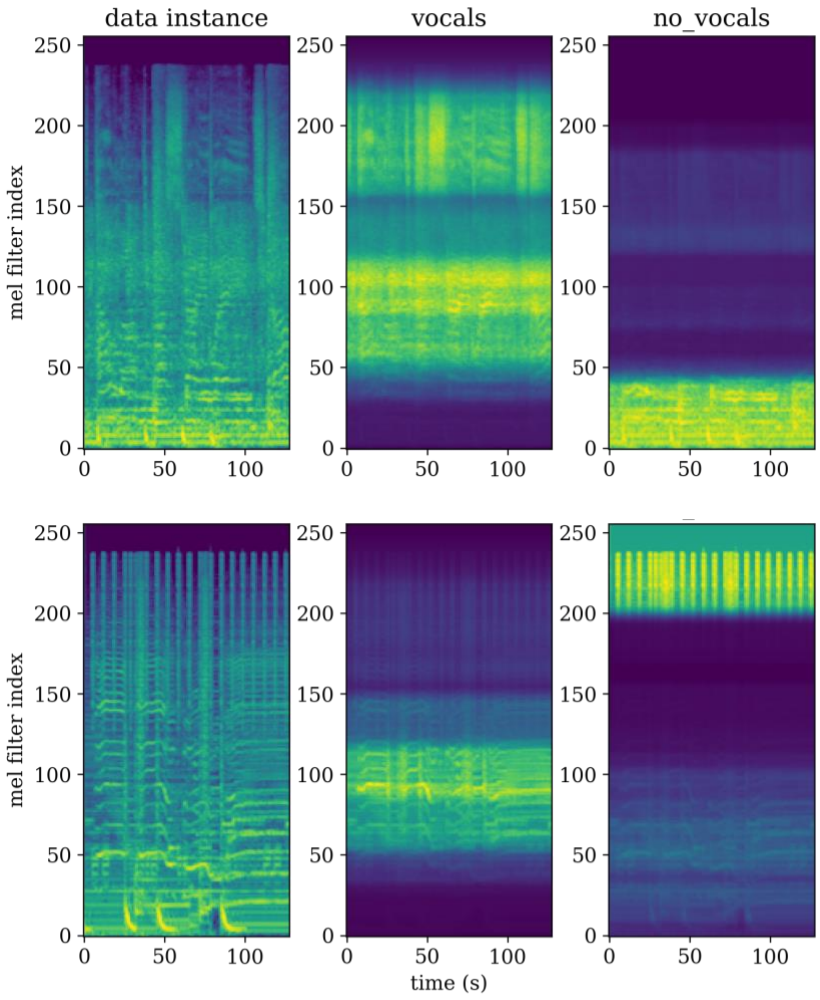

Figure 3.2: Example of two input instances masked by the attention maps. In the first column there are the mel spectrograms of the two inputs, and the other plots represent the same mel-spectrograms masked by their attention maps for the given class.

In figure 3.2 we show the plotted attention maps for two different input instances, $X_{821}$and X_{290}, both of them with "vocals" as label. In the top row the attention maps show that the model concentrates in the middle-range of the mel-scaled frequency to make its prediction for the "vocals" class and in the bottom for the "no_vocals". For the other instance it concentrates on the percussive onsets in the high range for the "no_vocals", while for the "vocals" it concentrates around the middle, and it highlights some frequencies that appear to be vocals. In both cases the maps are very localized, and they don't seem to isolate any particular elements, but it rather highlights areas with different intensities.

3.3.2 Post-Hoc Analysis

Our post-hoc analysis consists of various trial and error operations in order to choose the best possible interpretation method. We use the tools provided by the iNNvestigate library [25], which allows us to plot and compare various saliency maps using different visual interpretation methods.

In order for our model to be interpreted by iNNvestigate methods, some layers of the model have to be altered slightly or replaced. It is complicated for iNNvestigate to interpret lambda layers; thus we replace the lambda functions with equivalent reshape layers. Furthermore, layers with leakyRelu activations are also not able to be interpreted by the library, so they have been changed to Relu activations. AttProtos also includes a part of code for the model_encoder_mask, which is detected as a model on its own by iNNvestigate and it cannot be analysed. So this had to be incorporated in the main code of the model instead. The model with all the modifications is renamed AttProtos2, and it is trained using the same extracted features and hyperparameters used for the training of the original AttProtos. It is then used as the model we analyse in the post-hoc analysis.

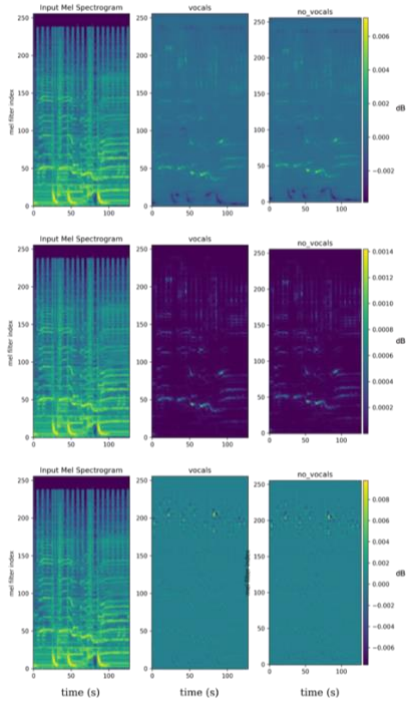

Figure 3.3: An example of Innvestigate tools used for the interpretable analysis of our model. The same input is used for analysing the model with lrp.sequential_preset_a in the first row, lrp.alpha1_beta0 in the middle, and gradient at the bottom.

We experiment with LRP methods, as well as gradient, deep_taylor and others. The saliency maps for 4 methods we used for comparison are shown in figure 3.3. The LRP methods, which have significantly better resolution than the rest, and the vocals are detected in much more detail than in the intrinsic analysis. We decide to continue our project experiments with LRPαβ where α = 1 and β = 0, as it offered the sharpest and most useful results out of all the methods provided by the tools of the library.

3.3.3 Observations and improvements

In both analyses, there are some instances where the explanations for the model predictions are providing some insight into the issues that arise after training.

We discover that the model explanations highlight elements of the mel spectrogram which did not seem to be related to vocals. This can be justified by the large variety of instruments that form the different accompaniment stems in each song, which might have misled our model on its predictions and it considers artifacts of other instrument sounds as vocals. A lot of the highlighted elements appear as percussive in nature, which means that our model might be detecting drum and other percussive elements as vocals, as the majority of the mixtures with vocals also include drums.

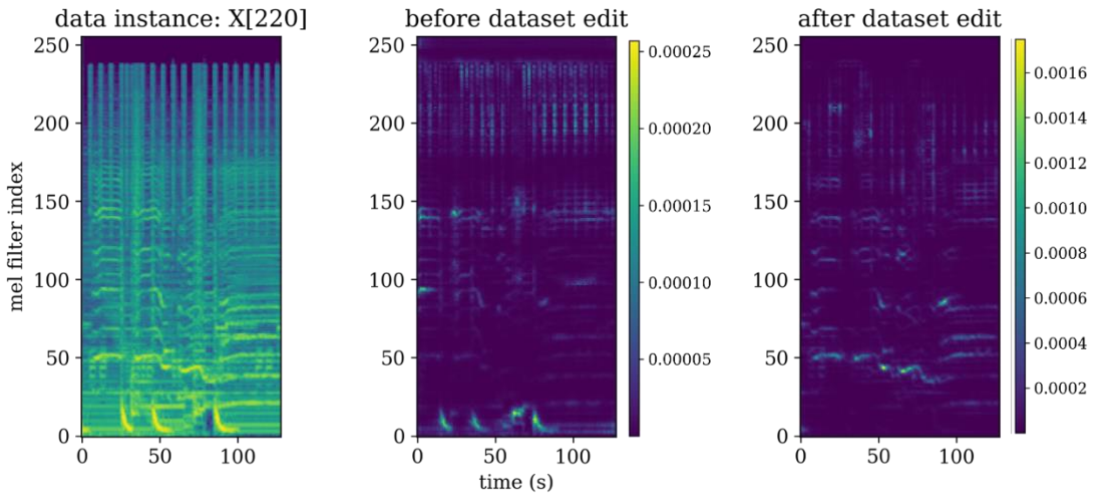

Figure 3.4: The input data instance X[220], and the LRP saliency maps before and after the data was edited.

For this reason, we repeated the training stage after altering the train set of MUSDB18Mixtures, by replacing some of the mixtures with vocals in the “vocals”

labeled data, with only isolated vocals. To do that we replaced half of the "vocals" labeled mixtures, with their corresponding isolated vocal ground truths, while keeping the same naming format. This assisted our model to define better where it should concentrate to detect the vocals.

The new analysis results as they can be seen in Figure 3.4. showed some improvement, not only in detecting the vocals but also in returning slightly sharper results.

3.4 Classification Accuracy

The final classification accuracy for both AttProtos and the altered version AttProtos2 is very high as it can be seen in Table 3. The "no_vocals" labeled excerpts have a higher prediction accuracy, while the accuracy for "vocals" labeled data decreased compared to the accuracy before editing the dataset which was at 85.3% for AttProtos and 83.5% for AttProtos2. However, this means that the model may have become slightly more discriminative when predicting vocals than before, which backs the results shown by the interpretable analyses.

Table 3: Class-wise and macro-average accuracy of AttProtos & AttProtos2 trained classifiers.

	Classification Accuracy		
Model	vocals	no_vocals	Macro-Average
AttProtos	85.1%	93.3%	88.38%
AttProtos2	81.0%	95.7%	89.20%

3.5 Reconstruction with masks

Masking is an essential part of how many modern source separation approaches approximate sources from a mixture, specifically approaches that use time-frequency representation as input.

To reconstruct the singing voice we use the maps from the analyses and we convert them to masks which are later applied on the input spectrogram before turning them back to waveform.

3.5.1 Computing masks

Intrinsic masks

Even though the attention maps from the intrinsic analysis are not capturing the vocal elements in detail, we carry out the process of masking and reconstruction for investigation and comparison purposes.

The same process we followed for plotting the attention maps in the intrinsic analysis provide the calculated saliency needed for the mask. Given that it returns contrastive saliency maps for the two labels, the mask for accompaniment could be used to reconstruct the accompaniment source, but as the main focus of this thesis is singing voice separation we only us the accompaniment map to apply soft masking when calculating the vocals mask. Soft masking is achieved with this equation:

$$mask_{voc} = \frac{Sal_{voc}}{Sal_{voc} + Sal_{acc}}$$

where Sal_{voc} is the computed saliency for the vocals and Sal_{acc} is the computed saliency for the accompaniment. To compute the mask for vocals we divide the saliency of vocals by the sum of the two saliencies.

Then to normalise the mask we apply the sigmoid function:

$$S(mask_{voc}) = \frac{1}{1 + e^{-mask_{voc}}}$$

and then we threshold the values between 0-1:

$$mask_{voc} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

The mask is calculated for every segment of the mel-spectrogram input.

Post-Hoc masks

For the post hoc analysis masks we use the saliency map we obtain from LRPaβ with a=1 and β=0. First, we discard the negative values of the saliency map, and then we apply

sigmoid function and thresholding like we did for the intrinsic masks. However, as we don't have the saliency for accompaniment so we don't apply soft masking.

3.5.2 Reconstructing the estimated vocals

We carry out 2 different operations to reconstruct the vocal estimations for all the mixtures in the test set. One by using the mask calculated with the post-hoc method and the other using the intrinsic mask. The reconstruction process is the same for both.

The phase is needed for reconstructing the estimated signal back to the waveform domain, thus for each data instance we export the original phase of the original mixture using the feature extractor of DCASE-models.

For each segment of the input mel-spectrograms X_i, we calculate the mask, and then the segment of the mel-spectrogram is converted to a power spectrogram. Then the mask is applied on the power spectrogram before we transform it back to a waveform using the Inverse Short Time Fourier Transform (ISTFT). We later use overlap add operation to connect the segmented signal and we export the reconstructed signal as a wav file.

A crucial point in reconstructing the estimated signals, is the complete alignment of its samples with the original mixture, and of course with the ground truth signal that will be used for evaluation. Thus the overlap add operation needs the accurate overlap parameters in order to return a correct reconstruction without a single sample being misaligned. To make sure that the reconstructions are correct in the time axis, we also investigate whether the reconstruction of the original signal without masking, matches exactly the original input waveform when we plot them together. If the waves are in sync, then the estimated sources are reconstructed correctly.

3.6 Evaluation

As our method is currently able to estimate only one of the two sources from our audio mixtures on a satisfactory level, we only evaluate the system's performance on the separation of the vocals sources. The evaluation metrics we use are Source-to-Distortion-Ratio (SDR) to evaluate the quality of the separated audio, and Normalized

SDR (NSDR) which represents the improvement of SDR from the original mixture to the separated signal.

SDR is calculated by the equation:

$$SDR(S_e, S_r) = 10log_{10}\left[\frac{< S_e, S_r >^2>}{\|S_e\|^2 \|S_r\|^2 - < S_e, S_r >^2}\right],$$

where S_e is the estimated signal and S_r is the reference signal.

NSDR is calculated by subtracting the SDR of the original mixture signal S_m to the reference signal S_r, from the SDR of the estimated signal S_e to the reference signal S_r.

$$NSDR(S_e, S_r, S_m) = SDR(S_e, S_r) - SDR(S_m, S_r)$$

For our evaluation we use the museval library [27], that utilises the bss_eval tools which are commonly used for evaluating source separation tasks.
The estimated sources are the vocals that have been reconstructed, while the reference sources are the isolated vocal stems which were segmented according to the time stamps used to create MUSDB18Mixtures. Thus, the reference sources for "vocals" labelled data, include the ground truth isolated vocals, but for "no_vocals" labelled data there is silence or very few audible elements. This silence can bring ambiguity into the evaluation results, and the tools we use return an error when a reference or estimate source is completely silent.
We overcome this issue by concatenating all the excerpts of the estimated sources, reference sources and mixtures, to evaluate the SDR and NSDR on the full songs of the original MUSDB18 dataset. By using the full duration of the song for the evaluation, there is no signal that includes only silence. We evaluate the performance on the 50 songs of the test set of the original dataset.
To evaluate the SDR and NSDR for the whole songs, we use a bool argument in the code. The evaluation results for full songs are saved in a csv table, together with the

name and the duration of the song. The SDR is computed on frames of 1s, and for each song we have an average SDR value.
The Global NSDR (GNSDR) and Global SDR (GSDR), are computed as the weighted averages of NSDR and SDR with the duration of each audio used as the weight.

4. Results and Discussion

4.1 Results

As it is shown in table 4.1 the GNSDR for the vocals separated with masks of the post-hoc analysis is 7.01 dB, while for the estimated sources of the intrinsic masks, the GNSDR is much lower at 2.63 dB. Similarly the GSDR for the separation with the intrinsic masks is really low at -3.55 dB and for the separation with post-hoc masks it is 0.83dB.

Table 4.1: Weighted-average GSDR and GNSDR in dB, for the models AttProtos, and AttProtos2.

Model	GSDR(dB)	GNSDR(dB)
AttProtos (Intrinsic)	-3.55	2.63
AttProtos2 (Post-Hoc)	0.83	7.01

As the performance of the separation with intrinsic methods is not satisfactory for now, we will investigate and discuss the separation results of the separation carried out through the post-hoc method.
In figure 4.1 we show the box-plot representation of the distribution of SDR and NSDR values per song. It can be observed that the distribution is relatively smooth apart from two outliers which have scored really low SDR and really high NSDR. Most SDR values lie between 0 and 2.175 dB, which means that the performance of our system is restricted in this range of separation quality. However, it is interesting to investigate some of these values and compare the efficiency of the two metrics we use.

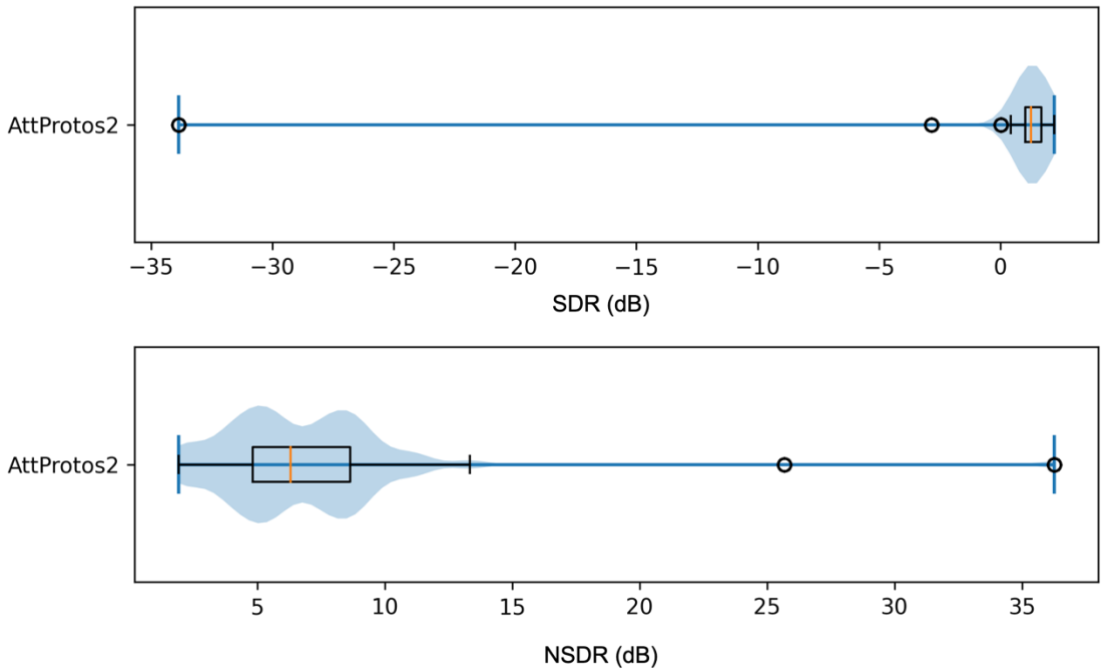

Figure 4.1: Weighted box-plots on violin-plots for SDR and NSDRevaluation results on full songs with Post-Hoc masks.

4.2 Discussion

In table 4.2 we show the four songs with the highest NSDR values. The songs “PR – Happy Daze” and “PR – Oh No” are the outliers that we can also see in the boxplots in Figure 4.1. Even though they have the highest NSDRs, their SDRs are some of the lowest in the evaluation results. NSDR might not be the best evaluation metric for songs like this. It is understood that the improvement of SDR is really high for some cases because of how the source to distortion ratio for the original mixture to the reference source, is already really low. This means that these mixtures include accompaniments with a lot of prominent elements and instruments with particular sound qualities that really affect the way the SDR is calculated.

Table 4.2: The four songs with the highest NSDR values for the estimated vocals, as well as their corresponding SDR values calculated for the original input mixtures and the estimated vocals.

Song	$SDR(S_m, S_r)$ (dB)	$SDR(S_e, S_r)$ (dB)	$\boldsymbol{NSDR}$ **(dB)**
PR - Happy Daze	-70.140	-33.869	**36.271**
PR - Oh No	-28.506	-2.833	**25.673**
Punkdisco - Oral Hygiene	-12.789	0.551	**13.340**
BKS - Bulldozer	-9.999	1.328	**11.327**

When listening to the mixtures of these outlier cases, it is proven that these songs share common elements. They are all electronic songs, with a lot of artificial sounds and limited vocals, which are usually repeated along with strong beat patterns. The evaluation results for these songs and others with similar qualities are understandable, but not as reliable. It is possible that the model finds it difficult to isolate vocals in such occasions, where the majority of the song is electronic in nature, or where there are intense percussive sounds.

On the other hand, in table 4.3 we show the four songs with the highest SDR values. When listening to these four songs, the accompaniments consist of more acoustic sounding instruments, such as guitar, drums, bass and piano. It is then proven that the model performs better on songs that are acoustic in nature. When it comes to the NSDR values, they are closer to the weighted-average GNSDR. Also comparing the SDR of the mixture to reference sources with the SDR for the estimated vocals, the results seem more coherent for these songs than for the songs in table 4.2.

Table 4.3: The four songs with the highest SDR values for the estimated vocals, and their corresponding NSDR values.

Song	$SDR(S_m, S_r)$ (dB)	$\boldsymbol{SDR(S_e, S_r)}$ **(dB)**	$NSDR$ (dB)
Moosmusic - Big Dummy Shake	-5.342	**2.175**	7.517
BKS - Too Much	-3.012	**2.188**	5.200
Detsky Sad - Walkie Talkie	-2.836	**2.227**	5.063
Angels In Amplifiers - I'm Alright	-3.418	**2.231**	5.649

Recent works in the field using spectrogram as input, such as D3net [11] and Open-Unmix [12], have reported state-of-the-art SDR on the MUSDB18 dataset. D3net average SDR results are 6.01 dB without using extra training data, and 6.63 dB with extra training data. Open-Unmix has reported 5.33 dB as average SDR without extra training data and 6.316 dB with extra training data. The difference between the performance of those models and our method's is apparent, however there is room for improvement.

4.3 Future Work

The intrinsic explanations of AttProtos are currently not able to produce sharper results with higher resolution, but this is something that could be addressed in the future. In that case, this model that is interpretable in nature won't only give useful insights into the prediction process but could play an important part in the separation process. Its inner workings can also be used to enhance the separation, or to facilitate the computation of better masks. Another improvement in the future can be the use of discriminative LRP such as in [28], or other interpretable methods in order to achieve the separation for all sources. The evaluation would return more complete and reliable results if there are estimated sources for all the sources of the input mixture. Furthermore, the input representation could be just the STFT without transforming it to Mel Spectrograms, as this affects the quality of the separation.

5. Conclusion

Our evaluation shows that the performance of the method we propose at its current state is not in line with the performance of other state-of-the-art models. However, the interpretable methods used, the analysis and the investigations have shed light on reasons why this happens, and how it can be improved. The new methods examined in this thesis could serve as the basis for further research which could assist various tasks in the source separation field.

6. Bibliography

[1] Zinemanas, P., Rocamora, M., Fonseca, E., Font, F. & Serra, X. Toward interpretable polyphonic sound event detection with attention maps based on local prototypes. In *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)* (2021).

[2] Zinemanas, P., Rocamora, M., Miron, M., Font, F. and Serra, X. An Interpretable Deep Learning Model for Automatic Sound Classification. *Electronics*, 10(7), 850 (2021).

[3] Manilow, E., Seetharman, P., Salamon. J., *Open Source Tools & Data for Music Source Separation*. (2020). at <https://source-separation.github.io/tutorial;>

[4] Schulze-Forster. K. Informed Audio Source Separation with Deep Learning in Limited Data Settings. *Signal and Image processing* (2021).

[5] Défossez, A, Usunier, N, Bottou, L, Bach, F. Music Source Separation in the Waveform Domain. *arXiv.org* (2021).

[6] Parekh, D., Kharah, D., Suthar, K., Shirsath, V. Audio Stems Separation using Deep Learning. *International Journal of Engineering Research & Technology (IJERT),* 10(03) (2021).

[7] Petermann, D., Chandna, P., Cuesta, H., Bonada, J., Gomez, E. Deep Learning Based Source Separation Applied To Choir Ensembles (2020).

[8] López, S., F.. Music Source Separation Using Deep Neural Networks. Universitat Politécnica de Catalunya (2020).

[9] Défossez, A. Hybrid Spectrogram and Waveform Source Separation. *MDX Workshop (ISMIR)* (2021).

[10] Griffin, D. &Lim, J. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2), 236–243, (1984).

[11] Takahashi, N. &Mitsufuji, Y. D3Net: Densely connected multidilated DenseNet for music source separation. *arXiv.org* (2022). at: <https://arxiv.org/abs/2010.01733>

[12] Stöter, F., Uhlich, S., Liutkus, A. & Mitsufuji, Y. Open-Unmix - A Reference Implementation for Music Source Separation. *Journal of Open Source Software,* 4(41), 1667 (2019).

[13] Stolter, D., Ewert, S. & Dixon, S. Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation. In P*roceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)* (2018).

[14] Ozerov, A., Philippe, P., Bimbot, F. & Bribonval, R. Adaptation of Bayesian Models for Single-Channel Source Separation and its Application to Voice/Music Separation in Popular Songs. *In IEEE Transaction on Audio Speech and Langauge Processing* 15(5) (2007).

[15] Geng, H., Hu, Y. & Huang, H. Monaural Singing Voice and Accompaniment Separation Based on Gated Nested U-Net Architecture. *Symmetry,* 12(6), (2020).

[16] Goodfellow, I., Bengio, Y. & Courville., A. Deep Learning. *MIT Press,* (2016).

[17] Molnar, C. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (2nd ed.), (2022). at: <christophm.github.io/interpretable-ml-book/>

[18] Arias-Duart, A., Parés, F., Garcia-Gasulla, D. & Gimenez-Abalos, V. Focus! Rating XAI Methods and Finding Biases**,** *arXiv.org* (2022)**.**

[19] Hägele, M., Seegerer, P., Lapuschkin, S. *et al.* Resolving challenges in deep learning-based analyses of histopathological images using explanation methods. *Sci Rep* **10, 642**3 (2020). at: <https://doi.org/10.1038/s41598-020-62724-2>

[20] Markert, K., Parracone, R., Kulakov, M., Sperl, P., Kao, Y. & Bottinger, K., Visualizing Automatic Speech Recognition – Means for a Better Understanding? In Proceedings ISCA Symposium on Security and Privacy in Speech Communication (2021).

[21] Bharadhwaj, H. Layer-wise relevance propagation for explainable deep learning based speech recognition. *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (2018).

[22] Huang, X., Jamonnak, S., Zhao, Y., Heng Wu, T. & Xu, W. A Visual Designer of Layer-wise Relevance Propagation Models. (2021).

[23] Zinemanas, P., Hounie, I., Cancela, P., Font, F., Rocamora, M. & Serra, X. DCASE-models: a Python library for computational environmental sound analysis using deep-learning models. In Proceedings of the Fifth Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2020), 240-4, (2020).

[24] Tafii, Z., R., Liutkus, A., Fabian-Robert, S., Mimilakis, S. & Bittner, R. The MUSDB18 corpus for music separation. (2017). at: <https://doi.org/10.5281/zenodo.1117372>

[25] Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K. R., Dähne, S., & Kindermans, P. J. iNNvestigate neural networks! *Journal of Machine Learning Research,* 20, (2019).

[26] McFee, B., Colin, R., Dawen, L., Ellis, D., McVicar, M., Battenberg, E. & Nieto, O. librosa: Audio and music signal analysis in python. *In Proceedings of the 14th python in science conference*,18-25. (2015).

[27] Fabian-Robert, S., Liutkus, A. & Nobutaka, I. The 2018 Signal Separation Evaluation Campaign. Latent Variable Analysis and Signal Separation: 14th International Conderence, 293-305, (2018).

[28] Jindong, G. , Yinchong, Y., & Volker, T. Understanding Individual Decisions of CNNs via Contrastive Backpropagation. *arXiv.org* (2019).