ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MATHEMATICS SECTION

COMPUTATIONAL SCIENCE AND ENGINEERING

MASTER THESIS

Audio Blind Source Separation for Noise Reduction

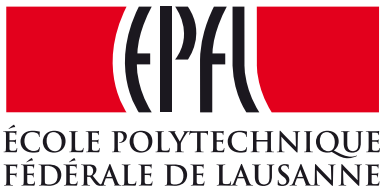
Vincent POLLET

Carried out in the Signal Processing Laboratory 2 (LTS2)

Supervisors: Pr. Pierre VANDERGHEYNST
Dr. Benjamin RICAUD
Helena PEIC TUKULJAC
Rodrigo PENA

In collaboration with Logitech Europe S.A.
Supervisor: Dr. Milos CERNAK

January 17, 2019



Abstract

In video-conferencing applications, it is crucial for user experience to attenuate or remove undesirable audio sources interfering with speech signals. Speech enhancement can be achieved using source separation algorithms to separate speech from all other competing sources. State of the art audio source separation algorithms are machine learning based and require to be trained on massive amount of audio data. For training, audio mixtures and the corresponding separate sources recordings must be compared, therefore the algorithms are typically trained on artificially generated mixtures which limits the amount and variety of available data. Recently, an audio separation algorithm trained on *weakly* labeled audio recordings has been proposed [12, 13]. Weakly labeled recordings refer to audio recordings whose audio content has been labeled, and is much more abundant and easy to produce than artificial mixtures from clean sources. This thesis investigate the use of this method for speech enhancement. The algorithm attains a PESQ MOS LQO score of 1.9 on mixtures of non-overlapping sources, and of 1.2 on mixtures of overlapping sources.

Contents

Introduction

Noise reduction is one of the broader and most widespread topic of signal processing, as any signal is bound to be corrupted up to a certain point by noise, whether it is during its recording, digitization, processing, transfer or even storage. If the effects of this noise becomes too important, the information contained in the signal of interest might be completely lost. The topic of noise removal or noise reduction thus regroups all the technology developed in order to minimize the effect of noise by preventing it from corrupting the signal in the first place, or removing its presence from a signal after it has been corrupted. The later - removing noise from a signal - can be seen as a particular instance of a source separation problem where the goal is to recover two different signal sources once they have been mixed together, one source been the signal of interest and the other noise. Source separation methods can thus be of interest when designing a noise reduction pipeline if the signal considered is bound to be corrupted by an important level of noise.

Logitech produces video-collaboration devices, and is therefore interested in providing the best signal quality to its customers. In the field of video-collaboration, video and audio data must be recorded from a user and transmitted to another user in real time, without deterioration of the signals during the entire processing. The noise been generated by the signal processing is typically controlled to minimize the impact on the users ; however the signal can be corrupted by noise as early as its recording: when a user is handling a meeting in a environment that is noisy to begin with, for instance in a cafeteria where other people are talking, in a park with dogs barking, at home with air conditioning, etc... Virtually any kind of audio noise can corrupt the signal of interest - the user's speech - at the time of the recording and there is a need for a technology able to remove this noise. Source separation techniques naturally come to mind to perform the noise reduction in this situation.

Speech enhancement is currently a very active area of research due to the increasing use of video calls in both private and professional environments, as well as the democratization of tools relying on automatic speech recognition such as personal assistants. Noise removal technique can be tailored to enhance the speech quality for this specific applications. For instance removing periodic noises can be successfully achieved by using hand-tailored filters [3], but in an application such as video-collaboration, a vast variety of noises are expected and the type of noise appearing in a recording is not known in advance. Therefore general blind audio source separation methods which have been designed to handle the separation of audio sources without prior knowledge on the sources properties look promising.

Many algorithms have been proposed for tackling signal separation problems. One of the most widespread method, beamforming, can be used provided the sources are recorded by an array of sensors [31]. It is widely used for speech enhancement, and can be used for audio separation provided that the signal sources do not come from the same direction. Its performances therefore depend on the hardware used to capture the signals as well as the spatial distribution of the input sources. That second element can not be controlled in practice in the field of video-collaboration.

Other methods such as independent component analysis [9] aim at separating signal sources by finding statistically independent components. The method, as it was initially proposed, separates as many signal components as there are input channels, which makes its application in a field such as video-collaboration where the number of sources and input sensors might be unknown and can vary from a user to another difficult. The method has been extended to be applied in a single channel case [1] ; nonetheless its application to audio source separation remains difficult because many parameters that depend on the time-frequency characteristics of the signals must be tuned to obtain usable separation results. This aspect suggests that the source separation problem from a single channel - an under-determined problem - can be solved in practice by using additional information gathered from data.

With the recent democratization of machine learning many methods that learn a representation from the data to solve the problem have been proposed. Non-negative matrix factorization (NNMF) is a method that learns, in an unsupervised fashion, a decomposition of an audio spectrogram on a basis of spectral vectors [26]. The vectors can either be learned whenever the algorithm is applied to a new audio recording, yielding a basis for individual sources for this recording ; or the basis can be learned off-line on a training set and then used for the separation of new audio samples. The method can be integrated in a hidden Markov Model to account for temporal dependencies [21] and has been the state of the art for many years. It has recently been surpassed by other clustering approaches that are solely based on machine learning techniques [7, 4].
Deep learning methods have recently been proposed for the task of single channel audio separation. Auto-encoder approaches can be used for learning optimal separation masks (in time-frequency representation) from data using deep feed-forward or convolutional networks [6]. These method have the limitation that the number of sources that can be separated is fixed at the training stage, but they typically are faster than clustering methods. Recent research tries to combine both the clustering and the masking methodologies in hybrid models [16].

Supervised machine learning method performances heavily depend on the quantity and variety of data available for training. Unfortunately in the field of audio separation, the quantity of available data is limited. Indeed: in order to train a supervised model to separate an audio signal from noise, one needs examples of recordings containing both signal and noise *and* the corresponding sources in separate recordings. To achieve this, training is typically conducted on artificial mixes: multiple sources are used to create an artificial audio mixture whose individual sources are known. Therefore, the amount of data available for training is directly linked to the amount of clean recordings of each sources. While there exists a vast quantity of audio recordings containing several sources, the amount of very clean recording for a particular audio source can be quite limited. For instance, there exists many audio recordings containing both human speech and keyboard strokes, however the amount of audio recordings containing only keyboard strokes is in comparison much smaller. The data available for training a model (in a supervised manner) at separating speech from keyboard noise is therefore limited by the small amount of keyboard-only recordings. Moreover, the mixing step required to produce the training mixtures is crucial. It has to be performed in such a way that the generated mixtures are representative of the operating conditions of the model. In the video-collaboration field, there is an almost infinite variety of potential audio sources and a vast variety of mixing conditions of those sources (many possible signal to noise ratio, room acoustics, recording sensors...). For industrial applications,

a model should be very robust to this variety, which is difficult to ensure when the training data is artificially generated from a small set of examples.

Recently, an audio separation deep learning method trained on *weakly* labeled data has been proposed in [12, 13]. Weakly labeled audio recordings refer to audio recordings for which only the presence of the source is known, but the individual recordings of the audio sources are not available. This is a major difference in training procedure compared to frameworks used to train auto-encoders. It allows to train models using much more abundant data, and removes the mixture generation step otherwise required which is limiting the training data variety. In this thesis, we propose to investigate the performances of this method at the task of audio separation, and in particular its potential application to speech enhancement.

The thesis is organized as follows: in section 1 we describe the audio separation framework and the training procedure used for training on weakly labeled recordings. Several variations to the method proposed in [12, 13] are proposed. In section 2.1 we test the method on the TUT Rare Sound Event 2017 data set [20]. In section 2.2 we apply the method on a portion of AudioSet [5], a massive data set of weakly labeled audio (and video segments, not used here), and evaluate the method at the task of speech separation. In section 3, we consider a different audio processing front-end, per-channel energy normalization, for processing the audio passed to the audio separation models.

1 Audio separation models trained on weakly labeled data

1.1 Audio separation framework

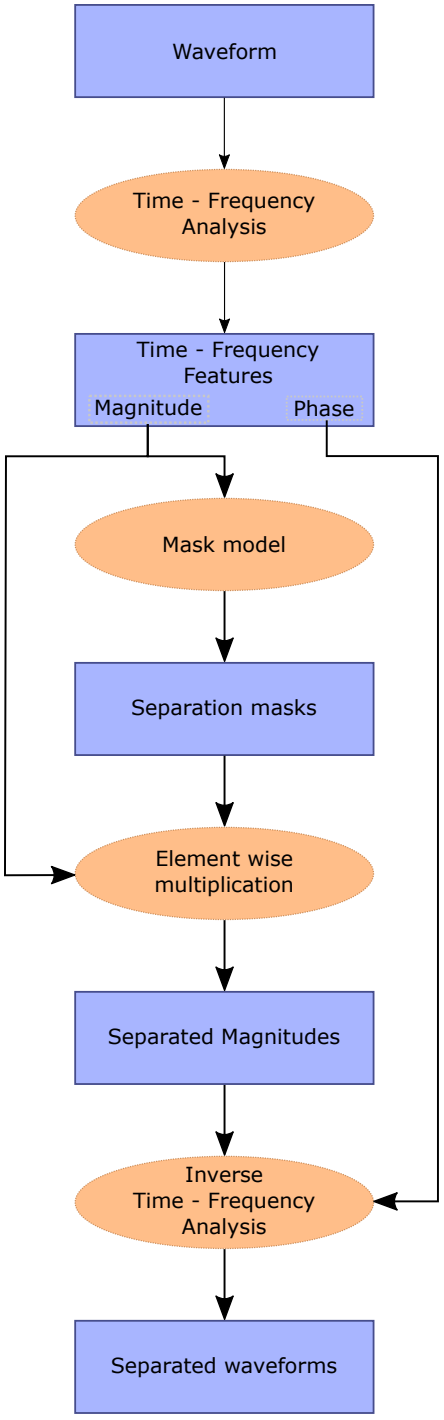
As many audio separation algorithms, the investigated method works with the time-frequency representation of audio signals and aims at producing "separation masks" that capture the contribution of each audio source to a mixture. A diagram of such a method is shown in figure 1.1a. The separation pipeline goes as follows:

- A time-frequency representation of the audio mixture is computed and the magnitude and phase information are separated.
- The magnitude representation is fed to the "mask model" which outputs, for each audio source, a ratio mask whose values are the ratio of the magnitude of the corresponding source by the magnitude of the mixture.
- Each mask is multiplied to the mixture magnitude, producing separated magnitudes for each source.
- The separated magnitudes are combined with the mixture phase, then converted back to audio waveform.

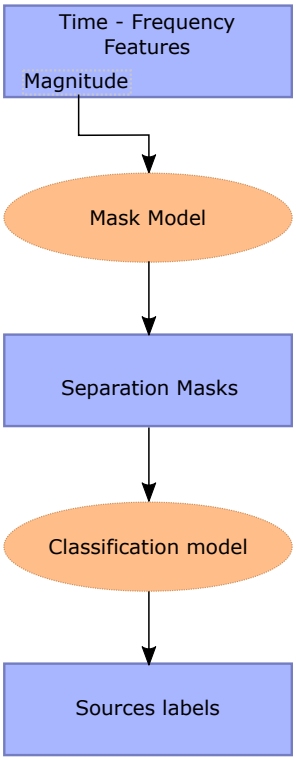
Usually, the mask model is trained by comparing its outputs to the ground-truth ratio masks of the sources, which can be computed because the separated sources of the training mixtures are known. In our case the training is performed using weakly labeled data, thus the ground-truth ratio masks are not known and the training procedure must be adapted.

In the training stage, the separation masks are passed to another model, which we call the "classification model". This model task is to predict, given a separation mask, the presence or the absence of the source corresponding to the input mask in the mixture. The training framework is illustrated in figure 1.1b. The mask and the classification model are trained jointly, which allows to train the mask model only using weak labels.

Of course, since the mask model outputs are not directly compared to the ground-truths during training, the training procedure does not directly optimizes the separation masks. The fact that mask model actually learns separation masks has to be ensured by choosing a smart architecture of both the mask and the classification models.



(a) Diagram of the audio separation framework based on magnitude masking.



(b) Diagram of the framework used to train a source separation model using weakly labeled data.

FIGURE 1.1

1.2 Mask model

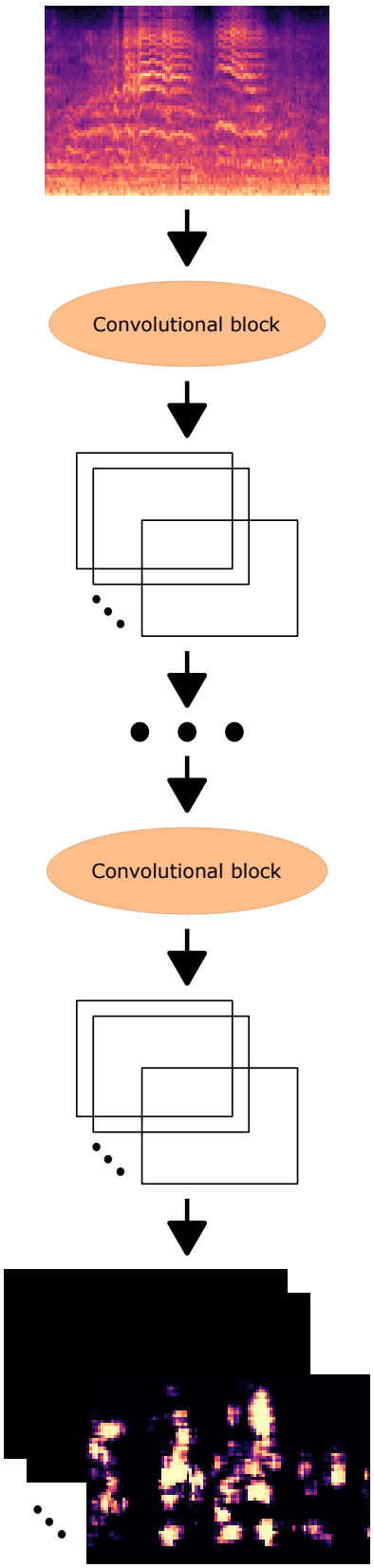


FIGURE 1.2: Diagram of the mask model architecture.

The mask model purpose is to produce, given the magnitude of the time-frequency representation of an input example, ratio masks predicting the contribution of each possible source at each point of the time-frequency representation of the mixture. In order to do this, the model needs some context information about a point surroundings. Convolutional neural networks (CNN) have been shown to perform very well at image segmentation tasks (that is for each pixel of an image, determining to which category this pixel belongs) [14] and are natural candidates for the task. The chosen architecture for the mask model is a stack of convolutional blocks as is illustrated on figure 1.2. Each convolutional block starts with a convolution layer using 5×5 kernels, and a stride of 1. Leaky-ReLU [17] functions are used as non-linearity after all convolution layers of the mask model, except the last one. Indeed, the mask model outputs are expected to be ratio masks, and to ensure that they can be interpreted this way a sigmoid or a softmax non-linearity is used as the final non-linearity. On the one hand, the sigmoid activation lets the model predict the ratio masks of each source independently. On the other hand, the softmax activation ensures that the sum, in each point of the time-frequency representation, of the separation masks of all sources is equal to one ; thus ensuring that the sum of the separated magnitudes is equal to the magnitude of the mixture.

The softmax activation choice is particularly adapted when the sources to separate

are the only possible sources in the input mixture, since in this case the magnitude of the mixture is only formed by the combination of the source to separate. In the case where sources to separate do not account for all possible sources in the mixture (for instance we train a model to separate speech and laughter, but the training mixture also contain music, which the model is not trained to separate), the sigmoid activation gives the model more flexibility as it can chose to discard the mixture magnitude, while a softmax activation forces all the mixture magnitude to be kept in the outputs.

In order to preserve the shape of the input throughout the model, zero-padding is applied to the input before each convolution layer and pooling layers are not used in the network. Batch normalization is used to accelerate training [10] and a dropout regularization is applied during training after each convolution layer, except the last one, with a probability of 0.1, in order to improve the model generalization capabilities.

1.3 Classifier model

In [12, 13], 3 kind of classification models architectures are proposed, all of them been a global pooling operation that in one step sums up the information contained in a separation mask down to a label prediction:

- GMP: Global maximum pooling. It is equal to the maximum value of the mask.
- GAP: Global average pooling. It is equal to the average value of the mask.
- GWRP: Global weighted rank pooling. Let $\{i_1, \ldots, i_n\}$ be the indices sorting the mask values in descending order. Then

$$\text{GWRP}(\text{mask}) = \frac{1}{Z(d_c)} \sum_{j=1}^{T \times F} (d_c)^{j-1} \text{mask}(i_j)$$

  where T and F are the masks dimension in time, respectively frequency, and $Z(d_c) = \sum_{j=1}^{T \times F} d_c^{j-1}$ is a normalization term. GWRP is an intermediate between GMP and GAP, allowing to give more importance to the biggest values in the mask than GAP (which uses an equal weight for each mask value) but less than GMP which only considers the maximal value. In all our experiment the value of d_c has been set to 0.999.

In addition to these 3 pooling operations, the following architectures have been studied:

- FC: A fully connected layer connects the mask values to a single output. The result is then passed through a sigmoid non-linearity.
  This method is motivated by the fact that the GMP, GAP and GWRP models all use pre-set weights when aggregating the mask values. In the GMP case the weight of the maximal value is 1 and the weights of the other values is 0. In the GAP case all the weights are 1. In the GWRP case the weights values start at 1 for the maximal value and then decrease according to the value of d_c^{j-1}. The idea behind the fully connected layer is to let the classification model learn the weights to attribute to each frequency bin on its own.

- FC-sort: A fully connected layer connects the *sorted* values of the mask, in descending order, to a single output. The result is then passed through a sigmoid non-linearity.
  One direct drawback of using an FC classifier is the that it is not translation invariant. By sorting the values in descending order before going through the fully connected layer, we allow for the connections from the masks to the output to change from one training example to the next depending on the masks values, in a similar fashion to GWRP ; however the classifier weights can now be learned by the network.

- "x"-layer-CNN: A convolutional neural network with x convolution layers, using an aggressive pooling strategy and followed by a fully connected layer with a single output and non-linearity. This architecture has the capacity of leveraging the information contained in the 2 dimensional nature of the masks, information which is necessarily discarded by the FC and FC-sort and pooling based classifier. Using a convolutional neural network with aggressive pooling strategy allows to aggregate the values in the masks while preserving the 2d structure information.

- RNN: a recurrent neural network. The separation masks can be seen as time series where the series values at a time step are the mask frequency bins at that time step. Recurrent neural networks design make them suitable for treating such time series.

Several classifiers performances will be compared in section 2.1. For simplicity these classifier models have been named by their architecture components. Table 1.1 reports the models hyper-parameters.

1.4 Time-Frequency representation

The time-frequency representation used as input to the mask model is the log Mel spectrogram, which is a very common representation for audio processing using neural networks.

The discrete short term Fourier transform (DSTFT) of the audio waveform is first computed and the magnitude and phase component are separated. The magnitude component is the spectrogram of the audio waveform. The spectrogram representation is linear in frequency, but human perception of frequencies is linear for low frequencies and logarithmic in higher regions. To account for this, it is common practice in audio processing to scale the spectrogram using the Mel scale, in order to work with frequency bins that are perceptually equidistant. A logarithm transform is then applied to the spectrogram values in order to reduce the dynamic range of the values. While this is a common audio processing front-end, there is no guarantee that it is optimal for the task at hand. In section 3, the effect of an other audio processing front-end, the per-channel energy normalization, is explored.

The DSTFT is computed using the scipy library [11] signal module. All used audio signals were re-sampled to 16 kHz before processing. A window size of 64 ms (1024 samples) was used, with 32 ms overlap and no padding. A Hamming window function is used to reduce spectral leakage. These parameters ensure that the reconstruction of the waveform is possible using the discrete inverse short term Fourier transform (DISTFT) when the separated waveform need to be constructed.

Model name	Hyper-parameters
GMP	-
GAP	-
GWRP	$d_c = 0.999$
FC	-
FC-sort	-
1-layer CNN	5×5 kernel are used, with a stride of 3. leaky-ReLU non-linearity is used.
2-layers CNN	All convolution layers hyper-parameter values are identical to the 1-layer CNN
3-layers CNN	All convolution layers hyper-parameter values are identical to the 1-layer CNN.
RNN	A LSTM [8] architecture is used, with a hidden size of 40. The last hidden state is passed to a fully connected layer with an output size of 1 and sigmoid non-linearity.

TABLE 1.1: Summary of the different classification model architectures.

The mapping to the Mel scale is accomplished by building a filter bank of triangular filters using the librosa library [18], then multiplying the filter bank matrix and the spectrograms. 64 Mel filters were used in all experiments.

Mask application
The input to the mask model are the log Mel spectrograms of the mixtures. Therefore the separation steps could be accomplished by the following steps. First applying the masks on the mixture log Mel spectrogram to get separated spectrograms for each sources. Then taking the exponential of the separated log Mel spectrograms to return in linear domain, and multiplying by a normalized pseudo-inverse of the Mel filter bank matrix to re-scale the separated spectrogram to a linear frequency scale, and finaly compute the DISTFT to generate the corresponding audio waveform. However, as argued in [29] and verified in practice in section 2.1.1 applying the masks in this fashion is not optimal because small errors in the mask values are exponentially amplified when computing the exponential, and then spread on multiple frequency bins by the inverse Mel scaling. The effect is avoided by first mapping the masks to a linear frequency representation, and directly apply them on the mixture spectrogram in this representation.

1.5 Audio separation metrics

Objective evaluation of the quality of separated signals is a difficult problem, and a computational criterion describing the perceived separation quality of any kind of audio recordings is still to be found. In this thesis we will use the criteria defined in

[28] that can be computed for any kind of audio sources and are often reported for audio separation results.

- Signal to Distortion Ratio: (SDR): $\text{SDR} = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2}$
- Signal to Inference Ratio: (SIR): $\text{SIR} = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}$
- Signal to Artifact Ratio: (SAR): $\text{SAR} = 10\log_{10}\frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2}$

where s_{target} is the orthogonal projection of the separated source onto the target source, and e_{interf} is the orthogonal projection of the other separated sources onto the target source. We refer to [28] for the definition of e_{noise} and e_{artif}.

The SDR measures the relative importance (in dB) of the separated source with respect to its reference source. The SIR the relative importance between the separated source and other sources interfering. SAR quantifies the absence of distortions and burbling artifacts. Higher values of SDR, SIR and SAR indicate better separation performances. These criteria value presented in this thesis have been computed using the `bss_eval_images` function of the mir_eval library [24]. Due to rounding errors the precision of the criteria becomes imprecise for high SDR conditions.

These measurements do not always correlate to the perceived quality of the separated audio, and in the field of speech enhancement in particular it is often preferred to asses other measures such as the perceptual evaluation of speech quality score (PESQ MOS-LQO score) [25]. Unfortunately this measure is only defined to assess speech quality and can not be computed for measuring the separation quality of other type of sources. The PESQ MOS-LQO score ranges between 1 (very bad) to 5 (excellent) and reflects the average opinion score given by listeners to the speech quality.

Maximal performance: When computing the above metrics with the separated source equal to the target source, one obtain SDR, SIR and SAR scores around 20 dB and PESQ MOS-LQO score of 4.6.

2 Experiments

In this section, we report experiments conducted to evaluate the method and the different classification models architectures. The training of the models has been implemented using the pytorch deep learning library [22]. The code base is available at https://github.com/4p0pt0Z/Audio_blind_source_separation and the thesis results were obtained with the commit 69326f8. The cross-entropy loss is used for training the models against the weak labels. All presented models have been trained for 300 epochs, using the ADAM optimizer with an initial learning rate of 10^{-3}, decreasing by half every 50 epochs. Unless stated otherwise, a mask model containing 3 convolutional blocks is used.

2.1 TUT Rare Sound Events 2017 data set

The method has first been tested on mixes generated from the TUT Rare Sound Events 2017 data set [20]. The data set contains clean recordings of 3 audio events: baby cries, glass breaks and gunshots. The events are split in a training and an evaluation set, which contain 106, respectively 42, baby cries events, 96, respectively 43, glass break events and 134, respectively 53, gunshots events. The events are mixed with background scene recordings from the TUT Acoustic Science 2016 data set [19] to generate 4 seconds mixtures. The mixing procedure used in [12] has been re-used, producing 1008 mixtures for training and 134 mixtures for evaluation.

2.1.1 Audio separation method validation

As a first step to validate the audio separation framework, we train a mask model containing 3 convolutional blocs using GWRP as a classification model (which is the model achieving the best performances in [12, 13]). We evaluate the separation performance of this model when the mask is applied to the log Mel spectrogram of the mixture (and the separated log Mel spectrogram are converted back to audio waveform) or directly to the magnitude of the DSTFT (cf section 1.4).
With the first method, the model obtains an average SDR of 1.4, against 2.3 for the second method. Casual listening of the separated audio confirms that applying the masks to the log magnitude of the mixture causes the separated audio to be corrupted with high amplitude artifacts, that come from the exponential amplification of the separated log magnitudes. Therefore, in the following, the separated sources are obtained by applying the mask directly to the input mixture DSTFT magnitude.

2.1.2 Classification models performances

In this section we compare the separation performances of the method when the same mask model architecture (3 convolutional blocks) is trained using the different kind of classification models described in 1.3. The results are reported in table 2.1. A

separation mask example for the 3 audio events and all models is displayed in figure 2.1. The following observations can be made from the results:

- The SDR and SIR values of glass break separation are very low for all classification models. This suggests that the problem comes from the mask model which does not learn proper separation masks for this class.
- GAP and GWRP are the only classification models that achieve high SDR, SIR and SAR for the baby cry and gun shot classes. The FC and FC-sort, all CNN and the RNN models produce either values which are close to 0 or strongly negative.
- SAR values are positive for all classification models and audio events, so the separation framework does not create too much burbling artifacts. However, the values are much smaller for gun shot events than for the baby cry and glass break events, which can be imputed to 'holes' in the detected event regions as can be seen in the GWRP separations masks (figure 2.1c). The SIR scores are also very low for this class, showing that the separated gun shots are corrupted with other sources as well.
- The mask produced by training with the GAP classification model for the baby cry and gun shot classes appear to over-estimate the event presence, while the masks obtained with the GMP model (figure 2.1a) heavily under-estimate the event presence and the masks obtained with the GWRP model cover an area which is closer to the event region in the spectrograms. This confirms the observations described in [13].
- The masks obtained with GAP and GWRP models capture the general location in time-frequency of the events to separate. However they do not capture the fine structure of the event spectrogram such as the wave-like shapes visible in a baby cry spectrogram. They are more similar to binary separation masks than ratio masks, so we can expect the method to perform reasonably well on mixtures of non-overlapping sources, but to have difficulty handling the case of separating sources overlapping in time and frequency.
- The mask obtained when training with the FC, FC-sort, all the CNN and the RNN models do not correspond to separation masks, explaining the poor separation results. Indeed, the masks do not capture the presence of audio events with high values ; instead some of the masks seem to capture the opposite signal: (figure 2.1d and 2.1e for baby cry events, figure 2.1f, 2.1g, 2.1h and 2.1i for baby cry and gun shot events). Moreover, most of the values outside of the regions covered by events are around 0.5 which implies that background noise will be included in the separated audio resulting in poor SIR scores. Interestingly, though these masks do not capture the desired features from the input, the structure of the events spectrogram appears to be better preserved than in the masks obtained with GAP or GWRP models.

The fact that the masks obtained with non-pooling methods show behaviors which is the opposite of what is expected (events region are associated with values close to 0) is a sign that the classification model has too much expressive power. The fully connected, convolution or recurrent layers of the classification model can indeed learn negative or zero weights, which affects the learning of the mask model. With global pooling strategies, the values of the masks, which are always positive,

were summed up with positive weights. Therefore, if a region of the mask had to be zeroed for the separation, the mask was forced to take zero value in this region, otherwise a erroneous contribution would be made to the pooling output. If the classification model can use negative weights, errors in the mask can be corrected by the classification model before the computation of the training objective. Therefore the mask model is not forced to learn the separation masks to optimize the training objective and instead converges to a different state. It therefore appears that the classification model should perform some kind of positive combination of the values in the mask for the method to work properly.
Nonetheless, the representation learned by the mask model when the classification model is too expressive, though not suited for audio source separation, shows that it is able to capture much more precise signal structures than what is obtained when using global pooling operations (obtained masks are closer to binary masks than ratio masks), which discards all the 2 dimensional information of the mask model output. It is possible that a middle ground solution that would take advantage of the properties of pooling operations as well as use the structural information that the mask model produces would improve the separation results, for instance a convolution layer which kernel weights are constrained to positive values. This would prevent the mask model from learning features unrelated to the ratio mask, but not necessarily prevent the mask model from taking non-zero value outside of the regions of interest as global pooling operation do.

To apply more constraint on the mask model, we can also add a regularization term to the training objective, for instance $\mathcal{L}1$ or $\mathcal{L}2$ regularization, that would penalize inadequate masks activation.

As seen in this section, even though the classification model is only used during the joint training of the mask and classification models, it has a enormous impact on the ability of the method to perform audio source separation. The joint training of the mask and classification model has been proposed as a solution for training the mask model using weakly labeled data ; however it is not required that the classification model *learns* at the same time than the mask model. The classification model could be trained alone on a data set of labeled ratio masks (the classification mask is trained to predict a audio source presence given a ratio mask). Once its performance are deemed satisfactory, the mask model could be trained using the classification mask, but without re-training the latter. This set-up could still be used for training the mask model from weakly labeled data, but it requires the existence of a data set of labeled ratio mask for pre-training the classification model.

	SDR (dB)			SIR (dB)			SAR (dB)		
	Baby Cry	Glass Break	Gun Shot	Baby Cry	Glass Break	Gun Shot	Baby Cry	Glass Break	Gun Shot
GMP	2.0 ± 1.1	-0.4 ± 0.6	0.55 ± 0.09	6 ± 10	-19 ± 18	-63 ± 46	7 ± 5	5 ± 3	$\mathbf{4 \pm 8}$
GAP	$\mathbf{8 \pm 4}$	-7.2 ± 1.7	-0.6 ± 2.7	9 ± 9	-13 ± 13	$\mathbf{-62 \pm 46}$	12 ± 4	$\mathbf{20 \pm 9}$	1.6 ± 3.8
GWRP	7 ± 4	-7 ± 2	$\mathbf{4 \pm 3}$	$\mathbf{13 \pm 10}$	-19 ± 17	-64 ± 49	10 ± 5	13 ± 7	1.7 ± 4.2
FC	-0.04 ± 0.8	$\mathbf{0.9 \pm 1.0}$	-4 ± 3	-9 ± 7	-11 ± 20	-63 ± 45	12 ± 4	7 ± 3	3 ± 6
FC-Sort	-0.8 ± 1.2	-2.7 ± 1.5	-0.9 ± 1.2	-10 ± 4	-12 ± 13	-69 ± 45	$\mathbf{16 \pm 4}$	19 ± 9	1.9 ± 4.2
1-layer CNN	0.15 ± 0.3	-1.6 ± 1.6	-4 ± 2	-8 ± 11	-12 ± 16	-73 ± 46	1.2 ± 1.5	14 ± 7	2 ± 5
2-layers CNN	-0.2 ± 1.0	-5.6 ± 1.7	-2.8 ± 1.7	-10 ± 7	-18 ± 15	-71 ± 45	7 ± 2	11 ± 5	2 ± 5
3-layers CNN	-1.5 ± 1.3	0.6 ± 1.6	-1.7 ± 2.0	-11 ± 8	$\mathbf{-10 \pm 17}$	-71 ± 47	10 ± 3	9 ± 4	1.3 ± 3.1
RNN	-0.7 ± 0.9	-6.0 ± 1.8	-2.6 ± 1.6	-10 ± 5	-16 ± 17	-66 ± 44	13 ± 3	15 ± 7	2 ± 5

TABLE 2.1: Separation performances of a mask model with 3 convolutional blocks, when trained with different kind of classification models. The mask model last non-linearity is sigmoid. The confidence interval indicates one standard deviation.
The highlighted values are the best performance measured across the trained models (maximum of each column).

2.1.3 Mask model training optimization

In this section we investigate modifications to the mask model training framework that can boost the performances of the audio separation.

As briefly mentioned in the previous section, the loss function can be modified to include a term of the form $\sum_j \mathbb{1}_{j \notin \text{label}} \sum_{t=1}^{T} \sum_{f=1}^{F} |M(t,f)|$ that penalizes a masks activation when the corresponding source is not present. The new training objective is a linear combination of the cross entropy loss and and the regularization term:

$$\mathcal{L}(\text{label}, \text{prediction}) = (1-\lambda)\mathcal{L}_{CE} + \lambda * \sum_j \mathbb{1}_{j \notin \text{label}} \sum_{t=1}^{T} \sum_{f=1}^{F} |M(t,f)| \qquad (2.1)$$

where $\mathcal{L}_{CE}$ is the cross entropy loss. $\lambda = 0.3$ has been used in practice for all experiments.

The data set is composed of 3 audio event classes which are mixed with background noise. If we set the mask model to also produce a separation mask for the background class (and the classification model to predict the presence of the background class), then our model predicts a segmentation mask for each possible audio sources in the training mixtures. In this set-up, it makes sense to use the softmax function as last non-linearity of the mask model (cf section 1.2). Indeed, we know for a fact that the magnitude of the mixture has to be the sum of all the sources separated by our model, which can be enforced in the mask model structure by the softmax activation.

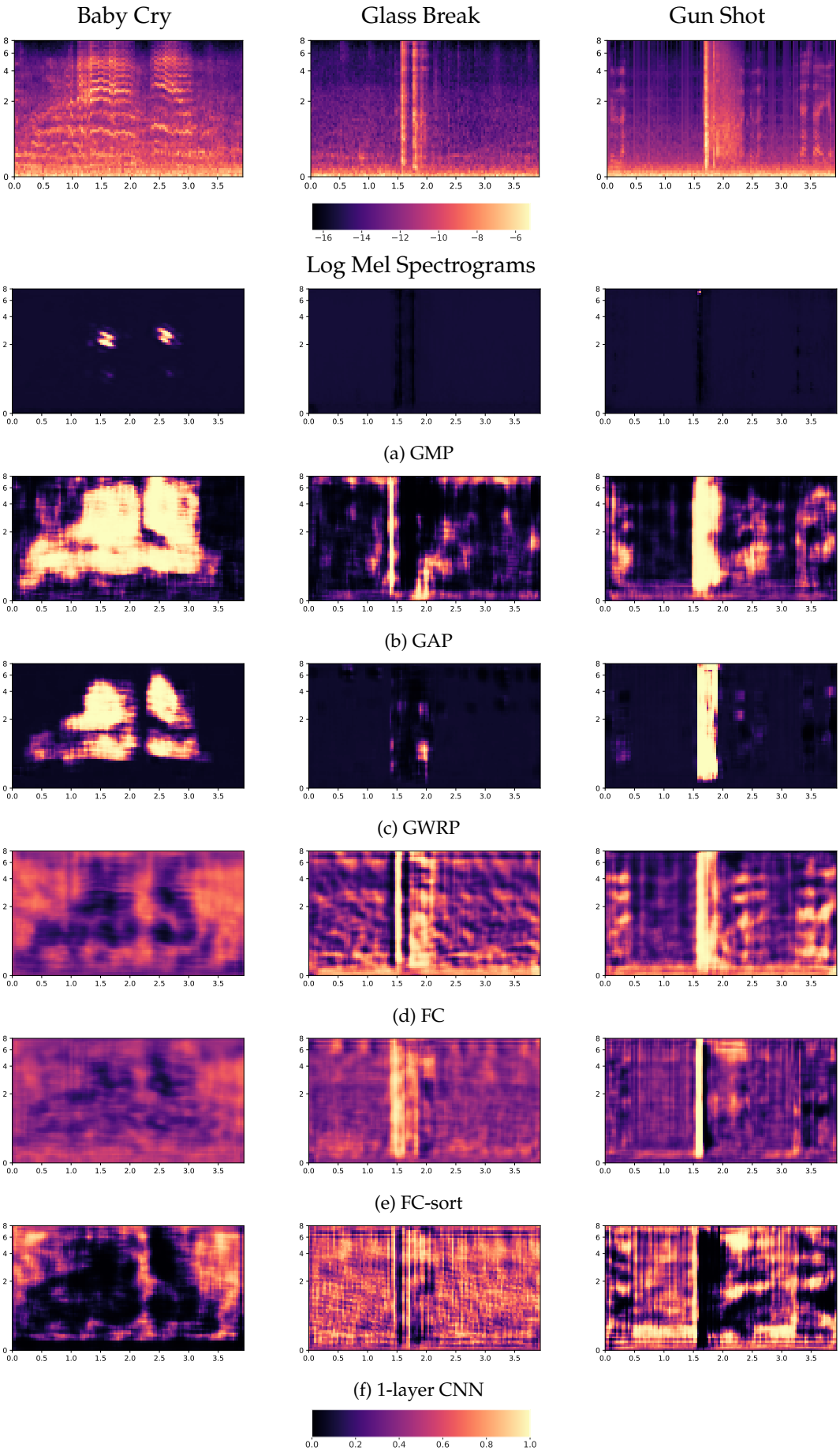
Finally, we investigate the effect of the number of convolutional blocks in the mask model on its separation performances when it is trained jointly with a GWRP classification model (the best performing classification model so far).

Modification results

Separation performances of a GMP, GAP and a GWRP model with the updated loss function and softmax activation as the mask model last block activation function are reported in table 2.2. The performances of mask models with different number of convolutional blocks (trained with a GWRP classification model) are also reported. The masks corresponding to these model predictions are shown in figure 2.2.

For the baby cry and the gun shot classes, SDR, SIR and SAR scores improved by 1 or 2 dB with respect to models trained with the previous procedures. In comparison to the previous training procedure, the masks produced with this modifications take smaller values (closer to 0) at points which are not part of an audio events (figure 2.1b and 2.2b for the baby cry class for instance).

Increasing the number of convolutional blocks in the mask model does not change much the source separation results. However as the number of blocks increases, the separation masks seem to preserve less and less the structure of the audio events: on figure 2.2f, the baby cry mask is detected as a single region, while on figure 2.2d the two clusters are distinguished. When using only 2 convolution blocks, the masks activated areas is smaller than the event area ; possibly because the receptive field of a point in the mask with respect to the input is too small compared to the baby cry event area. This suggests 3 convolutional blocks is an optimal choice for this tasks as it corresponds to a big enough receptive field for the separation masks, but is a small enough number of blocks for the information not to be averaged too much. This last point in particular could be an indication that residual connections between the mask model layers could help increase the precision of the separation masks.



(a) GMP

(b) GAP

(c) GWRP

(d) FC

(e) FC-sort

(f) 1-layer CNN

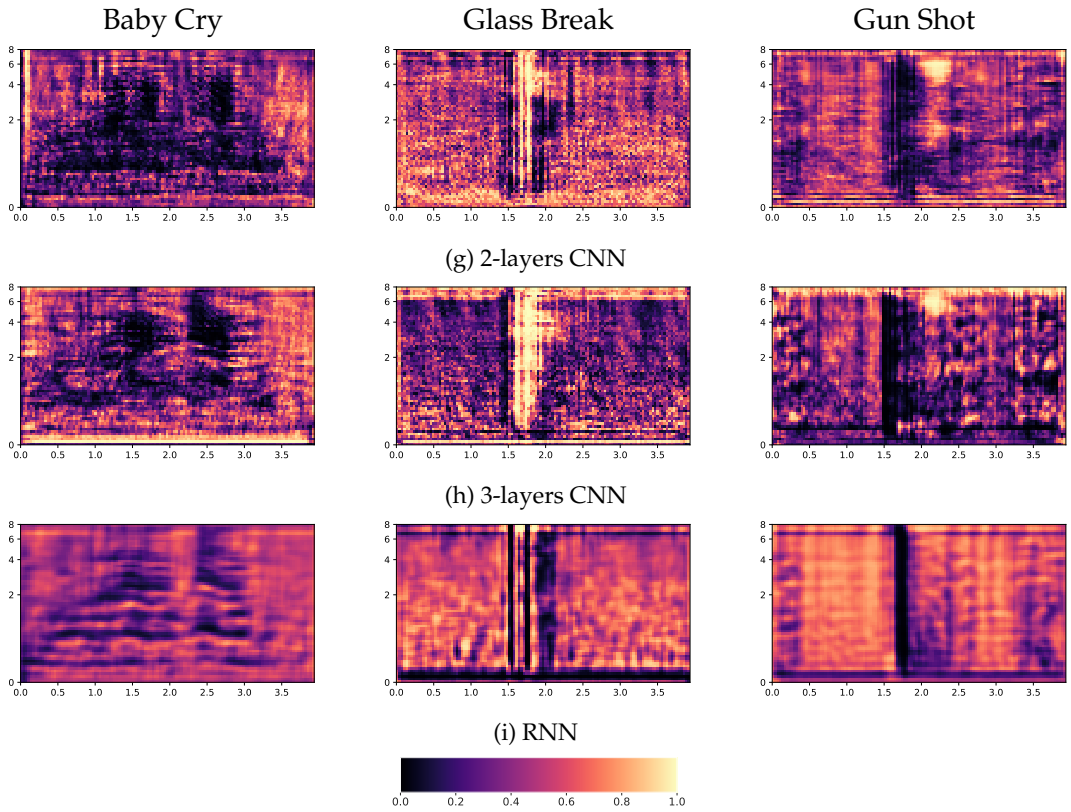
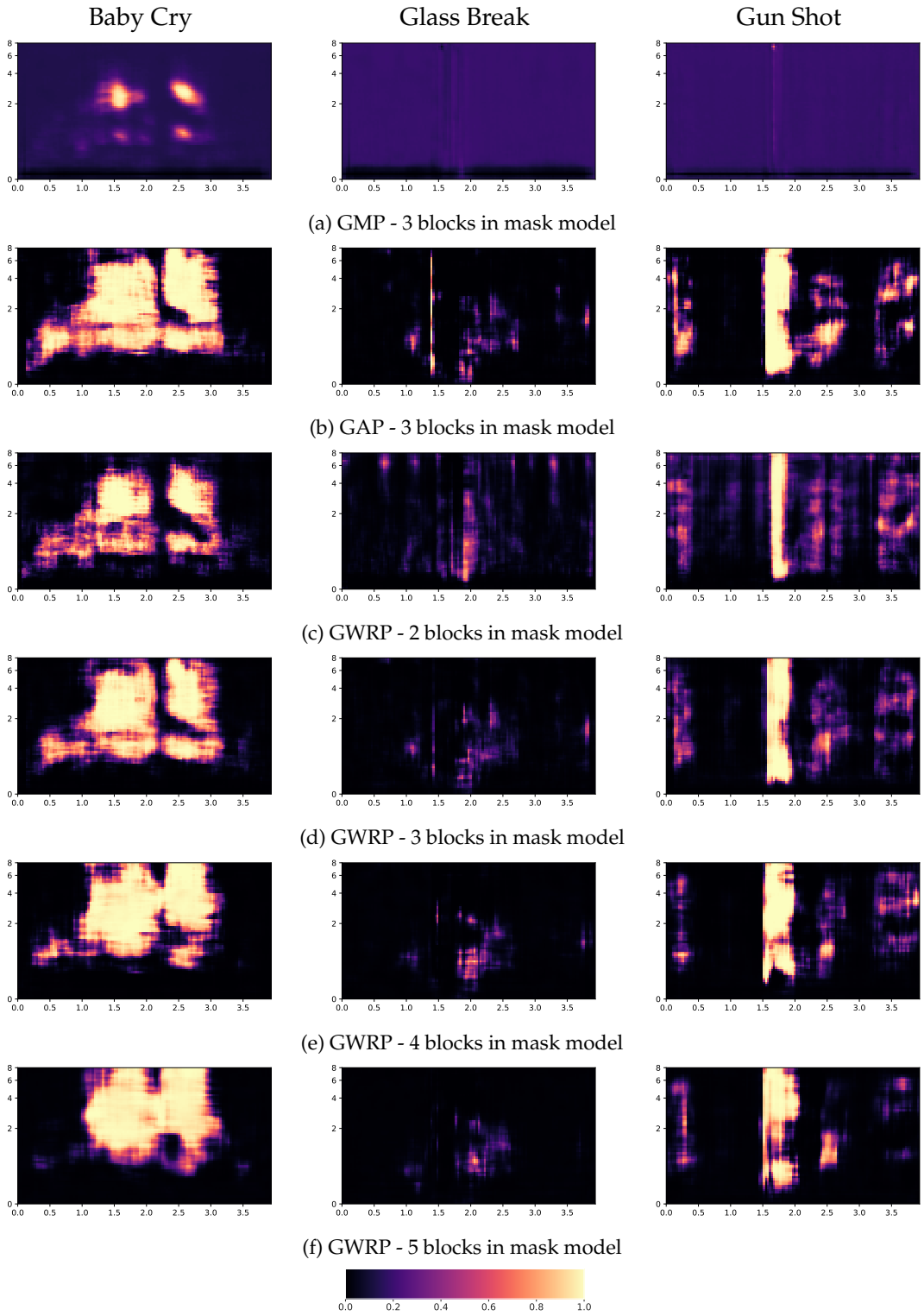


FIGURE 2.1: Separation masks obtained when training the mask model jointly with the different classifiers models. The mask model contains 3 convolutional blocks and its last layer activation function is sigmoid. Each column displays, at the top position, the log Mel spectrogram of an audio event. Below this event are displayed the separation mask corresponding to this source. The Horizontal axis represents time (in seconds) and the vertical axis represents the center of the Mel frequency bins (in kilo-Hertz)

	SDR (dB)			SIR (dB)			SAR (dB)		
	Baby Cry	Glass Break	Gun Shot	Baby Cry	Glass Break	Gun Shot	Baby Cry	Glass Break	Gun Shot
GMP 3 blocks in mask model	4 ± 2	$\mathbf{-4.7 \pm 1.7}$	1.1 ± 0.3	9 ± 10	$\mathbf{-16 \pm 16}$	-64 ± 46	11 ± 3	$\mathbf{17 \pm 8}$	$\mathbf{3 \pm 6}$
GAP 3 blocks in mask model	$\mathbf{11 \pm 5}$	-6.7 ± 1.9	3 ± 4	19 ± 12	-24 ± 16	-62 ± 51	12 ± 5	13 ± 8	1.2 ± 3.1
GWRP 2 blocs in mask model	8 ± 5	-7.6 ± 1.8	$\mathbf{4 \pm 3}$	20 ± 12	-24 ± 18	$\mathbf{-62 \pm 52}$	11 ± 5	16 ± 8	1.4 ± 3.4
GWRP 3 blocs in mask model	10 ± 5	-7.8 ± 1.6	$\mathbf{4 \pm 3}$	$\mathbf{21 \pm 11}$	-23 ± 18	-64 ± 52	12 ± 6	$\mathbf{17 \pm 8}$	1.1 ± 2.9
GWRP 4 blocs in mask model	10 ± 5	-7.9 ± 1.6	1.6 ± 1.9	$\mathbf{21 \pm 11}$	-23 ± 18	-68 ± 52	$\mathbf{13 \pm 5}$	17 ± 9	0.7 ± 2.3
GWRP 5 blocs in mask model	10 ± 5	-7.9 ± 1.5	1.5 ± 2.0	18 ± 12	-22 ± 15	-67 ± 47	11 ± 6	$\mathbf{17 \pm 8}$	0.9 ± 3.1

TABLE 2.2: First and second row: Separation performances of mask models when trained jointly with GMP and GAP classification model.
Rows 2 to 5: separation performance of mask models containing different number of convolution blocks when jointly trained with GWRP classification models.
The mask model last block activation is softmax and a regularization term is used in the loss function with coefficient $\lambda = 0.3$.
The highlighted values are the best performance measured across the trained models (maximum of each column).



(a) GMP - 3 blocks in mask model

(b) GAP - 3 blocks in mask model

(c) GWRP - 2 blocks in mask model

(d) GWRP - 3 blocks in mask model

(e) GWRP - 4 blocks in mask model

(f) GWRP - 5 blocks in mask model

FIGURE 2.2: Separation masks obtained when training several mask model architectures jointly with the different classifiers models. The mask model last block activation function is softmax. The training is used using a regularization term with $\lambda = 0.3$. The Horizontal axis represents time (in seconds) and the vertical axis represents the center of the Mel frequency bins (in kilo-Hertz)

2.2 AudioSet database

In this section, we apply the method of audio source separation trained from weakly labeled data to a bigger data set, for which we do not have the reference sources that compose the audio mixtures. The used data set is a portion of AudioSet [5], a huge audio and video data set extracted from YouTube videos. AudioSet is composed of around 2.1 millions segments of approximately 10 seconds. Each segment is assigned labels indicating the presence or absence of the 632 audio event classes which are currently part of the set ontology.

For this experiment, only the audio segments of the following classes have been used: fire alarm (921), smoke alarm (548), dog bark (2632), tire screeching (1557), car honk (3707), baby laugh (863), baby cry (2390), glass break (845) and (computer) keyboard (2042) ; totaling 15 505 recordings and around 43 hours of audio. These segments have been annotated with segmentation information by CloudFactory, a company specialized in data labeling: for each recording, the precise starting and ending point of the audio events were obtained using the ELAN annotation software [23]. Using this information, 3.3 seconds (independent) segments have been extracted from all the 10 seconds recordings, with the corresponding weak labels. In addition to the before mentioned classes, the presence of speech has also been annotated as it appeared in the segments.
The distribution of the classes in the segment is illustrated on figure 2.3. Since 3.3 second segments have been extracted from the 10 seconds initial segments, many of the produced segments actually do not contain an event from any of the classes (referred to as 'other'). The classes are not mutually exclusive: a segment can have multiple labels. The overlapping of the initial classes is almost negligible (except for the pair fire alarm - smoke alarm, figure 2.4), but the speech class often appears together with other classes.

The most successful model architecture reported in section 2.1.3: a mask model using 3 convolutional blocks and softmax activation function in its last block, has been trained using this data. The separation masks corresponding to examples of each class in the data set are shown in figure 2.5. Unfortunately, since the reference sources of the audio mixtures are not available, it is not possible to compute SDR, SIR and SAR scores.

The separation masks obtained show that the separation method has captured the most representative characteristics of the majority of the audio events, as the masks highlight areas containing the most important part of the audio events.
However for the fire alarm, smoke alarm, honk, tire screeching and glass break examples, the separation masks covers an area which is much smaller than the area of the audio event. This certainly translates to poor SDR performances because parts of the audio event will be lost during the audio separation. This is confirmed by casual listening of the separated audio files. On the contrary, the separation masks examples of the speech, keyboard and dog bark classes seem to cover a larger area than their audio event, which translates to poor SIR performance as the separated audio will contain parts of other sources.
Remarkably, the classes for which the masks are broader than the audio events are the one which are the most frequent in the data set. We can suspect that the model is biased toward these classes because of the imbalance of the training set: classes that appears more frequently during training are assigned broader areas, while classes which appear rarely like glass break are assigned very small areas. A data balancing strategy could be incorporated in the training procedure to reduce this unwanted

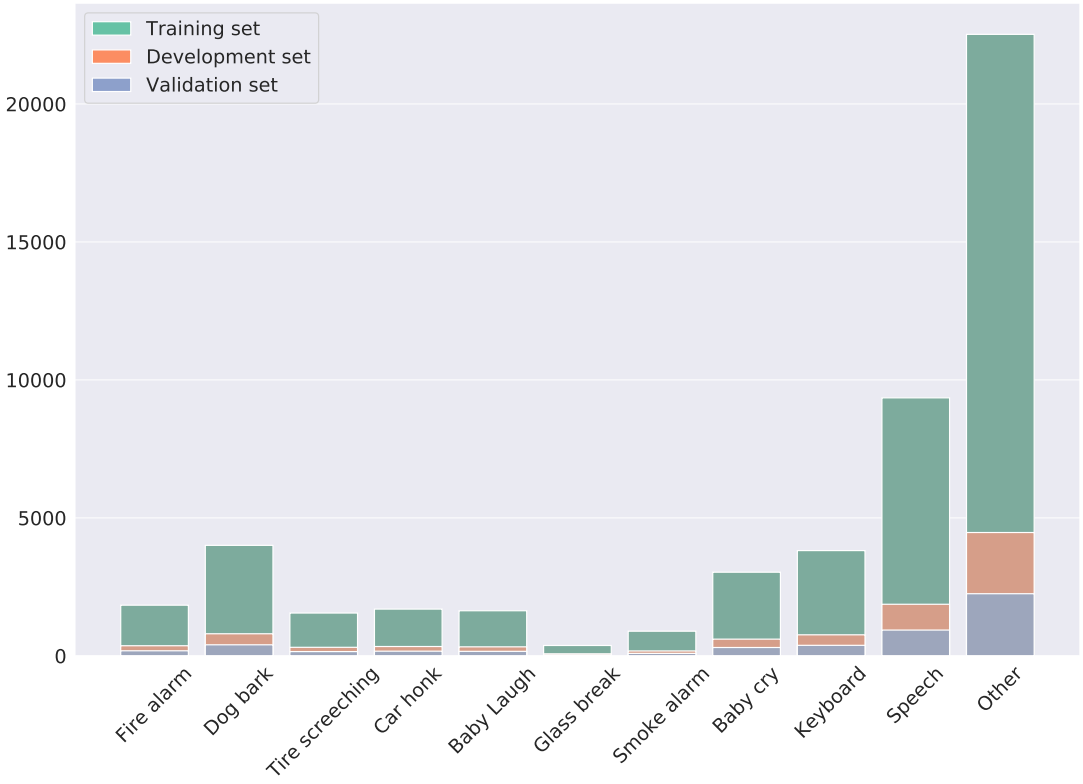


FIGURE 2.3: Distribution of events in the re-labeled portion of AudioSet used for training. To monitor the models during the training, it is useful to regularly evaluate it against unseen examples. To achieve this, the data set is split in disjoint training (80%), development (10%) and validation (10%) sets.

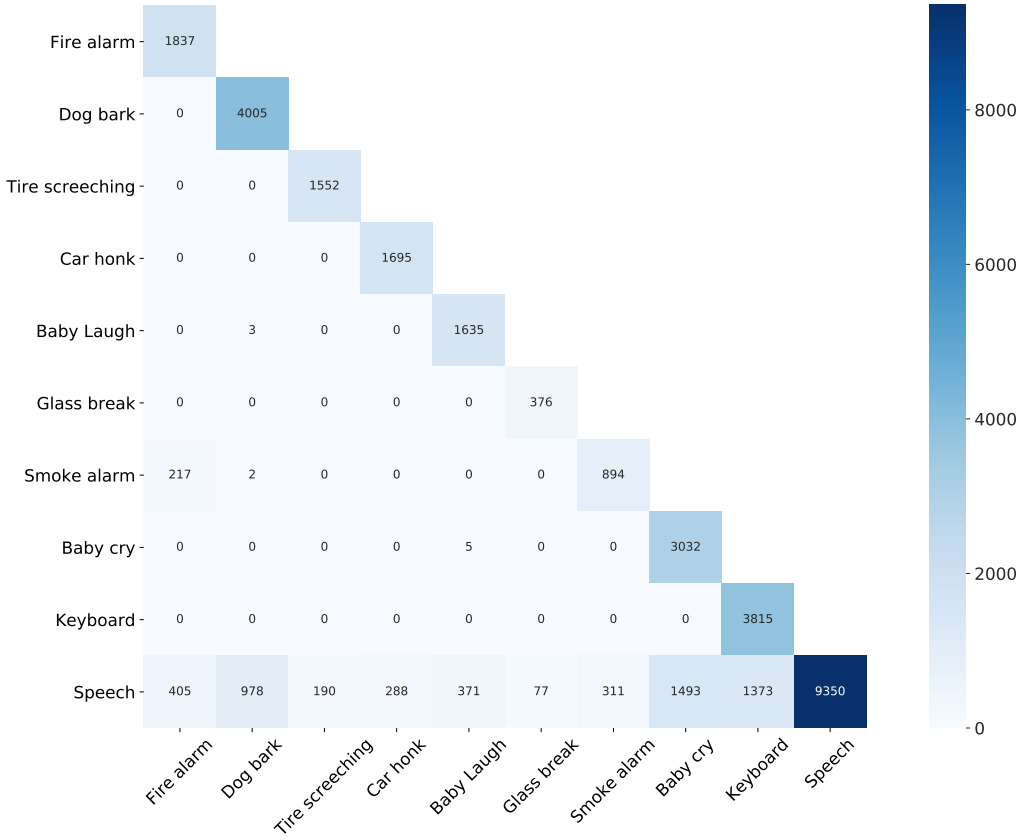


FIGURE 2.4: Overlap distribution of the events in the re-labeled portion of AudioSet used for training.

	non-overlapping mixture	overlapping mixture
GWRP	1.9	1.2
RNNoise	1.3	1.5

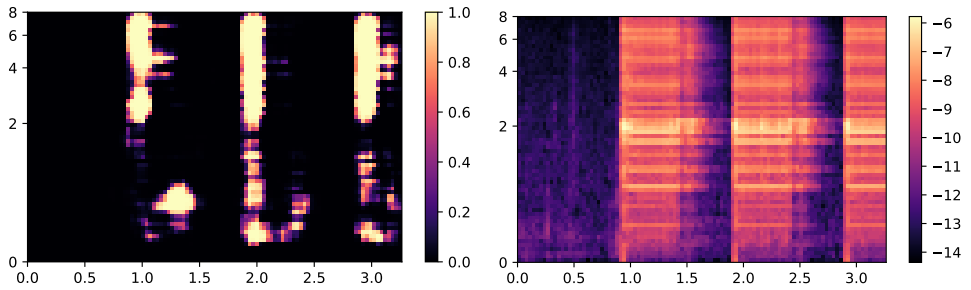
TABLE 2.3: PESQ MOS-LQO scores of GWRP model trained on the AudioSet subset, calculated on the separated speech extracted from the audio files corresponding to the log Mel spectrograms of figure 2.6

bias.

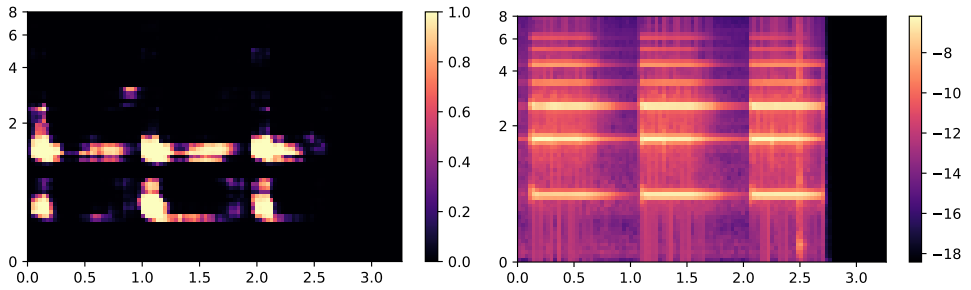
Speech enhancement
One of the most frequent noise in the field of video-collaboration is keyboard noise, which corrupts speech signals. Figure 2.6 regroups two examples of this specific case. Figure 2.6a shows the spectrogram of a recording containing *non-overlapping* segments of speech and keyboard noise. Figure 2.6b displays the spectrogram of the separated speech obtained with model described above. The separation step has successfully selected the area of the spectrogram containing speech and increased their magnitude relatively to the background. The area containing keyboard noise have been lower in comparison, but they have not been completely removed. The area containing speech also appears blurred: the separation method reduces the quality of the recording.
On figure 2.6c, the spectrogram of a recording containing *overlapping* segments of speech and keyboard ; the spectrogram of the corresponding separated speech is displayed on figure 2.6d. In this case, separation has enhanced all the part containing speech, amplifying the overlapping keyboard noise at the same time.

The PESQ MOS-LQO scores have been computed on the audio examples corresponding to the log Mel spectrograms of figure 2.6 and reported in table 2.3. For comparison, the open source model RNNoise proposed in [27] for speech enhancement based on a hybrid DSP-deep learning framework scores have also been computed. This model is trained to perform speech enhancement, unlike our method which was designed for general audio source separation, and is considered as a baseline algorithm for speech enhancement. The GWRP model performs better than RNNoise on the recording where speech and keyboard do not overlap, but it performs worse on the segment with overlapping events. Indeed, as discussed in section 2.1.2, the separation masks produced by the GWRP model mostly consists of binary values - either 0 or 1, but do not learn the magnitude *ratio* between the sources, which is the information required to correctly separate overlapping events. However, this is not an inconvenient when the events do not overlap.

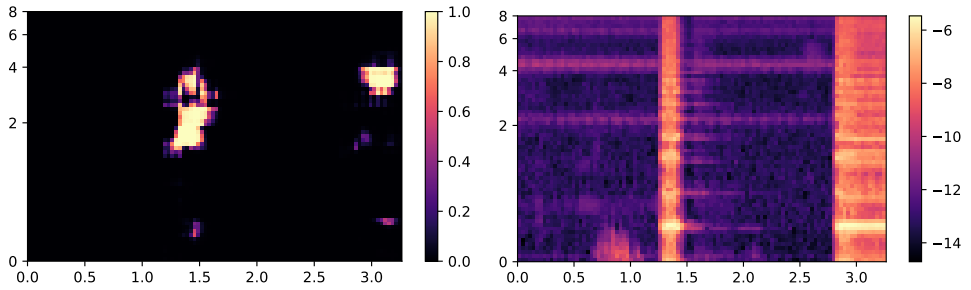
Listening to the separated speech by the GWRP model, we find that the key strokes noise is better suppressed than noise reduction operated by the RNNoise model. However, the separation also creates many distortions in the speech, while RNNoise has a much lighter effect. This effect is only indirectly measured in the PESQ scores, but is important for speech intelligibility.



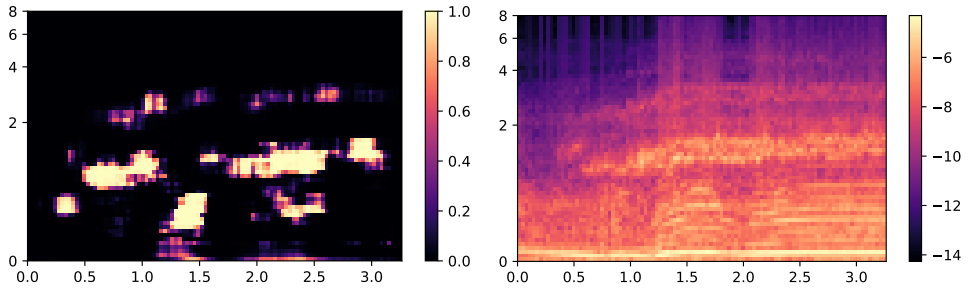
(a) Fire Alarm



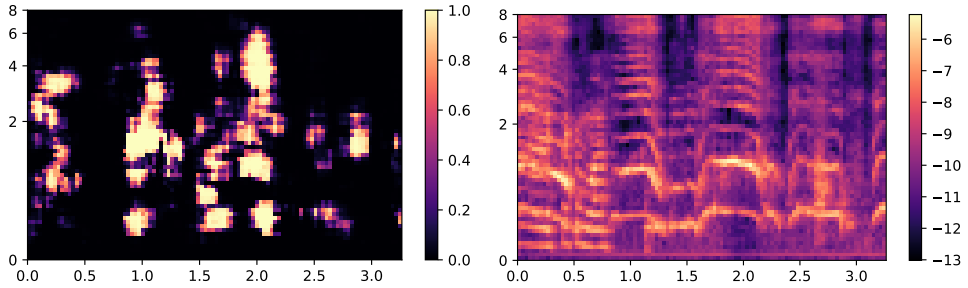
(b) Smoke Alarm



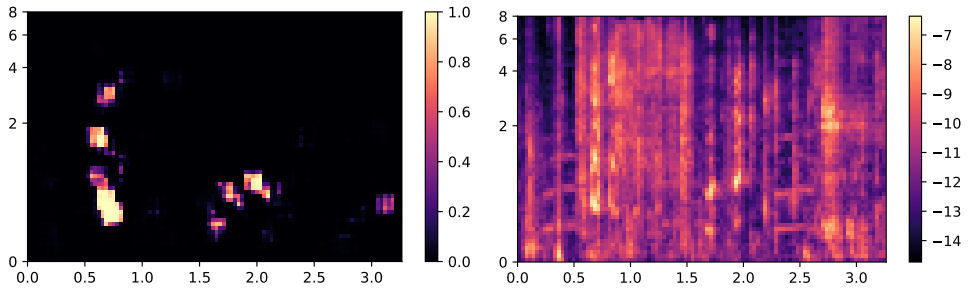
(c) Honk



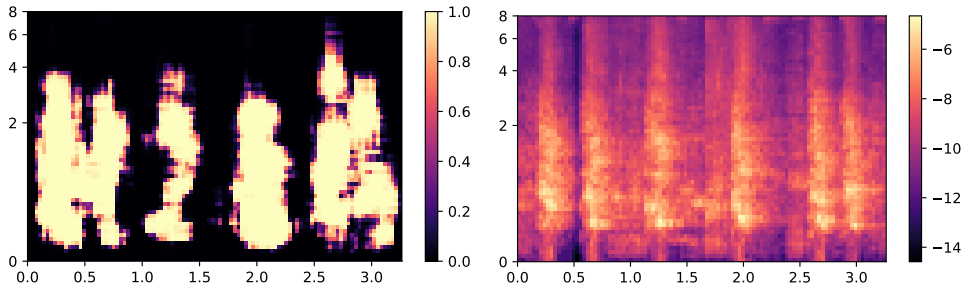
(d) Tire screeching



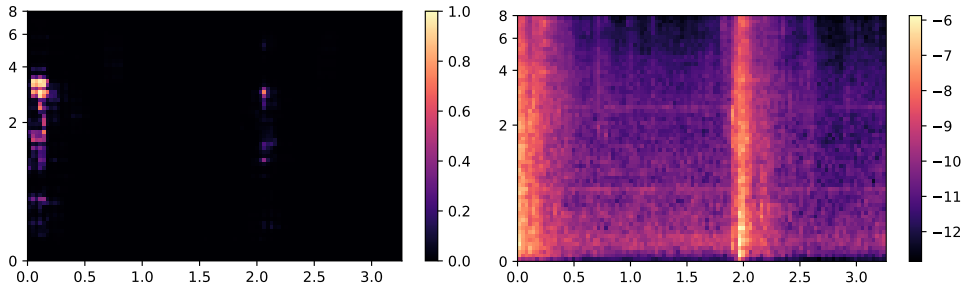
(e) Baby cry



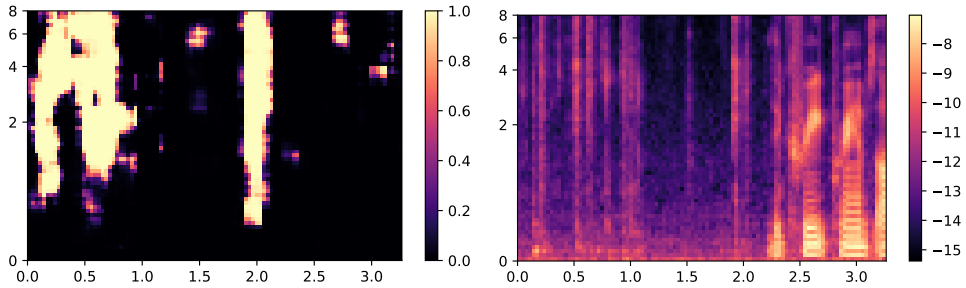
(f) Baby laughter



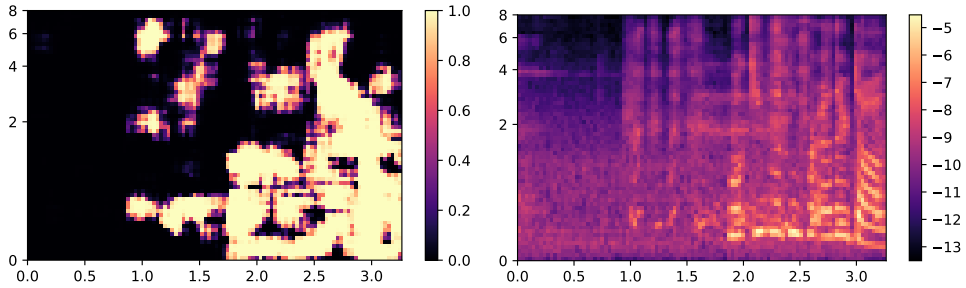
(g) Dog Bark



(h) Glass break

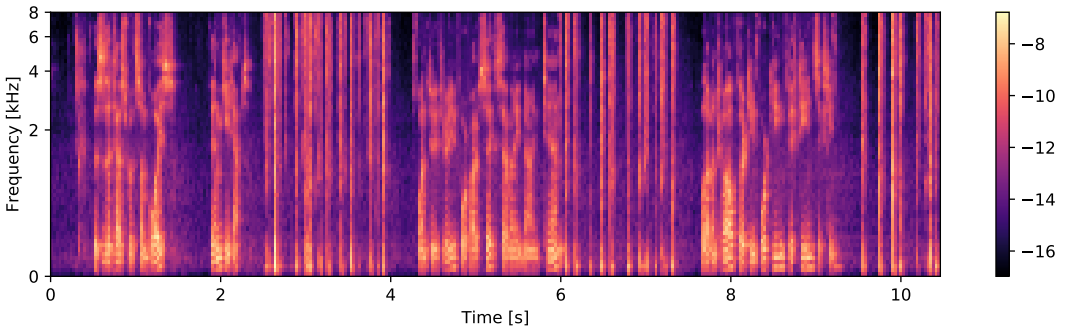


(i) Keyboard

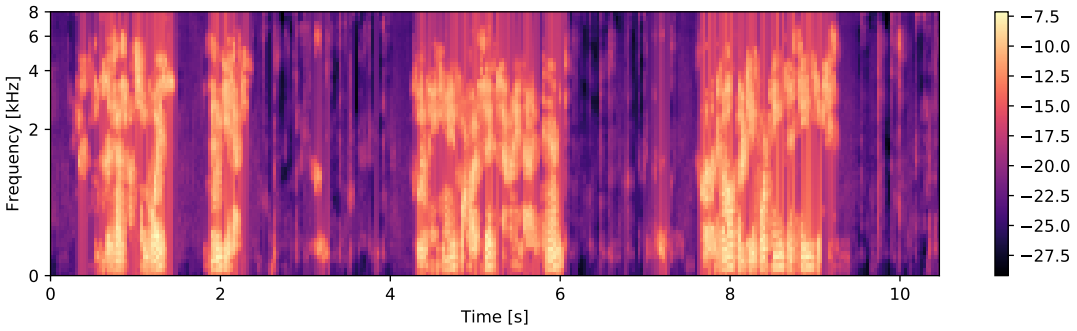


(j) Speech

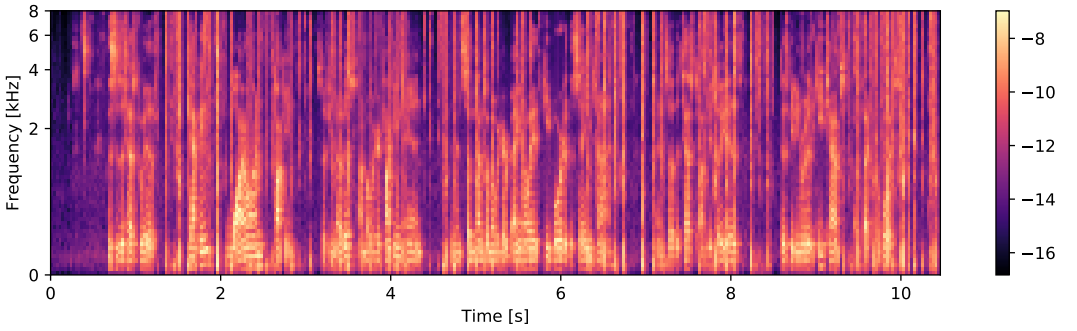
FIGURE 2.5: Example of a separation masks. The right column is the log Mel spectrogram of a audio segment containing the audio event indicated in caption. The left column displays the separation masks associated to the class present in the event. The horizontal axis is time in second and vertical axis is frequency in kilo-Hertz



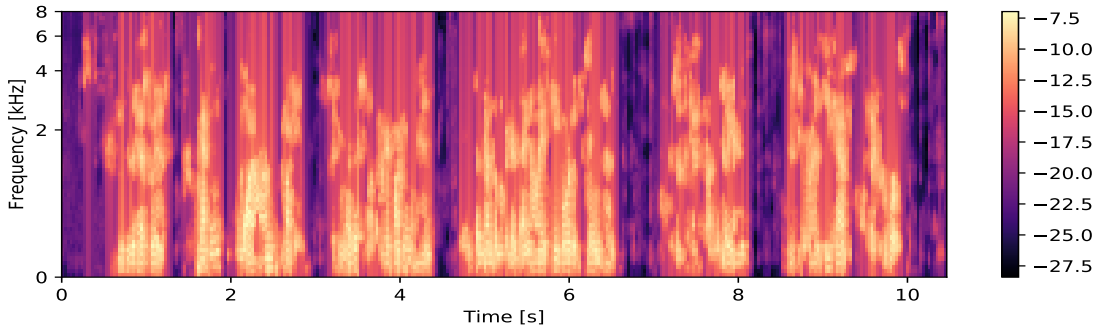
(a) Spectrogram of an audio recording containing non-overlapping segments of male voice and keyboard noise.



(b) Spectrogram of the separated speech from the audio recording corresponding to figure 2.6a



(c) Spectrogram of an audio recording containing voice and keyboard noise overlapping in time.



(d) Spectrogram of the separated speech from the audio recording corresponding to figure 2.6c

FIGURE 2.6

3 Per-Channel Energy Normalization

The log Mel spectrogram is a widely used time-frequency representation when working with machine learning techniques on audio data. The reason for using the Mel scaling is to transform the linear frequency scale of a spectrogram to the Mel scale which is designed to replicate the human perception of frequencies [2]. The log transform is used for another reason: to reduce the dynamic range of the spectrogram values to facilitate its use in a gradient based optimization method. It is however not the only available transform for performing dynamic range compression, other mapping like power transforms can achieve the same goal (cf figure 3.1). In this section, we investigate the use of the per-channel energy normalization (PCEN) processing method as a front-end for the models trained on the subset of AudioSet described in section 2.2. The PCEN combines a power transform with a filtering processing removing stationary noise from the Mel spectrograms. It is therefore able to highlight audio events in the spectrogram. As seen in section 2.1, the masks produced by the GWRP models capture the general area of the events, but not their precise structure. We hypothesize that using the PCEN front-end to highlight the events in the model input can help the model learn more precise masks.

3.1 Definition

The PCEN transform has first been proposed in [30]. Let E be a Mel spectrogram and $E(t, f)$ the spectrogram value at the time step t and frequency bin f. The PCEN transform is defined by (3.1).

$$PCEN(t, f) = \left(\frac{E(t, f)}{(\epsilon + M_s(t_n, f))^{\alpha}} + \delta \right)^{r} - \delta^{r} \tag{3.1}$$

where δ, α, r and ϵ are parameters of the transformation and $M_s(t, ft)$ is a smoothed version of $E(t, f)$ obtained using a first order infinite response filter:

$$M(t, f) = (1 - s)M(t - 1, f) + sE(t, f). \tag{3.2}$$

[30] proposes the following default values: $r = 0.5$, $\delta = 2$, $s = 0.057$, $\alpha = 0.98$ and $\epsilon = 10^{-6}$.

The PCEN processing combines two signal processing elements in one operation:

- Dynamic range compression. The parameters r and δ are the parameters of a shifted power transform accomplishing dynamic range compression.
- Stationary and quasi-stationary noise filtering. The smoother M is a low pass filtered version of E. The normalization of E by M removes the stationary and slowly varying component from E. The parameter s parametrizes the cut-off frequency of the filtering and the parameter α controls the amount of noise

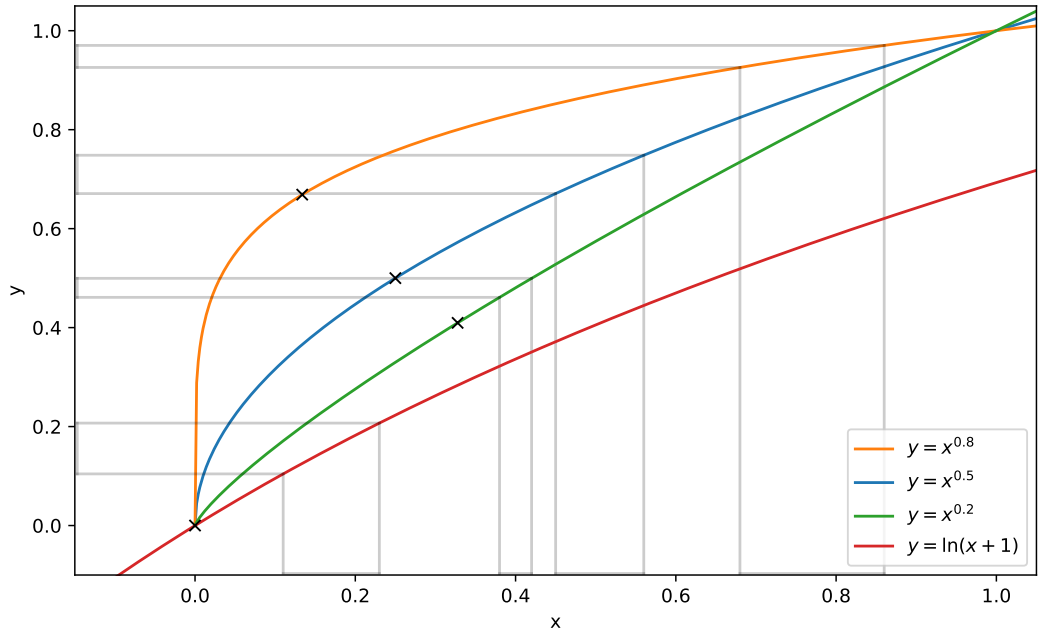


FIGURE 3.1: Examples of dynamic range compression through power functions and log function. The black crosses mark the point above which the transformation function are contraction mappings.

removal to perform: $\alpha = 1$ leads to perfect stationary signal cancellation, while a smallest value allows a portion of the stationary signal to be preserved. ϵ is a parameter used to control the numerical stability of the cancellation.

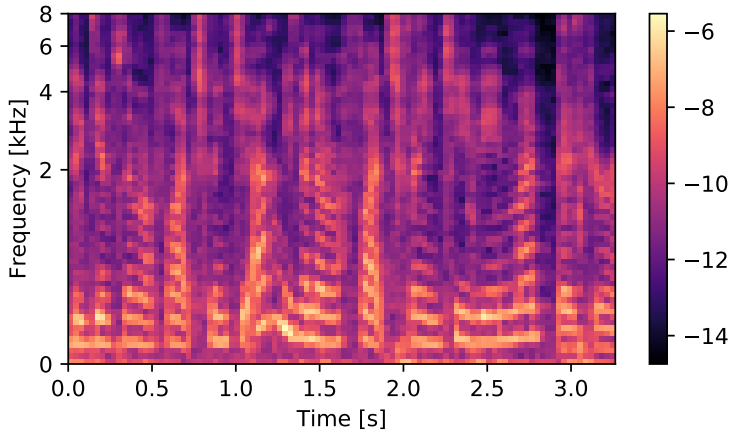
A study of the PCEN parameters signification and interpretation has been conducted in [15].

3.2 Stationary noise filtering

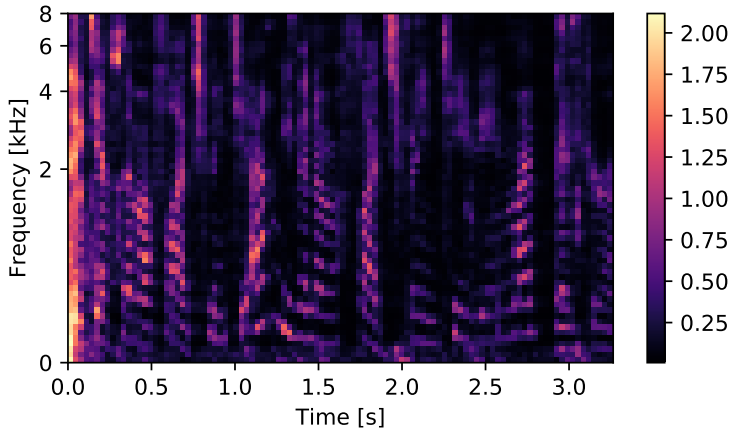
The PCEN ability to suppress stationary noise from the input spectrogram is illustrated on figure 3.2. Figure 3.2a displays the log Mel spectrogram of an audio recording ; the PCEN processing of the same audio, using the parameters values suggested in [30] is displayed in figure 3.2b, shows that the background noise has been removed.
The initialization of M should be taken into consideration: an artifact can be seen for the first time steps of the PCEN output in figure 3.2b which is due to an initialization to 0. In order to avoid such artifacts, the filtering operation can be applied twice: once forward, and then backward in time. The total filtering operation is then ensured to have a null phase. Forward backward filtering has been used in place of the forward only definition in (3.2) for the rest of the thesis.

3.3 A trainable front-end

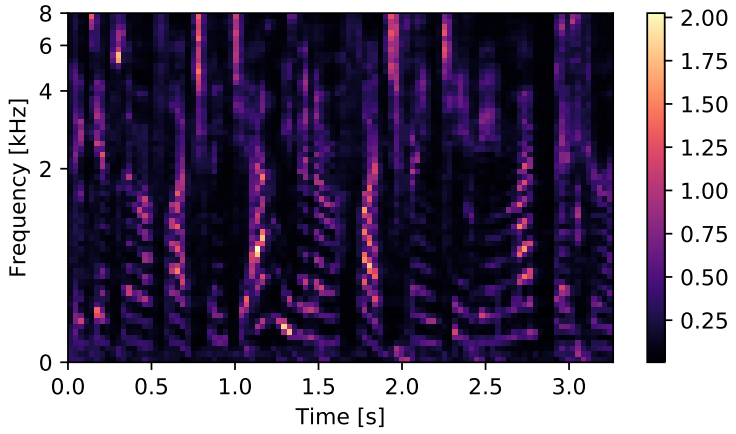
The operation of the PCEN processing are derivable with respect to the parameters r, δ and α, therefore the optimization of these parameters can be conducted jointly as the training of a the audio source separation models. Moreover, the value of these parameters can be generalized and be considered frequency dependent:



(a) Log Mel spectrogram



(b) PCEN processing with default parameters from [30]. The filter used for the computation of M is zero initialized, creating an artifact on the first time steps.



(c) PCEN processing with the default parameters from [30], using forward-backward filtering for the computation of M.

FIGURE 3.2: Illustration of the stationary noise canceling property of the PCEN processing.

$r = r(f), \delta = \delta(f), \alpha = \alpha(f)$.
The filtering operation can also be trained, either by directly training the s parameter, or by following the method proposed in [30]: instead of using a unique smoother M_s (whose s parameter could be trained), one can chose to consider a fixed set of smoother $\{M_{s_k}\}_{k=1,\dots,K}$ (where K is the number of smoothers) and learn a combination of those smoother outputs:

$$M(f,t) = \sum_{k=1}^{K} w_k(f) M_{s_k}(f,t) \tag{3.3}$$

$$\text{s.t. } w_k(f) \geq 0 \text{ and } \sum_{k=1}^{K} w_k(f) = 1 \tag{3.4}$$

The same model than in section 2.2 has been trained, with the difference that the log transform applied to the mask model is replaced by a trainable PCEN processing. In order to investigate the behavior of the PCEN processing with respect to noise, Gaussian noise is mixed at 5 dB signal to noise ratio (SNR) to each element of the data set, and another model is trained on this noisy data.

The learned values for the parameters r, δ and α are displayed in figure 3.3. r and δ parametrize the dynamic range compression applied to the Mel spectrograms. The range of the values of a Mel spectrogram depend on the energy contained in the audio samples. Unless there is an imbalance in the energy distribution with respect to frequency in the data set, these parameters take roughly the same values for all frequency bins which is the observed behavior. The value of r is around 0.5 which is the recommended value for indoor application by [15]. The value of α is also stable with respect to frequency and at 0.8 it is much lower than the recommended value of 0.98 for indoor application.
Figure 3.4a displays the weights w_k learned by model trained on the clean data set, and figure 3.4b the weights learned during training on the noisy data set. The model trained on the noisy data set does not learn to favor a specific smoother. This is to be expected since, with such a high level of noise, the background stationary noise is completely hidden by the added Gaussian white noise which is frequency independent and thus can not be removed by the smoothers. However, the model trained on the clean data set clearly gives a higher weight to the smoother corresponding to the higher s value, that is the smoother that lets the most signal unfiltered. Moreover, the value of α being small, the effect of the smoother is even more attenuated.

The model trained on the clean data set has learned parameter values aiming at minimizing the effect of the stationary noise removal of the PCEN. The hypothesis that the PCEN filtering operation would help the model better focus on the audio events is invalidated: on the contrary it appears that the filtering is hiding useful information from the model and that is effect has to be minimized. The model is trained for separating events which have very different time properties: key strokes or glass breaks are for instance very short and sudden events, while tire screeching or alarms are more stationary events. Learning a single filtering operation can highlight some events while penalizing others, hence having a negative impact overall.

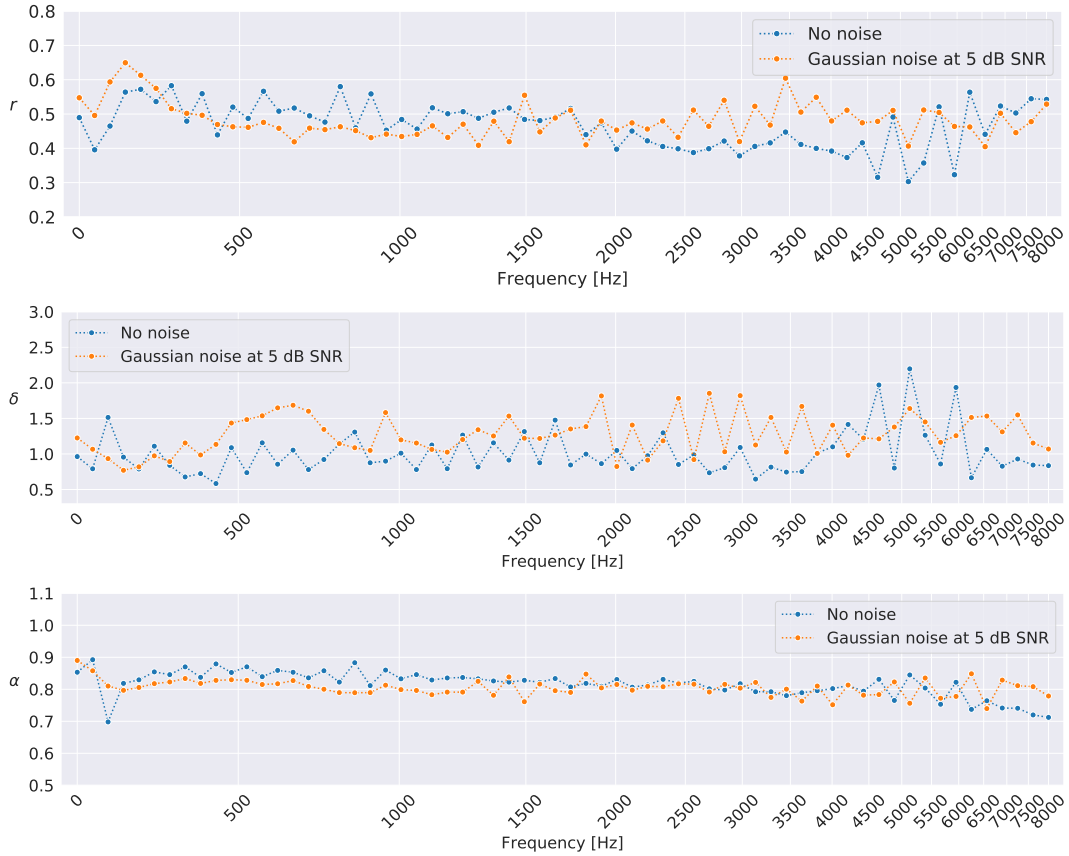
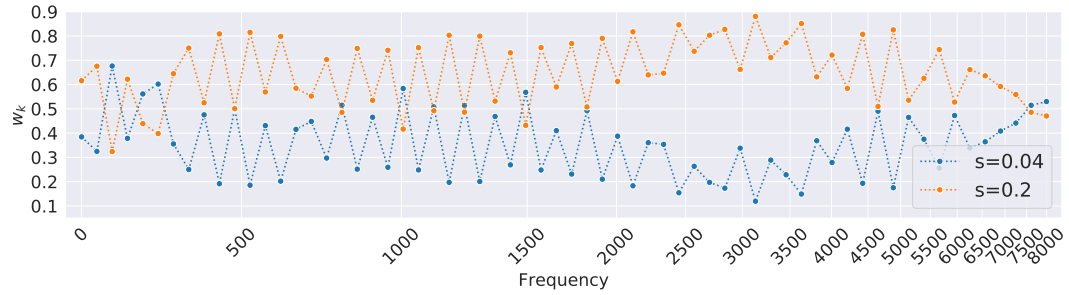
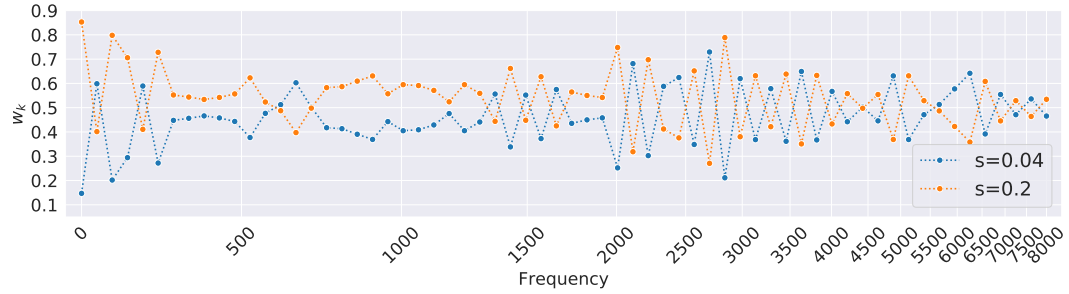


FIGURE 3.3: Learned PCEN parameters with respect to frequency.



(a) Distribution of the w_k parameters with respect to frequency for the model trained on the clean data set.



(b) Distribution of the w_k parameters with respect to frequency for the model trained on the noisy data set (Gaussian noise mixed to each audio sample at 5 dB signal to noise ratio).

FIGURE 3.4

3.4 Multi-PCEN training

Learning a single PCEN transform is not profitable for the audio separation model because, since the model must separate classes that have very different time-frequency signature, one of the classes to separate is likely to be penalized by the PCEN filtering component. To circumvent this problem, we investigate the possibility of learning not one, but multiple PCEN transform at the same time. In order to to this, the input to the model is changed from been one log Mel spectrogram to a stack of PCEN processed spectrograms. This stack is obtained by passing the same Mel spectrogram to a set of trainable PCEN layers ; their outputs can then be regrouped and used as input to the mask model. Having several transformation to tune, the network is expected to learn to maximize its input information by learning PCEN transforms that put emphasis on different kind of audio sources.

The experiment is conducted with a set of 4 PCEN transform. The training curves of the PCEN parameters are reported in figure 3.5. The PCEN layers appear to specialize in the processing of a specific audio sources:
The green PCEN learns a very high value for the parameter s, which translates to a M value very close to E. The corresponding value of α is away from 1, so the division E/M in equation (3.1) is not trivial. The value of r is above 0.5 and the value of δ is bellow 1.0 which correspond to a minimal use of the dynamic range compression mechanism. This PCEN tries to enhance quasi-stationary (high s) events with relatively small energy (high r and low δ).
The red PCEN removes stationary noise (low s and high α). The dynamic range compression is small (low value of r and high value of δ) which favors high values to stand out in the result. This PCEN parameters favors the detection of high energy and sudden events.
The blue PCEN uses an important dynamic range compression (high r and small δ) which brings the low energy events of the input spectrogram closer to the high energy events. The smoother M has a low cut-off frequency (low s) which removes stationary noises, but its application is limited by a small value of α. This PCEN favors low energy and sudden events such as glass breaks.

To compare the results of this multi-PCEN approach to the log transform, the PESQ MOS-LQO score of speech separation is computed on the same recordings than in section 2.2 and reported in table 3.1. The multi-pcen performs worse on the recording where the speech and keyboard are non-overlapping, and similarly on the case where they overlap. Listening to the audio samples, we discover that the speech separated by the model using the multi-PCEN front-end is very distorted: the slowly varying elements of the speech signal are attenuated. It appears that the filtering operation of the learned PCEN transform removes part of the speech information before it is fed to the separation model, which results in poorer separation performances.

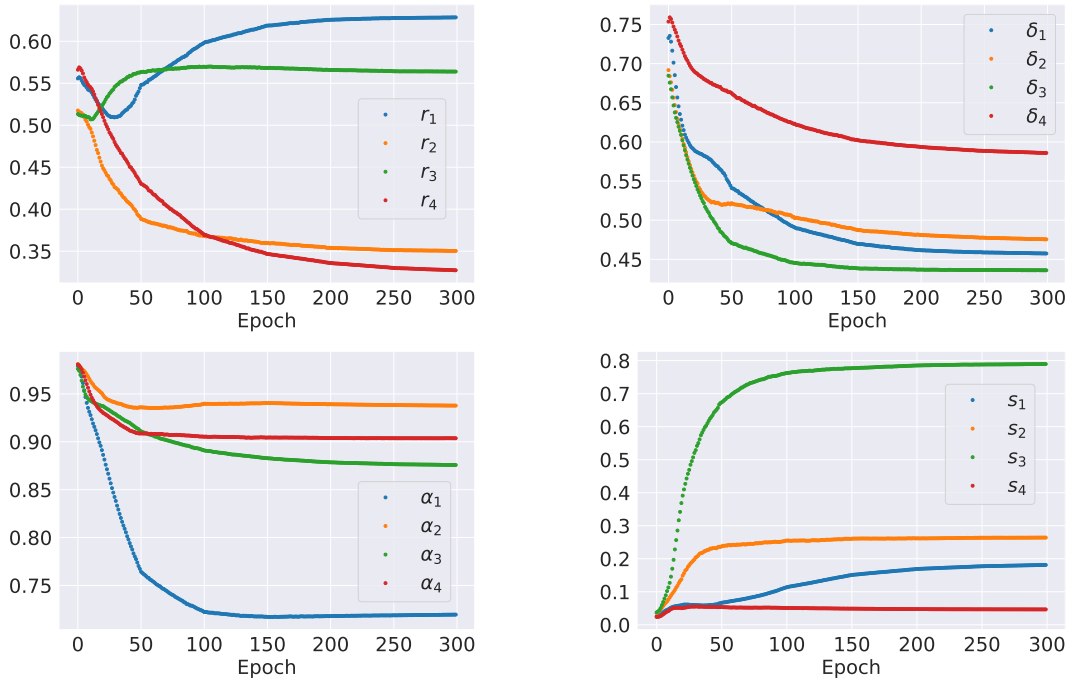


FIGURE 3.5: PCEN parameters values during training. A color identify a PCEN transform.

	Non-overlapping mixture	Overlapping mixture
log transform	1.9	1.2
4 PCEN transform	1.2	1.2

TABLE 3.1: PESQ MOS-LQO scores of GWRP model trained on the AudioSet subset, calculated on the separated speech extracted from the audio files corresponding to the log Mel spectrograms of figure 2.6

Conclusion

An audio separation framework has been implemented for training models from weakly labeled data and generate separated audio sources from mixture using separation masks. It is able to train this method using weakly labeled data because the model doing the separation (the mask model) is jointly trained with a classifier model (the classification model).

Several models architecture have been investigated for the classification model which lead to a better understanding of the training algorithm under-lying assumptions. Performances measurements on the TUT Rare Sound Events 2017 data set show that the best architecture is the global weighted rank pooling proposed in [12, 13]. The global pooling classifier have the advantage of aggregating the values of the separation masks using positive-only weights, while the other considered approaches could use negative weights which was interfering with the mask model learning. Classification mapping that have the same property while leveraging the 2 dimensional information of the mask models could improve the method performances.

The best performing model architecture has been trained on a portion of Audioset, demonstrating that the method is able to capture relevant information from a complex data set. The separation method is then tested on the problem of speech enhancement and achieves similar speech enhancement performances than the open source RNNoise model [27]. However it introduces many distortions in the separated speech compared to the processing operated by RNNoise. The method struggles much more to separate source signals when they overlap in time and frequency.

While the method is effective at separating non-overlapping events, it also induces many distortions. The effect on speech intelligibility should be measured more precisely to assess the method performances at speech enhancement. In comparison, RNNoise model introduces much less distortion but does not suppress the interfering events as well.

A regularization term has been proposed in order to include into the training objective the fact that the masks of absent sources should take all-zero values. The use of the softmax function for predicting the separation masks has been shown to improve the separation performances of the model.

A recent trainable front end for audio processing, the per-channel energy normalization technique, has been investigated in the topic of audio separation. The filtering component of the front-end removes speech information from the spectrogram, which results in poorer separation performance for the model compared to log Mel spectrogram input.

The presented method work on mono-channel audio recordings. However, most audio separation framework algorithms use information from multiple channels for the separation. An extension of the method to leverage such information is expected to improve the performances.

Bibliography

[1] J. S. Calderón-Piedras, Á. D. Orjuela-Cañón, and D. A. Sanabria-Quiroga. "Blind source separation from single channel audio recording using ICA algorithms". In: *2014 XIX Symposium on Image, Signal Processing and Artificial Vision*. Sept. 2014, pp. 1–5. DOI: 10.1109/STSIVA.2014.7010168.

[2] M. Cernak, A. Asaei, and A. Hyafil. "Cognitive Speech Coding: Examining the Impact of Cognitive Speech Processing on Speech Compression". In: *IEEE Signal Processing Magazine* 35.3 (May 2018), pp. 97–109. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2761895.

[3] A. Chaudhari and S. B. Dhonde. "A review on speech enhancement techniques". In: *2015 International Conference on Pervasive Computing (ICPC)*. Jan. 2015, pp. 1–3. DOI: 10.1109/PERVASIVE.2015.7087096.

[4] Zhuo Chen, Yi Luo, and Nima Mesgarani. "Deep attractor network for single-microphone speaker separation". In: *CoRR* abs/1611.08930 (2016). arXiv: 1611.08930. URL: http://arxiv.org/abs/1611.08930.

[5] J. F. Gemmeke et al. "Audio Set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2017, pp. 776–780. DOI: 10.1109/ICASSP.2017.7952261.

[6] Emad M. Grais and Mark D. Plumbley. "Single Channel Audio Source Separation using Convolutional Denoising Autoencoders". In: *CoRR* abs/1703.08019 (2017). arXiv: 1703.08019. URL: http://arxiv.org/abs/1703.08019.

[7] John R. Hershey et al. "Deep clustering: Discriminative embeddings for segmentation and separation". In: *CoRR* abs/1508.04306 (2015). arXiv: 1508.04306. URL: http://arxiv.org/abs/1508.04306.

[8] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[9] A. Hyvärinen and E. Oja. "Independent component analysis: algorithms and applications". In: *Neural Networks* 13.4 (2000), pp. 411–430. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(00)00026-5. URL: http://www.sciencedirect.com/science/article/pii/S0893608000000265.

[10] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 448–456. URL: http://dl.acm.org/citation.cfm?id=3045118.3045167.

[11] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: http://www.scipy.org/.

[12] Qiuqiang Kong et al. "A joint separation-classification model for sound event detection of weakly labelled data". In: *CoRR* abs/1711.03037 (2017). arXiv: 1711.03037. URL: http://arxiv.org/abs/1711.03037.

[13] Qiuqiang Kong et al. "Sound Event Detection and Time-Frequency Segmentation from Weakly Labelled Data". In: *CoRR* abs/1804.04715 (2018). arXiv: 1804.04715. URL: http://arxiv.org/abs/1804.04715.

[14] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[15] V. Lostanlen et al. "Per-Channel Energy Normalization: Why and How". In: *IEEE Signal Processing Letters* 26.1 (Jan. 2019), pp. 39–43. ISSN: 1070-9908. DOI: 10.1109/LSP.2018.2878620.

[16] Yi Luo et al. "Deep clustering and conventional networks for music separation: Stronger together". eng. In: vol. 2017. IEEE, 2017, pp. 61–65. ISBN: 9781509041176.

[17] Andrew L. Maas. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: 2013.

[18] Brian McFee et al. *librosa/librosa: 0.6.2*. Aug. 2018. DOI: 10.5281/zenodo.1342708. URL: https://doi.org/10.5281/zenodo.1342708.

[19] A. Mesaros, T. Heittola, and T. Virtanen. "TUT database for acoustic scene classification and sound event detection". In: *2016 24th European Signal Processing Conference (EUSIPCO)*. Aug. 2016, pp. 1128–1132. DOI: 10.1109/EUSIPCO.2016.7760424.

[20] A. Mesaros et al. "DCASE 2017 Challenge Setup: Tasks, Datasets and Baseline System". In: *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Nov. 2017, pp. 85–92.

[21] Gautham J. Mysore, Paris Smaragdis, and Bhiksha Raj. "Non-negative Hidden Markov Modeling of Audio with Application to Source Separation". In: *Latent Variable Analysis and Signal Separation*. Ed. by Vincent Vigneron et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 140–148. ISBN: 978-3-642-15995-4.

[22] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS-W*. 2017.

[23] SNijmegen: Max Planck Institute for Psycholinguistics. *ELAN*. https://tla.mpi.nl/tools/tla-tools/elan/. Version 5.2.

[24] Colin Raffel et al. "MIR EVAL: A Transparent Implementation of Common MIR Metrics". In: *ISMIR*. 2014.

[25] A. W. Rix et al. "Perceptual Evaluation of Speech Quality (PESQ)-a New Method for Speech Quality Assessment of Telephone Networks and Codecs". In: *Proceedings of the Acoustics, Speech, and Signal Processing, 200. On IEEE International Conference - Volume 02*. ICASSP '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 749–752. ISBN: 0-7803-7041-4. DOI: 10.1109/ICASSP.2001.941023. URL: http://dx.doi.org/10.1109/ICASSP.2001.941023.

[26] Mikkel N. Schmidt and Rasmus Kongsgaard Olsson. "Single-channel speech separation using sparse non-negative matrix factorization". In: *INTERSPEECH*. 2006.

[27] Jean-Marc Valin. "A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement". In: *CoRR* abs/1709.08243 (2017). arXiv: 1709.08243. URL: http://arxiv.org/abs/1709.08243.

[28] E. Vincent, R. Gribonval, and C. Fevotte. "Performance measurement in blind audio source separation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4 (July 2006), pp. 1462–1469. ISSN: 1558-7916. DOI: 10.1109/TSA.2005.858005.

[29] Y. Wang, A. Narayanan, and D. Wang. "On Training Targets for Supervised Speech Separation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.12 (Dec. 2014), pp. 1849–1858. ISSN: 2329-9290. DOI: 10.1109/TASLP.2014.2352935.

[30] Yuxuan Wang et al. "Trainable Frontend For Robust and Far-Field Keyword Spotting". In: (2016).

[31] Shinji Watanabe et al. *New Era for Robust Speech Recognition : Exploiting Deep Learning*. eng. Cham Switzerland: Springer, 2017. ISBN: 978-3-319-64679-4.