End-to-End Mono to Binaural Conversion with Conv-TasNet

Pau Márquez Julbe
pau.marquez.julbe@estudiantat.upc.edu

José Adrián Rodríguez Fonollosa
jose.fonollosa@upc.edu

Abstract

Binaural audio creates a 3-D stereo sound sensation that makes the listener feel like actually being in the room where the sounds are recorded. However, this kind of recordings are expensive and require expertise. A data-driven approach to mono-to-binaural conversion with the help of visual information has already been studied[4], but we face the problem with a different method.

Our main idea is to combine a source separation neural network with visual information since the mono-to-binaural problem can be reformulated as a source separation problem, where each source is associated to a given object in the image. With this information, the neural network should be able to distribute each sound proportionally to each ear and get the desired binaural audio.

As end-to-end deep learning approaches have shown to surpass frequency space based models in audio processing, the former ones are elected to tackle this problem. In order to do that we are going to use Conv-TasNet, a Temporal Convolutional Network based architecture for speech separation.

1. Introduction

As human beings, we perceive the world as a combination of simultaneous sensory stimulus and we are able to automatically combine all of them to have a better understanding of what surrounds us. More concretely, the auditory system is able to capture spatial information by itself. As Lord Rayleigh says in the duplex theory, humans are able to localise sound sources with the difference in time between the different sounds that reach the ears (interaural time differences, ITDs) and the difference in the amplitude between the sounds (interaural level differences, ILDs). However, this spatial information cannot be captured trivially by devices. Ear shaping, separation and orientation has to be cloned as shown in Fig. 1. This kind of microphone gives the user a 3D sound sensation that takes the listener to a much more realistic feeling. This is a valuable feature for audiophiles, AR/VR applications and many more applications.

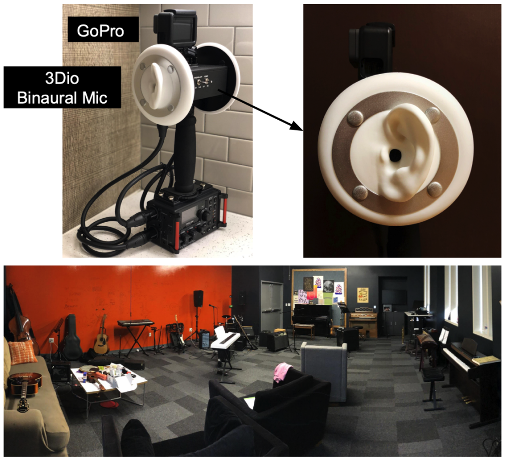

Figure 1. Binaural microphone used to collect the FAIR-Play dataset. Reprinted from [4]

In the recent years, end-to-end neural networks have been a successful approach on deep learning algorithms applied to audio problems, as compared to spectrogram based approaches, which tended to be the most used to solve this kind of problems. Some of the main reasons why end-to-end approaches have had the edge over spectrogram based ones are the following: the computation of the spectrogram has an overhead in terms of latency, as the time window of the short time fourier transform (STFT) needs to be long enough to get a high frequency resolution (and even longer when working with music clips). Furthermore, an upper bound on the accuracy is created as the reconstruction of the phase is not a trivial task that, even using phase reconstruction methods, it is not easy to reconstruct. Some end-to-end approaches are [3] and [13], which replace the STFT with a data-driven feature extraction module.

In this paper we are approaching end-to-end mono to binaural conversion with Conv-TasNet [11] as the base network. This problem has already been studied by [4], our benchmark and starting point.

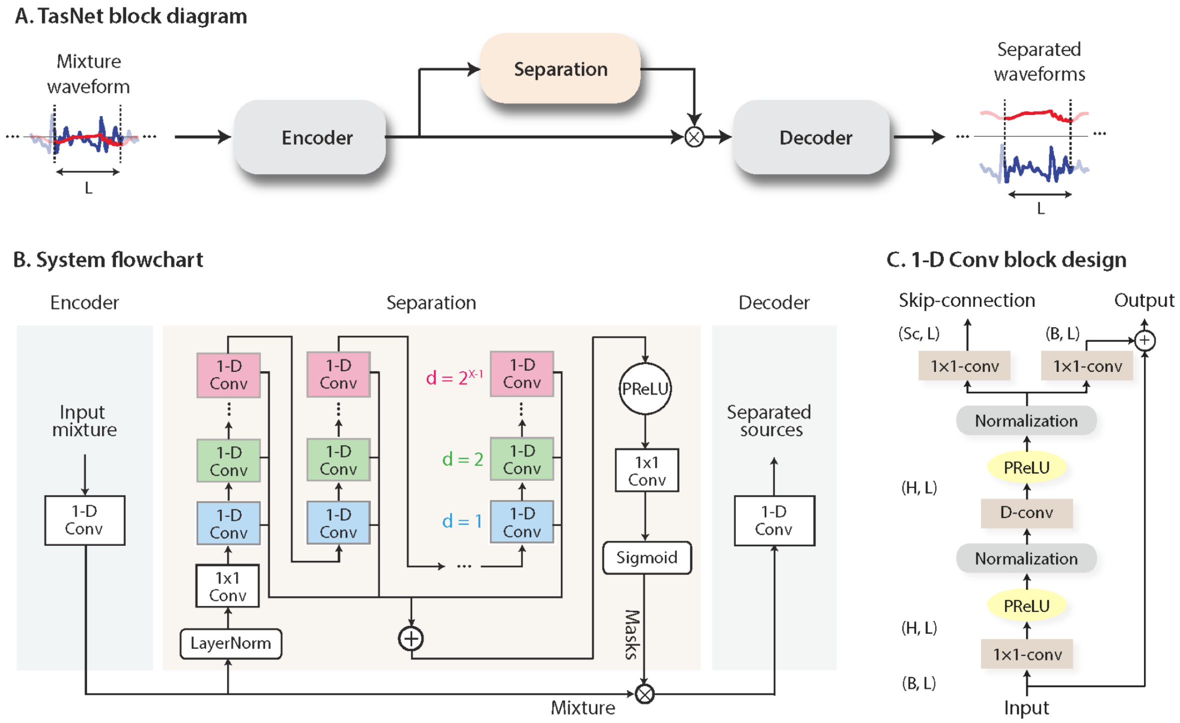

Figure 2. Conv-TasNet's architecture. Reprinted from [11]

2. Architectures

Our approach to convert mono audio to binaural with visual information uses the Conv-Tasnet [11] architecture, an end-to-end neural network that reached state-of-the-art results in speech separation. As a reference we have taken the Mono2Binaural [4] network, which, through a convolutional encoder-decoder architecture, computes a spectrogram mask.

2.1. Mono2Binaural

It is a U-net style architecture that includes five convolutional layers and five transposed convolution layers to encode and decode, respectively. The visual features of a given frame are extracted using a ResNet-18 [5] and concatenated to the audio encoding through the channel's dimension.

The input of the network is a complex-valued spectrogram from a 0.63 seconds clip of audio and a frame in the middle of those 0.63 seconds. The prediction is a complex mask of the short-time Fourier transform of the difference between the left and the right channels. With this, the inverse short-time Fourier transform (iSTFT) is applied to the masked input and the difference signal is recovered.

The loss to be optimized in is the euclidean distance between the complex spectrogram of the ground truth's difference signal and the complex spectrogram predicted by the neural network.

2.2. Conv-TasNet

Conv-TasNet is the network in which we have based our studies on to approach the end-to-end mono to binaural problem. It consists of an encoder, a separator and a decoder module. An overview of the network's flow is the following:

1. **Encoder:** The encoder module outputs N channels through a 1D convolution with a stride half the size of the kernel size L. Thus, the temporal information is kept in the embedding through the dimensions.

2. **Separator:** The separation module takes the encoded audios (of N channels) and, through a 1×1 convolutional block, it adapts the number of channels to B. Then the output is given to a temporal convolutional network (TCN) [1] [8] [9] based module that, in summary, is X dilated 1D convolutional blocks. These TCN are applied R times sequentially.

   After that, a 1×1 convolutional block adapts the number of channels to have a shape of $C \times N$, where C is the number of sources we want to separate. Finally, a sigmoid function is applied to get the C masks. Those are applied to the encoded audio and the result is given to the decoder.

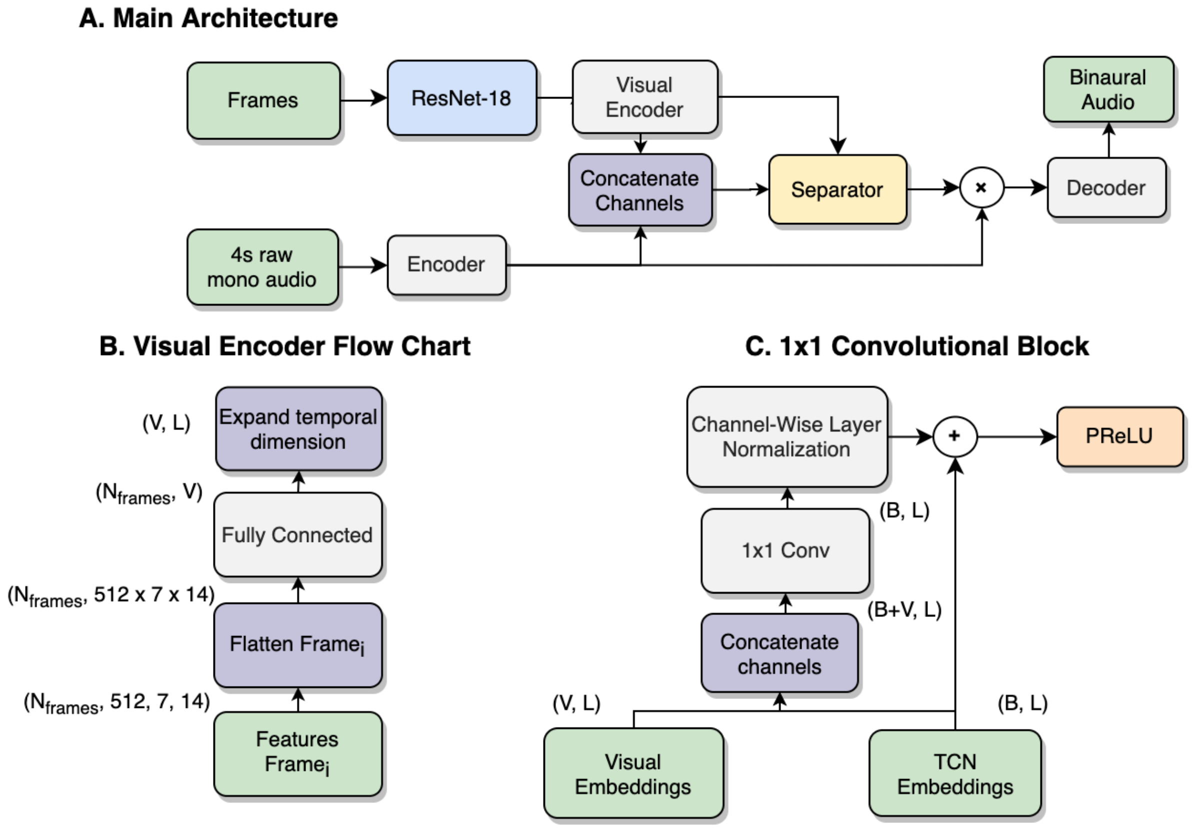

Figure 3. (A) Encode both visual and audio information and give it to the Separation module. Visual encodings are also given to the separation module separately in order to apply the 1×1 Convolutional block after every TCN. Finally the mask is applied and the decoder outputs the binaural audio. (B) Turns the features from ResNet into temporal embeddings taking into account the frame's index. V is the number of channels that the visual encodings have and L is the length of the audio encoding. (C) Is the skip connection module that is inserted after every TCN module except the last one. Visual embeddings and TCN's output are concatenated through channels and a 1×1 convolutional layer is applied to go back to B channels. This convolution pads the input in such a way that the output's last dimension is the same as the input's dimension. After that, a channel-wise layer normalization is applied and the output is summed to the input TCN embeddings.

3. **Decoder:** For each output audio C, the decoder applies a 1×1 convolution to convert the encoding's N channels into $L \cdot ac$ (ac is the output's number of channels, 1 for mono and 2 for stereo) channels. Then through a reshape and a transposition, the shape turns into (ac, K, L) where K is the last encoding's dimension. Through an overlap-add method, the audio's temporal dimension is recovered resulting into the final sources.

For a clearer understanding see the figure 2 and table 2.2.

2.3. Adapting Conv-Tasnet to video

As Conv-TasNet is used for speech separation, it does not have spatial information about the source of sounds that, combined, turn into the final audio.

In order to introduce such information, feature extraction has been applied with the ResNet-18 [5] neural network. Concretely, only the convolutional layers are kept. This features are given to the network in different ways, as explained in the following section.

Symbol	Description
N	Number of filters in the autoencoder
L	Kernel size of the encoder 1×1 convolution
B	Number of channels given to TCNs
H	Number of channels inside the TCN
P	Kernel size of the TCN
X	Number of convolutional blocks in each TCN
R	Number of repeats of the TCN
C	Number of speakers
V	Number of channels of the encoded frames

Table 1. Conv-TasNet network's hyperparameters.

2.3.1 Main approach

As the separation module is applied after encoding the audio, our approach has been to create another encoder module that encodes the visual features into a tensor that has the same temporal dimension as the encoded audio.

To do so, the ResNet features from the convolutional layer are flattened for each frame and a fully connected layer encodes those features into *V* dimensions. Then, the frame's index (temporal information) and the embedding dimension are transposed. With this, the last dimension is expanded to have the same shape as the encoded audio.

After that, both encoded tensors have been concatenated through the channel's dimension and given to the separation module increasing the N parameter inside this module. See figure 3.

Furthermore, after every repetition of the TCN except the last one, its output is given to the 1×1 Convolutional block described in figure 3. The output of this block is given back to the next TCN.

This kind of neural network modules respect the temporal information of the embeddings, as applying the 1×1 Convolution is the same as applying the same fully connected layer to every time-step of the embedding. Specifically,

$$Y = W \times \begin{bmatrix} X_{TCN} \\ X_{visual} \end{bmatrix} \tag{1}$$

where $W \in \mathbb{R}^{B \times B+V}$, $X_{TCN} \in \mathbb{R}^{B \times L}$ and $X_{visual} \in \mathbb{R}^{V \times L}$. Remind that *L* is the encoding (temporal) dimension.

3. Experiments

3.1. Dataset

Our network has been trained and tested on the FAIR-Play [4] dataset. It contains 1871 10s clips of video with the corresponding binaural audios which were collected specifically to solve this task.

3.2. Metrics

Five different metrics have been used:

1. **STFT Distance:** The euclidean distance between the ground-truth and the predicted complex spectrograms of the left and right channels:

$$\mathcal{D}_{\{STFT\}} = \|\mathbf{X}^L - \tilde{\mathbf{X}}^L\|_2 + \|\mathbf{X}^R - \tilde{\mathbf{X}}^R\|_2 \tag{2}$$

2. **Envelope Distance:** Following [12], we take the envelope distance as the euclidean distance of the envelope of each channel. Let $\phi(x(t))$ be the Hilbert transform of signal $x(t)$:

$$\mathcal{D}_{\{ENV\}} = \|\phi(x^L(t)) - \phi(\tilde{x}^L(t))\|_2 + \|\phi(x^R(t)) - \phi(\tilde{x}^R(t))\|_2 \tag{3}$$

N	L	B	H	P	X	R	V	Norm	C
512	16	256	512	3	8	3	512	gLN [11]	2

Table 2. Hyperparameters of the selected model.

3. **Reconstruction Distance:** The euclidean distance between the ground-truth and the predicted $x(t)$:

$$\mathcal{D}_{\{REC\}} = \|x^L(t) - \tilde{x}^L(t)\|_2 + \|x^R(t) - \tilde{x}^R(t)\|_2 \tag{4}$$

4. **Channel's Difference Distance:** The euclidean distance between the difference of the channels. This loss is based on the loss used in our benchmark, as explained in section 2.1:

$$\mathcal{D}_{\{DIFF\}} = \|(x^L(t) - x^R(t)) - (\tilde{x}^R(t) - \tilde{x}^L(t))\|_2 \tag{5}$$

5. **Scale Invariant Signal to Noise Ratio:** Metric commonly used for source separation replacing the source-to-distorsion ratio (SDR) [6] [10] [14]. It is defined as:

$$\begin{cases} s_{target} := \frac{\langle \tilde{s}, s \rangle s}{\|s\|^2} \\ e_{noise} := \tilde{s} - s_{target} \\ \textit{SI-SNR} := 10 log_{10} \frac{\|s_{target}\|^2}{\|e_{noise}\|^2} \end{cases} \tag{6}$$

where $\tilde{s} \in \mathbb{R}^{1 \times T}$ and $s \in \mathbb{R}^{1 \times T}$ are the estimated and clean sources. To ensure scale invariance, s and $\tilde{s}$ are normalized to have zero mean.

6. **Signal to Distorsion Ratio:** Commonly used metric for source separation. It is defined as:

$$\textit{SDR} := 10 log_{10} \frac{\|s\|^2}{\|\tilde{s} - s\|^2} \tag{7}$$

where $\tilde{s} \in \mathbb{R}^{1 \times T}$ and $s \in \mathbb{R}^{1 \times T}$ are the estimated and clean sources.

3.3. Implementation

We implemented the training configuration as close to the Conv-Tasnet's configuration as possible; the audio is processed in 4 second clips, an initial learning rate of $1e^{-3}$ and it is halved if the validation loss is not improved in 3 consecutive epochs. Adam [7] is used as the optimizer and a gradient clipping of maximum L2-norm of 5 is applied. It has been trained during 75 epochs with a batch size of 32.

	STFT	ENV	REC	DIFF	SI-SNR	SDR
Conv-TasNet (with image)	1.114	0.162	$7.424e^{-3}$	$24.72e^{-3}$	5.336	2.842
Conv-TasNet (without image)	1.273	0.171	$8.462e^{-3}$	$26.35e^{-3}$	5.024	2.556
Mono2Binaural (Benchmark)	0.879	0.135	$5.874e^{-3}$	$21.40e^{-3}$	6.40	3.49
Mono-Mono	1.195	0.156	$7.95e^{-3}$	$26.48e^{-3}$	5.544	2.845

Table 3. Quantitative results of the models

Conv-TasNet is trained on a sampling frequency of 8kH but music requires it to be greater. Thus, we have followed Mono2Binaural, which uses a 16kHz sampling rate and a normalization of the average power of the signal to 0.1.

As the objects move slow in the training data, frames from the video are given every 0.5 seconds, even though it should not be a problem for the network to have a greater frame frequency, as the frames (temporal) dimension is expanded to have the same temporal dimension as the audio's encoding (see 2.3.1) and the processing is done mostly temporal-wise.

The loss used is a combination of the reconstruction distance and the STFT distance:

$$L := D_{REC} + \alpha D_{STFT} \tag{8}$$

where α is $1e^{-6}$, selected based on the difference of the scales of the different metrics.

The hyperparameters are based on the study made by [2] on Conv-TasNet's application in music source separation. Figure 2 shows the selected hyperparameters.

The python code can be found on github[1]. It is based on Mono2Binaural[2] and Demucs[3].

Finally, we have focused our studies on the non-causal case scenario so this network could not be used in real time, even though Conv-TasNet has also shown very good results for speech separation in its causal implementation.

3.4. Results

In order to evaluate our network results, we have computed the metrics for both the benchmark network and a mono to mono model that just outputs the input's mono audio. To obtain the benchmark's results, we have used the weights[4] of their model trained in the same dataset split that we did. Concretely, the *split1* from the splits folder given by the FAIR-Play dataset.

Our network has been able to recover the audio signal without distortion. Concerning the sound source distribution into the stereo channels, it does capture spatial information and the sound gives a stereo sensation, but the separation of sounds is not accurate enough and sources from different locations are panned into the same ear.

Furthermore, our network is very conservative when assigning a sound to a channel, meaning that, for instance, if the binaural microphone pans the whole sound into the left, the network might pan it to the left but not as much as desired.

A possible reason for that could be that our model outputs the left and right channels, instead of the difference between those, like our benchmark did. However, some of the configurations that we trained used this kind of loss and the results were very similar.

Most of the networks tended to the same metrics. There might be different reasons for that but the most likely one is that the tested architecture did not have the capacity to learn the spatial information that turns into deciding the right panning. It could be because of a too large dimensionality reduction of the visual features or the way that the visual data was combined with the audio data.

However, as shown in the table 2, the visual information has helped the network to get much better results. Without the visual information, the network has even worse results than predicting just the mono input. It can be expected, as there is an implicit reconstruction error in the network and without visual information there is no way to know where the different sound sources should be located. That is the reason why the network basically learned to output the same mono audio that was given as an input.

4. Conclusions

The approach has not been successful in terms of mono to binaural conversion (the main goal of this study). It has shown that the way we combined the visual and the audio information with the Conv-TasNet was not powerful enough to pan the audio to the correct channel.

Nonetheless, it has been proved that visual temporal information can be given into the Conv-TasNet architecture and use it to learn temporally-dependent image information, as the metrics are considerably better with image than without image.

Aknowledgements: Thanks to José Adrián Rodríguez Fonollosa.

[1] https://github.com/paumarquez/mono2binaural-conv-tasnet

[2] https://github.com/facebookresearch/2.5D-Visual-Sound

[3] https://github.com/facebookresearch/demucs

[4] https://drive.google.com/drive/folders/1qh6PikVtqWXM-pz$_2EF6AemoX7Q6tMgQ$

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv e-prints*, page arXiv:1803.01271, Mar. 2018.

[2] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254*, 2019.

[3] Szu-Wei Fu, Tao-Wei Wang, Yu Tsao, Xugang Lu, and Hisashi Kawai. End-to-End Waveform Utterance Enhancement for Direct Evaluation Metrics Optimization by Fully Convolutional Neural Networks. *arXiv e-prints*, page arXiv:1709.03658, Sept. 2017.

[4] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *CVPR*, 2019.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385, Dec. 2015.

[6] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. Single-Channel Multi-Speaker Separation using Deep Clustering. *arXiv e-prints*, page arXiv:1607.02173, July 2016.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec. 2014.

[8] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal Convolutional Networks for Action Segmentation and Detection. *arXiv e-prints*, page arXiv:1611.05267, Nov. 2016.

[9] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal Convolutional Networks: A Unified Approach to Action Segmentation. *arXiv e-prints*, page arXiv:1608.08242, Aug. 2016.

[10] Yi Luo, Zhuo Chen, and Nima Mesgarani. Speaker-independent Speech Separation with Deep Attractor Network. *arXiv e-prints*, page arXiv:1707.03634, July 2017.

[11] Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation. page arXiv:1809.07454, Sept. 2018.

[12] Pedro Morgado, Nuno Vasconcelos, Timothy Langlois, and Oliver Wang. Self-Supervised Generation of Spatial Audio for 360 Video. *arXiv e-prints*, page arXiv:1809.02587, Sept. 2018.

[13] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. SEGAN: Speech Enhancement Generative Adversarial Network. *arXiv e-prints*, page arXiv:1703.09452, Mar. 2017.

[14] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.