A cappella: Audio-visual Singing Voice Separation (Supplementary Material)

Juan F. Montesinos*
juanfelipe.montesinos@upf.edu
Venkatesh S. Kadandale*
venkatesh.kadandale@upf.edu
Gloria Haro
gloria.haro@upf.edu

Universitat Pompeu Fabra
Barcelona, Spain

1 Models' details and comparison

1.1 LLCP

The LLCP model is one of the baseline models in this work. It is basically the model used in Ephrat et al. [1] for audio-visual speech separation but with a small modification. We use an implementation[1] of LLCP that differs from that of Ephrat et al. only in the final fully-connected layer. The original model uses a fully-connected layer of size 1024. Due to the GPU memory limitations, we replace this layer with a fully-connected layer of size 512. Yet, note that, among the considered models in this work, LLCP has the highest number of trainable parameters (see Table 1). Again, due to limited GPU memory, we use a reduced batch size of 4 for training the LLCP model.

1.2 The different Y-Net variants

One of our design criterion was to develop a network which can exploit motion information with the least possible number of parameters, in order to avoid overfitting. In Table 1 we can find a comparison between the models in terms of the amount of trainable parameters. All our networks have much less parameters compared to classic computer vision networks; they also have less parameters than the LLCP model which is one of our baseline models. It is interesting to highlight that the highest performance is reached by the network with the least amount of parameters, Y-Net-g.

1.2.1 U-Net

U-Net is an encoder-decoder architecture with the skip connections in between. It is widely used in several fields such as image segmentation [5], image to image conversion [2], or blind source separation (BSS) [3], to name a few. This is why there are several variations of

* These authors contributed equally to this work.

[1] https://github.com/vitrioil/Speech-Separation

Architecture	Visual Network	Audio-visual network
ResNet18	11.4M	-
3D-ResNet18	33.4M	-
LLCP	2.6M	15.3M
Y-Net-m	3.1M	11.4M
Y-Net-g	1.3M	9.4M

Table 1: Number of parameters (M for million) for the different architectures compared to common networks in computer vision (ResNet18 and 3D-ResNet18).

it, each of them incorporating different modifications adapted to the corresponding field. The original architecture consists of a set of blocks composed of two consecutive convolutions, max pooling and an activation. The up-sampling blocks rely on up-convolutions to restore the feature maps to the original size. On contrary, the U-Net variants typically used in BSS have a single convolution and use bilinear upsampling not to overfit. In order to improve the performance of the architecture, our U-Net's blocks consist of a single 5×5 convolution followed by leaky ReLU and a frequential or a spatial max-pooling. Each decoder block consists of a single 3×3 convolution followed by batch normalisation, ReLU and bilinear upsampling. This allows us to finely control the dimensions of the intermediate feature maps and the amount of blocks. This way we can keep enough temporal resolution to ensure a proper audio-visual synchronisation when we want to fuse the visual features with the audio features. Note that not only the architecture is important but the input too. U-Net performs a center crop to align the incoming features from the skip connections if the shapes are mismatching. We have chosen the STFT parameters, the track duration and the audio sampling rate such that our spectrograms are computationally efficient and do no require to be center cropped.

Block #	Type of block	Output channels	Kernel	Padding	Output shape
1	Spatial	32	5×5	2	128 × 128
2	Spatial	64	5×5	2	64 × 64
3	Spatial	128	5×5	2	32 × 32
4	Spatial	256	5×5	2	16 × 16
5	Frequential	256	5×5	2	8 × 16
6	Frequential	256	5×5	2	4 × 16
-	Bottleneck	256	3×3	1	8 × 16

Table 2: Encoder details of U-Net, which is the audio encoder of the Y-Net models. Note that the encoder shapes are symmetrical. The input tensor has the shape of $2 \times 256 \times 256$.

1.2.2 Y-Net-m

We use a light 3D-ResNet-like network to be capable of extracting motion information at a low cost. It is a 3-block CNN network where the first block uses 3D convolutions while the latter blocks use 2D convolutions. This results in a light but powerful 3-million-parameter network. It has three and ten times less parameters than traditional ResNet18 or 3D-ResNet18, respectively, as shown in Table 1. Details about the network can be found in

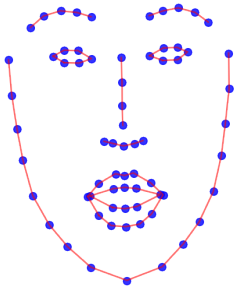

Figure 1: Graph with spatial edges (i.e. without temporal edges) fed into Y-Net-g.

Block #	Output channels	Output shape
1	32	100×68
2	32	100×68
3	64	50×68
4	64	50×68
5	128	25×68
6	128	25×68
7	256	13×68
8	256	13×68

Table 4: Y-net-g blocks in terms of output channels. Note that the output shape keeps the dimensionality of the spatial graph. The input tensor has the shape of $2 \times 100 \times 68$.

Table 3.

Block #	Type of block	Output channels	Kernel	Padding	Stride	Output shape
0	Basic Stem	64	3×7×7	1×2×2	1×3×3	100×48×48
1	Spatio-temporal	64	3×3×3	1×1×1	1×1×1	100×48×48
2	Spatial	128	3×3	1×1	2×2	100×24×24
3	Spatial	256	3×3	1×1	2×2	100×12×12

Table 3: Y-Net-m blocks in terms of output channels, type of block and convolution parameters. Note that the output shape keeps the temporal dimensionality. The input tensor has the shape of $3 \times 100 \times 96 \times 96$.

1.2.3 Y-Net-g

In Y-Net-m, we were using frames cropped around the lips to minimise the possibility the network learns visual singer identity to perform the separation. Face landmarks are an appearance-less representation which encode motion and position information. We treat the landmarks as a spatio-temporal graph $G = (J, E)$ where the node set can be defined as $J = j_{ti} \quad \forall\, t = 1,...,T, i = 1,...,N$. Each node's vector is a tuple encoding the cartesian position of the associated landmark in the image. As it is a spatio-temporal graph, the nodes are connected both spatially and temporally. This way, we can consider two types of edges, the spatial ones, defined as shown in Figure 1; and the temporal ones, where each node in the graph is connected to the same nodes of adjacent spatial graphs.

We used a variation of the graph convolutional network from [6], details about the number of blocks can be found in Table 4.

2 Training Setup

All the models have been trained using stochastic gradient descent, with a momentum of 0.8 and a weight decay of 10^{-5}. The learning rate is 0.01. Batch size of 10. In case of Y-Net-m,

we use pretrained weights from Kinetics [4] and its statistics to normalise the input frames.

3 Demo videos

We also provide demo videos[2] for the qualitative assessment of the performance of our models and the baseline models (U-Net and LLCP). The demos cover examples from the real-world multi-voice *a cappella* music videos, our multi-voice music video, single voice plus accompaniment, and two lead voices plus accompaniment samples from our unseen-unheard test subsets. We recommend using headphones to listen to the demos. The demos show that our Y-Net performs clearly better than the baseline models in such challenging examples.

References

[1] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. In *SIGGRAPH*, 2018.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. doi: 10.1109/CVPR.2017.632.

[3] A Jansson, E Humphrey, N Montecchio, R Bittner, A Kumar, and T Weyde. Singing voice separation with deep U-Net convolutional networks. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 23–27, 2017.

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[6] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[2] https://ipcv.github.io/Acappella/demos/