EE698R: Project Report - Audio Source Separation

Saaransh Aggarwal (190729)
Videeta Sharma (190959)

I. Introduction

The problem statement is a modified version of Music Source Separation, where we have to split the given audio file into voices and accompaniments(background). We have followed the paper Music Source separation using Hourglass networks. We have used Stacked Hourglass Network to perform the task at hand. The paper mentions that a stacked hourglass network is used to separate music sources. It was originally intended for human pose estimation in natural photos. The network builds masks for each audio source by learning features from a spectrogram image at various scales. As it travels across stacked hourglass modules, the estimated mask is refined. The paper proposes that this network can be used to detect sources for the source separation problem. We have tried to follow this paper and implement the approach given by their authors by making some minute changes. We use only a single network for both voice and accompaniments to keep the space and time complexity in check.

II. Recent Work

Separating singing voice or sounds of individual instruments from a mixture has grabbed a lot of attention in recent years. Due to its end-to-end learning characteristic, deep neural networks that are used in computer vision research can be directly applied to audio signal processing area with minor modifications. Since the magnitude spectrogram of an audio signal can be treated as a 2D single-channel image, convolutional neural networks (CNNs) have been successfully used in various music applications, including the source separation task. Simple deep feed-forward networks consisting of multiple fully-connected layers showed reasonable performance for supervised audio source separation tasks. Since an audio signal is time series data, it is natural to use a sequence model like recurrent neural networks (RNNs) for music source separation tasks to learn temporal information.It is proven that FCNs are also effective in signal processing.

III. Dataset And Pre Processing

Convolutional neural networks (CNNs) have been effectively used in several music applications, including the source separation because the magnitude spectrogram of an audio signal may be considered as a 2D single-channel image. The CNNs use spectrogram images of a music signal as inputs to create masks for each separate music source. An hourglass module extracts both holistic features from low-resolution feature maps and fine details from high-resolution feature maps using low-resolution feature maps.

We have used the MIR 1K dataset to implement this paper. There are 11 male voices and 8 female voices in the MIR 1K dataset. We have used 8 Males and 6 females for creating our training dataset and the remaining 3 males and 2 females have been used in the test dataset. The training to test data points ratio is $80 : 20$. To begin with, we have first resampled our audio file to $8000Hz$ to increase the duration of spectrograms in a batch and to speed up training. For each sample, magnitude spectrograms of mixed and separated sources are generated, which are divided by the maximum value of the mixed spectrogram for data normalization. The STFT has been calculated with a window size of 1024 and a hop size of 256. The spectrograms have 512 frequency bins and the width of the spectrogram depends on the duration of the music sources. For all the music sources, the width of the spectrogram is at least 64 - corresponding to approximately duration of 2 seconds. Thus, we fix the size of an input spectrogram to $512, 64$. Hence, the size of the feature maps at the lowest resolution is $32, 4$.

IV. Model Architecture - Hourglass Module

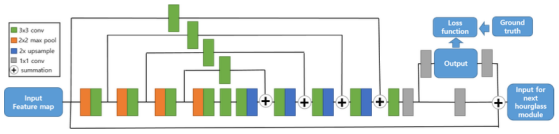

Fig. 1. Hourglass Module

A single hourglass module consists of a set of 4 downsampling layers coupled with convolutional layers and 4 upsampling layers coupled with convolutional layers. There are 4 skip connections in 1 module which pass the information between matching resolution feature maps, the connected is made through a $3x3$ convolutional layer. In the downsampling step, $2x2$ max pool layer has been used and in the upsampling step, nearest neighbour interpolation has been used. This upsampling and downsampling helps the model to capture the features at different scales. Finally, the feature

maps are passed through $1x1$ convolution layers to generate output - masks for each source : Voice and background. After this, the input given to this module is added to output which has been convolved to match it's dimension to input of the module, which is $512, 64, 64$ and then this is passed as input to the next hourglass module. The output masks calculated by each module are multiplied with the mixed spectrogram to obtain the estimated masks, which are then used to calculate the loss for the current module. This intermediate supervision helps improve the training speed and the performance of the network.

V. Model Architecture - Stacked Hourglass

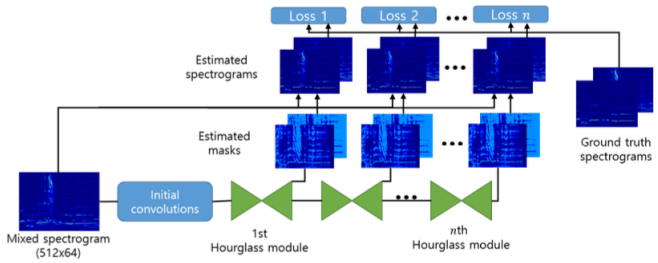

Fig. 2. Stacked Hourglass

The paper proposes a model of 4 hourglasses modules stacked together. The input mixed spectrogram is made to pass through initial convolutions- 4 layers : the first one with kernel size of $7, 7$ and rest with kernel size of $3, 3$ with output channel dimension of each layer being $64, 128, 128, 64$, after which the size of input is $512, 64, 64$. This is passed through the 4 hourglass modules and the loss of each module is added to a final $loss$ variable which is then updated. Our aim is to minimise the total loss and hence find the best parameters of the model. The output dimension of the network is $512, 64, 2$, where $512, 64$ represents the size of the input spectrogram that was passed to the model and 2 represents the number of sources we need to separate, which is 2 in our case.

VI. Training

We have trained the model using a batch size of 4. The Adam optimiser has been used with a learning rate of 0.001. We have used the following loss function:

$$J(i,j) = ||Y_i - XM_{ij}||$$

$$J = \sum_{i=1}^{C}\sum_{j=1}^{D} J(i,j)$$

Here, M_{ij} is the mask generated for Audio Source i in Hourglass module j. X is the input spectrogram and Y_i is the ground truth for audio source i. $J(i,j)$ is the loss for audio source i from hourglass module j. The final loss, J is the sum of losses for all audio sources from each hourglass module, D is the number of hourglass modules in the network and C is the number of audio sources to be separated. We directly used the output of the last $1, 1$ convolutional layer as the mask. While it is natural to use the sigmoid function to restrict the value of the mask to [0,1], we empirically found that not applying the sigmoid function boosts the training speed and improves the performance. Since sigmoid activations vanish the gradient of the inputs that have large absolute values, they may diminish the effect of intermediate supervision

VII. Evaluation

In the test phase, the magnitude spectrogram of the input source is cropped to network input size and fed to the network sequentially. The output of the last hourglass module is used for testing. We set the negative values of output masks to 0 in order to avoid negative magnitude values. The masks are multiplied by the normalized magnitude spectrogram of the test source and unnormalized to generate spectrograms of separated sources. We did not change the phase spectrogram of the input source, and it is combined with the estimated magnitude spectrogram to retrieve signals for separated sources via inverse STFT. After this, the audio is sampled back to $16000Hz$ and these outputs are used to calculate the evaluation metrics : GNSDR, GSAR and GSIR. To compare the performance as suggested in the class presentations, We have implemented 3 types of stacked hourglass models, with $1, 2$ and 4 hourglass modules respectively. We have also implemented a baseline CNN model to compare our results.

VIII. Observations

The number of trainable parameters in these models are:

1) SH1 : 8868038
2) SH2 :17475276
3) SH4 : 34689752
4) CNN : 525574

We did not witness the pattern of increase in performance in $SH1$, $SH2$ and $SH4$, which was mentioned in the paper. Our $SH4$ gave results comparable to the $SH1$ and $SH2$ model. The training approach used by us is different than the one mentioned in paper - they have used the voices of only 2 people - namely $abjones$ and amy to train the model, whereas we have made a $80 - 20$ split of data ensuring proportionate distribution of male and female voices. The paper mentions using 15000 iterations to train these models, whereas we have trained our models on 1500 iterations, but the results are almost comparable. We trained the $SH - 4$ model on 500 and 1500 iterations and witnessed an increase in performance in the second case. The training time was approximately 10 minutes for $SH1$, 20 minutes for $SH2$, 30 minutes for $SH4$ with 500 iterations and around 45 minutes for $SH4$ with 1500 iterations on GPU. The paper mentioned a training time of 3 hours.

IX. Results

For Accompaniments

Model	GNSDR	GSIR	GSAR
SH1	9.08708454	12.57614819	12.13160477
SH2	9.14332844	12.92863707	11.92048093
SH4 - 500	8.0546265	11.76520215	11.04173583
SH4 - 1500	9.41683138	15.56715204	11.01817018
CNN	-6.03407712	-3.02142562	2.06437262

For Voice

Model	GNSDR	GSIR	GSAR
SH1	9.79673213	15.98145621	11.32328618
SH2	9.93643997	16.77771608	11.30910882
SH4 - 500	8.74925667	14.95189058	10.40547879
SH4 - 1500	9.11034063	14.40147653	11.20639122
CNN	-0.88577581	5.08411082	1.78525219

The results that we were trying to replicate are :

Singing voice			
Method	GNSDR	GSIR	GSAR
MLRR [37]	3.85	5.63	10.70
DRNN [6]	7.45	13.08	9.68
ModGD [23]	7.50	13.73	9.45
U-Net [8]	7.43	11.79	10.42
SH-1stack	10.29	15.51	12.46
SH-2stack	10.45	15.89	12.49
SH-4stack	**10.51**	**16.01**	**12.53**

Accompaniments			
Method	GNSDR	GSIR	GSAR
MLRR [37]	4.19	7.80	8.22
U-Net [8]	7.45	11.43	10.41
SH-1stack	9.65	13.90	12.27
SH-2stack	9.64	13.69	**12.39**
SH-4stack	**9.88**	**14.24**	12.36

Fig. 3. Results of the paper

X. DEMO

The audio obtained by us after separating the sources on few examples from the $MIR-1K$ dataset can be accessed here.

References

[1] Sungheon Park, Taehoon Kim, Kyogu Lee, Nojun Kwak(2018), "Music Source Separation using Stacked Hourglass Networks", in 19th International Society for Music Information Retrieval Conference, Paris, France, 2018