INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Transfer Learning for Improving Singing-voice Detection in Polyphonic Instrumental Music

Yuanbo Hou, Frank K. Soong†, Jian Luan‡, Shengchen Li**

*Beijing University of Posts and Telecommunications
†Microsoft Research Asia
‡Microsoft Search Technology Center Asia, XiaoIce

{hyb,shengchen.li}@bupt.edu.cn, {frankkps,jianluan}@microsoft.com

Abstract

Detecting singing-voice in polyphonic instrumental music is critical to music information retrieval. To train a robust vocal detector, a large dataset marked with *vocal* or *non-vocal* label at frame-level is essential. However, frame-level labeling is time-consuming and labor expensive, resulting there is little well-labeled dataset available for singing-voice detection (S-VD). Hence, we propose a data augmentation method for S-VD by transfer learning. In this study, clean speech clips with voice activity endpoints and separate instrumental music clips are artificially added together to simulate polyphonic vocals to train a *vocal/non-vocal* detector. Due to the different articulation and phonation between speaking and singing, the vocal detector trained with the artificial dataset does not match well with the polyphonic music which is singing vocals together with the instrumental accompaniments. To reduce this mismatch, transfer learning is used to transfer the knowledge learned from the artificial speech-plus-music training set to a small but matched polyphonic dataset, *i.e.*, singing vocals with accompaniments. By transferring the related knowledge to make up for the lack of well-labeled training data in S-VD, the proposed data augmentation method by transfer learning can improve S-VD performance with an *F-score* improvement from 89.5% to 93.2%.

Index Terms: Singing-voice detection, music information retrieval, transfer learning, data augmentation

1. Introduction

Singing-voice detection (S-VD) is to detect vocal frames of given music clips. Successful detection of singing voice regions in polyphonic music is critical to music information retrieval (MIR) [1] tasks, such as music summarization [2], retrieval [3], transcription [4], genre classification [5], and vocal separation [6].

Recently, deep learning has been applied to S-VD. Deep neural networks [7] are used to estimate an ideal binary spectrogram mask that represents the spectrogram bins in which the vocal is more prominent than the accompaniments. Convolutional neural networks (CNN) have been used to boost the performance in MIR [8], with an efficient model built on temporal and timbre features. Recurrent neural networks (RNN) are employed to predict time-frequency masks of multiple source signals, then masks are multiplied with the original signal to obtain the desired isolated source [9]. Above models can be refined with more accurate frame-level labels, also known as strong labels [10]. However, labeling strong label is time-consuming, hence usually datasets have been used with only small number of songs with strong labels in training.

To overcome the limitation of lack of frame-level labeled training data in S-VD, we propose a data augmentation [11, 12] method for S-VD by transfer learning. Transfer learning [13] extracts representations learned from a source task and applies to a similar but different target task. Transfer learning can alleviate the problem of insufficient training data for the target task and tend to generalize the model. Many transfer learning methods [14–16] related to S-VD use strong labels, and some methods even need clean singing recordings. Datasets with strong labels or clean singing recordings are scarce. However, clean speech corpora and instrumental music datasets are widely available in the Internet, and the endpoints of clean speech can be easily detected. Hence, these clean speech clips and instrumental music clips can be artificially added together to simulate polyphonic vocals for training a vocal detector. To make up for the lack of well-labeled training data in S-VD, this paper proposes to transfer the latent representations of vocal detector in speech-plus-music domain to detect singing voice in polyphonic music domain. Given a source domain $D_S = \{X_S, f_S(X)\}$ and source task T_S, a target domain $D_T = \{X_T, f_T(X)\}$ and target task T_T. In this paper, X_S denotes audio clips synthesized by speech clips and instrumental music, T_S is speech activity detection, and f_S is latent representations mapping function learned by the convolutional layers. X_T denotes polyphonic music and T_T is S-VD. Transfer learning [13] aims to improve the learning of the target mapping function $f_T()$ in D_T using the information in T_S and D_S.

To investigate the performance of data augmentation by transfer learning in S-VD and explore the possibility of transferring the knowledge from speech to singing voice, the learned representation which retains relevant information of speech clips, will be transferred to S-VD which is a similar but different target task. Although there is difference between speaking and singing, and vocals characteristics may also vary with the change of accompaniments [17], they still have useful similarities to be exploited. In addition, sharing knowledge of voice between speech clips and the singing voice enable the detector to understand human voice, speech or singing vocal, in a more general and robust form.

The main contributions of this paper are: 1) to overcome the lack of frame-level labeled training data in S-VD, we propose a data augmentation method for S-VD by transfer learning; 2) we investigate the performance of transferring representations learned in speech activity detection to detect singing voice, and find the lower convolutional layers learn more basic local representations which are more effective for detecting vocals in polyphonic music; 3) patterns of convolutional filters are visually analyzed, and the learned knowledge of voice between detectors trained with synthesized audio clips and polyphonic music

*Work performed as an intern at Microsoft Research Asia.

http://dx.doi.org/10.21437/Interspeech.2020-1806

clips is compared.

The rest of the paper is organized as follows. Section 2 shows the proposed method. Section 3 describes experiments and analyzes results in detail. Section 4 gives the conclusions.

2. Proposed method

The proposed method for S-VD is illustrated in Figure 1. To overcome the lack of well-labeled training data in S-VD, transfer learning extracts knowledge of voice from the source task and applies it to the target task to detect singing voice. This is crucial for our task, where the training data for the target task is insufficient to train a good detector model. In the source task, CNN is trained for detecting speech activity frames in synthesized audio clips. The knowledge of voice learned from the large-scale dataset in source task is then transferred to the target task. Due to the different articulation and phonation between speaking and singing [17], the target task is more challenging. So a convolutional recurrent neural network (CRNN) is trained with a small set of data collected in the target task to detect the vocal frames.

2.1. Source task: speech activity detection

The source task is to detect the speech activity endpoints in the synthetic audio clips to learn the representations of voice. For the good performance of CNN in MIR [18, 19], CNN is used as the detector in the source task. Figure 2 shows the details of CNN. The waveforms of synthetic audio clips are converted to log mel spectrogram, which is a 2D representation that approximates human auditory perception. This computationally efficient input has been shown to be effective in MIR tasks such as music classification [20].

To comprehensively consider the contextual information of audio, the input of CNN is a moving data block, consisting of the preceding L frames and the succeeding L frames of the current frame, the shift between succeeding blocks is one frame. Each block contains *(2L+1)* frames. L determines the range of contexts visible in the model at every frame.

The detector consists of a series of convolutional and pooling layers. To preserve the time resolution of the input, pooling is applied to the frequency axis only. As shown in Figure 2, where (64, (3, 3)) corresponds to (convolutional filters, (receptive field in time, frequency)). Pooling layer is specified by (pooling length in time, frequency). In addition, to reduce the gradient vanishing problem in deep networks training, gated linear units (GLUs) [21] are used in convolutional layers. They provide a linear path for gradient propagation while keeping nonlinear capabilities through the sigmoid operation. Given W and V as convolutional filters, b and c as biases, X as the input features or the feature maps of interval layers and σ as sigmoid function, GLUs are defined as:

$$Y = (W * X + b) \odot \sigma(V * X + c) \tag{1}$$

where the symbol $\odot$ is the element-wise product and $*$ is the convolution operator. By weighting time-frequency units according to their unique time positions, GLUs can help network attend to voice and ignore unrelated accompaniments.

The source task aims to detect whether there is speech in a frame, which is a binary classification task. If sigmoid function with one unit is used in the last layer of the CNN, thresholds are needed to determine the label of each frame. To avoid the impact of thresholds on detection results, softmax function with two output units are used in the last layer. The label corresponding to the larger output probability is used as the final label of each frame.

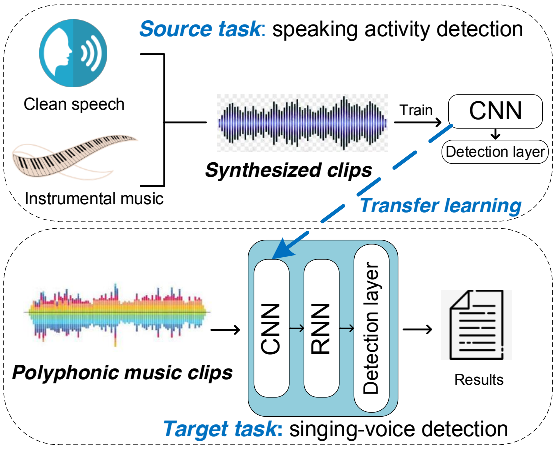

Figure 1: *Framework of the proposed method.*

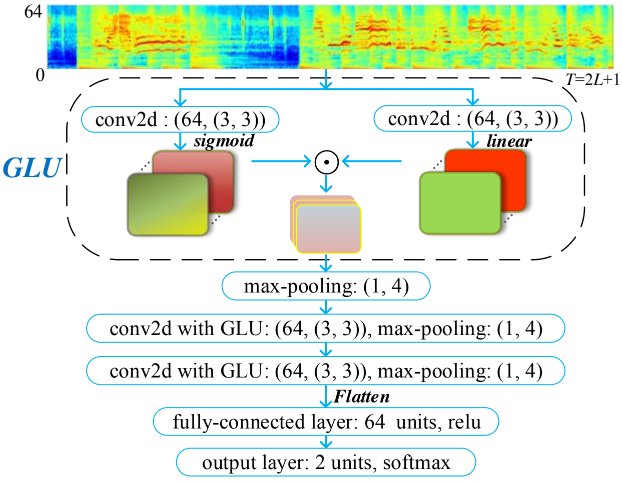
64
0
T=2L+1
GLU
conv2d : (64, (3, 3))
sigmoid
conv2d : (64, (3, 3))
linear
⊙
max-pooling: (1, 4)
conv2d with GLU: (64, (3, 3)), max-pooling: (1, 4)
conv2d with GLU: (64, (3, 3)), max-pooling: (1, 4)
Flatten
fully-connected layer: 64 units, relu
output layer: 2 units, softmax

Figure 2: *Details of the CNN architecture in the source task.*

2.2. Target task: singing-voice detection

When the detection aims at polyphonic songs, relying on the CNN trained on the artificial synthesized audio clips may be inadequate, because both articulation and phonation between speech and singing are different [17]. In addition, the vocals in polyphonic music will change together with the accompaniments. It is known that singing voice evolves in songs, which can bring more variation to the vocal representations.

Compared with the source task based on synthesized data, the target task is more challenging. Vocals, which change together with the accompaniments, are difficult to detect in polyphonic music, so a recurrent layer is added to the CNN to capture the long-term temporal contextual information of audio signal. In the target task, the detector is a convolutional recurrent neural network (CRNN), which adds a recurrent layer after the last convolutional layer of the CNN in Figure 2. The rest of the CRNN is consistent with the CNN in Figure 2.

There are two modes for transferring knowledge from the source task to the target task depending on whether the transferred parameters are updated during the training phase in the target task. In this paper, a comparative study is conducted to investigate the effects of two modes on the proposed system.

2.3. Visualizing the patterns of convolutional filters

It is difficult to display or measure the knowledge in speech and sing voice directly. Fortunately, convolutional layers in the model can extract the features of the input data, which are indirect representations of the knowledge contained in the speech and singing voice. To intuitively inspect the differences of knowledge in speech and singing voice, the gradient ascent [22] is used to show the patterns learned from the data by convolutional filters. Given X is a blank input image, x is the point in X, η is learning rate and $a_{ij}(x)$ is the output of the filter at (i, j) after convolution. The pattern of the filter can be calculated by:

$$X = X + \eta \partial a_{ij}(x) / \partial x \tag{2}$$

The visualization method applies gradient descent to the value of the input image of a convolutional layer so as to maximize the response of a specific filter. Repeat this step many times, the resulting image will be one that the chosen filter is maximally responsive to, *i.e.* the pattern of the filter.

3. Experiments and results

3.1. Dataset and Experiments Setup

For the source task, artificially synthesized audio clips are required to train the CNN, which is able to learn the spectral and temporal features of speech signal. For this reason, a private clean speech corpus from Microsoft XiaoIce group with 100 speakers, each speaker recorded about 20 minutes of speech, in total for about 34.5 hours, was artificially added together with an instrumental music dataset at signal-to-noise ratio of 0 dB to simulate polyphonic music clips. The endpoints of voice in the clean speech are detected, hence the frame-level label of the synthesized polyphonic audio clips are obtained, accordingly.

For the target task, the dataset consisting of 120 polyphonic songs is divided into training and validation sets. The test set consists of another 60 polyphonic songs. Each song in the target task is about 4 minutes long and there is no intersection of singers in training, validation and test sets. These songs are annotated with frame-level *on*/*off* labels as the ground truth representing the singing voice is on or not in each audio frame. More details, source codes and samples, please see here[1].

In training, log mel spectrogram is extracted using STFT with Hamming window length of 40 ms, which has sufficient time and frequency resolution. An overlap of 50% between two adjacent windows is used to smooth the spectrograms. Then 64 mel filter banks are applied. Dropout and normalization are used to prevent over-fitting. Both the source and target tasks are binary classification tasks, hence Adam optimizer [23] is used to minimize the binary cross entropy.

Given the frame-wise detection results for each frame, we can calculate precision (*P*), recall (*R*) and *F-score* (*F*) of the detection performance. They are defined as:

$$P = \frac{N_{tp}}{N_{tp} + N_{fp}}, R = \frac{N_{tp}}{N_{tp} + N_{fn}}, F = \frac{2P \cdot R}{P + R} \tag{3}$$

where N_{tp}, N_{fp} and N_{fn} are the numbers of true positives, false positives and false negatives, respectively. Higher *P*, *R* and *F* indicate a better performance [24].

3.2. Results and analysis

To consider the long-term contextual information of the audio clips, the input of CNN is a block totaling (*2L+1*) frames. Figure 3 shows the results of CNN trained with blocks of different lengths, on the x-axis is different values of T frames, *i.e.* (*2L+1*) frames, and on the y-axis is *F-score*. The comparison in Figure 3 reveals that performance of detector does not improve monotonically with increased length of input block, and setting *T*=25 achieved a good trade-off between *F-score* and computational complexity. Consequently, this value is used for all later experiments.

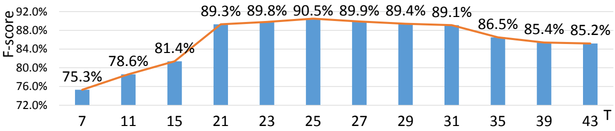

Figure 3: *Results of different input lengths in the source task.*

Given L_1, L_2 and L_3 denote the first, the second and the third convolutional layer with GLUs, L_{all} denotes all convolutional layers. In transfer learning, the L_i in CRNN in the target task will accept the learned parameters of L_i in CNN in the source task. In *Fixed* mode, the parameters of L_i in CRNN will no longer be updated during the backpropagation, other layers of CRNN are trained normally. In *Fine-tuning* mode, the L_i in CRNN will continue to adapt its parameters with the target dataset. Due to the limitation of space, *F-score* of two modes on the test dataset of target task, and the number of trainable parameters (*N.params*) are shown in Table 1.

As shown in Table 1, the performance of transferring the all convolutional layers of the CNN in the source task and freeze them yields the worst result. However, transferring L_1 with fine-tuning yields the best result. Transferring the knowledge of L_2 or L_3 does not perform as well as L_1. This may due to lower level convolutional layers may contain more generic features (e.g. edge or frequency detectors) that are useful for both source and target tasks. They learn the basic and local features of voice, but high level convolutional layers may become more irrelevant in learning some high level representations. The singing voice in the target task is more complex than the speech in the source task, because the singing voice will change with the polyphonic accompaniments. Hence, the high level representations of voice learned by the higher convolutional layers from speech may not match the target task, resulting transferring this knowledge does little in helping the target task. To show the difference between source domain and target domain more intuitively, Figure 4 shows the results of high-dimensional acoustic features clustering of synthesized polyphonic audio samples and singing voice samples in polyphonic music by *t-SNE* [25]. It can be seen from the Figure 4 that the features of the synthesized polyphonic audio samples in the source task are clearly separated from the features of the actual singing voice samples in the target task after high-dimensional clustering. Therefore, the synthesized polyphonic audio samples cannot completely simulate the characteristics of the singing voice in polyphonic instrumental music, which leads to the fact that the knowledge learned by the vocal detector from the source task cannot be fully applied to the target task.

Table 1: *The results of two different transfer modes.*

Transferred layer	***Fixed***		***Fine-tuning***	
	F-score	*N.params*	*F-score*	*N.params*
L_1	91.9%	20.58K	**93.2%**	20.72K
L_2	91.7%	13.33K	92.0%	20.72K
L_3	91.1%	13.33K	91.7%	20.72K
L_{all}	82.6%	5.79K	92.3%	20.72K

[1] https://github.com/moses1994/singing-voice-detection

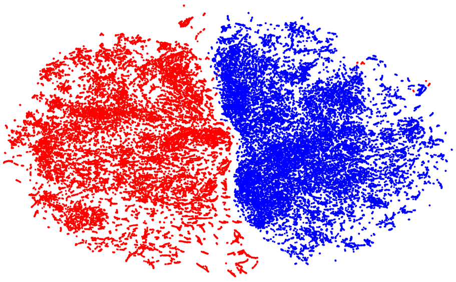

Figure 4: *Visualization of features distribution using t-SNE [25], the red points and blue points denote singing voice samples in the target task and synthesized polyphonic audio samples in the source task, respectively.*

To gain deeper insights of the knowledge in the source and target tasks, we visualized the learned patterns of filters in convolutional layers with GLUs. Due to the limitation of space, patterns which are randomly selected from different filters, is shown in Figure 5. Please see here[1] for more details. In Figure 5, for the same model in a task, L_1 learns more obvious basic local features of the input spectrogram than L_2 and L_3. For different models in the two tasks, compared with the learned patterns of L_2 and L_3, the patterns of L_1 in the two task are more similar. This may be the reason why transferred L_1 performs best in Table 1. Since the local representations of voice learned in the source task and target task are relatively similar, transferring this knowledge to target domain can help the model obtain a more general and robust vocal detection. For the L_2 and L_3, the high level representations they learned from different domain is quite different, hence transferring this knowledge provides little help to the target task.

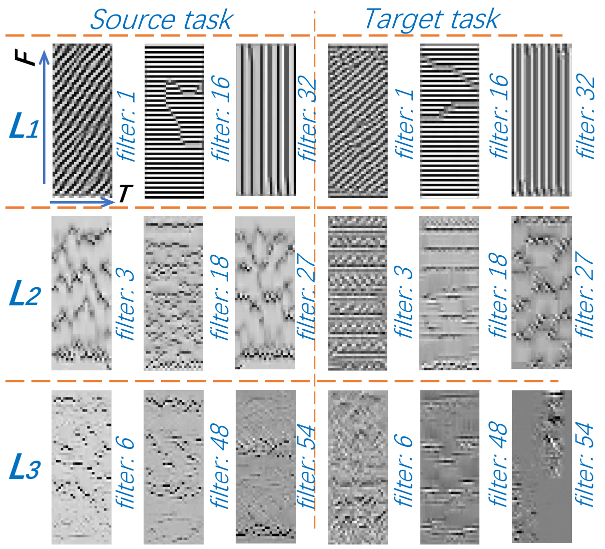

Figure 5: *Patterns of different filters in L_i, for each subgraph, the x-axis is time (T) and the y-axis is frequency (F).*

When the optimal transfer mode is determined, the detection results on the test set in the target task are shown in Table 2. The baseline is a deep CNN architecture with 3-by-3 2D convolution layers [26] trained directly with the dataset in the target task. And [26] implies that CNN may benefit from looking at a varying range of time and frequency to learn vocal-specific characteristics, such as timbre [27]. For most polyphonic songs in Table 2, the results of the proposed data augmentation method by transfer learning have better *F-score* higher than the baseline. A very robust sample of detection results is shown in Figure 6.

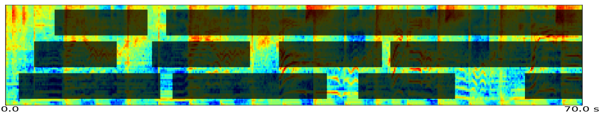

Figure 6: *From top to bottom, they are visualization of the ground truth, the results of proposed method and baseline, respectively. Shaded parts indicate singing voice activity.*

The singing-voice detector trained by the transfer learning was also tested on MUSDB18 [28] to compare the performance on the publicly available music dataset. MUSDB18 contains 150 tracks (∼10h duration) of different styles, the 150 tracks are split into 100 tracks for training, and 50 for testing. The detection results on the test set in MUSDB18 are shown in Table 3. In addition to the precision, the model trained by transfer learning in this paper is better than the baseline in recall and *F-score*. The reason may be that the training data in this paper has more types of samples, and the model can learn more different information in the process of transfer learning.

Table 2: *The detection results on the test set in the target task.*

Polyphonic	***Frames***		***Baseline [26]***			***Transfer learning***		
song	*off*	*on*	*P (%)*	*R (%)*	*F (%)*	*P (%)*	*R (%)*	*F (%)*
No.1	2938	5384	82.8	85.4	84.1	92.9	97.3	**95.1**
No.2	4166	7476	83.0	90.0	86.3	89.6	98.6	**93.9**
No.3	4945	5754	86.5	91.6	89.0	89.3	96.5	**92.8**
No.4	3390	6098	79.6	91.4	85.1	84.1	91.8	**87.8**
No.5	5844	8366	96.4	93.2	89.7	88.4	92.9	**90.6**
No.6	2744	4793	84.5	92.3	88.2	86.5	91.7	**89.1**
No.7	6423	2911	89.5	94.4	**91.9**	86.7	93.7	90.1
No.8	1475	4561	90.2	94.3	92.2	91.0	97.8	**94.2**
No.9	2458	9922	66.6	89.9	76.5	70.8	91.4	**79.8**
•••	•••	•••	•••	•••	•••	•••	•••	•••
No.60	3218	7220	96.5	95.7	96.1	95.8	97.7	**96.8**
Overall			86.1	93.2	89.5	90.1	96.0	**93.2**

Table 3: *The detection results on the test set in MUSDB18 [28]*

Baseline [26]			***Transfer learning***		
P (%)	*R (%)*	*F (%)*	*P (%)*	*R (%)*	*F (%)*
96.83	81.64	88.61	92.98	96.57	94.74

4. Conclusions

To overcome the limitation of insufficient frame-level labeled training data in S-VD, this paper proposes a data augmentation method for S-VD by transfer learning. Due to the shortage of well-labeled polyphonic music data, a training set of clean speech and instrumental music are added together to construct the basic training dataset. The knowledge learned from the artificial training set is then transferred to a small but more matched dataset of singing vocals with instrumental accompaniments, by adapting the corresponding detector parameters to make a better singing voice detector.

By analyzing the patterns of filters, we found the patterns learned from the source task does not match well with target task. This mismatch can be reduced by fine-tuning the convolutional filters parameters at the lower layers of the model. By transferring the related knowledge to make up for the lack of well-labeled training data in S-VD, the proposed data augmentation method by transfer learning can improve S-VD performance with an F-score improvement from 89.5% to 93.2%.

5. References

[1] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[2] B. Logan and S. Chu, "Music summarization using key phrases," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, vol. 2, pp. II749–II752.

[3] N. Hu, R. B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 185–188.

[4] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180.

[5] C. Xu, N. C. Maddage, and X. Shao, "Automatic music classification and summarization," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 441–450, 2005.

[6] A. Mesaros, T. Virtanen, and A. Klapuri, "Singer identification in polyphonic music using vocal separation and pattern recognition methods." in *International Society for Music Information Retrieval*, 2007, pp. 375–378.

[7] S. D. You, C. Liu, and W. Chen, "Comparative study of singing voice detection based on deep neural networks and ensemble learning," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, p. 34, 2018.

[8] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples." in *International Society for Music Information Retrieval*, 2016, pp. 44–50.

[9] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks." in *International Society for Music Information Retrieval*, 2014, pp. 477–482.

[10] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *EUSIPCO*, 2016, pp. 1128–1132.

[11] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition—tangent distance and tangent propagation," in *Neural networks: tricks of the trade*. Springer, 1998, pp. 239–274.

[12] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[14] P. H., M. E. P. D., K. Y., and M. G., "Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity," in *International Society for Music Information Retrieval Conference*, 2013.

[15] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *International Society for Music Information Retrieval*, 2017.

[16] A. Van D. O., S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *International Society for Music Information Retrieval*, 2014.

[17] J. Wolfe, M. Garnier, and J. Smith, "Vocal tract resonances in speech, singing, and playing musical instruments," *HFSP Journal*, vol. 3, no. 1, pp. 6–23, 2009.

[18] A. L. Berenzweig and D. P. W. Ellis, "Locating singing voice segments within music signals," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pp. 119–122.

[19] T. Zhang, "Automatic singer identification," in *International Conference on Multimedia and Expo. ICME'03. Proceedings*, vol. 1, 2003, pp. 1–33.

[20] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proceedings of The 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 2003, pp. 282–289.

[21] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017, pp. 933–941.

[22] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The 3rd International Conference for Learning Representations, San Diego*, 2015.

[24] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[25] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[26] K. Lee, K. Choi, and J. Nam, "Revisiting singing voice detection: A quantitative review and the future outlook," *arXiv preprint arXiv:1806.01180*, 2018.

[27] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2744–2748.

[28] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372