Exploring Knowledge Distillation Techniques in Singing Voice Detection

Project-II (EE47008) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Electrical Engineering

by

Soumava Paul

(16EE10056)

Under the supervision of

Prof. K. Sreenivasa Rao

Department of CSE

Indian Institute of Technology Kharagpur

Spring Semester, 2019-20

June 10, 2020

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: June 10, 2020
Place: Kharagpur

Soumava Paul
16EE10056

DEPARTMENT OF CSE

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled “**Exploring Knowledge Distillation Techniques in Singing Voice Detection**” submitted by **Soumava Paul** (Roll No. 16EE10056) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Electrical Engineering is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2019-20.

Date: June 10, 2020
Place: Kharagpur

Prof. K. Sreenivasa Rao
Department of CSE
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Singing Voice Detection (SVD) has been an active area of research in Music Information Retrieval. Currently, two deep neural network-based methods, one based on CNN and the other on RNN, exist in literature that learn optimized features for the VD task and achieve state-of-the-art performance on common datasets. Both these models have a huge of parameters (1.4M for CNN and 65.7K for RNN) and hence not suitable for deployment on devices like smartphones or embedded sensors with limited capacity in terms of memory and computation power. The most popular method to address this issue is known as knowledge distillation in deep learning literature (in addition to model compression) where a large pretrained network known as the teacher is used to train a smaller student network. However, to the best of our knowledge, such methods have not been explored yet in the domain of SVD. In this thesis, efforts have been made to investigate this issue using both conventional as well as ensemble knowledge distillation techniques. Through extensive experimentation on the publicly available Jamendo dataset, this project shows that, not only it's possible to achieve comparable accuracies with far smaller models (upto 1000x smaller in terms of parameters), but fascinatingly, in some cases, smaller models trained with distillation, even surpass the current state-of-the-art models on voice detection performance.

Keywords: Singing Voice Detection, Accelerating CNNs/RNNs, Knowledge Distillation, Soft Labels, Ensemble Knowledge Distillation.

Acknowledgements

I wish to express my sincere thanks to Prof. K. Sreenivasa Rao for his supervision on this project, his suggestions and guidance. I would also like to thank Prof. Jiaul Paik for providing me with all the necessary facilities for this research.

I am grateful to M. Gurunath Reddy, my PhD advisor for this project, for his continuous assessment, inputs, sincere advice and encouragement. Because of him, I got the opportunity to explore an important area of research in Music Information Retrieval.

Contents

Chapter 1

Introduction & Related Work

Singing Voice Detection is a binary classification problem in Music Information Retrieval (MIR), where the task is to identify singing voice in an audio segment of duration 100 to 200 ms. Efficient Voice Detection Systems can also aid in other MIR tasks such as melody extraction[Hsu et al., 2009] or artist recognition[Berenzweig et al., 2002]. Early voice detection approaches [Lehner et al., 2014, Ramona et al., 2008] usually relied on complex hand-engineered audio features, which have now gone out of favour with the advent of deep neural networks. Some examples of such features include MFCCs (MelFrequency Cepstral Coefficients), PLPs (Perceptual Linear Predictive Coefficients) and LFPCs (Log Frequency Power Coefficients). These were used as inputs to classification systems like Support Vector Machines, Hidden Markov Models, Random Forests or Artificial Neural Networks. Two of the most popular and recent DNN based approaches [Leglaive et al., 2015, Schlüter and Grill] show that neural networks have the capacity to learn complex features relevant to a particular task like SVD, from very low-level audio representations such as STFT or Melspectrograms. Lee et al. [2018] performed evaluation of these 2 algorithms under a common benchmark protocol so that pre-processing procedures do not give any of the 2 models an undue advantage in terms of performance. The source code released by Lee et al. [2018] is the starting point of our work.

While these models have considerably improved voice detection performance as well as eliminated reliance on hand-crafted features, little attention has been paid to the optimal structure of these models - whether we actually need the huge memory and computation power needed to train these models. Also, from a deployment point of view, these networks are all the more non-optimal owing to their sufficiently large inference times. In this thesis, we try to address this issue using the basic principles of knowledge distillation [Hinton et al., 2015]. Our experiments show that the optimal CNN-based model is actually more than **250 times smaller** than the model used by Schlüter and Grill, and the optimal RNN-based model is at least **2.5 times smaller** than the one used by Leglaive et al. [2015]. Moreover, we show that, with knowledge distillation on smaller student models, it's possible to obtain accuracies substantially higher than the current state-of-the-art.

Chapter 2

Proposed Methods

2.1 Basics of Knowledge Distillation

The basic idea behind knowledge distillation is to have the student network trained with not only the information provided by true labels (also called hard targets) but also by using soft targets produced by the teacher network (also referred to as cumbersome model) as a regularizer. This helps the student network learn to mimic the teacher's behaviour. In accordance with Hinton et al. [2015], a scaling hyperparameter referred to as temperature is used to regulate the softness of the targets from the teacher network as shown in figure 2.1.

The generalized formula for Softmax is then given by:

$$p_i = \frac{\exp\left(s_i/\tau\right)}{\sum_j \exp\left(s_j/\tau\right)} \tag{1}$$

where s_i and p_i are the logit produced for the i^{th} class and the corresponding class probability respectively and τ is the softness regulating temperature.

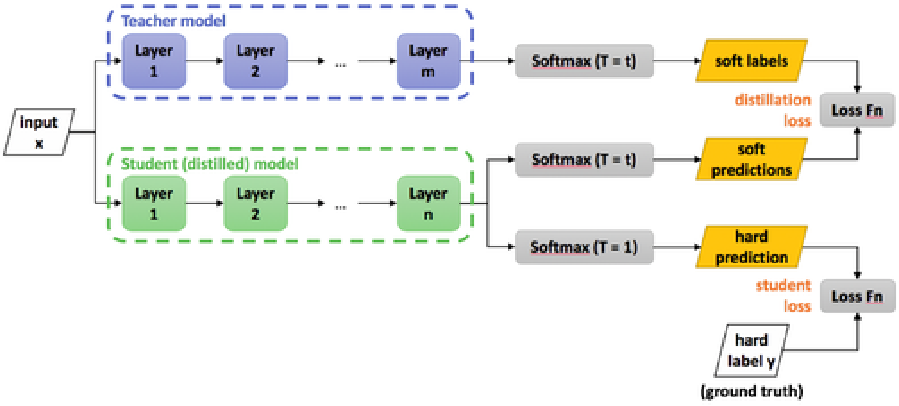


Figure 2.1: Schematic diagram of Knowledge Distillation

A KL-divergence loss is taken between the softmax probabilities of the teacher and the student, represented as q and p respectively, raised to temperature τ (between 2-20 in our experiments) and is given by:

$$L_{KD} = \tau^2 KLD\left(q, p\right) \tag{2}$$

The combined loss is given by:.

$$L_{\text{Total}} = (1 - \lambda)L_{CE} + \lambda L_{KD} \tag{3}$$

where L_{CE} is the cross-entropy loss between the student's softmax probabilities (at $\tau = 1$) and the correct labels (hard targets) and λ is a second hyperparameter controlling the trade-off between the 2 losses. Generally, better results are obtained by keeping λ close to 1, i.e., weight on L_{KD} higher.

2.2 Knowledge Distillation with Schluter CNN

Before starting the discussion specific to knowledge distillation, we first describe the model architecture and feature inputs in Schluter CNN [Schlüter and Grill]:

2.2.1 Feature Inputs

[Schlüter and Grill] use melspectrograms (80-D) of 115 consecutive frames (1.6 seconds) of an audio signal as input feature. The label of the central frame is considered as the label of a particular sample. The resulting input dimension is 80x115. The Mel bank is normalized to have zero mean and unit variance over the train data.

2.2.2 Model Architecture

The Schluter CNN employs three types of feedforward neural network layers: 2D convolutional layers with 3x3 kernels and Leaky ReLU activations (Xu et al. [2015]), maxpooling layers with 3x3 kernels and dense layers. To describe the architecture, we use the following shorthand notations: 'ConvX' denotes a conv layer with X channels and Leaky ReLU activation, 'Max' stands for maxpool, and 'DenseX' denotes a dense layer with X neurons. The architecture can then be described as Conv64-Conv32-Max-Conv128-Conv64-Max-Dense256-Dense64-Dense2. Additionally, there are two dropout layers with dropout probability 0.2 in between the final 3 dense layers. Note that the original model was trained with a singular output dimension and binary cross entropy loss where appropriate thresholds were used on the output probability (between 0 and 1) to infer the presence or absence of a voice label. However, since the main objective of this thesis is knowledge distillation, we needed a softmax activation on the output. Hence, in our modified model, the output dimension is 2 with a softmax activation and the model is trained with categorical cross-entropy loss. In Table 3.1, we compare the performance of our base model with the benchmarks by Lee et al. [2018] and show that the two sets of accuracies are almost identical. This justifies the above mentioned modification to Schluter CNN.

2.2.3 Building Student Networks from Base Model

The base model has a total of 14,08,290 (**1.4M**) parameters. To build student models, we focus on reducing the number of channels per layer (except maxpool of course). The lowest number of channels in any particular layer in the architecture mentioned above is 32. We define a variable ”filter scale” (abbreviated as **FS**) that can have any of the values in {2, 4, 8, 16, 32}. As the name suggests, we divide the number of channels in each layer by FS to reduce the overall number of model parameters. Higher the value of FS, lower the number of parameters. We denote each resulting student network as **FSX**. Student networks trained with distillation are denoted as **KD-FSX**. These notations are later used in Section 3.3. As shown in Table 2.1, the highest reduction achieved with student networks in terms of parameters is nearly **1000x** with FS32.

Table 2.1: Number of parameters of Student Models

Model	Parameters
FS2	3,52,402
FS4	88,266
FS8	22,150
FS16	5,580
FS32	1417

2.3 Knowledge Distillation with Leglaive RNN

Similar to the CNN case, we first describe the model architecture and feature inputs in Leglaive RNN [Leglaive et al., 2015]:

2.3.1 Feature Inputs

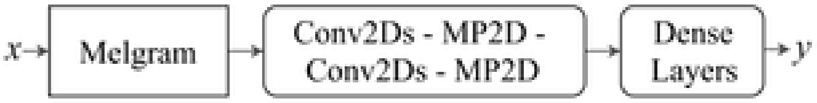
As show in Figure 2.2, Leglaive et al. [2015] first apply a double stage harmonic-percussion source separation (HPSS) [Ono et al., 2008] on the audio signal as pre-processing, to extract signals relevant to the singing voice. The main idea behind HPSS is to decompose the spectrogram of the input signal into one spectrogram smooth in time direction (associated to harmonic components), and another spectrogram smooth in frequency direction (associated to percussive components). Features are then extracted from a filter bank on a Mel scale and the resulting melspectrograms of the harmonic and percussive components are concatenated as input to the BiLSTM network described below. This gives 80-dimensional feature vectors for 218 frames (3.5 seconds) per sample. Additionally and similar to Schluter CNN, the mel bands are normalized to have zero mean and unit standard deviation over the train set. However, unlike Schluter CNN, Leglaive RNN infers the label of each of the 218 time frames.

2.3.2 Model Architecture

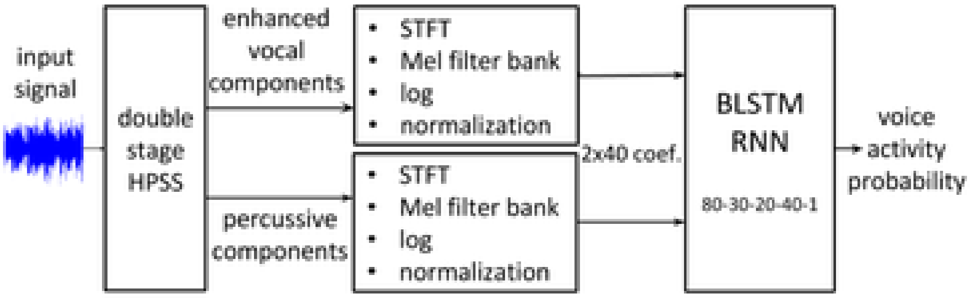
The model is made up of 3 layers of stacked Bidirectional LSTM (BiLSTM) units, of sizes 30, 20, and 40. Finally, this is followed by a shared dense layer (also known as time distributed dense) to yield predictions for the 218 output time frames (3.5 seconds). Similar to 2.2.2, the original model was trained with a singular output dimension for each of the 218 frames and binary cross entropy loss. Keeping in mind our main objective of experimentation with distilled student models, we re-implement Leglaive RNN with 2 output dimensions with softmax activation, for each frame. Comparison with Lee et al. [2018]'s implementation is shown in Table 3.4. Almost identical numbers justify this modification to the teacher network.

2.3.3 Building Student Network from Base Model

Leglaive RNN has a total of 65,682 (**65.7K**) parameters. Reducing only the number of LSTM units won't have too much of an effect in terms of parameter count reduction. So, we completely remove the last 2 BiLSTM layers and that gives close to **2.5x** reduction in size for the student model with a total of 26, 762 (**26.8K**) parameters. For future reference, the teacher is abbreviated as **LRNN** (Leglaive RNN) and the student as **SRNN** (Small RNN). Student network trained with distillation is denoted as **KD-SRNN**.



(a) Proposed Approach in Schluter CNN



(b) Proposed Approach in Leglaive RNN

Figure 2.2: Figures taken from Lee et al. [2018] and Leglaive et al. [2015] respectively.

2.4 Ensemble Knowledge Distillation

In this section, we propose to train both CNN and RNN based student networks using 2 teachers, the 2 models discussed above. For this we pick the feature inputs used by [Schlüter and Grill]. However, for these features, we do not have any prior RNN based models in literature. To get around this problem, we train the Leglaive RNN architecture itself on this data, the only difference being, the RNN now predicts labels of the central time frame as in the case of Schluter CNN. The parameter count remains unchanged from 2.3.2. We first benchmark the performance of LRNN and SRNN on the new data in Table 3.6. The input dimension of Schluter CNN is same as before (80x115); that of Leglaive RNN is 115x80 (time dimension comes first in RNNs). Similar to the proposition in Hinton et al. [2015], we combine the predictions of the teachers in the ensemble (here 2) by taking the arithmetic or geometric mean of their predictions as soft targets. The CNN based students are denoted as **ENKD-FSX** and the RNN based one as **ENKD-SRNN**. These notations are reused in Section 3.6. As shown in Table 3.7, we achieve **state-of-the-art performance** on the Jamendo dataset with this formulation.

Chapter 3

Experiments

3.1 Dataset and Protocol

For our experiments, we use the Jamendo dataset, a publicly available dataset with singing voice annotations, that was also used by Schlüter and Grill and Leglaive et al. [2015]. It contains 93 copyright-free songs, collected and annotated by [Ramona et al., 2008]. For comparison to existing results [Lee et al., 2018], we follow the official split of 61 files for training and 16 files each for validation and testing. We use the validation set for hyperparameter tuning and report test results corresponding to the highest validation accuracy. In accordance with Lee et al. [2018], for test set, we report accuracy, precision, recall, F-measure, False Positive Rate (FPR) and False Negative Rate (FNR). Our source code is built on top of this public repository by Lee et al. [2018].

3.2 Reproducing Results of Schluter CNN

Here we compare the results of our implementation (output dimension 2 and trained by categorical cross-entropy loss) with the implementation by Lee et al. [2018]. Note that, similar to Lee et al. [2018], we also do not apply data augmentation for a fair comparison with Leglaive et al. [2015]. As shown in Table 3.1 below, our numbers only differ by small magnitudes.

Model	Acc.	Precision	Recall	F-Measure	FPR	FNR
Original Implementation	**86.8**	**83.7**	89.1	**86.3**	**15.1**	10.9
Ours	85.4	81.3	**89.3**	85.1	17.9	**10.7**

Table 3.1: Comparison of our Schluter CNN implementation with Lee et al. [2018]

3.3 Knowledge Distillation with Schluter CNN

3.3.1 Performance of Student Networks without Distillation

Here we show the performance of the student networks from Table 2.1. These are trained similar to the teacher network. As shown in Table 3.2 below, with our smaller student networks, we get better results across **all performance measures**, with respect to both ours and the original implementation in Table 3.1. Interestingly enough, **FS32**, having **1000x** lesser parameters than the teacher, succeeds in improving on 2 performance measures, Precision and FPR, over Schluter CNN. This clearly indicates, that smaller models can be equally effective in SVD and require further attention.

Model	Acc.	Precision	Recall	F-Measure	FPR	FNR
FS2	**87.0**	81.8	**92.8**	**86.9**	18.0	**7.2**
FS4	85.2	80.4	90.1	85.0	19.1	9.9
FS8	**87.0**	84.0	89.1	86.5	14.8	10.9
FS16	86.5	84.5	86.6	85.7	13.5	13.4
FS32	83.7	**85.8**	77.9	81.7	**11.2**	22.1

Table 3.2: Performance of Student Derivatives from Schluter CNN

3.3.2 Performance of Distilled Student Networks

Here we present the performance of the student networks from 3.3.1, trained with distillation by the teacher network. As shown in Table 3.3 below, performance **falls** on almost every measure with the student models. This is expected since, as per Table 3.2, student networks already have a lesser tendency to overfit, i.e., they actually perform better than Schluter CNN. So, additional supervision from the teacher actually ends up hurting the student network. However, as we show in Section 3.6, performance of **all FSX models** improve with additional supervision from an LSTM-based[Hochreiter and Schmidhuber, 1997] teacher network.

Model	Acc.	Precision	Recall	F-Measure	FPR	FNR
KD-FS2	87.0	83.7	89.4	86.4	15.2	10.6
KD-FS4	**87.3**	**84.5**	88.3	**86.6**	**13.6**	11.7
KD-FS8	86.6	83.1	89.5	86.2	15.9	10.5
KD-FS16	85.0	80.0	**90.4**	84.9	19.7	**9.6**
KD-FS32	81.5	79.4	81.3	80.3	18.4	18.7

Table 3.3: Performance of Student Models with Knowledge Distillation

3.4 Reproducing Results of Leglaive RNN

Similar to Section 3.2, we compare our implementation of Leglaive RNN with that of Lee et al. [2018]. Here the output dimension is 2 for the time distributed dense layer and the model is trained by categorical cross-entropy loss. As shown in Table 3.4 below, our numbers only differ by small magnitudes.

Model	Acc.	Precision	Recall	F-Measure	FPR	FNR
Original Implementation	87.5	**86.1**	87.2	86.6	**12.2**	12.8
LRNN (**Ours**)	**88.2**	85.7	**89.7**	**87.8**	13.2	**10.3**

Table 3.4: Comparison of our Leglaive RNN implementation with Lee et al. [2018]

3.5 Knowledge Distillation with Leglaive RNN

3.5.1 Performance of Student Network with and without Distillation

As shown in Table 3.5 below. KD-SRNN comprehensively outperforms SRNN on all performance measures, showing the benefits of knowledge distillation in this context. Although SRNN performs slightly worse than LRNN on some of the measures, KD-SRNN surpasses LRNN (our implementation) on all measures and is inferior to the original implementation [Lee et al., 2018], only on Precision and FPR.

Model	Acc.	Precision	Recall	F-Measure	FPR	FNR
SRNN	87.6	84.5	90.6	87.5	14.6	9.4
KD-SRNN	**88.9**	**85.7**	**91.5**	**88.5**	**13.4**	**8.5**

Table 3.5: Performance Comparison of **SRNN** and **KD-SRNN**

3.6 Ensemble Knowledge Distillation

3.6.1 Benchmarking LSTM Performance

First we show the performance of LRNN and SRNN on the data used by Schluter CNN in Table 3.6 below. We see that its performance is considerably lower than Schluter CNN or any of the **FSX** derivatives. Despite this, including it as a teacher in the ensemble considerably improves the performance of all **ENKD-FSX** and **ENKD-SRNN** models, as shown in Table 3.7 next.

Model	Acc.	Precision	Recall	F-Measure	FPR	FNR
LRNN	**82.3**	**78.2**	**85.8**	**81.8**	**20.9**	**14.2**
SRNN	76.1	70.5	83.8	76.6	30.6	16.2

Table 3.6: Performance of LRNN and SRNN on data used by Schlüter and Grill

3.6.2 ENKD Models

Here we present the performance of all CNN and RNN based student networks trained by Ensemble Knowledge Distillation. AM (Arithmetic Mean) and GM (Geometric Mean) denote method by which teacher predictions were combined.

	Acc.		Precision		Recall		F-Measure		FPR		FNR	
Model	AM	GM	AM	GM	AM	GM	AM	GM	AM	GM	AM	GM
ENKD-FS2	87.9	87.7	86.3	86.5	87.9	87.1	87.1	86.8	12.1	11.8	12.1	12.9
ENKD-FS4	86.8	**88.4**	84.4	**86.8**	88.0	88.4	86.1	**87.6**	14.2	**11.7**	12.0	11.6
ENKD-FS8	87.7	87.1	86.2	81.9	87.6	**92.8**	86.9	87.0	12.2	17.8	12.4	**7.2**
ENKD-FS16	86.4	86.5	84.2	83.4	87.1	88.1	85.6	85.8	14.3	15.0	12.9	11.9
ENKD-FS32	84.3	84.2	82.3	82.8	84.4	83.3	83.3	83.1	15.8	15.0	15.6	16.7
ENKD-SRNN	84.4	83.1	81.1	81.5	86.5	82.5	83.7	82.0	17.5	16.3	13.5	17.5

Table 3.7: Performance Comparison of ENKD Variants

3.7 Summary of Results

- Smaller student networks built from Schluter CNN actually show lesser overfitting tendencies and hence perform better than the base model in several cases. That is why knowledge distillation experiments with the base model as a teacher, fail to get any substantial improvement over the student networks' performance.

- Knowledge Distillation experiments with Leglaive RNN produce new state-of-the-art results on the features used by Leglaive et al. [2015], with a smaller student model, **SRNN**.

- With Ensemble Knowledge Distillation, we get a new state-of-the-art model on the features used by Schlüter and Grill, **ENKD-FS4** (mainly based on accuracy) that improves current best reported results (Table 3.1) by **1.6%**. This technique also proves quite beneficial for SRNN, improving its accuracy by **8.3%** to **84.4%** (higher than even **LRNN**). However, this accuracy is still lower than that of **ENKD-FS4** by a fair amount, proving that CNN is still the better algorithm for this kind of data.

Chapter 4

Conclusion and Future Work

In this project, we have shown that application of knowledge distillation techniques on Singing Voice Detection can be a new direction of research in the field of Music Information Retrieval. Our experiments show that smaller models trained with distillation can achieve comparable and in some cases, even higher accuracies than current state-of-the-art models. For future work, we plan to extend our methods to other publicly available SVD datasets like MIR-1K [Hsu and Jang, 2009], RWC Popular Music Database [Goto et al.] and MedleyDB [Bittner et al., 2014]. With respect to knowledge distillation, another interesting experiment can be training student models on a subset of the training data and comparing their performance with student/teacher models that learn from the entire data. We also plan to explore multi-step knowledge distillation techniques in the future, like the Teacher Assistant model (Mirzadeh et al. [2019]), where an intermediate-sized network can supposedly help bridge the knowledge gap between student and teacher better and hence aid in more efficient distillation.

Bibliography

Adam Berenzweig, Daniel PW Ellis, and Steve Lawrence. Using voice segments to improve artist classification of music. 2002.

Rachel M. Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. Medleydb: A multitrack dataset for annotation-intensive mir research. In *ISMIR*, 2014.

Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical and jazz music databases.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the mir-1k dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2009.

Chao-Ling Hsu, Liang-Yu Chen, Jyh-Shing Roger Jang, and Hsing-Ji Li. Singing pitch extraction from monaural polyphonic songs by contextual audio modeling and singing harmonic enhancement. In *ISMIR*, 2009.

Kyungyun Lee, Keunwoo Choi, and Juhan Nam. Revisiting singing voice detection: A quantitative review and the future outlook. *arXiv preprint arXiv:1806.01180*, 2018.

Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125. IEEE, 2015.

Bernhard Lehner, Reinhard Sonnleitner, and Gerhard Widmer. Towards light-weight, real-time-capable singing voice detection.

Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484. IEEE, 2014.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019.

Nobutaka Ono, Kenichi Miyamoto, Jonathan Le Roux, Hirokazu Kameoka, and Shigeki Sagayama. Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. In *2008 16th European Signal Processing Conference*, pages 1–4. IEEE, 2008.

Mathieu Ramona, Gaël Richard, and Bertrand David. Vocal detection in music with support vector machines. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1885–1888. IEEE, 2008.

Jan Schlüter and Thomas Grill. Exploring data augmentation for improved singing voice detection with neural networks.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.