Neural Computing and Applications (2020) 32:1037–1050
https://doi.org/10.1007/s00521-018-3933-z

DEEP LEARNING FOR MUSIC AND AUDIO

Singing voice separation using a deep convolutional neural network trained by ideal binary mask and cross entropy

Kin Wah Edward Lin[1] · B. T. Balamurali[1] · Enyan Koh[1] · Simon Lui[1] · Dorien Herremans[1,2]

Received: 16 December 2017 / Accepted: 3 December 2018 / Published online: 13 December 2018

Abstract
Separating a singing voice from its music accompaniment remains an important challenge in the field of music information retrieval. We present a unique neural network approach inspired by a technique that has revolutionized the field of vision: pixel-wise image classification, which we combine with cross entropy loss and pretraining of the CNN as an autoencoder on singing voice spectrograms. The pixel-wise classification technique directly estimates the sound source label for each time–frequency (T–F) bin in our spectrogram image, thus eliminating common pre- and postprocessing tasks. The proposed network is trained by using the Ideal Binary Mask (IBM) as the target output label. The IBM identifies the dominant sound source in each T–F bin of the magnitude spectrogram of a mixture signal, by considering each T–F bin as a pixel with a multi-label (for each sound source). Cross entropy is used as the training objective, so as to minimize the average probability error between the target and predicted label for each pixel. By treating the singing voice separation problem as a pixel-wise classification task, we additionally eliminate one of the commonly used, yet not easy to comprehend, post-processing steps: the Wiener filter postprocessing. The proposed CNN outperforms the first runner up in the Music Information Retrieval Evaluation eXchange (MIREX) 2016 and the winner of MIREX 2014 with a gain of 2.2702–5.9563 dB global normalized source to distortion ratio when applied to the iKala dataset. An experiment with the DSD100 dataset on the full-tracks song evaluation task also shows that our model is able to compete with cutting-edge singing voice separation systems which use multi-channel modeling, data augmentation, and model blending.

Keywords Singing voice separation · Convolutional neural network · Ideal binary mask · Cross entropy · Pixel-wise image classification

1 Introduction

Humans have an exceptional ability to separate different sounds from a musical signal [3]. For instance, some musicians can distinguish the guitar part from a song and transcribe it, and most non-musician listeners are able to hear and sing along to lyrics of a song. Machines, however, have not yet mastered the ability to separate voices in music, despite the steep increase in the amount of research on artificial intelligence and music over the past few years [8, 19, 28, 48, 50, 66]. In this paper, we focus on the task of singing voice separation from a polyphonic musical piece, i.e., the automatic separation of a musical piece into two music signals: the singing voice and its music accompaniment. Some singing voice separation (SVS) systems [48, 52, 65, 66] take this one step further by separating the music accompaniment into different types of musical instruments. In this research, we focus on the first task of

This work is supported by the MOE Academic fund AFD 05/15 SL and SUTD SRG ISTD 2017 129.

✉ Kin Wah Edward Lin
edward_lin@mymail.sutd.edu.sg

B. T. Balamurali
balamurali_bt@sutd.edu.sg

Enyan Koh
enyan_koh@mymail.sutd.edu.sg

Simon Lui
simon_lui@sutd.edu.sg

Dorien Herremans
dorien_herremans@sutd.edu.sg

[1] Singapore University of Technology and Design, Singapore, Singapore

[2] Institute of High Performance Computing, A*STAR, Singapore, Singapore

separating the singing voice from its music accompaniment. The potential applications of automatic singing voice separation are plentiful and include melody extraction/annotation [12, 56], singing skill evaluation [35], automatic lyrics recognition [46], automatic lyrics alignment [71], singer identification [37] and singing style visualization [34]. These applications are not only useful for researchers in the field of music information retrieval (MIR), but extend to commercial applications such as music for karaoke systems [71].

We propose a novel convolutional neural network (CNN) approach for extracting a singing voice from its musical accompaniment. The key innovations in this design are the inclusion of Ideal Binary Mask (IBM) [70] as the target label and the use of cross entropy [47] as the training objective. This particular combination of IBM with cross entropy loss has proven to be extremely effective for image classification [49]. In the case of singing voice separation, the IBM represents a binary time $\times$ frequency matrix, whereby a '1' indicates that the target energy is larger than the interference energy within the corresponding time–frequency (T–F) bin and '0' indicates otherwise. The training is guided by cross entropy, i.e., the average of the probability error between the predicted and the target label for each T–F bin. Additionally, we pretrain the weights of the CNN by training it as an autoencoder using singing voice spectrograms. The proposed network design enables us to leverage the power of CNNs for pixel-wise image classification, i.e., classifying each individual pixel of an image [32, 42]. This is done performing multiclass classification (one class per sound source) for each T–F bin in our spectrogram, thus directly estimating the soft mask. This allows us to eliminate one of the very commonly used postprocessing step, the Wiener filter [12, 13, 22, 48, 52, 65, 66] (see Sect. 2).

We set up an experiment to test the proposed system with state-of-the-art models for SVS. When training our model on the iKala dataset [5], we achieve 2.2702–5.9563 dB Global normalized source-to-distortion ratio (GNSDR) gain when compared to two state-of-the-art SVS systems [6, 26]. A second experiment, on the full-track songs from the DSD100 dataset [41], shows no statistically significant difference between the proposed system and the current state-of-the-art systems. These experimental results suggest the need for a dataset agnostic model, meaning that instead of blindly feeding more data to models (which greatly improves training time), there is a need for efficient and effective models that perform well across different dataset, even with limited data. In the current research, we work toward this goal by using a network architecture that has shown to be effective in the field of image classification, and use a validation procedure during training and postprocessing to ensure that our CNN generalizes better. Furthermore, when designing our novel architecture, we trained and tested the model on two different datasets, such that the final optimized architecture would perform well across these datasets.

In the next section, an overview of the current state-of-the-art in voice separation models is given, followed by a description of our proposed CNN model with a formal definition of IBM and cross entropy. We then describe the details of the experimental setup and the training methodology and present the results. Finally, conclusions regarding our proposed model and future research are offered.

2 Related work

This section presents existing research in the field of singing voice separation. Experienced readers, who are familiar with the basics of the field, may skip to the sixth paragraph of this section for a detailed description of some of the latest state-of-the-art models. For a more comprehensive overview of the research undertaken in the last 50 years in this field, we refer the reader to the overview article [55].

The most popular preprocessing method in the field of singing voice separation involves transforming the time-domain signal into a spectrogram [4, 15, 16, 24, 26, 29, 67, 69]. Given that the value of each time–frequency (T–F) bin in the magnitude spectrogram X is nonnegative, existing research on blind source separation (BSS) typically applies techniques such as Independent Subspace Analysis (ISA) [4] and Nonnegative Matrix Factorization (NMF) [33]. The former, ISA, is a variant of Independent Component Analysis (ICA), which has previously been used to solve the cocktail party problem [7]. Independent Component Analysis is built upon the assumption that the number of mixture observation signals is equal to or greater than the target sources. The ISA variant, however, relaxes this constraint by using the nonnegative spectrogram X. The second technique often used for blind source separation, NMF, decomposes X into two nonnegative matrices L and R. The product of these two matrices approximates X, such that $LR \approx X$, with D being the difference, such that $D = X - LR$. The matrix D is later assumed to have the timbral characteristics of the singing voice.

NMF was the most widely adopted BSS technique in the 2000s [9, 11, 14, 15, 67, 69]. The main difference between the various NMF-based methods is how the objective function is formulated. A typical formulation could be, $\min ||X - LR||^2$ or $\min Div(X||LR)$, where Div is the Kullback–Leibler divergence function. The popularity of

NMF is partly due to the fact that the two matrices (L and R) can easily be interpreted as a set of different types of musical instruments (or different tracks in the music), which we refer to as I. To understand this interpretation, let us first assume the columns of L to be the frequency/tone basis functions l_i and the rows of R to be the time basis functions r_i, where i is one of the musical instrument (or tracks) in the music. The factorized matrices (L and R) can be decomposed as the sum of the outer product of the basis functions, such that $LR = \sum_{i \in I} l_i \times r_i$. Thus, a frequency basis function l_i can be interpreted as the timbre of instrument i. The corresponding set of time basis functions r_i indicate how the sound of instrument i evolves during the music. Additionally, I is sometimes divided into two groups by posing constraints for the set of harmonic or pitched instruments (e.g. piano), $h \in I$, and the set of the percussion instruments (e.g. drum), $p \in I$ [15, 29, 69].

A related technique, Robust Principal Component Analysis (rPCA), has also been applied to source sound separation [38]. It uses an augmented Lagrange multiplier to *exactly*[1] separate X into a low rank matrix and sparse matrix, $X = \sum_{i \in I} l_i \times r_i - D$, was widely adopted since 2012 [24]. The resulting factorized matrix LR is a low rank approximation of X. The use of rPCA in source separation is motivated by the fact that (i) that the basis function of LR approximates the spectrogram of the musical accompaniment component in the mixture signal; and (ii) D is a sparse matrix that closely approximates the spectrogram of the separated singing voice. To better understand this, note that $X \approx LR$ and $X \approx \sum_{i \in I} l_i \times r_i$. If the number of musical instruments $|I|$ is the reduced rank of X, then LR is a low rank approximation of X. Since the singing voice falls in between the harmonic instruments and percussion instruments, it is assumed to be represented by D.

Ikemiya et al. [26] use rPCA to obtain a sparse matrix, which is treated as a vocal time–frequency mask, and a vocal spectrogram. They then estimate the vocal F0 contour in this spectrogram in order to form a harmonic structure mask. By combining these two masks, they are able to better perform singing voice separation. This method, referred to as IIY, is the winner of MIREX 2014.[2] Chan et al. [5] use the annotation of the vocal F0 contour to form a sparsity mask, which they then use as the input for rPCA to obtain a better vocal spectrogram. There exist several other approaches for source separation, such as the use of a similarity matrix [40, 53]. Based on the MIREX 2014 results[2], however, none of them outperform the rPCA-based methods. Hence, rPCA has become the de facto baseline in recent years.

Inspired by the influential work of Krizhevsky et al. [32] on large-scale image classification from natural images, the use of deep learning has recently gained a lot of attention. Most deep-learning-based SVS systems [6, 12, 22, 44, 66] are trained to match the network input (i.e., the magnitude spectrogram of the mixture signal), with the target label (i.e., the ground truth magnitude spectrogram of the target sound source). Given enough training data, neural networks are typically able to estimate good approximations any continuous function [20], in this case, the magnitude spectrogram for each of the sound sources is estimated. These magnitude spectrograms, however, are not yet a good representation of the different sources. Contrary to intuition, these systems require a Wiener filter postprocessing step, in which a soft mask is calculated for the estimated magnitude spectrograms for every target sound source. These masks are then multiplied with the original magnitude spectrogram of the mixture signal to recreate each estimated signal. Using these soft masks typically gives a better separation quality than directly using the network output to synthesize the final signal [66]. This suggests that we should skip the Wiener filter postprocessing and design a network to learn a soft mask directly.

Recent advances in the field of computer vision [42] have greatly advanced image classification techniques by moving away from the image level toward the pixel-level. Pixel-wise classification aims at classifying each individual pixel in an image. The task of classifying each T–F bin of a spectrogram into a vocal or non-vocal component can be considered as a pixel-wise classification problem.

Creating the pixel-wise ground truth for image segmentation typically involves extensive human effort. Luckily, this is not the case in SVS research as we can simply calculate the ground truth mask from a training set which contains the separated signals (see Sect. 3.2). Simpson et al. [59] and Grais et al. [18] perform singing voice separation using IBM as the target label for training a deep feed-forward neural network. In this research, however, we opt to use a convolutional neural network architecture, which has proven to greatly improve the performance of image classification tasks [32, 42]. A similar CNN architecture for SVS, abbreviated in what follows as MC, has been proposed by [6]. This method was the first runner up in the MIREX 2016 competition.[3] The architecture proposed in this research improves the dimensions of the convolutional layer and introduces a cross entropy loss function, which greatly improves performance.

[1] NMF-based methods do not have this strong constraint. After their optimization process, it likely happens that the rank of LR cannot be reduced to $|I|$, or that D is not a sparse matrix.

[2] http://www.music-ir.org/mirex/wiki/2014:Singing_Voice_Separation_Results.

[3] http://www.music-ir.org/mirex/wiki/2016:Singing_Voice_Separation_Results.

Other state-of-the-art alternatives to using a CNN include the use of recurrent neural networks (RNN) [22] and bi-directional long short-term memory (BLSTM) Networks [66]. These networks are designed to capture temporal changes and may therefore not be necessary in a voice separation context.

Jansson et al. [28] were the first to tackle SVS tasks by using a deep convolutional U-net in which the network predicts the soft mask. Their system shows remarkable performance on two datasets, iKala and MedleyDB [2]. It should be noted, however, that while their network was tested on iKala and MedleyDB, it was trained on a gigantic dataset (the equivalent of two months worth of continuous audio) supplied by industry [25]. This is much larger than the iKala and DSD100 training sets used in this research, which contain a total of, respectively, 76 minutes and 216 minutes of audio. The performance of similar U-net architectures [61, 62] trained on these smaller training set (e.g. DSD100) perform much worse than the original model. We can thus conclude that the remarkable performance reported by Jansson et al. [28] is mainly depended on the tremendous large training set, instead of the U-net architecture [25].

In this paper, we explore a CNN-based method with soft mask prediction further improve the state-of-the-art in SVS systems. The next section will describe our proposed system in more detail.

3 CNN network design

In this section, we first describe how the original mixture signal is transformed into a set of spectrogram excerpts, which are used as the input of the proposed CNN model. We then outline the network architecture, along with a formal definition of IBM and cross entropy. Next, we discuss issues related to the implementation and design of the CNN. Finally, an outline is given of how the network output is transformed into two separated signals, the singing voice and music accompaniment.

3.1 Preprocessing

In the preprocessing stage, the actual input for the CNN is created. First, we apply a short-time Fourier transform (STFT) on the mixture signal x to obtain the magnitude spectrogram X and the phase spectrogram pX. For each Fast Fourier transform (FFT) step, we use the Hann windowing function [51] with a window size W of 46.44 ms, a hop size H of 11.61 ms and a 4× zero padding factor. By setting the sampling rate f_S at 22.05 kHz, each FFT step is with size $N = 4096$, $W = 1024$ and $H = 256$. This STFT configuration was chosen based on the authors' previous study on sinusoidal partials tracking [36].

Sinusoidal partials tracking (PT) is a peak-continuation algorithm that links up the spectral peaks into a set of tracks. Each track models a time-varying sinusoid. The tracks are called partials when they represent the deterministic part of the audio signal. In the previous PT study, the average length of a singing voice partial was found to be around 9 continuous frames and the 4× zero padding factor improved the separation quality of the ideal case. Hence we can assume that these settings should allow for enough temporal and spectral cues in order to properly train the CNN. The input of the proposed CNN consists of an image snapshot of X with a shape of (9×2049), which is a spectrogram excerpt of $(9 \times 1000)/22{,}050 = 104.49$ ms and 11.025 kHz.

3.2 Network architecture with ideal binary mask and cross entropy

Table 1 shows the network architecture of the proposed CNN along with the configuration and the corresponding number of trainable parameters and features. We adopt the CNN architecture developed by Schlüter [57] for voice detection. For that task, the network was trained on weakly labeled music.[4] The resulting saliency map, created through guided backpropagation of the CNN, shows the singing voice in the T–F bin level.

In the current research, we use the IBM as the target label instead of weak labels. IBM can be formally defined as follows. Let the $F \times T$ matrix X denote the magnitude spectrogram, whereby F is the number of frequency bins, $F = (\lfloor \frac{N}{2} \rfloor + 1)$ with N as the FFT size, and T is the number of frames. Given the magnitude spectrogram of the voice X_V and of the music accompaniment X_S, the IBM of the singing voice, which is a $F \times T$ matrix B, is calculated as,

$$B[n,t] = \begin{cases} 1, & \text{if } \mathrm{X_V}[\mathrm{n,t}] > \mathrm{X_S}[\mathrm{n,t}] \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $t \in [1, T]$ is the time index and $n \in [1, F]$ is the frequency bin index. The IBM of the music accompaniment is denoted as $\overline{B} = |1 - B|$.

The resulting matrix B forms the target label of the neural network. Together with the network predictions, $Y[n, t]$, formed by the sigmoid output of the final layer, we can calculate the cross entropy over all T–F bins, as:

$$\begin{aligned} C[n,t] = {} & B[n,t] \times -log(Y[n,t]) \\ & + (1 - B[n,t]) \times -log(1 - Y[n,t]) \end{aligned} \quad (2)$$

[4] Each piece of music only has one annotation that indicates whether the music contains vocals or not.

Table 1 Network architecture of the proposed CNN along with the configuration and the corresponding number of trainable parameters and features

Layer	Configuration	Number of trainable parameters
Input	Input size is (9×2049) Num. of features is $(9 \times 2049) = 18{,}441$	N/A
Convolution	32@ (3×12), Stride 1 Zero Pad, ReLU	$(3 \times 12) \times 32 + 32 = 1184$
Convolution	16@ (3×12), Stride 1 Zero Pad, ReLU	$(3 \times 12) \times 32 \times 16 + 16 = 18{,}448$
Max pooling	Non-overlap (1×12) reshapes input size to $(9 \times 12) = 1539$ Num. of features is $(9 \times 171) \times 16 = 24{,}624$	N/A
Convolution	64@(3×12), Stride 1 Zero Pad, ReLU	$(3 \times 12) \times 16 \times 64 + 64 = 36{,}928$
Convolution	32@(3×12), Stride 1 Zero Pad, ReLU	$(3 \times 12) \times 64 \times 32 + 32 = 73,760$
Max-pooling	Non-overlap (1×12) reshapes input size to $(9 \times 15) = 135$ Num. of features is $(9 \times 15) \times 32 = 4{,}320$	N/A
Dropout	with probability 0.5	N/A
Fully connected	2048 Neurons, ReLU	$4{,}320 \times 2{,}048 + 2{,}048 = 8{,}849{,}408$
Dropout	With probability 0.5	N/A
Fully connected	512 Neurons, ReLU	$2{,}048 \times 512 + 512 = 1{,}049{,}088$
Output	18,441 Neurons, Sigmoid Reshape (9×2049) Singing Voice IBM Label to match these Neurons	$512 \times 18{,}441 + 18{,}441 = 9{,}460{,}233$
Objective function	Cross entropy	Total: 19,489,049

The training objective of our proposed network minimizes the cross entropy. This type of objective function performs better then that often used softmax function, as it is tailored to the fact that each T–F bin can have multiple labels. Unlike a pixel in an image whose value is paired with the desired label, the value of a T–F bin in the magnitude spectrogram of a mixture signal is roughly the sum of the T–F bin of the singing voice and its accompaniment.

Alternative training objectives were explored, such as minimum mean square error (MMSE) with both IBM and Ideal Ratio Mask (IRM) [72] as the target label. We found, however, that the MMSE does not decrease much with IRM and IBM and that cross entropy also does not decrease much with IRM. We therefore opted to integrate IBM with a cross entropy training objective.

To improve the network performance, the weights were first initialized with Xavier's initializer [17]. To further improve these initial weights, the CNN trained as an autoencoder using spectrogram excerpts of the ideal singing voice for 300 epochs. These initial weights allow us to train the resulting separation network much more efficiently.

An often used technique to speed up a model's convergence is Batch Normalization (BN) [27]. This technique requires a number of extra parameters and increases the training time for each epoch. When implementing BN in our network, we did not notice an improvement in training time, and most importantly, there was no improvement of the separation quality. We therefore opted not to include BN in the proposed system. Similarly, we also did not find an improvement of separation quality and training time when we used the skip connection method [21] and the method of converting the fully connected layer to a convolutional layer [42]. Hence, both methods were not included in the proposed CNN.

Existing network architectures commonly apply a (3×3) filter in the convolutional layers. Because we applied $4\times$ zero padding factor in the frequency domain during the STFT calculation, we set the convolutional filter size to be (3×12), whereby 3 represents the time and 12 the frequency bin. The time dimension in the pooling layer was not reduced as this can introduce jitter and other artifacts. The frequency dimension in the max pooling layer, however, was reduced. This process is roughly analogous to Mel-frequency calculation, which has been empirically proven to provide useful features for audio classification tasks [43, 45, 64]. The number of features maps in each convolutional layer is halved compared to the

original voice-detection CNN architecture [57], so as to shorten the training time, and most importantly, to avoid degradation of the separation quality. Finally, the dropout [60] settings and ReLU activations [32] are preserved as in the original architecture.

3.3 Postprocessing

The goal of the singing voice separation task is to get two isolated music signals: voice and accompaniment. We therefore need to convert the estimated soft mask by network into two audio signals. In order to do this, the CNN output is first reshaped from $(1 \times 18{,}441)$ to (9×2049) in order to reconstruct the 9 frames. The estimated network output, before postprocessing, is considered to be the *soft mask* of the estimated singing voice spectrogram, meaning that the value for each T–F can range from 0 to 1. This assumption is justified by the fact that IBM was selected as the target label during training and thus used to calculate the cross entropy with sigmoid function. The value of each T–F bin in the soft mask can be interpreted as the probability e that the T–F bin belongs to the singing voice.

To further improve the separation quality, we carry out the following optional refinement using the validation set. For a threshold θ, we set e to zero when $e < \theta$. Based on an experiment using the validation set (see Sect. 4), we set θ to be 0.35 for the iKala dataset and 0.15 for the DSD100 dataset.

The neural network architecture described above takes 9 audio frames as input. In order to estimate a single soft mask M_V for separating the singing voice from an *entire* song, we follow a two step approach inspired by Schlüter [57]. First, overlapping spectrogram excerpts (each 9 frames long) are fed into the network with a hop size of 1 frame. The middle frames of each estimated soft mask is then concatenated to create M_V. These two steps are illustrated in Fig. 1. The soft mask M_S for obtaining the *music accompaniment* from a test song can be calculated by $1 - M_V$.

Finally, the isolated signing voice signal is obtained by calculating the inverse TFT (iSTFT) of the element-wise multiplication between the estimated M_V and X, and the original phase spectrogram pX. Similarly, we can obtain the isolated musical accompaniment signal by calculating the iSTFT of the element-wise multiplication between M_S and X using pX. In the case of a stereo recording, all of the procedures mentioned above should be carried out for each channel separately.

4 Experiment setup

The separation quality of the proposed CNN model is evaluated and compared to other state-of-the-art SVS systems. This is achieved by using two datasets that are specifically designed for the SVS task. Before discussing the results of our experiment in the next section, a brief description of the music clips in each dataset is given, together with how these are divided into development and test sets. We then describe the evaluation procedure and discuss how the proposed CNN should be properly trained, so that a state-of-the-art results can be obtained.

4.1 iKala dataset

The iKala dataset [5] is a public dataset specifically created for the SVS task. Each clip in the dataset is recorded in a CD quality wave file and sampled at 44.1 kHz, with two channels. One channel consists of the ground truth singing voice V, and the other one forms the ground truth music accompaniment S. The mixture signal M is simply the sum of V and S. There are 6 singers, of which three were female and three male. The singing voice tracks were almost entirely performed by one or more of these singers. The musical accompaniment tracks were all performed by professional musicians. Each clip is 30 s long and contains non-vocal regions with varied duration. The language of the lyrics is either English, Mandarin, Korean, or Taiwanese. The dataset contains 352 music clips, 100 of them are reserved for the evaluation of the MIREX[5] singing voice separation task and are not publicly available. Among the remaining 252 clips, 137 of these clips are labeled *Verse* and 115 clips as *Chorus*.

In order to properly evaluate our proposed model, the 252 music clips in the iKala dataset were randomly divided into 3 sets, namely training, validation, and test set. The training set consisted of 152 ($\sim$ 60%) clips, 50 ($\sim$ 20%) music clips form the validation set and 50 ($\sim$ 20%) the test set. The details of each set are described in Table 2.

4.2 Evaluation under iKala dataset

In line with the MIREX2016 evaluation procedures, we use a standard quality assessment tool for evaluating SVS systems called BSS Eval Version 3.0 [68]. For each estimated/original clip, four quality metrics are calculated in order to assess the separation quality, namely source-to-distortion Ratio (SDR), source Image-to-spatial distortion Ratio (ISR), source-to-interferences ratio (SIR), and sources-to-artifacts ratio (SAR). The global separation quality for each clip in terms of singing voice is measured by the normalized SDR (NSDR). This ratio is calculated as

$$\mathrm{NSDR}(\overline{\mathrm{V}}, \mathrm{V}, \mathrm{M}) = \mathrm{SDR}(\overline{\mathrm{V}}, \mathrm{V}) - \mathrm{SDR}(\mathrm{M}, \mathrm{V}) \quad (3)$$

Here, $\overline{V}$ represents the audio signal of the estimated singing voice. The overall singing voice separation quality on a test

[5] http://www.music-ir.org/mirex/wiki/MIREX_HOME.

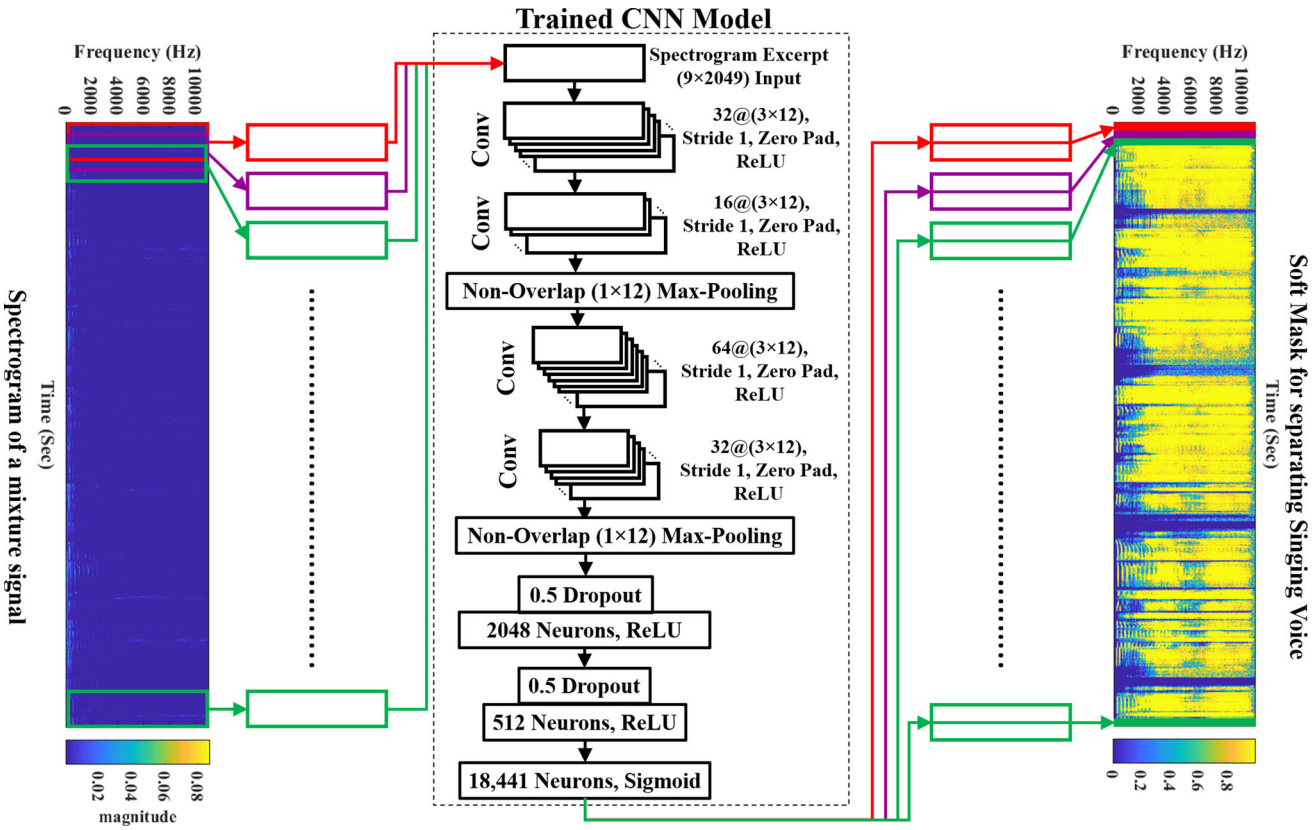


Fig. 1 Architecture for estimating a soft mask based on an entire track

set is determined by the global NSDR (GNSDR). This ratio is calculated as

$$\mathrm{GNSDR} = \frac{1}{|\Lambda|}\sum_{i\in\Lambda}\mathrm{NSDR}(\overline{\mathrm{V}}_i, \mathrm{V}_i, \mathrm{M}_i) \tag{4}$$

whereby Λ is a set of test clips; and the total number of the test clips is represented by $|\Lambda|$. A better separation quality is reflected by a larger GNSDR. Similarly to the quality of the singing voice, the above formula can be modified to calculate the separation quality of the music accompaniment by replacing V by S and $\overline{V}$ by $\overline{S}$ respectively. The GNSDR calculation is computationally expensive; hence, we used parallel processing through a GPU[6] to accelerate this process.

4.3 DSD100 dataset

The DSD100 dataset [41] is a public dataset, specifically created for evaluating source separation algorithms capable of separating professionally produced music recordings into either two stereo signals (i.e., music accompaniment and singing voice), or five stereo signals (i.e., singing voice, music accompaniment, drums, bass and other). There are four wave files for each recording, in addition to the mixed recording wave file: the ground truth singing voice V, drums U, bass A and other O. The ground truth music accompaniment S is simply the sum of U, A and O. The mixture signal M is the sum of V and S. The recordings are all in English, and feature different artists and genres. For example, the genres includes Rap, Rock, Heavy Metal, Pop and Country. The time duration ranges from 2 min and 22 s to 7 min and 20 s, with an average duration of 4 min and 10 s. There are 100 recordings, that are evenly distributed over the development (dev) set and the test set. We used the dev set to create the training and validation set by following the procedures described in Sect. 4.5.

4.4 Evaluation under DSD100 dataset

To enable easy comparison with other algorithms, we follow the evaluation procedure of the SiSEC 2016 MUS track, and use BSS Eval version 3.0 [68] to assess the separation quality of our SVS algorithm. In order to assess the separation quality of whole songs; however, we carry out the procedures below instead.

The stereo mixture signal of each recording is first divided into a set of 30-s-long music clips with 15-s overlap. We then exclude music clips which are smaller than 30 s or yield NaN (Not a Number) SDR values for the

Table 2 The training, validation and test set split based on the iKala dataset

	Music clips		Total clips
	Verse	Chorus	
Training	10174, 21025, 21031, 21032, 21033, 21035, 21038, 21039, 21040, 21054, 21055, 21059, 21060, 21063, 21064, 21069, 21076, 21086, 31081, 31099, 31101, 31104, 31107, 31109, 31113, 31114, 31119, 31134, 31136, 31143, 45305, 45358, 45359, 45362, 45367, 45368, 45378, 45381, 45382, 45386, 45387, 45388, 45389, 45390, 45393, 45398, 45404, 45414, 45415, 45421, 45423, 45428, 45429, 45434, 54173, 54186, 54191, 54192, 54194, 54205, 54223, 54226, 54245, 54246, 61670, 61671, 61673, 61674, 66558, 66564, 66565, 71706, 71710, 71711, 71719, 80612	10171, 10174, 21033, 21035, 21038, 21040, 21054, 21056, 21057, 21059, 21061, 21063, 21068, 21074, 21075, 21083, 21086, 31047, 31075, 31083, 31101, 31103, 31112, 31113, 31115, 31118, 31135, 45305, 45358, 45361, 45363, 45367, 45368, 45369, 45378, 45382, 45384, 45386, 45387, 45392, 45398, 45406, 45413, 45422, 45424, 45425, 45428, 45429, 54189, 54190, 54192, 54202, 54211, 54220, 54221, 54223, 54226, 54233, 54236, 54239, 54243, 54245, 54246, 54249, 61647, 61671, 61676, 61677, 66556, 66557, 71710, 71716, 71719, 71720, 71726, 90586	152
Validation	10161, 10171, 21068, 31092, 31129, 31139, 31142, 45369, 45384, 45400, 45409, 45417, 45422, 45435, 54016, 54189, 54219, 54242, 66559, 66560, 66563, 66566, 71712, 71720, 90586	10170, 21025, 21045, 21073, 21084, 31092, 31100, 31129, 31137, 31143, 45381, 45385, 45389, 45416, 45419, 45435, 54173, 54183, 54210, 54212, 54228, 66559, 66561, 66563, 71711	50
Test	21045, 21058, 21061, 21062, 21071, 21073, 21075, 21084, 31083, 31117, 31132, 31135, 31137, 31144, 45391, 45392, 45410, 45412, 45416, 45418, 45431, 54190, 54213, 54216, 54227, 54233, 54243, 54247, 54249, 54251, 61647, 66556, 71723, 80614, 80616, 90587	10161, 10164, 21058, 31093, 31109, 31116, 31126, 31134, 31139, 45412, 45415, 54194, 54213, 54227	50

The numbers represent the file name of the corresponding wave file

singing voice. The NaN SDR values mostly occur at the beginning and end of the recording, where there is no singing voice.

We refer to the set of 30-s-long clips for a recording r as Λ_r. In order to assess the singing voice separation quality of a SVS algorithm, we first calculate the representative ($\mathrm{SDR_r}$) value of a recording r by averaging the singing voice SDR for each clip i in r, such that $SDR_r = \frac{1}{|\Lambda_r|}\sum_{i\in\Lambda_r} \mathrm{SDR(i)}$. The singing voice separation quality of a SVS algorithm is represented by the median of these $\mathrm{SDR_r}$ over the test set. The separation quality of other sound sources can be calculated similarly.

4.5 Training

The training instances were created by dividing each training song into a set of (9×2049) spectrogram excerpts (one spectragram for each 9 frames) using a hop size of 8 frames (92.88 ms). Since there is an overlap of only 1 frame, the training instances are concise. In the case of stereo recordings, each channel was processed in the same manner, but we chose to alternatingly use the spectrogram excerpts from one or the other channel, in order to have the same number of training instances as for the single channel. This procedure reduces the number of training instance significantly, yet preserve most of the information of each channel. Both datasets are evaluated on the basis of 30-s music clips. Using our network setup, a 30-s music clips equates to $30 \times 1000/92.88 = 323$ input slices. For the iKala dataset, there are 152 clips of 30 s, resulting in $323 \times$

$152 = 49{,}096$ training instances. For the DSD100 dataset, there are 347 clips of each 30 s, resulting in $323 \times 347 = 112{,}081$ training instances. For each clip, we randomly shuffle the training instances for the purpose of regularization. In a similar fashion, validation instances are created using the set of validation songs. They are used for parameter initialization and model selection.

We use the Tensorflow [1] version of the ADAM [31] optimizer with its default values, to train a CNN for each dataset. The network is updated per batch of 171 instances. A BizonBox[6] with NVIDIA GTX TITAN X was used to train both CNNs. Each training epoch needed around 2 min and 6 min for the iKala and DSD100 dataset respectively. For regularization purposes, we used 50% dropout [60] and shuffled the training instances. The target values were set to 0.02 and 0.98 instead of 0 and 1, as suggested by Schlüter [57]. This method prevents overfitting more so than L2 weight regularization.

All trainable parameters in our CNN were initialized with Xavier's initializer [17]. In order to even further improve the set of initial parameters for the SVS task, the CNN is first treated as an *auto-encoder* by pretraining it with spectrogram excerpts of the ideal singing voice for 300 epochs. The model with the lowest cross entropy loss for the validation set is then selected as the initial model for the actual training with the full network. After this parameter initialization, the proposed CNN is trained by feeding it the spectrogram excerpts of the mixture signal and the corresponding singing voice IBM as the target label. Figure 2 shows the evolution of the cross entropy loss for each dataset. Note that we also plot the cross entropy loss of the test set for the sake of completeness. The final model is selected based on the lowest cross entropy loss on the validation set, which is 0.4509 and 0.3625, for the iKala and DSD100 dataset, respectively. The selected model for the iKala and DSD100 dataset are trained with 242 epochs and 280 epochs, respectively, in order to ensure that the validation set has the lowest cost. The separation quality results of these models on the test set are described in the next section.

5 Experimental results

Using the **iKala dataset**, the proposed CNN was compared with the first runner up (MC) of MIREX 2016 [6], the winner (IIY) of MIREX 2014 [26] and the rPCA baseline [24]. A comparison of our model with the winner of MIREX 2016 [44] and MIREX 2015 [12] was not possible, as both winners do not share sufficient information to ensure a fair comparison. For example, they do not share their trained model, information on the training set, nor their separation results for each music clip.[7] The results[8] of our experiment are displayed in Fig. 3. The CNN proposed in this paper achieves the highest GNSDRs for both singing voice and music accompaniment: 9.5774 dB and 9.2484 dB, respectively. For the singing voice, our system achieves 2.2702 dB higher than MC, 5.0908 dB higher than IIY, and 5.9071 dB higher than rPCA. For the music accompaniment voice, the proposed CNN achieves 2.3804 dB higher than MC, 5.9563 dB higher than IIY, and 6.5947 dB higher than rPCA. To further justify that our CNN outperforms the others, we perform a one-way ANOVA, the results of which are summarized in Table 3. The p values confirm that the proposed CNN achieves a statistically significant GNSDR difference (< 0.01) compared to the other systems.

Secondly, the **DSD100 dataset** was used to compare the proposed CNN to the SVS systems that participated in the SiSEC 2016 MUS track.[9] This track included 10 blind source separation methods: CHA [6], DUR [10], KAM [39], OZE [52], RAF [40, 53, 54], HUA [24] and JEO [30], and 14 supervised learning methods, which use different types of deep neural networks, including GRA [18], KON [23], UHL [66], NUG [48], STO [63] and their variants, e.g. UHL1 and UHL2. Given the published details of their separation results,[10] we are able to show the SDR distribution[8] for each SVS algorithm in Fig. 4. Based on the median values in the test set, the proposed CNN ranks 3rd and 8th in term of the separation quality of the singing voice and the music accompaniment, respectively. Its performance is just behind UHL and NUG which use multi-channel modeling [48], data augmentation [66], and model blending [66]. When interpreting these results, one should keep in mind that we only used 1×10^5 training instances to train the CNN (without data augmentation), whereas UHL was trained on 2×10^6 instances. This further illustrates the effectiveness of our network design. The result also shows that our proposed way of proprocessing training instances effectively reduces the size of the required training set. Furthermore, unlike the UHL1 model, our model does not require us to train a model separately for each channel.

To evaluate the significance of the difference in performance, a pairwise two-tailed Wilcoxon signed-rank test with Bonferroni correction [58] was performed. Figure 5

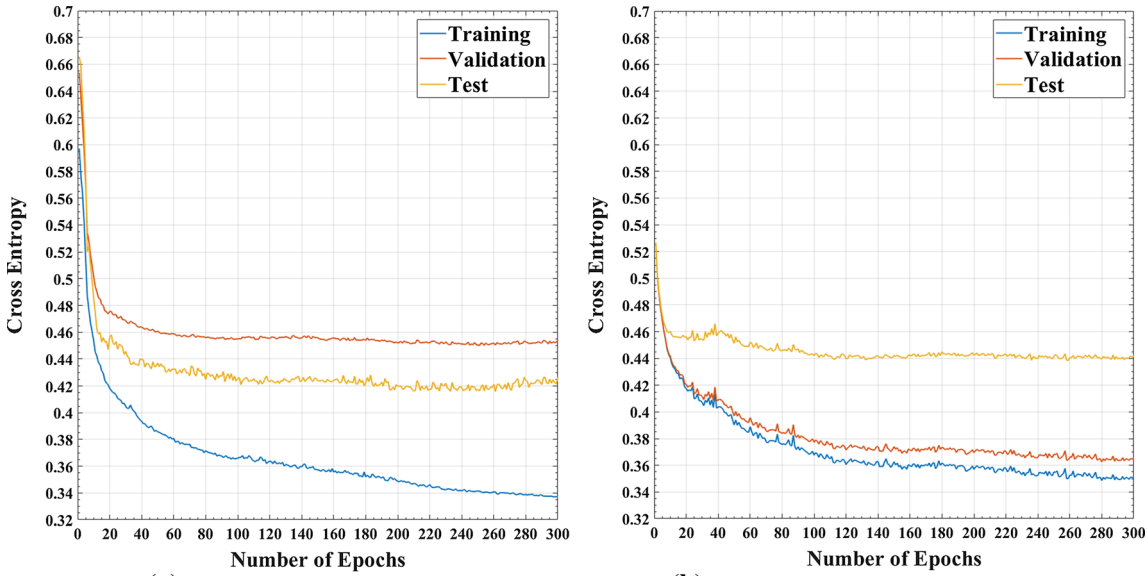
[6] https://bizon-tech.com/.

[7] The 2016 winner [44] has created a web service for others to try their separation method, however, each separated clip is only 10 s long.

[8] Readers who are interested in other evaluation metrics of our CNN model, may refer to https://kinwahedwardlin.wordpress.com.

[9] http://sisec17.audiolabs-erlangen.de/.

[10] https://github.com/faroit/sisec-mus-results.



(a) The loss function for iKala Dataset **(b)** The loss function for DSD100 Dataset

Fig. 2 Evolution of the cross entropy loss for each dataset during training. The lowest cross entropy loss of the validation set is 0.4509 and 0.3625 for the iKala and DSD100 dataset, respectively. The final selected model for the iKala and DSD100 dataset was trained with 242 epochs and 280 epochs, respectively

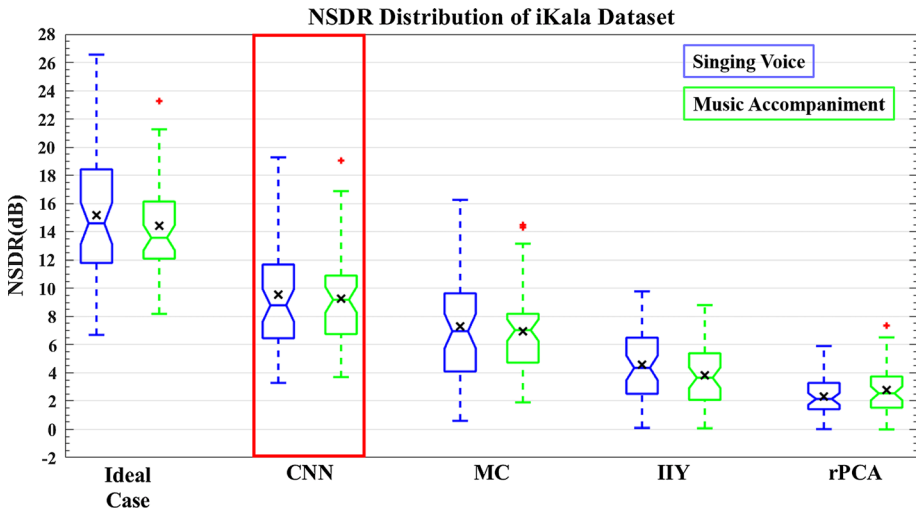


Fig. 3 The NSDRs distribution of each SVS algorithm. The marks x indicate the GNSDRs of each SVS algorithm. The left bar represents the ideal GNSDR: 15.1944 dB for singing voice, and 14.4359 dB for musical accompaniment

Table 3 The significant GNSDR difference between each pair of the SVS systems evaluated by a One-way ANOVA test

Pair	Singing voice		Music accompaniment	
	F (1,98)	p value	F (1,98)	p value
CNN, MC	8.4989	0.0044	9.2806	0.0002
CNN, IIY	57.9684	1.676×10^{-11}	76.0115	9.7516×10^{-16}
CNN, rPCA	59.7874	9.4109×10^{-12}	147.3874	3.0223×10^{-21}
MC, IIY	17.9755	5.0706×10^{-5}	35.8675	3.4918×10^{-8}
MC, rPCA	22.838	6.1939×10^{-6}	66.96450	1.0299×10^{-12}
IIY, rPCA	1.5871	0.2107	1.5620	0.2143

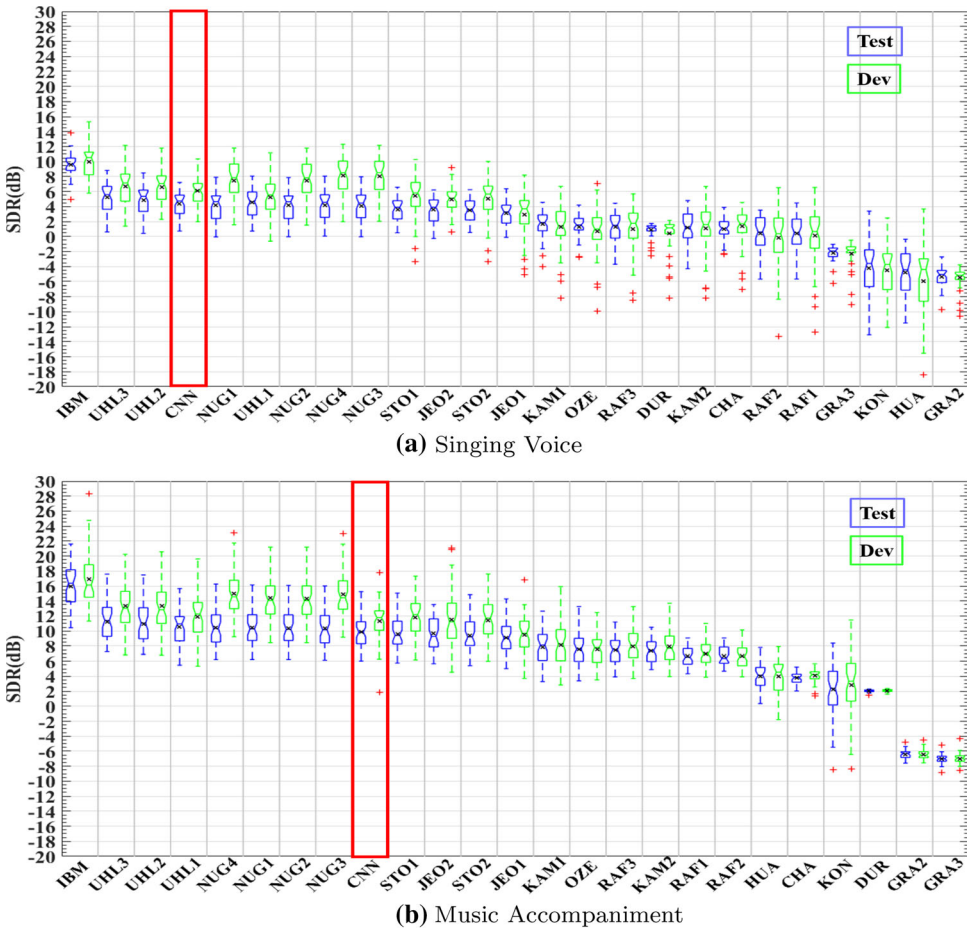


Fig. 4 The SDR distribution for the dev and test set, sorted by the median values of the test set for all SVS algorithms. For the Test set, our CNN achieves 4.7385 dB and 9.8567 dB for the singing voice and its accompaniment, respectively. For Dev set, our CNN achieves 6.1632 dB and 11.7888 dB for the singing voice and its accompaniment respectively

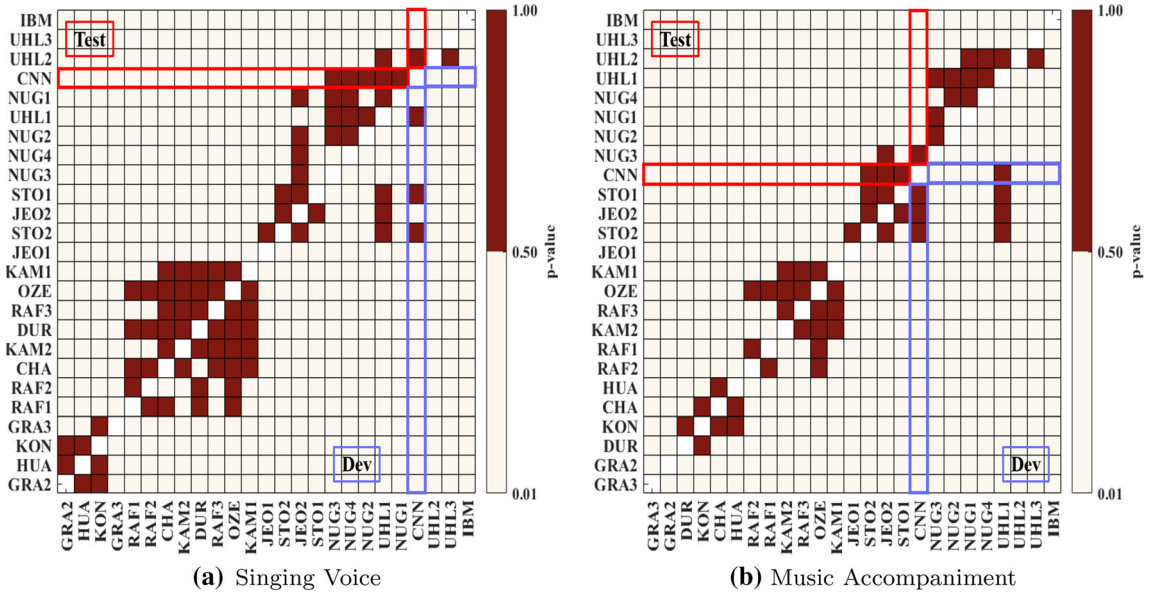


Fig. 5 p values of the pairwise difference of Wilcoxon signed-rank test over different pairs of SVS systems. The upper triangle represents the result of the test set and the lower triangle represents the result of the dev set. Values $p > 0.05$ indicate no significant differences between two SVS systems. Note that the Labels of SVS systems are different in these two sub-figures. They are based on the ranking shown in Fig. 4

summarizes the results. There is no statistical difference, in terms of separation quality of the singing voice, between our CNN, UHL(1,2), and NUG(1-4). This relativizes the importance of Fig. 4. The only significant difference is with UHL3, which uses model blending between UHL1 and UHL2. This results suggest that our CNN might be a suitable candidate for blending with other state-of-the-art systems.

Jansson et al. [28] reported a remarkable performance by using their U-Net architecture trained on a huge industry dataset. We refrained from directly comparing our CNN with the U-net as we are not able to replicate their extraordinary performance when training on the smaller iKala and DSD100 training set. Nevertheless, by looking the empirical results[11] reported by similar U-nets [61, 62], we are confident that our CNN is able to compete with the U-net architecture.

6 Conclusion

A singing voice separation model inspired by recent advances in image processing, e.g. pixel-wise image classification, is presented in this paper. Details of the full design process of this model are given, including preprocessing steps such as how the mixture signal can be transformed to form the model's input. The full architecture of the proposed convolutional neural network is discussed, which includes an Ideal binary mask component as the prediction target label. Our unique network approach includes IBM target labels, cross entropy loss, and pretraining the CNN as an autoencoder on singing voice spectrogram segments.

Computational results based on the iKala and DSD100 dataset show that the proposed system can compete with cutting-edge voice separation systems. On the iKala dataset, our model reaches 2.2702–5.9563 dB Global GNSDR gain over the two best performing algorithms [6, 26]. Second, on the DSD100 dataset, no statistically significant difference was found between the proposed model and current state-of-the-art (non-fused) systems [41]. Audio examples resulting from this paper are available online,[12] together with the spectrogram plots, source code and trained models.

In future research, it would be interesting to further improve the quality of the separated music accompaniment, e.g., by dedicated training on specific instruments in the music accompaniment, and systematically studying the effect of the model's components on the separation quality, such as the choices for the number of feature maps in each layers.

[11] For iKala, the GNSDRs for both singing voice and music accompaniment are 9.50 dB and 6.34 dB, respectively; for DSD100, the SDRs for both singing voice and music accompaniment are 2.83 dB and 6.71 dB, respectively.

[12] https://kinwahedwardlin.wordpress.com/.

Compliance with ethical standards

Conflict of interest The authors of this manuscript certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements) or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: Large-scale machine learning on heterogeneous systems https://www.tensorflow.org/, software available from https://www.tensorflow.org
2. Bittner RM, Salamon J, Tierney M, Mauch M, Cannam C, Bello JP (2014) Medleydb: a multitrack dataset for annotation-intensive mir research. In: International society for music information retrieval conference (ISMIR). pp 155–160
3. Bregman AS (1994) Auditory scene analysis: the perceptual organization of sound. MIT Press, Cambridge
4. Casey M, Westner A (2000) Separation of mixed audio sources by independent subspace analysis. In: International computer music conference (ICMC)
5. Chan T, Yeh T, Fan Z, Chen H, Su L, Yang Y, Jang R (2015) Vocal activity informed singing voice separation with the ikala dataset. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 718–722
6. Chandna P, Miron M, Janer J, Gómez E (2017) Monoaural audio source separation using deep convolutional neural networks. In: International conference on latent variable analysis and signal separation (LVA/ICA),
7. Cherry EC (1953) Some experiments on the recognition of speech, with one and with two ears. J Acoust Soc Am 25(5):975–979
8. Chuan CH, Herremans D (2018) Modeling temporal tonal relations in polyphonic music through deep networks with a novel image-based representation. In: AAAI conference on artificial intelligence (AAAI)
9. Dessein A, Cont A, Lemaitre G (2010) Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In: International society for music information retrieval conference (ISMIR). pp 489–494
10. Durrieu JL, David B, Richard G (2011) A musically motivated mid-level representation for pitch estimation and musical audio source separation. IEEE J Sel Top Signal Process 5(6):1180–1191

11. Eggert J, Korner E (2004) Sparse coding and NMF. IEEE international joint conference on neural networks. vol 4, pp 2529–2533
12. Fan ZC, Jang JSR, Lu CL (2016) Singing voice separation and pitch extraction from monaural polyphonic audio music via DNN and adaptive pitch tracking. In: IEEE international conference on multimedia big data (BigMM)
13. Fan ZC, Lai YL, Jang JSR (2017) Svsgan: singing voice separation via generative adversarial network. In: arXiv:1710.11428
14. Févotte C, Bertin N, Durrieu JL (2009) Nonnegative matrix factorization with the itakura-saito divergence: with application to music analysis. Neural Comput 21(3):793–830
15. FitzGerald D, Gainza M (2010) Single channel vocal separation using median filtering and factorisation techniques. ISAST Trans Electr Signal Process 4(1):62–73
16. Fujihara H, Goto M, Kitahara T, Okuno HG (2010) A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval. IEEE Trans Audio Speech Lang Process 18(3):638–648
17. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: International conference on artificial intelligence and statistics
18. Grais EM, Roma G, Simpson AJR, Plumbley MD (2016) Single-channel audio source separation using deep neural network ensembles. In: Audio engineering society convention 140
19. Herremans D, Chuan CH, Chew E (2017) A functional taxonomy of music generation systems. ACM Comput Surv 50(5):69:1–69:30
20. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4(2):251–257
21. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: The IEEE conference on computer vision and pattern recognition (CVPR)
22. Huang PS, Kim M, Hasegawa-Johnson M, Smaragdis P (2014) Singing-voice separation from monaural recordings using deep recurrent neural networks. In: International society for music information retrieval conference (ISMIR). pp 477–482
23. Huang PS, Kim M, Hasegawa-Johnson M, Smaragdis P (2015) Joint optimization of masks and deep recurrent neural networks for monaural source separation. IEEE/ACM Trans Audio Speech Lang Process 23(12):2136–2147
24. Huang P, Chen S, Smaragdis P, Hasegawa-Johnson M (Mar 2012) Singing-voice separation from monaural recordings using robust principal component analysis. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 57–60
25. Humphrey E, Montecchio N, Bittner R, Jansson A, Jehan T (2017) Mining labeled data from web-scale collections for vocal activity detection in music. In: Proceedings of the 18th ISMIR conference
26. Ikemiya Y, Itoyama K, Yoshii K (2016) Singing voice separation and vocal F0 estimation based on mutual combination of robust principal component analysis and subharmonic summation. IEEE/ACM Trans Audio Speech Lang Process 24(11):2084–2095
27. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning (ICML). pp 448–456
28. Jansson A, Humphrey E, Montecchio N, Bittner R, Kumar A, Weyde T (2017) Singing voice separation with deep u-net convolutional networks. In: International society for music information retrieval conference (ISMIR). pp 745–751
29. Jeong IY, Lee K (2014) Vocal separation from monaural music using temporal/spectral continuity and sparsity constraints. IEEE Signal Process Lett 21(10):1197–1200
30. Jeong IY, Lee K (2017) Singing voice separation using rpca with weighted l_1-norm. In: International conference on latent variable analysis and signal separation (LVA/ICA). Springer, Berlin, pp 553–562
31. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
32. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp 1097–1105
33. Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Advances in neural information processing systems. pp 556–562
34. Lin KWE, Anderson H, Agus N, So C, Lui S (2014a) Visualising singing style under common musical events using pitch-dynamics trajectories and modified traclus clustering. In: International conference on machine learning and applications (ICMLA). pp 237–242
35. Lin KWE, Anderson H, Hamzeen M, Lui S (2014b) Implementation and evaluation of real-time interactive user interface design in self-learning singing pitch training apps. In: Joint proceedings of international computer music conference (ICMC) and sound and music computing conference (SMC)
36. Lin KWE, Anderson H, So C, Lui S (2017) Sinusoidal partials tracking for singing analysis using the heuristic of the minimal frequency and magnitude difference. In: Interspeech. pp 3038–3042
37. Lin KWE, Feng T, Agus N, So C, Lui S (2014c) Modelling mutual information between voiceprint and optimal number of mel-frequency cepstral coefficients in voice discrimination. In: International conference on machine learning and applications (ICMLA). pp 15–20
38. Lin Z, Chen M, Ma Y (2010) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Tech. rep., UILU-ENG-09-2214, UIUC
39. Liutkus A, Fitzgerald D, Rafii Z (2015) Scalable audio separation with light kernel additive modelling. In: IEEE International conference on acoustics, speech and signal processing (ICASSP). pp 76–80
40. Liutkus A, Rafii Z, Badeau R, Pardo B, Richard G (2012) Adaptive filtering for music/voice separation exploiting the repeating musical structure. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 53–56
41. Liutkus A, Stöter FR, Rafii Z, Kitamura D, Rivet B, Ito N, Ono N, Fontecave J (2017) The 2016 signal separation evaluation campaign. In: International conference on latent variable analysis and signal separation (LVA/ICA). Springer, Berlin, pp 323–332
42. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE Conference on computer vision and pattern recognition (CVPR)
43. Loughran R, Walker J, O'Neill M, O'Farrell M (2008) The use of mel-frequency cepstral coefficients in musical instrument identification. In: International computer music conference (ICMC)
44. Luo Y, Chen Z, Hershey JR, Roux JL, Mesgarani N (2017) Deep clustering and conventional networks for music separation: Stronger together. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 61–65
45. Mauch M, Fujihara H, Yoshii K, Goto M (2011) Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In: International society for music information retrieval conference (ISMIR). pp 233–238
46. Mesaros A, Virtanen T (2010) Automatic recognition of lyrics in singing. EURASIP J Audio Speech Music Process 1:546047
47. Nielsen MA (2015) Neural networks and deep learning. Determination Press, New York

48. Nugraha AA, Liutkus A, Vincent E (2016) Multichannel audio source separation with deep neural networks. IEEE/ACM Trans Audio Speech Lang Process 24(9):1652–1664
49. Oh SJ, Benenson R, Khoreva A, Akata Z, Fritz M, Schiele B (2017) Exploiting saliency for object segmentation from image level labels. In: IEEE conference on computer vision and pattern recognition (CVPR). pp 4410–4419
50. den Oord AV, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. In: Advances in neural information processing systems. pp 2643–2651
51. Oppenheim AV, Schafer RW (2009) Discrete-time signal processing, 3rd edn. Prentice Hall Press, Upper Saddle River
52. Ozerov A, Vincent E, Bimbot F (2012) A general flexible framework for the handling of prior information in audio source separation. IEEE Trans Audio Speech Lang Process 20(4):1118–1133
53. Rafii Z, Pardo B (2012) Music/voice separation using the similarity matrix. In: International society for music information retrieval conference (ISMIR). pp 583–588
54. Rafii Z, Pardo B (2013) Repeating pattern extraction technique (repet): a simple method for music/voice separation. IEEE Trans Audio Speech Lang Process 21(1):73–84
55. Rafii Z, Liutkus A, Stoter FR, Mimilakis SI, FitzGerald D, Pardo B (2018) An overview of lead and accompaniment separation in music. IEEE/ACM Trans Audio Speech Lang Process (TASLP) 26(8):1307–1335
56. Salamon J, Bittner R, Bonada J, Bosch JJ, Gómez E, Bello JP (2017) An analysis/synthesis framework for automatic F0 annotation of multitrack datasets. In: International society for music information retrieval conference (ISMIR)
57. Schlüter J (2016) Learning to pinpoint singing voice from weakly labeled examples. In: International society for music information retrieval conference (ISMIR). pp 44–50
58. Simpson AJR, Roma G, Grais EM, Mason RD, Hummersone C, Liutkus A, Plumbley MD (2016) Evaluation of audio source separation models using hypothesis-driven non-parametric statistical methods. In: European signal processing conference (EUSIPCO). pp 1763–1767
59. Simpson AJ, Roma G, Plumbley MD (2015) Deep karaoke: extracting vocals from musical mixtures using a convolutional deep neural network. In: International conference on latent variable analysis and signal separation (LVA/ICA). pp 429–436
60. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
61. Stoller D, Ewert S, Dixon S (2017) Adversarial semi-supervised audio source separation applied to singing voice extraction. arXiv:1711.00048
62. Stoller D, Ewert S, Dixon S (2018) Jointly detecting and separating singing voice: a multi-task approach. In: International conference on latent variable analysis and signal separation. Springer, Berlin, pp 329–339
63. Stter FR, Liutkus A, Badeau R, Edler B, Magron P (2016) Common fate model for unison source separation. In: IEEE International conference on acoustics, speech and signal processing (ICASSP). pp 126–130
64. Sturm BL, Morvidone M, Daudet L (2010) Musical instrument identification using multiscale mel-frequency cepstral coefficients. In: European signal processing conference. pp 477–481
65. Uhlich S, Giron F, Mitsufuji Y (2015) Deep neural network based instrument extraction from music. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 2135–2139
66. Uhlich S, Porcu M, Giron F, Enenkl M, Kemp T, Takahashi N, Mitsufuji Y (2017) Improving music source separation based on deep neural networks through data augmentation and network blending. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 261–265
67. Vembu S, Baumann S (2005) Separation of vocals from polyphonic audio recordings. In: International society for music information retrieval conference (ISMIR). pp 337–344
68. Vincent E, Gribonval R, Fevotte C (2006) Performance measurement in blind audio source separation. IEEE Trans Audio Speech Lang Process 14(4):1462–1469
69. Virtanen T (2007) Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. IEEE Trans Audio Speech Lang Process 15(3):1066–1074
70. Wang D (2005) On ideal binary mask as the computational goal of auditory scene analysis. Springer, New York, pp 181–197
71. Wang Y, Kan MY, Nwe TL, Shenoy A, Yin J (2004) Lyrically: automatic synchronization of acoustic musical signals and textual lyrics. In: ACM international conference on multimedia. ACM, Cambridge, pp 212–219
72. Wang Y, Narayanan A, Wang D (2014) On training targets for supervised speech separation. IEEE/ACM Trans Audio Speech Lang Process (TASLP) 22(12):1849–1858