Positions, Channels, and Layers: Fully Generalized Non-Local Network for Singer Identification

I-Yuan Kuo, Wen-Li Wei, Jen-Chun Lin
Institute of Information Science, Academia Sinica, Taiwan
iyuan.i.kuo@gmail.com, lilijinjin@gmail.com, jenchunlin@gmail.com

Abstract

Recently, a non-local (NL) operation has been designed as the central building block for deep-net models to capture long-range dependencies (Wang et al. 2018). Despite its excellent performance, it does not consider the interaction between positions across channels and layers, which is crucial in fine-grained classification tasks. To address the limitation, we target at singer identification (SID) task and present a fully generalized non-local (FGNL) module to help identify fine-grained vocals. Specifically, we first propose a FGNL operation, which extends the NL operation to explore the correlations between positions across channels and layers. Secondly, we further apply a depth-wise convolution with Gaussian kernel in the FGNL operation to smooth feature maps for better generalization. More, we modify the squeeze-and-excitation (SE) scheme into the FGNL module to adaptively emphasize correlated feature channels to help uncover relevant feature responses and eventually the target singer. Evaluating results on the benchmark artist20 dataset shows that the FGNL module significantly improves the accuracy of the deep-net models in SID. Codes are available at *https://github.com/ian-k-1217/Fully-Generalized-Non-Local-Network*.

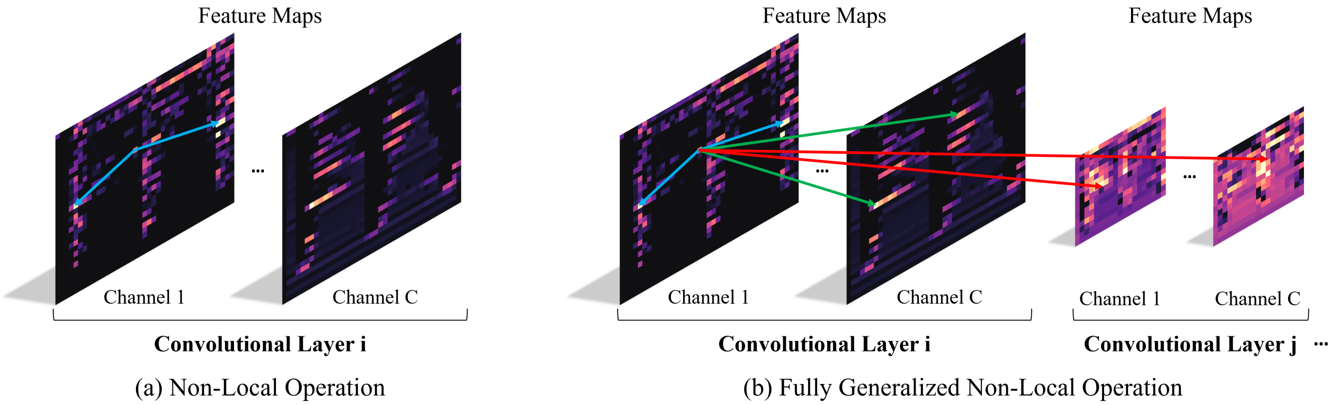
Introduction

The ability of humans to identify singers under limited guidance is remarkable. Take, for example, humans can quickly learn to identify singers by listening to only a few clips of music from those singers. Even without prior knowledge about singers, the human auditory system has evolved to be able to handle such a task by performing different functionalities that include exhibiting *attention* for specific frequency bands, capturing *long-range dependencies* of audio features as a whole, and extracting distinctive cues for comparison. All these can be done under the influence of background instrumental music and subtle sound variations from different singers (fine-grained vocals).

The functionalities provided by the human auditory system were a perfect match to a particular class of deep learning algorithms called *attention mechanism*. Attention mechanism is an attempt to mimic human brain action, that is, to selectively concentrate on a few relevant things, while ignoring others in deep-net models. It not only tells where to focus, but also improves the feature representation by capturing long-range spatial (or spatio-temporal) dependencies (Wang et al. 2018; Vaswani et al. 2017; Bahdanau, Cho, and Bengio 2015; Ramachandran et al. 2019). Among a mass of attention mechanisms, a non-local (NL) operation that belongs to the self-attention mechanism has recently been proposed, and has achieved great success in various vision and audio processing tasks (Wang et al. 2018; Hsieh et al. 2019; Jung et al. 2020; Zhang et al. 2019; Li et al. 2019). As illustrated in Figure 1 (a), the NL operation computes the response at a position in an image (or audio frame) by attending to all positions and taking their weighted average in an embedding space to achieve the goal of capturing long-range dependencies. Despite its excellent performance, the original NL module only considers the global spatial (or spatio-temporal) correlation by merging channels, which would lose important cues across channels and layers.

To mimic the functionalities of the human auditory system and improve the effectiveness in singer identification (SID) task, this study proposes a fully generalized non-local (FGNL) module that extends the NL module by learning explicit correlations among all of the elements (*positions*) across *channels* and *layers*, as shown in Figure 1 (b). Specifically, FGNL module contributes in three key aspects. First, we propose the FGNL operation, which scales up the representation power of NL operation to attend the interaction among feature maps across channels and layers and reveal the mutual similarity of the corresponding parts. Second, we suppress the noise in each feature map by integrating the Gaussian smoothing filter into the FGNL operation. Third, we modify the squeeze-and-excitation (SE) scheme (Hu, Shen, and Sun 2018) into the end of the FGNL module to adaptively recalibrate channel-wise feature responses by explicitly modeling inter-dependencies among channels. Figure 2 illustrates the details of the FGNL module.

To the best of our knowledge, our work is the first to introduce the attention mechanism for solving the SID task. Extensive experimental results show that: 1) Compared with the NL module, our FGNL module can capture richer feature representations and distinctive cues for prediction, and achieve the state-of-the-art results on the SID task; 2) The proposed FGNL module is flexible in the sense that it can be integrated into different deep-net architectures and be trained in an end-to-end fashion.



(a) Non-Local Operation

(b) Fully Generalized Non-Local Operation

Figure 1: Compared with the original non-local (NL) operation computes the response at each position by attending to all other positions in a single channel, the proposed fully generalized non-local (FGNL) operation further considers correlations among all of the positions across channels and layers.

Related Work

As our goal is to develop attention mechanisms for capturing richer feature representations and distinctive cues so that they could be used to facilitate the SID task. We discuss relevant literature and recent progress on both topics.

Singer Identification

SID is a classic task in the field of music information retrieval (MIR) (Nasrullah and Zhao 2019; Hsieh et al. 2020; Zhang et al. 2020; Van, Quang, and Thanh 2019). It aims to automatically identify the performing singers in given audio clips to facilitate the management of music libraries. There are two main challenges in the SID task. First, due to subtle differences in vocal organs, singers may have similar singing voices (fine-grained vocals), resulting in small inter-class variations (Hsieh et al. 2020; Sundberg 1989). As the number of singers to be considered increases, this issue becomes crucial. Second, since the songs in each singer's albums usually contain instrumental accompaniment, it is difficult for the SID model to extract vocal-only features from such recordings, which will reduce the generalization ability of the SID model (Hsieh et al. 2020; Van, Quang, and Thanh 2019; Sharma, Das, and Li 2019; Rafii et al. 2018; Sturm 2014).

With the success of deep learning, deep-net models such as convolutional neural network (CNN) and recurrent neural network (RNN) have been widely used to address both challenges. For the first challenge, the core behind these methods is to learn discriminative feature representations for singers to be identified. For example, Nasrullah and Zhao (Nasrullah and Zhao 2019) introduce an end-to-end trainable convolutional recurrent neural network (CRNN) to learn the discriminative feature representations and their temporal dependency to achieve the SID task. Hsieh *et al.* (Hsieh et al. 2020) further add a branch in CRNN to incorporate melody features for better performance. Van *et al.* (Van, Quang, and Thanh 2019) use the bidirectional long short-term memory (LSTM) network to learn the temporal dependency of feature representations for SID. Zhang *et al.* (Zhang et al. 2020) use the WaveNet to learn feature representations directly from the raw audio waveform in the time domain to identify singers. Despite the recent success of CNN and RNN, both convolutional and recurrent operations can only process a local neighborhood (Wang et al. 2018), making it difficult to learn non-local context relations between audio feature representations, which is essential for distinguishing fine-grained vocals.

Regarding the second challenge, the key is to separate the vocal parts of the given audio clips to minimize the influence of instruments on the learning of the SID model. For example, Van *et al.* (Van, Quang, and Thanh 2019) combine a gated recurrent unit (GRU) on U-Net to separate the vocal parts from the song that with mixed background accompaniment. Sharma *et al.* (Sharma, Das, and Li 2019) introduce an end-to-end trainable Wave-U-Net to learn the separation of singing voices, thereby eliminating the interference of background accompaniment on singer identity cues. Hsieh *et al.* (Hsieh et al. 2020) use an open source tool called Open-Unmix (Stöter et al. 2019), which combines a three-layer bidirectional LSTM and multiplicative skip connection to separate the vocal and instrumental tracks of music, and has made great progress. As source separation technology has become more mature and has been successfully used to improve the performance of SID task, in this study, we integrate the source separation model (Stöter et al. 2019) into our system and focus on solving the first challenge by introducing attention mechanism.

Attention Mechanism

Attention mechanism has enjoyed widespread adoption as a computational module for modeling sequences because of its ability to capture long-range dependencies and selectively concentrate on the relevant subset of the input (Vaswani et al. 2017; Bahdanau, Cho, and Bengio 2015; Devlin et al. 2019; Yu et al. 2018). For example, Bahdanau *et al.* (Bahdanau, Cho, and Bengio 2015) present for the first time an attention mechanism and combine it with the RNN encoder-decoder in a neural machine translation model to allow selective attention to relevant information from a variable length source sentence. Vaswani *et al.* (Vaswani et al. 2017) further propose a Transformer architecture to draw global dependencies between input and output. This architecture entirely replaces recurrence with self-attention, and greatly improves the performance of machine translation. Such a self-attention mechanism has also been extended to other language representation models such as BERT (Devlin et al. 2019) and achieved the state-of-the-art results.

Creating attention mechanisms to compensate for the weakness of convolution has also become an emerging theme in vision tasks (Hu, Shen, and Sun 2018; Wang et al. 2018; Ramachandran et al. 2019; Woo et al. 2018; Roy, Navab, and Wachinger 2019; Bello et al. 2019). For example, Hu *et al.* (Hu, Shen, and Sun 2018) present a channel-wise attention mechanism to explicitly model the inter-dependencies between the channels of its spatial features. It is intended to select the useful feature maps and suppress the others by considering the global information of each channel. Woo *et al.* (Woo et al. 2018) and Roy *et al.* (Roy, Navab, and Wachinger 2019) further explore both spatial and channel-wise attentions, and verify that using both is superior to using only the channel-wise attention. Recently, Wang *et al.* (Wang et al. 2018) show that self-attention is an instantiation of non-local mean (Buades, Coll, and Morel 2005), and present a NL operation for the convolution-based deep-net models to capture long-range dependencies. Specifically, the NL operation computes the correlation matrix between each spatial point in the feature maps to generate an attention map, and then perform the attention-guided dense context information aggregation. Such a NL operation has become the core component for various deep-net architectures to capture non-local context relations, and has been successfully applied in various fields, including vision, audio, *etc.* (Wang et al. 2018; Li et al. 2019). Despite its excellent performance, the original NL operation only considers the global spatial (or spatio-temporal) correlation by merging channels, which may miss subtle but important cues across channels and layers in fine-grained classification tasks. In this work, we propose the FGNL operation, which extends the NL operation to further explore the explicit correlations among all of the elements (positions) across channels and layers to obtain richer feature representations and distinctive cues.

Approach

In this section, we elaborate the proposed FGNL module. We first revisit the original NL operation (Wang et al. 2018). Then, we will introduce three extensions of FGNL module in detail, including FGNL operation, Gaussian smoothing filter, and modified squeeze-and-excitation (MoSE) scheme.

Revisiting NL Operation

The original NL operation (Wang et al. 2018) is revisited in matrix form shortly. Given the input feature map $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times C}$ for the NL operation, the goal is to obtain a response $\mathbf{Y} \in \mathbb{R}^{T\times H\times W\times \frac{C}{m}}$, which aims to capture the non-local context relations (*i.e.*, long-range dependencies) across the whole feature map by weighting sum of the features at all positions,

$$\mathbf{Y} = f(\theta(\mathbf{X}), \phi(\mathbf{X}))g(\mathbf{X}), \tag{1}$$

where T denotes the number of input video frames (when the input is a single image, T can be ignored), H and W denote the height and width of the feature map, C is the number of channels, m is a reduction ratio, which refers to the bottleneck design used for reducing the computational complexity (Wang et al. 2018), $f(\cdot,\cdot)$ represents the pairwise function, which calculates the affinity between all positions, and $\theta(\cdot)$, $\phi(\cdot)$, and $g(\cdot)$ are learnable transformations recommended to be implemented by using 1×1 or $1\times 1\times 1$ convolution (Wang et al. 2018). Thus, the transformations can be written as

$$\theta(\mathbf{X}) = \mathbf{X}\mathbf{W}_\theta \in \mathbb{R}^{N\times\frac{C}{m}}, \tag{2}$$

$$\phi(\mathbf{X}) = \mathbf{X}\mathbf{W}_\phi \in \mathbb{R}^{N\times\frac{C}{m}}, \tag{3}$$

and

$$g(\mathbf{X}) = \mathbf{X}\mathbf{W}_g \in \mathbb{R}^{N\times\frac{C}{m}}, \tag{4}$$

parameterized by the weight matrices $\mathbf{W}_\theta$, $\mathbf{W}_\phi$, and $\mathbf{W}_g \in \mathbb{R}^{C\times\frac{C}{m}}$, respectively. Here N denotes the collapsing of all the spatial or spatio-temporal positions in one dimension, *i.e.*, $N = HW$ or $N = HWT$. In the implementation, the original NL operation provides multiple options for f. For simplicity, we choose the dot product as an example, *i.e.*,

$$f(\theta(\mathbf{X}), \phi(\mathbf{X})) = \theta(\mathbf{X})\phi(\mathbf{X})^\intercal, \tag{5}$$

where the size of the resulting pairwise function $f(\cdot,\cdot)$ denotes as $\mathbb{R}^{N\times\frac{C}{m}} \times \mathbb{R}^{\frac{C}{m}\times N} \to \mathbb{R}^{N\times N}$. Thus, by substituting equations (2) to (5) into (1), the response $\mathbf{Y}$ can be obtained as

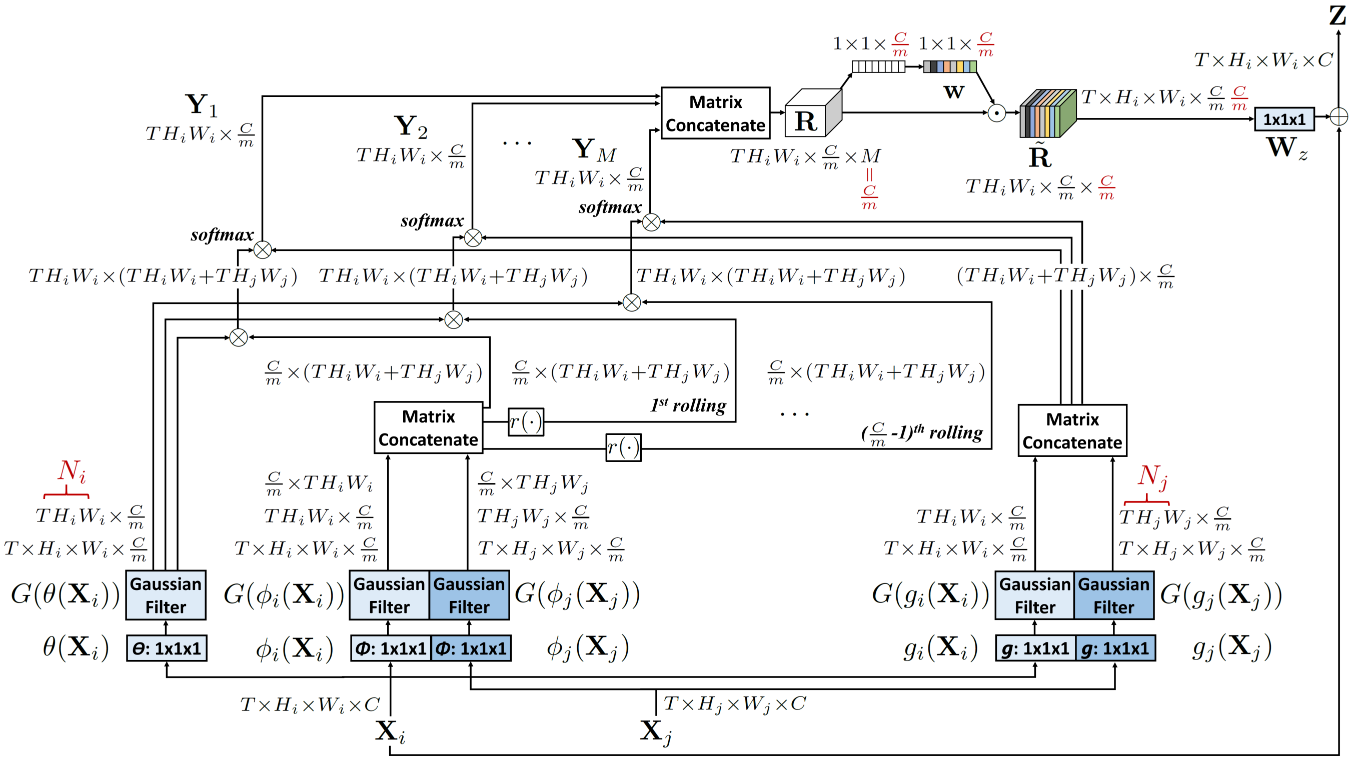
$$\mathbf{Y} = \mathbf{X}\mathbf{W}_\theta\mathbf{W}_\phi^\intercal\mathbf{X}^\intercal\mathbf{X}\mathbf{W}_g, \tag{6}$$

where the pairwise matrix $\mathbf{X}\mathbf{W}_\theta\mathbf{W}_\phi^\intercal\mathbf{X}^\intercal \in \mathbb{R}^{N\times N}$ encodes the mutual similarity between any positions of the input feature. The effect of NL operation can be understood as the self-attention mechanism (Vaswani et al. 2017) in the sense that each position (row) in the resulting $\mathbf{Y}$ is a linear combination of all the positions (columns) of $\mathbf{X}\mathbf{W}_g$ weighted by the corresponding row of the pairwise matrix.

Our FGNL Module

The original NL operation aims to capture the long-range dependencies between any two positions in one convolutional layer. However, it only calculates the dependencies of any two positions in each channel separately, and aggregates all channel information in one convolutional layer together through a joint location-wise matrix $f(\theta(\mathbf{X}), \phi(\mathbf{X}))$. Thus, it will lose the interaction between positions across channels and layers. To this end, we generalize the original NL operation so that the long-range dependencies between any positions of any channels and layers can be modeled.

We formulate the proposed FGNL module as follows. Given a set of input feature maps $\mathbf{F} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_L\}$ for the FGNL module, the goal of the FGNL operation



$\mathbf{Z}$
$1\times1\times\frac{C}{m}$
$1\times1\times\frac{C}{m}$
$\mathbf{w}$
$T\times H_i\times W_i\times C$
Matrix Concatenate
$\mathbf{R}$
$\tilde{\mathbf{R}}$
$T\times H_i\times W_i\times\frac{C}{m}\frac{C}{m}$
1x1x1
$\mathbf{W}_z$
$\mathbf{Y}_1$
$TH_iW_i\times\frac{C}{m}$
$\mathbf{Y}_2$
$TH_iW_i\times\frac{C}{m}$
$\mathbf{Y}_M$
$TH_iW_i\times\frac{C}{m}$
$TH_iW_i\times\frac{C}{m}\times M$
$\frac{C}{m}$
$TH_iW_i\times\frac{C}{m}\times\frac{C}{m}$
softmax
softmax
softmax
$TH_iW_i\times(TH_iW_i+TH_jW_j)$
$TH_iW_i\times(TH_iW_i+TH_jW_j)$
$TH_iW_i\times(TH_iW_i+TH_jW_j)$
$(TH_iW_i+TH_jW_j)\times\frac{C}{m}$
$\frac{C}{m}\times(TH_iW_i+TH_jW_j)$
$\frac{C}{m}\times(TH_iW_i+TH_jW_j)$
$\frac{C}{m}\times(TH_iW_i+TH_jW_j)$
1st rolling
$(\frac{C}{m}$-1)th rolling
Matrix Concatenate
$r(\cdot)$
$r(\cdot)$
Matrix Concatenate
N_i
N_j
$TH_iW_i\times\frac{C}{m}$
$T\times H_i\times W_i\times\frac{C}{m}$
$\frac{C}{m}\times TH_iW_i$
$TH_iW_i\times\frac{C}{m}$
$T\times H_i\times W_i\times\frac{C}{m}$
$\frac{C}{m}\times TH_jW_j$
$TH_jW_j\times\frac{C}{m}$
$T\times H_j\times W_j\times\frac{C}{m}$
$TH_iW_i\times\frac{C}{m}$
$T\times H_i\times W_i\times\frac{C}{m}$
$TH_jW_j\times\frac{C}{m}$
$T\times H_j\times W_j\times\frac{C}{m}$
$G(\theta(\mathbf{X}_i))$
Gaussian Filter
$G(\phi_i(\mathbf{X}_i))$
Gaussian Filter
Gaussian Filter
$G(\phi_j(\mathbf{X}_j))$
$G(g_i(\mathbf{X}_i))$
Gaussian Filter
Gaussian Filter
$G(g_j(\mathbf{X}_j))$
$\theta(\mathbf{X}_i)$
θ: 1x1x1
$\phi_i(\mathbf{X}_i)$
Φ: 1x1x1
Φ: 1x1x1
$\phi_j(\mathbf{X}_j)$
$g_i(\mathbf{X}_i)$
g: 1x1x1
g: 1x1x1
$g_j(\mathbf{X}_j)$
$T\times H_i\times W_i\times C$
$\mathbf{X}_i$
$T\times H_j\times W_j\times C$
$\mathbf{X}_j$

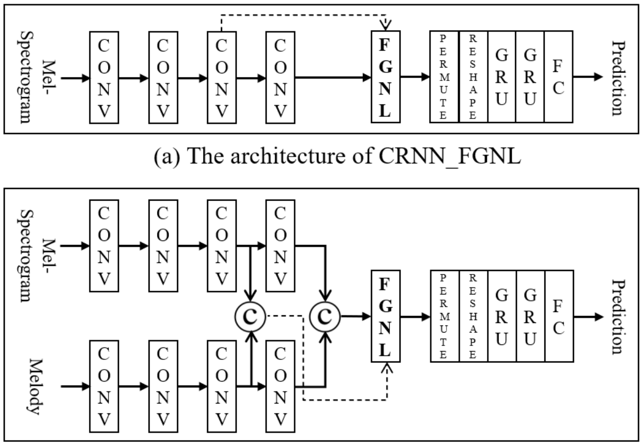
Figure 2: A spatio-temporal FGNL module. The feature maps are shown as the shape of their tensors, *e.g.*, $T\times H_i\times W_i\times C$ for C channels (proper reshaping is performed when noted). θ, ϕ, and g denote $1\times1\times1$ convolutions, $\otimes$ denotes matrix multiplication, $\odot$ denotes the element-wise product, and $\oplus$ denotes element-wise sum. The computation of softmax is performed on each row.

is to obtain a set of non-local context responses $\mathbf{R} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_M\}$, where L represents the number of layers and M is the number of responses. For the sake of clarity, in the following, we use two-layer input feature maps $\mathbf{X}_i \in \mathbb{R}^{T\times H_i\times W_i\times C}$ and $\mathbf{X}_j \in \mathbb{R}^{T\times H_j\times W_j\times C}$, namely $\mathbf{F} = \{\mathbf{X}_i, \mathbf{X}_j\}$ as an example to explain the FGNL operation. (See Figure 2.) To this end, each response $\mathbf{Y}_k$ in $\mathbf{R}$ can be calculated by weighting sum of the features at all positions,

$$\begin{aligned}\mathbf{Y}_k = f(G(\theta(\mathbf{X}_i)), r([G(\phi_i(\mathbf{X}_i)), G(\phi_j(\mathbf{X}_j))]))\\ \times [G(g_i(\mathbf{X}_i)), G(g_j(\mathbf{X}_j))].\end{aligned} \tag{7}$$

Similar to the NL operation, $\theta(\cdot)$, $\phi_i(\cdot)$, $\phi_j(\cdot)$, $g_i(\cdot)$, and $g_j(\cdot)$ are learnable transformations. In the implementation, we set the number of channels represented by the weight matrices $\mathbf{W}_\theta$, $\mathbf{W}_{\phi_i}$, $\mathbf{W}_{\phi_j}$, $\mathbf{W}_{g_i}$, and $\mathbf{W}_{g_j}$ in the above transformations to $\frac{1}{m}$ of the number of channels in $\mathbf{X}_i$ and $\mathbf{X}_j$. Here m is a reduction ratio, and is set to 32 in our experiments, and $G(\cdot)$ represents the Gaussian smoothing filter, which suppresses noise by performing the depth-wise convolution between the feature map and the Gaussian kernel. A two-dimensional Gaussian kernel, *i.e.*, $G(p,q) = \frac{1}{2\pi\sigma^2}e^{-(p^2+q^2)/2\sigma^2}$ is adopted, where p and q represent the spatial coordinates of the feature map (*i.e.*, resulting from $\theta(\mathbf{X}_i)$ or $\phi_i(\mathbf{X}_i)$ or $\phi_j(\mathbf{X}_j)$ or $g_i(\mathbf{X}_i)$ or $g_j(\mathbf{X}_j)$), and σ is the standard deviation. As σ grows, the feature map becomes smoother, providing more noise suppression capabilities. $[\cdot,\cdot]$ is the operation of matrix concatenation, and $r(\cdot)$ represents a rolling function, which rolls the elements of the matrix along the channel axis. Thus, by concatenating the matrices from different layers and subsequently rolling the matrix along the channel axis, the long-range dependencies between the positions across channels and layers can be obtained through the operation of the pairwise function $f(\cdot,\cdot)$. Similar to (5), we choose the dot product as the operation for f, and then normalize it by using the softmax computation. The size of the resulting pairwise function $f(\cdot,\cdot)$ denotes as $\mathbb{R}^{N_i\times\frac{C}{m}}\times\mathbb{R}^{\frac{C}{m}\times(N_i+N_j)}\rightarrow\mathbb{R}^{N_i\times(N_i+N_j)}$. Here N_i and N_j denote the collapsing of all the spatial or spatio-temporal positions for layer i and j, respectively. To this end, the response $\mathbf{Y}_k \in \mathbb{R}^{N_i\times\frac{C}{m}}$ can be calculated by the linear combination between the two matrices resulted from $f(\cdot,\cdot)$ and $[G(g_i(\mathbf{X}_i)), G(g_j(\mathbf{X}_j))]$. As a result, $\mathbf{R} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_{M=\frac{C}{m}}\}$ can be obtained by repeating the operation of (7), which rolls matrix $r([G(\phi_i(\mathbf{X}_i)), G(\phi_j(\mathbf{X}_j))]$ $\frac{C}{m}-1$ times along the channel axis.

Besides capturing the long-range dependencies with FGNL operation, we further explore the relatedness between each response $\mathbf{Y}_k$ in $\mathbf{R}$ through the modified squeeze-and-excitation (MoSE) scheme, which adaptively recalibrates each response $\mathbf{Y}_k$ by considering inter-dependencies over the M responses. Specifically, the squeeze step spatially summarizes each response with global average pooling,



(a) The architecture of CRNN_FGNL

(b) The architecture of CRNNM_FGNL

Figure 3: The proposed CRNN_FGNL and CRNNM_FGNL architectures. © means concatenation.

while the excitation function emphasizes those responses that play crucial roles in identifying the target. In between the squeeze layer and the excitation layer, we use two convolutional layers that connect ReLU and softmax activations respectively to modify the original SE scheme (Hu, Shen, and Sun 2018). We depict the MoSE operation as follows:

$$\mathrm{MoSE}(\mathbf{R}) = \mathbf{w}, \quad \tilde{\mathbf{R}} = \mathbf{w} \odot \mathbf{R}, \tag{8}$$

where $\tilde{\mathbf{R}}$ is a re-weighted set of non-local context responses, $\mathbf{w} \in \mathbb{R}^M$ is the excitation vector, and $\odot$ denotes the element-wise product.

Finally, as in the design of the NL module (Wang et al. 2018), we use residual connection to generate the output feature representation (map) $\mathbf{Z} \in \mathbb{R}^{T \times H_i \times W_i \times C}$ (refer to Figure 2) of the FGNL module as follows:

$$\mathbf{Z} = \tilde{\mathbf{R}}\mathbf{W}_z + \mathbf{X}_i, \tag{9}$$

where $\mathbf{W}_z$ is a learnable weight matrix, which can be implemented by using 1×1 or $1 \times 1 \times 1$ convolution (*i.e.*, depends on frame-wise (spatial) classification or sequence-wise (spatio-temporal) classification task), and the number of channels in $\mathbf{W}_z$ is scaled up to match the number of channels in $\mathbf{X}_i$. "$+\mathbf{X}_i$" denotes a residual connection (He et al. 2016). Such a residual connection allows us to insert a new FGNL module into any pre-trained model, without breaking its initial behavior (*e.g.*, if $\mathbf{W}_z$ is initialized as zero). As a result, by further considering the re-weighted non-local context responses $\tilde{\mathbf{R}}$, the information in $\mathbf{Z}$ is richer so $\mathbf{Z}$ can be regarded as enhanced $\mathbf{X}_i$.

Experiments

To demonstrate the effectiveness of the proposed FGNL module, we conduct SID experiments on the benchmark *artist20* dataset (Ellis 2007), which includes a total of 1,413 complete songs collected from 20 artists (singers). In the experiments, album-split (Hsieh et al. 2020; Nasrullah and Zhao 2019) is employed, which ensures that the songs from the same album are split either in the training, validation, or the test set, to eliminate additional clues provided by the album. All evaluated deep-net models are trained with audio clips of length $\{3\text{s}, 5\text{s}, 10\text{s}\}$. Among them, 90% audio clips (so-called frames) are used for training and the rest are used for testing. The data in the validation set is split from 10% of the training data.

Evaluation Protocols

We integrate the proposed FGNL module into two state-of-the-art SID models, the convolutional recurrent neural network (CRNN) (Nasrullah and Zhao 2019) and the convolutional recurrent neural network with melody (CRNNM) (Hsieh et al. 2020) in order to compare performance. For both CRNN and CRNNM, we follow their original architecture settings as benchmarks. Briefly, the CRNN architecture is defined as a stack of four convolutional layers, two GRU layers, and one fully connected (FC) layer. The CRNNM architecture is basically the same as CRNN, except that CRNNM also includes a branch related to melody. Such a melody branch consists of a stack of four convolutional layers, and its output will be concatenated to the main branch of the CRNN for subsequent processing. For the proposed FGNL networks, we insert the FGNL module into the network architecture of CRNN and CRNNM, respectively named CRNN_FGNL and CRNNM_FGNL. For CRNN_FGNL, as shown in Figure 3 (a), we insert a FGNL module after the fourth convolutional layer to model the non-local context relations between the feature maps of the fourth and third convolutional layers. For CRNNM_FGNL, as shown in Figure 3 (b), we insert the FGNL module after the fourth convolutional layer of the mel-spectrogram and melody branches to model the non-local context relations between the feature maps of the fourth and third convolutional layers. For the training of the above deep-net models, we apply random initialization for the weights, a constant learning rate of 10^{-4}, the dropout and batch normalization to avoid over-fitting, and the Adam solver (Kingma and Ba 2015) for optimization. Each model is trained by using back-propagation algorithm (including back-propagation through time algorithm) with the objective of softmax cross entropy under the supervision of the ground truth artist (singer) label. The meta-parameters of each model are set based on the validation error.

To evaluate whether the background accompaniment will affect the generalization ability of the above deep-net models, two evaluation settings are considered, including the *original audio file* and the *vocal-only*. The difference between them is that the vocal-only setting further employs the Open-Unmix toolkit (Stöter et al. 2019) to separate the vocal parts from each audio file in training and test. In the experiments, we report the evaluation results of each deep-net model at the frame level and the song level. Specifically, at the frame level, each *t*-length (3s, 5s, or 10s) audio spectrogram is treated as an independent sample, and the performance is measured by taking the F1 score across all samples in the test set. For evaluation at the song level, majority voting will be applied to select the most frequent frame level

		Original Audio File						Vocal-Only						
		Frame Level			Song Level			Frame Level			Song Level			
Model	**Type**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**#Parameters**
CRNN	Average	0.44	0.45	0.48	0.57	0.55	0.58	0.42	0.46	**0.51**	0.72	0.74	0.74	394,516
(Nasrullah and Zhao 2019)	Best	0.46	0.47	0.53	0.62	0.59	0.60	**0.44**	**0.48**	**0.53**	0.76	0.79	0.77	
CRNN_FGNL (Ours)	Average	**0.52**	**0.54**	**0.55**	**0.72**	**0.73**	**0.73**	**0.44**	**0.47**	**0.51**	**0.79**	**0.80**	**0.79**	584,141
	Best	**0.54**	**0.57**	**0.58**	**0.76**	**0.79**	**0.78**	**0.44**	**0.48**	**0.53**	**0.81**	**0.82**	**0.83**	
CRNNM	Average	0.47	0.47	0.51	0.62	0.61	0.65	**0.42**	0.46	0.49	0.73	0.75	0.73	778,772
(Hsieh et al. 2020)	Best	0.48	0.50	0.53	0.67	0.68	0.69	0.43	**0.47**	0.50	0.75	0.79	0.75	
CRNNM_FGNL (Ours)	Average	**0.54**	**0.55**	**0.58**	**0.74**	**0.74**	**0.73**	**0.42**	**0.47**	**0.52**	**0.77**	**0.83**	**0.81**	1,175,381
	Best	**0.55**	**0.57**	**0.63**	**0.82**	**0.81**	**0.83**	**0.44**	**0.47**	**0.53**	**0.83**	**0.84**	**0.86**	

Table 1: The average and best F1 scores of the frame level and the song level in various length settings. Each *t*-length (3s, 5s, or 10s) experiment repeats three independent runs. Bold is the comparison winner of the same series (CRNN or CRNNM) model.

		Original Audio File						Vocal-Only					
		Frame Level			Song Level			Frame Level			Song Level		
Model	**Type**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**
CRNN (Nasrullah and Zhao 2019)	Average	0.44	0.45	0.48	0.57	0.55	0.58	0.42	0.46	**0.51**	0.72	0.74	0.74
	Best	0.46	0.47	0.53	0.62	0.59	0.60	**0.44**	**0.48**	**0.53**	0.76	0.79	0.77
CRNN_NL	Average	0.51	0.52	0.54	0.71	0.69	0.69	0.42	0.46	0.50	0.77	0.78	0.76
(w/o the cues across channels and layers)	Best	0.53	0.53	0.55	0.76	0.74	0.74	0.43	0.46	0.51	**0.81**	0.81	0.79
CRNN_FGNL_LIGHT	Average	0.51	**0.54**	0.54	0.70	**0.73**	0.69	0.43	**0.47**	**0.51**	0.77	0.77	0.77
(w/o the cues across layers)	Best	**0.54**	0.55	0.55	**0.78**	0.77	**0.78**	**0.44**	**0.48**	**0.53**	0.80	0.80	0.82
CRNN_FGNL (Ours)	Average	**0.52**	**0.54**	**0.55**	**0.72**	**0.73**	**0.73**	**0.44**	**0.47**	**0.51**	**0.79**	**0.80**	**0.79**
	Best	**0.54**	**0.57**	**0.58**	0.76	**0.79**	**0.78**	**0.44**	**0.48**	**0.53**	**0.81**	**0.82**	**0.83**

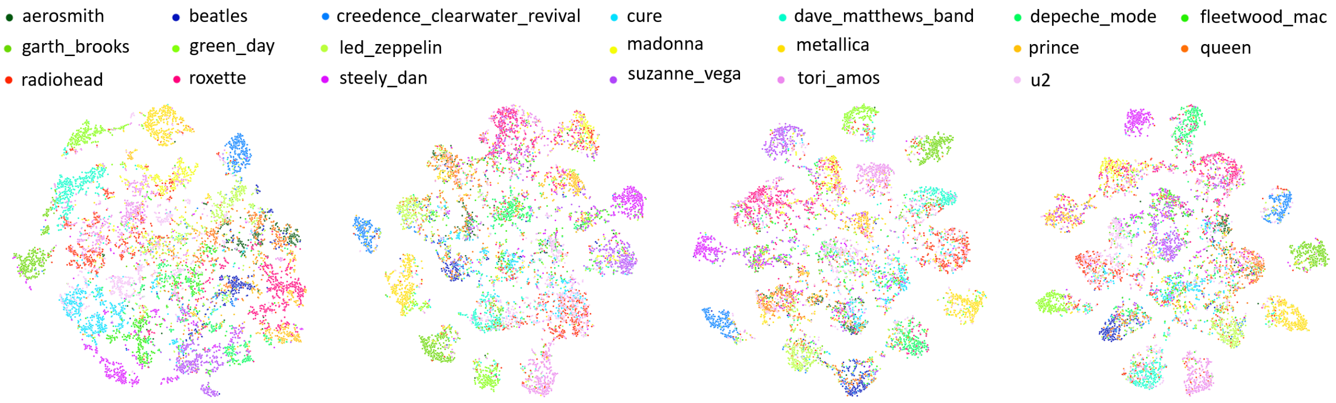
Table 2: Ablation experiments of CRNN with three attention modules, including NL (Wang et al. 2018), FGNL_LIGHT, and FGNL. Each *t*-length experiment repeats three independent runs. Bold indicates the comparison winner of the model.

artist prediction as the final prediction for each song. Note that in the implementation, if the confidence (softmax output) of the test frame is less than 0.5, it will be removed and will not participate in voting (Nasrullah and Zhao 2019). The F1 score is then reported by song to quantify performance.

Results and Comparisons

For all the above competition methods, Table 1 summarizes the average and best test F1 scores of the frame level and the song level resulted from three independent runs. For the comparison between CRNN and CRNNM, similar to the results in (Hsieh et al. 2020), the results first show that CRNNM is better than CRNN in most settings. Such results indicate that further consideration of melody-related features is positive for SID. However, although CRNNM outperformed CRNN, the performance is still limited. One explanation for this may be that both convolutional and recurrent operations in CRNNM only consider a local neighborhood (Wang et al. 2018), so it is difficult to capture the non-local context relations (*i.e.*, long-range dependencies) between audio features to distinguish singer. To tackle the issue, we introduce the attention mechanism and develop the FGNL module to explicitly model the correlations among all of the positions in the feature map across channels and layers. By further integrating the FGNL module, the results support that CRNN_FGNL and CRNNM_FGNL can learn richer feature representations and distinctive cues to complete SID. That is, compared with the original CRNN and CRNNM, CRNN_FGNL and CRNNM_FGNL achieve great improvements. In addition, it is noteworthy that the improvement of the FGNL module is not just because it adds the number of parameters to the baseline model. To see this, we note that in Table 1, CRNN_FGNL has better performance than CRNNM but has fewer parameters.

For comparing the original audio file setting with the vocal-only setting at the frame level and the song level, as shown in Table 1, we first notice that the vocal-only setting at the frame level performs worse than the original audio file setting. Such results indicate that a model trained with the original audio files may benefit from the additional information in the accompaniment. This is supported by another observation. It is observed that the model can identify the singer, even if some segments (*e.g.*, intro, inter, or outro) in the song do not contain the vocals. However, it is interesting that the vocal-only setting at the song level performs better than the original audio file setting. This is because in song level prediction, lower confidence frames will be removed and will not contribute to the voting. Although the accompaniment in the original audio file setting will provide extra information, the confidence is usually low. This is because the accompaniment in some vocal segments of the song could confuse the identification. In this case, the source separation technique, which is used to separate the human voice from the original audio, would increase the identification confidence of the SID model. Thus, the results indicate that source separation plays a positive role when considering the identification confidence in the song level. All in all, the proposed FGNL module makes significant improvements to CRNN and CRNNM in both the original audio file and the vocal-only settings.



(a) CRNN (b) CRNN_NL (c) CRNN_FGNL_LIGHT (d) CRNN_FGNL (Ours)

Figure 4: Visualization of the embeddings (projected into 2-D space by t-SNE) under the original audio file setting of the 5-sec frame level test samples. From left to right are CRNN, CRNN_NL, CRNN_FGNL_LIGHT, and CRNN_FGNL.

		Original Audio File						**Vocal-Only**					
		Frame Level			**Song Level**			**Frame Level**			**Song Level**		
Model	**Type**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**	**3s**	**5s**	**10s**
CRNN_FGNL	Average	0.52	**0.54**	0.54	0.71	**0.73**	0.72	**0.44**	0.46	0.50	0.78	0.79	0.76
without Gaussian smoothing	Best	0.53	0.55	0.55	0.77	**0.80**	0.74	**0.45**	0.47	0.52	0.81	**0.83**	0.81
CRNN_FGNL	Average	0.52	0.53	0.54	0.71	0.72	0.71	0.43	0.45	**0.51**	0.78	0.78	0.78
without MoSE	Best	**0.54**	0.54	0.55	0.77	0.77	0.77	0.44	0.46	**0.53**	0.81	0.82	**0.83**
CRNN_FGNL	Average	**0.53**	**0.54**	0.54	**0.72**	0.71	0.70	**0.44**	0.46	**0.51**	**0.80**	0.76	0.77
with Gaussian smoothing and SE	Best	**0.54**	0.55	0.57	**0.78**	0.79	**0.78**	0.44	**0.48**	0.52	**0.83**	0.78	**0.83**
CRNN_FGNL	Average	0.52	**0.54**	**0.55**	**0.72**	**0.73**	**0.73**	**0.44**	**0.47**	**0.51**	0.79	**0.80**	**0.79**
with Gaussian smoothing and MoSE	Best	**0.54**	**0.57**	**0.58**	0.76	0.79	**0.78**	0.44	**0.48**	**0.53**	0.81	0.82	**0.83**

Table 3: Ablation experiments of CRNN_FGNL with and without Gaussian smoothing, MoSE, and SE (Hu, Shen, and Sun 2018) mechanisms. Bold indicates the comparison winner of the model.

To verify whether the cues across channels and layers in the proposed FGNL module are useful, we conducted ablation experiments under the CRNN architecture. Specifically, the proposed CRNN_FGNL is compared with the CRNN_FGNL_LIGHT (*i.e.*, without the cues across layers), the CRNN_NL (Wang et al. 2018) (*i.e.*, without the cues across channels and layers), and the original CRNN. All attention modules (*i.e.*, NL, FGNL_LIGHT, and FGNL) are inserted after the fourth convolutional layer of the CRNN architecture (Nasrullah and Zhao 2019). For performance comparison, it is obvious from Table 2 that CRNN_NL is superior to CRNN in almost all settings. The results confirm that by further introducing NL module to model the non-local context relations of audio features, SID performance can indeed be improved. Despite its excellent performance, it can only compute the response at each position by attending to all other positions in each channel separately, which will lose important information between positions across channels and layers. Compared with CRNN_NL, the CRNN_FGNL_LIGHT demonstrates that by further considering the cues across channels, the performance can indeed be improved. The results can be further verified by visualizing the feature embedding in each competing model. To this end, we employ t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton 2008) to project the computed embedding vectors to a 2-D space for visualization. Briefly, for each of the above models, we regard the output of the last layer of GRU in the CRNN architecture as embedding and visualize it through t-SNE. For space limit, we visualize the four competing models under the setting of the original audio file at the 5-sec frame level. The audio samples of testing set are drawn and colored according to the ground truth artist (singer) labels in Figure 4. It can be seen from the result of CRNN_FGNL_LIGHT that samples from different singers are fairly well-separated in the embedding space. The result of CRNN_NL looks chaotic and less separated, suggesting again that a model taking the cues across channels (*i.e.*, CRNN_FGNL_LIGHT) may achieve SID better. Finally, as shown in Table 2 and Figure 4, by simultaneously exploring the correlations between positions across channels and layers, the CRNN_FGNL can indeed capture richer feature representations and distinctive cues to facilitate the identification of singers. Overall, CRNN_FGNL achieves the best performance among the above competing models.

Finally, to evaluate whether integrating the Gaussian smoothing filter and the modified squeeze-and-excitation (MoSE) scheme into the FGNL module (refer to Figure 2) can improve the generalization ability of the model, ablation experiments are further conducted in CRNN_FGNL (refer to Figure 3(a)). Four settings are considered, including

CRNN_FGNL without Gaussian smoothing, CRNN_FGNL without MoSE, CRNN_FGNL with Gaussian smoothing and squeeze-and-excitation (SE) (Hu, Shen, and Sun 2018), and CRNN_FGNL with Gaussian smoothing and MoSE (our full version). Table 3 summarizes performance comparison. It is observed that the CRNN_FGNL with Gaussian smoothing and MoSE outperforms CRNN_FGNL without Gaussian smoothing and CRNN_FGNL without MoSE, demonstrating that generalizing the ability of the model could result from both using the Gaussian smoothing filter to suppress the noise of the feature map, and using the MoSE scheme to recalibrate the channel-wise feature responses. Besides, comparing the original SE and the proposed MoSE scheme, the results show that CRNN_FGNL with Gaussian smoothing and MoSE is better than CRNN_FGNL with Gaussian smoothing and SE. Such results indicate that using the convolutional layer and softmax operation instead of fully connected layer and sigmoid operation in SE module can indeed increase the generalization ability of the model. All in all, the ablation experiments show that the Gaussian smoothing filter and the MoSE scheme improve the SID performance for the deep-net model. More experiments and a demo video can be found at *https://github.com/ian-k-1217/Fully-Generalized-Non-Local-Network*.

Conclusions

We have introduced a new attention mechanism called the fully generalized non-local (FGNL) module, which can better capture the non-local context relations (*i.e.*, long-range dependencies) of audio features to help identify fine-grained vocals. The results have demonstrated that the FGNL module significantly improves the accuracy of the deep-net models in singer identification (SID) task and achieves the state-of-the-art level. Moreover, it is shown that the proposed FGNL module is superior to the popular non-local (NL) module (Wang et al. 2018) by explicitly modeling the rich inter-dependencies between any positions across channels and layers in the feature space, while the NL module only considers the correlations between positions along the specific channel. Based on the promising outcomes, our future work will focus on developing more effective loss functions to improve the fineness of the learned feature representation. We also plan to expand the scale of the experiments to other tasks in the future, such as vision tasks.

Acknowledgments

This work was supported in part by MOST grant 107-2218-E-001-010-MY2. The third author would like to thank his best friend, Chao-Wei Chen, for his suggestion and encouragement.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, 1–15.

Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; and Le, Q. V. 2019. Attention Augmented Convolutional Networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 3286–3295.

Buades, A.; Coll, B.; and Morel, J.-M. 2005. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 60–65.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Ellis, D. P. W. 2007. Classifying Music Audio with Timbral and Chroma Features. In *Proceedings International Society for Music Information Retrieval Conference (ISMIR)*, 339–340. [Online] https://labrosa.ee.columbia.edu/projects/artistid/.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hsieh, T.-H.; Cheng, K.-H.; Fan, Z.-C.; Yang, Y.-C.; and Yang, Y.-H. 2020. Addressing The Confounds Of Accompaniments In Singer Identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

Hsieh, T.-I.; Lo, Y.-C.; Chen, H.-T.; and Liu, T.-L. 2019. One-Shot Object Detection with Co-Attention and Co-Excitation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2725–2734.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7132–7141.

Jung, Y.; Kim, D.; Woo, S.; Kim, K.; Kim, S.; and Kweon, I. S. 2020. Hide-and-Tell: Learning to Bridge Photo Streams for Visual Storytelling. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 11213–11220.

Kingma, D. P.; and Ba, J. L. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Li, X.; Li, Y.; Li, M.; Xu, S.; Dong, Y.; Sun, X.; and Xiong, S. 2019. A Convolutional Neural Network with Non-Local Module for Speech Enhancement. In *Proc. Interspeech*, 1796–1800.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9: 2579–2605.

Nasrullah, Z.; and Zhao, Y. 2019. Music Artist Classification with Convolutional Recurrent Neural Networks. In *International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Rafii, Z.; Liutkus, A.; Stöter, F.-R.; Mimilakis, S. I.; FitzGerald, D.; and Pardo, B. 2018. An Overview of Lead and Accompaniment Separation in Music. *IEEE/ACM Transactions on Audio, Speech and Language Processing* 26(8): 1307–1335.

Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; and Shlens, J. 2019. Stand-Alone Self-Attention in Vision Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 68–80.

Roy, A. G.; Navab, N.; and Wachinger, C. 2019. Recalibrating Fully Convolutional Networks With Spatial and Channel 'Squeeze & Excitation' Blocks. *IEEE Transactions on Medical Imaging* 38(2): 540–549.

Sharma, B.; Das, R. K.; and Li, H. 2019. On the Importance of Audio-Source Separation for Singer Identification in Polyphonic Music. In *Proc. Interspeech*, 2020–2024.

Stöter, F.-R.; Uhlich, S.; Liutkus, A.; and Mitsufuji, Y. 2019. Open-Unmix - A Reference Implementation for Music Source Separation. *Journal of Open Source Software* [Online] https://sigsep.github.io/open-unmix/.

Sturm, B. L. 2014. A Simple Method to Determine if a Music Information Retrieval System is a "Horse". *IEEE Transactions on Multimedia* 16(6): 1636–1644.

Sundberg, J. 1989. *The Science of the Singing Voice*. Northern Illinois University Press.

Van, T. P.; Quang, N. T. N.; and Thanh, T. M. 2019. Deep Learning Approach for Singer Voice Classification of Vietnamese Popular Music. In *Proceedings of the Tenth International Symposium on Information and Communication Technology (SoICT)*, 255–260.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 5998–6008.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-Local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7794–7803.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *International Conference on Learning Representations (ICLR)*.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2019. Self-Attention Generative Adversarial Networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 7354–7363.

Zhang, X.; Gao, Y.; Yu, Y.; and Li, W. 2020. Music Artist Classification with WaveNet Classifier for Raw Waveform Audio Data. *arXiv preprint arXiv:2004.04371v1* .