IMPROVING THE PERFORMANCE OF AUTOMATED AUDIO CAPTIONING VIA INTEGRATING THE ACOUSTIC AND TEXTUAL INFORMATION

Technical Report

Zhongjie Ye[1], *Helin Wang*[1], *Dongchao Yang*[1], *Yuexian Zou*[1,2,*]

[1] ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China
{*zhongjieye@stu.pku.edu.cn*, *wanghl15@pku.edu.cn*
dongchao98@stu.pku.edu.cn, *zouyx@pku.edu.cn*}

ABSTRACT

This technical report describes an automated audio captioning (AAC) model for the the Detection and Classification of Acoustic Scenes and Events (DCASE) 2021 Task 6 Challenge. In order to utilize more acoustic and textual information, we propose a novel sequence-to-sequence model named KPE-MAD, with a **k**eyword **p**re-trained **e**ncoder and a **m**ulti-modal **a**ttention **d**ecoder. For the encoder, we use pre-trained classification model on the AudioSet dataset, and finetune it with keywords of nouns and verbs as labels. In addition, a multi-modal attention module is proposed to integrate the acoustic and textual information in the decoder. Our single model achieves the SPIDEr score of 0.279 in the evaluation splits. And our best ensemble model by optimizing CIDEr-D via the reinforcement learning, achieves the SPIDEr score of 0.291. Our code[1] and models will be released after the competition.

Index Terms— Audio caption, pre-training, multi-modal attention, keyword classification

1. INTRODUCTION

Automated audio captioning (AAC) is a new and challenging task that involves different modalities. It could be described as generating a textual description (i.e. caption) given an audio signal, where the caption should be as close as possible to a human-assigned one [1]. In contrast to automatic speech recognition which just converts speech to text, AAC converts environmental sound to text. It is also different from sound event detection (SED) [2] and audio tagging (AT) [3, 4] tasks, which output exact labels with start and end time or not. Generating accurate captions needs more information, including identification of sound events, acoustic scenes, spatio-temporal relationships of sources, foreground versus background discrimination, concepts, and physical properties of objects and environment [5].

One of the challenges of AAC is the lack of training data. Typical datasets in AAC, are Audio Caption [6], Audio Caps [7] and Clotho [5]. The Clotho [5] dataset is published by DCASE 2020 and expanded in DCASE 2021. Now it contains 5,929 audio samples and 29,645 captions. However, the scales of the datasets of AAC are quite small, comparing to datasets of image captioning, such as MS COCO [8] which contains one million captions and over 16k images.

Through previous work and competitions in AAC, there are amounts of algorithms proposed [6, 9, 10] based on sequence-to-sequence model. M. Wu *et al.* [6] straightly sent the output of encoder to the decoder. It would result in that acoustic information wouldn't be fully utilized. H. Wang *et al.* [10] proposed a temporal attention mechanism in the decoder, which could utilize more acoustic information for each time step. Both of them adopt a strategy of training the whole audio caption model directly, which would cause that the encoder couldn't sufficiently learn the representations of audios because of the lack of data. In addition, Y. Wu *et al.* [9] proposed a pre-training method in the task by extracting the top 300 words with the highest frequency, and achieved good results. Before training the whole audio captioning model, they pre-trained the convolutional neural network (CNN) encoder with 300 labels. However, the extracted words, through frequency, may contain some meaningless words such as until, onto, etc. Besides, the use of textual information could be further exploited.

To address the above issues, we propose a novel AAC model which combines the keyword pre-trained CNN encoder and a decoder with multi-modal attention module, named KPE-MAD. On the official evaluation splits of Clotho dataset [5], our proposed single model could achieve the SPIDEr score of 0.279 (baseline system is 0.051) and our best ensemble model could achieve the SPIDEr score of 0.291 by optimizing CIDEr-D via a reinforcement learning method, *i.e.* SCST [11].

The organization of the paper is as follows. Section 2 introduces our proposed KPE-MAD. We present our experimental results and evaluations in Section 3. Finally, we give concluding remarks and possible future directions in Section 4.

2. SYSTEM ARCHITECTURE

In this section, our proposed KPE-MAD model is introduced and its architecture is shown in Figure 1. Specifically, our KPE-MAD consists of a keyword pre-trained encoder and a multi-modal attention decoder. Firstly, the encoder is pre-trained with keywords which are extracted from captions in the training data. Then, we use the pre-trained encoder, multi-modal attention module which aligns the acoustic and textual information, and a decoder based on the long-short term memory (LSTM). In the following subsection, we will introduce details about KPE-MAD model.

*Yuexian Zou is the corresponding author

[1]`https://github.com/WangHelin1997/DCASE2021_Task6_PKU`

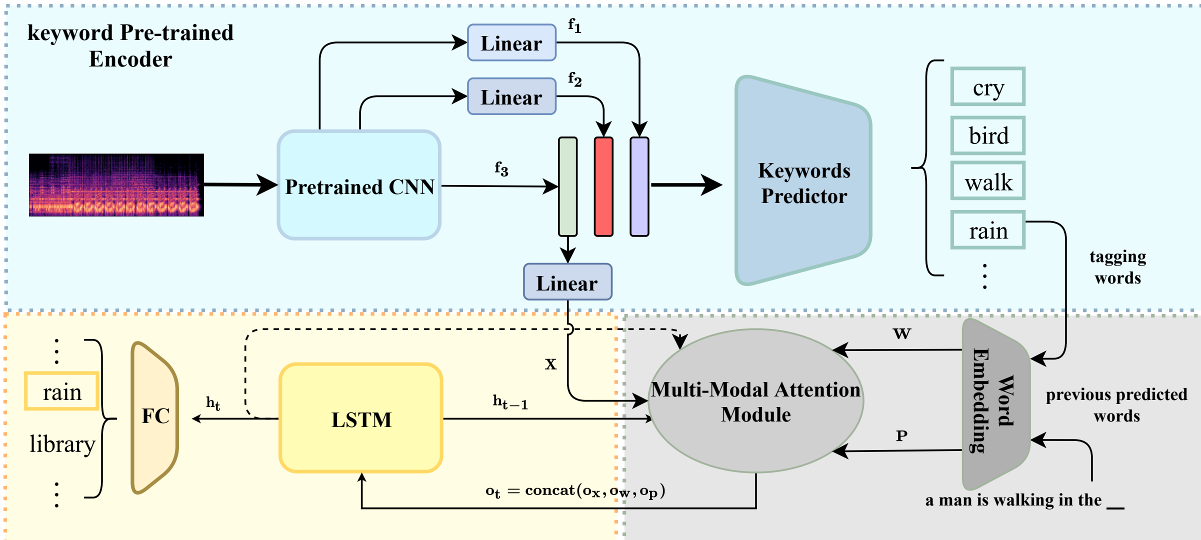

Figure 1: The architecture of our KPE-MAD caption model.

2.1. Keyword Pre-trained Encoder

The CNN encoder, which are widely used in the DCASE community [9, 10, 12, 13], plays an important role in extracting robust time-frequency information from raw audios. Meanwhile, with the development of large-scale pre-training approaches, lots of pre-trained models such as VGGish [14], PANNs [15] could improve the performance of downstream tasks. In this work, we use the pre-trained ResNet38[2] [15], which performs well in AudioSet dataset [16], as our backbone network. As Section 1 states, Y. Wu *et al.* [9] selected keywords by the highest frequency which is sometimes unreasonable. Apparently it is quite difficult for us to correctly recognize these adverbs and conjunctions from an audio sample. Instead, we extract some more meaningful words, such as nouns and verbs (e.g., bird, cry, etc.) as labels.

Firstly, Natural Language Toolkit (NLTK[3]) which is a powerful open-source tool is applied to extract words from each caption. And we choose the nouns and verbs, and get rid of some useless words through handcrafted useless vocabulary such as make, go, others, etc.

Then, the verbs in keywords vocabulary are transformed into their original forms and the nouns aren't changed, because their plural forms have different meanings. Finally, we choose N keywords with highest frequency from the modified keywords vocabulary, and use them as class labels for pre-training.

In the training phase of the encoder, we combine all the keywords form the 5 captions of each audio to form the training label which is a multi-hot vector. Each word of captions is transformed into their original forms according to the above rules. When a keyword occurs in the keywords vocabulary, the corresponding position of the multi-hot vector is set to 1, otherwise 0.

As Figure 1 illustrates, the pre-trained ResNet38 ($Conv\_1$ to $Conv\_6$) is used as our backbone, as detailed in Table 1, which consists of 6 convolution blocks. We refine it with fusion of multi-level features, *i.e.* the features after $Conv\_3$, $Conv\_4$ and $Conv\_6$. Then a global average pooling (GAP) layer and a linear layer are applied to each level of the feature, which are FC_1, FC_2 and FC_3. The last classification layer (CLS) utilizes the information combining the output of the three level features to classify keywords, which could enforce CNN model to learn more diversity of information.

More specifically, $Conv\_1$ consists two convolotional layers and a pooling layer applied to the log mel spectrogram. Each of $Conv\_2$ to $Conv\_5$ contains some basic blocks, which are mainly parts of ResNet38 and introduce shortcut connections between con-

Table 1: The architecture of the keyword pre-trained encoder(KPE). GAP means the global average pooling layer. Linear(128, 2048) means that the input dimension of the fully-connected layer is 128 and the output dimension is 2048. We take FC_1 as an example that the input features firstly go through the global average pooling layer, and then are passed into a fully-connected layer with ReLU activation function.

X	log mel spectrogram
Conv_1	(Conv 3 × 3 @ 64, BN, ReLU) ×2 Pooling 2 × 2
Conv_2	(BasicB @ 64) ×3 Pooling 2 × 2
Conv_3	(BasicB @ 128) ×4 Pooling 2 × 2
Conv_4	(BasicB @ 256) ×6 Pooling 2 × 2
Conv_5	(BasicB @ 512) ×3 Pooling 2 × 2
Conv_6	(Conv 3 × 3 @ 2048, BN, ReLU) ×2 Pooling 2 × 2
$\mathbf{FC_1}$	$GAP, Linear(128, 2048), ReLU$
$\mathbf{FC_2}$	$GAP, Linear(256, 2048), ReLU$
$\mathbf{FC_3}$	$GAP, Linear(2048, 2048), ReLU$
CLS	$Linear(6144, 300), Sigmoid$

[2] https://github.com/qiuqiangkong/audioset_tagging_cnn
[3] https://github.com/nltk/nltk

volutional layers, and a 2×2 pooling layer. We use f_1, f_2 and f_3 to represent the output of FC_1, FC_2 and FC_3 respectively, and $\hat{y}$ represents the output of CLS and GAP means global average pooling. Then we use f_1, f_2 and f_3 to obtain the predictions $\hat{y} \in \mathbb{R}^N$ where N is the number of keywords.

$$\hat{y} = \sigma(Linear(concat(f_1, f_2, f_3))) \tag{1}$$

Given the ground truth $y \in \mathbb{R}^N$, the keyword pre-trained encoder could be optimized by:

$$\mathcal{L}_{bce}(y, \hat{y}) = -\sum_{i=1}^{N} y(i) log\ \hat{y}(i) \tag{2}$$

where σ means sigmoid activation function, $\hat{y}$ is the output of the CLS. Standard binary cross entropy loss is used as the loss function, which is defined as the negative log likelihood of the expected keyword y_i given transcription $\hat{y}_i$ at the position i.

2.2. Multi-Modal Attention Decoder

Unlike the existing audio caption models, we further incorporate acoustic information with textual information into generating captions: we propose a multi-modal attention module to align them. The high-level representation of acoustic features is denoted as $\boldsymbol{X} = \{x_1, ..., x_L\} \in \mathbb{R}^{L \times C_1}$, which is the output of FC_3 of the keyword pre-trained encoder. The textual features contain the keywords $\boldsymbol{W} = \{w_1, ..., w_K\}$ that is the K outputs of keyword pre-trained encoder, and the previous words $\boldsymbol{P} = \{p_1, ..., p_{t-1}\}$ that contain all the generated words before time step t. Both of them are transformed into continuous vectors by randomly initialized embedding layer, $\boldsymbol{W} \in \mathbb{R}^{K \times C_2}$ and $\boldsymbol{P} \in \mathbb{R}^{(t-1) \times C_2}$. And we align the acoustic and textual information by a multi-modal attention module.

Firstly, they are transformed into the same latent space, where X is turned to $\hat{\boldsymbol{X}} \in \mathbb{R}^{T \times C}$, W becomes $\hat{\boldsymbol{W}} \in \mathbb{R}^{K \times C}$ and P becomes $\hat{\boldsymbol{P}} \in \mathbb{R}^{(t-1) \times C}$. Then the hidden states as intermediaries connect $\hat{\boldsymbol{X}}$, $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{P}}$, by a attention mechanism that is shown in Figure 2. Taking the acoustic information for example: given the previous time step LSTM hidden state h_{t-1}, we use a single fully-connected layer followed by a softmax function to generate the attention distributions α of acoustic features in time axis. Finally, the gated linear unit (GLU) [17] is applied to the output of the attention module, to control how much information should flow into the next layer. Bellow are the definitions of acoustic attention module $\boldsymbol{\Psi_x}$:

$$\boldsymbol{A} = ReLU((\hat{\boldsymbol{X}}\boldsymbol{W_i^T} + b_i) \oplus (h_{t-1}\boldsymbol{W_s^T} + b_s)) \tag{3}$$

$$\alpha = softmax(\boldsymbol{A}\boldsymbol{W_n} + b_n) \tag{4}$$

$$o_x = GLU([\hat{\boldsymbol{X}} \otimes \boldsymbol{\alpha}, h_{t-1}]) \tag{5}$$

where $\boldsymbol{W_s} \in \mathbb{R}^{M \times H}$, $\boldsymbol{W_i} \in \mathbb{R}^{M \times C}$, $\boldsymbol{W_n} \in \mathbb{R}^{M}$ are transformation matrixes that map acoustic features and hidden states to the same dimension. Here, $b_s \in \mathbb{R}^M$, $b_i \in \mathbb{R}^M$, and $b_n \in \mathbb{R}^1$. We denote $\oplus$ as the element-wise addition of a matrix and a vector, and $\otimes$ as the element-wise multiplication of a matrix and a vector. The output $o_x \in \mathbb{R}^C$. For GLU [17] operation, it implements a simple gating mechanism over the output $\mathcal{Y} = [\mathcal{A}, \mathcal{B}] \in \mathbb{R}^{2d}$:

$$GLU([\mathcal{A}, \mathcal{B}]) = \mathcal{A} \otimes \sigma(\mathcal{B}) \tag{6}$$

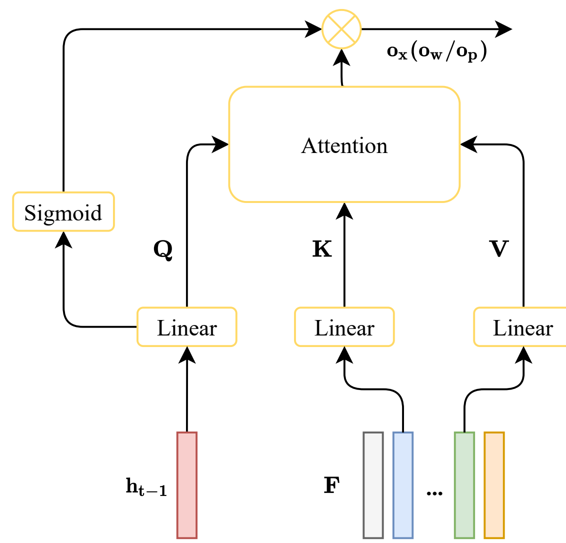

Figure 2: The architecture of the attention mechanism. F could represent acoustic features or textual features.

where $\mathcal{A} \in \mathbb{R}^d, \mathcal{B} \in \mathbb{R}^d$ are the inputs to the non-linearity, $\otimes$ is the point-wise multiplication and the output $GLU([\mathcal{A}, \mathcal{B}]) \in \mathbb{R}^d$ is half the size of $\mathcal{Y}$. The gates $\sigma(\mathcal{B})$ control which inputs $\mathcal{A}$ of the current context are relevant [18].

As for the textual information, the same structure of attention module is applied to keywords and previous words, and the outputs are $o_w \in \mathbb{R}^C$, $o_p \in \mathbb{R}^C$ respectively. Finally, we add them with o_x. Then, they are sent to calculate hidden state in current time and probabilities of each word:

$$\begin{aligned} h_0 &= GAP(\hat{\boldsymbol{X}}) \\ h_t &= LSTM(h_{t-1}, Add(o_x, o_w, o_p)) \\ v_t &= Linear(h_t) \end{aligned} \tag{7}$$

where h_0 represents global information of acoustic features in the time dimension. $v_t \in \mathbb{R}^{|\Sigma|}$ is a probability vector, and $|\Sigma|$ is a predefined dictionary including all words. Then, the current word is chosen with the highest probability and added to previous words $\boldsymbol{P}$ for the next iteration of LSTM.

2.3. Data Augmentation

In order to avoid over-fitting and increase data diversity, SpecAugment [19], SpecAugment++ [20], Mixup [21] and Label smoothing [22] are used in the training phase. For Mixup method, it is just used in the training of the keyword encoder. And label smoothing is just used while training the whole captioning model.

3. EXPERIMENT

Experiment setups: We choose $N = 300$ keywords for pre-training encoder and the dimension of fully-connected layers C_1 and C_2 are 512. The decoder LSTM has 512 hidden units, word embedding size is also set to 512. To mitigate overfitting, dropout regularization is used in the word embedding layer with a rate of 0.5, and LSTM decoder layers with a rate of 0.25. In the phase of training the encoder, firstly the CNN backbone is frozen up, trained

Table 2: The performance of different models in Clotho [5] evaluation splits

Model	BLEU1	BLEU2	BLEU3	BLEU4	ROUGEL	METEOR	CIDEr	SPICE	SPIDEr
Baseline [5]	0.378	0.119	0.050	0.017	0.263	0.078	0.075	0.028	0.051
Temporal attention model [10]	0.489	0.285	0.177	0.107	0.325	0.148	0.252	0.091	0.172
Transformer model [9]	0.534	0.343	0.230	0.151	0.356	0.160	0.346	0.108	0.227
KPE-MAD (w/o rl)	0.578	0.381	0.257	0.169	0.381	0.181	0.433	0.125	0.279
KPE-MAD (w/ rl)	0.579	0.384	0.261	0.172	0.386	0.181	0.436	0.128	0.282
KPE-MAD_ensemble (w/o rl)	0.586	0.391	0.268	0.180	0.388	0.180	0.440	0.125	0.282
KPE-MAD_ensemble (w/ rl)	**0.590**	**0.395**	**0.272**	**0.183**	**0.394**	**0.182**	**0.453**	**0.129**	**0.291**

by Adam optimizer with the initial learning rate of 1×10^{-3}. We then finetune the whole keyword encoder with the learning rate of 5×10^{-4}. Next, the strategy of training the whole caption model is the same as the keyword pre-trained encoder, and the difference is that the multi-modal attention decoder is trained for 30 epochs with the learning rate of 3×10^{-4} and finetuned for 15 epochs with the learning rate of 2×10^{-5}. Finally, we optimize CIDEr-D score with SCST [11] for another 10 epochs with an initial learning rate of 1×10^{-6}. In the inference stage, we adopt beam search with a beam size of 4 that is implemented to achieve best decoding performance.

Experimental results: We compare performance of our model with baseline model [5], a temporal attention model [10] and a transformer model [9]. The results are shown in table 2, which demonstrate that our proposed model has a great improvement over previous models. Our single KPE-MAD model achieves a SPIDEr score of 0.279. KPE-MAD(w/ scst) uses SCST [11] to optimize the CIDEr-D and achieves 0.282. Then we ensemble three KPE-MAD models which are trained with different seeds, with or without reinforcement learning, which achieve 0.282 and 0.291, respectively. Comparing with other state-of-the-arts, our proposed method with keyword pre-training encoder and multi-modal attention decoder can obviously improve the performance of AAC.

4. CONCLUSION

The technical report describes our proposed KPE-MAD model, which focuses on fusing multi-modal information by introducing keyword pre-trained encoder and multi-modal attention decoder. In the future work, we would concentrate on how to align the multi-modal information more effectively to improve the performance of the AAC.

5. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, “Automated audio captioning with recurrent neural networks,” in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 374–378.

[2] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[3] H. Wang, Y. Zou, D. Chong, and W. Wang, “Modeling label dependencies for audio tagging with graph convolutional network,” *IEEE Signal Processing Letters*, vol. 27, pp. 1560–1564, 2020.

[4] H. Wang, Y. Zou, and W. Wang, “A global-local attention framework for weakly labelled audio tagging,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 351–355.

[5] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[6] M. Wu, H. Dinkel, and K. Yu, “Audio caption: Listen and tell,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 830–834.

[7] C. D. Kim, B. Kim, H. Lee, and G. Kim, “Audiocaps: Generating captions for audios in the wild,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[8] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” *arXiv preprint arXiv:1504.00325*, 2015.

[9] Y. Wu, K. Chen, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, “Audio captioning based on transformer and pre-training for 2020 dcase audio captioning challenge,” DCASE2020 Challenge, Tech. Rep, Tech. Rep., 2020.

[10] H. Wang, B. Yang, Y. Zou, and D. Chong, “Automated audio captioning with temporal attention,” Tech. Rep., DCASE2020 Challenge, Tech. Rep., 2020.

[11] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008–7024.

[12] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, “A transformer-based audio captioning model with keyword estimation,” *arXiv preprint arXiv:2007.00222*, 2020.

[13] H. Wang, Y. Zou, D. Chong, and W. Wang, “Environmental sound classification with parallel temporal-spectral attention,” *Proc. Interspeech 2020*, pp. 821–825, 2020.

[14] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, “Cnn architectures for large-scale audio

classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[15] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[16] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[17] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 933–941.

[18] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1243–1252.

[19] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[20] H. Wang, Y. Zou, and W. Wang, "Specaugment++: A hidden space data augmentation method for acoustic scene classification," *arXiv preprint arXiv:2103.16858*, 2021.

[21] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.