Interspeech 2024
1-5 September 2024, Kos, Greece

PFCA-Net: Pyramid Feature Fusion and Cross Content Attention Network for Automated Audio Captioning

Jianyuan Sun[1], *Wenwu Wang*[2], *Mark D. Plumbley*[2]

[1]Department of Computer Science, University of Sheffield, UK
[2]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK
jianyuan.sun@sheffield.ac.uk, w.wang@surrey.ac.uk, m.plumbley@surrey.ac.uk

Abstract

Automated audio captioning (AAC) aims to generate textual descriptions for a given audio clip. Despite the existing AAC models obtaining promising performance, they struggle to capture intricate audio patterns due to only using a high-dimensional representation. In this paper, we propose a new encoder-decoder model for AAC, called the Pyramid Feature Fusion and Cross Context Attention Network (PFCA-Net). In PFCA-Net, the encoder is constructed using a pyramid network, facilitating the extraction of audio features across multiple scales. It achieves this by combining top-down and bottom-up connections to fuse features across scales, resulting in feature maps at various scales. In the decoder, cross-content attention is designed to fuse the different scale features which allows the propagation of information from a low-scale to a high-scale. Experimental results show that PFCA-Net achieves considerable improvement over existing models.
Index Terms: Pyramid feature fusion, high-dimensional representation, cross-context attention network

1. Introduction

Automated audio captioning (AAC) is the task of generating text descriptions of an audio clip, which can be performed in real-time, as the audio is being played, or offline, on pre-recorded audio files. It is a useful tool for providing accessibility to media for the impaired of hearing, generating subtitles for audio in a television program, and for content translation and summarization [1, 2].

A popular approach to AAC is to use an encoder-decoder architecture, where the encoder is used to extract features from an audio signal, while the decoder is used to generate text descriptions based on the audio features. In the early years, a recurrent neural network (RNN) [3] is used in the encoder to extract the audio features. For example, Drossos et al. [1] use a multi-layered and bi-directional gated recurrent unit (GRU) as the encoder, and a multi-layered GRU as the decoder. Subsequently, convolutional neural networks (CNNs) are applied in audio captioning, such as the use of convolutional recurrent neural network (CRNN) as the encoder by combining CNNs and RNNs in [4], and the use of the VGG network [5] by Kim et al. [6]. Moreover, Mei et al. [7] use a pre-trained CNN model, called PANNs, pre-trained on a large dataset AudioSet [8], as the encoder, and then a transformer with multi-head attention as the decoder to predict the captions according to the audio features.

Recently, Liu et al. [9] use contrastive learning to improve audio captioning performance in a data scarcity scenario. Chen et al. [10] introduce an audio-head and a text-head to extract audio and text information in the pre-trained encoder. Both methods aim to learn embeddings that are close together for positive audio-text pairs and far apart for negative audio-text pairs, by combining the cross-entropy loss with a contrastive loss [9, 10]. In addition, Xu et al. [11] explore the local and global audio information by using transfer learning. Most existing methods use a shallow transformer decoder (with only two transformer blocks) due to the limited amount of data available [7, 12, 13]. Deep transformer architectures, from natural language processing (NLP), are also used for AAC. For example, Xu et al. employed the pre-trained BERT [14] as the decoder to generate the captions [15]. Yuma et al. [16] introduced a cascaded system that utilizes a pre-trained large-scale language model to direct the generation of audio captions. This approach assists in alleviating the challenges stemming from the limited training data available for audio captioning tasks.

Although existing methods achieve promising results, most of them encode the audio signals with a high-dimensional representation. This may not be sufficient as audio signals contain acoustic events, scenes, and noise and clutter with different scales. In this paper, we propose a new encoder-decoder model called the Pyramid Feature Fusion and Cross Context Attention Network (PFCA-Net) for AAC, motivated by the success of the feature pyramid networks (FPN) developed for object detection tasks [17]. In PFCA-Net, we design an improved FPN in the encoder for extracting multi-scale audio features, by upsampling lower-resolution feature maps with a bottom-up pathway and then combining these upsampled maps with higher-resolution feature maps from the CNN with a top-down pathway [17, 18]. This allows the network to capture contextual information from a wide range of scales, thus offering advantages over traditional CNNs. In addition, we design a cross-content attention scheme to fuse the features at different scales, which enables effective information propagation among them.

The main contributions of this work are summarized as follows: (1) a new encoder-decoder model is proposed called PFCA for AAC. In the encoder, an improved FPN is proposed that consists of a combination of the bottom-up, the top-down, and the lateral connections to fuse features across scales and achieve a high-level feature map at different scales. Our work is also the first attempt to use a pyramid neural network to learn the feature embedding for audio captioning. (2) In the decoder, cross-context attention is designed to integrate the features of different scales, which allows effective information propagation between the multi-scale features.

The remainder of the paper is organized as follows. Section 2 discusses the proposed PFCA-Net in detail. Section 3 presents experimental results. Finally, Section 4 concludes the paper.

10.21437/Interspeech.2024-1268

2. Proposed method

In this section, we introduce the proposed PFCA-Net, which consists of a feature pyramid network (FPN) encoder and a decoder based on a transformer with cross-content attention, as shown in Fig. 1.

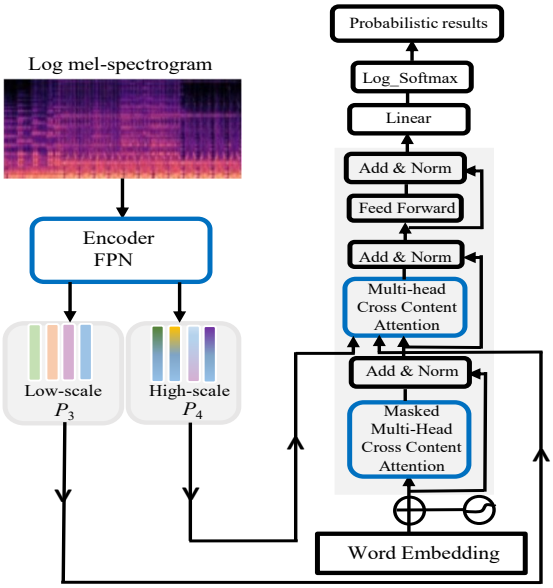

Figure 1: *The architecture of the proposed PFCA-Net.*

2.1. Encoder: Feature Pyramid Network (FPN)

The original FPN [17] is based on ResNets [19], and the construction of the pyramid network includes a bottom-up pathway, a top-down pathway, and lateral connections, where the bottom-up pathway involves computing a feature hierarchy consisting of feature maps at different scales with a scaling step of 2. This bottom-up pathway is also called the feed-forward computation of the backbone. In particular, many layers produce output maps of the same size and these layers are called in the same network stage. In the original FPN, it designs one pyramid level for each stage and chooses the output of the last layer of each stage. Because the deepest layer of each stage has a feature with more information [17]. In the ResNets [19], it uses the feature output of the last residual block of each stage.

For the top-down pathway and lateral connections of the original FPN, the top-down pathway creates higher resolution features by upsampling feature maps with stronger semantic information obtained from higher pyramid levels. These features are improved by incorporating information from the bottom-up pathway through lateral connections. Each lateral connection combines feature maps of the same spatial size from both the bottom-up and top-down pathways. While the bottom-up feature map has lower-level semantics, its activations are more accurately localized since it underwent fewer subsampling steps. The upsample spatial resolution uses a factor of 2 by employing the nearest neighbor upsampling for simplicity. Then, the upsampled map is merged with the corresponding bottom-up map by performing a 1×1 convolutional layer for the lateral connections to reduce channel dimensions firstly and using element-wise addition.

Inspired by the original FPN [17, 18], the structure of the improved FPN encoder is shown in Fig. 2. In this paper, the improved FPN is based on pre-trained CNN10, i.e., PANNs [20]. It also has the bottom-up pathway C_i, the top-down pathway M_i, and the lateral connections L_i to construct pyramid levels. Where PANNs consist of four convolutional blocks and two linear layers. Each convolutional block has two layers with a kernel size 3×3, followed by the normalization and ReLU activation layer. The channel numbers for these blocks are $64, 128, 256$, and 512. Additionally, a 2×2 average pooling is applied for downsampling. After the final convolutional block, a global average pooling is performed, followed by two linear layers to produce the final feature representation.

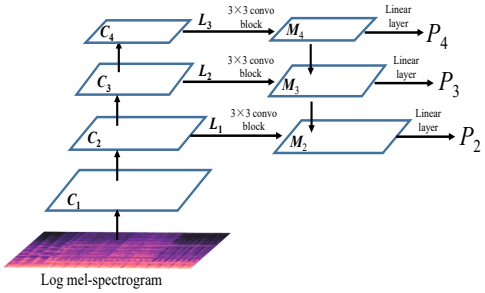

Figure 2: *The architecture of the proposed FPN encoder in PFCA-Net.*

In Fig. 2, it can be observed that the improved FPN has the bottom-up pathway C_i, the top-down pathway M_i, and the lateral connections L_i (need to note in the figure). The bottom-up pathway C_i is the feed-forward computation of the output of i-th stage convolutional block. Furthermore, the symbol $i = 0, \cdots, I$ represents the i-th stage convolutional block of the network, where $i = 0$ indicates the input log mel-spectrogram image and $I = 4$ is the total number of stages in the improved FPN. To enhance performance, each lateral connection L_i merges feature maps of identical spatial dimensions obtained from both the bottom-up pathway C_i and the top-down pathway M_i. Therefore, the shallow feature layers M_2 and M_3 also contain the high-level information as the deep feature layer M_4. The top-down feature map M_i is computed by M_{i+1} and C_i as follows,

$$M_i = \begin{cases} F_i(M_{i+1}, L_i(C_i)), & i < I \\ L_i(C_i), & i = I. \end{cases}$$

The fusion operation F_i comprises two essential steps. Initially, M_{i+1} undergoes an upsampling process. Subsequently, the upsampled M_{i+1} is integrated with lateral information through element-wise addition. The lateral information is derived from lateral connections, which involve a 3×3 convolutional block operation. Following each feature representation M_i (where $i = 2, 3, 4$), a linear layer is applied to produce the final feature representations P_i with dimensions $1024, 256$, and 128, corresponding to $i = 2, 3, 4$. In contrast to the original FPN [17], our FPN employs a 3×3 convolution block operation rather than the 1×1 convolution operation in lateral connections. Additionally, we employ a linear layer to obtain P_i from M_i instead of utilizing the 3×3 convolution operation in the original FPN.

2.2. Decoder: Transformer with Cross-Content Attention

The PFCA-Net decoder includes three parts, i.e., the word embedding layer, a transformer block, and a linear layer. The input

words are coded through the word embedding layer into word vectors of fixed dimensions and then fed into the transformer decoder. The word vectors are obtained through a pre-trained Word2Vec model on all caption corpus [21]. The decoder of PFCA-Net is a transformer block that uses multi-head attention with a fixed number of heads. We set the number of heads as 4 and the dimension of the hidden layer is 128 in the decoder of PFCA-Net.

For each head, we proposed cross-content attention based on the multi-scale feature outputs from the encoder. More specifically, the high-scale feature P_4 output by the FPN encoder, the input ground truth, and the low-scale feature P_3 are transformed into query set Q, key set K, and value set V, respectively, through matrix multiplication with three learnable matrices W_Q, W_K, $W_V \in \mathbb{R}^{d \times d_k}$, where d is the dimension of the features and d_k is the dimension of the attention heads. Before the high-scale feature P_4 and the low-scale feature P_3 transformer into query set Q and value set V, a linear layer is applied to make P_4 and P_3 have the same dimension that is 128. Then, the dot-product attention is calculated as

$$Attn(Q, K, V) = Soft\max(\frac{QK^T}{\sqrt{d_k}})V.$$

The outputs of multi-heads are ensembled by a linear transformation matrix $W_o \in \mathbb{R}^{(h \times d_k) \times d_k}$, as follows

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W_o.$$

The difference between the original self-attention and our cross-content attention lies in the method for computing the attention. For the original self-attention, the query set Q and the value set V are both calculated from high-dimensional features. In our cross-content attention, the query set Q and the value set V are calculated from different scale features. In this way, the proposed cross-content attention enables PFCA-Net to better understand the relationships among different scale features and more effectively capture semantic information among features.

3. Experiments

In this section, we perform experimental studies of the proposed FPCA-Net and compare it with other existing methods for AAC on two public datasets, i.e., the Clotho [22] and AudioCaps [6] datasets.

3.1. Datasets

Clotho [22] is a well-known audio captioning dataset containing audio clips of various lengths between 15 and 30 seconds collected from the Freesound archive. The audio clips are accompanied by 5 captions annotated by different Amazon Mechanical Turk employees, with each caption containing 8-20 words. In our experiment, we use the Clotho v2 dataset, specifically the version released for Task 6 of the DCASE 2021 Challenge. The Clotho v2 dataset is split into three parts: a development set with 3839 samples, a validation set with 1045 samples, and an evaluation set with 1045 samples. To comply with the settings of the baseline methods, we merge the development and validation sets to form a training set of 4884 samples. The evaluation set is used as the test set. In addition, audio clips are combined with one of their five captions as a training sample in the training set.

AudioCaps [6] is a large audio captioning dataset that includes $50,000$ audio clips, each with a duration of 10 seconds. The dataset is split into three parts, with $49,274$ audio clips for training, 497 clips for validation, and 957 clips for testing. Each audio clip in the training set has one caption and each audio clip in the validation and test sets has 5 captions. Each caption contains 3 to 20 words.

3.2. Data Pre-Processing

For each audio clip, a 1024-point Hanning window with a hop size of 512 points is employed to obtain 64-dimensional log mel-spectrograms as the input of the proposed FPCA-Net. In addition, we apply the SpecAugment method to augment the log mel-spectrogram of audio clips in the training data using "zero-value masking" and "mini-batch based mixture masking". The captions in the Clotho and AudioCaps datasets are converted to lowercase, with punctuation removed. We also add two special tokens, "<sos >" and "<eos >", at the beginning and end of each caption.

3.3. Experimental Setups

We train the proposed FPCA-Net model by employing the Adam optimizer [23]. The batch size is set to 32. The training epoch is 30 with an initial learning rate of 5×10^{-4}. Additionally, the pre-trained Word2Vec model [24] is employed to obtain the word embedding for all the captions in the Clotho and AudioCaps datasets.

3.4. Performance Metrics

To evaluate the performance of the models, we use multiple metrics including machine translation metrics such as BLEU [25], METEOR [26], and ROUGE [27], as well as captioning-specific metrics such as CIDEr [28], SPICE [29], and SPIDEr [30]. BLEU measures the n-gram precision of the generated text, METEOR is a word-to-word matching metric that calculates the harmonic mean of recall and precision, and ROUGE is an F-measure based on the longest common subsequence. CIDEr uses the term frequency-inverse document frequency to calculate the score, SPICE uses captions from scene graphs to determine the F-score, and SPIDEr is the mean score of CIDEr and SPICE.

3.5. Compared Models

For the Clotho dataset, we compare the proposed FPCA-Net with four existing models. The first one is the PANNs-Trans model [7] which is an encoder-decoder framework consisting of a pre-trained CNN10 encoder called PANNs and a transformer decoder. The second is the CL4AC model [9] which aims to correct the inaccurate audio-text alignment with a contrastive learning loss, based on the PANNs-Trans model [7]. The third one is the AT-CNN10 [11] model that explores the local and global audio information by using transfer learning based on audio tagging and acoustic scene classification techniques. The fourth model is the PreCNN-Transformer [12] model, which is similar to the PANNs-Trans model [7], but its encoder is a CNN pre-trained for an acoustic event tagging task.

For the AudioCaps dataset, we also compared with the PANNs-Trans [7] and the AT-CNN10 [11]. In addition, we used the Pre-Bert model [15], GPT-2 model [15], and TopDown-Att [6]. Here, Pre-Bert is a transformer model using the Pre-trained BERT [14] from the Natural Language Processing (NLP).

To verify the impact of different scale features on model performance, the results on Clotho and AudioCaps datasets us-

Table 1: *The experimental results on the Clotho and AudioCaps datasets.*

Dataset	Model	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	METEOR	$CIDE_r$	SPICE	$SPIDE_r$
Clotho	PANNs-Trans [7]	0.561	0.364	0.243	0.159	**0.375**	0.172	0.391	0.120	0.256
	CL4AC model [9]	0.553	0.349	0.226	0.143	0.374	0.168	0.368	0.115	0.242
	AT-CNN10 [11]	0.556	0.363	0.242	0.159	0.368	0.169	0.377	0.115	0.246
	PreCNN-Transfor [12]	0.534	0.343	0.230	0.151	0.356	0.160	0.346	0.108	0.227
	PFCA-Net (P_4 and P_2)	0.555	0.356	0.234	0.147	0.167	0.369	0.359	0.119	0.239
	PFCA-Net (our model)	**0.564**	**0.366**	**0.246**	**0.160**	**0.375**	**0.174**	**0.401**	**0.123**	**0.262**
AudioCaps	PANNs-Trans [7]	0.667	0.491	0.350	0.248	0.468	0.229	0.643	0.165	0.404
	Pre-Bert [15]	0.667	0.491	0.354	0.247	0.475	0.232	0.654	0.167	0.410
	AT-CNN10 [11]	0.655	0.476	0.335	0.231	0.467	0.229	0.660	0.168	0.414
	GPT-2 [16]	0.655	0.476	0.335	0.231	0.467	0.229	0.660	0.168	0.414
	TopDown-Att [6]	0.614	0.446	0.317	0.219	0.450	0.203	0.593	0.144	0.369
	PFCA-Net (P_4 and P_2)	0.671	0.501	0.367	0.264	0.231	0.482	0.655	0.168	0.411
	PFCA-Net (our model)	**0.678**	**0.507**	**0.374**	**0.268**	**0.486**	**0.234**	**0.698**	**0.173**	**0.436**

Table 2: *The generated captions for the test audio clips from the AudioCaps dataset.*

Audio clip	Yti66RjZWTp0.wav	Yz4uELRI6p08.wav
Ground truth	1. a male speaks as metal clicks and a gun fires once.	1. loud **laugh** ting and **mumbling** with s person laughing faintly and briefly in the distance.
	2. a man speaks and a weapon cocks and fires.	2. an older **woman laughs** and **titters**.
	3. a man speaks while loading a gun cocking it and shooting.	3. a woman makes **noises** and laughs happily.
	4. a **male speaks** as metal **clicks** and a **gun** fires once.	4. laughing and some mumbling.
	5. a man speaks and then gunfire takes place.	5. laughing followed by a short groan then more laughing.
PANNs-Trans	a man speaks followed by several gunshots.	a person laughs and then laughs.
PFCA-Net	a **man speaks** followed by several loud **clicks** (our model) and a **gun** shots.	a **woman laughs** and **speaks**.

Table 3: *The generated captions for the test audio clips from the Clotho dataset.*

Audio clip	TheGym.wav
Ground truth	1. **people converse** in a very large echoing room.
	2. **a group of people indistinctly chatter** in the background.
	3. in the background **a group of people indistinctly chatter**.
	4. **an inaudible group of people converse** in a very **large echoing room**.
	5. **many people talking** in a **enclosed space bar** or **restaurant** while **music plays**.
PANNs-Trans	**many people talking** in a **restaurant**.
PFCA-Net (our model)	**a large group of people** are **talking** in a **restaurant** with **music plays**.

ing the different scale feature combinations based on the proposed model are also given in Table 1. Moreover, in Table 2 and Table 3, we show some predicted captioning results for the PANNs-Trans model [7] and the proposed FPCA-Net.

3.6. Results

Table 1 shows the performance of the proposed FPCA-Net and the compared algorithms. Note that, for a fair comparison, the results of the PANNs-Trans model without reinforcement learning are reported in Table 1. The compared models are all based on the encoder-decoder architecture. The encoder aims to extract the high-scale acoustic feature, while the decoder generates textual descriptions solely based on this high-scale representation. From Table 1, we see that our proposed FPCA-Net outperforms other existing models that only use high-scale representation. The FPCA-Net obtains the best performance compared to the existing models. Moreover, Table 2 and Table 3 also illustrate that the proposed FPCA-Net model can capture more scene information.

As described in the subsection 2.2, we know that the proposed FPCA-Net mainly uses the feature P_4 and P_3. In Table 1, we also give the results that the proposed FPCA-Net uses the feature P_4 and P_2 instead of using features P_4 and P_3, i.e., FPCA-Net (P_4 and P_2). From the results, the proposed FPCA-Net uses the features P_4 and P_3 to perform better than using the features P_4 and P_2. This result also verifies the fact of the limited representation ability of lower-scale features. However, under this result, the performance of our proposed FPCA-Net uses P_4 and P_2 is better than the existing models only using the high-dimensional feature in most cases. In particular, the performance of the proposed FPCA-Net using features P_4 and P_2 outperforms other existing models on the AudioCaps dataset.

4. Conclusion

We have presented a new encoder-decoder model called FPCA-Net for AAC. Inspired by the success of the pyramid neural network in the object detection tasks, we improved the pyramid network and used it as the encoder of FPCA-Net to extract the multi-scale audio features. While FPNs have achieved considerable success in the visual domain, their application in the audio field has been limited. Our work is the first attempt to use the pyramid network in the audio captioning task. Moreover, to fully learn the multi-scale feature information in the decoder, we have designed cross-content attention to fuse the features of different scales, which allows effective information to be propagated from low-scale to high-scale features. Experimental results have shown that FPCA-Net outperforms other existing algorithms on Clotho and AudioCaps datasets.

5. Acknowledgment

This work was partly supported by a Go Fund Award from the Grantham Centre for Sustainable Futures of the UK, titled 'A New Mobile Technology of Hearing-impaired' (Grant number X/008802-16-58), and a Grant EP/T019751/1 'AI for Sound' from the Engineering and Physical Sciences Research Council (EPSRC) of the UK. For open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising. This publication is supported by multiple datasets, which are openly available at the locations referenced in this paper.

6. References

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

[2] F. Zalkow and M. Müller, "CTC-Based learning of chroma features for score–audio music retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2957–2971, 2021.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[4] J. Sun, Z. Wang, H. Yu, S. Zhang, J. Dong, and P. Gao, "Two-stage deep regression enhanced depth estimation from a single RGB image," *IEEE Transaction Emerging Topics in Computing*, vol. 10, no. 2, pp. 719–727, 2022.

[5] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, "CNN architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 131–135.

[6] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: generating captions for audios in the wild," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 119–132.

[7] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. Zhao, S. Li, T. Ko, H. Tang, X. Shao, M. Plumbley, and W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021, pp. 206–210.

[8] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "AudioSet: an ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 776–780.

[9] X. Liu, Q. Huang, X. Mei, T. Ko, H. Tang, M. D. Plumbley, and W. Wang, "CL4AC: A contrastive loss for audio captioning," in *Detection and Classification of Acoustic Scenes and Events 2021*, 2021, pp. 196–200.

[10] C. Chen, N. Hou, Y. Hu, H. Zou, X. Qi, and E. S. Chng, "Interactive auido-text representation for automated audio captioning with contrastive learning," in *Conference of the International Speech Communication Association*, 2022, pp. 2773–2777.

[11] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, "Investigating local and global information for automated audio captioning with transfer learning," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 905–909.

[12] K. Chen, Y. Wu, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on transformer and pre-trained CNN," in *Detection and Classification of Acoustic Scenes and Events*, 2020, pp. 21–25.

[13] Q. Han, W. Yuan, D. Liu, X. Li, and Z. Yang, "Automated audio captioning with weakly supervised pre-training and word selection methods," in *Detection and Classification of Acoustic Scenes and Events*, 2021, pp. 6–10.

[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[15] X. Liu, X. Mei, Q. Huang, J. Sun, J. Zhao, H. Liu, M. D. Plumbley, V. Kilic, and W. Wang, "Leveraging pre-trained BERT for audio captioning," in *European Signal Processing Conference*. IEEE, 2022, pp. 1145–1149.

[16] Y. Koizumi, Y. Ohishi, D. Niizumi, D. Takeuchi, and M. Yasuda, "Audio captioning using pre-trained large-scale language model guided by audio-based similar caption retrieval," *arXiv preprint arXiv:2012.07331*, 2020.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[18] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-Net: pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7448–7458, 2020.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[20] 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 Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013.

[22] K. Drossos, S. Lipping, and T. Virtanen, "CLOTHO: an audio captioning dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 736–740.

[23] D. P. Kingma and J. Ba, "ADAM: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Association for Computational Linguistics*, 2002, pp. 311–318.

[26] A. Agarwal and A. Lavie, "METEOR: an automatic metric for mt evaluation with high levels of correlation with human judgments," *Proceedings of WMT-08*, 2007.

[27] L. C. ROUGE, "A package for automatic evaluation of summaries," in *Proceedings of Workshop on Text Summarization of ACL*, 2004.

[28] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDER: consensus-based image description evaluation," in *IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[29] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: semantic propositional image caption evaluation," in *European conference on computer vision*, 2016, pp. 382–398.

[30] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of spider," in *IEEE international conference on computer vision*, 2017, pp. 873–881.