Expert Systems With Applications 231 (2023) 120664

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Using various pre-trained models for audio feature extraction in automated audio captioning

Hyejin Won [a,1], Baekseung Kim [a,1], Il-Youp Kwak [a,1], Changwon Lim [a,b,*]

[a] Department of Applied Statistics, Chung-Ang University, Seoul, 06974, Republic of Korea
[b] Institute for Community Care and Health Equity, Chung-Ang University, Seoul, 06974, Republic of Korea

ARTICLE INFO

Keywords:
Audio captioning
Acoustic scene detection
Transfer learning
Encoder–decoder
Convolutional neural network
Transformer

ABSTRACT

The DCASE automated audio captioning challenge aimed to construct a model that generates captions describing given audio. Our team developed a CNN14 encoder (pre-trained on AudioSet data) along with a Transformer decoder model that ranked sixth place in the competition. Many teams utilized pre-trained networks, and it was evident that more research into their utilization was required. This paper presented comprehensive experiments conducted with various encoder networks for the proposed system, including CNN10, CNN14 ResNet54, AST, VGGNet, and EfficientNet. The pre-trained networks of CNN10, CNN14, ResNet54, and AST were trained on AudioSet data, while the pre-trained networks of AST, VGGNet, and EfficientNet were trained on ImageNet data. The best outcomes were achieved when the pre-trained CNN10, trained on AudioSet data, was utilized as an encoder with the Transformer serving as a decoder, and fine-tuning applied. Moreover, a qualitative study confirmed that our model generates plausible captions for different types of audio.

1. Introduction

In the field of computer vision, image captioning has garnered significant attention. The objective is to generate a caption that accurately describes the image from a given photograph. Similarly, in the audio domain, research on audio captioning has been pursued with a similar goal. Image captioning can be particularly helpful for individuals with visual impairments, as it provides them with crucial information about their surroundings (Makav & Kılıç, 2019). Likewise, audio captioning can be utilized to provide a textual description of audio signals to those who are deaf or hard of hearing. As captioning technologies continue to evolve, they hold great potential for enhancing the quality of life for individuals with disabilities. Recent research in audio processing highlights the significance of recognizing and explaining nonverbal sounds, rather than simply translating them into other languages, as seen in closed captions for audiovisual content on streaming platforms such as Netflix (Xu, Wu, & Yu, 2022). This paper aims to provide a comprehensive overview of the current state and future prospects of captioning technology in both computer vision and audio processing.

Automated Audio Captioning (AAC) is a system that uses machine learning approaches to develop captions that explain the given audio data, as shown in Fig. 1. For instance, when presented with the sound of thunder or dripping rain, the AAC System generates the caption 'Rain is pouring down while thunder is occurring.' This fascinating subject has been the focus of many researchers who participated in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenges and workshops in 2020 and 2021, including (Chen et al., 2020; Han, Yuan, Liu, Li, & Yang, 2021; Pellegrini, 2020; Perez-Castanos, Naranjo-Alcazar, Zuccarello, & Cobos, 2020; Takeuchi, Koizumi, Ohishi, Harada, & Kashino, 2020; Xu, Dinkel, Wu, & Yu, 2020). In the DCASE AAC for 2021, the Clotho v2 dataset (Drossos, Lipping, & Virtanen, 2020) was used, which contains 6974 audio clips ranging from 15–30 s in length, with 5 captions each consisting of 8–20 English words.

In the AAC problem for the DCASE 2020 competition, Drossos, Adavanne, and Virtanen (2017) presented a baseline model using an encoder–decoder structure. This architecture was widely used in AAC design and involved a multi-layered, bi-directional Gated Recurrent Unit (GRU) encoder and a multi-layered decoder. Takeuchi et al. (2020) employed data augmentation, multi-task learning, and post-processing using a Long Short-Term Memory (LSTM) decoder to achieve the best results. Chen et al. (2020) proposed a pre-training stage for the encoder before combining the Transformer decoder to create the second-best model. Perez-Castanos et al. (2020) experimented with a Residual Network (ResNet) encoder and an LSTM decoder with the gammatone feature as an input. Pellegrini (2020) introduced the Listen–Attend

* Corresponding author at: Department of Applied Statistics, Chung-Ang University, Seoul, 06974, Republic of Korea.
E-mail addresses: whj9492@cau.ac.kr (H. Won), kbs778@cau.ac.kr (B. Kim), ikwak2@cau.ac.kr (I.-Y. Kwak), clim@cau.ac.kr (C. Lim).
[1] Authors contributed equally.

https://doi.org/10.1016/j.eswa.2023.120664
Received 7 May 2022; Received in revised form 18 March 2023; Accepted 30 May 2023
Available online 5 June 2023

Table 1
Top ten teams and their models for DCASE 2021 AAC Competition.

Authors	Rank	Encoder	Decoder	SPIDEr (Test)
Yuan et al.	1	CNN14, ResNet38, Wavegram-Logmel-CNN14	Transformer	0.310
Xu et al.	2	CNN10	LSTM	0.296
Xinhao et al.	3	CNN10	Transformer	0.294
Ye et al.	4	ResNet38	LSTM	0.280
Chen et al.	5	ResNet38	Meshed-memory Transformer	0.262
Won et al.	**6**	**CNN14**	**Transformer**	**0.249**
Narisetty et al.	7	Conformer, Wavegram-Logmel-CNN14	Transformer	0.236
Labbe et al.	8	CNN14	LSTM	0.221
Liu et al.	9	CNN10(from scratch)	Transformer	0.184
Eren et al.	10	Wavegram-Logmel-CNN14	GRU	0.182

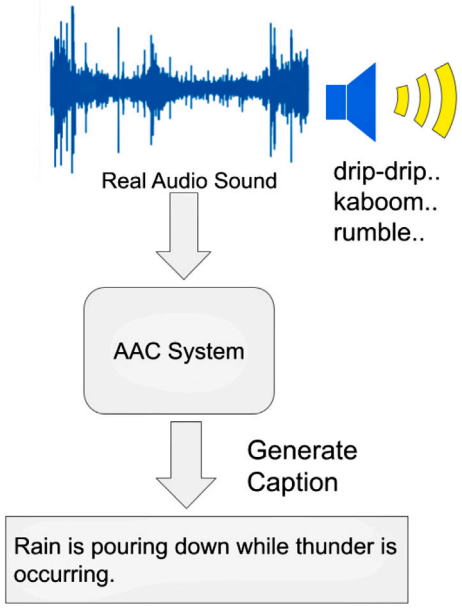

Fig. 1. Illustration of the Automated Audio Captioning system.

Spell (LAS) (Chan, Jaitly, Le, & Vinyals, 2015) architecture with a listener encoder and speller decoder. Xu et al. (2020) utilized a Convolutional Recurrent Neural Network (CRNN) encoder and a GRU decoder with fine-tuning via reinforcement learning.

Researchers could utilize a large amount of audio data, such as AudioSet (Gemmeke et al., 2017; Kong et al., 2019), to learn crucial sound-related feature representations. Koizumi, Masumura, Nishida, Yasuda, and Saito (2020) used a Transformer decoder and a pre-trained VGGish model (Hershey et al., 2017), while Xu, Dinkel, Wu, Xie, and Yu (2021) proposed pre-trained 10-layer Convolutional Neural Network (CNN) and CRNN5 encoder networks with GRU decoder.

The use of external data was permitted in the DCASE 2021 AAC competition. All of the top ten teams in the challenge employed models combined CNN-based encoders with LSTM, GRU or Transformer decoders as shown in Table 1. Nine among them employed pre-trained models from extensive audio data as their CNN-based encoders (Chen, Zhang, Wang, & Deng, 2021; Han et al., 2021; Labbé & Pellegrini, 2021; Mei et al., 2021; Narisetty, Hayashi, Ishizaki, Watanabe, & Takeda, 2021; Özkaya Eren & Sert, 2021; Won, Kim, Kwak, & Lim, 2021; Xu, Xie, Wu, & Yu, 2021; Yang & Sijun, 2021; Ye, Wang, Yang, & Zou, 2021). Interestingly, all the nine teams used models in the Pretrained Audio Neural Networks (PANNs) (Kong et al., 2020) as their pretrained models. PANNs were 15 pre-trained audio classification models using large-scale AudioSet data, including CNN10, CNN14, and ResNet38.

Han et al. (2021) employed 14-layer CNN, 38-layer ResNet, and Wavegram-Logmel-CNN14 as encoders, along with a Transformer as a decoder. Xu, Xie, et al. (2021) reported successful results through the utilization of reinforcement learning with a CNN10 encoder and GRU decoder. Chen et al. (2020) applied a ResNet38 encoder and a meshed-memory Transformer as the decoder. This modification of the Transformer was initially used for image captioning (Cornia, Stefanini, Baraldi, & Cucchiara, 2020). The majority of the competing teams employed PANNs as encoders, and the results were largely influenced by the effectiveness of their usage. The question is whether there exist better pre-trained networks than PANNs. Spectrogram, one of the audio preprocessing techniques, can be viewed as a two-dimensional image providing frequency and time-axis information. Consequently, even though pre-trained weights ought to be applied to a domain similar to the original data, pre-trained models from ImageNet can still be expected to yield a good performance on audio domain (Palanisamy, Singhania, & Yao, 2020). There are numerous pre-trained networks available from the image domain, such as EfficientNet (Tan & Le, 2019) and VGGNet (Simonyan & Zisserman, 2014). The recently proposed Audio Spectrogram Transformer (AST) (Gong, Chung, & Glass, 2021a) trained on AudioSet is also available. Therefore, in this paper, we explore the application of a variety of pre-trained networks with transfer learning to the AAC task, such as CNN14, ResNet54, CNN10, EfficientNet, VGGNet, and AST. The key contributions of this paper are summarized as follows:

- We present a novel system composed of a pre-trained encoder network and a Transformer decoder that achieves high performance on the Automated Audio Captioning (AAC) task 6.
- We conduct extensive experiments to compare the performance of several pre-trained models of PANNs, including CNN14, ResNet54, and CNN10. These models demonstrated excellent performance in the AAC competition and also showed good performance in the image domain.
- We show that the best performance is achieved in the scenario where the pre-trained CNN10, trained on AudioSet, is utilized as an encoder with the Transformer serving as a decoder, and fine-tuning applied.
- We consider how the caption is generated from AudioSet consisting of 527 labels. Our best model creates captions well to explain the corresponding audio even if the word of the AudioSet label is not in the caption.

This paper is organized as follows. Section 2 reviews relevant prior work. Section 3 presents the proposed system for the AAC. Section 4 describes the dataset used and discusses the experimental results. In Section 5, we draw conclusions and discuss potential future research directions.

2. Related works

2.1. Image captioning and audio captioning

Previous research regarding captioning tasks was actively conducted in the field of image captioning which generates sentences associated with the content of images (Vinyals, Toshev, Bengio, & Erhan,

2015). Image captioning models are capable of automatically generating captions for an image. This task requires algorithms to not only recognize the items in an image but also to capture and describe their relationships in natural language. As a result, sequence-to-sequence models are typically used, which employ an encoder to extract features from the image and a decoder to generate a caption. In AAC, many studies convert raw audio data into mel spectrogram and then extract features from the mel spectrogram using an encoder. Image and mel spectrogram feature extraction share many similarities, in that an encoder extracts features from three-dimensional data consisting of horizontal, vertical, and channels, which are then passed to a decoder that acts as a text generator. The state-of-the-art models for image captioning are as follows: one that employs a Faster region-based convolutional neural networks (Faster R-CNN) model with top-down attention as an encoder and an LSTM structure as a decoder to create sentences (Anderson et al., 2018), and another that uses a Meshed-Memory Transformer model, which has a memory-augmented encoder and a meshed decoder (Cornia et al., 2020). For audio captioning, we also utilize the encoder–decoder structure, which involves an encoder that converts audio into a latent embedding and a decoder that acts as a text generator.

2.2. Transfer learning

Transfer learning is a process of utilizing the features learned from one domain of data to another related domain (Chollet, 2017). For instance, a pre-trained network from vast ImageNet data can be used to create a classification model for classifying monkeys, dogs, cats and birds.

The steps for transfer learning can be outlined as follows: (1) Take layers from a pre-trained model. (2) Make the weights of layers non-trainable. (3) Add some trainable layers on top of the non-trainable layers to connect the pre-trained network and the output layer for solving the problem. (4) Train the weights of newly added trainable layers. (5) Make some last non-trainable layers trainable and retrain the trainable weights with a small learning rate (Chollet, 2017). It is observed that transfer learning is especially useful when the size of training data is small. Deep learning models are susceptible to overfitting when the size of training data is small. However, pre-trained networks which have been trained on a vast amount of data are capable of avoiding overfitting even with limited training data.

Neyshabur, Sedghi, and Zhang (2020) studied how transfer learning can lead to good performance and can perform well on any layer of the network, and found that the success of transfer learning depends on the latter layer. Li, Grandvalet, and Davoine (2020) observed that the use of L2 penalty in pre-trained models is key to achieving good performance in transfer learning. Guo et al. (2019) proposed TransTailor, a type of pruning technique, to reduce FLoating point Operations Per Second (FLOPS) and improve accuracy as a way to resolve structural discrepancies between a pre-trained model and a model used in a target task during transfer learning.

Recently, many studies implemented transfer learning in the audio domain. Similar to the utilization of pre-trained models trained on ImageNet data in the image domain, pre-trained models trained on AudioSet data were often utilized in the audio domain. PANNs and VGGish, both trained on AudioSet data, are two commonly employed pre-trained networks. Therefore, we used a pre-trained network through transfer learning in the AAC task.

2.3. Transformer

Regarding sequence modeling such as language modeling and machine translation, recurrent neural networks, particularly LSTM and GRU, were renowned as state-of-the-art methodologies (Chung, Gülçehre, Cho, & Bengio, 2014; Hochreiter & Schmidhuber, 1997). However, it is known that RNN has a problem with long-term dependency. Even though the long-term dependencies are memorized using a state vector in LSTM and GRU, sequential computing still has inherent constraints in terms of speed (Vaswani et al., 2017). To address the problem of sequential calculation, attention methods were studied. As the title of the Transformer paper, 'attention is all you need', implies, sequential computing was eliminated in the Transformer (Vaswani et al., 2017). This model was initially introduced in 2017 at the thirty-first conference on Neural Information Processing Systems and achieved state-of-the-art in machine translation. As a result of the parallel system, learning of Transformer is faster than networks based on recurrent or convolutional layers. The attention mechanism in the Transformer is used so that the model can learn global dependencies between input and output.

Currently, models utilizing the Transformer are employed not only in machine translation but also in other tasks. Additionally, new models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018), Detection with Transformer (DETR) (Carion et al., 2020), and Generative Pre-Training (GPT)-3 (Brown et al., 2020) using the Transformer as the base have been actively developed. The Transformer structure has been applied not only in natural language processing but also in various fields. For instance, Akbari et al. (2021) utilized only the encoder portion of the Transformer for multimodal tasks that involved audio, video and text, achieving state-of-the-art performance on multiple downstream tasks such as video action recognition, audio event classification, and image classification. In the field of self-supervised learning, wav2vec 2.0 has emerged, which improves the performance of existing wav2vec models by applying Transformer structures that can mask current locations and infer masked locations from surrounding data (Baevski, Zhou, Mohamed, & Auli, 2020).

3. Proposed methods

The architecture of our proposed system is illustrated in Fig. 2. We use a log-mel spectrogram to extract input audio features and a pre-trained encoder network to generate context vectors. The Transformer decoder then takes the context vector as input and produces the probability of the caption words as its output.

3.1. Encoder network

The CNN and ResNet models were pre-trained on AudioSet, while the VGGNet and EfficientNet models were pre-trained on ImageNet. For AST, two datasets were utilized for pre-training, leading to two versions of pre-trained AST models. We employ a log-mel spectrogram to extract the audio features of the input signal and a pre-trained encoder network to generate the corresponding context vectors.

3.1.1. Pre-trained audio neural networks

Kong et al. (2020) proposed Pre-trained Audio Neural Networks (PANNs) based on the AudioSet dataset, which contains 5000 h of audio and 527 sound labels. The 15 pre-trained models, such as CNN10, CNN14, ResNet38, and ResNet54, have been made accessible to the public. Audio snippets in the dataset were obtained from YouTube videos (Gemmeke et al., 2017). AudioSet is a repository of over 2 million audio recordings, which includes a balanced train dataset of 22,160 audio files, with at least 50 files for every sound class. The evaluation dataset consisted of 20,371 audio files, each with a length of 10 s, or padded with zeros if shorter. For audio-related applications, the pre-trained models (CNN10, CNN14 and ResNet54) of multi-label classification could be utilized, achieving state-of-the-art performance for the 527 sound classes. The encoders of these models, CNN10, CNN14 and ResNet54, are summarized in Tables 2–4, respectively, for the AAC model, excluding the fully connected layers.

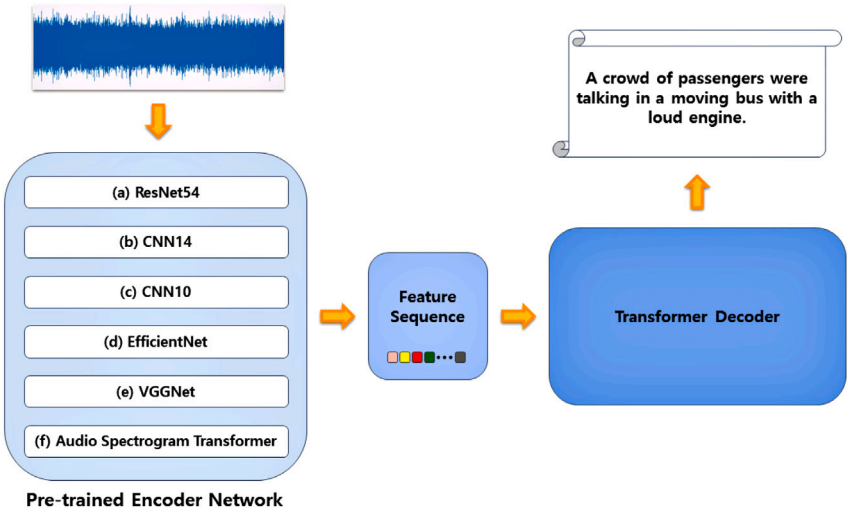

Fig. 2. Overview of the proposed system for the AAC.

Table 2
CNN10 architecture.

CNN10
Log-mel spectrogram 64 mel bins
(3 × 3 @64,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @128,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @256,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @512,BN,ReLU)×2

BN: Batch Normalization.

Table 3
CNN14 architecture.

CNN14
Log-mel spectrogram 64 mel bins
(3 × 3 @64,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @128,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @256,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @512,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @1024,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @2048,BN,ReLU)×2

BN: Batch Normalization.

Table 4
ResNet54 architecture.

ResNet54
Log-mel spectrogram 64 mel bins
(3 × 3 @512,BN,ReLU)×2
Pooling 2 × 2
(bottleneckB@64)×3
Pooling 2 × 2
(bottleneckB@128)×4
Pooling 2 × 2
(bottleneckB@256)×6
Pooling 2 × 2
(bottleneckB@512)×3
Pooling 2 × 2
(3 × 3 @512,BN,ReLU)×2

BN: Batch Normalization; bottleneckB: Bottleneck Block.

3.1.2. Audio spectrogram transformer

In recent years, CNNs have been widely used for both audio and visual tasks, due to their effectiveness in capturing features in a specific geographic area (Gulati et al., 2020). However, the Transformer approach is also capable of capturing global aspects through self-attention. Audio Spectrogram Transformer (AST) is an attention-mechanism based classification model, which consists of a Transformer encoder structure and does not rely on CNN. AST is similar in design to Vision Transformer (ViT) (Dosovitskiy et al., 2020), yet modified to cater to the audio classification task. This includes four modifications: (1) AST utilizes single-channel spectrograms as input, while ViT uses three-channel picture data; (2) ViT requires a fixed input size of either (224, 224) or (384, 384), whereas AST allows for a variable input size depending on the length of the audio spectrogram; (3) ViT does not consider overlapping a patch size of (16, 16), whereas AST does consider overlapping patch samples for positional embedding adaptation; and (4) the last classification layer of ViT was discarded, and a new one was initialized for AST (Gong et al., 2021a). We employ two versions of the AST model pre-trained on AudioSet and ImageNet, respectively, to extract features from audio signals, in our proposed AAC system.

AST has been pre-trained after extracting the feature to a size of (128,1024) with log-mel spectrogram from the AudioSet. However, when fine-tuning the pre-trained model with Clotho data, information is lost when the time dimension is set to 1024 due to the relatively longer playback time of Clotho data. To prevent such information loss, we have modified the log-mel spectrogram with the size of (128,1024). As shown in Fig. 3, this preprocessing involves dividing the mel spectrogram into odd-numbered and even-numbered sections in the time dimension and stacking them in the frequency dimension to create a log-mel spectrogram of (128,1024).

The AST model is employed as the encoder of the AAC model, as shown in Fig. 4. A 16 × 16 patch size is selected with an overlap of 6 in both time and frequency dimensions. For the AAC encoder, the special classification token [CLS], flatten layer, and Multi-Layer Perceptron (MLP) layer are removed. The 2D patch samples are flattened to serve as Transformer encoder input data. A linear projection layer is then utilized to map to a 768-dimensional context vector and add positional embedding to each patch embedding, to comprehend the spatial structure. Finally, the output from the Transformer encoder is sent to the Transformer decoder.

3.2. ImageNet pre-trained models

Models pre-trained on ImageNet data are frequently used in computer vision applications (Hussain, Bird, & Faria, 2018). Audio spectrograms could be considered as 2D images, making them suitable for

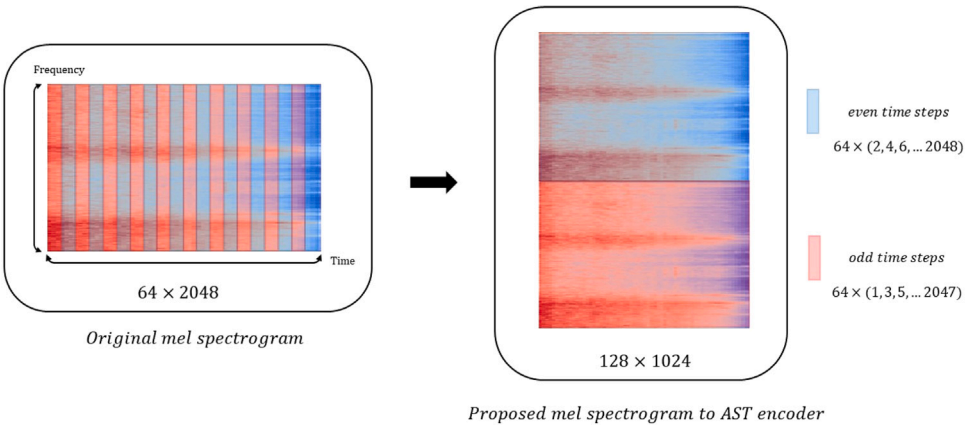

Fig. 3. Pre-processing of the mel spectrogram to match the input size for the pre-trained AST.

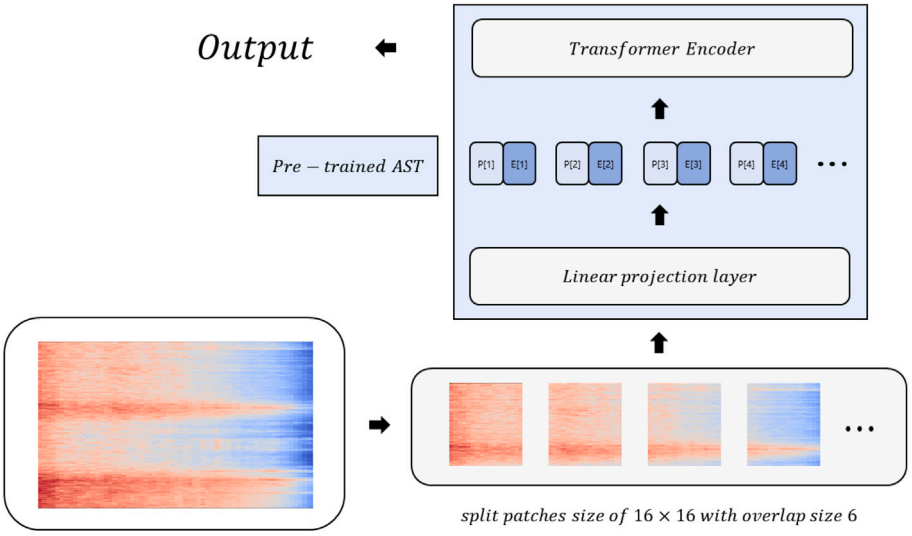

Fig. 4. Proposed AST encoder architecture.

use with these models. In our experiments, we employ VGGNet and EfficientNet architectures, which are widely used for sound classification (Gong, Chung, & Glass, 2021b; Qian, Bi, Tan, & Yu, 2016).

VGGNet is popular for its simple structure, which is composed of repeating identical CNN blocks (Simonyan & Zisserman, 2014). An important feature of VGGNet is that the filter size of all convolution layers is 3. While using a small filter size, the depth of the model is increased. VGG16 and VGG19 are the two most popular VGGNet models, with 16 and 19 weight layers, respectively; we used the VGG16 pre-trained network.

Simply stacking layers deeply, as in VGGNet, is one type of model scaling strategy. However, since the number of layers is manually adjusted, this method is inefficient. On the other hand, EfficientNet, based on MnasNet (Tan et al., 2019), is applied compound scaling. This compound scaling approach updates the parameters based on the relationship among model depth, width, and the resolution of the input image.

For pre-trained weights of VGGNet and EfficientNet, the input dimension for channels is 3 due to the structure of the ImageNet dataset. To match our input dimension, the channel is adjusted to 1.

3.3. Decoder network

Our Transformer decoder architecture is depicted in Fig. 5. A conventional Transformer decoder with multi-head self-attention is used. It is a two-layers Transformer with a hidden dimension of 192 with four heads. Positional encoding is added to input embedding to give positional information. Input to the decoder is a word embedding feature obtained using the Word2Vec model. It then goes to the masked multi-head attention module, which returns a query vector for the following multi-head attention module. The encoder network output is used to generate the key and value vectors for the attention module. The Transformer block is iterated twice before feeding the output tensor into a dense layer. Finally, a dense layer with a softmax activation function generates probabilities of the caption words.

4. Experiments

4.1. Dataset and data pre-processing

Clotho v2 is an audio captioning dataset that includes five captions for each CD-quality audio clip (44.1 kHz sampling rate, 16-bit sample width) (Drossos et al., 2020). The audio snippets last between 15 and 30 s, and each caption has ranges from eight to twenty words. We use this dataset in the experiment. It comprises 6974 audio samples, 34,870 captions, and approximately 4500 words. Clotho v2 is divided into four parts: development, validation, evaluation, and testing. The captions are accessible to the public solely for the development, validation, and evaluation parts. Clotho v2 was selected for the DCASE 2021 AAC competition due to its ability to handle various types of audio content.

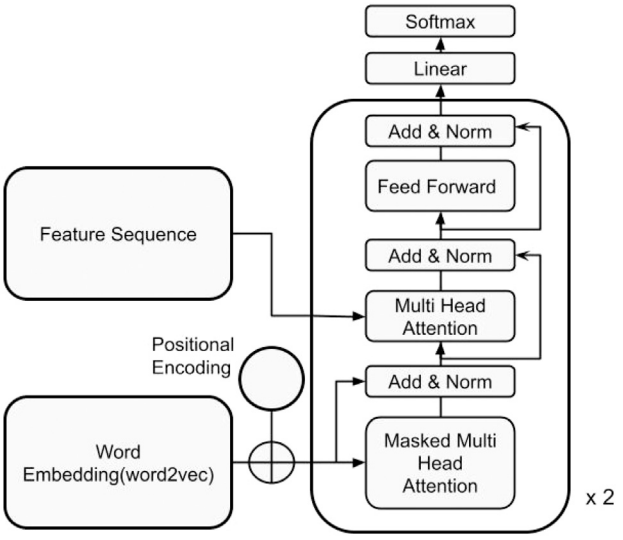

Fig. 5. Transformer decoder.

In this study, as in the DCASE 2021 competition, the Clotho v2 dataset is utilized for model training and evaluation.

Three different types of input features are used for the study. The first type is designed for PANNs pre-trained networks (specifically, CNN10, CNN14, and ResNet54). The second and the third are for AST models and ImageNet pre-trained models, respectively. In the case of PANNs models, the 44.1 kHz recording is downsampled to 32 kHz, and a log-mel spectrogram is extracted using the preprocessing codes provided by PANNs.[2] For the AST model, the 44.1 kHz recording is downsampled to 16 kHz, and the audio waveform is transformed into a log-mel spectrogram feature according to the procedure described in Gong et al. (2021a). Finally, for the ImageNet pre-trained models, a Hann window of size 1024 with 50% overlap is applied to the 44.1 kHz audio data, and 64 log mel-band energies are extracted from each window frame. For the number of time windows, we compute the maximum time window number, T, across sample datasets. To have a fixed size T for the input feature on our model, we pad zero to the time dimension. Additionally, we use the pre-trained Word2Vec model (Mikolov, Chen, Corrado, & Dean, 2013) from the Gensim python package (Rehurek & Sojka, 2010) for the word embedding. The pre-trained Word2Vec embedding is fine-tuned using caption sentences of the training set. We employ Spec Augment (Park et al., 2019) as a data augmentation technique, where frequency and time masks are randomly applied to the log-mel spectrogram to increase the robustness of the training.

4.2. Experimental setup

4.2.1. Hyperparameters

In training, a batch size of 8 is utilized with a learning rate of 10^{-4}. L2 regularization is applied to all trainable parameters with factor $\lambda = 10^{-6}$. The Adam optimizer (Kingma & Ba, 2015) is employed and the Stochastic Weight Averaging (SWA) method (Izmailov, Podoprikhin, Garipov, Vetrov, & Wilson, 2018) is applied to boost performance. Dropout with a probability of $P = 0.2$ is applied to the encoder and Transformer decoder.

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

4.2.2. Training procedure

There are three steps to the training procedure: (1) transfer learning for the encoder network, (2) training the Transformer decoder network while setting the weights of the pre-trained encoder network non-trainable and using A Clotho data caption embedding through Word2Vec, and (3) unfreezing the weights of the last convolution block from the encoder network and fine-tuning the trainable weights using a low learning rate. Six pre-trained networks are employed as the encoder network in the transfer learning stage: CNN10, CNN14, ResNet54, AST, VGG16, and EfficientNet. All five captions per audio are used as a reference for metric and loss calculation. Beam search is utilized to improve decoding performance in the inference stage, with a beam size of 3. Word2vec embeddings are pre-calculated using 1000 epochs of training. There are 30 epochs of training with a learning rate 0.0001 in step (2) the initial round of training, freezing encoder network parameters. Finally, the last convolution blocks of pre-trained encoder networks are unfrozen during the fine-tuning stage, which continues for another 30 epochs with a learning rate 0.00001.

4.3. Evaluation metrics

We present our evaluation of the proposed system using the metrics from the AAC task at the DCASE 2021 challenge. These metrics can be divided into two categories: machine translation metrics and captioning metrics. The machine translation metrics include Bilingual Evaluation Understudy (BLEU)$_n$ (Papineni, Roukos, Ward, & Zhu, 2002), Recall-Oriented Understudy for Gisting Evaluation (ROUGE)$_L$ (Lin, 2004), and Metric for Evaluation of Translation with Explicit Ordering (METEOR) (Lavie & Agarwal, 2007). BLEU is a precision-based metric that calculates a weighted geometric mean of an adjusted precision of n-grams between anticipated and ground truth captions. The adjusted precision calculation penalizes the geometric mean computation, favoring short predicted captions and thus penalizing forecasted captions that are shorter than the truth. BLEU$_n$ ($n \in \{1, 2, 3, 4\}$) refers to n-grams with typical lengths of one to four (Chen et al., 2015; Papineni et al., 2002). ROUGE$_L$, a statistic based on the Longest Common Subsequence (Lin, 2004), is used to compute an F-measure between the predicted and ground truth captions. The calculation of this F-measure is oriented towards recall, using a value for the $\beta = 1.2$ (Chen et al., 2015; Lin, 2004). METEOR (Lavie & Agarwal, 2007) is a recall-based statistic, where recall is weighted higher than precision (Chen et al., 2015). It produces a harmonic mean of precision and recall of caption segments between the expected and ground truth captions, and uses word alignment to match exact words, stems of words, synonyms, and paraphrases in the anticipated and ground truth captions. This alignment is computed over segments of the captions while limiting the number of chunks required.

The captioning metrics are Consensus-based Image Description Evaluation (CIDEr) (Vedantam, Lawrence Zitnick, & Parikh, 2015), Semantic Propositional Image Caption Evaluation (SPICE) (Anderson, Fernando, Johnson, & Gould, 2016), and a linear combination of these two metrics called SPIDEr (Liu, Zhu, Ye, Guadarrama, & Murphy, 2017). CIDEr calculates a weighted sum of the cosine similarity between the predicted and ground truth captions for n-grams of length $n \in [1, 4]$. Term Frequency Inverse Document Frequency (TF-IDF) weighting is used to calculate the cosine similarity for each n-gram (Chen et al., 2015; Vedantam et al., 2015). SPIDEr is a combination of CIDEr and SPICE that examines the anticipated captions' fluency and semantic qualities.

4.4. Experimental results

Table 5 summarizes the experimental results of the proposed models and the baseline system provided by the DCASE 2021 challenge. The baseline model is a seq2seq model with three bi-directional GRU layers as an encoder and two GRU layers as a decoder. Among the models

Table 5
Score for model performance on evaluation data. The values with the best performance are shown in bold.

Model	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	METEOR	CIDEr	SPICE	SPIDEr
Baseline model									
BGRU + GRU	0.378	0.119	0.050	0.017	0.263	0.078	0.075	0.028	0.051
From scratch									
CNN14 + Transformer	0.466	0.262	0.156	0.092	0.309	0.137	0.208	0.087	0.148
ResNet54 + Transformer	0.459	0.253	0.152	0.084	0.312	0.131	0.182	0.085	0.133
AST + Transformer	0.209	0.104	0.058	0.031	0.084	0.209	0.202	0.090	0.146
Pre-trained with AudioSet									
CNN14 + Transformer	0.550	0.361	0.244	0.160	0.375	**0.172**	0.401	**0.121**	0.261
CNN10 + Transformer	**0.569**	**0.378**	**0.257**	**0.172**	**0.380**	0.170	**0.420**	0.119	**0.269**
ResNet54 + Transformer	0.540	0.345	0.230	0.152	0.361	0.161	0.383	0.109	0.246
AST + Transformer	0.488	0.322	0.212	0.157	0.350	0.159	0.337	0.115	0.226
Pre-trained with ImageNet									
VGGNet + Transformer	0.522	0.342	0.232	0.154	0.368	0.158	0.352	0.104	0.228
Efficientnet + Transformer	0.513	0.336	0.230	0.156	0.360	0.157	0.343	0.107	0.225
AST + Transformer	0.210	0.109	0.064	0.035	0.080	0.205	0.254	0.094	0.174

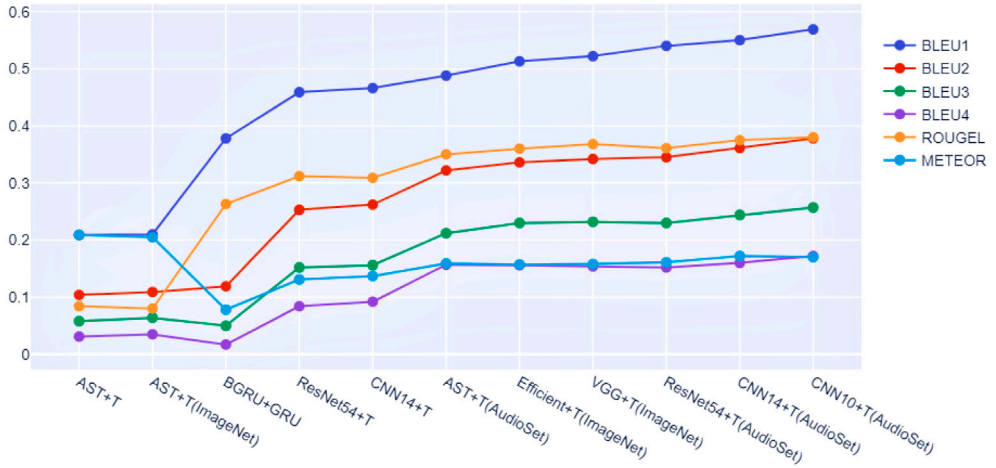

Fig. 6. Graphical presentation of model performance on evaluation data (machine translation metrics).

with a pre-trained encoder, the CNN10 and ResNet54 models trained on AudioSet data outperform the VGGNet and EfficientNet models trained on ImageNet data, with SPIDEr scores of 0.269 and 0.246, respectively. The best performance across all models is achieved by the CNN10 pre-trained encoder with a Transformer decoder, with a SPIDEr score of 0.269.

Models with pre-trained encoders (using AudioSet or ImageNet) generally show superior performance with SPIDEr scores ranging from 0.174 to 0.269, compared to models trained with Clotho data (from scratch), whose scores range from 0.133 to 0.148.

Figs. 6 and 7 demonstrate that models with good performance in both machine translation metric scores and captioning metric scores are arranged to the right. The models that use CNN10 and CNN14 pre-trained networks on AudioSet data show the best performance in both machine translation metric scores and captioning metric scores. Pre-trained networks based on ImageNet data have slightly lower performance, and models trained from scratch without pre-trained networks perform the worst in terms of both machine translation and captioning metrics.

4.5. Ablation studies

4.5.1. Fine-tuning

In the transfer learning process, it is known that the fine-tuning step can improve performance by making some blocks of the encoder network trainable and then re-training them with a small learning rate. Thus, we experiment with fine-tuning the last one or two convolution blocks of the CNN10 and ResNet54 models to check the effects of fine-tuning. Table 2 shows the structure of the CNN10 embedding layers, which consist of four convolution blocks. Each convolution block in the CNN10 model consists of two 3 by 3 convolution layers. The ResNet54 embedding consists of six convolution blocks as shown in Table 4. We unfreeze the last one or two convolution blocks of CNN10 and ResNet54 in the experiment. Table 6 shows the results of comparing a default from-scratch model (freezing all encoder networks) and two fine-tuning models (unfreezing last one or two convolution layers) using models with CNN10 and ResNet54 encoders. Both ResNet54 and CNN10 encoder models achieve better overall performance for all scoring metrics by fine-tuning. Particularly, the CNN10 encoder (unfreezing last two convolution layers) with a Transformer decoder model achieved the best performance across all scoring metrics, including BLEU, ROUGE, METEOR, CIDEr, SPICE and SPIDEr, with scores of 0.569, 0.378, 0.257, 0.172, 0.380, 0.170, 0.420, 0.119 and 0.269 respectively. This suggests that fine-tuning through Clotho data is more effective than simply fixing the pre-trained weights without fine-tuning, as it was well-known.

4.5.2. Study on feature extraction methods

We explore three feature extraction methods: (1) the log-mel spectrogram, (2) the constant Q transform (CQT) spectrogram (Lidy & Schindler, 2016), and (3) the gammatone filter spectrogram. The CQT is a time–frequency representation with geometrically separated frequency bins and equivalent Q-factors (ratios of center frequencies to bandwidths) across all bins. The spectrogram of a gammatone filter is

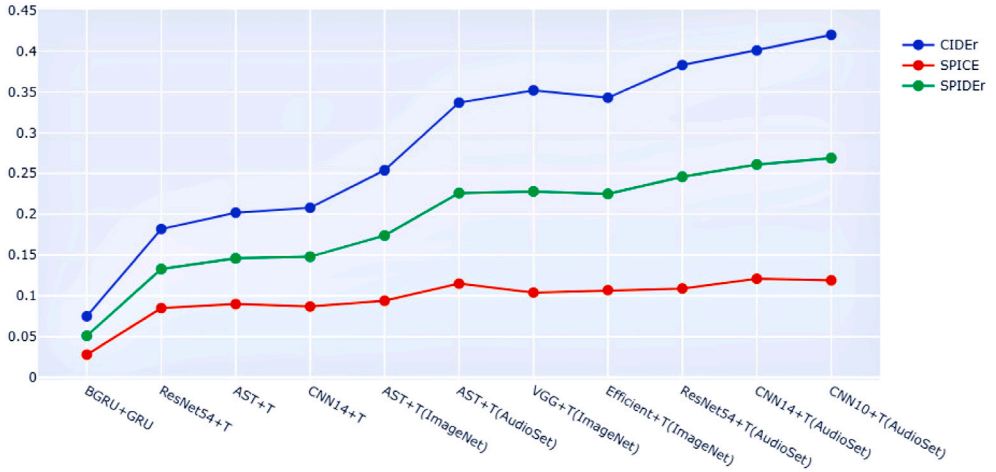

Fig. 7. Graphical presentation of model performance on evaluation data (captioning metrics).

Table 6
Evaluation scores for pre-trained encoder models on evaluation data with respect to the level of fine-tuning. The values with the best performance are shown in bold. Transformer is used as the decoder for all the models.

Level of fine-tuning	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	METEOR	CIDEr	SPICE	SPIDEr
ResNet54									
Without fine-tuning	0.540	0.345	0.230	0.152	0.361	0.161	0.383	0.109	0.246
Unfreezing last 1 block	0.545	0.360	0.244	0.164	0.366	0.166	0.393	0.111	0.252
Unfreezing last 2 blocks	0.549	0.358	0.239	0.157	0.366	0.168	0.397	0.117	0.257
CNN10									
Without fine-tuning	0.511	0.332	0.225	0.149	0.360	0.155	0.356	0.108	0.232
Unfreezing last 1 block	0.549	0.348	0.227	0.142	0.370	0.169	0.353	0.113	0.233
Unfreezing last 2 blocks	**0.569**	**0.378**	**0.257**	**0.172**	**0.380**	**0.170**	**0.420**	**0.119**	**0.269**

Table 7
Evaluation scores of CNN14 + Transformer model using three different types of spectrogram on evaluation data. The values with the best performance are shown in bold.

Type of spectrogram	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	METEOR	CIDEr	SPICE	SPIDEr
Log-mel	0.451	0.275	0.172	0.100	0.318	0.133	0.197	0.087	0.142
CQT	0.436	0.288	0.191	0.115	0.339	0.132	0.222	0.090	0.156
Gammatone	**0.467**	**0.312**	**0.215**	**0.142**	**0.358**	**0.148**	**0.306**	**0.096**	**0.201**

created by decomposing the input audio signal into the time–frequency domain to use a bank of gammatone filters, then downsampling the filter-bank replies along the time dimension (Ayoub, Jamal, & Arsalane, 2016).

We compare the performance of the CNN14 + Transformer model using these three features without using transfer learning, as there are no pre-trained models with CQT and gammatone features available. Table 7 shows the performance results. In most metrics, the model employing the CQT spectrogram outperforms the model utilizing the log-mel spectrogram, except for $BLEU_1$ and METEOR. The model using the gammatone spectrogram, however, outperforms the other two models in all evaluation metrics. If a pre-trained model with the gammatone spectrogram is available, we may be able to get even better results.

4.5.3. Processing embedding matrix in AST

The pre-trained weights provided by AST are learned through a log-mel spectrogram of (128,1024) as explained in Section 3.1.2. We consider the following methods to change the shape of the log-mel spectrogram and conduct an experiment to compare the effect on the performance of the model:

- Simple cut and paste: We attempt to resize the existing log-mel spectrogram from (64, 2048) to (128, 1024) by performing the following steps. Initially, the log-mel spectrogram has a size of (64, 2048). We divide the time dimension in half, resulting in two (64, 1024) sized spectrograms. Then, we concatenate them in the frequency dimension, resulting in a (128, 1024) dimensional feature map.
- Time preserving cut and paste: There is a risk of losing temporal information when using the above resizing method. As illustrated in Fig. 3, the log-mel spectrogram is divided into odd-numbered sections and even-numbered sections in the time dimension and stacked in the frequency dimension to make a log-mel spectrogram size of (128,1024)

Table 8 shows the results of both processing methods for AST embedding. In all evaluation metrics, the time-preserving cut and paste method works better than the simple cut and paste method in terms of the model performance.

4.6. Qualitative results

4.6.1. Analyzing predicted captions

Fig. 8 displays the results of the proposed CNN10 (pre-trained using AudioSet) + Transformer model on the Clotho dataset. The generated captions were usually semantically similar to the answer captions, indicating satisfactory performance. However, there were occasions where the model mis-predicted captions. For example, for the fourth caption, 'Water spraying against a surface and dripping in the background,' the model predicted 'A shower is running with water in the

Table 8
Evaluation scores of AST + Transformer model using two different feature processing methods on evaluation data. The values with the best performance are shown in bold.

Feature processing method	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	METEOR	CIDEr	SPICE	SPIDEr
Simple cut and pasting	0.377	0.251	0.194	0.145	0.301	0.131	0.309	0.101	0.205
Time preserve cut and pasting	**0.488**	**0.322**	**0.212**	**0.157**	**0.350**	**0.159**	**0.337**	**0.115**	**0.226**

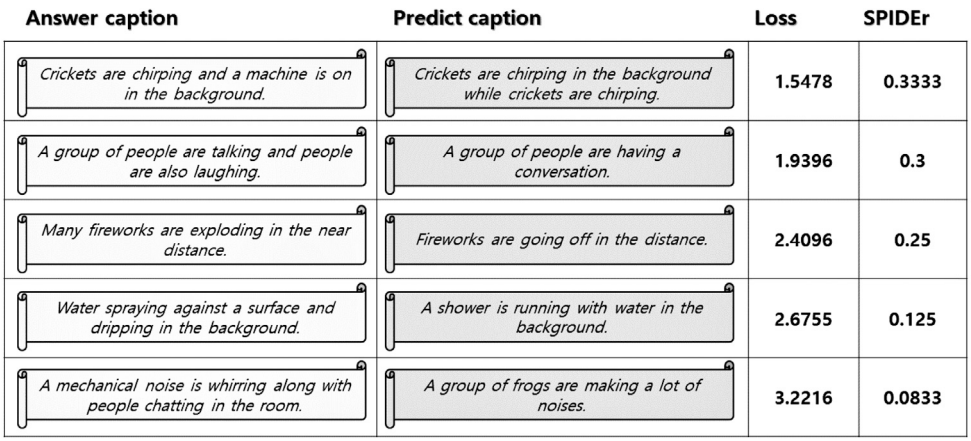

Fig. 8. Examples of answer and predicted caption.

background,' which might not be the exact answer caption. However, the sound of water being sprayed through the shower could be heard in the audio corresponding to the caption, indicating that the predicted caption is also legitimate. For the fifth answer caption, 'A mechanical noise is whirring along with people chatting in the room,' our model completely mis-predicted the caption as 'A group of frogs are making a lot of noises.' Although the actual audio corresponding to the caption sounded similar to the frog's noisemaking, the predicted caption was not accurate. The loss and SPIDER scores for these results are also displayed in Fig. 8.

Overall, our CNN10 + Transformer model outputted captions that were semantically similar to the answer captions, although there were cases where the predicted captions were different from the actual answer captions. In such cases, however, the predicted captions were still often meaningful.

4.6.2. Evaluation on a real world sound data

We evaluated the performance of our AAC model on the AudioSet dataset. The AudioSet dataset consists of 527 classes (Gemmeke et al., 2017) and the captions were generated using our proposed CNN10 + Transformer model for five sample audio data from five different classes. The following is the audio categories, along with their generated captions:

- Rain — A heavy rainstorm is falling heavily and hitting the ground
- Bird — Birds are chirping and singing in the background
- Water — A person is washing dishes in a bucket of water
- Train, Rail transport — A train is passing by a train station
- Car, Vehicle — A motorcycle is revving its engine and then speeding up

The five classes selected above overlap the top 30 most frequently used words in captions of the Clotho data. As such, the presence of several similar words indicates that our model is able to create appropriate captions by listening to the audio corresponding to the labels.

To evaluate the performance of our model on audio data that was not included in the Clotho dataset, we generated captions for five samples that contain words that were not present in the Clotho dataset. The results were as follows:

- Plucked string instrument — A person is playing a guitar
- Female speech — A woman is talking
- Bass drum — A person is playing a drum while music is playing
- Cock-a-doodle-doo — A variety of birds are chirping and whistling loudly

There are labels such as bass drum and plucked string instrument in the AudioSet, and they are classified by those labels. However, the captions of the Clotho data did not provide this level of detail. Nevertheless, the captions generated by our AAC model still contain words such as drum and guitar, indicating that the model is able to generate meaningful captions even with a limited vocabulary. To further improve the accuracy of the model, more diverse audio and caption data could be used for training.

We were able to gain some insight into AAC tasks with these qualitative results. The challenge of being able to accurately differentiate between similar sounds, such as mechanical noise and frog's noise, was still an issue to be addressed. If an audio captioning system was implemented for the deaf community, such a problem could lead to disastrous outcomes. Therefore, our goal is to develop a system that can clearly distinguish environmental sounds, despite the existence of many similar sounds in the real world.

The above qualitative results with the corresponding audio are publicly available via our YouTube channel.[3]

5. Discussion

5.1. Real world use case of AAC

Recent research in the audio domain has highlighted the importance of not only translating non-voice sounds into other languages, but also of identifying and explaining them. This is analogous to closed captions in audiovisual resources from streaming platforms such as Netflix (Xu et al., 2022). AAC technologies can be used to facilitate deaf individuals by providing a text interpretation of audio signals in their environment. As AAC technologies become more advanced, they can provide more convenient services, such as assistive robots for deaf people and more dynamic closed captions from streaming platforms.

[3] https://youtu.be/6Qxa7iD1juw

5.2. Why CNN10 and CNN14 encoders worked better?

In our study, pre-trained models from AudioSet and ImageNet data were imported through transfer learning and demonstrated good performance. Of the pre-trained models, CNN10 and CNN14 models performed exceptionally well in comparison to ResNet54, AST, CNN10, and CNN14. This was because deep models that worked well in the image domain did not necessarily perform well in the audio domain, as they tended to lose frequency-wise information while stacking many layers (Kim, Chang, Lee, & Sung, 2021; Koutini, Eghbal-zadeh, & Widmer, 2021). Thus, a model design that considers the receptive field is necessary (Koutini et al., 2021). The superior performance of the CNN10 and CNN14 models can be attributed to their more appropriate receptive field.

5.3. Future work

For further research, there are a number of topics to be considered. For instance, creating a training dataset by adding captions to numerous and diverse voices, such as the Clotho dataset, requires a lot of time and effort. To address this, studies involving self-supervised learning or weakly supervised learning, which require less data labeling, could be conducted. Additionally, while this paper focused on log-mel spectrograms as a feature, research into more diverse features may also be beneficial.

6. Conclusion

This paper presented the results of our team's participation in the AAC Task of the DCASE 2021 competition. We explored the use of external data by proposing an encoder pre-trained on the AudioSet data, along with a transformer decoder. Additionally, we evaluated several pre-trained encoders (AST, ResNet, VggNet, CNN, and ResNet) on the task. Furthermore, we analyzed the real-world use case of AAC through qualitative analysis.

Our results showed that the models using pre-trained encoders outperformed those using Clotho data from scratch, with SPIDER scores ranging from 0.174 to 0.269. Furthermore, we found that pre-trained networks based on AudioSet outperformed models based on ImageNet, with SPIDER scores ranging from 0.226 to 0.269. The CNN10 + Transformer model, pre-trained with AudioSet, achieved the best results across 7 out of the 9 evaluation metrics, with a SPIDER score of 0.269. Additionally, the CNN14 encoder (pre-trained with AudioSet) and ResNet54 encoder (pre-trained with AudioSet) models with the Transformer decoder achieved the second and third highest scores, respectively. Upon adaptation of pre-trained encoders, our fine-tuning strategy yielded superior performance compared to the baseline, with a SPIDER score improvement from 0.232 to 0.269 for the CNN10 + Transformer model. Qualitatively, the AAC model generated satisfactory captions for general audios. In addition, one limitation was the lack of the use of unlearned words. To further enhance its predictive capability, It would be better if the model is trained on a larger amount of data that covers a wider range of situations.

CRediT authorship contribution statement

Hyejin Won: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft. **Baekseung Kim:** Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Il-Youp Kwak:** Conceptualization, Funding acquisition, Investigation, Methodology, Writing – review & editing, Supervision. **Changwon Lim:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is publicly available.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (RS-2023-00208284, 2021R1F1A1056516).

References

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., et al. (2021). Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, *34*, 24206–24221.

Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *European conference on computer vision* (pp. 382–398). Springer.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., et al. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6077–6086).

Ayoub, B., Jamal, K., & Arsalane, Z. (2016). Gammatone frequency cepstral coefficients for speaker identification over VoIP networks. In *2016 international conference on information technology for organizations development* (pp. 1–5). IEEE.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, *33*, 12449–12460.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer.

Chan, W., Jaitly, N., Le, Q. V., & Vinyals, O. (2015). Listen, attend and spell. CoRR abs/1508.01211..

Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollár, P., et al. (2015). Microsoft COCO captions: Data collection and evaluation server. CoRR.

Chen, K., Wu, Y., Wang, Z., Zhang, X., Nian, F., Li, S., et al. (2020). Audio captioning based on transformer and pre-trained CNN. In *Proceedings of the detection and classification of acoustic scenes and events 2020 workshop* (pp. 21–25).

Chen, Z., Zhang, D., Wang, J., & Deng, F. (2021). *Audio captioning with meshed-memory transformer*: *Technical report*, DCASE2021 Challenge.

Chollet, F. (2017). *Deep learning with python* (pp. 143–159). Chapter 5. Deep learning for computer vision.

Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR abs/1412.3555.

Cornia, M., Stefanini, M., Baraldi, L., & Cucchiara, R. (2020). Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10578–10587).

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. CoRR abs/2010.11929.

Drossos, K., Adavanne, S., & Virtanen, T. (2017). Automated audio captioning with recurrent neural networks. In *2017 IEEE workshop on applications of signal processing to audio and acoustics* (pp. 374–378). IEEE.

Drossos, K., Lipping, S., & Virtanen, T. (2020). Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 736–740). IEEE.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., et al. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing* (pp. 776–780). IEEE.

Gong, Y., Chung, Y., & Glass, J. R. (2021a). AST: Audio spectrogram transformer. CoRR.

Gong, Y., Chung, Y.-A., & Glass, J. (2021b). Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3292–3306.

Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., et al. (2020). Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020, 21st annual conference of the international speech communication association* (pp. 5036–5040). ISCA.

Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., & Feris, R. (2019). SpotTune: Transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4805–4814).

Han, Q., Yuan, W., Liu, D., Li, X., & Yang, Z. (2021). Automated audio captioning with weakly supervised pre-training and word selection methods. In *Proceedings of the 6th detection and classification of acoustic scenes and events 2021 workshop* (pp. 6–10). Barcelona, Spain.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., et al. (2017). CNN architectures for large-scale audio classification. In *2017 Ieee international conference on acoustics, speech and signal processing* (pp. 131–135). IEEE.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hussain, M., Bird, J. J., & Faria, D. R. (2018). A study on cnn transfer learning for image classification. In *UK workshop on computational intelligence* (pp. 191–202). Springer.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. CoRR abs/1803.05407.

Kim, B., Chang, S., Lee, J., & Sung, D. (2021). Broadcasted residual learning for efficient keyword spotting. In *Proc. Interspeech 2021* (pp. 4538–4542). Brno: ISCA.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)* (pp. 1–15).

Koizumi, Y., Masumura, R., Nishida, K., Yasuda, M., & Saito, S. (2020). A transformer-based audio captioning model with keyword estimation. arXiv.

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

Kong, Q., Yu, C., Xu, Y., Iqbal, T., Wang, W., & Plumbley, M. D. (2019). Weakly labelled audioset tagging with attention neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(11), 1791–1802.

Koutini, K., Eghbal-zadeh, H., & Widmer, G. (2021). Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 1987–2000.

Labbé, E., & Pellegrini, T. (2021). IRIT-UPS DCASE 2021 audio captioning system. *Technical report*, DCASE2021 Challenge.

Lavie, A., & Agarwal, A. (2007). METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation* (pp. 228–231).

Li, X., Grandvalet, Y., & Davoine, F. (2020). A baseline regularization scheme for transfer learning with convolutional neural networks. *Pattern Recognition*, *98*, Article 107049.

Lidy, T., & Schindler, A. (2016). CQT-based convolutional neural networks for audio scene classification. In *Proceedings of the detection and classification of acoustic scenes and events 2016 workshop, vol. 90* (pp. 1032–1048). IEEE Budapest.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Liu, S., Zhu, Z., Ye, N., Guadarrama, S., & Murphy, K. (2017). Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision* (pp. 873–881).

Makav, B., & Kılıç, V. (2019). A new image captioning approach for visually impaired people. In *2019 11th international conference on electrical and electronics engineering* (pp. 945–949). IEEE.

Mei, X., Huang, Q., Liu, X., Chen, G., Wu, J., Wu, Y., et al. (2021). An encoder-decoder based audio captioning system with transfer and reinforcement learning. In *Proceedings of the 6th detection and classification of acoustic scenes and events 2021 workshop* (pp. 206–210). Barcelona, Spain.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Nariseti, C., Hayashi, T., Ishizaki, R., Watanabe, S., & Takeda, K. (2021). Leveraging state-of-the-art ASR techniques to audio captioning. In *Proceedings of the 6th detection and classification of acoustic scenes and events 2021 workshop* (pp. 160–164). Barcelona, Spain.

Neyshabur, B., Sedghi, H., & Zhang, C. (2020). Hat is being transferred in transfer learning? W. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems, vol. 33* (pp. 512–523). Curran Associates, Inc..

Özkaya Eren, A., & Sert, M. (2021). *Audio captioning using sound event detection*: *Technical report*, DCASE2021 Challenge.

Palanisamy, K., Singhania, D., & Yao, A. (2020). Rethinking CNN models for audio classification. arXiv preprint arXiv:2007.11154.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318).

Park, D. S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E. D., et al. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019, 20th annual conference of the international speech communication association* (pp. 2613–2617). ISCA.

Pellegrini, T. (2020). *IRIT-UPS DCASE 2020 audio captioning system*: *Technical report*, DCASE2020 Challenge.

Perez-Castanos, S., Naranjo-Alcazar, J., Zuccarello, P., & Cobos, M. (2020). Listen carefully and tell: An audio captioning system based on residual learning and Gammatone audio representation. In *Proceedings of the detection and classification of acoustic scenes and events 2020 workshop* (pp. 150–154).

Qian, Y., Bi, M., Tan, T., & Yu, K. (2016). Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(12), 2263–2276.

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Takeuchi, D., Koizumi, Y., Ohishi, Y., Harada, N., & Kashino, K. (2020). Effects of word-frequency based pre- and post- processings for audio captioning. In *Proceedings of the detection and classification of acoustic scenes and events 2020 workshop* (pp. 190–194).

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., et al. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2820–2828).

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems, vol. 30* (pp. 1–11). Curran Associates, Inc..

Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4566–4575).

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Won, H., Kim, B., Kwak, I.-Y., & Lim, C. (2021). Transfer learning followed by transformer for automated audio captioning. In *Proceedings of the 6th detection and classification of acoustic scenes and events 2021 workshop* (pp. 221–225). Barcelona, Spain.

Xu, X., Dinkel, H., Wu, M., Xie, Z., & Yu, K. (2021). Investigating local and global information for automated audio captioning with transfer learning. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing* (pp. 905–909). IEEE.

Xu, X., Dinkel, H., Wu, M., & Yu, K. (2020). A crnn-gru based reinforcement learning approach to audio captioning. In *Proceedings of the detection and classification of acoustic scenes and events workshop* (pp. 225–229).

Xu, X., Wu, M., & Yu, K. (2022). A comprehensive survey of automated audio captioning. arXiv preprint arXiv:2205.05357.

Xu, X., Xie, Z., Wu, M., & Yu, K. (2021). The SJTU system for DCASE2021 challenge task 6: Audio captioning based on encoder pre-training and reinforcement learning. *Technical report*, DCASE2021 Challenge.

Yang, L., & Sijun, B. (2021). The DCASE2021 challenge task 6 system : Automated audio caption. *Technical report*, DCASE2021 Challenge.

Ye, Z., Wang, H., Yang, D., & Zou, Y. (2021). Improving the performance of automated audio captioning via integrating the acoustic and textual information. In *Proceedings of the 6th detection and classification of acoustic scenes and events 2021 workshop* (pp. 40–44). Barcelona, Spain.