# CAU SUBMISSION TO DCASE 2024 TASK6: SUPERVISED AUDIO CAPTIONING SYSTEM WITH ConvNeXt AND TRANSFORMER ARCHIRECTURES

Technical Report

*Hyunhee Hong*[*]

Chung-Ang University
Department of Applied Statistics,
Seoul, South Korea,
hhh9900@naver.com

*Yunjung Lee*[†]

Chung-Ang University
Department of Applied Statistics,
Seoul, South Korea,
dldbswjd106@naver.com

## ABSTRACT

This technical report presents an audio captioning system designed to generate descriptive textual captions for audio inputs. The system employs an encoder-decoder architecture, utilizing ConvNeXt-Tiny for acoustic feature extraction and a Transformer decoder for generating the captions. The audio inputs are sampled at 32kHz, and the system learns its own word embeddings. Data augmentation techniques, specifically mixup and label smoothing, are applied to enhance the training process. The model is trained using supervised learning with a cross-entropy loss function and optimized using the AdamW optimizer. The system comprises 11,914,777 learnable parameters and 29,388,303 frozen parameters, with a total training duration of 10,760 seconds on a single NVIDIA 20TF-V100 GPU. It was trained and validated on the Clotho development-training and development-validation subsets, respectively. The system achieved competitive results, with a METEOR score of 0.187644, a CIDEr score of 0.457634, a SPICE score of 0.132791, a SPIDEr score of 0.295213, a SPIDEr-FL score of 0.293949, and a FENSE score of 0.509337, using a vocabulary size of 564 unique words. This report underscores the effectiveness of leveraging advanced neural network architectures and robust data augmentation techniques for the task of audio captioning.

***Index Terms***— ConvNeXt-Tiny, Transformer, ACC, SPICE, SPIDEr-FL

## 1. INTRODUCTION

The focus of this technical report is an audio captioning system designed to generate descriptive textual captions for audio inputs. Audio captioning is a complex task that ~~involves understanding the audio co~~ntent and translating it into coherent and contextually appropriate text. To achieve this, we employed a sophisticated architecture consisting of an encoder and a decoder. Specifically, we used ConvNeXt-Tiny as the encoder for extracting acoustic features and a Transformer decoder for generating the captions. This combination leverages the strengths of both convolutional networks in handling audio data and transformers in managing sequential data.

## 2. SYSTEM DESCRIPTION

The system processes audio inputs sampled at a rate of 32kHz. For acoustic representation, it utilizes features extracted through a pre-trained ConvNeXt-Tiny model, ensuring a robust and high-quality feature set. Unlike some other models that rely on pre-trained word embeddings such as Word2Vec or BERT, this system learns its own word embeddings, tailored specifically to the task at hand.

To enhance the training process, the system employs data augmentation techniques, specifically mixup and label smoothing. These methods help to improve the generalization capabilities of the model by providing more diverse training examples and smoothing the target labels. The core architecture of the system follows an encoder-decoder scheme, a popular choice for tasks involving sequence-to-sequence learning, such as audio captioning.

The learning scheme is supervised, meaning the model is trained with labeled data, learning to predict the output captions based on the input audio. The system does not use an ensemble approach, operating instead as a single model.

For word modelling, a transformer architecture is employed. Transformers are known for their efficiency and effectiveness in capturing long-range dependencies in sequential data, making them well-suited for generating coherent and contextually appropriate captions.

[*] Thanks to ABC agency for funding.

[†] Thanks to XYZ agency for funding.

## 3. ARCHITECTURE

### 3.1. Encoder

The encoder in our system is ConvNeXt-Tiny, a convolutional neural network specifically pre-trained for audio feature extraction. ConvNeXt-Tiny is effective in capturing detailed acoustic features, which are crucial for understanding the nuances in audio inputs. This model processes the raw audio signal and outputs a high-dimensional representation that encapsulates the essential characteristics of the audio.

### 3.2. Decoder

The decoder is a Transformer, an architecture renowned for its ability to handle sequential data with long-range dependencies. The Transformer decoder takes the encoded audio features from ConvNeXt-Tiny and generates the corresponding text caption. The attention mechanisms within the Transformer allow it to focus on different parts of the audio representation while generating each word in the caption, ensuring that the generated text is coherent and contextually relevant.

### 3.3. Loss Function

The system uses cross-entropy as its loss function. Cross-entropy loss is widely used for classification tasks, including sequence-to-sequence problems like audio captioning. It measures the performance of the model by comparing the predicted sequence of words against the true sequence, providing a robust metric for training. The goal is to minimize this loss, thereby improving the accuracy and quality of the generated captions.

## 4. EXPERIMENT

### 4.1. Dataset Used

For training, the system uses the Clotho development-training subset. The Clotho dataset includes audio and corresponding captions, with the development-training subset consisting of 3839 seconds of audio data, each paired with five captions. This subset is crucial for both audio and word modelling. The dataset is accessible via the following URL: Clotho Development-Training.

The system's validation process utilizes the Clotho development-validation subset, which includes 1045 seconds of audio data, also with five captions per audio file. This subset provides a reliable means of evaluating the system's performance during training. The validation dataset is accessible via the following URL: Clotho Development-Validation.

### 4.2. Settings

AdamW optimizer is used with a learning rate set at 5e-4. AdamW is a variant of the Adam optimizer that includes a weight decay parameter, which helps to prevent overfitting by regularizing the weights. The weight decay parameter for the optimizer is set to 2, striking a balance between learning efficiency and regularization.

To ensure stable training, the system applies gradient clipping with a norm type of L2 and a value of 1. Gradient clipping helps to prevent the gradients from exploding, which can occur in deep networks and lead to unstable training dynamics.

The system comprises 11,914,777 learnable parameters, with an additional 29,388,303 frozen parameters, which are derived from the feature extractor and other components of the model. This brings the total number of parameters involved at inference time to 41,303,080. The training duration of the entire system is 10,760 seconds, utilizing a single NVIDIA 20TF-V100 GPU.

### 4.3. Metrics

The submitted systems will be evaluated based on their performance on the withheld evaluation split. For the evaluation, it is essential that the captions generated by the systems do not contain any punctuation marks and that all letters are lowercase. Participants are therefore advised to optimize their methods accordingly. The primary metrics reported for each submitted method will include METEOR, CIDEr, SPICE, SPIDEr, SPIDEr with fluency error detection (denoted as SPIDEr-FL), FENSE, and Vocabulary. Among these, the FENSE metric will be the main criterion for ranking the methods. Detailed information on the FENSE metric can be found in the corresponding paper, available online. Additionally, several contrastive metrics will also be reported, including the traditional SPIDEr metric and some recent model-based metrics for captioning. The Vocabulary metric, which corresponds to the number of unique word types used by the system across all generated captions in the development-testing subset, has been introduced this year as a new contrastive metric. It is important to note that these contrastive metrics will not affect the overall team rankings but provide valuable insights into the system's performance. All these metrics can be computed using the aac-metrics package, ensuring standardized evaluation across different systems.

### 4.4. Results

On the development-testing split, the system achieves the following performance metrics: a METEOR score of 0.187644, a CIDEr score of 0.457634, a SPICE score of 0.132791, a SPIDEr score of 0.295213, a SPIDEr-FL score of 0.293949, and a FENSE score of 0.509337. The vocabulary size utilized by the system is 564 unique words.

## 5. CONCLUSION

This technical report provides a comprehensive overview of the submitted system, detailing its implementation, complexity, datasets used, and performance metrics. The system leverages ConvNeXt-Tiny for acoustic feature extraction and a transformer architecture for word modelling within an encoder-decoder framework. Through supervised learning, the system is trained and validated on the Clotho dataset, demonstrating competitive results across multiple evaluation metrics. This approach underscores the effectiveness of combining modern neural network architectures with robust data augmentation techniques to tackle the task of audio captioning.