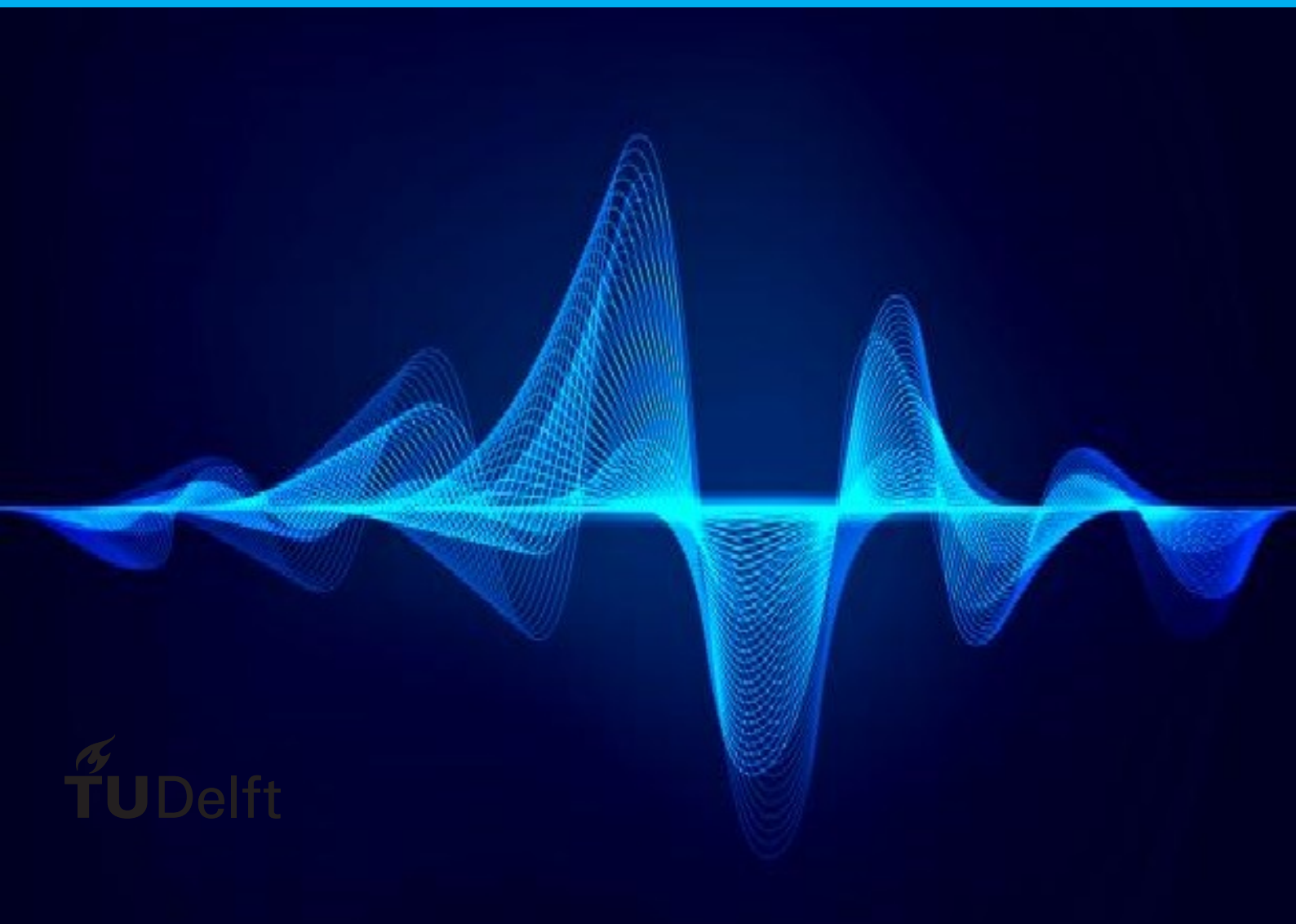
Visually grounded fine-grained speech representations learning

Tian Tian

Multimedia Computing Group - TU Delft



Visually grounded fine-grained speech representations learning

by

Tian Tian

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 25, 2020 at 10:00 AM.

Student number:	4818776	
Project duration:	November 1, 2019 – August 25, 2020	
Thesis committee:	Dr. Odette Scharenborg,	TU Delft, Associate Professor
	Dr. Elvin Isufi,	TU Delft, Assistant professor
	Dr. Nava Tintarev,	TU Delft, Assistant professor

TU Delft

Contents

Abstract

Visually grounded speech representation learning has shown to be useful in the field of speech representation learning. Studies of learning visually grounded speech embedding adopted speech-image cross-modal retrieval task to evaluate the models, since the cross-modal retrieval task allows to jointly learn both modalities and find their relationships. Specifically, the two modalities, i.e., audio and visual, were jointly embedded into a common space where the speech embeddings and the image embeddings were learned in this process. The obtained embeddings were evaluated by cross-modal retrieval task to see the model performance. Currently, the studies worked on visually grounded speech representation learning trained on scene-based datasets, such as Flickr8k, etc., which learn different objects to infer a new scene. The works that investigating the visually grounded speech representation model's ability to combine different attribute information to infer new objects are lacking. Therefore, this thesis presented a visually grounded speech representation model trained on the fine-grained datasets that contain high level details of objects to learn attribute information associated with objects to infer new objects. The proposed model adopted dual-encoder structure and used different DNN models to extract visual and audio features. An adapted batch loss was used to calculate the similarities between two modalities. Experiments were conducted to test the model performance: 1) The parameter adjusting to obtain a better-performed model. 2) Comparing with state-of-the-art models in speech-image cross-modal retrieval field and fine-grained text-image cross-modal retrieval field. 3) Ablation studies to evaluate components in the model. 4) Research on attention module to see its effectiveness. The results indicated that the proposed model was able to learn the relationships between attributes and objects to retrieve new visual objects and outperformed other visually grounded speech learning models.

1

Introduction

Learning spoken language plays an essential role in some intelligent systems that need to interact with humans, such as systems that require to recognize speech. Standard automatic speech recognition (ASR) systems are trained based on large amounts of transcribed speech data. However, among the around 7,000 languages in world, half of them do not have a written form [1], which called unwritten language. For these unwritten languages, ASR is not available since the textual transcriptions are lacking. Inspired by the way human infants learn and understand speech from exposure to spoken language and from watching visual scenes and different gestures, recent research ([5, 11, 20, 31]) that worked on speech learning learned the semantic embedding of spoken language by visual grounding. In these studies, two modalities, i.e., audio and visual, were embedded into a common space in which the speech embeddings and image embeddings were learned. The learned embeddings were evaluated by speech-image cross-modal retrieval task that the instances of one modality are retrieved by the instances of another modality. The better retrieval performance indicates the better learned embeddings.

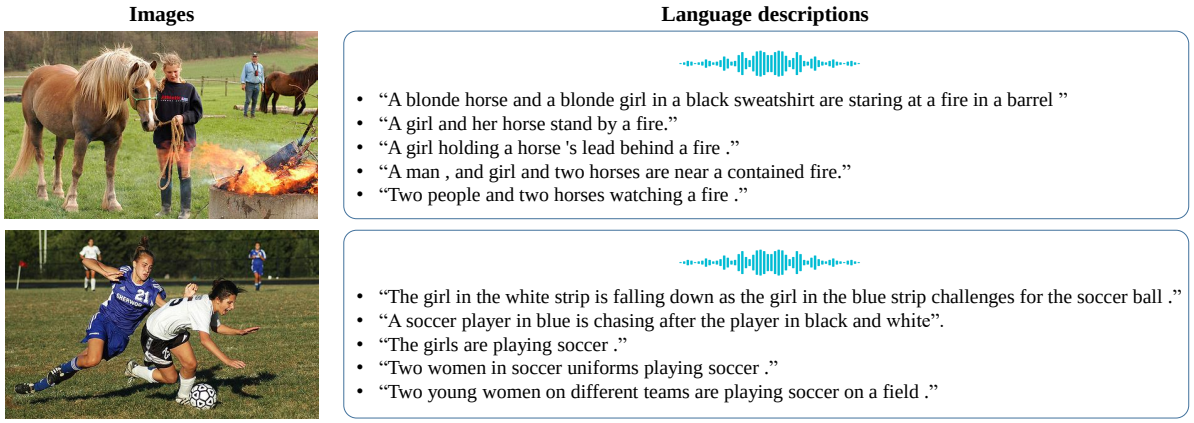


Figure 1.1: Examples of scene images and their corresponding language descriptions from Flickr8k dataset [15].

In the field of visually grounded speech representation learning, typically the speech representations are learned based on scene image datasets such as Flickr8k [15], MSCOCO [27], and Places [56]. These image datasets come with spoken captions which describe the corresponding scene images by describing the objects occurred in images, the objects' actions or states, and sometimes the attributes of the objects. For instance, The left-hand side of Figure 1.1 presents images taken from Flickr8k dataset, while the right-hand side of the figure show the corresponding image descriptions. For ease of the reader, the captions are presented in text. These textual descriptions are not used during visually grounded speech representation learning. For instance, the first image describes a scene which consists of different objects, such as a horse, girl, fire, etc., and the objects' actions or states, such as stand, staring, holding, etc. Moreover, some combinations of attributes and objects can appear in the corresponding captions to identify objects, such as “blond” (attribute) “horse” (object). Recent visually grounded

speech embedding models [5, 11, 31] were learned based on the scene-based datasets and combined different elements occurred in the images such as objects, actions or states, and sometimes attributes of objects to retrieve new scenes. However, the models focus more on learning a complete scene that contains different objects but not focuses on learning an object that contains attributes. Currently, there are no related works of learning visually grounded speech embedding that focus on investigating whether the model has the ability to learn attribute information to infer new objects. Recognizing the relationship between attributes and objects to infer new instances by combining different attributes and objects learned from known instances is an essential ability of human beings. For example, when people are told that a "zebra" is a horse-like animal combined by its unique black and white striped coat, people's knowledge of "horse" and "black and white stripes" allows them to recognize zebras even if they have never seen one before. Thus, in this thesis, I aim to train a visually grounded speech representation learning model to investigate the probability of the model to learn attribute information to infer new objects. However, in order to investigate this ability, databases that contain many attribute and object pairs are needed. Fine-grained image datasets refer to the datasets that contain hard-to-distinguish object classes, such as species of birds, flowers, etc. Fine-grained images from the same class describe the same kinds of objects and the differences between different fine-grained image classes are subtle. This characteristic of fine-grained image datasets makes the learned model pay more attention to the attributes of objects to identify different objects. This kind of dataset provides me with the chance to investigate whether new instances can be recognized by learning the relation of attributes and objects. Figure 1.2 shows two bird images from the fine-grained dataset CUB-200 [48], and the textual sentences on the right of each image are their corresponding captions. The two birds in the two images are from different classes (bird species) but they are described in the same form, which is the combinations of attributes and objects, i.e., brown body, white belly, yellow on its chest, etc. According to the related methods [5, 11, 31] to learn visually grounded speech embeddings, a fine-grained visually grounded speech embedding model can also be implemented by embedding fine-grained images and corresponding speech descriptions jointly into a common space to learn their relationships and evaluated by speech-image cross-modal retrieval task. If the model can be used to retrieve new bird categories based on known bird images and their corresponding spoken captions, it can show the ability to learn the relationships between attributes and objects, which have not been investigated by previous works.

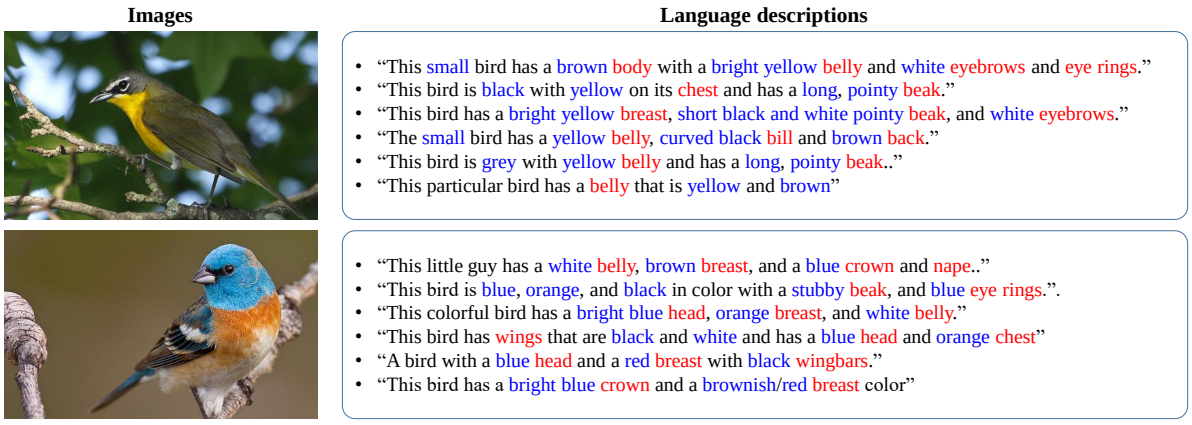


Figure 1.2: Examples of images from the CUB-200 dataset. In the captions, the words of attributes (in blue) and objects (in red) are marked with different colors. [15]

In short, with this thesis I aim to investigate whether the visually grounded speech learning method can learn the relationships between attributes and objects to infer new objects. To that end, I propose a visually grounded fine-grained speech embedding network.

1.1. Research questions

The main research question of this thesis is:

Can the visually grounded speech representation learning model combine different

attribute information to retrieve new visual objects?

According to the previous introduction and discussion to the main task of this thesis, the main research question is given. The question is to investigate whether the proposed visually grounded speech representation model to be able to learn attribute information and combine different attributes and objects together to infer new objects. To implement this work, a specific task is needed. Related works [10–12] adopted cross-modal retrieval task to evaluate the performance of their learned model. Thus, the specific task will be investigated in this thesis. Moreover, fine-grained datasets contain attribute and objects information in both images and captions, which can be used to learn the attribute information. Thus, the datasets used in this thesis will also be investigated.

The models in related works [10–12] were all trained on deep neural networks. Recent years, deep neural networks (DNN) show great potential in multimedia learning field, such as ASR [32], image recognition [13], natural language processing (NLP) [46], etc. DNN-based speech-image retrieval methods [5, 10–12, 31] become mainstream in speech-image cross-modal retrieval field. Thus, DNNs will be good alternatives to construct my speech-image cross-modal retrieval model. Moreover, the appropriate structure of the model by using DNNs is a question to be investigated in this thesis.

In addition to the model structure, the loss function is also essential in the deep learning field. A loss function is to optimize the parameter values in a neural network model. Loss functions map a set of parameter values for the network onto a scalar value that indicates how well those parameters accomplish the task the network is intended to do. To obtain a good-performing DNN model in this thesis, an appropriate loss function for the proposed model is required. In cross-modal retrieval field including speech-image retrieval and fine-grained image-text retrieval, there are several loss functions proposed in the literatures [10, 41, 54, 55] that would be useful to my work and I will describe them in the next chapter (chapter 2).

Specifically, in the deep learning field, the attention mechanism is developed rapidly for image recognition [13] and natural language processing [46]. It was adopted to pay more considerable attention to certain factors when processing the data. Thus, I considered whether the attention mechanism would be useful for learning speech and image representations.

According to the previous discussions about the work of this thesis, the main research question can be divided into sub-tasks to better look into the work. The relevant sub-questions are presented as follow:

- *What kind of tasks and datasets can be used to evaluate the ability of the learned speech embedding model on inferring new visual objects?*
- *What is the appropriate deep learning model structure for feature extraction of speech and images?*
- *What is the appropriate loss function for the visually grounded speech representation learning model?*
- *Is the attention mechanism useful for visually grounded speech semantic learning?*

According to the first sub-question, the specific task will be introduced in chapter 2. Besides, the datasets that used to evaluate the proposed model will be introduced in detail in chapter 3. In order to answer the second sub-question, the deep learning model that is proposed in this thesis to learn visually grounded fine-grained speech representation is introduced in chapter 4. The evaluation of the proposed model is in chapter 5. According to the third sub-question, different loss functions will be investigated and the appropriate one for the proposed model will be introduced in chapter 4 and evaluated in chapter 5. In order to better look into the last sub-question, a comprehensive introduction to the attention mechanism is presented in chapter 2. The specific implementation and evaluation of the attention mechanism will be introduced in chapter 4 and chapter 5 respectively.

1.2. Thesis outline and contributions

This thesis consists of six chapters. The next chapter (Chapter 2) will provide a comprehensive theoretical background on visually grounded speech representation and cross-modal retrieval. Some required deep learning knowledge and related works in the visually grounded speech representation field will also be discussed. In Chapter 3, the datasets used in this thesis and the pre-processing operations for the

datasets will be discussed. In Chapter 4, the proposed model for visually grounded speech representation learning will be introduced. In Chapter 5, the experimental setup, the results obtained from the experiments will be presented and discussed. Finally, this thesis will have a conclusion, a discussion for the limitations of my work and future works for improvement in Chapter 6.

The main contributions of this thesis are presented in the following:

- A visually grounded speech embedding learning model was proposed in this thesis which was able to learn fine-grained semantic information.
- Extensive experiments showed that the proposed method achieves state-of-the-art performance for the speech-based image cross-modal retrieval task not only on the fine-grained datasets but also on the scene image dataset Flickr8k.
- Using an adapted batch loss function, the performance of the model was substantially improved for the retrieval task.
- A detailed analysis showed the effectiveness of the attention mechanism and other components in the proposed model.

2

Background and related work

A comprehensive introduction to visually grounded speech model learning and related works of this task will be shown in section 2.1, which gives a basic understanding of visually grounded speech representation learning. Moreover, the specific task to evaluate my visually grounded fine-grained speech learning is investigated to be cross-modal retrieval in section 2.1. An introduction to the cross-modal retrieval task will be presented in section 2.2, which includes the reason to choose it (section 2.2.1), and its definition and types (section 2.2.2). Among all types of cross-modal retrieval methods, DNN-based cross-modal retrieval is the mainstream method and is adopted in this thesis. Its concept and several related deep learning techniques are introduced in detail in section 2.3. In section 2.4, the speech-image cross-modal retrieval task is introduced. Additionally, several speech-image retrieval methods [5, 11, 31, 42] for this task will be introduced. I will compare the results of my proposed model with those of the models of above-mentioned papers in chapter 5. Finally, some text-image cross-modal retrieval methods in [41, 54, 55] will also be introduced and compared with, because my model learned a lot from these works. An introduction to the text-image cross-modal retrieval and the above text-image retrieval methods will be presented in section 2.5.

2.1. Visually grounded speech representation learning

The meaning of a piece of speech can be obtained by learning its semantic information. Traditional speech recognition systems usually transcribe speech into text information to learn the semantic information of the speech. The transcription requires massive transcribed speech data. The cost of collecting these resources is enormous, so it is no surprise that ASR is available for very few of the more than 7,000 languages spoken across the world [38]. People found that humans are capable of learning language from raw sensory input, and furthermore learn to communicate long before they are able to read [31]. Inspired by human learning and understanding language, many works have moved towards more realistic inputs (visual input) instead of the textual input, while modelling speech [11].

Visually grounded speech representation learning has aroused much attention during the past few years. Researchers considered to jointly learn visual and audio input to implement visually grounded speech representation learning [10–12]. Harwath et al. [10] adapted text-based caption-image retrieval (e.g. [21]) and showed that it is possible to perform speech-image retrieval to jointly learn speech and image. Subsequently, in a series of studies by Harwath et al., [11, 12] use image captioning datasets to learn spoken language from the visual context through a convolutional neural network model. Chrupala et al. [5] projects spoken utterances and images to a joint semantic space to train a multi-layer recurrent highway network model of language acquisition from visually grounded speech signal. Kamper et al. [19] trained an image-to-words multi-label visual classifier to predict a set of words referring to aspects of the scene, and then used soft labels to train a neural network to map spoken captions to these soft unordered word targets. So that the speech model can predict which words occur in a spoken utterance. Similarly, Kamper et al. [20] extended the work in [19] in which all utterances in a speech collection that are semantically relevant to a given query keyword could be retrieved.

Above-mentioned works aim to learn visually grounded speech embedding with untranscribed speech but through different forms of tasks. [5, 10–12] applied cross-modal retrieval task that the items in one

modality could be retrieved by another to evaluate their models. [19, 20] focus on speech unit discovery and spoken word prediction by visual labels. My thesis is more similar to the former papers. Thus, cross-modal retrieval between speech and image is adopted to evaluate the performance of the proposed model in this thesis.

2.2. Cross-modal retrieval

Cross-modal retrieval between two modalities allows users to get the results of one media type by submitting a query of another media type [36]. Learning the relationships between two modalities is the main idea of cross-modal retrieval of two media types. This thesis aims to learn the visually grounded speech representation that involves the two modalities of speech and image. The reason for choosing this task will be explained in the following. Subsequently, the concept of cross-modal retrieval will be given.

2.2.1. The reason for choosing cross-modal retrieval

Since visually grounded speech representation learning is an abstract concept, it needs to be implemented by a specific task. Most studies [5, 11, 12, 31] adopted cross-modal retrieval task, which can jointly learn two modalities to find their relationships. The mainstream method of cross-modal retrieval is to learn a shared space for features of different media types and measure the similarities among them in one common space [36]. Thus, it provides a method to learn both modalities jointly and their relations. Moreover, the retrieval is to rank the items in one modality by relevance to a query item in another modality and vice versa, which could evaluate the performance of modality representation, such as speech representation. Therefore, in this thesis, the cross-modal retrieval between image and speech will be used to jointly learn two modalities and using the items of one modality to retrieve another.

2.2.2. The definition and types of cross-modal retrieval

In general, the definition [36] of cross-modal retrieval is as follows. I will use two media typess X and Y as examples to give the definition for cross-modal retrieval. The training data is denoted as $\mathscr{D}_{tr} = \{X_{tr}, Y_{tr}\}$, in which $X_{tr} = \{\boldsymbol{x}_p\}^{n_{tr}}$, where n_{tr} represents the number of media instances for training and x_p denotes the p-th media instance. Similarly, $Y_{tr} = \left\{\boldsymbol{y}_p\right\}^{n_{tr}}$. x_p and y_p represents instances of the two modalities that describe relevant semantics and can be seen as matched pairs. For the datasets divided into classes, such as fine-grained datasets, class labels for training data can be denoted as $\left\{c_p^X\right\}^{n_{tr}}$ and $\left\{c_p^Y\right\}^{n_{tr}}$, which indicate the classes that the instances belong to. The goal of cross-modal retrieval is to compute similarities $\mathbf{sim}\left(\boldsymbol{x}_p, \boldsymbol{y}_p\right)$ and retrieve relevant instances of one modality by a query of another modality.

The main challenge of cross-modal retrieval is the problem of the "modality gap", which means that the representation of different media types is inconsistent and located in different feature spaces. Thus, it is challenging to measure the similarity between them. Many methods have been proposed to solve this problem by analyzing the rich correlations in cross-modal data. As mentioned, the mainstream method of cross-modal retrieval is "common space learning methods" that aims to learn the intermediate public space for the characteristics of different media types and measure the similarity between them [36]. Among the methods in common space learning, DNN-based methods take deep neural network as the basic model and aim to make use of its strong abstraction ability for cross-modal correlation learning. With the great advance of deep learning, recent works in speech-image retrieval [5, 10–12, 31] apply DNN to learn both modalities. DNN has a strong ability to learn nonlinear relationships and is the commonly used method that perform public space learning on data of different modalities. Thus, this thesis will apply DNN-based common space learning method to implement image-speech retrieval task. Next section will discuss the DNN-based cross-modal retrieval and introduce several essential deep learning techniques.

2.3. Deep neural network techniques for cross-modal retrieval

With the rapid development of deep learning, deep neural networks (DNN) have shown their potential in cross-modal applications. Inspired by the great success of DNNs in representation learning [26], several DNN-based methods have been proposed to learn the complex nonlinear transformations for

cross-modal retrieval. It turns out that deep neural network architectures and training schemes are essential for cross-modal retrieval tasks [10–12]. A comprehensive introduction to DNNs and related techniques is given in the following sections.

2.3.1. Deep neural network.

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers [17]. DNNs compose computations performed by many layers. Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains, such as convolutional neural networks (CNNs) in computer vision [40], recurrent neural network in natural language processing [29], etc.

DNNs used to model multimodal representation learning was first proposed in [33]. Following this idea, some similar deep architectures are proposed and achieve improvement in cross-modal retrieval. Recent works of DNN-based cross-modal retrieval between two modalities adopted an architecture that consists of two subnetworks coupled at the code layers and jointly computes the correlation loss [36]. Some works consist of two auto-encoders, such as independent component multimodal auto-encoder (ICMAE) [53] to jointly learn the two modalities in a shared space and reconstruct the original input of one modality into another modality. Other works [10, 11] consist of two encoders (a part of the auto-encoder), which dropped the decoder part of the auto-encoder, and only learn the embeddings of two modalities to calculate the similarity. My model is also a two-encoder structure networks model. Thus, before giving an introduction to the encoder, the auto-encoder is needed to be introduced.

2.3.2. Auto-Encoder.

The auto-encoder is an unsupervised learning technique in which people leverage neural networks for the task of representation learning. In a standard auto-encoder presented in Figure 2.1, the network can be viewed as consisting of two parts: an encoder function $h = f(x)$ and a decoder that produces a reconstruction $r = g(h)$ [17]. Internally, h is a hidden layer. In the figure, input x is mapped to an output r through an internal representation h. Traditionally, auto-encoders were used for dimensionality reduction or feature learning, so they were designed to be unable to learn to copy the input perfectly. They are forced to prioritize which aspects of the input should be copied and learned useful properties of the data.

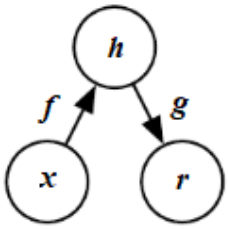


Figure 2.1: The general structure of an autoencoder [17]

Convolutional neural network (CNN) is often used to extract visual features. Harwath and Glass [10] applied Region Convolutional Neural Network (RCNN) [8] to do the image feature extraction which demonstrated the power of the CNN features in speech-image cross-modal retrieval. Thus, an introduction to CNNs will be presented in the following.

2.3.3. CNNs.

In deep learning, convolutional neural networks are a special kind of multi-layer neural networks, which are designed to recognize visual patterns directly from pixel images with minimal pre-processing [17]. The attractive feature of CNN is its ability to exploit spatial or temporal correlation in data. A CNN is a feed forward multi-layer network, where each layer uses many convolution kernels for multiple transformations [25]. Convolution operations help extract useful features from locally correlated data points. The output of the convolution kernel is then assigned to a non-linear processing unit (activation function), which not only helps to learn the abstraction, but also embeds the non-linearity in the feature space.

Recent years, the improvement of CNN models increased rapidly to make CNN scalable to large, heterogeneous, complex and multi-class problems [22]. The innovation of CNNs has different aspects, such as modification of processing units, parameter and hyper-parameter modification strategies, design patterns and connectivity of layers, etc. [17] During the development of CNNs, different networks were adopted in speech-image cross-modal retrieval task. Some commonly used CNN models in the speech-image retrieval field were summarized roughly as below. [11][5][12] adapted VGG-16, [9][55] used VGG-19, [41] used ResNet-101, [31][30] used ResNet-152, etc. VGG and Residual network are two types of the CNN architecture, which are used frequently in the speech-image cross-modal retrieval task for the image feature extraction. The two CNN networks will be discussed in the following.

VGG Neural Networks [45] are proposed by the Visual Geometry Group in [45]. The network makes the improvement over AlexNet [24] by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3X3 kernel-sized filters one after another. Multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network, which enables it to learn more complex features and have lower cost.

VGG released two different CNN models, specifically a 16-layer model and a 19-layer model. The main difference between the two VGG networks is the depth. The structure of the VGG network is very consistent. From the beginning to the end, 3x3 convolution and 2x2 max pooling are used. The advantage of VGG is the simple structure. The entire network uses the same size of convolution kernel size (3x3) and maximum pooling size (2x2). Also, the combination of several small filter (3x3) convolutional layers is better than one large filter (5x5 or 7x7) convolutional layer. It is verified that the performance can be improved by continuously deepening the network structure.

However, VGG consumes more computing resources and uses more parameters (not the parameters in 3x3 convolution), resulting in more memory usage. Most of the parameters are from the first fully connected layer, and VGG has 3 fully connected layers. Compared with other methods, such as AlexNet and GoogleNet [17], VGG has a large parameter space.

Residual Neural Network (ResNet) [13]. In 2015, the concept of skip connections introduced by ResNet [13] for the training of deep CNNs gained popularity, and afterwards, this concept was used by most of the succeeding networks. The residual network is one of the most successful convolutional neural network architectures for feature extraction. Shortly after the first success of convolutional neural networks, the scientific community realized that it was necessary to have a deeper network to avoid overfitting the data set. However, stacking more layers in the network will cause the gradient vanishing problem which makes the gradient infinitely small and meaningful learning no longer possible. The core idea of ResNet is introducing a so-called "identity shortcut connection" that skips one or more layers, as shown in the Figure 2.2.

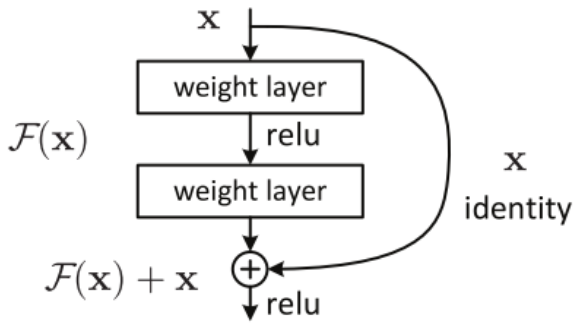


Figure 2.2: Example for a residual layer [13]

The logic behind the residual unit to solve the gradient vanishing problem can be explained as follows. Imagine a network A whose training error is x. Now build Network B by stacking more layers on top of A. These new layers do nothing, just copy the output of the previous A. These new layers are called C. This means that network B should have the same training error as A but in fact B is worse. The only reason is that it is not easy to learn the identity mapping with the added layer C. In order to solve this problem, the residual module establishes a direct connection between input and output, so that the newly added layer C only needs to learn new features based on the original input

layer, that is, to learn the residual. Using the residual module, a 152 layers residual network can be trained. The accuracy of ResNet is higher than VGG, and the calculation efficiency is also higher than VGG according to [13]. ResNet mainly uses 3x3 convolution, which is similar to VGG and insert a short-circuit connection into the residual network.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2.3: Architectures for different forms of ResNet from [13]. The size of building blocks is show in brackets, with number of blocks stacked.

Since the ResNet is more relevant to my work, the detailed internal structure of different forms of ResNet will be introduced. There are five main forms of ResNet: ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152. Figure 2.3 shows the 5 different internal architectures of ResNet. Each network consists of three main parts: input, output and intermediate convolution part. Although ResNet has rich variants, they all follow the above-mentioned structure. The differences between networks are mainly in the block parameters and the convolution part. Two blocks were proposed in [13], one is a basic block and the other is a "bottleneck" building block, as shown in Figure 2.4. In the figure, the left one is Basic Block, which is used for ResNet with less than 50 layers (usually ResNet18, ResNet34). The right is "BottleNeck" building block, which is used for ResNet greater than or equal to 50 layers (usually ResNet50, ResNet101, and ResNet152). In a deeper network, BottleNeck contain fewer parameters, but can also maintain the performance improvement. However, from ResNet50 to ResNet152, top5-error decreased by 1% (top1-error did not even 1%), but FLOPs (floating point operations per second) increased by about 3 times. Thus, it is important to choose the appropriate network for specific task.

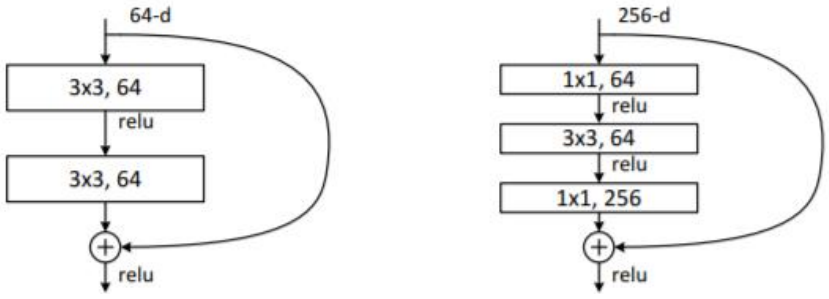


Figure 2.4: Two deeper residual functions in [13]. Left: a building block (on 56×56 feature maps) for ResNet34. Right: a "bottleneck" building block for ResNet-50/101/152.

Harwath and Glass [10] treated speech frames as 1-channel images to use CNNs to encode speech. However, recurrent neural networks (RNN) have been shown to well deal with sequence data such as speech. For example, some RNN variants, such as Gated Recurrent Unit (GRU) and long short-term memory (LSTM) [14], have been applied to encode speech [5][31]. Thus, the next section will discuss the RNN and its variants.

2.3.4. RNN with its variants.

Recurrent Neural Networks (RNNs) are a class of neural networks that allow previous outputs to be used as inputs and having hidden states. RNNs can use their internal memory to process arbitrary sequences of inputs. In general, the CNN and DNN are not good at dealing with the temporal information of input data. Therefore, in research areas that contain sequential data, such as text, audio, and video, RNNs are dominant. In RNN, the data travel both forward and backward by introducing loops in the network. However, when data passes through units in RNN for a long time, the connection of related information may be lost. As a result, some variants of RNNs were proposed to preserve the relevant information in the data, such as Gated Recurrent Unit (GRUs) [4], Long Short-Term Memory (LSTM) [14], etc.

LSTM [14]. When RNNs are used to train long sentences, the shortcoming of RNNs of not capable to handle long-term dependencies emerges. As the gap between the related inputs grows, it is difficult to learn the connection information. Thus, LSTM was proposed, which improved the memory capacity of the standard recurrent cell by introducing a "gate" into the cell. Figure 2.5 shows an LSTM cell. In the figure, there are two states to convey information in LSTM which are cell state and hidden state. Comparing to traditional RNN, LSTM adds a cell state that able to store long memory in the network. $c(t-1)$ and $c(t)$ represent the cell state at time $t-1$ and t respectively. The LSTM has the ability to remove or add information to the cell state, regulated by gates. $h(t-1)$ and $h(t)$ denote hidden states. $x(t)$ is the input at time t. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The sigmoid(σ) layer outputs numbers between zero and one, describing how much of each component should be let through. The tanh layer is to push the values to be between 1 and 1. Thus, the whole LSTM can be divided into following steps.

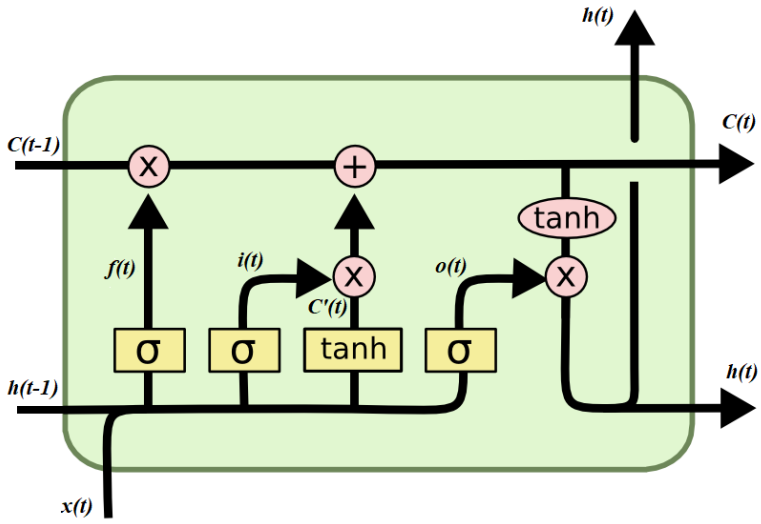


Figure 2.5: The Long Short-Term Memory (LSTM) cell [14]

The first step is to decide what information is going to be forgotten from the cell state. This decision is made by sigmoid(σ) layer called the "forget gate layer." It looks at $h(t-1)$ and $x(t)$, and outputs a number between 0 and 1 for each number in the cell state $C(t-1)$. Here:

$$f(t) = \sigma\left(W_f \cdot [h(t-1), x(t)] + b_f\right) \tag{2.1}$$

The next step is to decide what new information is going to be stored in the cell state. This has two parts. First, a sigmoid(σ) layer called the "input gate layer" decides which values to be updated. Next, a tanh layer creates a vector of new candidate values, $C'(t)$, that could be added to the state. In the next step, they are combined to create an update to the state. The functions are formulated as:

$$\begin{aligned} i(t) &= \sigma\left(W_i \cdot [h(t-1), x(t)] + b_i\right) \\ C'(t) &= \tanh\left(W_C \cdot [h(t-1), x(t)] + b_C\right) \end{aligned} \tag{2.2}$$

The third step is to update the old cell state $C(t-1)$ into the new cell state $C(t)$. The formula is shown

below:

$$C(t) = f(t) * C(t-1) + i(t) * C'(t) \tag{2.3}$$

Finally, it needs to decide what is going to be outputted. First, a sigmoid layer is run which decides what parts of the cell state is going to be outputted. Then, the cell state is going through Tanh and multiply it by the output of the sigmoid gate. The functions are formulated as:

$$\begin{aligned} o(t) &= \sigma\left(W_o\left[h(t-1), x(t)\right] + b_o\right) \\ h(t) &= o(t) * \tanh\left(C(t)\right) \end{aligned} \tag{2.4}$$

Thus, in LSTM, the gates could control what information to preserved and which is forgotten. Because of their powerful learning capacity, LSTM works tremendously well and have been widely used in various kinds of tasks, such as NLP [29], speech recognition [18], etc.

Gated Recurrent Unit(GRU) [4]. Another RNN variant commonly used in natural language representation is the Gated Recurrent Unit. Like LSTM, it is also proposed to solve the problems of long-term memory. The performance of GRU and LSTM are almost the same in some cases, but GRU is computationally cheaper [47].Compared with LSTM, the GRU has fewer parameters which can greatly improve the training efficiency. Besides, GRUs have been shown to exhibit even better performance than LSTM on certain smaller datasets [6].

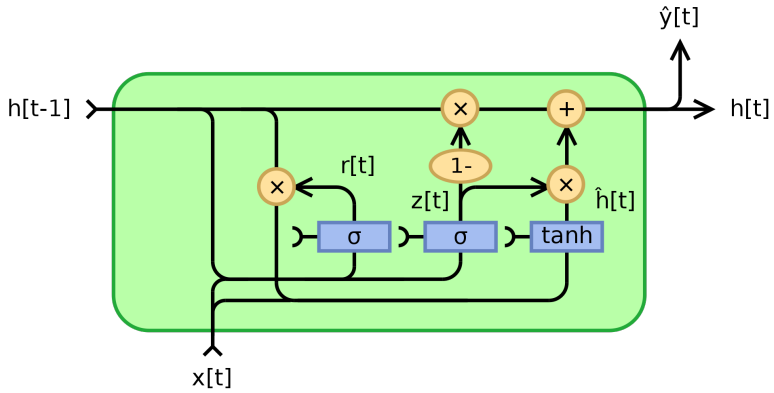


Figure 2.6: The Gated Recurrent Unit (GRU) cell [17]

Figure 2.6 shows the architecture of a GRU cell. The input and output structure of GRU is the same as that of a normal RNN, and its internal structure is similar to a LSTM, with some gates. A GRU combines the forget gate and input gate from the LSTM into a single "update gate", and merges the cell state and hidden state. The resulting model is simpler than the standard LSTM model and is becoming more popular. The two gates of GRU, reset gate r_t (see Eq. 2.5) and update gate z_t (see Eq. 2.6), have the same calculation method.

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{2.5}$$

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{2.6}$$

The calculation of the candidate hidden layer $\tilde{h}_t$ is as follows:

$$\tilde{h}_t = \tanh\left(W x_t + r_t U h_{t-1}\right) \tag{2.7}$$

Which can be regarded as new information at the current moment, where r_t is used to control how much previous memory needs to be retained, for example, if r_t is 0, then $\tilde{h}_t$ contains information about the current word only. Finally, z_t controls how much information needs to be forgotten from the hidden layer h_{t-1} at the previous moment, how much hidden layer information $\tilde{h}_t$ needs to be added at the current moment, and finally h_t is obtained. The difference between this process and LSTM is that there is no output gate in GRU, which saves many parameters. The equation is shown below:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{2.8}$$

For GRU and LSTM, on one hand, GRU has fewer parameters, the training time is shorter and requires less data to generalize. On the other hand, if there is enough data, the strong expressive power of LSTM may produce better results.

2.3.5. The attention mechanism.

In addition to above deep neural networks, some new techniques have been proposed, such as "attention", to deal with feature representation. The essence of the attention mechanism is inspired by the human visual attention mechanism. Roughly speaking, when our vision perceives things, it is generally not a scene that is seen all the time from beginning to end but is often observed and pays attention to a specific part according to needs. When people find that a scene often shows something they want to observe in a certain part, people will learn to focus on that part when a similar scene appears again in the future. In fact, the attention mechanism is actually a series of attention distribution coefficients, that is, a series of weight parameters. When some regions are learned to be focused on, the attention mechanism allocates high weight to those regions. The attention mechanism can improve the ability of the neural network to process information just like how the human brains handle overloaded information.

The attention mechanism is widely applied in many fields, such as natural language processing (NLP) [29], computer vision [7], etc. However, different applications have different kinds of attention mechanism. In the fields of NLP, the attention mechanism is usually used to learn natural language which can be represented as a sequence, while for the computer vision tasks, such as object detection and image recognition, the input will be an image which do not contain any sequence information. Thus, based on different applications, two types of attention mechanism are adopted.

RNN-based attention. In general, RNN and its variants can be trained in a sequence-to-sequence (seq2seq) architecture which has an input layer, an output layer and an autoencoder between input and output. However, the traditional seq2seq model lacks discrimination of the input sequence, therefore the attention mechanism is introduced in [4] to solve this problem. The attention mechanism computes weights of each hidden states and learns the alignment between the input sequence and the output sequence. Recent years, different types of RNN-based attention mechanism have been proposed, such as global attention [28], local attention [28] and self-attention [46], where self-attention performs the best. The traditional attention mechanism calculates the weights based on the hidden state of the source sequence and target sequence, and the result is the dependency between each word on the source sequence and each word on the target sequence. The self-attention is different. It is carried out in the source sequence and the target sequence respectively which can capture some syntactic features between words in the one sentence. Thus, after using the self-attention mechanism, it will be easier to capture the long-distance interdependent features in the sentence.

Attention in computer vision. In the field of image recognition, it is usually used to classify the objects in the image. For example in an image recognition task, a bird is classified to recognize its species. In order to distinguish different birds, in addition to grasping the picture as a whole, more attention is paid to local information, such as head, body, feet, etc., while for the background information in the image is not necessary for the bird recognition. Therefore, the introduction of the attention mechanism in the field of image recognition allows the deep learning model to pay more attention to certain local information.

2.3.6. Gradient-weighted Class Activation Mapping (Grad-CAM) [43]

Although the CNN model shines in various tasks, it has always been criticized for lack of interpretability. In response to this problem, in the past few years, in addition to seeking explanations from the theoretical level, researchers have also proposed some visual methods to intuitively understand the internal mechanism of CNN. Grad-CAM was proposed based on CAM [56], which both methods have done a relatively sufficient study on the visualization technology of Class Activation Map. Before introduced Grad-CAM, CAM is needed to be mentioned. When studying global average pooling (GAP), the authors of CAM found that GAP is not only a regularity, but also reduces over-fitting. After a little improvement, it can make CNN have the ability to locate. Though the RCNN has been explained that CNN contains the location information of the target, when performing the classification task, the fully connected layer is introduced and discards the spatial location information of the target. Thus, CAM replaced the fully connected layer by GAP in the last convolutional layer, which will output the average of each feature map, and then perform a weighted summation to get the final output. Correspondingly, the weight of the classification layer could be used to weight the last layer of feature map. Usually the value of each position on the feature map is activated when there is a certain pattern in its perception field, and the final class activation map is a linear combination of these patterns. For simplicity, the authors directly used upsampling to restore the class activation map to the same size as the original

image. By superimposing, we can know which areas are closely related to the final classification result.

Unlike CAM, the feature weighting coefficient of CAM is the weight of the classifier, and the weighting coefficient of Grad-CAM is obtained through back propagation. Although CAM is simple, it requires to modify the structure of the original model, which needs to retrain the model. If the cost of training is very high, it is almost impossible to retrain for the CAM. Thus, Grad-CAM was proposed to solve the problem. It uses the global average of the gradient to calculate the weight. First, the definition of the weight of feature map k to classification c is:

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \tag{2.9}$$

Then the result is weighted on the last layer of feature maps, and linearly combined to the ReLu activation function to get:

$$L_{\text{Grad-CAM}}^c = \text{Re}\,LU\Big(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}}\Big) \tag{2.10}$$

The size of the obtained heatmap is consistent with the size of the feature map.

2.4. Speech-image cross-modal retrieval

As mentioned, speech-image cross-modal retrieval is adopted to evaluate the visually grounded speech embedding learning task. Different models have been proposed to learn the image and speech representation in speech-image cross-modal retrieval field. Four models will be used to compare with my model are shown in Table 2.1. In the table, each row represents a method to be compared. The first column represents the reference of each method. The second column and third column show the model of both encoders in each work. If there is a work containing multiple models or tasks, the best performed model, or the most related model is shown. The fourth column is the loss function corresponds to their models. The last column shows the datasets that used to train their models. Harwath et al.[11] proposed a classical model that have dual-encoder to encode the two modalities. Merkx et al.[31] proposed a state-of-the-art model which used advanced DNN technologies, such as the attention mechanism. These two models [11][31] will be reproduced in this thesis to compare with my model that trained on fine-grained datsets. Chrupała et al.[5] and Scharenborg et al.[42] presented multiple tasks in their work. The image-speech cross-modal retrieval task is one of their tasks. Their models trained on the Flickr8k dataset which can be directly compared with my model that trained on the Flickr8k dataset. Moreover, Merkx et al.[31] also trained on the Flickr8k dataset, which will also be compared with my model that trained on the Flickr8k dataset. A comprehensive introduction to the above-mentioned methods will be given as follows.

Table 2.1: Networks, loss functions and the datasets that used to train the models in different comparison works.

Method	Image encoder	Speech encoder	Loss function	Dataset
Harwath et al.[11]	VGG-16	CNN	bi-modal triplet loss	Places [56]
Merkx et al.[31]	ResNet-152	CNN + Bi-GRU + Attention	bi-modal triplet loss	Flickr8k [15]
Chrupała et al.[5]	VGG16	CNN + Attention	bi-modal triplet loss	Flickr8k
Scharenborg et al.[42]	VGG16	pBLSTM	bi-modal triplet loss	Flick8k

Unsupervised Learning of Spoken Language with Visual Context [11]. The authors investigate deep neural network architectures for the purpose of learning high-level semantic concepts across both audio and visual modalities. They adopted dual-encoder structure for their model which one encoder is the image encoder and another is the speech encoder. For the visual encoding, VGG16 layer network was used to process the images. They discarded the classification layer and took the 4096-dimensional activations of the penultimate layer to represent the input image features. For the

speech encoding, the authors used a log mel-filterbank spectrogram to represent the spoken audio caption associated with each image and treated the spectrogram as a 1-channel image to extract features by CNNs. The model was designed to calculate a similarity score for any given image and caption pairs, where the score should be high if the caption was relevant to the image and low otherwise. The final layer of each branch outputs a vector of activations, and the dot product of these vectors is taken to represent the similarity between the image and the caption. The loss is the bi-modal triplet loss that is similar to a standard triplet loss [3]. The triplet loss function is defined as:

$$\mathscr{L} = \sum_{j=1}^{B} \max\left(0, S_j^c - S_j^p + 1\right) + \max\left(0, S_j^i - S_j^p + 1\right) \tag{2.11}$$

Where each batch consists of B ground truth pairs, each of which is paired with one impostor image and one impostor caption randomly sampled from the same batch. S_j^p denote the cosine similarity score between the j-th ground truth pair, S_j^c is the score between the original image and the impostor caption and S_j^i is the score between the original caption and the impostor image. This loss function encouraged the model to assign a higher similarity score to a ground truth image/caption pair than a mismatched pair by a margin of 1.

Language learning using Speech to Image retrieval [31] proposed a state-of-the-art model which was also a dual-encoder architecture model with one for the image encoding and another for the speech encoding. The two encoders were trained to make the embeddings of an image-caption pair lie close to each other in the embedding space. For the image encoder, the image features were extracted by a pre-trained image recognition model: ResNet-152 [13]. For the speech encoder, the authors used two types of acoustic features: Mel Frequency Cepstral Coefficients (MFCCs) and Multilingual Bottleneck (MBN) features, and started with an embedding layer (for text caption encoding, not for speech caption) and then fed into an RNN, followed by a self-attention layer. Their loss function is also the bi-modal triplet loss as used in [11], shown in Equation 2.11.

Representations of language in a model of visually grounded speech signal [5] used a multi-layer, gated recurrent neural network (RHN) to model the temporal nature of speech signal. The model consists of two parts: a speech encoder, and an image encoder. The speech encoder starts from MFCC speech features, while the image encoder starts from features extracted with a VGG-16 pre-trained on ImageNet, then goes through a simple linear projection, followed by normalization to unit L2 norm:

$$\mathrm{enc}_i(\mathrm{i}) = \mathrm{unit}(\mathrm{Ai} + b) \tag{2.12}$$

The speech encoder consists of a 1-dimensional convolutional layer, whose output feeds into a Recurrent Highway Network with 4 layers and 2 microsteps, whose output in turn goes through an attention-like lookback operator, and finally L2 normalization:

$$\mathrm{enc}_u(\mathbf{u}) = \mathrm{unit}\left(\mathrm{Attn}\left(\mathrm{RHN}_{\mathrm{k,L}}\left(\mathrm{Conv}_{\mathrm{s,d,z}}(\mathbf{u})\right)\right)\right) \tag{2.13}$$

The loss function is same as the bi-modal triplet loss used in [11], shown in Equation 2.11.

Speech Technology for Unwritten Languages [42] presents three speech technology applications, which are end-to-end (E2E) speech-to-translation, speech-to-image retrieval and image-to-speech. Here, the second task which is speech-to-image retrieval will be mainly discussed since it is the closest to my work. Images are pre-trained using VGG16. The speech encoder starts from MFCC speech features and is modeled as a pyramidal bidirectional long-short term memory network (pyramidal biLSTM): a bi-LSTM with three hidden layers. The loss function is also the bi-modal triplet loss shown in Equation 2.11.

The above four papers mainly focus on visually grounded speech representation learning that using speech-image cross-modal retrieval.

2.5. Text-image cross-modal retrieval

The text-image cross-modal retrieval task has many applications, such as image captioning [21], image labeling [40], etc., which attracts a lot of interests. Methods in text-image cross-modal retrieval have been developed rapidly and diversified. Some advanced models and loss functions of text-based image retrieval can be extended to learn visually grounded speech representation, such as [41, 54, 55]. In

[10][11], the authors adapted text-based caption-image retrieval [21] to learn their speech-image cross-modal retrieval models. Since my work focuses on learning fine-grained image-speech retrieval which does not have any experience in speech-image cross-modal retrieval field, it is necessary to learn from current works in fine-grained caption-image retrieval, such as text-image cross-modal retrieval. Moreover, some text-image cross-modal retrieval works worked heavily on solving the problems which are the large variance of different modality types of input and the difficulty of measuring the distance between the multi-modal features. The above works will be useful to help my work. Three text-image cross-modal retrieval models [41, 54, 55] are chosen to be comparison methods to be compared with my model in chapter 5.3, since all of the three works trained their models on fine-grained datasets, such as CUB-200 [48], Oxford-102 [35], etc. Table 2.2 summarizes the three text-to-image models' networks, loss functions and datasets that used to train their models. Their text encoder will not be introduced in this thesis.

Table 2.2: Networks, loss functions and datasets used to train the models in different comparison works.

Method	Image encoder	Loss function	Dataset
Zhen et al.[55]	VGG-19	model proposed loss	Wikipedia [37]
Zhang and Lu[54]	MobileNet [16]	softmax loss	CUB-200 [48] & Oxford-102 [35]
Sarafianos et al.[41]	ResNet152	KL divergence loss + matching loss + adversarial loss	CUB-200 & Oxford-102

Deep Supervised Cross-modal Retrieval [55]. Zhen et al. [55] presented a novel cross-modal retrieval method, called Deep Supervised Cross-modal Retrieval (DSCMR) that aims to find a common representation space, in which the samples from different modalities can be compared directly. Specifically, DSCMR minimizes the discrimination loss in both the label space and the common representation space to supervise the model learning discriminative features. The entire model includes two sub-networks, one for the image encoding and another for text encoding. The convolutional layers of the deep neural network for image modality are the VGG-19. They generated 4, 096-dimensional feature vector from $fc7$ layer (the seventh layer which is a fully connected layer) as the original high-level semantic representation for image encoding. To ensure that the two sub networks learn the common representation space in the form of images and text, they force the two to share the weights of their last layer. Intuitively, this can generate as similar representations as possible for images and text samples from the same category. Finally, a linear classifier with the parameter matrix is connected to these two sub-networks to learn discriminative features by exploiting the label information. The objective function contains three parts: the discrimination loss in the label space, the discrimination loss of all samples from both modalities and modality invariance loss. The discrimination loss in the label space is defined as:

$$\mathscr{J}_1 = \frac{1}{n}\left\|\mathbf{P}^T\mathbf{U}-\mathbf{Y}\right\|_F + \frac{1}{n}\left\|\mathbf{P}^T\mathbf{V}-\mathbf{Y}\right\|_F \tag{2.14}$$

where $\|\mathring{\mathrm{u}}\|_F$ denotes the Frobenius norm, $\mathbf{P}$ is the projection matrix of the linear classifier. The discrimination loss of all samples from both modalities in the common representation space directly is defined as:

$$\begin{aligned}\mathscr{J}_2 = &\underbrace{\frac{1}{n^2}\sum_{i,j=1}^{n}\left(\log\left(1+e^{\Gamma_{ij}}\right)-S_{ij}^{\alpha\beta}\Gamma_{ij}\right)}_{\text{inter-modalities}}\\ &+\underbrace{\frac{1}{n^2}\sum_{i,j=1}^{n}\left(\log\left(1+e^{\Phi_{ij}}\right)-S_{ij}^{\alpha\alpha}\Phi_{ij}\right)}_{\text{image modality}}\\ &+\underbrace{\frac{1}{n^2}\sum_{i,j=1}^{n}\log\left(1+e^{\Theta_{ij}}\right)-S_{ij}^{\beta\beta}\Theta_{ij}\Big)}_{\text{text modality}}\end{aligned} \tag{2.15}$$

To eliminate the cross-modal discrepancy, the modality invariance loss was proposed to minimize the

distance between the representations of all image-text pairs. Technically, it was formulated as follow:

$$\mathcal{J}_3 = \frac{1}{n}\|\mathbf{U}-\mathbf{V}\|_F \tag{2.16}$$

The overall loss function of the method DSCMR is:

$$\mathcal{J} = \mathcal{J}_1 + \lambda\mathcal{J}_2 + \eta\mathcal{J}_3 \tag{2.17}$$

where the hyper-parameters λ and η control the contributions of the last two components, and n is the number of the input instances.

Overall, the model DSCMR learns the common representations which are both discriminative and modality-invariant for cross-modal retrieval. The model achieved its goal by minimizing the discrimination loss (in the common representation space and the label space) and the modality invariance loss simultaneously.

Deep Cross-Modal Projection Learning for Image-Text Matching [54] proposed a cross-modal projection matching (CMPM) loss and a cross-modal projection classification (CMPC) loss for learning discriminative image-text embeddings. The image-text matching architecture consists of three components: a visual CNN to extract image features, a bi-directional LSTM (Bi-LSTM) to encode text features, and a joint learning module for associating the cross-modal representations. For an image, the authors employed MobileNet [16] to extract its initial feature from the last pooling layer. Its novel image-text matching loss termed Cross-Modal Projection Matching (CMPM), incorporates the cross-modal projection into Kullback–Leibler divergence to associate the representations across different modalities.

The matching loss from image to text in a mini-batch is computed by:

$$\mathcal{L}_{i2t} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} p_{i,j}\log\frac{p_{i,j}}{q_{i,j}+\epsilon} \tag{2.18}$$

Where $p_{i,j}$ is the probability of matching x_i to z_j, $q_{i,j}$ is the true matching probability of (x_i, z_j) after normalization, since there might be more than one matched text samples for x_i in a mini-batch. Thus, the bi-directional CMPM loss is calculated by the sum of the text-to-image loss and image-to-text loss. For image-text matching with identity-level annotations, the classification loss applied to each modality helps to learn more discriminative features. However, the matching relationship of image-text pairs may not be fully utilized in a single classification task. The author has developed a novel classification function in which cross-modal projection is integrated into the norm-softmax loss to further enhance the compactness of matching embedding. The re-formulated norm-softmax loss is defined as:

$$\begin{aligned}&\mathcal{L}_{ipt} = \frac{1}{N}\sum_{i} -\log\left(\frac{\exp\left(\boldsymbol{W}_{y_i}^{\top}\hat{\boldsymbol{x}}_i\right)}{\sum_j \exp\left(\boldsymbol{W}_j^{\top}\hat{\boldsymbol{x}}_i\right)}\right)\\ &\text{s.t.}\left\|\boldsymbol{W}_j\right\| = r, \hat{\boldsymbol{x}}_i = \boldsymbol{x}_i^{\top}\overline{\boldsymbol{z}}_i\cdot\overline{\boldsymbol{z}}_i\end{aligned} \tag{2.19}$$

where $\hat{\boldsymbol{x}}_i$ denotes the vector projection of image feature x_i onto normalized text feature $\overline{\boldsymbol{z}}_i$. The final CMPC loss is also the sum of the bidirectional loss.

Adversarial Representation Learning for Text-to-Image Matching [41]. Sarafianos et al. [41] introduced a Text-Image Modality Adversarial Matching approach that learned modality-invariant feature representations using adversarial and cross-modal matching objectives. The TIMAM model consists of three parts: the feature extraction module which extracts textual and visual features using their corresponding DNN models, the identification and cross-modal projection losses that match the feature distributions originating from the same identity, and an adversarial discriminator that pushes the model to learn modality-invariant representations for effective text-image matching.

For the visual representations, ResNet-152 is used as its backbone network. The loss function consists of three parts. There are two loss functions for identification and cross-modal matching. The identification loss is a norm-softmax cross entropy loss. The cross-modal projection matching loss which incorporates the cross-modal projection into the KL divergence measure to associate the representations across different modalities is used to address the problem of no association between the representations of the two modalities. Then, an adversarial loss to train the adversarial neural network is used, which

is used to reduce the modality gap. The authors trained a discriminator by loss function $\mathscr{L}_D$ and a generator $\mathscr{L}_G$. The discriminator is optimized by Equation 2.20:

$$\mathscr{L}_D = -\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{y_i \sim y}\left[\log D\left(\boldsymbol{y}_i\right)\right] + \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim \boldsymbol{x}}\left[\log\left(1 - D\left(\boldsymbol{x}_i\right)\right)\right]\right) \tag{2.20}$$

Thus, the overall loss function is the sum of above three loss functions.

3

Datasets and Preparation

This thesis is targeting the previously untouched challenge of fine-grained visually grounded speech representation learning. Therefore, it is necessary to have a detailed look at available datasets for the task. In the following, the datasets used for the experiments will be introduced (see Section 3.1). Additionally, the processing of raw data will be introduced in section 3.2.

3.1. Datasets

The datasets used in this thesis contain two types: fine-grained dataset and scene dataset. The fine-grained datasets are used to train the visually grounded fine-grained speech representation model that can learn the relationships between attribute and objects. The image scene dataset is used to train a scene-based visually grounded speech representation learning model to see the capability of the proposed model and to directly compare with other models in speech-image cross-modal retrieval field. A summary of the number of classes, images, and captions of each dataset can be seen in Table 3.1. In the table, the first column shows the three datasets used in this work. The first two datasets are fine-grained datasets, while Flickr8k is a scene image dataset. The second column is divided into three columns which represents the number of classes, the number of images and the number of spoken captions in the training set. The third and fourth column represents test set statistics and validation set statistics respectively. There is no overlap in classes between the training set, test set, and validation set. Please note that while the fine-grained datasets have classes (i.e., different flower and bird species, respectively), Flickr8k does not have classes. The last column summarizes the total number of classes, images, captions of the three datasets. A comprehensive introduction to the three datasets will be given in the following sections.

Dataset	Training set			Test set			Validation set			Overall		
	class	images	captions	class	images	captions	class	images	captions	class	images	captions
Flower (Oxford102)	82	7034	70340	20	1155	11550	-	-	-	102	8189	81890
Bird (CUB200)	150	8855	88550	50	2933	29330	-	-	-	200	11788	117880
Flickr8k	-	6000	30000	-	1000	5000	-	1000	5000	-	8000	40000

Table 3.1: Overview over the datasets

3.1.1. Fine-grained datasets.

Fine-grained datasets contain images and language descriptions. The dataset is divided into different classes and each class represents one species, i.e., flower species and bird species. The difference between classes in subtle, so-called fine-grained. In this thesis, two fine-grained datasets were used for model learning which are 102 Category Flower Dataset (Oxford-102) [35] and Caltech-UCSD Birds-200-2011 (CUB-200) [48]. The two datasets are commonly used in the fine-grained image learning [50]. The language descriptions of the two datasets were collected by [39], which were text descriptions. In order to fit into the visually grounded speech representation learning task, spoken descriptions were synthesized by Text-to-Speech (TTS) model [44]. A comprehensive introduction to the two datasets and the spoken description synthesis will be given in the following.

102 Category Flower (Oxford-102) Dataset.

Figure 3.1: The example images of each class in Oxford-102 dataset

Figure 3.1 shows an example image of each class in the Oxford-102 (flower) dataset. The flowers in this database are all commonly occurring flowers in the United Kingdom. Each class consists of between 40 and 258 images. Figure 3.2 shows the distribution of the number of images over all 102 classes. Passion flower has the greatest number of images and eustoma, mexican aster have the least, which is 40 per class. Each image has 10 language descriptions.

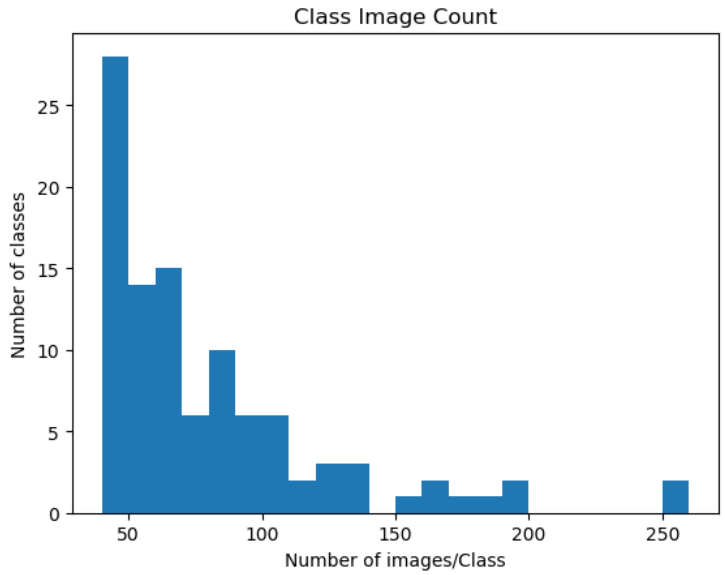


Figure 3.2: The distribution of the number of images over the 102 classes.

Caltech-UCSD Birds-200-2011 (CUB-200) Dataset.
Caltech-UCSD Birds-200-2011 (CUB-200-2011) is an image dataset with photos of 200 bird species (mostly North American). Figure 3.3 shows examples of the 200 bird species. Each class has between 40 and 60 images. The distribution of number of images per class is shown in Figure 3.4. Similarly, each image is also described by ten different text descriptions.

Spoken descriptions of fine-grained datasets. The fine-grained datasets only contain text descriptions. In order to obtain spoken captions, Tacotron 2 [44] is used in this thesis to synthesize spoken description.

Tacotron 2 is a neural network architecture for the speech synthesis, which synthesizes speech directly from text. Figure 3.5 shows the internal architecture of tacotron 2. The system is composed of a recurrent sequence-to-sequence feature prediction network that maps character embeddings to mel-scale

Figure 3.3: The example images in 200 classes of the CUB-200 (Birds) dataset

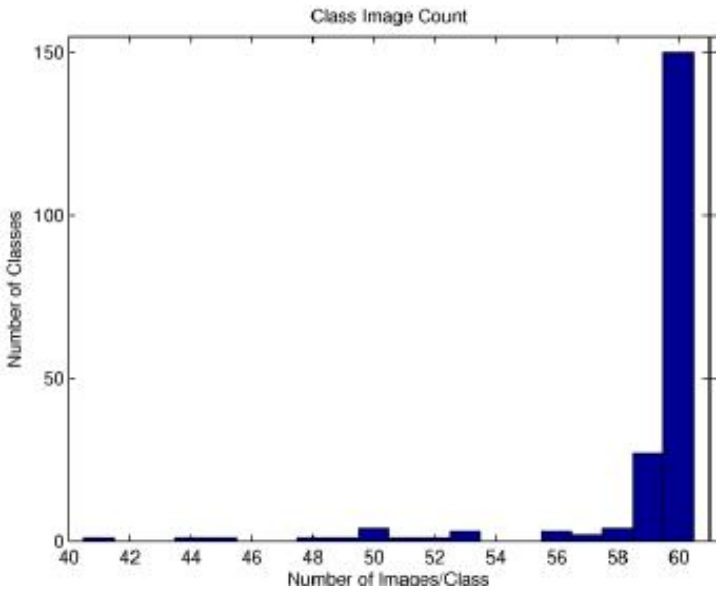


Figure 3.4: The distribution of the number of images per class in CUB-200 (Birds) dataset

spectrograms, followed by a modified WaveNet model acting as a vocoder to synthesize time-domain waveforms from those spectrograms [44]. Mel spectrograms are computed through a short-time Fourier transform (STFT) using a 50 ms frame size, 12.5 ms frame hop, and a Hann window function. The Mean Opinion Score (MOS) of synthesized speech reaches 4.53 comparable to a MOS of 4.58 for professionally recorded speech [44]. All text descriptions of Oxford-102 flowers dataset and CUB-200 birds dataset are synthesized into speech.

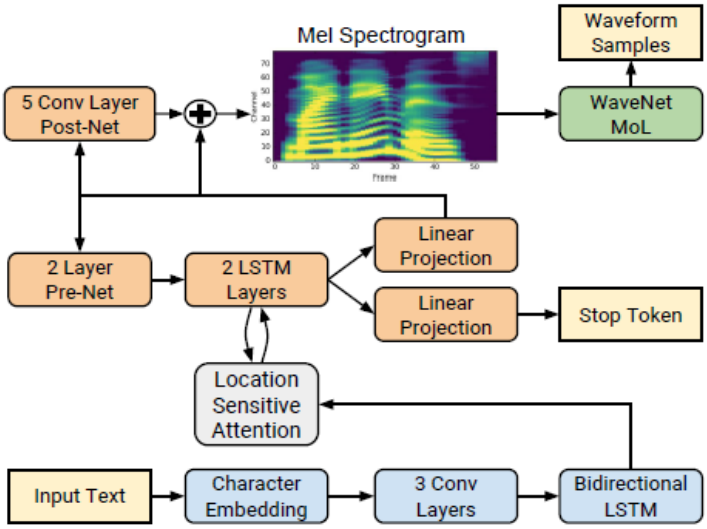


Figure 3.5: Block diagram of the Tacotron 2 system architecture [44].

3.1.2. Scene-based dataset

In this thesis, the scene-based dataset: Flickr8k audio caption corpus [10] is adopted. One reason for using Flickr8k is that the speech descriptions in the fine-grained datasets are synthesized while Flickr8k has real speech. Moreover, this dataset is common in speech-image cross-modal retrieval field, which can make my model directly compare with the related models in this field. Thus, my model was also trained on Flickr8k dataset to test the performance.

[15] presented Flickr8k dataset with captions. Each image in the dataset is associated with five different captions that describe the entities and events depicted in the image that were collected via crowdsourcing (Amazon Mechanical Turk) [15]. Figure 1.1 (see Chapter 1) showed two example images and corresponding five captions selecting from Flickr8k dataset. However, the captions presented by [15] were text descriptions. To make the Flickr8k dataset apply to speech-image retrieval task, [10] presented Flickr 8k Audio Caption Corpus which contains audio captions. It was collected in 2015 to investigate multimodal learning schemes for unsupervised speech pattern discovery [10].

3.2. Image and Speech data pre-processing

The images and spoken captions are pre-processed before training.

3.2.1. Image pre-processing

To extract image features, all images were resized such that the smallest side is 256 pixels while keeping the aspect ratio intact. I took random crop to fix the image size to 256 × 256 where random horizontal flip was applied to change some images, for the purpose of the data enhancement. Specifically, before the random crop, the images in CUB-200 dataset applied bounding boxes, which provided by [48]. The purpose is to locate the approximate position of the bird in the image. A simple cropping diagram has shown in Figure 3.6. Finally, images were rescaled to 244 × 244 to fit the input size of ResNet101.

3.2.2. Speech pre-processing

The input of the speech encoder consists of log Mel filter bank spectrograms. It is computed by 40 Mel-spaced filter banks with 25 ms Hamming window and 10 ms shift.

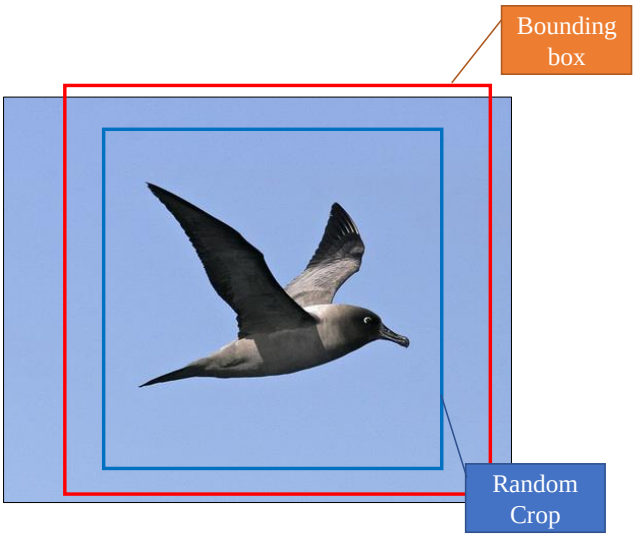


Figure 3.6: An example image processing from Bird dataset.

4

Approach

The following sections provide a detailed explanation of the proposed model, including the architecture and loss function.

4.1. Model architecture

In line with the two-branch architecture used in previous speech-image cross-modal retrieval models [11][12][31][54], my model also uses a dual-encoder structure consisting of a speech encoder and an image encoder. The two encoders are then mapped into a common feature space without any further guidance from the outside. The overall architecture of my visually grounded fine-grained speech embedding learning model is shown in Figure 4.1.

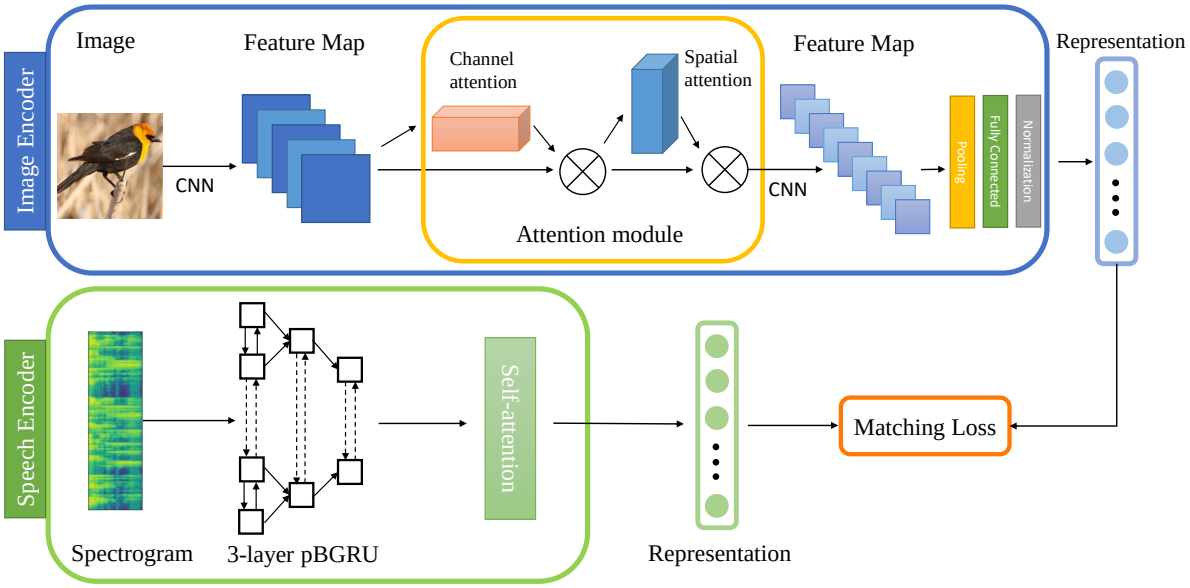


Figure 4.1: The structure of the VFSEL (visually grounded fine-grained speech embedding learning) model. The model consists of two parts: the image encoder and the speech encoder. The resulting representation of two parts will match together to calculate the matching loss.

The input of the image encoder (see top half of Figure 4.1) is a batch of processed images of the same size. The pre-trained ResNet-101 model is adopted to extract the image features. An image attention module is used inside the ResNet-101. Before generating the final representation of images, pooling layer, fully connected layer and normalization layer are applied. The input of the speech encoder (see

bottom half of Figure 4.1) is a batch of processed Mel Filterbanks with the same size. Then, a three-layer pyramidal Bi-GRU followed by a self-attention module is applied to represent speech. The image and speech representations are then used to compute the proposed batch loss to optimize the model. A comprehensive introduction to the two encoders will be presented in the following sections.

4.1.1. Image encoder

The top half of Figure 4.2 shows the internal architecture of the image encoder. The bottom left of the figure shows the structure of Bottleneck layer consisting several layers. The bottom right of the figure shows the structure of the image attention module which is taken from [49]. The output size of each block is shown on the right of each block in Figure 4.2. Take the first output size as an example, 122 × 122 × 64 denotes a feature map of 64 channels of convolution size of 122 × 122. The number shown on the Bottleneck block represents the number of similar Bottleneck blocks. The overall image encoder adopted the structure of ResNet-101 [13] pretrained on ImageNet [40] to extract visual features of the pre-processed images but made some changes to adapt the visually grounded speech embedding learning task. The changes to the original ResNet-101 model is marked by red dotted frame in the figure. The original architecture of ResNet-101 is shown in Figure 2.3 (see Section 2.3). The ResNet-101 model is designed for image classification task. It contains a 1,000 dimension fully connected (FC) layer and a softmax layer that outputs a probability distribution of classes. The two layers are the last two layers in ResNet-101. However, my task do not need the last softmax layer, so the softmax layer is dropped. Moreover, the dimension of fully connected layer is set to 2,048 instead of 1,000 in my image encoder, since after applying FC, the image encoder will output a 1 × 2,048 dimensional embedding vector for each image to be consistent with the output size of my speech encoder. Besides, a batch normalization layer is added between the average pooling layer and the FC to reduce the covariance shift. The ResNet-101 generally consists of four bottleneck blocks. The bottleneck block contains convolutional layers, batch normalization layers and RELU layers as shown in the bottom left of the Figure 4.2. Those bottleneck blocks are used to learn image features and for dimension reduction. It reduces convolution size and increase the number of channels. Besides, an attention layer is added between the third bottleneck and the fourth bottleneck. The attention mechanism is used to better learn the visual features that pay attention to important regions of images. The detailed introduction to the attention module is given in the following.

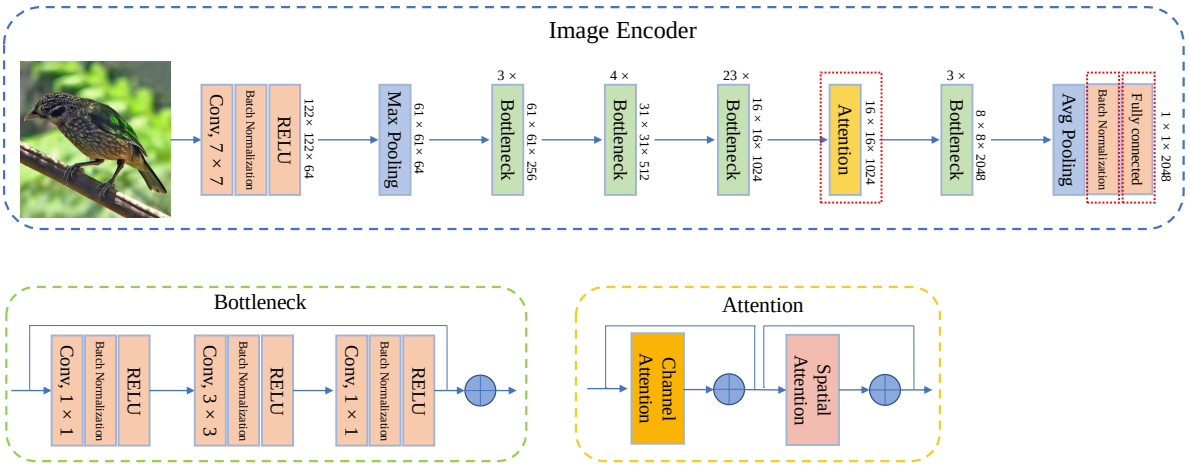


Figure 4.2: The internal structure of the image encoder.

Image attention module.

The Image attention module used in this model was taken from Convolutional Block Attention Module (CBAM) [49]. The goal of the attention module is to increase image representation power by focusing on important visual features and suppressing unnecessary ones. Since the fine-grained image representation need to focus more on discriminative features to recognize objects, the attention mechanism would help with the feature extraction.

The image attention module explored both spatial and channel-wise attention based on an efficient

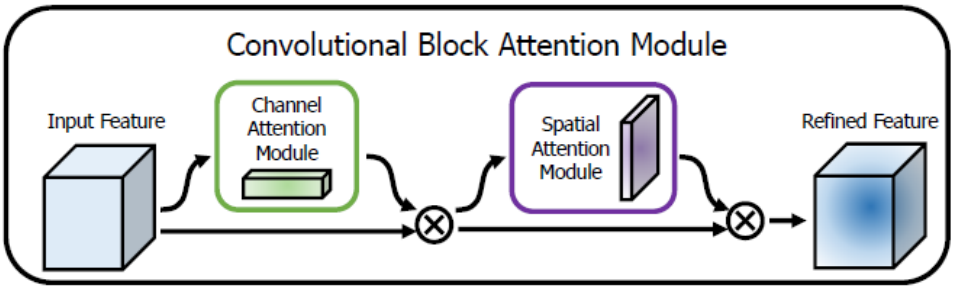


Figure 4.3: The overall Image attention module [49]

architecture [49]. Given an intermediate feature map $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$ taken from the previous convolutional layer as the input feature of the attention module, the module sequentially infers attention maps along two separate dimensions, a 1D channel attention map $\mathbf{M_c} \in \mathbb{R}^{C \times 1 \times 1}$ and a 2D spatial attention map $\mathbf{M_s} \in \mathbb{R}^{1 \times H \times W}$. The input feature $\mathbf{F}$ multiplied with two attention maps successively, as illustrated in Figure 4.3. The process of the overall attention mechanism can be summarized as:

$$\begin{aligned} \mathbf{F}' &= \mathbf{M_c}(\mathbf{F}) \otimes \mathbf{F} \\ \mathbf{F}'' &= \mathbf{M_s}(\mathbf{F}') \otimes \mathbf{F}' \end{aligned} \tag{4.1}$$

Where $\otimes$ represents element-wise multiplication. During the multiplication process, the input feature $\mathbf{F}$ multiplied with the channel attention matrix $\mathbf{M_c}(\mathbf{F})$ to get an intermediate output $\mathbf{F}'$. Then the intermediate feature map $\mathbf{F}'$ multiplied with the spatial attention matrix $\mathbf{M_s}(\mathbf{F}')$ to obtain the final refined output $\mathbf{F}''$. The Figure 4.4 shows two computation processes to get each attention weight matrix. The top half of Figure 4.4 shows the internal architecture of channel attention module which has been shown in Figure 4.3. The bottom half of Figure 4.4 shows the internal structure of spatial attention module that has been shown in Figure 4.3.

Channel attention module [49] was used to discover the inter-channel relationship of features. A color image usually contains three channels: red, green and blue. A color image can be represented as a matrix of dimensions $w \times h \times c$, where w is the width of the image, h is its height and c is the number of channels. In general, the number of channels will increase, and the size of the image will decrease by applying convolutional networks. A convolutional-net layer usually consists of multiple channels (typically hundreds of channels). Each channel describes different aspects of the previous layer. The channel attention module focuses on 'what' is meaningful during training according to the different filters in CNNs. The top half of Figure 4.4 describe the computation process to get the channel attention map $\mathbf{M_c}$. In the channel attention module frame, max pooling and average pooling are adopted simultaneously to address different aspects. Average pooling was commonly used for aggregating spatial information in CNNs and max-pooling gathered another important clue about distinctive object features to infer finer channel-wise attention [49]. The overall computation process can be described as follow:

First, the spatial information of the feature map was aggregated by using the average-pooling and max-pooling operations to generate two different spatial descriptors: $\mathbf{F^c_{avg}}$ and $\mathbf{F^c_{max}}$ [49]. Both descriptors were then forwarded to a shared network (Shared MLP in the Figure 4.4) to produce the channel attention map $\mathbf{M_c} \in \mathbb{R}^{C \times 1 \times 1}$. The shared network consists of a multi-layer perceptron (MLP) with a hidden layer. The hidden activation size is set to $\mathbb{R}^{C/r \times 1 \times 1}$, here r represents the reduction ratio. After applying the shared network to each descriptor, element-wise summation to merge the output feature vectors is used. In short, the formula for calculating channel attention is:

$$\begin{aligned} \mathbf{M_c}(\mathbf{F}) &= \sigma(MLP(\text{AvgPool}(\mathbf{F})) + MLP(\text{MaxPool}(\mathbf{F}))) \\ &= \sigma\left(\mathbf{W_1}\left(\mathbf{W_0}\left(\mathbf{F^c_{\text{avg}}}\right)\right) + \mathbf{W_1}\left(\mathbf{W_0}\left(\mathbf{F^c_{\text{max}}}\right)\right)\right) \end{aligned} \tag{4.2}$$

Where $AvgPool$ and $MaxPool$ denote the average-pooling and maximum-pooling operation, σ denotes the sigmoid function, $\mathbf{W_0} \in \mathbb{R}^{C/r \times C}$, and $\mathbf{W_1} \in \mathbb{R}^{C \times C/r}$. Note that the MLP weights, W_0 and W_1 are shared for both inputs of $\mathbf{F}_{\text{avg}}$ and $\mathbf{F}_{\text{max}}$, and the ReLU activation function is followed by W_0.

Spatial attention module [49] was utilized to explore the inter-spatial relationship of features. Unlike channel attention, the spatial attention focuses on 'where' is an informative part, which is complementary to the channel attention. Similar to channel attention, in the bottom half of Figure 4.4,

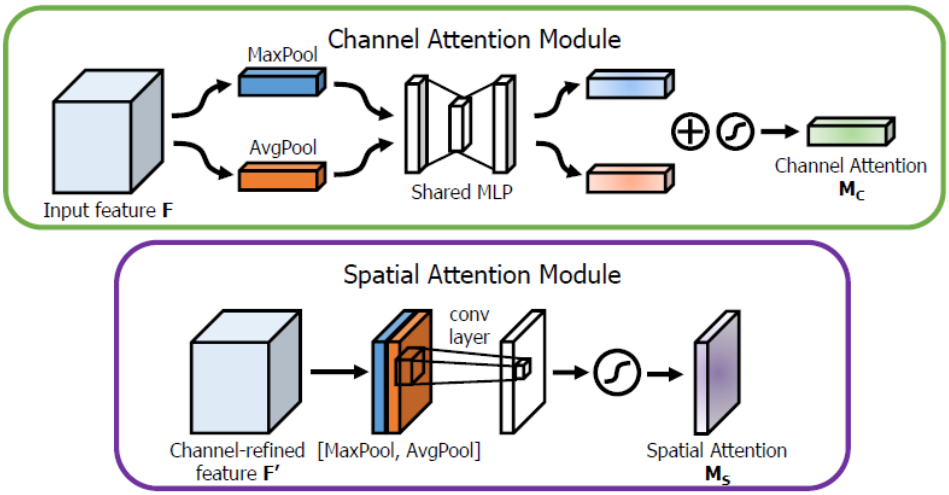


Figure 4.4: The internal structure of the channel attention module and the spatial attention module. [49]

average-pooling and max-pooling operations are firstly used along the channel axis and then concatenated to generate an efficient feature descriptor. The pooling operation along the channel axis is shown to be effective in highlighting informative regions [49]. After concatenating the two feature descriptors $\mathbf{F}^{\mathbf{s}}_{\mathbf{avg}}$ and $\mathbf{F}^{\mathbf{s}}_{\max}$, a convolution layer to generate a spatial attention map $\mathrm{M_s(F)} \in \mathrm{R}^{H \times W}$ is applied to encode where to emphasize or suppress. The detailed operation is described as follow:

The channel information of a feature map is aggregated by using two pooling operations, generating two 2D maps: $\mathbf{F}^{\mathbf{s}}_{\mathbf{avg}} \in \mathbb{R}^{1 \times H \times W}$ and $\mathbf{F}^{\mathbf{s}}_{\max} \in \mathbb{R}^{1 \times H \times W}$. Each represents the average pool feature and maximum pool feature in the channel. Then connect them through a standard convolution layer and convolve them to generate a 2D space attention map. The spatial attention is computed as follow:

$$\begin{aligned} \mathbf{M_s}(\mathbf{F}) &= \sigma\left(f^{7\times7}([\mathrm{AvgPool}(\mathbf{F}); \mathrm{MaxPool}(\mathbf{F})])\right) \\ &= \sigma\left(f^{7\times7}\left(\left[\mathbf{F}^{\mathbf{s}}_{\mathbf{avg}}; \mathbf{F}^{\mathbf{s}}_{\max}\right]\right)\right) \end{aligned} \tag{4.3}$$

Where $f^{7\times7}$ represents a convolutional layer with a filter size of 7×7.

4.1.2. Speech encoder

The speech encoder is modeled as a pyramidal Bidirectional Gated Recurrent Unit [4] (pBGRU) followed by a self-attention module, as illustrated in Figure 4.2. In the figure, the speech encoder inputs a sequence of $[x_1, x_2, ..., x_T]$ on the left of the Figure to each recurrent unit and outputs a sequence of $[h_1, h_2, ..., h_v]$ processed by a 3-layer pBGRU. Learned from [42] to encode speech data, 3-layer pyramidal bi-GRU is chosen since it can reduce the training time and computing resources. The model is adapted in this thesis by choosing GRU instead of LSTM because GRU performs better in this case. Finally, a self-attention module is used (see the green frame in Figure 4.5). The self-attention module is taken from [31], because it shows a good performance and better than other self-attention modules. The self-attention module calculates an attention weight matrix $\mathbf{M}_{att}$ which then multiplies with the hidden states $[h_1, h_2, ..., h_v]$ to get a sequence of weighted hidden states. The yellow frame with "Mat Mul" inside means matrix product. Finally, all the hidden states are summed over to get the final output. The output size of the speech encoder is [B × E], where B is the batch size and E represents the embedding size. The detailed introduction to the 3-layer pBGRU and the self-attention module will be given.

Three-layer pyramidal Bidirectional GRU module. The 3-layer pBGRU is a bi-GRU with three hidden layers, in which the input to each layer is the concatenation of two consecutive units from the left layer (see 3-layer pBGRU in Figure 4.5). The three-layer pBGRU is first introduced in [2]. The module utilizes a Bi-directional Gated Recurrent Unit (BGRU) with a pyramidal structure. The pyramidal bidirectional GRU which is an alternative bidirectional GRU that reduces the time dimension. It reduces the length V of h, from T of the input x. In each successive stacked pBGRU layer, the time resolution was reduced by a factor of 2. In a standard deep GRU architecture, the

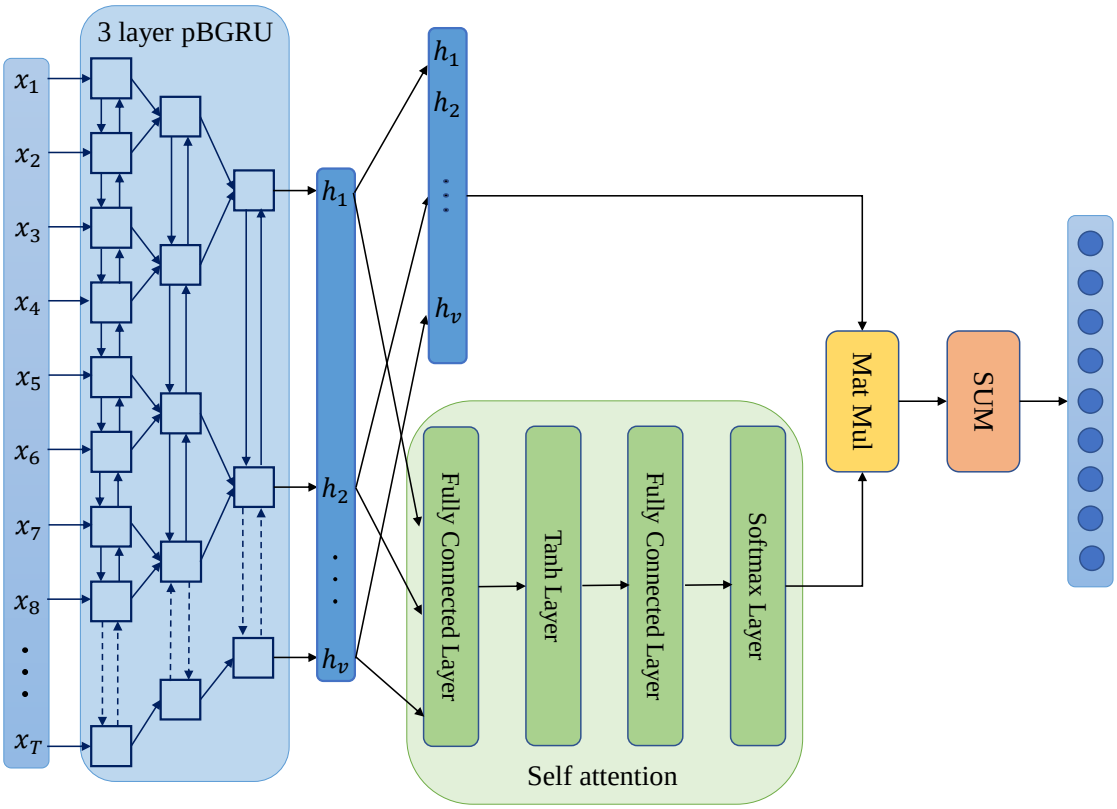


Figure 4.5: The structure of the speech encoder

output at the i-th time step, from the j-th layer is computed as follow:

$$h_i^j = \text{BGRU}\left(h_{i-1}^j, h_i^{j-1}\right) \tag{4.4}$$

In the pBGRU model, the outputs are concatenated at consecutive steps of each layer before feeding it to the next layer:

$$h_i^j = \text{pBGRU}\left(h_{i-1}^j, \left[h_{2i}^{j-1}, h_{2i+1}^{j-1}\right]\right) \tag{4.5}$$

In this model, the 3-layer pBGRUs is used to reduce the time resolution $2^3 = 8$ times and make the attention model to extract the relevant information from a smaller number of times steps.

Self-attention module. The self-attention mechanism is applied to focus on meaningful positions in a sequence of speech data. Even though Bi-GRU could alleviate the problem of long-term memory of RNNs, they may lose some connections when the sequence is too long. The self-attention will be easier to capture the long-distance interdependent features.

Given a speech representation $\mathbf{h} = (h_1, \ldots, h_t)$, the self-attention layer computes a weighted sum over all the hidden GRU states:

$$\mathbf{a}_t = \text{softmax}\left(V \tanh\left(W\mathbf{h}_t + \mathbf{b}_w\right) + \mathbf{b}_v\right) \tag{4.6}$$

$$\text{Att}\left(\mathbf{h}_1, \ldots, \mathbf{h}_t\right) = \sum_t \mathbf{a}_t \circ \mathbf{h}_t \tag{4.7}$$

Where $\mathbf{a}_t$ is the attention vector for hidden state $\mathbf{h}_t$ and W, V, $\mathbf{b}_w$ and $\mathbf{b}_v$ denotes the weights and biases. The applied attention is then the sum over the Hadamard product between all hidden states $(\mathbf{h}_1, \ldots, \mathbf{h}_t)$ and their attention vector. I use 1024 units for W and 2048 units for V.

4.2. Loss Function

After the representation vectors of the spoken caption and the image are obtained, the loss that aims to make the matched speech-image pairs closer in the embedding space is calculated. The loss used in

my model is a matching loss which contains two parts. The first part is defined as the negative log posterior probability that the images are matched with their corresponding speech descriptions and the second part is also the negative log posterior probability that the speech descriptions are matched with their corresponding images. This loss is taken from a Deep Attentional Multimodal Similarity Model (DAMSM) loss in [51]. And it is used because it performs better than triplet loss which will be discussed in chapter 5.

Thus, given a batch of speech-image representation pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_i^n$, batch size of n, the overall matching loss function is defined as:

$$\mathscr{L}_m = -\sum_{i=1}^{n} \log P\left(\boldsymbol{x}_i|\boldsymbol{y}_i\right) - \sum_{i=1}^{n} \log P\left(\boldsymbol{y}_i|\boldsymbol{x}_i\right) \tag{4.8}$$

where $P\left(\boldsymbol{x}_i|\boldsymbol{y}_i\right)$ the posterior probability of $\boldsymbol{y}_i$ matching with $\boldsymbol{x}_i$, and vice versa. The $P\left(\boldsymbol{x}_i|\boldsymbol{y}_i\right)$ is defined as:

$$P\left(\boldsymbol{x}_i|\boldsymbol{y}_i\right) = \frac{\exp\left(\gamma S\left(\boldsymbol{y}_i, \boldsymbol{x}_i\right)\right)}{\sum_{j=1}^{n} M_{i,j} \exp\left(\gamma S\left(\boldsymbol{y}_i, \boldsymbol{x}_j\right)\right)} \tag{4.9}$$

where γ is a smoothing factor determined by experiments with a value of 13. $S\left(\boldsymbol{y}_i, \boldsymbol{x}_i\right)$ is the cosine similarity between $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$. In this batch of pairs, only $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ are matching pairs and treat other $n-1$ of $\boldsymbol{x}$ as mismatching targets. Specifically, to deactivate the effects of other pairs from the same class, a mask $M_{i,j}$ is applied, where:

$$M_{ij} = \begin{cases} 0, & \text{if } y_i \text{ matches } x_j \& i \neq j \\ 1, & \text{otherwise} \end{cases} \tag{4.10}$$

5

Experiments and Results

After having discussed the datasets and the architecture of my model, the specific implementation and the evaluation of the model are presented in this Chapter. The experiments presented here are designed to test the model performance and investigate the effectiveness of each module in the model. Moreover, some experiments are designed to tune hyper-parameters to get the best-performing model. In this chapter, the measures to evaluate the model will be introduced first in Section 5.1. The detailed settings of the model training will be introduced in Section 5.2. Then, the details of the experimental implementations are described in Section 5.3. The experimental results are provided in Section 5.4.

5.1. Evaluation metrics

As mentioned, the proposed visually grounded speech learning model will be trained on both fine-grained datasets and a scene image-based dataset. The two kinds of datasets are different in structure: the fine-grained datasets are divided into classes while the scene image-based dataset is not. Consequently, different metrics should be used for the evaluation of my model depending on the type of database. For the fine-grained datasets, "R(ank)@1" (see Section 5.1.1) and "mean Average Precision (mAP)@50" (see Section 5.1.2) are adopted, and for the scene image-based dataset, "R(ank)@k" (see Section 5.1.1) and "Med r" (see Section 5.1.3) are used. The above metrics are commonly used metrics in cross-modal retrieval.

5.1.1. R(ank)@k

Image-caption retrieval performance on fine-grained datasets is typically evaluated using R(ank)@1 [39, 54, 55]. Moreover, speech-image retrieval performance is also typically evaluated using R(ank) [31]. The most commonly used metrics are R@1, R@5 and R@10, in short R(ank)@k.

R@k is the percentage of items for which the correct image or caption was retrieved in the top k, as mentioned in [31]. R@k is then defined as:

$$R@k = \frac{|\{\text{ number of correct items (sucessfully retrieved in the top k)}\}|}{|\{\text{ number of queries }\}|}. \tag{5.1}$$

For example, in the case of speech-based image retrieval, an image is used as a query to retrieve the corresponding spoken captions. If the image has more than one spoken caption and at least one caption was retrieved in top k results, the item for this query should be correct. The final R@k is the percentage of correct items in all queries.

For the fine-grained image datasets, all images from the same class can be described by the same spoken caption, and one image can also be described by different spoken captions from the same class. The number of relevant items for a query is large and the probability of correct items will be high. Thus, for evaluation on the fine-grained datasets, only R@1 is adopted which indicates the percentage of the queries where the top-1 result is the ground truth, according to [54].

For the scene image dataset Flickr8k, each spoken caption only has one corresponding image. Moreover, each image has five spoken captions. According to the evaluation metrics in [31], R@1, R@5 and R@10 were used to evaluate their models. The larger R@k means better retrieval performance.

5.1.2. Mean average precision (mAP)

As mentioned before, in the fine-grained datasets, each class has more than one image, and each image contain multiple spoken captions. In the speech-based image retrieval, a query, i.e., a spoken caption, should retrieve all matching images. However, the evaluation metric of R@1 is hard to comprehensively reflect the retrieved results. It only cares about whether the top-1 retrieved result is correct. That is the R@1 does not evaluate the ranking information in all retrieved results. Thus, following [54, 55], the mean Average Precision (mAP) is also adopted to evaluate the retrieval performance on the fine-grained databases. The mAP is a commonly used evaluation metric in information retrieval field, and it is defined as

$$\text{mAP} = \frac{\sum_{q=1}^{Q} \text{AP(q)}}{Q} \tag{5.2}$$

where Q is the number of queries, AP is the Average Precision which is defined as

$$\text{AP} = \frac{\sum_{k=1}^{n} (P(k) \times \text{rel}(k))}{\text{number of relevant images}} \tag{5.3}$$

where n is the number of the retrieved images or spoken captions to be taken into account, $P(k)$ is the fraction of correct results in the top-k retrieved instances, and $rel(k)$ is an indicator that equals 1 if the retrieved instance at rank k is correct. Here, the n is set as 50, which means only the top-50 retrieved instances are taken into account, so the evaluation metric is denoted as mAP@50.

5.1.3. Med r

Metric R@k does not care about the ranking of instances within top k. To focus on the ranking of retrieved results and also to give a more direct comparison with the state-of-the-art methods on speech-image cross-modal retrieval, the "med r" is adopted for the retrieval performance on Flickr8k according to [31]. The metric med r is the median rank of the correct retrieved instances and lower value of med r means better performance on the retrieval task, because the correct results are at a higher position in the ranking.

5.2. Training settings

The proposed model in this thesis was implemented with PyTorch-1.4.0 [1] which is an open source machine learning library. In this section, a brief introduction to PyTorch and details of experimental settings will be presented.

5.2.1. Pytorch

As an increasingly popular deep learning framework, PyTorch has basically become the most commonly used framework in the field of deep learning [34]. PyTorch is a Torch-based Python open source machine learning library for applications such as natural language processing. It is mainly developed by Facebookd's artificial intelligence team. It not only can achieve powerful GPU acceleration, but also supports dynamic neural networks. PyTorch provides two advanced functions: Tensor calculation with powerful GPU acceleration (such as Numpy) and Deep neural network including automatic derivation system. In addition to Facebook, organizations such as Twitter, GMU, and Salesforce have adopted the PyTorch architecture. Comparing with other machine learning libraries, such as Tensorflow, Caffe, etc., the code of PyTorch is more concise and intuitive, the underlying code is easier to understand. My model uses PyTorch as the underlying framework to write and execute the following modules: model definition, data processing and loading, training model and testing model.

5.2.2. Implementation details

The neural networks (see section 4.1) are stopped training when the number of epochs is above 120. The networks are trained by matching loss (see section **??**) and are optimized with the ADAM optimizer [23]. In the ADAM optimizer, the learning rate is set to 1×10^{-4}, weight decay of 1×10^{-5}, betas of [0.95,0.999] which are coefficients used for computing running averages of the gradient and its square. The other parameters in ADAM are set to default values. The batch size is 64 and the embedding dimension for image encoder and speech encoder is set to 2048.

[1] https://pytorch.org/

5.3. Experiments

First, an experiment to study the effect of a hyperparameter on the model and find the appropriate value for it to get the best performing model was conducted (see Section 5.3.1). After obtaining the final model, to validate the effectiveness of the model on the task of visually grounded speech representation learning, a cross-modal retrieval task was first conducted on the scene image dataset Flickr8k (see Section 5.3.2 for more details on the experiment). This allows me to directly compare my model to the state-of-the-art models in the field of visually grounded speech representation learning. Then, crucially, the cross-modal retrieval experiments were run on the two fine-grained datasets, which allows me to investigate the feasibility of the proposed visually grounded speech representation learning method to learn the relationships between attributes and objects, and to combine different attributes to retrieve new objects (see Section 5.3.3). The fine-grained cross-modal retrieval experiments contain two parts: a quantitative experiment which evaluates the model on the entire test sets by two metrics: R@1 and mAP@50 and a qualitative experiment which drill down into the task to see what the model actually retrieves. Moreover, ablation studies on the fine-grained datasets are carried out to assess the impact of components in the model were presented (see Section 5.3.4). Finally, further research on the image attention mechanism was conducted, which also included two parts: a quantitative experiment to see the effect of the attention mechanism on the cross-modal retrieval performance and a qualitative experiment to investigate what the attention mechanism learned (see Section 5.3.5).

5.3.1. Parameter sensitivity analysis

The purpose of this experiment is to analyze the sensitivity of the hyper-parameter γ in Eq. 4.8 (see Section 4.2) and optimize it to get a better-performed model. This experiment consists of two parts: investigate the parameter sensitivity and tune its value to get the optimal performance. First, according to the original setting of γ which is 10 in [51], I expanded its value range from 1 to 100 and selected 1, 5, 10, 20, 50 and 100 as its value to see how this parameter affected the model performance. The model performance was tested on the fine-grained datasets and evaluated by R@1 which is easier to calculate compared to mAP@50. Then, the best performing value range was selected and further analyzed. I reduced the value interval to 1. The value of γ to make the model performed the best among all values was selected to be the final value of hyper-parameter γ.

5.3.2. Cross-modal retrieval on Flickr8k

In this experiment, the proposed model was evaluated on the scene image dataset Flickr8k, and compared to several recent, state-of-the-art models trained on Flickr8k [5, 31, 42]. The results of the recent, state-of-the-art models were directly obtained from the respective papers. The networks and loss functions of the three works as well as my model are shown in Table 5.1. Each row represents a method. Each column, except first column, shows one part of a model, i.e., image encoder, speech encoder, loss function. The specific introduction to the three models can be found in section 2.4.

Following [31], both speech-to-image retrieval, i.e., using spoken captions as queries to retrieve the corresponding image, and image-to-speech retrieval, i.e., using images as queries to retrieve the corresponding spoken captions, were conducted in this thesis. The retrieval performance was evaluated by R@k, (k=1, 5, 10) (see Section 5.1.1) and Med r (see Section 5.1.3). Higher R@k and lower Med r mean better retrieval results, indicating better performance on the visually grounded speech representation learning task.

Method	Image encoder	Speech encoder	Loss function
Chrupała et al.[5]	VGG16	CNN + Attention	bi-modal triplet loss
Scharenborg et al.[42]	VGG16	pBLSTM	bi-modal triplet loss
Merkx et al.[31]	ResNet152	CNN + Bi-GRU + Attention	bi-modal triplet loss

Table 5.1: Networks and loss functions in different comparison works.

5.3.3. Cross-modal retrieval on Fine-grained datasets

For the key experiments in this thesis, the proposed model was trained and tested on two fine-grained datasets: CUB-200 and Oxford-102 (see Section 3.1.1). For the fine-grained datasets, the training set and test set are class-disjoint, which means that the learned model should have the ability to tell the difference between attributes and objects, so that it can infer new instances by combining different attributes and objects learned from the training set. Thus, this experiment aims to show the ability of my speech-image cross-modal retrieval model to learn the relationships between attributes and objects.

The cross-modal retrieval experiments were divided into two experiments: a quantitative experiment and a qualitative experiment. For the quantitative experiment, the proposed model was tested on the test sets of two fine-grained datasets and evaluated by R@1 and mAP@50 (see Section 5.1.1 and 5.1.2). Moreover, five related models were reproduced and tested in this thesis to compare with my model. Please note, that none of these models were previously tested on the fine-grained datasets. Table 5.2 shows the five compared models and my model. Each row represents a model. Each column, except the first column, shows one part of a model, i.e., image encoder, speech encoder, loss function. The dotted line divides two kinds of models. The models above the dotted line were originally designed for image-speech cross-modal retrieval task while the models under the dotted line were originally image-text cross-modal retrieval models. In this work I focus on the speech encoder and the loss function, that is why for all models the same image encoder was used, i.e., ResNet101. For the speech-image cross-modal retrieval models in [11] and [31], their speech encoders were reproduced according to their original design (see Section 2.4). For the text-image cross-modal retrieval models in [55], [54] and [41], their text encoders were replaced by my speech encoder (see Section 4.1.2). All loss functions used their original designed losses. The five models were also trained on two fine-grained datasets: CUB-200 and Oxford-102 and tested on their test sets. Moreover, the five models were evaluated by R@1 and mAP@50 to be consistent with my model evaluation. Both speech-to-image retrieval and image-to-speech retrieval were conducted for those models.

Table 5.2: Networks and loss functions in different comparison works. The models above the dotted line originally were speech-image cross-modal retrieval models. The models below the dotted line originally were text-image cross-modal retrieval models.

Method	Image encoder	Speech encoder	Loss function
Harwath et al.[11]	ResNet101	CNN	bi-modal triplet loss
Merkx et al.[31]	ResNet101	CNN + Bi-GRU + Attention	bi-modal triplet loss
Zhen et al.[55]	ResNet101	3-pBGRU + attention	model proposed loss
Zhang and Lu[54]	ResNet101	3-pBGRU + attention	softmax loss
Sarafianos et al.[41]	ResNet101	3-pBGRU + attention	KL divergence loss + matching loss + adversarial loss
The proposed model	ResNet101+attention	3-pBGRU + attention	batch loss

For the qualitative experiment, examples of good results and bad results were shown intuitively in two figures. Four queries that showed good performance, two image queries and two speech queries, were randomly selected. Here, good performance means that there are at least four good results in the top 5 retrieved results. Moreover, four queries that showed bad performance, two image queries and two speech queries, were randomly selected. Bad performance is defined as at least three wrong results in the top 5 retrieved results. This experiment provided a qualitative analysis of my model by visually displaying the results retrieved by the model. The qualitative analysis includes an explanation what the model learned, what the model failed to learn and the reason for the failure.

5.3.4. Ablation studies

An ablation study was carried out in which certain parts of the networks (see section 4.1) were removed to gain a better understanding of the networks' behavior. In my model, the components to be evaluated contain the image attention module, the speech attention module and the loss function. To evaluate the effectiveness of each component, this experiment was designed by removing and replacing each component separately. Each model variant will be tested on two fine-grained datasets: CUB-200 and Oxford-102 and evaluated by two metrics: R@1 and mAP@50. There are four variants of my model. The specific networks and losses of the model variants are shown in Table 5.3.

Table 5.3: Networks and loss functions in different model variants. r/p denotes "replaced by". w/o denotes "without". $\mathscr{L}_t$ represents bi-modal triplet loss. att means all attention modules in the model, att-I means the attention module in the image encoder and att-S means the attention module in the speech encoder.

Model	Image encoder	Speech encoder	Loss function
r/p $\mathscr{L}_t$	ResNet101+attention	3-pBGRU+attention	bi-modal triplet loss
w/o att	ResNet101	3-pBGRU	batch loss
w/o att-I	ResNet101	3-pBGRU + attention	batch loss
w/o att-S	ResNet101+attention	3-pBGRU	batch loss
The proposed model	ResNet101+attention	3-pBGRU + attention	batch loss

5.3.5. Research on image attention module

Finally, two sub-experiments were designed to do further research on the image attention module. The experiments were conducted with a quantitative evaluation and a qualitative evaluation. For the quantitative experiment, different model variants that have different image attention placement in the image encoder were tested on the two fine-grained datasets and evaluated by two metrics. The image attention module was proposed by [49] which was called Convolutional Block Attention Module (CBAM). According to its name, the authors plugged the module at every convolutional block in ResNet50. However, it did not perform well in my image encoder, possibly due to the different blocks of ResNet50 and ResNet101. Thus, only their attention module was adopted, so the placement of the attention module in the image encoder should be investigated. According to the architecture of ResNet101, the networks contain four "bottleneck" blocks (see Section 4.1.1) and the attention module was considered to be placed behind the "bottleneck". However, the feature size of the first and second bottleneck is too large to apply to the attention module, which require high computational cost. Thus, the quantitative experiment that tested the performance of the three model variants which were the model without the attention module, the model containing the attention module behind the third "bottleneck" and the model containing the attention module behind the fourth "bottleneck" was performed.

For the qualitative analysis, Gradient-weighted Class Activation Mapping (Grad-CAM) [43] was adopted to intuitively show what the attention module learned from fine-grained images. Grad-CAM is a recently proposed visualization method, which uses the gradients to calculate the important spatial regions in the convolutional layer (see Section 2.3.6). Important regions of an image are visualized in high-resolution detail. The redder the area in the image, the greater the contribution to the final prediction result, while the bluer the area shows, the smaller the contribution to the result. In this experiment, the Grad-CAM was applied to three model variants (same as the models in quantitative experiments) using images from the two fine-grained datasets. The images were randomly selected from different classes of the datasets and include two bird images and two flower images. The qualitative analysis would be the comparison between the attention regions of different model variants and the effectiveness of the image attention module.

5.4. Results and discussions

5.4.1. Parameter sensitivity analysis

The sensitivity of hyper-parameter γ in the loss function was investigated to see how it affected the model performance. Two graphs depicting the performance of models with different γ values evaluated by R@1 are shown in Figure 5.1. In each graph, there are three curves. The red one represents the R@1 score of speech-to-image retrieval, the green one represents the R@1 score of image-to-speech retrieval and the blue one is the sum of both sides. Moreover, each graph shows six scores with γ values of 1, 5, 10, 20, 50 and 100. According to the two graphs in the figure, the R@1 curves show the trend from increasing to decreasing, and when the value range of γ is [10,20], the model performs best.

Then, following the experiment settings, the value interval was reduced to 1 in the range of [10,20] to find the best performing model. There were 11 models having different values of γ to be tested. Figure 5.2 shows two graphs of R@1 score curves testing on 11 models of γ value of 10 to 20. Similarly, there are three curves in both graphs. According to the two graphs, the R@1 scores of different models are close, but there is a peak point when the value of γ is 13. Thus, when the value of γ is set to 13,

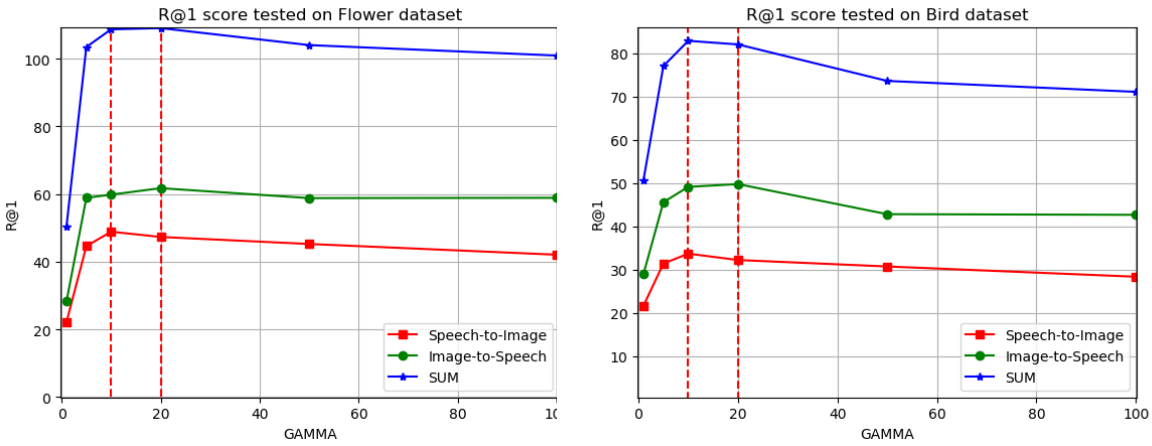


Figure 5.1: R@1 score in the value range of [1,100] of γ tested on Flower and Bird dataset

the best model performance is obtained.

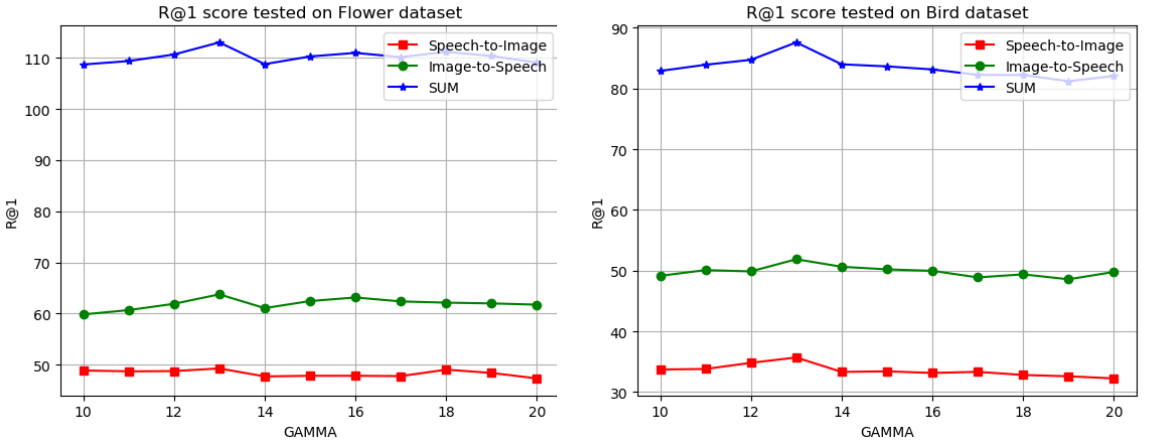


Figure 5.2: R@1 score in the value range of [10,20] of γ tested on Flower and Bird dataset

5.4.2. Cross-modal retrieval on Flickr8k

Cross-modal retrieval performance of the proposed method on Flickr8k and comparisons to other state-of-the-art works [5][42][31] are shown in Table 5.4. In the table, "Speech-to-Image" means speech-based image retrieval, i.e., using speech captions as queries to retrieve corresponding images, and vice versa. The retrieval performance was evaluated with R@k and Med r, and larger values of R@k and smaller values of med r mean better performance on the cross-modal retrieval task (see Section 5.1). Note that in the works of [5] and [42], only results on the task of "Speech-to-Image" are reported.

As shown in this table, the proposed method outperforms all other methods on both the "Speech-to-Image" and "Image-to-Speech" tasks on all evaluation metrics. Specifically, compared to the second-best method [31], the proposed method achieves 17.9% and 17.2% relative improvements, indicating the state-of-the-art performance of the proposed method on the visual-grounded speech representation learning task.

5.4.3. Cross-modal retrieval on Fine-grained datasets

The cross-modal retrieval results tested on two fine-grained datasets: CUB-200 and Oxford-102 of the proposed model are shown in this section. First, the quantitative experiment of the cross-modal retrieval evaluated by R@1 and mAP@50 is presented. Table 5.5 shows the retrieval performance evaluated using

Table 5.4: Cross-modal retrieval result on Flickr8k dataset. The best results are shown in bold.

Dataset	Flickr8k							
Method	Speech-to-Image				Image-to-Speech			
	R@1	R@5	R@10	med r	R@1	R@5	R@10	med r
Chrupała et al.[5]	5.5	16.3	25.3	48				
Scharenborg et al.[42]	7.3	21.8	32.1	—				
Merkx et al.[31]	8.4	25.7	37.6	21	12.2	31.9	45.2	13
The proposed model	**9.9**	**27.9**	**39.8**	**16**	**14.3**	**36.7**	**49.4**	**10**

R@1 and mAP@50 on the fine-grained test sets compared to the performance of five models (see Section 5.3.3). The best results are shown in bold.

According to table 5.5, the results of the proposed model are again much better than the results of previous four works. Overall, my model achieves state-of-the-art performance on visually-grounded speech representation learning and proves the ability to learn fine-grained visually grounded speech representation.

The results of the first two models [11][31] (see the first two rows above the dotted line in table 5.5) are significantly worse than the results of the other three models which were originally designed for text-image retrieval (see the three rows below the dotted line in table 5.5). According to the model structure of these five models that were reproduced in this thesis (see table 5.2), the first two models kept their speech encoders and loss functions, and the last three models only kept their loss functions when reproducing their models. Thus, their kept speech encoders and their loss functions (bi-modal triplet loss) did not perform well in this task. According to the results of three models that originally were text-image retrieval models, Sarafianos et al.[41] had the best results among the three models, which indicates that its loss function works the best on the model among three loss functions.

Table 5.5: Cross-modal retrieval results on CUB and Oxford datasets. The methods above the dotted line indicates they were originally designed for speech-image cross-modal retrieval task. The methods below the dotted line were originally designed for text-image cross-modal retrieval task. The best results are shown in bold.

Dataset	Oxford-102 (Flower)				CUB-200 (Bird)			
Method	Speech-to-Image		Image-to-Speech		Speech-to-Image		Image-to-Speech	
	R@1	mAP@50	R@1	mAP@50	R@1	mAP@50	R@1	mAP@50
Harwath et al.[11]	17.1	17.2	22.8	19.6	6.85	5.56	13.6	10.14
Merkx et al.[31]	10.6	10.1	17.9	14.6	9.56	8.38	12.29	10.25
Zhen et al.[55]	28.27	23.62	34.64	31.21	11.49	9.15	20.92	16.99
Zhang and Lu[54]	35.99	31.44	44.66	40.43	22.28	19.04	35.91	29.54
Sarafianos et al.[41]	37.06	32.61	47.35	41.43	27.01	22.77	43.00	35.52
The proposed model	**49.26**	**43.22**	**63.74**	**54.47**	**35.7**	**29.35**	**51.88**	41.52

Then, a qualitative experiment is presented to intuitively show the results retrieved by the proposed model (see Section 5.3.3). Figure 5.3 and 5.4 show the retrieval results which are good and bad respectively. In the two figures, the left-hand side shows the queries of images or speech descriptions, and the right-hand side shows the top five retrieved results. The symbol ✓ in green indicates that the retrieved result is correct, while the symbol × in red indicates the wrong result.

According to the figure of good performance, the good retrieval results (see Figure 5.3) show the ability of the proposed model to combine attributes and objects to infer new objects by learning fine-grained visually grounded speech representation.

According to the bad performance results, the proposed model retrieved some wrong results of the corresponding queries in the top 5 results obtained. There might be two reasons to explain the failure. The first one was the problem of my model. The proposed model missed key features to retrieve the correct result. For example, the first query of Figure 5.4 shows a pink flower with yellow stamen. The model erroneously retrieved a caption which refers to a star-shaped flower, while the query image is not star-shaped. The second reason might be the high similarities between different classes. The results retrieved by my model meet the requirements of the corresponding query, but due to the high degree of

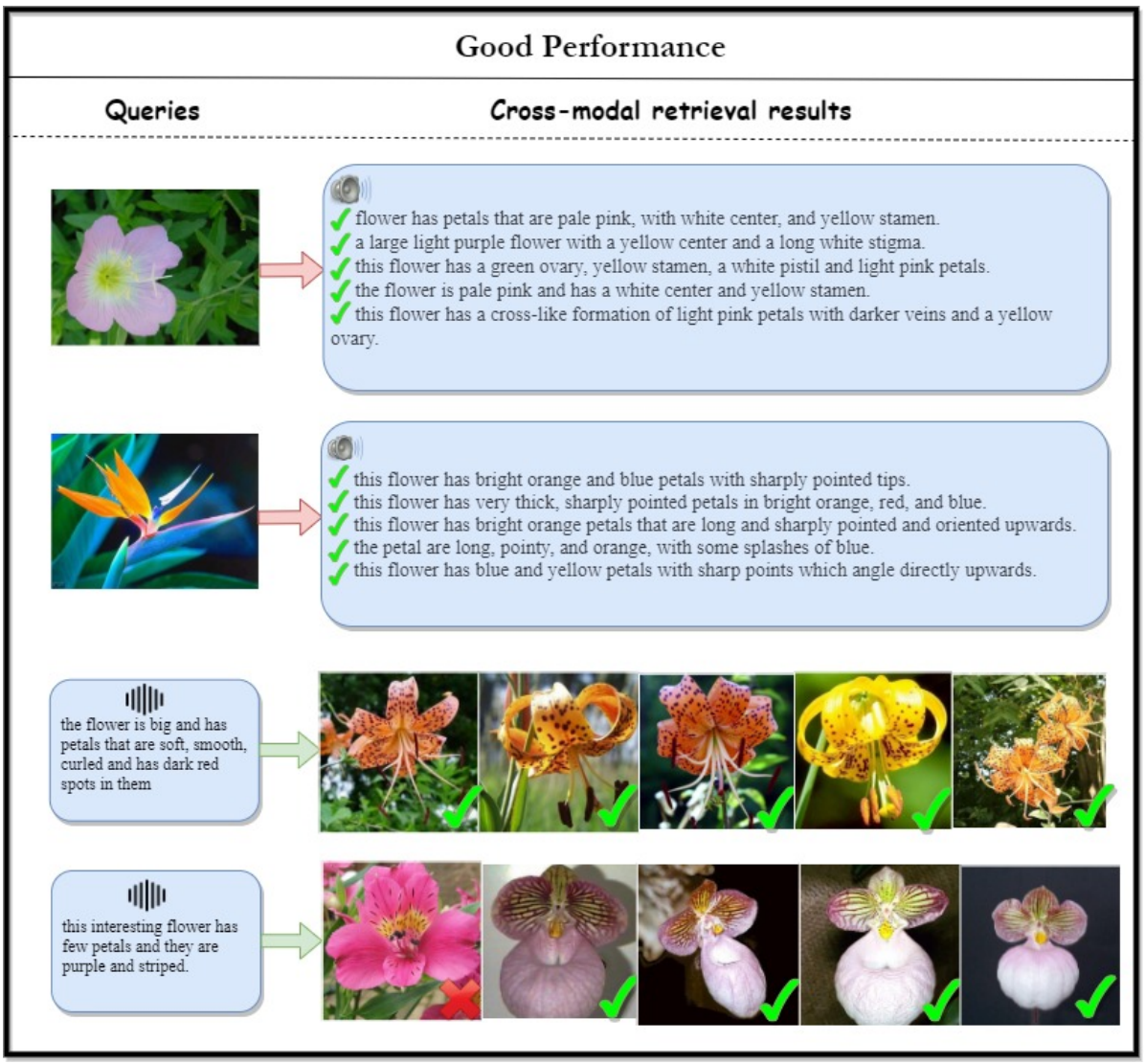


Figure 5.3: Qualitative results which have good performance. The four queries are randomly selected from Oxford-102 and CUB-200 test sets.

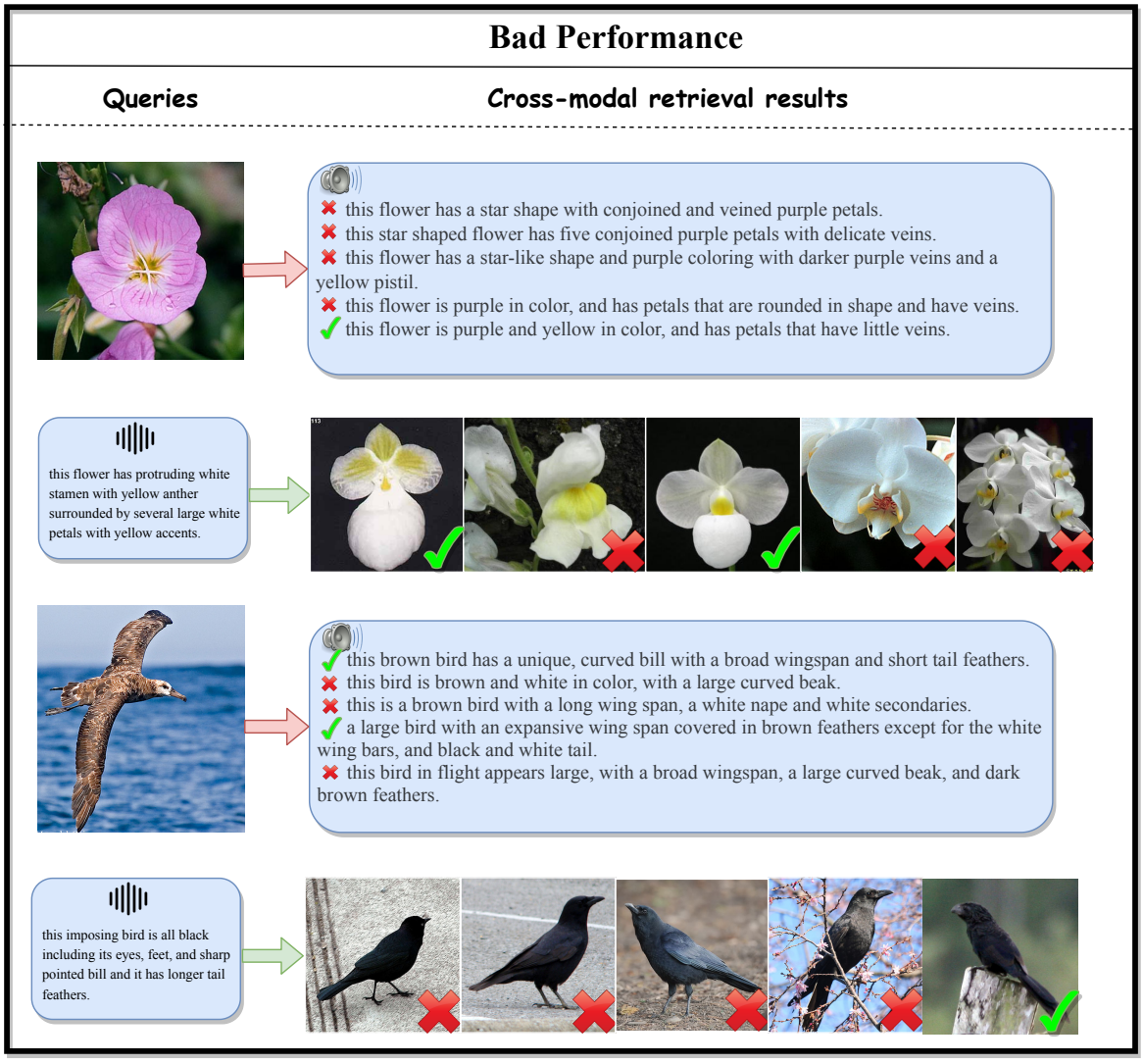


Figure 5.4: Qualitative results which have bad performance. The four queries are randomly selected from Oxford-102 and CUB-200 test sets.

similarity among categories, the model retrieved wrong results. One example is the fourth speech query in Figure 5.4 that describes "this imposing bird is all black including its eyes, feet, and sharp pointed bill and it has longer tail feather", which matches all its retrieved results. However, the black birds are really similar and hard to distinguish their difference. Thus, some wrong results are not caused by the proposed model. All in all, the above qualitative results show the ability of the proposed model to learn attribute information and the relationship between attributes and objects.

5.4.4. Ablation studies

The cross-modal retrieval results of the different variants of the model after removal of individual components are shown in Table 5.6. In the table, r/p means replaced by, $\mathscr{L}_t$ denotes bi-modal triplet loss, w/o means without, att means all attention modules in the model, att-I means the attention module in the image encoder and att-S means the attention module in the speech encoder. The best results are shown in bold.

The results show that the proposed model performs the best among other variants. This means that each component in the model is effective. Specifically, the results of the third row and the fourth row which indicate the model without the image attention module and the model without the speech attention module shows that the speech self-attention mechanism contributes more on the model performance than the image attention module. The triplet loss performs worse than the proposed batch loss.

Dataset	Oxford-102(Flower)				CUB-200(Bird)			
Method	Speech-to-Image		Image-to-Speech		Speech-to-Image		Image-to-Speech	
	R@1	mAP@50	R@1	mAP@50	R@1	mAP@50	R@1	mAP@50
r/p $\mathscr{L}_t$	23.12	20.6	30.41	24.48	18.33	16.38	22.31	19.76
w/o att	46.04	39.77	60.05	50.86	28.82	23.71	43.39	34.98
w/o att-I	48.55	41.95	61.35	53.36	31.79	25.86	46.58	37.68
w/o att-S	47.38	40.85	61.43	52.2	30.31	25.36	45.18	37.02
the proposed model	**49.26**	**43.22**	**63.74**	**54.47**	**35.7**	**29.35**	**51.88**	**41.52**

Table 5.6: Component analysis of the model. The best result is shown in bold.

5.4.5. Research on image attention module

Table 5.7 shows the quantitative results of three model variants of the location of the attention mechanism. In the table, "w/o" means without; "att-I" means the image attention module; "|" denotes behind; "B3" denotes the third "Bottleneck" block in ResNet101, i.e., "att-I|B3" represents that the image attention module is placed behind the third bottleneck.

Table 5.7: Cross-modal retrieval results between different attention placements testing on two fine-grained datasets. Best results showed in bold.

Dataset	Oxford-102(Flower)				CUB-200(Bird)			
Method	Speech-to-Image		Image-to-Speech		Speech-to-Image		Image-to-Speech	
	R@1	mAP@50	R@1	mAP@50	R@1	mAP@50	R@1	mAP@50
w/o att-I	48.55	41.95	61.35	53.36	31.79	25.86	46.58	37.68
att-I | B3	**49.26**	**43.22**	**63.74**	**54.47**	**35.7**	**29.35**	**51.88**	**41.52**
att-I | B4	47.79	41.9	59.43	51.18	33.89	27.8	49.18	39.99

The second model in the table is the proposed model in this thesis. Thus, the results show that the proposed model performs better than others. Specifically, for the Flower dataset, the model without image attention module performs even better than the model with the image attention which was placed behind the fourth "bottleneck" block.

Then, for the qualitative analysis, figure 5.5 shows the qualitative results (experiment settings see Section 5.3.5). In the figure, except for the first column and the column of the input image, each column represents a model using Grad-CAM to visualize the heatmap to an input image.

As mentioned, the more reddish regions the more contribution to the final prediction result. The Grad-CAM results show that all three variants pay attention to the objects. The difference is the model without the attention module has more attention regions than other two models, which means that the model learns more useless information, i.e., background information or common feature that all species have. Comparing the Grad-CAM on model "att-I|B3" and model "att-I|B4", it is hard to distinguish which model learns better, especially only picked four example images to evaluate. Overall, the qualitative results show the effectiveness of the image attention module in the proposed model.

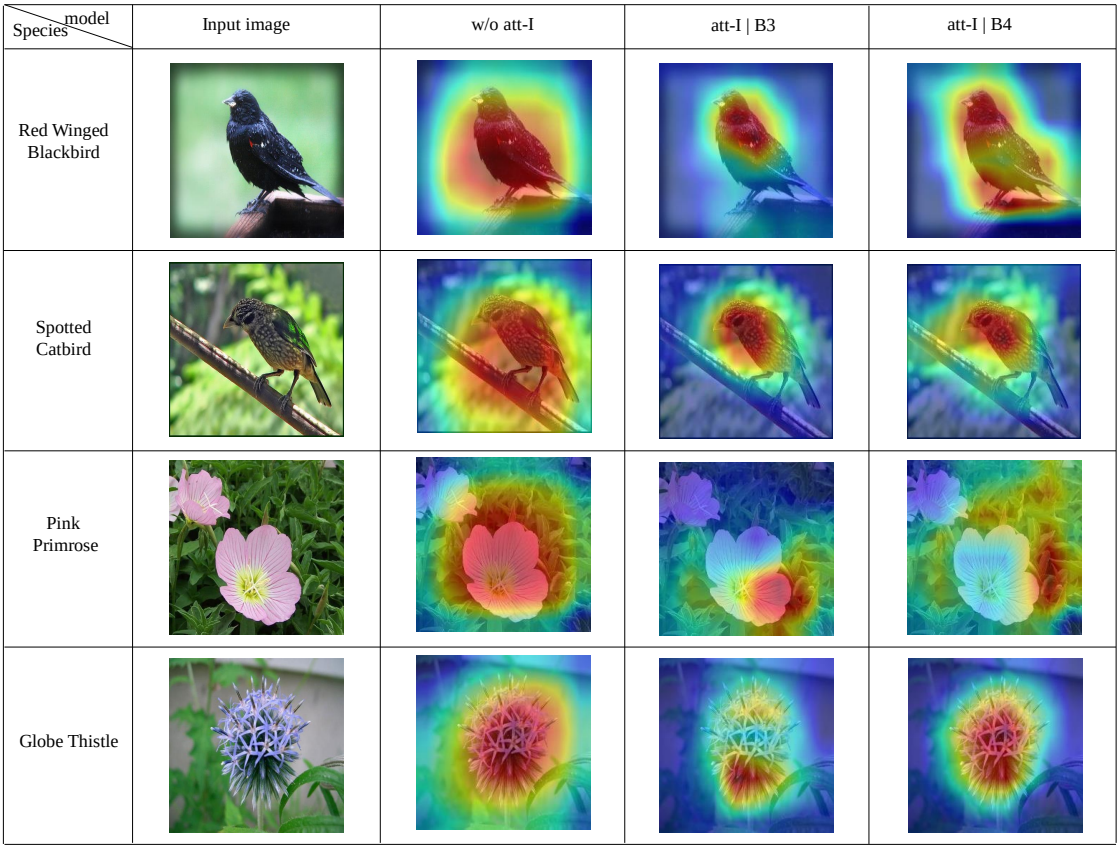


Figure 5.5: Grad-Cam [43] visualization results.

6

General discussion and conclusions

This chapter summarizes the outcomes of this thesis. In Section 6.1 the answers to the research questions proposed in Chapter 1.1 are given. Then, limitations of the used methodology are discussed in Section 6.2. Finally, future works to improve the existing works are discussed in the last Section 6.3.

6.1. Research questions

To conclude this thesis, the research questions proposed in Section 1.1 are answered. First, the sub-questions are considered, finally a general conclusion is given by answering the main research question.

What kind of tasks and datasets can be used to evaluate the ability of the learned speech embedding model on inferring new visual objects?

Visually grounded speech embedding learning was evaluated by cross-modal retrieval task in this thesis. The cross-modal retrieval task can establish the direct relationship between image and speech from paired image-audio data. Based on cross-modal retrieval task, the image and speech embeddings were learned jointly. Moreover, cross-modal retrieval which indicates using the instances of one modality to retrieve another's provides a way to evaluate the proposed model performance.

Fine-grained datasets contain hard-to-distinguish object classes and made the model learn more about the attribute information associated with objects. The model trained on fine-grained datasets learned different attribute information and was evaluated by cross-modal retrieval task to see whether it learned to combine attribute information to retrieve new objects. Thus, fine-grained datasets were adopted to train a visually grounded fine-grained speech representation learning model and to evaluate the model ability to infer new objects

What is the appropriate deep learning model structure for feature extraction of speech and images?

The proposed model was constructed by deep neural networks (see Section 4.1). Deep neural networks were chosen to encode the speech and images due to the well-developed techniques of DNNs. Moreover, most works in cross-modal speech-image retrieval field adopted DNNs. The proposed model was designed to be a dual-encoder architecture with an image encoder and a speech encoder. The image encoder consisted of a pre-trained ResNet and an image attention module to extract feature. The speech encoder applied a three-layer pyramidal bi-GRU followed by a self-attention module to learn speech embedding. Experiments conducted in section 5.4.2 and 5.4.3 showed that the proposed model achieved state-of-the-art results in cross-modal retrieval on both the scene dataset Flickr8k and the fine-grained datasets.

What is the appropriate loss function for the visually grounded speech representation learning model?

The adapted batch loss was adopted to optimize the proposed model (see section **??**). The proposed loss function calculated the posterior probability of each image-speech pair that was matched by calculating the similarity between a batch of image and speech data. Triplet loss is a commonly used loss function in speech-image cross-modal retrieval field. The results of the model applying batch loss were compared with the model using the triplet loss (see Section 5.4.4. The results showed that the proposed model using batch loss performed much better than using triplet loss.

Is the attention mechanism useful for visually grounded speech semantic learning?"

This question aims to evaluate the effectiveness of the attention mechanism. Both the image encoder and the speech encoder applied attention mechanisms that were the image attention module and the self-attention module respectively (see Section 4.1). To test the module's effectiveness, an ablation study was conducted (see Section 5.4.4) where each attention module was removed separately from the model. The results showed both image attention module and the self-attention module were effective in the proposed model. Moreover, Grad-Cam was used to visualize the important regions learned by the image attention module. The results showed the image attention module was able to learn meaningful and important regions in the image for the feature extraction.

Can the visually grounded speech representation learning model combine different attribute information to retrieve new visual objects?

In this thesis, a visually grounded fine-grained speech representation learning model was proposed to combine different attribute information associated with objects to infer new objects. The proposed model was tested on two fine-grained image datasets which contain many attribute-object pairs. The results of the quantitative experiment showed that the proposed model is able to retrieve new visual objects and outperforms other visually grounded speech learning models. The qualitative results and analysis also showed the proposed model's ability to learn the relationship between attributes and objects and infer new objects.

6.2. Discussion

In the following paragraphs the most important limitations of this study will be discussed. Firstly, the fine-grained datasets used in this thesis which are CUB-200 and Oxford-102 only contain text descriptions in their original datasets. The spoken descriptions for fine-grained datasets were synthesized by a single-speaker TTS model, and these synthesized speech utterances were used to train my model. This means that not only is the speech less variable than real speech due to it being synthesized, also the training and test spoken data came from the same one speaker. It is therefore easy to cause the model to overfit during training. Also, it makes this speech-based task to be simplified considerably which might not be convincing enough for the fine-grained image-speech retrieval model training. Thus, the multi-person speech data will be a solution to the problem.

Furthermore, the two attention mechanisms used in this thesis for speech and image only consider the acoustic and visual information individually within their local contexts. It may not sufficiently capture the correlations between modalities. Cross-modal attention mechanism, such as [52], could learn multi-modal interaction, which may improve the model performance in this thesis.

Lastly, the results of fine-grained image-speech cross-modal retrieval in this thesis still have room for improvement. In fact, comparing with the results of fine-grained image-text retrieval model, my model results are significantly lower than theirs. For example, the Text-Image Modality Adversarial Matching (TIMAM) model [41] achieved an R@1 score of 67.7% on the image-to-text retrieval and 70.6% on the text-to-image retrieval. The model was trained on CUB-200 and Oxford-102 datasets, which were the same datasets in this thesis. In fact, except for their text encoder, other components of their model, such as the image encoder, the loss function, have been applied to my model to see whether the model could be improved, but the performance did not improve. The large gap between their text-image retrieval model and my speech-image retrieval model might be the speech representation learning. Thus, the improvement on speech representation learning for this task should be considered.

6.3. Future works

According to the previous discussions for improvement, future research for this topic can be pursued in four directions.

Speech data collection. The fine-grained datasets require multi-speaker speech data to avoid overfitting. Spoken description data collected from multiple real speakers are considered to improve the capability of the model.

Qualitative analysis on self-attention module. Only the image attention module has been evaluated in a qualitative way to see what did the attention module learn. Visualize the self-attention module in the speech encoder is also important to learn its internal mechanism in speech representation learning. Drawing a heat map of the self-attention module can be a future work so that can qualitatively analyze what did the attention mechanism learn.

Cross-modal attention module. This technique could help to learn the relationships between two

modalities, i.e., speech and image. It learns correlations between multi-modal features which represent both visual and acoustic information might be more appropriate to cross-modal retrieval task.

Further research on speech representation learning. Speech representation learning was developed rapidly in the field of speech learning. The state-of-the-art techniques for speech representation can be adopted. For example, in [31], Multilingual Bottleneck (MBN) features for speech performed better than MFCCs, which can be used for speech encoding in my model.

Bibliography

[1] Lyle Campbell. Ethnologue: Languages of the world, 2008.

[2] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015.

[3] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3), 2010.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. Representations of language in a model of visually grounded speech signal. *arXiv preprint arXiv:1702.01991*, 2017.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] Mao Guo, Chenghu Zhou, and Jiahang Liu. Jointly learning of visual and auditory: A new approach for rs image and audio cross-modal retrieval. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[10] David Harwath and James Glass. Deep multimodal semantic embeddings for speech and images. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 237–244. IEEE, 2015.

[11] David Harwath, Antonio Torralba, and James Glass. Unsupervised learning of spoken language with visual context. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2016.

[12] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *Proceedings of the European conference on computer vision (ECCV)*, pages 649–665, 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[17] G Ian, B Yoshua, and C Aaron. Deep learning (adaptive computation and machine learning), 2016.

[18] Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[19] Herman Kamper, Shane Settle, Gregory Shakhnarovich, and Karen Livescu. Visually grounded learning of keyword prediction from untranscribed speech. *arXiv preprint arXiv:1703.08136*, 2017.

[20] Herman Kamper, Gregory Shakhnarovich, and Karen Livescu. Semantic speech retrieval with a visually grounded model of untranscribed speech. *IEEE ACM Trans. Audio Speech Lang. Process.*, 27(1):89–98, 2019.

[21] A Karpathy and L Fei-Fei. Deep visual-semantic alignments for generating image descriptions. department of computer science, 2017.

[22] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, pages 1–62, 2020.

[23] Diederik P Kingma and J Adam Ba. A method for stochastic optimization. arxiv 2014. *arXiv preprint arXiv:1412.6980*, 434, 2019.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[28] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[29] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[30] Danny Merkx and Stefan L Frank. Learning semantic sentence representations from visually grounded language without lexical knowledge. *Natural Language Engineering*, 25(4):451–466, 2019.

[31] Danny Merkx, Stefan L Frank, and Mirjam Ernestus. Language learning using speech to image retrieval. *arXiv preprint arXiv:1909.03795*, 2019.

[32] Yajie Miao, Mohammad Gowayyed, and Florian Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174. IEEE, 2015.

[33] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, 2011.

[34] Giang Nguyen, Stefan Dlugolinsky, Martin Bobák, Viet Tran, Álvaro López García, Ignacio Heredia, Peter Malík, and Ladislav Hluchỳ. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, 52(1):77–124, 2019.

[35] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[36] Yuxin Peng, Xin Huang, and Yunzhen Zhao. An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges. *IEEE Transactions on circuits and systems for video technology*, 28 (9):2372–2385, 2017.

[37] Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Nikhil Rasiwasia, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):521–535, 2013.

[38] Asya Pereltsvaig. *Languages of the world: An introduction.* Cambridge University Press, 2017.

[39] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[41] Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. Adversarial representation learning for text-to-image matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5814–5824, 2019.

[42] Odette Scharenborg, Laurent Besacier, Alan Black, Mark Hasegawa-Johnson, Florian Metze, Graham Neubig, Sebastian Stüker, Pierre Godard, Markus Müller, Lucas Ondel, et al. Speech technology for unwritten languages. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:964–975, 2020.

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[44] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arxiv 2014. *arXiv preprint arXiv:1409.1556*, 1409, 2014.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[47] W Wang et al. R-net: machine reading comprehension with self-matching networks. natural language computer group, microsoft reserach. asia, beijing. Technical report, China, Technical Report 5, 2017.

[48] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[49] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[50] Huapeng Xu, Guilin Qi, Jingjing Li, Meng Wang, Kang Xu, and Huan Gao. Fine-grained image classification by visual-semantic embedding. In *IJCAI*, pages 1043–1049, 2018.

[51] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[52] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10502–10511, 2019.

[53] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 187–196, 2014.

[54] Ying Zhang and Huchuan Lu. Deep cross-modal projection learning for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–701, 2018.

[55] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. Deep supervised cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10394–10403, 2019.

[56] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.