

A DISSERTATION  
SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN COMPUTER SCIENCE AND ENGINEERING

**Using a Resizable Hidden Module to Balance  
Performance and Cost in Image Captioning**



by

Yan Lyu

*March 2024*

© Copyright by Yan Lyu, March 2024

All Rights Reserved.

The thesis titled

*Using a Resizable Hidden Module to Balance Performance and Cost in Image Captioning*

by

Yan Lyu

is reviewed and approved by:

---

**Chief referee**

*Professor*

*Date*

Yong LIU

LIU Yong 

Feb. 21, 2024

---

*Professor*

*Date*

Qiangfu ZHAO

ZHAO Qiangfu 


Feb. 21, 2024

---

*Senior Associate Professor*

*Date*

Yoichi TOMIOKA

Yoichi Tomioka 

2024.02.21

---

*Associate Professor*

*Date*

Yan PEI

裴岩 

2024.02.21

THE UNIVERSITY OF AIZU

March 2024

# Contents

<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Image Captioning	1
1.1.1 Image Understanding	2
1.1.2 Natural Language Processing	5
1.2 Main Research Challenge in Image Captioning	6
1.3 Structure of Thesis	7
1.4 Motivations	8
1.5 Main Contributions	9
1.5.1 Chapter 2	10
1.5.2 Chapter 3	10
1.5.3 Chapter 4	11
1.5.4 Chapter 5	12
<b>Chapter 2 Literature Review and Preliminary Knowledge</b>	<b>14</b>
2.1 Literature Review	14
2.1.1 Multimodal Space-Based Methods	17
2.1.2 Visual Space-Based Methods	20
Supervised Learning	20
Unsupervised Learning	26
Reinforcement Learning	27
2.2 Preliminary Knowledge	29
2.2.1 Encoder–Decoder Architecture for Image Captioning	29
2.2.2 VGGNet	30
2.2.3 Darknet	32
2.2.4 Transformer	33
2.2.5 Vision Transformer	34
2.2.6 LSTM	34
<b>Chapter 3 End-to-end Image Captioning Based on Reduced Feature Maps of Deep Learners Pre-trained for Object Detection</b>	<b>36</b>
3.1 Introduction	37
3.2 Outline	39
3.3 Proposed Architecture	40
3.4 Design for Image Captioning	42
3.5 Dataset and Experiments Settings	46
3.6 Result Analysis with SOTA	49
3.7 Conclusion	50



<b>Chapter 4</b>	<b>Maintain a Better Balance Between Performance and Cost for</b>	
	<b>Image Captioning by a Size-Adjustable Convolutional Module</b>	<b>52</b>
4.1	Introduction	53
4.2	Outline	54
4.3	Proposed Network Architecture	55
4.4	Size-Adjustable Convolutional Module (SACM)	60
4.5	Dataset and Experiment Setting	66
4.6	Experimental Analysis	69
	4.6.1 Experimental Results on Flickr 8K	69
	4.6.2 Experimental Results on MS COCO	72
4.7	Result Comparison with SOTA	75
4.8	Qualitative Analysis	79
4.9	Conclusions and Future Works	79
<b>Chapter 5</b>	<b>Performance and Cost Balancing Image Captioning with Vision</b>	
	<b>Transformer</b>	<b>82</b>
5.1	Introduction	82
5.2	Outline	84
5.3	Proposed Method	85
5.4	Dataset and Experiment Settings	88
5.5	Empirical Analysis	90
5.6	Conclusion and Possible Future Work	94
<b>Chapter 6</b>	<b>Conclusion</b>	<b>96</b>

# List of Figures

Figure 1.1 Examples of a few images with sample captions from the Flickr 8K dataset [1]. . . . .	2
Figure 1.2 Image understanding . . . . .	3
Figure 1.3 Semantic Understanding . . . . .	6
Figure 1.4 Structure of Thesis . . . . .	7
Figure 2.1 An overall taxonomy of deep learning-based image captioning. . .	16
Figure 2.2 A summarized block diagram of the encoder-decoder and compositional architecture. . . . .	22
Figure 3.1 Some example images of Flickr8K dataset [1]. The reference captions are under the image. . . . .	38
Figure 3.2 Flowchart of the whole training process. The encoder extracts image features, while the decoder generates text descriptions by analysing the features. The estimated vector will be compared with the ground truth for loss measurement. . . . .	41
Figure 3.3 Encoder by VGG. Image is input into the model and feature-extracted by the VGG network without the last fully connected and softmax layers. The image is finetuned from [2]. . . . .	43
Figure 3.4 Encoder by Darknet-53. Image is input into the model and feature-extracted by the Darknet-53 with several convolutional layers. . . . .	44
Figure 3.5 Decoder by LSTM. With the threshold of vocab setting as 5, there are 2,550 words in the Flickr 8K dataset vocabulary. The feature maps and embedded caption will be input into the LSTM network during training. The <i>LSTM.out</i> means the output of the decoder part. . . . .	45
Figure 3.6 Some typical examples on the Flickr8k dataset. Humans generate the Ref sentences, and the other sentences are by the models of VGG and YOLO. The numbers after the predicted captions are the BLEU-4 scores. . . . .	49
Figure 4.1 Architecture of CNN plus RNN model. The CNN encoder extracts image features, while the RNN decoder generates text descriptions by analyzing the features. . . . .	56
Figure 4.2 Detailed architecture. The green dotted box includes the backbone, the following convolutional layers construct the SACM, and the blue dotted box contains the decoder. . . . .	62

Figure 4.3	The training process of the proposed model includes encoder, SACM, and decoder. The embedded target is input into the decoder only during training. This model is trained end-to-end. It means the optimizer can update all related parameters in the encoder, SACM, and decoder based on the loss function. . . . .	65
Figure 4.4	Loss values and four BLEU scores on training (left) and evaluation (right) By the end-to-end model with VGGNet as the encoder and LSTM as the decoder on the Flickr 8K dataset. The size of the final feature map is set as $S_f = 4096$ after removing the softmax and the last fully connected layer. Loss values and four BLEU scores on training (left) and evaluation (right) by Darknet-LSTM with $S_f = 128 \times 13 \times 13 = 21,632$ on the Flickr 8K dataset. . . . .	70
Figure 4.5	Loss values and four BLEU scores on training (left) and evaluation (right) by the end-to-end model with VGGNet as the encoder and LSTM as the decoder on the MS COCO dataset. The size of the final feature map is set as $S_f = 4096$ after removing the softmax and the last fully connected layer. Loss values and four BLEU scores on training (left) and evaluation (right) by Darknet-LSTM with $S_f = 128 \times 13 \times 13 = 21,632$ on the MS COCO dataset. . . . .	73
Figure 4.6	Visualization of our models with different encoders on validation images of Flickr 8K dataset. The wrong parts of a caption are marked. The GT caption is one of the five targets for evaluating the predicted sentence in the dataset. . . . .	80
Figure 5.1	Overview of the encoder-decoder model for image captioning. The encoder extracts image features, while the decoder generates text descriptions by analyzing the features. . . . .	83
Figure 5.2	Overview of ViT model [3]. The ViT model splits an image into fixed-size patches, linearly embeds each of them, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder. . . . .	86
Figure 5.3	Overview of the proposed model. To balance the performance and cost, we apply convolutional layers for feature map dimension reduction while maintaining the global information and the relationships between different regions. . . . .	87

# List of Tables

Table 2.1	Literature summary of image captioning. T-B, R-B, and E&D denote template-based, retrieval-based, and encoder-decoder methods.	25
Table 2.2	Comparisons among four CNN architectures [4]. #Multiply-adds and #Params denote the number of operations and the output of each neuron or node.	31
Table 3.1	Hyper Parameters Used in the Simulations	47
Table 3.2	BLEU Score and Testing Time by Different Models on Flickr8k Dataset (BLEU scores and testing time are all in average.)	50
Table 4.1	Settings of SACM with the original $128 \times 52 \times 52$ feature size. conv $k_x \times k_y \times c$ is a convolution of kernel size $k_x \times k_y$ with $c$ outputs channels. The last line is the final output $S_f$ size from SACM.	63
Table 4.2	The cost and performance of SACM with different feature sizes on the testing set for Flickr 8K. B@1 and B@4 denote BLEU-1 and BLEU-4 scores. “No.” denotes the number of learning epochs when the models received the best BLEU-4 score on the evaluation data.	71
Table 4.3	The cost and performance of SACM with different feature sizes on the testing set for MS COCO. B@1 and B@4 denote BLEU-1 and BLEU-4 scores. “No.” denotes the number of learning epochs when the models received the best BLEU-4 score on the evaluation data.	74
Table 4.4	Comparisons of our proposed Darknet-LSTM with SACM and some SOTA methods on Flickr 8K dataset. B@1, B@4, M, R, C, and P denote BLEU@1, BLEU@4, METEOR, ROUGE-L, CIDEr, and the model sizes.	76
Table 4.5	Comparisons among our proposed Darknet-LSTM with SACM and some SOTA methods on MS COCO dataset. B@1, B@4, M, R, C, and P denote BLEU@1, BLEU@4, METEOR, ROUGE-L, CIDEr, and the model sizes.	77
Table 5.1	Different Versions of ViT model and its Feature Size	85
Table 5.2	The cost and performance of our model with different versions of ViT as an encoder on the testing set for Flickr 8K. B@1 and B@4 denote BLEU-1 and BLEU-4 scores. “No.” denotes the number of learning epochs when the models received the best BLEU-4 score on the evaluation data.	91

Table 5.3	Number of operators of our models and predicting time on CPU and GPU machine . . . . .	92
Table 5.4	Comparisons of our proposed ViT-LSTM with convolutional feature reduction and some SOTA methods on Flickr 8K dataset. B@1, B@4, M, R, C, and P denote BLEU@1, BLEU@4, METEOR, ROUGE-L, CIDEr, and the model sizes. . . . .	93

[include-classes=abbrev,name=List of Abbreviations]

[include-classes=nomencl,name=List of Symbols]

y

# Acknowledgment

First, I would like to pay my great thankfulness and gratitude to everyone who gave me the strength and knowledge.

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Yong Liu, for his unwavering support, invaluable guidance, and continuous encouragement throughout my doctoral journey. Whenever I came up with a research idea, Prof. Liu assisted me in distilling its core and fundamental essence and formulating it more professionally and in detail. Whenever I completed a research experiment, Prof. Liu encouraged me to stay on track and improve based on our precious research. When I wrote a terrible draft, with utmost patience, Prof. Liu repeatedly revised and guided me until I had perfected it. Prof. Liu provided me with academic guidance and plenty of encouragement. Prof. Liu has transformed me from a confused, impatient researcher to a more consistent and focused researcher. I would not have been able to come this far without Prof. Liu's guidance and supervision.

I would like to thank Professor Qiangfu Zhao from the bottom of my heart, who has continuously provided me with constructive comments and advice even from my graduate study from the beginning of my PhD journey. I had chances to discuss our research ideas and experimental experiments in detail every two weeks. In every discussion, Prof. Zhao gives me supplementary advice and insightful observations from my empirical results. With Prof. Zhao's help, I have obtained in-depth knowledge about the fundamentals of computer vision and deep learning.

I am very grateful to Professor Yoichi Tomioka and Professor Yan Pei from the University of Aizu for their constructive critiques and valuable suggestions that have significantly enriched the quality of the final output of my research. I was fortunate to be able to invite Prof. Tomioka as an advisor in my research progress report. Through the progress report, Prof. Tomioka gave me several pieces of advice that helped me understand the practicality and scope of my research.

Second, I would like to extend my heartfelt appreciation to Professor Zixue Cheng from the University of Aizu for his unwavering assistance and support throughout my master's program and the initial two years of my doctoral journey at the University of Aizu. His guidance has illuminated my path, and their insights have been instrumental in helping me navigate the intricate landscape of sleep research, experimental setup, and methodologies. In every discussion or weekly meeting, Professor Zixue Cheng consistently presents captivating topics and offers perceptive viewpoints, motivating me to strive for continuous improvement.

I was privileged to work with and learn from several brilliant minds throughout my PhD journey. I would like to thank everyone in my lab. They have generously supported me in academics and also in day-to-day life. I wish them the best of luck and pray for their future endeavour.

I would like to thank my parents who always supported and encouraged me. It is

tough to help them while staying far from my family. But their unconditional love and prayer have given me the strength to work hard and make them proud.

Yan Lyu,  
April 2023,  
Aizuwakamatsu, Japan



# Abstract

Generating a description of an image is called image captioning. Image captioning is challenging because it involves understanding the main objects, their attributes and their relationships in an image. It also generates syntactically and semantically meaningful descriptions of the images in natural language. A typical image captioning pipeline comprises an image encoder and a language decoder. Convolutional neural networks (CNNs) are widely used as encoders, while long short-term memory (LSTM) networks are used as decoders. Various LSTMs and CNNs are used to generate meaningful and accurate captions. Traditional image captioning techniques have limitations in generating semantically meaningful and superior captions. With the appearance of transformer-based techniques, more and more large-scale models received better performance with higher computational resource requirements.

In this research, we focus on keeping the balance between performance and cost of image captioning, which can meet the needs of real-world mobile applications. This dissertation summarises our research in image captioning and proposes novel encoder-decoder models and a size-adjustable convolutional module (SACM) for feature dimension reduction for real-time image captioning.

In Chapter 3, we first introduced a novel end-to-end image captioning model architecture that combines a Darknet-based feature extractor with an LSTM-based caption generator. Unlike existing models that rely on pre-trained CNNs as intermediaries, the end-to-end image captioning model utilizes carefully designed feature extractors and caption generators to enhance caption quality. Our model allows a direct path from raw images to generated captions, simplifying the process. Empirical research supports the model's outstanding performance, and its low parameter requirements and efficiency make it well-suited for various practical applications. This chapter proved the effectiveness of convolutional-layer feature dimension reduction in image captioning.

Continuously, Chapter 4 focuses on striking a harmonious balance between performance and computational cost based on the Darknet-based model. We propose incorporating a size-adjustable convolutional module (SACM) as an intermediary step before decoding these features into coherent sentences. SACM is a size-adjustable convolutional module that consists of several convolutional layers for feature extraction and a few additional convolutional layers for dimension reduction. Increasing and decreasing the convolutional layers for dimension reduction can maintain the balance of performance and cost. The SACM can reduce feature dimension directly for sending to LSTM for caption generation. The SACM performs as a pipeline connecting the encoder and decoder to reduce feature dimensions, saving time and computational costs. After passing through SACM, the dimension-reduced feature maps go through LSTM to generate captions for the provided images.

The experimental results demonstrate the effectiveness of our approach. With the appropriately configured SACM, our model achieves remarkable performance on stan-

dard image captioning benchmarks. Leveraging a pre-trained object detection model and the size-adjustable convolutional module, our method demonstrates outstanding results on benchmark datasets while reducing the computational overhead substantially compared to existing approaches.

Lastly, in Chapter 5, we introduced a novel image captioning model that combines a vision transformer and LSTM, emphasizing its unique approach and real-time applicability while providing insights into its performance compared to established vision models. This approach represents an innovative departure from existing methods, which typically involve encoding and decoding stages. Importantly, this research marks the first utilization of the vision transformer method in image caption generation. Furthermore, we applied the convolutional layer for feature dimension reduction to emphasize the practicality of real-time image caption generation.

# Chapter 1

## Introduction

### 1.1 Image Captioning

Images are familiar in our daily lives, whether on the internet, in the news, or advertisements. Unlike pictures in articles and TV programs, most photos don't come with captions. While many people can easily understand images without captions, visually impaired individuals may need assistance comprehending them. Machine learning tools can help solve this issue by automatically interpreting photos, videos, and other media.

However, machines must first understand its semantics and context to provide a textual description of an image. For a long time, the goal of artificial intelligence (AI) has been to enable machines to see, understand, and describe the images around us [5]. Social media platforms like Facebook and Twitter can generate image descriptions that provide details such as the location, attire, and activities of the individuals in the image [6].

Image captioning is a complex problem in Artificial Intelligence that connects computer vision with natural language processing. It involves the understanding and description of the image's semantics, including the main objects, their attributes, poses, and interactions. Moreover, it requires inferring the underlying semantic meanings to generate captions [7, 8].

In the public image captioning dataset, Flickr 8K [1], several pictures with captions



(a) A small, pale bird bends down to examine a crumb.



(b) Three women dressed up in green and shamrocks.



(c) A man in red swim trunks jumps onto a bodyboard.

Figure 1.1: Examples of a few images with sample captions from the Flickr 8K dataset [1].

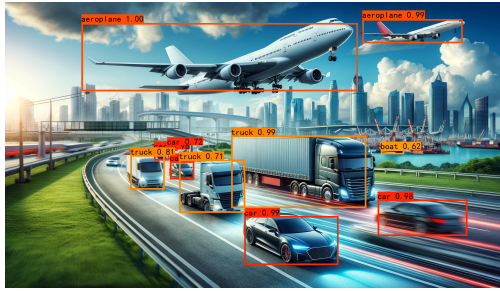
are provided, as shown in Figure 1.1. "A small, pale bird bends down to examine a crumb.", "Three women dressed up in green and shamrocks." and "A man in red swim trunks jumps onto a bodyboard." correspond to the images in Figure 1.1a, 1.1b and 1.1c, respectively. Image captioning can be used for various applications, including human-robot interaction, text-based image retrieval, and other similar tasks.

Image captioning is a crucial area of research involving automatically generating captions for images. This process requires both image understanding and language description generation. Image understanding is a fundamental problem of Computer Vision (CV), while language description generation is a part of Natural Language Processing (NLP) [9]. A typical image captioning framework comprises an image encoder that learns features from the image and a language decoder that generates captions for the image.

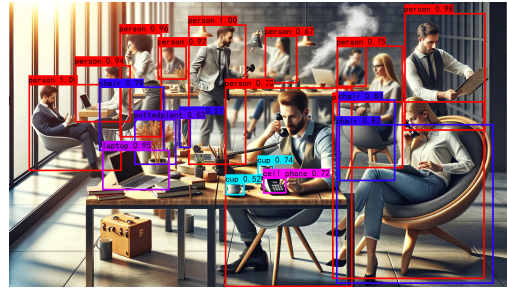
### 1.1.1 Image Understanding

Computer Vision (CV) refers to the ability of machines to perceive and comprehend items in an image. It encompasses various techniques to extract the necessary information from images. Much research is conducted in CV research, particularly in visual recognition and understanding. Visual recognition involves the identification, localization, and classification of objects present in an image.

Visual understanding requires object recognition and extracting the complete detail of the individual object and its associated relationship. The image captioning model



(a) An image of multiple objects: Car, Trunk and Aeroplane.



(b) An image of different items in the office: Person, Book, Cup, Cellphone, Chair, and Keyboard.

Figure 1.2: Image understanding

should correctly recognize multiple objects and their position relationship. Figure [1.2] shows a few examples of image understanding. Figure [1.2a] has three main entities: car, trunk and aeroplane, and Figure [1.2b] contains different items in the office or cafe, including person, book, cup, cellphone, chair and keyboard.

Feature extraction is a crucial aspect in object detection tasks. An object can possess multiple features rather than just one attribute. For instance, popular choices include colours, contour lines, and geometric lines or edges (gradient of pixel intensities) [10].

There are two types of feature-extracting methods: hand-crafted and learned methods. Hand-crafted methods include Local Binary Pattern (LBP) [11], Histogram of Oriented Gradients (HOG) [12], scale-invariant Feature Transform (SIFT) [13], and a combination of them. These techniques extract features from input data, but real-world image data is complex, redundant, and highly variable. Objects can appear differently from image to image, making hand-crafted features less robust and more computationally intensive. Therefore, extracting hand-crafted features from large and complex sets of images is not feasible.

In deep learning-based techniques, feature extraction methods are automatically learned. Convolutional Neural Networks (CNNs) are deep neural network architectures designed for working on images, videos, sound spectrograms in speech processing, character sequences in text, and so on [14,15]. Compared to hand-crafted features-based techniques, CNNs have made tasks much easier by automating the feature extraction process. With good accuracy levels, these networks can distinguish visual categories.

---

These advancements are now widely used in face detection and recognition, personal photo search, perception in robotics, self-driving cars, and other related fields [16].

Convolutional neural networks comprise one or more convolutional layers, followed by one or more fully connected layers. This architecture divides the lower layer into small regions known as receptive fields. Each receptive field is then mapped with the neurons of the upper layers to extract features. Below are some of the most popular CNN architectures:

LeCun Yann developed the first CNN architecture to identify handwritten digits provided by the U.S. Postal Service, LeNet, in the 1990s [14]. Alex Krizhevsky, Ilya Sutskever and Geoff Hinton developed AlexNet in 2012. Unlike the LeNet, AlexNet is more profound and extensive, with eight layers [17]. Szegedy et al. expanded GoogLeNet by adding an inception module to help reduce the number of parameters in the network [18]. Karen Simonyan and Andrew Zisserman grew the VGGNet with 16 convolutional layers and three fully connected layers [2]. The depth of the network is the main component for better performance. Instead of the  $11 \times 11$ ,  $7 \times 7$  and  $5 \times 5$  convolutions in AlexNet, the VGGNet performs  $3 \times 3$  convolutions and  $2 \times 2$  pooling from the beginning to the end.

Pooling is used to preserve more task-related information, compact representations, and robustness to noise and clutter [19]. They alleviate the problem of over-fitting. Then, an activation function produces a non-linear decision boundary from linear combinations of the weighted input [20]. Several pooling functions, such as max pooling [21], average pooling [22], and k-max pooling [15], are commonly used at the pooling stage.

He et al. developed ResNet [23], which features unique skip connections and heavy use of batch normalization. This network is also missing fully connected layers at the end of the network. In DenseNet, each layer connects with every other layer in the model in a feed-forward manner [24]. Therefore,  $L$  layers of DenseNet have  $L(L+1)/2$  direct connections. As a result, the feature map of all preceding layers is input to the current layer, and its feature maps are used as inputs to all subsequent layers.

### 1.1.2 Natural Language Processing

Generating text from an NLP standpoint involves a series of steps. The first step is content selection, where we identify the key elements of the input. The next step is text planning, where we organize the content. Finally, we move on to surface realization, which involves verbalizing the content. This requires tokenization, which means choosing the appropriate words, generating referential expressions using applicable pronouns, and combining related information through aggregation [25].

Recurrent Neural Network (RNN) [26] and Long Short-Term Memory (LSTM) [27] are two popular deep learning-based language models that have shown outstanding performances in many natural language processing tasks, including image captioning [28–32]. In image captioning, image features extracted from a CNN encoder are given as input to RNN or LSTM for decoding. The decoder predicts the probability of each word given the previous comments.

LSTM networks are a type of RNN that has special units in addition to standard units. LSTM units can actively maintain self-connecting loops involving an additional memory output. Thus, they can retain information in memory for long periods. Similarly, the Gated Recurrent Unit (GRU) used fewer gates to control the flow of information and remove the separate memory cells [33]. Furthermore, Bi-directional LSTM (Bi-LSTM) [34] computes information in both forward and backward directions. They combine the information using two hidden states and can preserve both past and future contexts.

CNNs can learn the internal hierarchical structure of the sentences, and they are faster in processing than LSTMs. Therefore, convolutional architectures have recently been used in other sequence-to-sequence tasks, such as conditional image generation [35] and machine translation [36–38].

Since the attention appeared, it has remarkably improved over encoder decoder-based NLP tasks [39–41]. Attention mechanisms such as soft attention [32] and hard attention [32] have also been used in image captioning methods [32, 42, 43]. In these methods, attention mechanisms can dynamically focus on the relevant parts of the image



(a) Caption 1: Dog.  
Caption 2: Black dog on the grass.



(b) Caption 1: Book.  
Caption 2: Book holding in hand.



(c) Caption 1: Car.  
Caption 2: Tilted car.

Figure 1.3: Semantic Understanding

while the output sequences are being produced.

## 1.2 Main Research Challenge in Image Captioning

Deep Learning-based techniques, specifically CNNs, have contributed substantially to image understanding. However, correct and precise recognition of objects in an image is one of the crucial requirements of image understanding. Despite comprehensive research in this area, accurate and precise recognition of multiple things is still a challenging problem [44].

Most existing image captioning models, including deep learning-based techniques, only focus on the factual description of an image. During feature learning, these methods compress the entire scene into a fixed vector representation. As a result, they often lose the information on relevant objects in the set [31, 32].

Image captioning is still challenging because it requires understanding the objects and attributes and inferring the underlying semantic information [45]. Figure 1.3 shows several examples of semantic knowledge. "Black dog on the grass" is semantically more meaningful than only "Dog" in Figure 1.3a. Similarly, "Book holding in hand" and "Tilted car" are semantically correct and meaningful for Figure 1.3b and Figure 1.3c, respectively. The context of the relationship between objects of an image plays a significant role in semantic understanding. A suitable context estimation can reduce the semantic gap between visual appearance and appropriate textual description of the image [46].



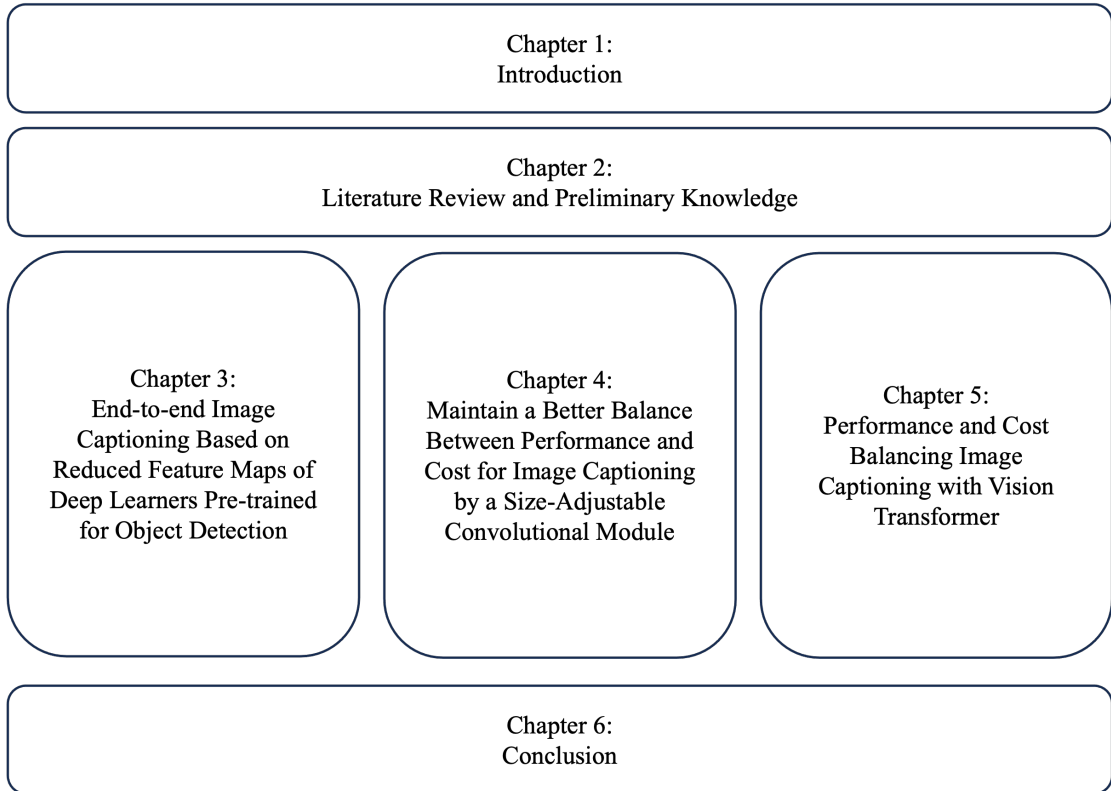


Figure 1.4: Structure of Thesis

Existing image captioning techniques use human-annotated authentic images for training and texting, which involve an expansive and time-consuming process. Moreover, much content, including images, is generated automatically, e.g., for news, illustration, artwork, promotion, human-computer interaction, and augmented reality. There is a need to use these generated or synthetic images for training and texting image captioning methods. There is also a need to create captions for the given photos.

### 1.3 Structure of Thesis

The thesis is mainly divided into seven chapters. The first chapter is the Introduction, which provides preliminary information on image captioning and its related research, image understanding, and natural language processing. Chapter 2 includes a literature review of image captioning and a relative preliminary knowledge introduction. Chapters 3, 4, and 5 are novel contributions to this thesis. Each of these chapters contains an introduction and detailed experiments. Finally, I conclude this dissertation

---

in Chapter 6.

## 1.4 Motivations

Given the overview and the research challenges of Sections 1.1 and 1.2, we have the following aims and objectives in this thesis:

- To train a model for high-quality image caption generation, which incorporates correct and relevant object information.
- Reducing the feature dimension to meet the limitation of computational resources for real-time generation.
- To select more valuable features of images for generating descriptions with correct segmentation information.

Two main motivations that inspired us and will be reflected in this dissertation.

**Motivation #1: It is essential to design an image captioning model that is accurate and affordable (reduces trainable parameters) in the mobile-level computational device.**

If we take a brief look at the literature review of image captioning and state-of-the-art practices in the deep learning-based image captioning task, we can realize that:

- Scaling up the SOTA image captioning model architecture will give superior image captioning performance, provided we have a proportional amount of training data and powerful hardware to train on them.
- Image captioning through feature extractors pre-trained for object recognition can produce highly accurate neural networks. However, such large-scale models are expensive procedures and can take days to train, even with a moderate amount of GPUs.
- As we aim to design a mobile device-oriented model for blind people, how to maintain performance and speed is one of the most crucial issues. Thus, our final goal is to design a mobile-device-oriented image captioning model that can

achieve performance comparable to the SOTA model practices while keeping computational and time costs in the training and testing phases.

**Motivation #2: Although CV and NLP received high performance for several years, their combination’s performance for image captioning is still sparse and has plenty of improvement opportunities. As a result, we aim to contribute a complete framework for image captioning with the help of the CNN feature extractor. We also shed light on several practical and potential research issues and opportunities for image captioning.**

The first image captioning model appeared in the literature titled *Show and Tell: A Neural Image Caption Generator* [31] for automatic image caption generation. The paper introduces an end-to-end neural network model that uses a CNN for image feature extraction and then feeds these features into an RNN to generate natural language descriptions associated with the images. Later, many image captioning models were proposed for the image captioning task and received SOTA performance.

This dissertation is approached to continue and overcome the sparsity of practice in image captioning tasks by implementing a complete framework based on the SOTA practices.

## 1.5 Main Contributions

This dissertation reports my three-year research on Keeping the Balance Between Performance and Computational Cost for Image Captioning. Chapter 2 summarises the literature review of image captioning. Chapters 3, 4 and 5 present our novel research works and experimental results. Chapter 3 first introduced our novel end-to-end image captioning model based on reduced feature maps for image captioning. Chapter 4 proposed a Darknet-based model with a Size-Adjusted Convolutional Module for feature dimension reduction to balance the performance and computational costs of image captioning. Chapter 5 introduced a Vision Transformer-Based image captioning model. Furthermore, considering the analysis of large-scale feature maps requires

---

much more computational cost, we proposed a model with feature dimension reduction. Finally, Chapter 6 includes several supplementary experiments that can guide us to both promising directions and directions to avoid. Our contributions to each chapter are summarized below:

### **1.5.1 Chapter 2**

This chapter proposes a literature review of existing image captioning methods, including template-based image captioning, retrieval-based image captioning and deep learning-based caption generation. In addition, we introduced the related preliminary knowledge for our proposed methods, such as encoder-decoder architecture, VGGNet, Darknet, Vision Transformer and LSTM.

### **1.5.2 Chapter 3**

Following the previous research, this chapter introduced a novel encoder-decoder model for image captioning tasks. The summary of contributions of this chapter is as follows:

- This chapter proposed a novel end-to-end image captioning model architecture that combines a Darknet-based feature extractor with an LSTM-based caption generator. Unlike existing models that rely on pre-trained CNNs as intermediaries, our model allows for a direct path from raw images to generated captions, simplifying the overall process.
- By training the entire model jointly, we ensured a closer alignment between the image feature extractor and the caption generator, resulting in image features better suited for the caption generation task. This approach enhances the quality and accuracy of the generated captions.
- Our model is designed with a low parameter configuration, making it highly suitable for resource-constrained environments. Given its reduced computational and

prediction time overhead, its efficiency is particularly advantageous for caption generation on mobile devices.

- Extensive empirical research and theoretical analysis were conducted on the Flickr 8K dataset, substantiating the effectiveness and efficiency of our proposed model. Furthermore, we conducted a comprehensive performance evaluation, comparing our model with baseline approaches and existing methods, highlighting its competitive advantages.

### 1.5.3 Chapter 4

The Darknet-based image captioning model in Chapter 2 achieved comparable performance by end-to-end training and feature dimension reduction. It proves the convolutional layer does work on keeping performance and cost balance by reducing the feature dimension. One simple question we try to address in Chapter 3 is: *what size of reduced feature dimension fits our proposed model better?*

The chapter thus performs research on this question through the following contributions.

- To address this challenge, the chapter introduces an innovative strategy—leveraging a deep learning model pre-trained for object detection to encode input images. This approach efficiently extracts features representing various objects within the image, streamlining the generating of captions. In addition, we reduced feature dimension by convolutional layer for losing less image information.
- Another critical innovation is integrating a size-adjustable convolutional module (SACM) before decoding image features into coherent sentences. SACM provides flexibility and adaptability in handling different types of images and improves the overall captioning performance.

---

## 1.5.4 Chapter 5

It is too difficult for blind people to live in this world to live much more comfortably. Helping them see the world more clearly is an exciting and vital task. Image captioning is a machine learning task of automatically generating a descriptive statement for the given image by combining image and natural language processing. The existing approaches divided the whole processing into encoding and decoding. This paper proposed a new idea captioning model based on a vision transformer and LSTM, which was compared with some other backbone vision models, such as VGGNet, YOLO, and so on. As we want to use it for real-time generation, the proposed and existing models are compared on both the BLEU score and training and testing time. Moreover, this is the first time the vision transformer method has been used on an image caption generation task.

Chapters 3 and 4 prove that the convolutional layer helps reduce feature dimension based on our proposed architecture. Chapter 5 proposed a more efficient and practical object recognition model, Vision Transformer, as the encoder and the convolutional dimension reduction module enhance the performance.

The contribution of Chapter 5 can be summarized as follows.

- The text highlights the significant challenges faced by blind individuals in navigating the world and underscores the importance of developing technologies to help them better understand their surroundings. It emphasizes the importance of image captioning as a machine-learning task that can bridge the gap between visual information and natural language processing.
- The paper introduces a new image captioning model that combines the power of a vision transformer and LSTM. This approach represents an innovative departure from existing methods, which typically involve encoding and decoding stages. Importantly, this research marks the first utilization of the vision transformer method in image caption generation.
- The study evaluates the proposed model against backbone vision models like VG-

GNet and YOLO. Furthermore, it emphasizes the practicality of real-time image caption generation and, as a result, assesses the proposed and existing models in terms of BLEU scores and training/testing time. This holistic evaluation aims to determine the model's effectiveness for real-world applications, particularly in providing visually impaired individuals with immediate, contextually relevant information about their surroundings.

In contrast, the image captioning model proposed in each chapter is the successor of the previous chapter with several incremental improvements, such as better metrics and computational efficiency.

# Chapter 2

## Literature Review and Preliminary Knowledge

This chapter proposes a literature review of existing image captioning methods, including template-based image captioning, retrieval-based image captioning and deep learning-based caption generation. In addition, we introduced the related preliminary knowledge for our proposed methods, such as encoder-decoder architecture, VGGNet, Darknet, Vision Transformer and LSTM.

### 2.1 Literature Review

Image captioning is a popular research area of Artificial Intelligence (AI) that deals with image understanding and a language description for that image. Image understanding needs to detect and recognize objects. It also needs to understand scene type and location, object properties and their interactions. Generating well-formed sentences requires a syntactic and semantic understanding of the language [31].

Template-based approaches have fixed templates with several blank slots to generate captions. In these methods, different objects, attributes, and actions are detected first, and then the blank spaces in the templates are filled. For example, Farhadi et al. [47] use a triplet of scene elements to fill the template slots for generating image captions. Li et al. extract the phrases related to detected objects, attributes and their relationships



for this purpose [48]. Kulkarni et al. proposed a Conditional Random Field (CRF) to infer the objects, attributes, and prepositions before filling in the gaps [49]. Template-based methods can generate grammatically correct captions. However, templates are predefined and cannot generate variable-length captions. Moreover, parsing-based language models were introduced in image captioning [50-54], which are more powerful than fixed template-based methods. Therefore, this chapter does not focus on these template-based methods.

Captions can be retrieved from visual space and multimodal space. In retrieval-based approaches, captions are retrieved from a set of existing captions. Retrieval-based methods first find visually similar images with their captions from the training dataset. These captions are called candidate captions. The captions for the query image are selected from these captions pool [55-58]. These methods produce general and syntactically correct captions. However, they cannot generate image-specific and semantically correct captions.

In deep machine learning-based techniques, features are learned automatically from training data, and they can handle a large and diverse set of images and videos. For example, CNNs are widely used for feature learning, and a classifier such as softmax is used for classification. RNN generally follows CNN to generate captions.

Novel captions can be generated from both visual space and multimodal space. A general approach of this category is to analyze the visual content of the image first and then generate image captions from the visual content using a language model [32, 43, 59, 60]. These methods can generate new captions for each semantically more accurate image than previous approaches. Most novel caption generation methods use deep machine learning-based techniques. Thus, deep learning-based novel image caption-generating methods are our main focus in this chapter.

Figure 2.1 shows an overall taxonomy of deep learning-based image captioning methods. The figure illustrates the comparisons of different categories of image captioning methods. Deep learning-based methods mostly use visual space and multiple space-based methods. Most public datasets, such as Flickr and MSCOCO datasets, have

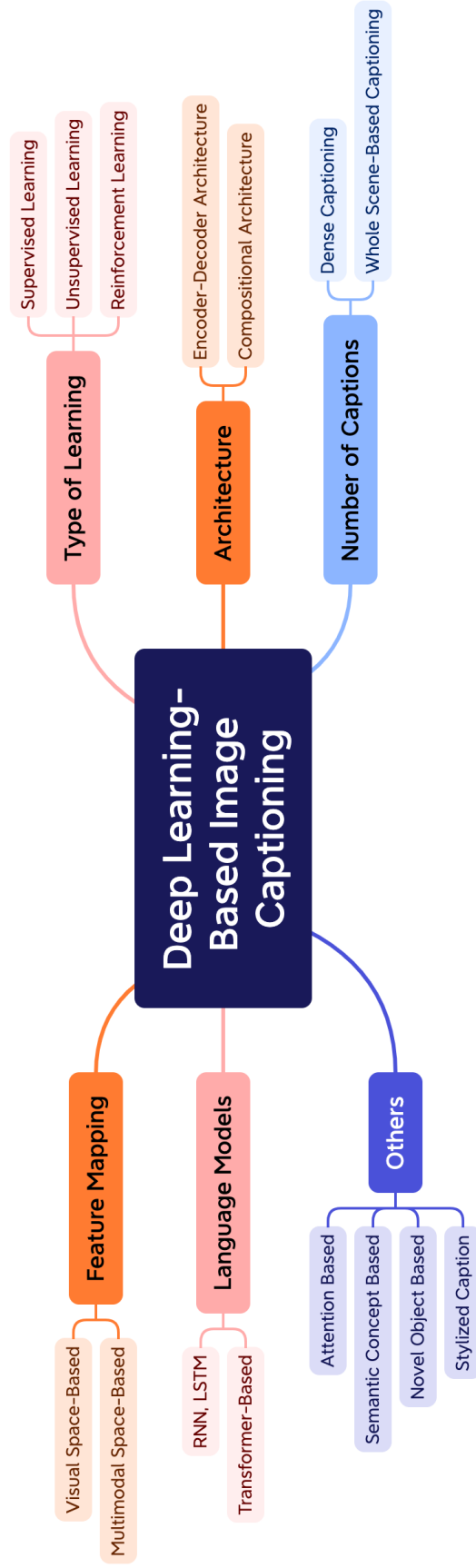


Figure 2.1: An overall taxonomy of deep learning-based image captioning.

the corresponding captions as text. In the visual space-based methods, the image features and the corresponding captions are independently input to the language decoder. In contrast, a multimodal space-based model learns a shared multimodal space from the images and the corresponding caption text. This multimodal representation is then input to the language decoder. Deep learning-based image captioning methods can also be categorised by learning techniques: supervised learning, reinforcement learning and unsupervised learning.

Generally speaking, captions are generated for a whole scene in the image. However, captions can also be generated for different regions of an image (Dense captioning). Image captioning methods can use either simple encoder-decoder architecture or compositional architecture. Most of such methods use LSTM as a language model. However, many methods use other language models such as CNN, RNN and transformer-based decoders.

Deep learning-based image captioning methods can generate captions from visual and multimodal spaces. Public image captioning datasets have the corresponding captions as text. In the visual space-based methods, the image features and the corresponding captions are independently passed to the language decoder. In contrast, a shared multimodal space is learned from the images and the corresponding caption text in a multimodal space case. This multimodal representation is then passed to the language decoder.

### **2.1.1 Multimodal Space-Based Methods**

Initially, Kiros et al. proposed an image captioning model with a CNN for extracting image features [59]. It used a multimodal space that jointly represents image and text for multimodal representation learning and image caption generation. It also introduces the multimodal neural language models such as the Modality-Biased Log-Bilinear Model (MLBL-B) and the Factored 3-way Log-Bilinear Model (MLBL-F) of reference [61] followed by AlexNet [17]. Unlike most previous approaches, this method does not rely on additional templates, structures, or constraints. Instead, it depends on the high-

---

level image features and word representations learned from deep neural networks and multimodal neural language models. The neural language models have limitations in handling a large amount of data and are inefficient in working with long-term memory [62].

Kiros et al. extended their work by learning a joint image sentence embedding where LSTM is used for sentence encoding and a new neural language model called the structure-content neural language model (SC-NLM) is used for image captions generations [63]. The SC-NLM has an advantage over existing methods in that it can extricate the structure of the sentence to its content produced by the encoder. It also helps them to achieve significant improvements in generating realistic image captions.

Karpathy et al. proposed a deep, multimodal model, embedding image and natural language data for bidirectional images and sentence retrieval tasks [64]. The previous multimodal-based methods use a common embedding space that directly maps images and sentences. However, this method works at a finer level and embeds fragments of images and fragments of sentences. This method breaks down the images into several objects and sentences into dependency tree relations (DTR) [65] and reasons about their latent, inter-modal alignment. It shows that the method achieves significant improvements in the retrieval task compared to the previous methods.

However, this method has a few limitations as well. Regarding modelling, the dependency tree can model relations easily, but they are not always appropriate. For example, a single visual entity might be described by a single complex phrase that can be split into multiple sentence fragments. The phrase "black and white dog" can be formed into two relations conjunct (CONJ, white and black) and adjectival modifier (AMOD, white and dog).

Mao et al. proposed a multimodal Recurrent Neural Network (m-RNN) method for generating novel image captions [66]. This method has two sub-networks: a deep CNN for images and a deep RNN for sentences. These sub-networks interact with each other in a multimodal layer to form the whole m-RNN model. Both images and fragments of sentences are given as input in this method. It calculates the probability distribution

to generate the next word of captions. This model has five more layers: Two-word embedding layers, a recurrent layer, a multimodal layer and a softmax layer.

In addition, Kiros et al. proposed a method built on a Log-Bilinear model and used AlexNet to extract visual features. This multimodal recurrent neural network method is closely related to the method of Kiros et al. The authors use a fixed-length context, whereas, in this method, the temporal context is stored in a recurrent architecture, which allows an arbitrary context length. The two word-embedding layers use one hot vector to generate a dense word representation. It encodes both the syntactic and semantic meanings of the words. The semantically relevant words can be found by calculating the Euclidean distance between two dense word vectors in embedding layers.

Most sentence-image multimodal methods [59, 64, 67, 68] use pre-computed word embedding layers and learn them from the training data. This helps them to generate better image captions than the previous methods. Many image captioning methods [63, 64, 69] are built on recurrent neural networks at the contemporary time steps. They use a recurrent layer for storing visual information. However, m-RNN uses both image representations and sentence fragments to generate captions. It utilizes the capacity of the current layer more efficiently, which helps to achieve a better performance using a relatively small dimensional recurrent layer.

Chen et al. proposed another multimodal space-based image captioning method. The method can generate novel captions from images and restore visual features from the description. It also can describe a bidirectional mapping between images and their captions. Many existing methods [56, 64, 68] use joint embedding to generate image captions. However, they do not use reverse projection that can generate visual features from captions. On the other hand, this method dynamically updates the visual representations of the image from the generated word. It has an additional recurrent visual hidden layer with RNN that makes reverse projection.

---

## 2.1.2 Visual Space-Based Methods

As shown in Figure 2.1, deep learning-based image captioning can be categorized into supervised learning, unsupervised learning and reinforcement learning in terms of the type of learning. Training data come with a desired output called *label* in supervised learning. On the contrary, unsupervised learning deals with unlabeled data. Reinforcement learning is another machine learning approach where an agent aims to discover data and/or labels through exploration and a reward signal.

### Supervised Learning

Supervised learning-based networks have successfully been used for many years in image classification [2,17,18,23], object detection [70-72] and attribution learning [73]. This process makes researchers interested in using them in automatic image captioning [5,31,66,74].

Dense captioning [75] proposes a fully convolutional localization network architecture composed of a convolutional network, a dense localization layer, and an LSTM [27] language model. The dense localization layer processes an image with a single, efficient forward pass, implicitly predicting regions of interest in the image. It requires no external region proposals, unlike Fast R-CNN or a full network of Faster R-CNN [72]. The working principle of the localization layer is related to the work of Faster R-CNN.

However, Johnson et al. use a differential, spatial soft attention mechanism [76, 77] and bilinear interpolation [77] instead of an ROI pooling mechanism [70]. This modification helps the method to backpropagate through the network and smoothly select the active regions. It uses the Visual Genome dataset for the experiments in generating region-level image captions.

Region-based descriptions are objective and detailed. The region-based method is known as dense captioning. However, there are some challenges in dense captioning. As regions are dense, one object may have multiple overlapping regions of interest. Moreover, it is challenging to recognize each target region for all the visual concepts.

The neural network-based image captioning methods work in just a simple end-to-

end manner. These methods are similar to the encoder-decoder framework-based neural machine translation [78]. In such networks, global image features are extracted from the hidden activations of CNNs and fed into an LSTM to generate a sequence of words.

To obtain a comprehensive understanding of objects and relationships in the images and generate fluent sentences to match the visual information, the encoder-decoder models often adopted the framework of CNN plus RNN. The CNN extracts the scene type to detect the objects and their relationships. After that, a language model uses the output of CNN to convert them into words, combined phrases that produce image captions. A simple image captioning model configuration is shown in the upper part of Figure 2.2.

Vinyan et al. proposed a Neural Image Caption Generator (NIC) method [31]. The method uses a CNN for image representations and an LSTM for captions generation. This special CNN uses a novel method for batch normalization, and the output of the last hidden layer of CNN is used as an input to the LSTM decoder. This LSTM can keep track of the objects that have been described using text. NIC is trained based on maximum likelihood estimation.

In generating image captions, image information is included in the initial state of an LSTM. The next words are generated based on the current time step and the previous hidden state. This process continues until it gets to the end token of the sentence. Since image information is fed only at the beginning of the process, it may face vanishing gradient problems. The role of words generated at the beginning is also weakening. Therefore, LSTM still faces challenges in generating long-length sentences [79,80].

Jia et al. proposed an extension of LSTM called guided LSTM (gLSTM) [81]. This gLSTM can generate long sentences. This architecture adds global semantic information to each gate and cell state of LSTM. It also considers different length normalization strategies to control the length of captions. Semantic information is extracted in different ways. First, it uses a cross-modal retrieval task to retrieve image captions, and then semantic information is extracted from them. The semantic-based information can also be extracted using a multimodal embedding space.



**Input Image**

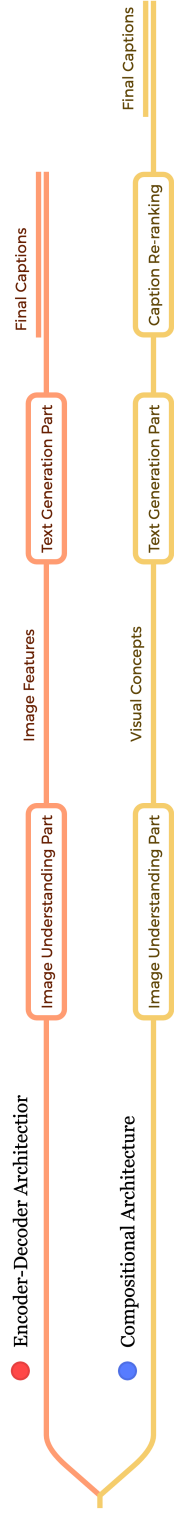


Figure 2.2: A summarized block diagram of the encoder-decoder and compositional architecture.



Mao et al. proposed a special text generation method for images [82]. This method can generate a description for a specific object or region that is called referring expression [83–89]. This expression can then infer the object or region being described. Therefore, the generated description or expression is quite unambiguous. This method uses a new dataset called ReferIt [89] based on the popular MSCOCO dataset to address the referring expression.

Previous CNN-RNN-based image captioning methods use unidirectional LSTM, which is relatively shallow in depth. The next word is predicted based on visual and previous textual contexts in unidirectional language generation techniques. Unidirectional LSTM cannot generate contextually well-formed captions. Moreover, recent object detection and classification methods [2, 17] show that deep, hierarchical methods are better at learning than shallower ones.

Wang et al. proposed a deep bidirectional LSTM-based method for image captioning [6]. This method is capable of generating contextually and semantically rich image captions. The proposed architecture consists of a CNN and two separate LSTM networks. It can utilize past and future context information to learn long-term visual language interactions.

Compositional architecture-based methods comprise several independent functional building blocks. First, a CNN extracts the semantic concepts from the image. Then, a language model generates a set of candidate captions. These candidate captions are re-ranked in the generation using a deep multimodal similarity model. A common block diagram of compositional network-based image captioning methods is shown in the lower part of Figure 2.2. A typical method of this category maintains the following steps:

- A CNN extracts image features.
- Visual concepts are obtained from visual features.
- Multiple captions are generated using the previously received information by a language model.

- 
- The generated captions are re-ranked using a deep multimodal similarity model to select high-quality image captions.

Fang et al. introduced generation-based image captioning [90]. It uses visual detectors, a language model, and a multimodal similarity model to train the model on an image captioning dataset. Image captions can contain nouns, verbs, and adjectives. A vocabulary is formed using the 1,000 most common words from the training captions. The system works with the image sub-regions instead of the full image. CNNs are used for extracting features for the sub-regions of an image. The features of sub-regions are mapped with the vocabulary words likely to be contained in the image captions.

Multiple instance learning [91] is used to train the model for learning discriminative visual signatures of each word. A maximum entropy [92] language model generates image captions from these words. A linear weighting of sentence features ranks generated captions. Minimum Error Rate Training [93] is used to learn these weights. The similarity between image and sentence can be easily measured using a common vector representation. A deep multimodal similarity model maps images and sentence fragments with the common vector representation. It achieves a significant improvement in choosing high-quality image captions.

The object detection model based on a faster R-CNN [70] with ResNet-101 was used to extract salient objects as regional visual features to generate image captions [42,70]. This model's final output performed non-maximum suppression for each object class using an intersection over union (IoU) threshold. All regions would be selected if any class-detection probability exceeded a confidence threshold. After that, the mean-pooled convolutional features were considered as features input into LSTM to generate captions. Indeed, it is not likely for LSTM to receive the complete information from all the predicted anchor boxes. For example, the pot, the cooker, and other similar items in a given image might show the same meaning of cooking.

The attention mechanism is an approach to decide whether to attend to visual or non-visual information at each step of the decoder part [94]. With the development of the attention mechanism, a two-level attention network was implemented based on

Table 2.1: Literature summary of image captioning. T-B, R-B, and E&D denote template-based, retrieval-based, and encoder-decoder methods.

	<b>Method</b>	<b>Main Property</b>	<b>Presented Papers</b>
	T-B	Fixed templates with several blanks are used to generate captions.	[47-49]
	R-B	The model finds a similar image from the training set, and then its corresponding caption is selected as a result.	[55-58]
	CNN+RNN	Introduced two-step approaches for image captioning of presenting images by CNNs and analyzing the presentation by RNNs.	[31,32,43,59,60]
E&D	CNN+RNN+Attention	Applied attention mechanism allows the model to focus on different regions at each step.	[94,95]
	CNN+RNN+Reinforcement Learning	The reinforcement model learns to optimize a reward function based on human evaluations.	[42]
	Transformer-Based	Applied the transformer architecture, originally designed for machine translation, for image captioning.	[96]
	Pretrained Vision-Language Model	Demonstrated the effectiveness of pre-trained models on large-scale vision-language datasets.	[97-99]

---

attributes and the attention mechanism [95]. With the attention mechanism and the multi-head architecture, transformers have been used in natural language processing and computer vision processes. A dual-level collaborative transformer for image captioning was developed in 2021. This model integrated regions and grids' appearance and geometry features with intra-level fusion based on comprehensive relation attention and dual-way self-attention [96]. Such grid features from transformer-based networks performed much better than previous results.

More and more large-scale models are designed for tasks related to computer vision, natural language processing, etc. The effectiveness of pre-trained large-scale models on image captioning has been proved in [97-99]. Large-scale models, however, often require a longer computation time and more memory. When the computational resources are limited, it is necessary to develop lightweight models for realizing the encoders and/or decoders in image captioning. The summarized literature review of deep learning-based methods is shown in Table 2.1.

## **Unsupervised Learning**

Unsupervised learning methods received good performance on machine translation. In the unsupervised machine translation methods [4, 26, 27], the source language and target language are mapped into a common latent space so that the sentences of the same semantic meanings in different languages can be well aligned, and the following translation can thus be performed. Unsupervised image captioning is similar in spirit to unsupervised machine translation. Unsupervised image captioning relies on a set of images, a set of sentences, and an existing visual concept detector.

The Functional Magnetic Resonance Imaging (fMRI) technique [100] introduced an unsupervised learning model to generate captions using human brain activity through a robust regression scheme. This method converts fMRI data into text features and uses LSTM to generate captions. They make text features and compare them with brain data. Lastly, they use text feature information with unlabeled data images.

Laina et al. proposed a method to align images and text in a shared latent represen-

tation structured through visual concepts [101]. This method is minimally supervised because it requires a standard, pre-trained image recognition model to obtain initial noisy correspondences between the image and the text domain. The translation from image features to text is learned from weakly paired images and text using a loss robust to noisy assignments and a conditional adversarial component.

Zhou et al. research the relationship between sentences and images to generate better captions. They proposed the TSGAN method [102], which means triple sequence generative adversarial nets. The image encoder uses CNN to encode images to different regions for extracting visual concepts. The triple cell proposed in this paper has an image generator, sentence generator and discriminator. The image generator generates image regions for different words. The sentence generator guides the generated captions, and the discriminator helps to improve captions by checking relevancy.

Feng et al. proposed a novel method to train an image captioning model unsupervised without using any paired image-sentence data [103]. They presented three training objectives, which encourage that 1) the generated captions are indistinguishable from sentences in the corpus, 2) the image captioning model conveys the object information in the image, and 3) the image and sentence features are aligned in the common latent space and perform bi-directional reconstructions from each other.

Unsupervised learning is a good choice for training with large-scale datasets. However, the unsupervised image captioning task is more challenging because images and sentences reside in two modalities with significantly different characteristics. The language source must contain sufficient visual concepts overlapping with the image domain to generate the initial assignments.

### **Reinforcement Learning**

A reinforcement learning approach is designed by many parameters such as agent, state, action, reward function, policy and value. The agent chooses an action, receives reward values and moves to a new state. The policies are defined by actions and the values by reward function. The agent attempts to select the action with the expectation of

---

having a maximum long-term reward. It needs continuous state and action information to guarantee a reward function.

Traditional reinforcement learning approaches face several limitations, such as the lack of guarantees of a reward function and uncertain state-action information. Policy gradient methods [104] are a type of reinforcement learning that can choose a specific policy for a specific action using gradient descent and optimization techniques. The policy can incorporate domain knowledge for the action that guarantees convergence. Thus, policy gradient methods require fewer parameters than reward function-based approaches.

Reinforcement learning-based image captioning methods sample the next token from the model based on the rewards they receive in each state. Policy gradient methods in reinforcement learning can optimize the gradient to predict the cumulative long-term rewards. Therefore, it can solve the non-differentiable problem of evaluation metrics.

In 2017, Ren et al. introduced a novel reinforcement learning-based image captioning method [105]. The architecture of this method has two networks that jointly compute the next best word at each time step. The "policy network" works as local guidance and helps to predict the next word based on the current state. "The value network" works as global guidance and evaluates the reward value, considering all the possible extensions of the current state. This mechanism can adjust the networks in predicting the correct words. Therefore, it can generate good captions similar to ground truth captions at the end. It used an actor-critic reinforcement learning model [106] to train the whole network. Visual semantic embedding [107,108] is used to compute the reward value in predicting the correct words. It also helps to measure the similarity between images and sentences, which can evaluate the correctness of generated captions.

Rennie et al. proposed another reinforcement learning-based image captioning method [109]. The method utilizes the test-time inference algorithm to normalize the reward rather than estimating the reward signal and normalization in training time. The test-time decoding is highly effective for generating quality image captions.

Zhang et al. proposed an actor-critic reinforcement learning-based image caption-

ing method [110]. The method can directly optimize non-differentiable problems of the existing evaluation metrics. The architecture of the actor-critic method consists of a policy network (actor) and a value network (critic). The actor treats the job as a sequential decision problem and can predict the next token of the sequence. The network will receive a task-specific reward in each state of the sequence. The job of the critic is to predict the reward. If it can predict the expected reward, the actor will continue to sample outputs according to its probability distribution.

In this section, we have reviewed deep learning-based image captioning methods. We have given a taxonomy of image captioning techniques, shown the major groups' generic block diagram, and highlighted their pros and cons. Although deep learning-based image captioning methods have achieved remarkable progress in recent years, a robust image captioning method that can generate high-quality captions for nearly all images is yet to be achieved. With the advent of novel deep-learning network architectures, automatic image captioning will remain an active research area for some time. However, the increasing parameters make it difficult to meet real-time applications' need to receive immediate, contextually relevant information.

## 2.2 Preliminary Knowledge

### 2.2.1 Encoder–Decoder Architecture for Image Captioning

To obtain a comprehensive understanding of objects and relationships in the images and generate fluent sentences to match the visual information, encoder-decoder models often adopted the framework of CNN plus RNN image captioning model configuration shown in Figure 4.1. Not only are they flexible, but they are also effective. Generally, global features are extracted from input images by a CNN model and then fed into an RNN model for sequence generation by transferring the image into a full grammatically and stylistically correct sentence. In some applications, a CNN was used for image representation, while an LSTM was used for caption generation. For example, the NIC (neural image caption generator) [31] and NIC V2 [111] followed such a framework.

---

The output of the last hidden layer of CNN was used as input for the LSTM-based decoder. In image captioning, image information was included in the initial state of LSTM. The NIC models show that improving results by directly maximizing the probability of the correct translation given an input sentence in an end-to-end fashion is possible. The end-to-end models use an RNN, which encodes the variable length input into a fixed dimensional vector. They then use the decoded vector to generate it into the desired output sentence. Therefore, it is natural to use the same approach to image captioning rather than inputting a sentence to translate it into a description.

### 2.2.2 VGGNet

The quality of image captioning mostly depends on the performance of extracting image features. Handcrafted (HC) features are task-specific because most real data are complex and have different semantic interpretations. Therefore, many human and material resources and a significant amount of time were spent on the feature extraction from a large dataset. Using traditional feature-extraction methods in image captioning tasks often involving large data sets is impractical. DL can learn from training data and automatically extract useful features so that even a large and complicated set of images and videos can be handled promptly nowadays. CNNs have been widely used for feature extraction, although they were originally built for classification or object detection tasks.

In image captioning, RNNs generally follow CNNs for caption generation [31]. GoogLeNet [18] had been used as a deep image processing network in some image captioning models. Moreover, VGGNet [2] and ResNet [4] have also been used as image feature extractors in some image caption systems [112]. VGGNet was invented by the Visual Geometry Group from the University of Oxford, which beat GoogLeNet and won the localization task in the ImageNet Large Scale Recognition Challenge (ILSVRC) 2014.

In the original VGGNet, there are three fully connected layers in front of the softmax layer for outputting classes of objects. It has 16 convolutional layers and is appealing



because of its uniform architecture. Using two layers of the  $3 \times 3$  filter, VGGNet could cover  $5 \times 5$  areas. By using three layers of the  $3 \times 3$  filter, it can cover  $7 \times 7$  effective areas. Therefore, large-size filters such as  $11 \times 11$  in AlexNet [17] and  $7 \times 7$  in ZFNet [?] are unnecessary. VGGNet is the community’s most preferred choice for extracting image features. The weight configuration of the VGGNet is publicly available and has been used as a baseline in many other applications.

Table 2.2 suggests that ResNet performs best among the four CNNs, including AlexNet, VGGNet, ResNet, and Inception-X Net, based on the accuracy of both Top-1 and Top-5. Although ResNet also has fewer parameters than VGGNet, VGGNet remains the most popular image feature extractor in applications and has the second-highest result in Table 2.2 [4].

Table 2.2: Comparisons among four CNN architectures [4]. #Multiply-adds and #Params denote the number of operations and the output of each neuron or node.

CONVOLUTIONAL NEURAL NETWORKS ARCHITECTURES					
Architecture	# Param	# Multiply-Adds	Top-1 Accuracy	Top-5 Accuracy	Year
Alexnet	61M	724M	57.1	80.2	2012
VGG	138M	15.5B	70.5	91.2	2013
Inception-V1	7M	1.43B	69.8	89.3	2013
Resnet-50	25.5M	3.9B	75.2	93	2015
Darknet-53	62M	65.86B	77.2	93.8	2018
ViT	86M	33.03B	88.6	97.9	2020

In the original VGGNet, the input image is resized into  $224 \times 224 \times 3$  and sent to the network until the first connected layer. Similar to the VGGNet used as the image presenter in previous vision tasks, the last fully connected layer and softmax layer was removed in our implementation so that the feature size became 4096 as input to the decoder. After that, the feature vectors were sent to the decoder directly. For the results presented in this paper, the weights of the VGG encoder were fine-tuned during the decoder training to let the predicted captions be near the ground-truth captions.

---

### 2.2.3 Darknet

VGGNet performs better on image classification in which there are fewer items. The image captioning tasks require the system to be capable of the prediction of multiple items and the background at the same time. Based on such considerations, Faster R-CNN was used as the backbone in the image captioning model [42]. The R-CNN model first used region proposal methods to generate potential bounding boxes and then applied a classifier to these predicted boxes. Finally, post-processing was used to refine the bounding boxes, eliminate duplicate detection, and re-score the boxes based on other objects in the scene. Such complex pipelines would be slow and hard to optimize because each component must be processed separately.

YOLO [113,114] framed object detection as a single regression problem from image pixels to bounding box coordinates and class probabilities. With the whole processing setting as a single network, it can be processed end-to-end directly on detection performance so that YOLO can learn the representations of objects well. YOLO evolved from YOLOv1 [113] to YOLOv8 [114] and has consistently focused on balancing speed and accuracy, aiming to deliver real-time performance without sacrificing the quality of the detection results.

The original YOLO model was designed with a single convolutional model to predict object locations and classes and enable real-time processing directly. However, the speed-oriented YOLOv1 cannot outperform the accuracy level for dealing with small objects or objects with overlapping bounding boxes.

The later designed YOLO models successfully addressed these limitations while maintaining real-time detection. For instance, YOLOv2 (YOLO9000) [115] with Darknet-19 introduced anchor boxes and pass-through layers to improve the localization of objects, resulting in higher accuracy. In addition, YOLOv3 with Darknet-53 enhanced the performance by employing a multi-scale feature extraction architecture for better object detection across various scales. With the development of backbones, YOLO models can maintain a faster speed and better performance simultaneously.

Models like YOLOv4 and YOLOv5 introduced innovations, such as new network

backbones, improved data augmentation techniques, and optimized training strategies. These developments led to significant gains in accuracy without drastically affecting the models' real-time performance [116]. Darknet-53 is therefore applied as a backbone of the encoder in our model so that captioning could focus on more points like the background and some small-scale details.

### 2.2.4 Transformer

The Transformer architecture, introduced in the seminal paper "Attention Is All You Need" [38] by Vaswani et al. in 2017, is a monumental milestone in natural language processing (NLP). This revolutionary neural network architecture redefined the way sequences of data, such as text, are processed and modelled. Although initially conceived for NLP tasks, the Transformer's versatility quickly transcended its original domain and found application in various other fields, marking a paradigm shift in the design of deep learning models.

The Transformer architecture leverages a mechanism known as self-attention. This innovation allows the model to weigh the importance of different elements in a sequence when making predictions, enabling it to effectively capture long-range dependencies and contextual information. Unlike traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs), Transformers process entire sequences in parallel, rendering them highly efficient and amenable to parallel computing. This architectural leap improved the performance of existing NLP tasks and paved the way for developing more capable and scalable models.

The significance of the Transformer extends beyond its application in natural language understanding and generation. Its attention mechanism has become a cornerstone in various fields, including computer vision, where it inspired the creation of the Vision Transformer (ViT) discussed earlier. Transformers have also found utility in speech recognition, recommendation systems, and scientific applications like protein folding prediction.

---

### 2.2.5 Vision Transformer

Based on the Transformer mentioned previously, the CV field has witnessed a paradigm shift with the introduction of the Vision Transformer (ViT) architecture. ViT represents a novel approach to processing visual information, departing from the conventional Convolutional Neural Networks (CNNs) that have long dominated the computer vision landscape. First proposed by Google Brain in 2020, ViT has rapidly gained traction and garnered widespread attention due to its remarkable performance and versatility.

At its core, ViT reimagines the treatment of images by treating them as sequences of fixed-size, non-overlapping patches. This departure from the grid-based processing of pixels in CNNs opens up new possibilities for capturing long-range dependencies and contextual information within images. ViT leverages self-attention mechanisms, initially popularized in natural language processing, to model intricate relationships between these patches. This self-attention mechanism endows ViT with the ability to understand global image context, making it particularly adept at handling complex scenes and large-scale images.

The advantages offered by ViT are manifold. It excels at tasks such as image classification, object detection, semantic segmentation, and more. Its remarkable scalability enables ViT to generalize effectively across various datasets and tasks without necessitating extensive architectural modifications. Moreover, ViT simplifies network design, reducing engineering complexity while maintaining competitive performance. Additionally, ViT has paved the way for intriguing prospects in transfer learning, fostering knowledge transfer between diverse domains.

### 2.2.6 LSTM

It is difficult for conventional RNNs to access long-range context because the back-propagated errors either inflate or decay over time due to the so-called vanishing gradient problem [117]. LSTM overcomes this problem and allows itself to model the self-learned context information. LSTM has a similar control flow to an RNN. It processes data, passing on information as it propagates forward. The differences are the

operations within the LSTM's cells. The updating of the hidden layer of LSTM is replaced by purpose-built memory cells. LSTM generates captions by making one word at a time, using a context vector, and considering the previously received hidden states and predicted words [112].

The LSTM model consists of a cell state and several gates. The cell state is a transport highway that transfers relative information down the sequence chain, like the memory. The cell state can carry relevant information throughout the processing of the sequence. Therefore, information from the earlier time steps can make its way to later time steps by reducing the short-term memory effect. As the cell state changes, information is added or removed to the cell state via gates. The gates decide which information is allowed in the cell state. The gates can learn what information should be kept or forgotten by training.

## **Chapter 3**

# **End-to-end Image Captioning Based on Reduced Feature Maps of Deep Learners Pre-trained for Object Detection**

Most existing models use the pre-trained CNNs as the encoder and train a natural language model for caption generation. Although some current models perform well with more complex architecture and larger parameter sizes, the pre-trained image feature is not the perfect fit for the caption generator. In addition, the increasingly tricky model architecture requires a higher cost of computational resources and predicting time. This chapter proposed an end-to-end model for image captioning. The architecture consists of a Darknet-based feature extractor and an LSTM-based caption generator. The whole model is trained on the entire set of training data together so that the extracted feature is more fitting to our caption generator. The backbone model is built with VGGNet as the encoder and LSTM as the decoder. The low parameter requirement of our model makes it very suitable in low computational environments and mobile device-oriented prediction. Extensive empirical study and theoretical analysis on Flickr 8K substantiate the effectiveness and efficiency of our proposed model. The

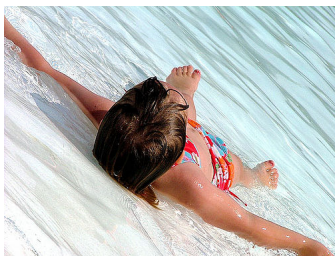
performance of our proposed method is compared with the backbone and some other existing methods.

## 3.1 Introduction

Based on the success of image and natural language processing, image captioning is a multimodal task, including visual and natural language processing. The research aims to summarize the classes of objects and background and their position relationship to generate the caption. For example, a successful model can say something like ‘a couple stands close at the water stage’ based on the image of Figure 3.1. The image recognition model can easily predict the “girl”, “ocean”, “stage”, “horse”, “snowboarders”, “hill”, “dog”, “backyard”, “people”, and the “water stage”, but measuring their position relationship is still difficult.

By analysing the ground truth of the public datasets of this task, it is not difficult to know that the outputs include objects’ names, backgrounds, and positional relationships. So, the application requires the recognition of important objects, backgrounds, and the relationship among them in the image. In [5,66], a multimodal recurrent neural network (m-RNN) method was proposed to explore the relationships between vision and text information and generate sentences to describe the content of a given image, where only the parameters of the m-RNN model were updated. In [118,119], a caption generation system was designed by using the most common words as the semantic attributes, in which both the global image feature and the semantic attribute vectors were used as input to an RNN.

Because of the two combined processes, hybrid systems with a deep network for image processing and another deep network for language processing are often applied. Similarly, the model in reference [31] uses a deep convolutional neural network (CNN) and a recurrent neural network (RNN) as encoder and decoder, respectively. Such as the GoogLeNet [18], which performed best in the ImageNet Large Scale Recognition Challenge (ILSVRC) 2014 classification competition, is used as the deep image processing



(a) A young girl is lying in the sand while ocean water is surrounding her.



(b) Two people stand by the water.



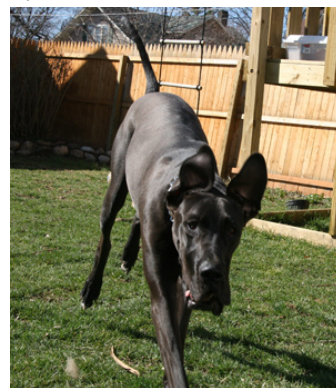
(c) A girl and her horse stand by a fire.



(d) Snowboarders sitting in the snow while skiers take the hill.



(e) A man with a shaved head is kissing another man on the cheek.



(f) A black dog in the backyard.

Figure 3.1: Some example images of Flickr8K dataset [1]. The reference captions are under the image.



network in existing image captioning models. In addition, the VGGNet [2] is also used as the image feature extractor in the existing image caption systems. Compared to the VGGNet, YOLO [113] performs much better in multiple objects' detection. YOLO, or You Only Look Once, is an object detection algorithm much different from the region-based algorithms seen above. In YOLO, a single convolutional network predicts the bounding boxes and the class probabilities for these boxes.

The primary purpose of this chapter is to investigate the potential of the object detection model for image captioning. Suppose we adopt the model as the encoder, that is, use the predicted annotations of the model as the inputs of an RNN-based decoder. In that case, we may not fully use the information provided by the hidden layers (often called the feature maps). On the other hand, if we feed the outputs of one of the hidden layers (i.e. one of the feature maps) directly to the decoder, the number of inputs for the decoder may become too large, and the cost both for training and testing of the decoder part can become prohibitive. In this paper, we propose using a reduced feature map, provided by a model pre-trained for object detection, as the input of the decoder. This way, we expect to use more information for captioning while keeping the budget for training and inference to an acceptable value.

## 3.2 Outline

We arrange this chapter in the following order.

- Section 3.3: An overview of the proposed architecture.
- Section 3.4: Detailed discussion on the image captioning model.
- Section 3.5: Detailed discussion about the dataset and experiments settings.
- Section 3.6: Discussion on results and comparison to state-of-the-art image captioning models.
- Section 4.9: Conclusion and possible future works.

---

### 3.3 Proposed Architecture

Image captioning has become a popular research topic in artificial intelligence (AI), including image understanding and natural language generation. According to the literature review of Chapter 2, most existing image caption models follow the encoder-decoder architecture. Based on this, the image captioning task is related to the image and natural language processing models. With machine learning and deep learning development, research on image processing has been improving rapidly. Visual models can solve more and more CV tasks. At the same time, Natural language pretraining has revolutionized the whole NLP research community.

However, most existing SOTA technologies are usually too heavy and expensive to be implemented in devices or systems with weak computing resources. This is why we still need relatively lightweight models for realizing the encoder and decoder in the image captioning system.

Unlike most existing image caption methods in which pre-trained encoders were often used, the method proposed in this chapter uses an end-to-end approach by simultaneously training both the encoder and decoder. The literature review shows that the VGGNet and ResNet were most commonly used as encoders. Among them, VGGNet was the second-place winner of the 2014 ILSVRC image classification competition, and ResNet was the first-place winner in 2015.

The challenges of image captioning include the comprehensive understanding of objects and relationships in the images and the generation of fluent sentences to match the visual semantics. According to previous research, it is easy to know that the encoder and decoder architecture is the most used one in recent research.

As shown in Figure 3.2 is the flow chart of the training processing. The model consists of the encoder and decoder parts, similar to most existing models. The encoder part extracts image features, and the decoder part is designed to generate text descriptions by analyzing the features. Unlike most existing models, which train the two parts separately, the model proposed in this paper is end-to-end. In other words, the parameters of the two-part update automatically together in our model to find what kind of

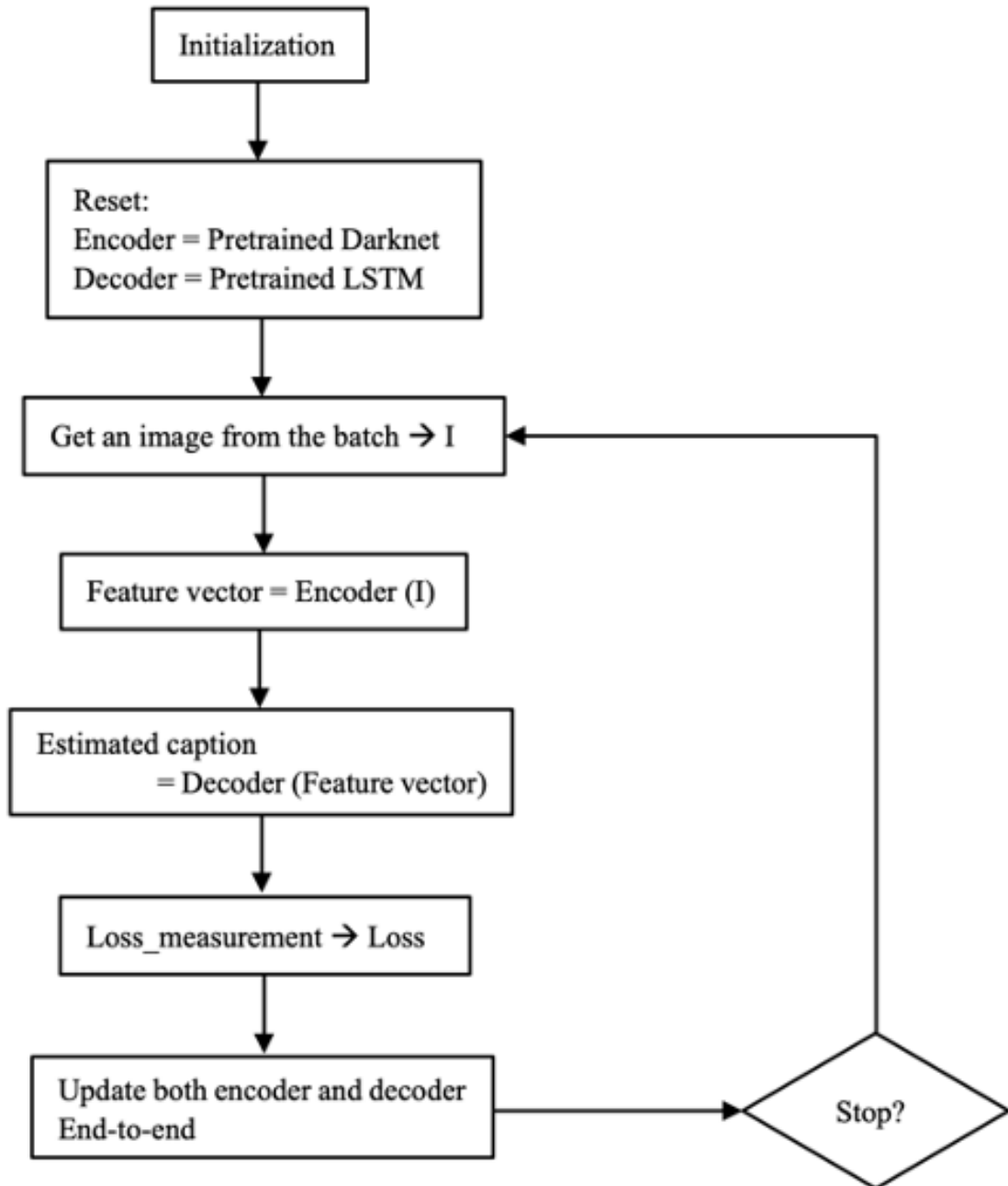


Figure 3.2: Flowchart of the whole training process. The encoder extracts image features, while the decoder generates text descriptions by analysing the features. The estimated vector will be compared with the ground truth for loss measurement.

---

features are more important for this task. After passing through the encoder and decoder, a meaningful text describing the image will be generated. This text vector will be compared with the reference vector during training to get the loss. Then, the encoder and decoder parameters will be optimized together based on the loss. Besides this, the vector will be transferred into the sequence of related words during the testing process. So far, the VGG is one of the most popular models used as an encoder of the image captioning architecture. In this chapter, Darknet-53 is applied as a feature extractor in the image captioning task and receives better results.

Apart from this, the proposed method is end-to-end, which means the training process is for training both the encoder and decoder simultaneously based on the  $\langle \text{data}, \text{ground truth} \rangle$  pairs instead of fixing the encoder and training the decoder only. In other words, providing ground truth information for the encoder is unnecessary. What's more, the reduced feature map is used in our proposed method. As we know, the feature map is generally more informative than the encoder output. However, the cost of training and prediction is proportional to the size of the feature map. A reduced feature map can reduce the training and reference cost and preserve the performance.

### 3.4 Design for Image Captioning

In the original VGGNet, there are three fully connected layers in front of the softmax layer for outputting classes of objects. To have the end-to-end implementation and fewer features, the last fully connected layer and softmax layer were removed so that the feature size becomes 4096 as input of the decoder part. We used the VGG as the baseline in this paper to compare the performance with our proposed model.

The encoder by the VGG model used in this paper is shown in Figure 3.3. Given images are resized into  $224 \times 224 \times 3$  and sent into the network until the first connected layer. After that, the feature vectors are transmitted directly into the decoder part.

Previous research made image caption generation a reality, but measuring and presenting the position relationship among items is still challenging. The Darknet53, de-

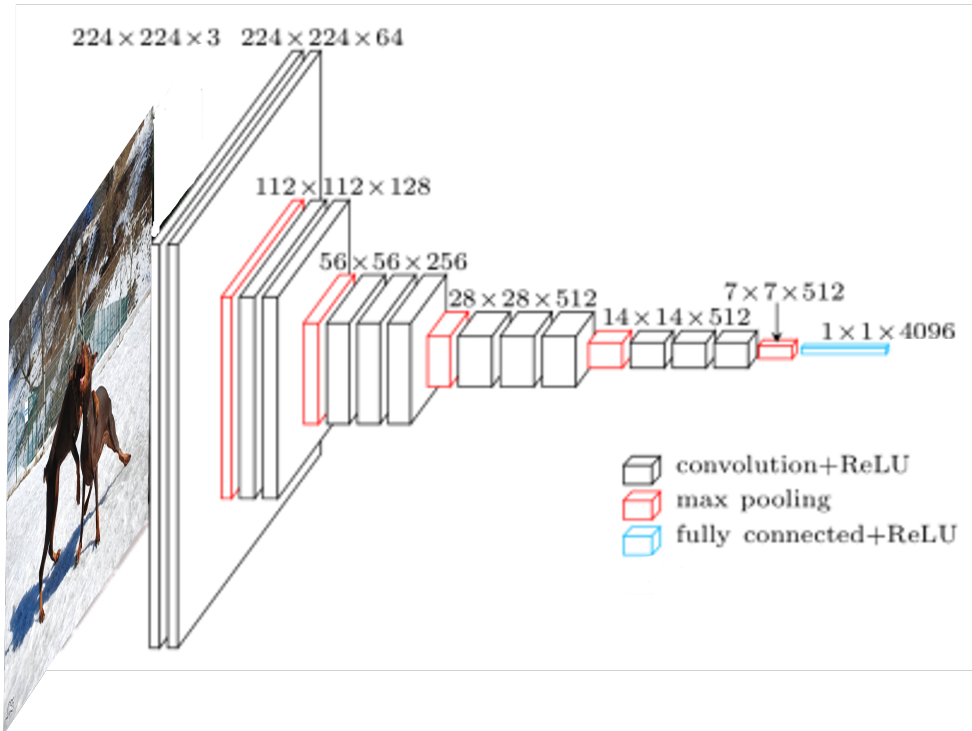


Figure 3.3: Encoder by VGG. Image is input into the model and feature-extracted by the VGG network without the last fully connected and softmax layers. The image is finetuned from [2].

signed as the backbone of image recognition, is applied as our proposed network’s backbone. Then, the residual blocks are kept in the network to help analyze the feature maps. At last, convolutional layers are introduced to contract the feature map for losing information about the given images as little as possible. After the convolutional layers, the reduced feature maps are sent into the decoder part to generate captions of the provided images. The encoder by Darknet-53 used in this paper is shown in Figure 3.4.

LSTM was used as a decoder part of this system. During the pre-processing, the captions will be filled with the “< unk >” for marking unknown words, the “< start >” to mark the start of a new sentence, and the “< end >” for ending the sentence. After that, a dictionary containing both words and their corresponding IDs will be set. The dictionary will be composed of the different words of the whole dataset. After that, the caption will be embedded into a matrix with their corresponding IDs.

As shown in Figure 3.5, there are 2,550 different words in the Flickr 8k dataset [1] used in this paper, so the dimension of  $d$  is 2,550. The feature maps and embedded caption will be input into the LSTM network during training. The *LSTM.Out* in the

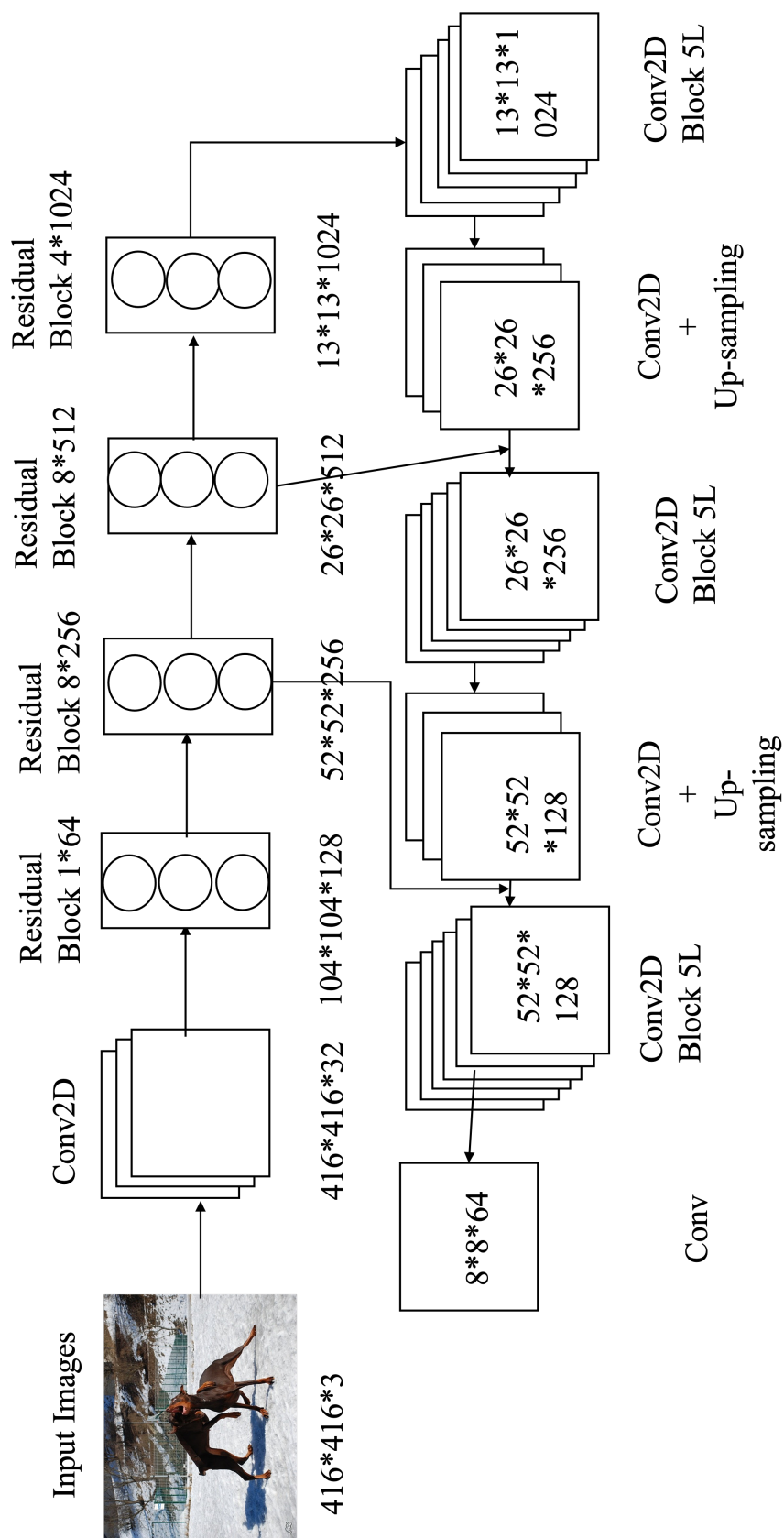


Figure 3.4: Encoder by Darknet-53. Image is input into the model and feature-extracted by the Darknet-53 with several convolutional layers.

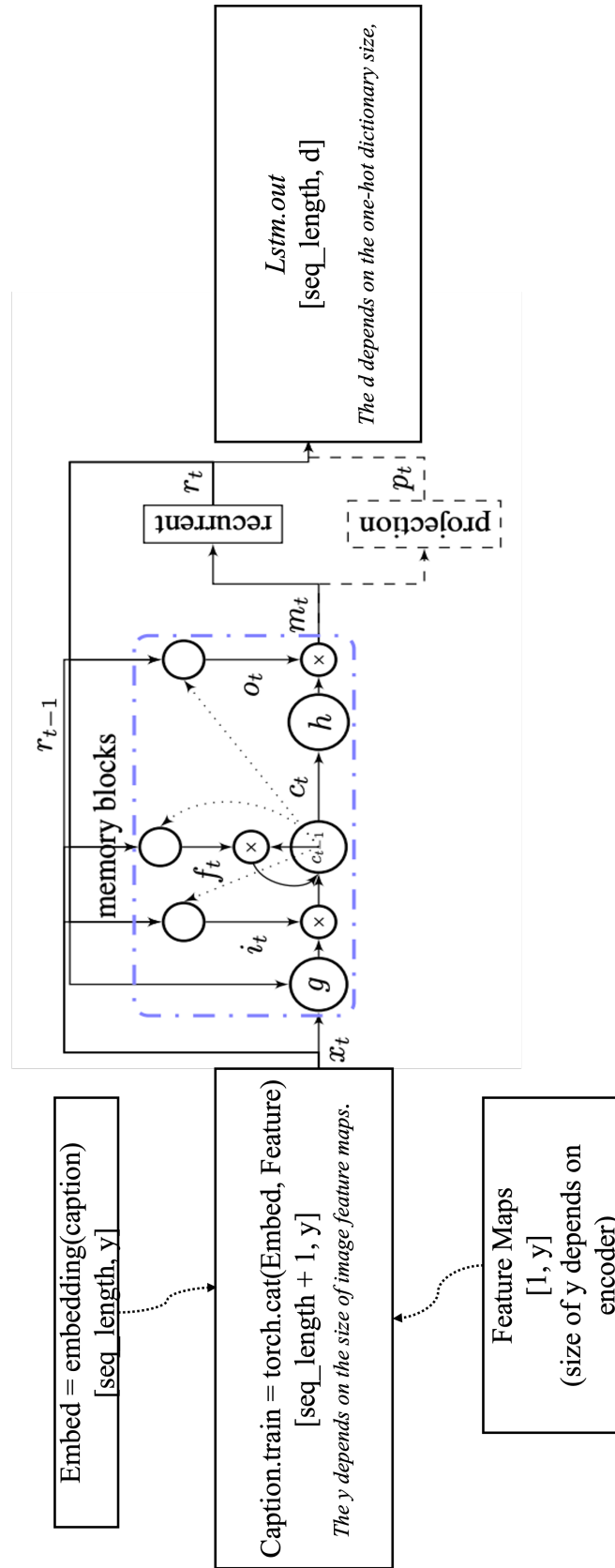


Figure 3.5: Decoder by LSTM. With the threshold of vocab setting as 5, there are 2,550 words in the Flickr 8K dataset vocabulary. The feature maps and embedded caption will be input into the LSTM network during training. The *LSTM.out* means the output of the decoder part.

---

picture corresponds to the output of the LSTM part. The vocab size determines the  $d$ . It will be used to get the loss with the embedded caption.

During the prediction, there should be a loop formed by the sequence length. The decoder part will predict the first matrix based on the feature map from the encoder to create a tensor of size  $[1, 2550]$  and a hidden state. The decoder will input the hidden state in the next cycle step to extend the word tensor. The final tensor contains all the probability of words in the vocab list. After that, the words with the highest probability will be chosen. At last, the loop will be stopped automatically once the " $\langle end \rangle$ " outputs.

### 3.5 Dataset and Experiments Settings

In the experiments, the similarity between predicted captions and ground truth and the training testing time was compared for our application.

Several well-known datasets are commonly used for image captioning experiments. All the datasets for image captioning consist of an image file and a text file that maps each image to one or more captions. Each caption is a sentence of words. Common Objects in Context (COCO) dataset [120] contains about 120 thousand images with five descriptions per image. There are also some smaller datasets, such as Flickr 8k dataset [1] with 8 thousand images, and the Flickr 30k dataset [121] with 30 thousand described images with five sentences for every image.

The dataset used in the experiments is the Flickr 8k, whose 8,000 images are in the ".jpg" format. There is also a text file with 40,000 captions in which five captions are provided for each image. The captions are set based on the diversity of human language. The 8,000 images are divided into three subsets: a training set with 6,000 images, a validation set with 1,000 images, and a testing set with the last 1,000 images.

Word embedding is one of the most popular representations of document vocabulary in the natural language processes. It can capture the context of a word in a document, semantic and syntactic similarity, relation with other words, etc. In other words, it is



used for extracting the features of the original sequences as the encoder. Similarly, the image feature extraction can be seen as image embedding. As the following part of the encoder is the natural language model, `embedding_size` is used to set the `input_size` of the LSTM part instead of feature size. Referring to the previous post, the hidden layer extracts high dimensional features for the words in the dataset.

In the end-to-end model by VGGNet and LSTM, input images are resized into  $224 \times 224 \times 3$ . The features are then extracted from the fully connected layer of VGG-19. The extracted  $1 \times 4096$  features are directly sent to LSTM with the 512 hidden dimensionality. The learning rate was  $10^{-4}$  for training this model.

In the end-to-end model by YOLO and LSTM, input images are resized into  $416 \times 416 \times 3$ . The features generated from the last convolutional layer of YOLO form a  $52 \times 52 \times 128$  vector. Through the newly added convolutional layers, the features are mapped into  $8 \times 8 \times 64$  so that the features from VGGNet and YOLO have the same size. The learning rate is initially  $10^{-4}$ . Other related hyperparameters are as shown in Table 3.1.

Table 3.1: Hyper Parameters Used in the Simulations

Model	learning algorithm	epochs	batch size	embedding size	hidden size
VGG-LSTM	Adam	200	1	4096	512
YOLO-LSTM	Adam	200	1	4096	512

The BLEU (Bilingual Evaluation Understudy) [122] score is commonly used for performance evaluation of natural language-related work. BLEU measures the closeness of translation by finding legitimate differences in the chosen words and their order between human translation and machine translation. Calculating BLEU involves counting the overlap of individual  $n$ -grams and obtaining a score by calculating the proportion of  $n$ -grams that are exact matches. With different  $n$ -grams settings, there are four BLEU scores from BLEU-1 to BLEU-4. The higher these scores are, the better the evaluated models' performance.

The BLEU-1 to BLEU-4 denotes unigram, bigram, trigram, and 4-gram, respec-

---

tively. The unigram (1-gram) is the degree of overlap of a single word. The bigram (2-gram) calculates the degree of overlap of two consecutive words. The trigram (3-gram) is the degree of overlap of three consecutive words. And the 4-gram (4-gram) is the degree of overlap of four successive words.

$$BLEU = BP \times \exp\left(\sum (1/n) \times \log(P_n)\right) \quad (3.1)$$

The Eq. 3.1 is the measurement method of BLEU, and BP is the factor used to penalize overly long or short-generated captions. The Brevity Penalty (BP) in the BLEU score is calculated to penalize the system’s translation if it generates shorter translations than the reference translations. The calculation involves comparing the length of the system’s translation ( $c$ ) to the length of the closest reference translation ( $r$ ). The formula of BP is shown as Eq. 3.2, where  $e$  is the base of the natural logarithm.

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \quad (3.2)$$

$P_n$  (*Precision at n-grams*) is the N-gram precision, representing the ratio of the number of n-grams in the generated caption to those in the reference caption. An *n-gram* is a contiguous sequence of  $n$  items (or words) from a given sample of text or speech. The  $n$  in *n-gram* represents the number of items in the sequence. By comparing *n-gram* matches between each candidate translation to the reference translations, BLEU tells whether the machine translation model is good. In our experiment, we set matches from one to four, commonly used in natural language model evaluation, to calculate the model performance.

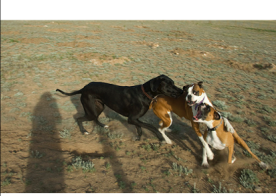



	<p><i>Ref:</i> Three dogs are playing in a field</p> <p><i>VGG:</i> a dog running on a beach (0.2892)</p> <p><i>YOLO:</i> two dogs running in the snow . (0.4953)</p>		<p><i>Ref:</i> A group of people gather around a large fountain.</p> <p><i>VGG:</i> a boy in a green shirt is walking through a fountain .(0.2526)</p> <p><i>YOLO:</i> a group of people are standing at a fountain in front of a white building. (0.6324)</p>
	<p><i>Ref:</i> A group of students in uniform stand in front of the gate .</p> <p><i>VGG:</i> &lt;start&gt; a group of people are standing on a &lt;unk&gt; road at night . &lt;end&gt; (0.2237)</p> <p><i>YOLO:</i> &lt;start&gt; a group of people are standing in a city street . &lt;end&gt; (0.5528)</p>		<p><i>Ref:</i> A girl in a blue swimsuit walks into the ocean .</p> <p><i>VGG:</i> &lt;start&gt; a man is standing in the ocean with a large stick in his hand . &lt;end&gt; (0.3686)</p> <p><i>YOLO:</i> &lt;start&gt; a young girl in a bikini is standing on the beach . &lt;end&gt; (0.6008)</p>

Figure 3.6: Some typical examples on the Flickr8k dataset. Humans generate the Ref sentences, and the other sentences are by the models of VGG and YOLO. The numbers after the predicted captions are the BLEU-4 scores.

### 3.6 Result Analysis with SOTA

Performance was compared among the two models and other state-of-the-art models on the Flickr8K dataset. Table 3.2 shows the BLEU-1 to BLEU-4 scores and testing times of different models on this dataset. The table shows that the models received higher scores from 1-gram to 4-gram than others.

As we can see from the table, the YOLO-LSTM architecture performs better than the VGG-LSTM model. Our YOLO-LSTM model performs 6.1% better in the BLEU-1, so we know that the YOLO-LSTM performs better predicting a single word. It means that the YOLO-LSTM is good at detecting the class of objects and background. What’s more, the BLEU-4 result of our YOLO-LSTM model is 9.2% better than the VGG-LSTM model. Figure 3.6 is some typical examples on the Flickr8K dataset. By comparing the captions of results from the VGG-LSTM and YOLO-LSTM, it could be found that the YOLO-LSTM model shows much better results on recognition of both main objects and the background in the image of larger size, such as the water and street. In addition, we can find that the YOLO-LSTM model is much better at measuring the relationship among the words. In other words, the YOLO-LSTM performs better on the positional relationship measurement among the objects and background.

The testing time of the VGG-LSTM and YOLO-LSTM models on NVIDIA GeForce RTX 3090 are shown in Table 3.2. We can see that the YOLO-LSTM model used a

Table 3.2: BLEU Score and Testing Time by Different Models on Flickr8k Dataset (BLEU scores and testing time are all in average.)

Model	1-gram	2-gram	3-gram	4-gram	Testing time
Neural Talk2 [5]	57.9	38.3	24.5	16.0	–
ATT-SVM + LSTM [119]	73	53	38	26	–
Multi-label CNN + Att-GRU [123]	72.9	53.4	41.2	30.7	–
AC-YOLO [124]	66.9	46.0	32.5	22.6	–
DLCT [96]	82.4	67.4	52.8	40.6	–
VGGNet (pre-trained) + LSTM	74.2	58.1	42.8	21.7	0.23s
VGGNet + LSTM (End-to-end)	79.8	63.3	46.6	34.2	0.23s
<b>YOLO + LSTM (Ours)</b>	<b>82.3</b>	<b>67.1</b>	<b>49.6</b>	<b>43.9</b>	<b>0.13s</b>

shorter time to test 1,000 images. According to the table, the testing time of YOLO-LSTM is 40% less than the VGG-LSTM for 1,000 images. As the original goal is to set a system for helping blind people see what is in front of them and live more comfortably in this world, the speed and accuracy of detection and the humanity of the natural language should be kept at a significant level. Based on the application of our system, the YOLO-LSTM model is a much better fit in the system to see and say what is in the image.

### 3.7 Conclusion

Based on some previous research, an image captioning model is proposed in this paper. The implemented model is tested and compared with the VGG-LSTM model on the Flickr 8k dataset. This paper proves that the YOLO-LSTM model fits the image captioning task. The result analysis shows that the YOLO-LSTM model performs much better than the VGG-LSTM and other existing models in a shorter prediction time. For the BLEU-1 score, the YOLO-LSTM received a 6.1% better performance; for the BLEU-4 score, the YOLO-LSTM received a 9.25% better performance. And for the predicting time, the YOLO-LSTM model is 40% faster than the VGG-LSTM model. Based on the research and experimental results, it is easy to say that the YOLO-LSTM

is a much better fit for applying the system to help blind people see the world with both accuracy and speed level.

In the future, we will continue exploring the issues discovered in this work. This chapter proved the effectiveness of reduced feature dimension, what size of reduced feature dimension can keep the performance with least computational cost is our continuous research point. Also, we endeavour to figure out a reasonable solution to building a model essentially generalizable to the complex real world.

## **Chapter 4**

# **Maintain a Better Balance Between Performance and Cost for Image Captioning by a Size-Adjustable Convolutional Module**

Image captioning is a challenging AI problem that connects computer vision and natural language processing. Many deep learning (DL) models have been proposed in the literature for solving this problem. So far, the primary concern of image captioning has been increasing the accuracy of generating human-style sentences for describing given images. As a result, state-of-the-art (SOTA) models are often too expensive to be implemented in computationally weak devices. In contrast, the primary concern of this paper is to maintain a balance between performance and cost. For this purpose, this chapter proposes using a DL model pre-trained for object detection to encode the given image so that features of various objects can be extracted simultaneously. We also propose adding a size-adjustable convolutional module (SACM) before decoding the features into sentences.

The experimental results show that the model with the adequately adjusted SACM could reach a BLEU-1 score of 82.3, a BLEU-4 score of 43.9 on the Flickr 8K dataset,

and a BLEU-1 score of 83.1 and a BLEU-4 score of 44.3 on the MS COCO dataset. With the SACM, the number of parameters is decreased to 108M, about 1/4 of the original YOLOv3-LSTM model with 430M parameters. Specifically, compared with mPLUG with 510M parameters, one of the SOTA methods, the proposed method can achieve almost identical BLEU-4 scores, but the number of parameters is 78% less than the mPLUG.

## 4.1 Introduction

There are a massive number of images appearing from different sources such as the internet, news, and advertisements. Unlike pictures in articles and TV programs, most images appear without captions in these sources. While most people have no difficulty understanding images without captions, visually impaired ones could face problems. Machine learning tools would help solve such problems by automatically interpreting images, videos, and other media.

Image captioning is a challenging AI problem that connects computer vision and natural language processing [112]. Many deep-learning (DL) models have been proposed for solving problems in both computer vision and natural language processing. The encoder-and-decoder architectures have been widely used for machine translation, transforming a sentence from one language to the target language. Such ideas have been applied to train a model with an image as input to generate captions based on a dictionary created from the given captions of the images by maximizing the probability of the correct words of the target sentence.

Besides natural language processing, image captions require object detection, recognition, location, properties, and interactions. Furthermore, generating human-style sentences requires a syntactic and semantic understanding of the language [31]. However, most proposed methods have not directly solved these problems in image captioning [112].

So far, the primary concern of image captioning has focused on increasing the ac-

---

---

curacy of generating human-style sentences for describing given images. As a result, state-of-the-art (SOTA) models are often too expensive to be implemented in computationally weak devices.

In contrast, the primary concern of this paper is to maintain a balance between performance and cost. For this purpose, we propose using a DL model pre-trained for object detection to encode the given image so that features of various objects can be extracted simultaneously. The Darknet, the model initially designed for object detection, has been used as the backbone to extract features of multiple objects in the image. We also propose adding a size-adjustable convolutional module (SACM) before decoding the features into sentences. The translated features from SACM have been used as input to a decoder implemented by long short-term memory (LSTM). The end-to-end image captioning system with Darknet, SACM, and LSTM is further trained simultaneously. After training, the system can automatically present an image and generate a descriptive caption in plain English.

The experimental results show that the system with a properly adjusted SACM could reach a BLEU-1 score of 82.3 and a BLEU-4 score of 43.9 on the Flickr 8K dataset, and a BLEU-1 score of 83.1 and a BLEU-4 score of 44.3 on the MS COCO dataset. The performance of our model with SACM is better than most of the existing models and comparable with that of the SOTA models. Our model size is much smaller than most SOTA models. With our proposed SACM, the number of parameters decreased to 108 M, about 1/4 of the original YOLOv3-LSTM model with 430 M parameters. At the same time, the proposed method can achieve almost identical BLEU-4 scores compared to the mPLUG, one of the SOTA methods, with a 78% smaller parameter size.

## 4.2 Outline

We arrange this chapter in the following order.

- Section [4.3](#): An overview of the proposed architecture.
- Section [4.4](#): Detailed discussion on the size-adjustable convolutional module.



- Section 4.5: Detailed discussion about the datasets and experiment settings.
- Section 4.6: Empirical analysis.
- Section 3.6: Discussion on results and comparison to state-of-the-art image captioning models.
- Section 4.9: Conclusion and possible future works.

### 4.3 Proposed Network Architecture

Captions can be generated from visual space and multimodal space, respectively, by novel image captioning methods. A general approach is to analyze the visual content of the image first and then generate image captions via the analysis of the visual content with a natural language model [32, 43, 59, 60]. Such methods can specifically generate captions with different lengths, styles, and relationships for each image. Therefore, these generated captions are semantically more accurate than previous methods. Most novel methods generate captions by analyzing information from visual space or multimodal space through DL.

Encoder–decoder approaches might be divided into convolutional neural networks (CNN), recurrent neural networks (RNNs), and transformer-based models. The CNN-RNN models use a CNN to encode images into vectorial representations. The vectors are adopted into an RNN-based decoder to analyze and provide a descriptive caption for the input image. For example, a special CNN used a novel method for batch normalization, while the output of the last hidden layer of CNN was used as an input to the LSTM decoder [31]. This LSTM decoder could keep track of the objects that had already been described using text. The CNN-RNN models are often trained in maximizing likelihood estimation.

To obtain a comprehensive understanding of objects and relationships in the images and generate fluent sentences to match the visual information, the encoder–decoder models often adopted the framework of CNN plus RNN image captioning model con-

---

figuration shown in Figure 4.1. Not only are they flexible, but they are also effective. Generally, global features are extracted from input images by a CNN model and then fed into an RNN model for sequence generation by transferring the image into a full grammatically and stylistically correct sentence. In some applications, a CNN was used for image representation, while an LSTM was used for caption generation. For example, the NIC (neural image caption generator) [31] and NIC V2 [111] followed such a framework. The output of the last hidden layer of CNN was used as input for the LSTM-based decoder.

In image captioning, image information was included in the initial state of LSTM. The end-to-end models use an RNN, which encodes the variable length input into a fixed dimensional vector. They then use the decoded vector to generate it into the desired output sentence. Therefore, it is natural to use the same approach to image captioning rather than inputting a sentence to translate it into a description.

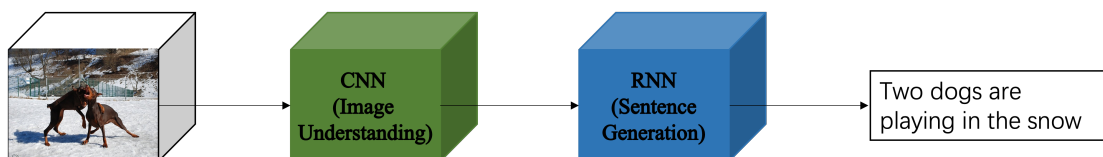


Figure 4.1: Architecture of CNN plus RNN model. The CNN encoder extracts image features, while the RNN decoder generates text descriptions by analyzing the features.

Compared with the transformer-based model, traditional CNNs have much fewer parameters. Faster R-CNN with ResNet101 was used as a feature extractor to generate image captions [42]. It proved the effectiveness of the object detection model as the encoder for the image captioning tasks. Most existing object detection methods, like DMP [125], R-CNN [71], and Faster R-CNN [70], made good use of classifiers for performing detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image.

Unlike two-stage models, YOLOv2 used Darknet-19 [115] as a feature extractor. YOLOv3 uses the Darknet-53 [126] network as a backbone with 53 convolutional layers. The experimental results proved that Darknet-53 was better than SOTA for having fewer floating point operations and more speed while maintaining similar accu-

racy [126]. Darknet-53 is better than ResNet-101 and 1.5 times faster than ResNet-101 as well. Darknet-53 has a similar performance to ResNet-152 but is two times faster. Darknet-53 also achieved the highest measured floating point operations per second. This means the network structure could better utilize the GPU and be more efficient and faster. Because ResNets have too many layers with less efficiency, Darknet-53 is selected as the backbone of our proposed image captioning system.

LSTM is used as the decoder in our proposed model. During the pre-processing, the captions will be filled with the "unk" for marking unknown words, the "start" for marking the start of a new sentence, and the "end" for indicating the end of the ground truth sentences. The one-hot encoding method is used in the experiment for training and predicting our implementation. A dictionary containing both words and their corresponding IDs will be set. With these processes, a dictionary of size  $D$  summarises all the different words corresponding to IDs in the dataset. The LSTM model is trained to predict each word of the target sentence after presenting the image and preceding words. During the decoding processing, the output of the LSTM at time  $t - 1$  is fed to the LSTM at time  $t$ . The unrolled version transforms all the recurrent connections into feed-forward connections, specifically if  $I$  is denoted as the input image.  $S = (S_0, \dots, S_N)$  is set as the target sentence with  $N + 1$  words. The unrolling procedure is as follows:

$$x_{-1} = \text{encoder}(I), m_{-1} = \text{None} \quad (4.1)$$

$$(s_t, m_t) = \text{LSTM}(x_{t-1}, m_{t-1}), t = 0, 1, \dots, N$$

$$s_t = \text{Linear}(s_t) \quad (4.2)$$

$$j_0 = \text{argmax}_t^j, j = 1, 2, \dots, D \quad (4.3)$$

$$S_t = s_t^{j_0} \quad (4.4)$$

$$x_t = W_e(S_t), t = 0, 1, \dots, N \quad (4.5)$$

The encoded features,  $x_{-1}$ , of image  $I$  are only input into LSTM at time  $t = -1$ .  $m_{-1}$  is set as none to inform LSTM about the boundary. From  $t = 0$  to  $t = N$ ,  $s_t$  is

---

the vector of the linear likelihood at time  $t$  of all words in the collected dictionary of size  $D$ .  $m_t$  is the memory at time  $t$ . At  $t = 0$ ,  $s_0$  and  $m_0$  are generated by LSTM with the encoded features and the boundary as input. From  $t = 1$ ,  $s_t$  and  $m_t$  are received with the information at the last step. The index  $j_0$  of the word that received the highest probability in  $s_t$  is indicated with the *argmax* function. Finally, the predicted word at time  $t$ ,  $S_t$ , is output from the dictionary. After prediction at time  $t$ , the predicted word is embedded by the word-embedding function  $W_e$  [127]. Word embeddings represent a word's semantics by efficiently encoding semantic information that might be relevant to the task at hand. From  $t = 0$ , the embedded vector  $x_t$  will be input into the LSTM with the memory at time  $t$  together. With such  $N$  words, the sentence  $S = (S_0, \dots, S_N)$  is generated.

While accuracy is important in image captioning, speed should also be considered, especially for mobile device-based real-time applications. By maintaining accuracy and achieving more stability with the reduced feature dimension, the processing time will be expected to decrease. For an image caption generator, the parameter size is related to the parameters of both the encoder and the decoder. The parameter size of an LSTM model can be calculated as follows:

$$\begin{aligned}
 P_S &= 4 \times (input\_size + hidden\_size) \times hidden\_size \\
 &+ 4 \times hidden\_size \times hidden\_size \times (num\_layers - 1) \\
 &+ output\_size \times (hidden\_size + 1)
 \end{aligned} \tag{4.6}$$

where *input\_size* is the size of the input vector, *hidden\_size* is the number of LSTM units in the hidden state, *num\_layers* is the number of LSTM layers, and *output\_size* is the size of the output vector at each time step. The factor 4 in the equation comes from the fact that LSTM has four gates, including an input gate, a forget gate, an output gate, and a cell gate.

From Equation (4.6), the number of parameters in an LSTM model depends on its input, hidden, and output sizes. If the input size is halved while the other data sizes remain the same, the weight matrix from the input layer to the hidden layer will have

half as many rows with the same number of columns that define the hidden size. This will result in the weight matrix with half as many elements by reducing the parameter size by approximately 1/4 of the original size. Therefore, the parameter size of an LSTM model would be reduced by about one-fourth of its original size if its input size were halved.

$$P_C = (C_W \times C_H \times C_I + 1) \times C_O \quad (4.7)$$

$$P_A = 3 \times E_S^2 \times N_H \quad (4.8)$$

Eq. 4.7 and 4.8 are the parameter size calculations of the convolutional layer and self-attention layer. In Eq. 4.7,  $P_C$  denotes the parameter size of the convolutional layer.  $C_W$  and  $C_H$  denote convolutional kernel width and height, respectively.  $C_I$  and  $C_O$  are input and output channels.  $P_A$  is the parameter size of the attention layer in Eq. 4.8.  $E_S$  and  $N_H$  correspond to the embedding size and number of heads in the attention layer.

In a self-attention layer, each position is required to calculate its attention with all other positions, which results in a relatively more significant number of parameters. This allows the self-attention layer to model global relationships and is suitable for handling long-range dependencies within sequences.

Conversely, only a local receptive field is considered in a convolutional layer, typically leading to fewer parameters. The size of the convolutional kernel determines the local patterns or features that can be captured. This makes convolutional layers suitable for capturing local structures and features within the input data. Suppose we can use the convolutional layer with a small-size kernel to help focus on the more critical information and remove other details. In that case, the parameter size will not increase so much. In this way, we can reduce the input of the decoder part to reduce the parameter size of the whole model.

---

## 4.4 Size-Adjustable Convolutional Module (SACM)

The final features from Darknet-53 are input to SACM for further feature extraction and dimension reduction without losing important information. The original Darknet-53 uses a residual network to generate residual blocks of different sizes. For corresponding blocks of various sizes, several convolutional layers and upsampling processes are designed to analyze the features of items in different sizes and to jump link with the residual blocks inside Darknet-53 to alleviate the gradient disappearance problem brought about by increasing depth in deep neural networks.

The following convolutional layers focus on detecting and localizing targets. These convolutional layers convert the feature maps into predicted feature maps at different scales to obtain information indicating the presence or absence of targets in a given region and the location and class of targets. The feature pyramid network (FPN) is applied to YOLOv3 to fuse the features at different levels. The upsampling layer can upsample the low-resolution feature map to the same size as the high-resolution feature map. This way, the semantic information from the shallower layers can be fused with the detailed information from the deeper layers by the upsampling operation. YOLOv3 designed this part for faster object detection, and we retained this part for global and local features.

Generally speaking, a larger feature map can provide richer spatial contextual information, and the model can better understand the relationship between the target and its surroundings. Nevertheless, for a mobile-device-oriented model, real-time detection is another important goal. According to Equation 4.6, the decode (i.e., the LSTM) has fewer parameters if the feature map is smaller. In other words, the smaller the feature map is, the lower the cost of predicting time and computational sources.

One of the primary considerations in this paper is to keep the performance while reducing the computation costs with a smaller-size feature map. For this purpose, we propose to insert a SACM between the encoder and the decoder. SACM is a size-adjustable convolutional module that consists of several convolutional layers for feature extraction and a few additional convolutional layers for dimension reduction. Increasing and de-

creasing the convolutional layers for dimension reduction can maintain the balance of performance and cost. The structure of the Darknet-SACM-LSTM model is shown in Figure 4.2.

Convolutional layers with a  $2 \times 2$  convolution kernel are applied in SACM for dimension reduction. Incorporating the convolutional layers is a way to form the original feature through  $2 \times 2$  filters or  $1 \times 1$  with non-linearity injection. With the  $2 \times 2$  convolution kernel, each output pixel of the layer is affected by only one pixel in a  $2 \times 2$  region of the input image after the convolution operation. Firstly, the parameter size will not increase so much with the small-size convolution kernel. For example, when a  $2 \times 2$  convolutional stack with  $C_i$  input channels and  $C_o$  output channels is set, the stack is parameterized by  $2^2 \times C_i \times C_o = 4C_i \times C_o$  weights. A convolutional layer with a  $1 \times 1$  convolution kernel is equivalent to a cross-channel parametric pooling layer [128]. When the output channel is smaller than the input channel, the convolutional layer can also be used for dimension reduction. Being compared to the pooling layer, the  $1 \times 1$  convolutional layer is a way to reduce the dimension without affecting the receptive fields of the convolutional layers. For the balance between the final output size and the performance, experiments of SACM with different convolutional layers with  $2 \times 2$  or  $1 \times 1$  kernels are set in the simulations. The settings of SACM are shown in Table 4.1. In addition, the parameters of all the size-adjusting layers do not increase so much. With the settings in the table, the module with three layers has the largest number of parameters, 1.05M. In other words, the parameter size of the whole structure decreases to nearly 1/8 of the original size after introducing these extra 1.05M parameters.

After processing by the first five convolutional layers, the size of the feature maps becomes  $52 \times 52 \times 128$ . The following adjustable convolutional layers can reduce feature dimension directly for sending to LSTM for caption generation. The SACM performs as a pipeline connecting the encoder and decoder to reduce feature dimensions, saving time and computational costs. After passing through SACM, the dimension-reduced feature maps go through to LSTM to generate captions for the provided images.

In this chapter, experiments were conducted to measure the relationship between

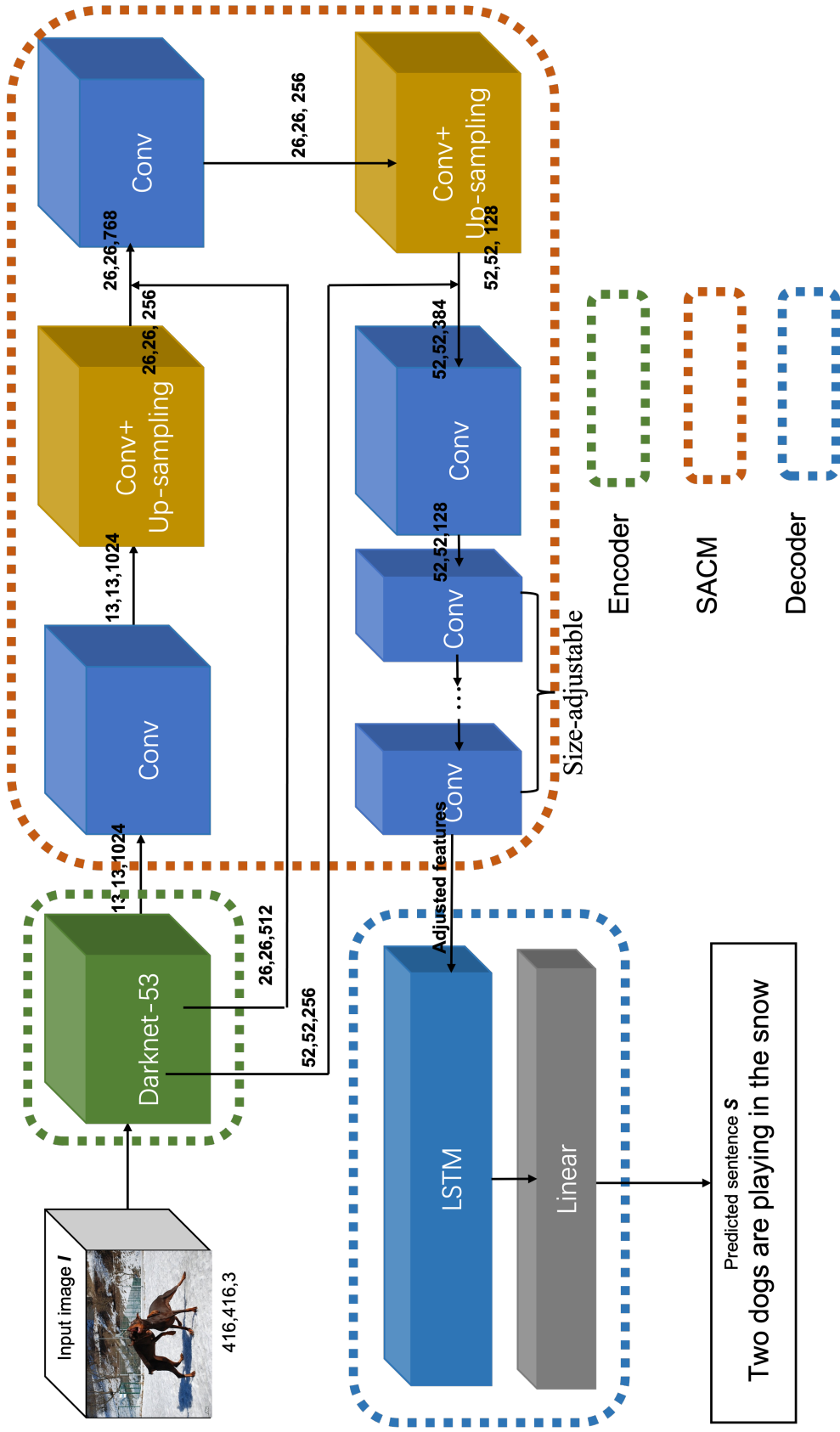


Figure 4.2: Detailed architecture. The green dotted box includes the backbone, the following convolutional layers construct the SACM, and the blue dotted box contains the decoder.



Table 4.1: Settings of SACM with the original  $128 \times 52 \times 52$  feature size. conv  $k_x \times k_y \times c$  is a convolution of kernel size  $k_x \times k_y$  with  $c$  outputs channels. The last line is the final output  $S_f$  size from SACM.

<b>SACM-A</b>	<b>SACM-B</b>	<b>SACM-C</b>	<b>SACM-D</b>	<b>SACM-E</b>
		input ( $128 \times 52 \times 52$ )		
conv2 $\times 2 \times 256$	conv2 $\times 2 \times 128$	conv2 $\times 2 \times 256$	conv2 $\times 2 \times 128$	conv2 $\times 2 \times 128$
		conv2 $\times 2 \times 256$	conv2 $\times 2 \times 128$	conv2 $\times 2 \times 128$
				conv1 $\times 1 \times 64$
$S_f: 256 \times 26 \times 26$	$S_f: 128 \times 26 \times 26$	$S_f: 256 \times 13 \times 13$	$S_f: 128 \times 13 \times 13$	$S_f: 64 \times 13 \times 13$

---

the feature size and balance of performance and speed. The final feature size, denoted as  $S_f$ , undergoes a reduction from half (1/2) of the original size ( $S_f = 256 \times 26 \times 26$ ) to a much smaller size of 1/32 of the original ( $S_f = 64 \times 13 \times 13$ ). This reduction is achieved by applying  $2 \times 2$  and  $1 \times 1$  convolutional layers. Our experiments also trained the SACM with the encoder and the decoder together.

With the trainable convolutional encoder of Darknet-53, the training process can be conducted by simultaneously training the encoder, SACM, and decoder with the *data, ground truth* pairs without fixing the encoder. Therefore, the proposed model's Darknet-53, SACM, and LSTM parameters are updated to find the features more useful for learning. The training processing is shown in Figure [4.3](#).

Since different people may give other descriptions of the same image, in a general image captioning dataset, each image usually has multiple captions corresponding to it. The Flickr 8K and MS COCO dataset contains five different ground truth captions for each image. Multiple annotations can provide more information and diversity to help the model learn different description styles, have different lexical usages, and learn different semantic expressions. Such a multi-labelling approach helps the model to better adapt to additional input images and generate diverse and higher-quality descriptions during testing. During training, each caption is set with the image, which it describes as a pair of input and ground truth. In other words, every image is input to the model five times with different captions. For example, the training set of the Flickr 8K dataset contains 6,000 images, so there are 30,000 pairs of input and ground truth in the training set.

Word2Index is the word-embedding structure used in this paper to map captions to vectors. At first, the structure collects all the unique words in the dataset to set a vocabulary. Equation [4.6](#) mentions that the vocabulary scale influences the parameter size. Large-scale vocabulary will increase storage and computational costs. Moreover, it is difficult for the model to obtain enough information from rare words that appear only once or twice and will also affect the model's prediction of high-frequency words. So, we set thresholds in our experiments at 5 to avoid the effect of rare words on the train-

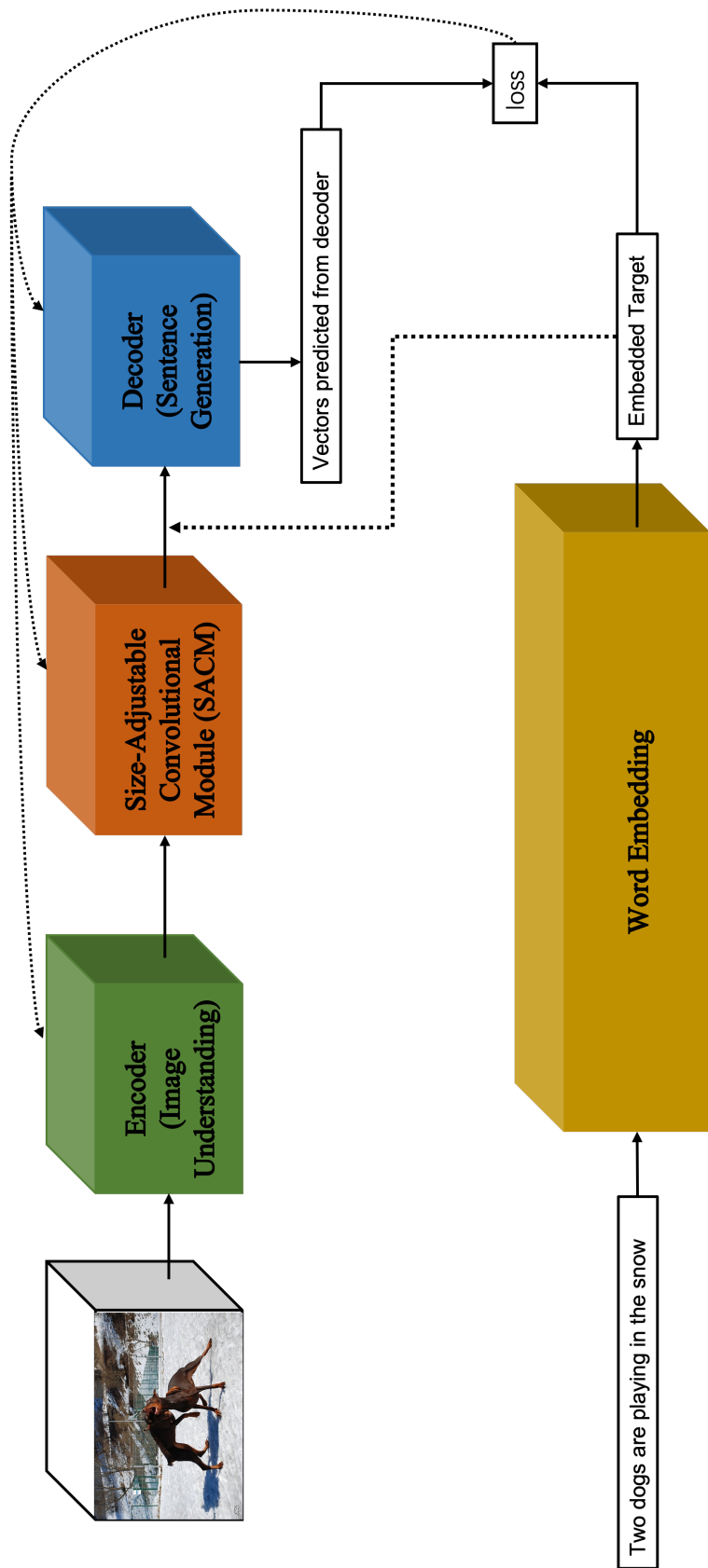


Figure 4.3: The training process of the proposed model includes encoder, SACM, and decoder. The embedded target is input into the decoder only during training. This model is trained end-to-end. It means the optimizer can update all related parameters in the encoder, SACM, and decoder based on the loss function.

---

ing effect of the model. Then, the structure maps each word to its unique corresponding index, which is set from 0, to construct the vocabulary for the dataset. The vector of size (sequence.length, index) can map the ground truth into a vector. The value of the corresponding index of the target word is 1, and the others are 0. In addition, the words not in the vocabulary will be instead of  $\langle unk \rangle$ .

During the prediction process of image captioning, the model generates a probability distribution at each time step. The word with the highest probability in the distribution is selected as the current output. The prediction continues until the termination marker is encountered or the maximum generation length is reached. Finally, the generated words are combined to form the final prediction result. Unlike the training process, the model selects only one best sentence as the final rendered image caption. To evaluate the model's performance, the evaluation metrics calculate the similarity between the generated subtitles and each ground truth to derive a composite score, thereby mitigating the effect of subjectivity on the evaluation results.

## 4.5 Dataset and Experiment Setting

The Flickr 8K dataset and MS COCO dataset were used in the experiments. Flickr 8K [56] is a popular dataset with 8000 images collected from Flickr. The training data comprises 6,000 images, while the test and evaluation data comprise 1,000 images separately. Each image in the dataset has five reference captions annotated by humans. The MS COCO dataset is extensive for image recognition, segmentation, and captioning. The dataset has more than 300,000 images and more than 2 million instances with 80 object categories and five captions per image. Many image captioning methods have been tested on these two datasets. To compare our model's performance on the MS COCO dataset with other results, the fixed training data used 118,287 images, while the evaluation and testing sets included 5,000 images, respectively.

The end-to-end VGG-LSTM model was used as a baseline to compare performance and speed with the end-to-end Darknet-LSTM model. In the VGG-LSTM experiment,

a pre-trained VGGNet-19 model is used as a feature extractor. To fit the pre-trained model, input images are transformed into  $224 \times 224 \times 3$ . As an end-to-end model, the model is fine-tuned for Flickr 8K and MS COCO datasets. The Adam optimizer was used with a base learning rate of  $10^{-5}$  for both datasets' models. The dimension of feature maps is 4096, while the dimension of hidden layers of LSTM is 512 in the VGG-LSTM model. The VGG-LSTM model is trained by minimizing the cross-entropy loss. The Adam optimizer with the same learning rate was applied to the end-to-end Darknet-LSTM model. The input of the original Darknet feature extractor is required to be  $416 \times 416 \times 3$ . The batch size is 1 for Flickr 8K and 50 for MS COCO datasets. The maximum epoch was set to 30. The model with the highest BLEU scores on evaluation data was used for testing. In addition, we also set experiments with the original Darknet as an encoder to compare the performance with our proposed model.

All experiments were run on a computer environment under Ubuntu 20.04, AMD Ryzen 9-3900X CPU with 32GB RAM, and GTX 3090 GPU with 24G memory. Pytorch was used for the deep learning framework. Following the previous research, the rule of captions with at most 20 words was set for both datasets. The specific vocabulary of words was built by particularly removing words that occurred fewer than five times. A vocabulary of 2550 words was created for Flickr 8K, while a vocabulary of 10,321 words was built for MS COCO.

The cross-entropy loss was measured throughout the whole training process. If the dictionary is of size  $D$ , the equation of the cross-entropy loss between the predicted word and the target word at time  $t$  is as follows:

$$Loss_t = - \sum_{j=1}^D T_{t,j} \log(s_{t,j}) \quad (4.9)$$

and the average loss of the sequence of length  $N$  is as follows:

$$Loss = \frac{1}{N} \sum_{t=1}^N Loss_t \quad (4.10)$$

where  $T_t$  is the ground truth of the given word at time  $t$ .  $T_{t,j}$  indicates the probability

---

---

of the  $j$ -th word in the dictionary at the current time step. For example, if the target word at time  $t$  is the 7th word in the dictionary,  $T_{t,7}$  is 1, and others are 0.  $s_{t,j}$  denotes the probability of the model predicting the  $j$ -th word at time  $t$ . The average loss  $Loss$  of the predicted sequence length  $N$  is calculated with the average function. The  $N$  is the same as the target sequence during the training process, while the  $N$  will be fixed in the prediction process. The measured losses in the experiments are the average of all cross-entropy losses between the prediction and the target captions.

Some evaluation metrics from machine translation were used in evaluations, including BLEU [122], METEOR [129], ROUGE [130], and CIDEr [131]. BLEU is used to analyze the co-occurrence of n-grams between the predicted captions and ground truth. The n-gram is often used to reflect the precision of the generated captions [122]. It compares a text segment with a set of references to compute a score correlating with a human’s quality judgement. The semantic propositional image caption evaluation METEOR is calculated based on the weighted harmonic average of single-word recall and precision [129], which can offset the shortcomings of BLEU. It also adds a word-net-based measurement to address issues of synonym matching. ROUGE [130] compares the generated word sequence and word pairs with reference descriptions. There are several different ROUGEs, such as ROUGE-L and ROUGE-N. The most widely used ROUGE-L, in which the longest identical fragment in the generated and ground-truth sentences is defined as the longest common sub-sequence, is selected as one of the evaluation metrics in the experiments. CIDEr [131] is an automatic caption evaluation metric based on consensus. It treats the sentence as a document and uses TF-IDF to calculate the weight of words. The consistency of the generated caption with the reference caption is measured by the cosine distance between the TF-IDF vector representations of two sentences.

## 4.6 Experimental Analysis

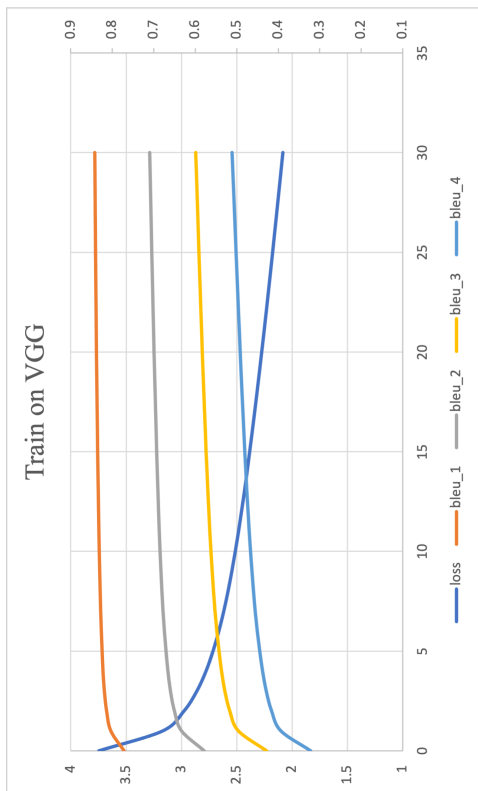
### 4.6.1 Experimental Results on Flickr 8K

VGGNet is used as an encoder of the baseline model in this paper. After removing the classifier, softmax, and last fully connected layer, the size of the feature maps is 4096. The changes of both loss values and BLEU scores from 1-gram to 4-gram on both training data (left) and evaluation data (right) are shown in Figure 4.4a and Figure 4.4b. Each figure’s horizontal ( $x$ ) axis represents the number of learning epochs. The left vertical ( $y$ ) axis represents the loss values, while the right vertical axis shows the values of BLEU scores. Although the training loss dropped throughout the training process, the evaluation loss slightly increased after 20 learning epochs. As expected, the BLEU scores were lower on the evaluation data than those obtained on the training data.

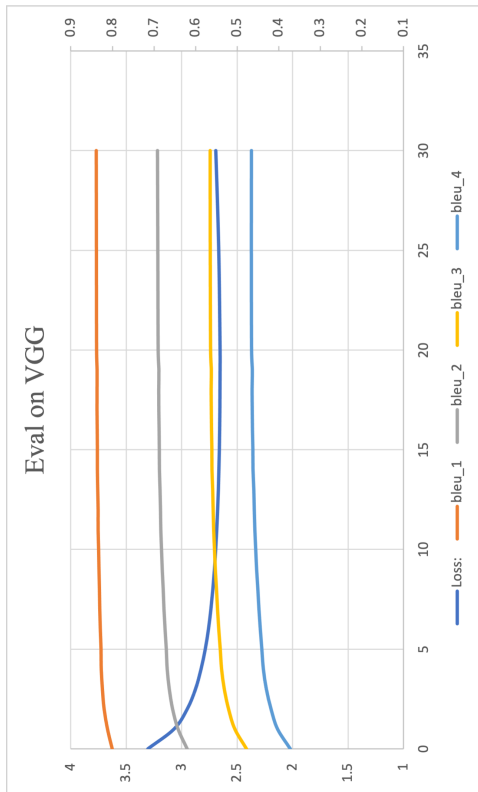
Because of the limited memory in our computer environment, the experiments on SACM feature selection are set from 1/2 ( $S_f = 256 \times 26 \times 26$ ) of the original size to 1/32 ( $S_f = 64 \times 13 \times 13$ ) of the original size. The training and predicting time cost of SACM with different feature sizes are shown in Table 4.2. For performance comparison, BLEU-1 and BLEU-4 scores and the cross-entropy loss by SACM on the testing set are also given in Table 4.2. The results suggest that SACM with  $S_f = 128 \times 13 \times 13$  features received the highest BLEU scores with a similar prediction speed to the baseline model of VGG-LSTM.

The results show that the model with the highest BLEU-1 score of 82.3% used  $S_f = 128 \times 13 \times 13$  features. Its BLEU-4 score is 0.439, the same as the model using  $S_f = 256 \times 26 \times 26$  features but higher than others. On testing 1000 images, the model with  $S_f = 128 \times 13 \times 13$  features used 3.9 min ran 15 min faster than the model with  $S_f = 256 \times 26 \times 26$  features but 1 min slower than the model using  $S_f = 64 \times 13 \times 13$  features. The baseline model could neither match the performance of the implemented models nor run faster than the model using  $S_f = 64 \times 13 \times 13$  features.

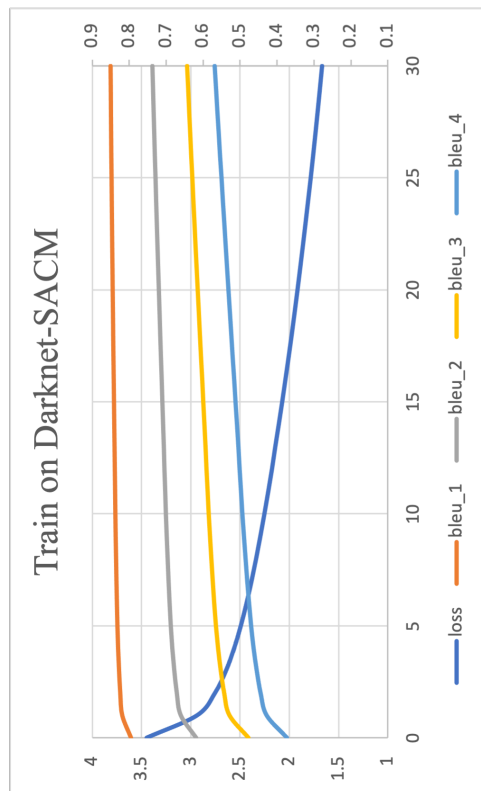
By comparing charts in Figure 4.4, it could be seen that SACM with  $S_f = 64 \times 13 \times 13$  reached a training loss of 1.7 lower than the training loss of 2.1 by VGG16-LSTM.



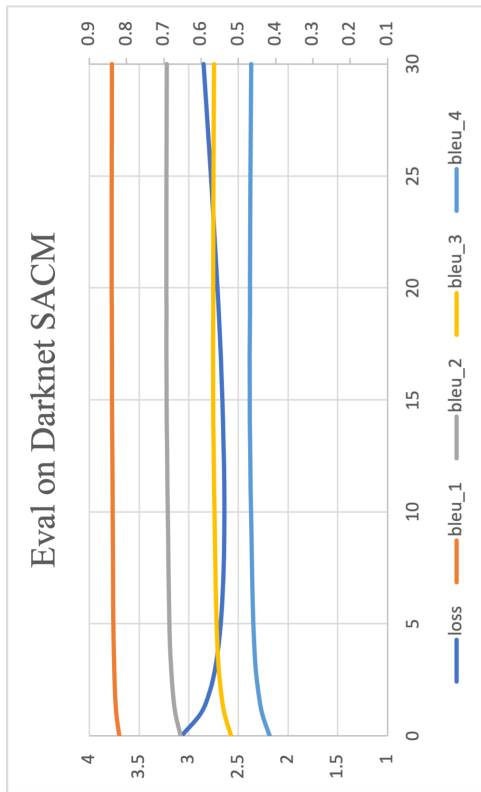
(a) Training on VGG16-LSTM.



(b) Evaluation on VGG16-LSTM.



(c) Training on Darknet-LSTM.



(d) Evaluation on Darknet-LSTM.

Figure 4.4: Loss values and four BLEU scores on training (left) and evaluation (right) By the end-to-end model with VGGNet as the encoder and LSTM as the decoder on the Flickr 8K dataset. The size of the final feature map is set as  $S_f = 4096$  after removing the softmax and the last fully connected layer. Loss values and four BLEU scores on training (left) and evaluation (right) by Darknet-LSTM with  $S_f = 128 \times 13 \times 13 = 21, 632$  on the Flickr 8K dataset.



Table 4.2: The cost and performance of SACM with different feature sizes on the testing set for Flickr 8K. B@1 and B@4 denote BLEU-1 and BLEU-4 scores. “No.” denotes the number of learning epochs when the models received the best BLEU-4 score on the evaluation data.

Final feature size $S_f$	Predicting time (on CPU)	Predicting time (on GPU)	B@1	B@4	loss	No.
VGG-LSTM (baseline)	2.6s	0.2s	0.798	0.342	2.66	10
$128 \times 52 \times 52$	N/A	N/A	-	-	-	-
$256 \times 26 \times 26$	5.5s	1.1s	0.822	0.439	2.61	19
$128 \times 26 \times 26$	4.6s	0.6s	0.817	0.428	2.65	19
$256 \times 13 \times 13$	3.6s	0.4s	0.790	0.419	2.68	17
<b><math>128 \times 13 \times 13</math></b>	3.5s	<b>0.2s</b>	<b>0.823</b>	<b>0.439</b>	<b>2.63</b>	<b>17</b>
$64 \times 13 \times 13$	<b>3.2s</b>	<b>0.2s</b>	0.809	0.431	2.64	19
darknet-LSTM	4.5s	0.9s	0.764	0.369	2.74	15

---

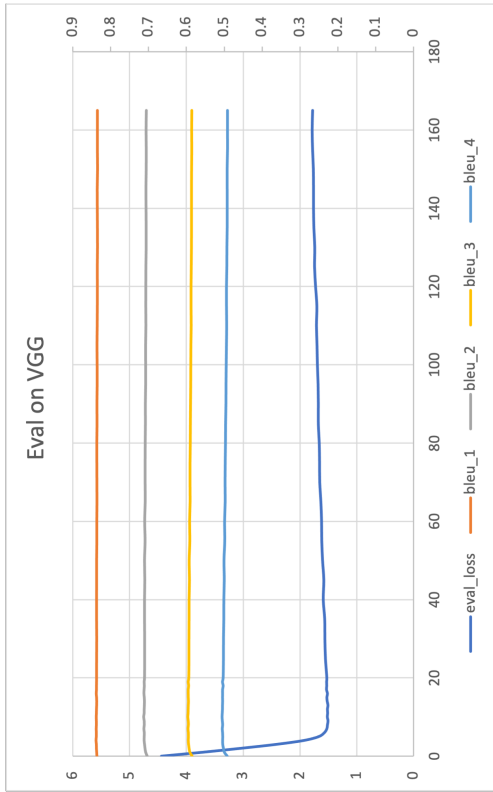
However, SACM could lead to overfitting on small data sets such as Flickr 8K. Using the evaluation loss to decide on the final learned model for solving the problems with fewer samples would be important. On Flickr 8K, all the models reached their highest BLEU-4 scores around the 20th training epoch.

## 4.6.2 Experimental Results on MS COCO

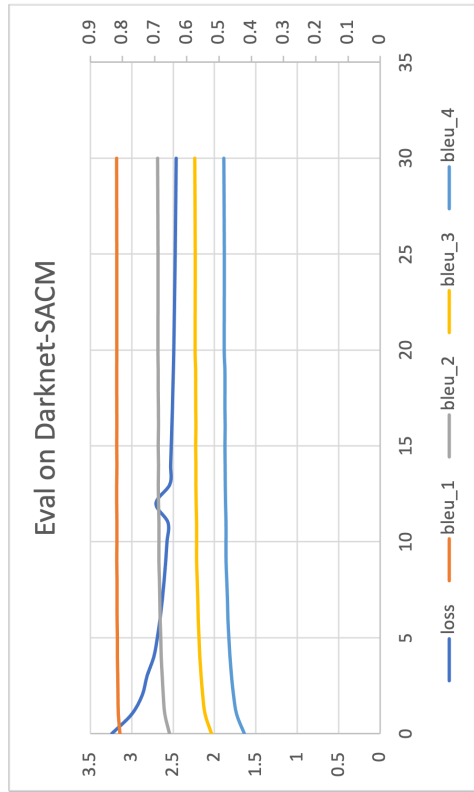
The training and evaluation results of the baseline encoder VGGNet on the MS COCO dataset were given in Figure 4.5a and Figure 4.5b. It can be seen that both the training loss and the evaluation loss were decreasing from the first epoch until the end. The loss values and BLEU scores changed more in the first 20 epochs. Unlike the Flickr 8K results, the best SACMs of different feature sizes on the MS COCO dataset were all received on the 30th epoch. That is, no overfitting appeared on the MS COCO dataset. Because the dictionary size for the MS COCO dataset is nearly five times larger than the Flickr 8K dataset, only the models using fewer features were tested. We also added the performance and time cost of Darknet-LSTM to the table for comparison.

The changes in the loss and BLEU scores on Darknet-SACMs with features of  $S_f = 128 \times 13 \times 13$  on both the training and evaluation sets for MS COCO are shown in Figure 4.5c and Figure 4.5d. For performance comparison, the performance of different SACMs on the MS COCO dataset is shown in Table 4.3. The  $B@1$ ,  $B@4$  and  $No.$  are the BLEU-1 and BLEU-4 scores and the epoch number of the model that received the best BLEU-4 score on the evaluation data. Similar results were obtained on MS COCO. SACM with features of  $S_f = 128 \times 13 \times 13$  outperformed on both BLEU-1 and BLEU-4 scores. Its BLEU-4 score is 0.443, which is higher than the others. The baseline model had the lowest BLUE scores. As for the running time, the time rose from 14.5 min to 31.2 min when the final feature size increased from  $S_f = 64 \times 13 \times 13$  to  $S_f = 256 \times 13 \times 13$  in SACM.

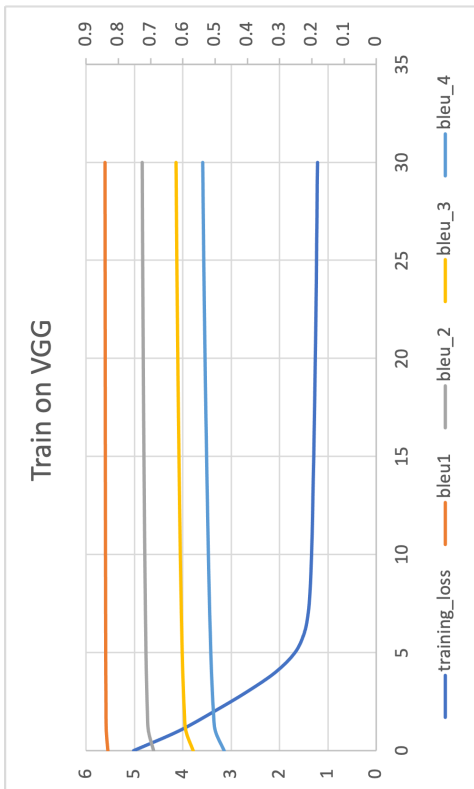
The changes in the loss and BLEU scores on Darknet-SACMs with features of  $S_f = 128 \times 13 \times 13$  on both the training and evaluation sets for MS COCO are shown in Figure 4.5c and Figure 4.5d. It is interesting to see that although the learned



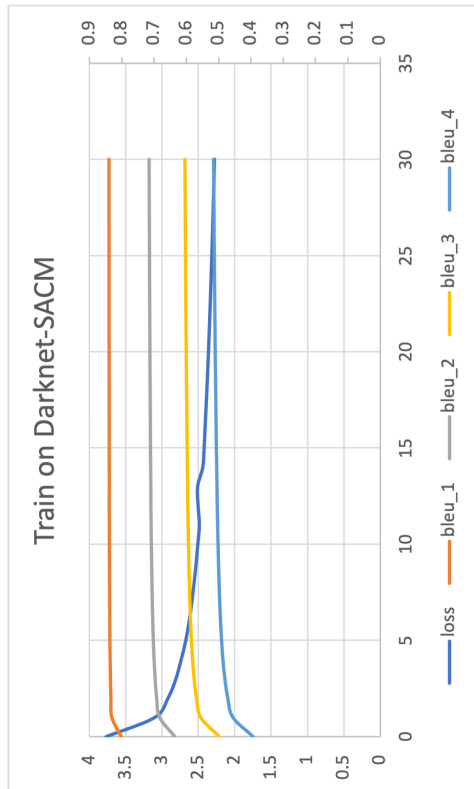
(b) Evaluation on VGG16-LSTM.



(d) Evaluation on Darknet-LSTM.



(a) Training on VGG16-LSTM.



(c) Training on Darknet-LSTM.

Figure 4.5: Loss values and four BLEU scores on training (left) and evaluation (right) by the end-to-end model with VGGNet as the encoder and LSTM as the decoder on the MS COCO dataset. The size of the final feature map is set as  $S_f = 4096$  after removing the softmax and the last fully connected layer. Loss values and four BLEU scores on training (left) and evaluation (right) by Darknet-LSTM with  $S_f = 128 \times 13 \times 13 = 21,632$  on the MS COCO dataset.

Table 4.3: The cost and performance of SACM with different feature sizes on the testing set for MS COCO. B@1 and B@4 denote BLEU-1 and BLEU-4 scores. “No.” denotes the number of learning epochs when the models received the best BLEU-4 score on the evaluation data.

Final feature size $S_f$	Prediction time (on CPU)	Predicting time (on GPU)	B@1	B@4	Loss	No.
VGG-LSTM(baseline)	2.6s	0.2s	0.778	0.376	2.51	10
$128 \times 52 \times 52$	N/A	N/A	-	-	-	-
$256 \times 26 \times 26$	N/A	N/A	-	-	-	-
$128 \times 26 \times 26$	N/A	N/A	-	-	-	-
$256 \times 13 \times 13$	3.8s	0.4s	0.828	0.434	2.63	30
<b><math>128 \times 13 \times 13</math></b>	3.5s	0.3s	<b>0.831</b>	<b>0.443</b>	<b>2.65</b>	<b>30</b>
$64 \times 13 \times 13$	<b>3.2s</b>	<b>0.2s</b>	0.820	0.438	2.63	30
darknet-LSTM	N/A	N/A	-	-	-	-

Darknet-SACMs had higher loss values on both the training and the evaluation set than the learned VGG-LSTM, the learned Darknet-SACMs were able to have higher BLEU scores. This indicates that the lower entropy-loss values might not necessarily lead to better captions.

## 4.7 Result Comparison with SOTA

The experiments in this paper proved the effectiveness of our proposed model and succeeded in dimension optimization. The encoder of the best model is set with Darknet as the backbone, and its output feature is  $128 \times 13 \times 13$ . To showcase our superiority, we compare state-of-the-art results on both the Flickr and MS COCO datasets in Tables [4.4](#) and [4.5](#), respectively. The best scores for each metric are highlighted in bold, and we also include the number of parameters used for prediction to compare prediction speeds.

As we can see from the table, our model shows competitive performance. Specifically, it outperforms the original CNN+RNN-based methods, indicating that our critical designs are effective for image captioning tasks. In addition, our model’s performance is better than many large-scale models. This suggests that further research on the encoder–decoder architecture could inspire new efforts in this area.

Table 4.4: Comparisons of our proposed Darknet-LSTM with SACM and some SOTA methods on Flickr 8K dataset. B@1, B@4, M, R, C, and P denote BLEU@1, BLEU@4, METEOR, ROUGE-L, CIDEr, and the model sizes.

Method	B@1	B@4	M	R	C	P
Neuraltalk2 [5]	57.9	16.0	-	-	-	31 M
D-CNN [132]	49.5	20.1	42.5	-	-	-
VGG16-LSTM [133]	62.6	28.7	-	-	-	138 M
Hard-attention [32]	66.0	31.4	24.8	50.3	68.9	149 M
Neural Baby Talk [134]	66.4	32.6	26.2	52.5	84.5	37.7 M
m-RNN [135]	66.9	32.8	25.5	51.1	75.8	180 M
SCST [109]	67.5	33.8	25.8	51.6	76.0	-
Vis-to-Lang [95]	72.9	30.7	27.9	-	54.3	157 M
ResNet with Attention [136]	55.6	33.5	-	-	-	-
AoANet [137]	67.4	33.5	26.7	52.7	84.7	115 M
CNN-Bi-GRU [138]	65.6	39.4	-	-	-	-
Darknet-LSTM (ours)	<b>82.3</b>	43.9	27.3	<b>65.1</b>	104.7	<b>97.7 M</b>
CATANIC [139]	78.8	<b>46.7</b>	-	63.8	<b>136.5</b>	300 M

Hybrid attention-based CNN-Bi-GRU [139] proposed a hybridized attention-based deep neural network (DNN) model. The model consists of an Inception-v3 convolutional neural network (CNN) encoder to extract image features, a visual attention mechanism to capture significant features, and a bidirectional gated recurrent unit (Bi-GRU) with an attention decoder to generate the image captions. CATANIC [139] applied the AoANet with DenseNet169 as the encoder to extract the initial features of the images and the modified transformer model as the decoder to transform the image feature vector into an image caption.

Table 4.5: Comparisons among our proposed Darknet-LSTM with SACM and some SOTA methods on MS COCO dataset. B@1, B@4, M, R, C, and P denote BLEU@1, BLEU@4, METEOR, ROUGE-L, CIDEr, and the model sizes.

Method	B@1	B@4	M	R	C	P
Hard Attention [32]	71.7	25.0	23.04	-	-	149 M
Adaptive Attention [94]	74.2	33.2	26.6	-	108.5	-
Actor-Critic Sequence [110]	77.8	33.7	26.4	55.4	110.2	-
Convolutional Image Captioning [140]	71.1	28.7	24.4	52.2	175	189.3 M
CNN Language Model [141]	72.6	30.3	24.6	-	96.1	-
SCST [109]	78.1	35.2	27.0	56.3	114.7	-
Up-Down [42]	80.2	36.9	27.6	57.1	117.9	108 M
GCN-LSTM [142]	77.4	37.1	28.1	57.2	117.1	-
SGAE [143]	81.0	38.5	28.2	58.6	123.8	-
AoANet (ResNeXt-101 Grid) [137]	81.0	39.4	29.1	58.9	126.9	115 M
X-Transformer [144]	81.9	40.3	29.6	59.5	131.1	11 B
RSTNet [145]	82.1	40.0	29.6	59.5	131.9	54 M/70 M
GET [146]	81.6	39.7	29.4	59.1	130.3	110 M
DLCT [96]	82.4	40.6	29.8	58.8	133.3	-
PureT [147]	82.8	41.4	30.1	60.4	136.0	-
ExpansionNet V2 [148]	<b>83.3</b>	42.1	30.4	60.8	138.5	129.6 M
BLIP-2 ViT-G OPT [149]	-	42.4	-	-	<b>144.5</b>	2700 M
Darknet-LSTM (ours)	83.1	44.3	<b>32.8</b>	<b>65.7</b>	148.0	<b>108.2 M</b>
OFA [150]	-	<b>44.9</b>	32.5	-	154.9	-
mPLUG [151]	-	<b>46.5</b>	32.0	-	<b>155.1</b>	510 M

Our model outperforms most previous models on the Flickr 8K dataset across all metrics in single and ensemble configurations. Our model outperforms other models by BLEU-1 and ROUGE-L. However, the CATANIC model has a slightly higher score in BLEU-4 and CIDEr-D despite having a parameter size almost twice as large as ours,

---

with differences of only 0.03 and 0.3, respectively.

More and more large-scale models are appearing and performing better and better. ExpansionNet V2 [148] applied block static expansion, which distributes and processes the input over a heterogeneous and arbitrarily big collection of sequences characterized by a different length compared to the input one. OFA [150] follows the previous research to adopt the encoder–decoder framework as the unified architecture. Both the encoder and the decoder are stacks of transformer layers. A transformer-based encoder layer consists of a self-attention and a feed-forward network (FFN). In contrast, a transformer-based decoder layer has a cross-attention network more than the encoder for building the connection between the decoder and the encoder output representations. The mPLUG [151] introduces a new asymmetric vision-language architecture with novel cross-modal skip-connections; it consists of  $N$  skip-connected fusion blocks to address two fundamental problems of information asymmetry and computation efficiency in cross-modal alignment. This model adapts the connected attention layer to each  $S$  asymmetric co-attention layer.

Our model achieved better results than the previous ExpansionNet V2 on the MSCOCO dataset, with improvements of 1.1 BLEU-4, 2.4 METEOR, 4.9 ROUGE-L, and 10.0 CIDEr-D. Compared to other models, our proposed model outperformed them by 0.3 METEOR and 0.57 ROUGE-L. However, our model was less efficient with the OFA and mPLUG on BLEU-4 and CIDEr scores. Despite this, our experiments have shown that the performance of approaches can be improved with larger datasets. Additionally, our best model could speed up predictions with a smaller size than attention-based and transformer-based large-scale models.

Ref. [32] introduced an attention-based image captioning model focusing on generating informative captions while considering computational efficiency to balance the performance and the cost. The method received a 71.8 BLEU-1 score and a 25.0 BLEU-4 score on the MS COCO dataset. In [94], the authors presented an adaptive attention mechanism that learns to attend to image regions for caption generation selectively, and this method received a 74.8 BLEU-1 score and 33.6 BLEU-4 score. The method pro-



posed in [110] was an actor-critic framework for training image captioning models. It tried to establish a balance via a trade-off between computational cost and captioning performance and received a 33.7 BLEU-4 score. Ref. [141] explored the use of language convolutional neural networks (CNNs) for image captioning, discussed the trade-off between computational cost and captioning performance, and received a 72.6 BLEU-1 score and 30.3 BLEU-4 score. Ref. [140] introduced a convolutional approach to image captioning that focused on reducing the computational cost while maintaining competitive performance. With the help of linear units, this model received a 71.1 BLEU-1 score and a 28.7 BLEU-4 score.

With several convolutional layers, our model performs better than most existing CNN+RNN models and transformer-based models and receives comparable results to those of the SOTA models with much smaller model sizes than the SOTA models.

## 4.8 Qualitative Analysis

Figure 4.6 shows prediction examples by our models with encoders with different output sizes on the Flickr 8K validation set, which shows reasonable prediction results. The wrong parts of a caption are marked. The GT caption is one of the five targets for evaluating the predicted sentence in the dataset. Compared with the ground truth captions, our best model with the encoded feature of size  $S_f = 128 \times 13 \times 13$  has obvious advantages in recognizing objects and some relative details. This advantage may come from the suitable feature with less loss of important information for the decoder.

## 4.9 Conclusions and Future Works

The detection of object classes and positions and their relationships should be considered in solving image captioning tasks. Therefore, the Darknet for object detection is the backbone of our proposed image captioning model. SACM, the size-adjustable convolutional module, is designed for feature extraction and dimension reduction in

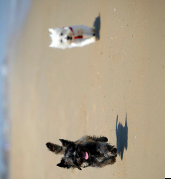

	64*13*13: a dog is laying on a hind of a grass, a stick in his mouth 128*13*13: a dog is laying on a grass, a ball in its mouth 256*13*13: a dog is laying on a grass, a tennis in his mouth 128*26*26: a dog is running on a grass, a ball in its mouth GT: a dog lays on his back with a favorite tennis ball in his mouth		64*13*13: a woman is sitting on a ground with <unk> <unk> the head 128*13*13: a man is sitting on a edge in front of building 256*13*13: a man is sitting on a edge surrounded of building <unk><unk> arm. 128*26*26: a man is sitting on the edge in of building GT: a man is sitting on the groundnext to the door of a building
	64*13*13: two dogs on a sand. 128*13*13: two dogs on a sand, a brown dog in a red collar. 256*13*13: two dogs running through a grass 128*26*26: two dogs jumping in a grass GT: two dogs running on a beach. 64*13*13: a brown dog is in the beach		64*13*13: a dog is running a purple collar 128*13*13: a dog is running in a collar 256*13*13: a dog is running a purple collar 128*26*26: a dog is running in a purple collar GT: a dog with a purple collar is running
	128*13*13: a brown dog is through a beach with a <unk> in its mouth 256*13*13: a brown dog is through a grass with a <unk> in the mouth 128*26*26: a dog brown is through a grass with a in of its mouth GT: a large dog runs on the beach with something hanging out of its mouth		64*13*13: a person is riding in the forest 128*13*13: a man biker is riding in in the forest 256*13*13: a man biker is riding in a forest 128*26*26: a man rider riding in the forest GT: a dirt biker rides through some trees
	64*13*13: two people stands in the street 128*13*13: a man and a woman are for cross the street 256*13*13: a man and a woman are for cross the street 128*26*26: a man and a woman with a <unk> are in the street GT: a man and a woman wait to cross the street		64*13*13: a girl jumps with a swing ball 128*13*13: a boy in a red shorts is playing a soccer ball on a beach 256*13*13: a boy is jumping into a ball 128*26*26: a boy in a red shorts is jumping for a soccer ball GT: a boy wearing a red bathing suit reaches for a soccer ball while running in sand
	64*13*13: a boy is jumping over the air 128*13*13: a boy in a blue shorts jumps a flip 256*13*13: a boy in a jeans jumping a flip into the ocean 128*26*26: a boy in a howts jumping a flip in the water GT: man with blue pants flipping in the air		64*13*13: a person on a red background 128*13*13: a man in a black shirt is down a street 256*13*13: a man in a black shirt is walking along a path 128*26*26: a man in a shirt is walking to a beach GT: an older man in a long black shiet is walking down a cobblestone street alone

Figure 4.6: Visualization of our models with different encoders on validation images of Flickr 8K dataset. The wrong parts of a caption are marked. The GT caption is one of the five targets for evaluating the predicted sentence in the dataset.

this paper. With the SACM, convolutional layers are applied for feature dimension reduction while losing less critical information on global and local features. With feature dimension reduction, the parameters of the whole model are smaller. With the convolutional layers, the feature size is reduced while expanding the depth of the network for receiving high-level semantic and contextual information. Faster implementation and better performance could be achieved simultaneously in our end-to-end image captioning system with a pre-trained Darknet, SACM, and LSTM.

The end-to-end neural network system proposed in this paper, Darknet-SACM-LSTM, is trained to maximize the likelihood of the correct words in the final sentence describing the given image. After training, our proposed systems can automatically generate a descriptive caption in plain English for a given image. Experiments on the Flickr 8K and MS COCO datasets show the robustness of our Darknet-SACM-LSTM system in terms of speed and several metrics of BLEU scores, METEOR, ROUGE, and CIDEr. By using one or more convolutional layers, SACM could reduce the number of features, speed up the predicting process, and maintain the performance of sentence quality measured by using both the cross-entropy loss and BLEU score.

The experimental results also indicate that neither the best training loss nor the best evaluating loss could let the learned systems with the highest metrics engage in image captioning. By modifying the cross-entropy loss function, it would be necessary to explicitly consider the relationships between items and their positions in the images. The modified loss functions might help the image captioning system to achieve better metrics. Meanwhile, all the parameters in our proposed Darknet-SACM-LSTM are trainable. It would be interesting to know which parts should be adaptive and which parts could be fixed. Even a faster system could be implemented by fixing some parameters besides the feature reductions.

# Chapter 5

## Performance and Cost Balancing

### Image Captioning with Vision

#### Transformer

By analysis of results in Chapter 3 and 4, the SACM module is practical in maintaining a balance between performance and cost. Although the parameter size of Darknet is much larger than VGGNet, our model still received good performance with shorter GPU machine time and similar time spent on the CPU machine. It proves that a large-scale model can also be used as an encoder of real-time image captioning. This chapter proposed a new image captioning model based on a vision transformer, which received comparable performance in image recognition. As we want to use it for real-time generation, the proposed and existing models are compared on both the BLEU score and training and testing time. Moreover, this is the first time the vision transformer method has been used on an image caption generation task.

#### 5.1 Introduction

Image captioning is a task to generate a descriptive sentence for a given image. It connects computer vision and language processing by extracting visual information and interpreting it in human language. The encoder-decoder architecture remains dominant

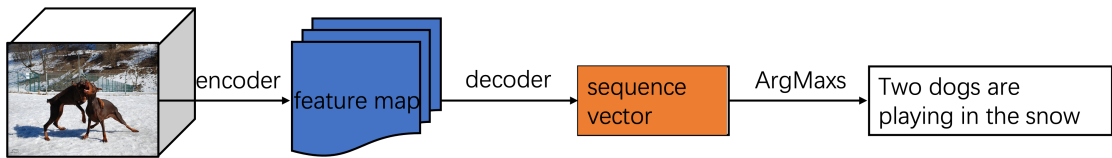


Figure 5.1: Overview of the encoder-decoder model for image captioning. The encoder extracts image features, while the decoder generates text descriptions by analyzing the features.

in current state-of-the-art (SOTA) models [152–161]. The encoder is responsible for encoding visual features. At the same time, the sentence decoder learns to generate the sequential sentence word-by-word, as shown in Figure 5.1.

Both the encoder and decoder contribute to the quality of generated captions. Convolutional neural networks (CNNs) are still the most prevalent in computer vision. To analyse the gap between visual processing and language processing, most existing architectures of image captioning follow the encoder-decoder architecture with CNN-RNN pipeline for image encoding and text generation [31, 32, 42, 142, 162].

For a long time, CNNs have been the preferred option for image-processing tasks. They excel at learning from vast amounts of image data and have achieved impressive results in tasks such as image classification, segmentation, and object detection. In vision, attention is applied in conjunction with convolutional networks or used to replace specific components of convolutional networks while keeping their overall structure in place. Vision Transformer (ViT) [3] attains excellent results compared to state-of-the-art convolutional networks, requiring substantially fewer computational resources.

The ViT model achieved excellent results and approached or surpassed the SOTA on multiple image recognition benchmarks. ViT employs self-attention mechanisms to capture global relationships in input images, providing more effective modelling of long-range dependencies and global structural information than the traditional CNN encoder. ViT offers benefits in scenarios where global dependencies and contextual understanding are critical. In the image captioning task, the global information and contextual relationships are essential for measuring relationships among items in the image. The pre-trained ViT model can also perform comparably to CNNs with a large-scale dataset such as ImageNet21k.

---

Previous research has proved the object detection model can be used as an encoder for image captioning. Our research aims to propose a comparable image captioning model that can generate an accurate depiction, including item classes and position relationships. The ViT model, which divides images into several patches and encodes patches with their position embedding information, can be a good choice for the encoder of the image captioning model.

However, suppose we adopt the original model as the encoder and input the predicted results to the decoder directly. The feature map with global information and contextual relationships provided by the hidden layers might be lost in that case. If we feed the outputs of the hidden layers to the decoder, the large input size will cause a layer cost of time and computational resources. In this chapter, we would like to find an excellent way to keep the cost of computational resources and the predicting time, whether to use feature maps directly extracted from a tiny version or reduced by convolutional layers.

Building upon previous research, we introduce a fine-tuned model, which combines a Vision Transformer as an encoder with an LSTM network serving as a decoder, with a particular emphasis on image captioning. Our motivation lies in leveraging this model for wearable devices to assist visually impaired individuals in perceiving their surroundings. However, we face a challenge in striking a balance between quality and speed, given the unique requirements of our application. Consequently, this paper delves into comparing the performance of feature maps with varying scales to address this challenge.

## 5.2 Outline

We arrange this chapter in the following order.

- Section [5.3](#): An overview of the proposed method.
- Section [5.4](#): Detailed discussion about the dataset and experiment settings.
- Section [5.5](#): Discussion on results and comparison to SOTA models.

Table 5.1: Different Versions of ViT model and its Feature Size

Model	Feature Size
Tiny-224	[batch_size, 198, 192]
Small-224	[batch_size, 198, 384]
Base-224	[batch_size, 198, 768]
Base-384	[batch_size, 578, 768]

- Section 5.6: Conclusion and possible future work.

### 5.3 Proposed Method

The Vision Transformer (ViT) model detects objects using an encoder with layer norm, multi-head attention, dropout, and an MLP block, as shown in Figure 5.2. For feeding images to the Transformer encoder, each image is split into a sequence of fixed-size, non-overlapping patches, which are then linearly embedded. A  $[CLS]$  token is added to represent an entire image, which can be used for classification. This process is similar to the caption-generating process, which is also input with a sequence of image representation and a  $[CLS]$  token. This is also one of the main reasons we chose ViT as the encoder of our model.

The original Vision Transformer was pre-trained using a resolution of  $224 \times 224$ . During fine-tuning, it is beneficial to use a higher resolution ( $384 \times 384$ ) than pre-training []. To fine-tune at higher resolution, the authors perform 2D interpolating the pre-trained position embeddings according to their location in the original image.

Facebook AI proposed four variants available: tiny, small, base-224 and base-384 versions. As the Vision Transformer expects each image to be of the exact resolution, one can use the corresponding image processor to resize and normalize images for the model. The patch and image resolutions used during pre-training or fine-tuning are reflected in the name of each checkpoint. These models are all pre-trained with the Imagenet-21k dataset. The four version models and the corresponding feature map size are shown in Table 5.1.

According to the table, the feature sizes of base-224 and base-384 versions are still

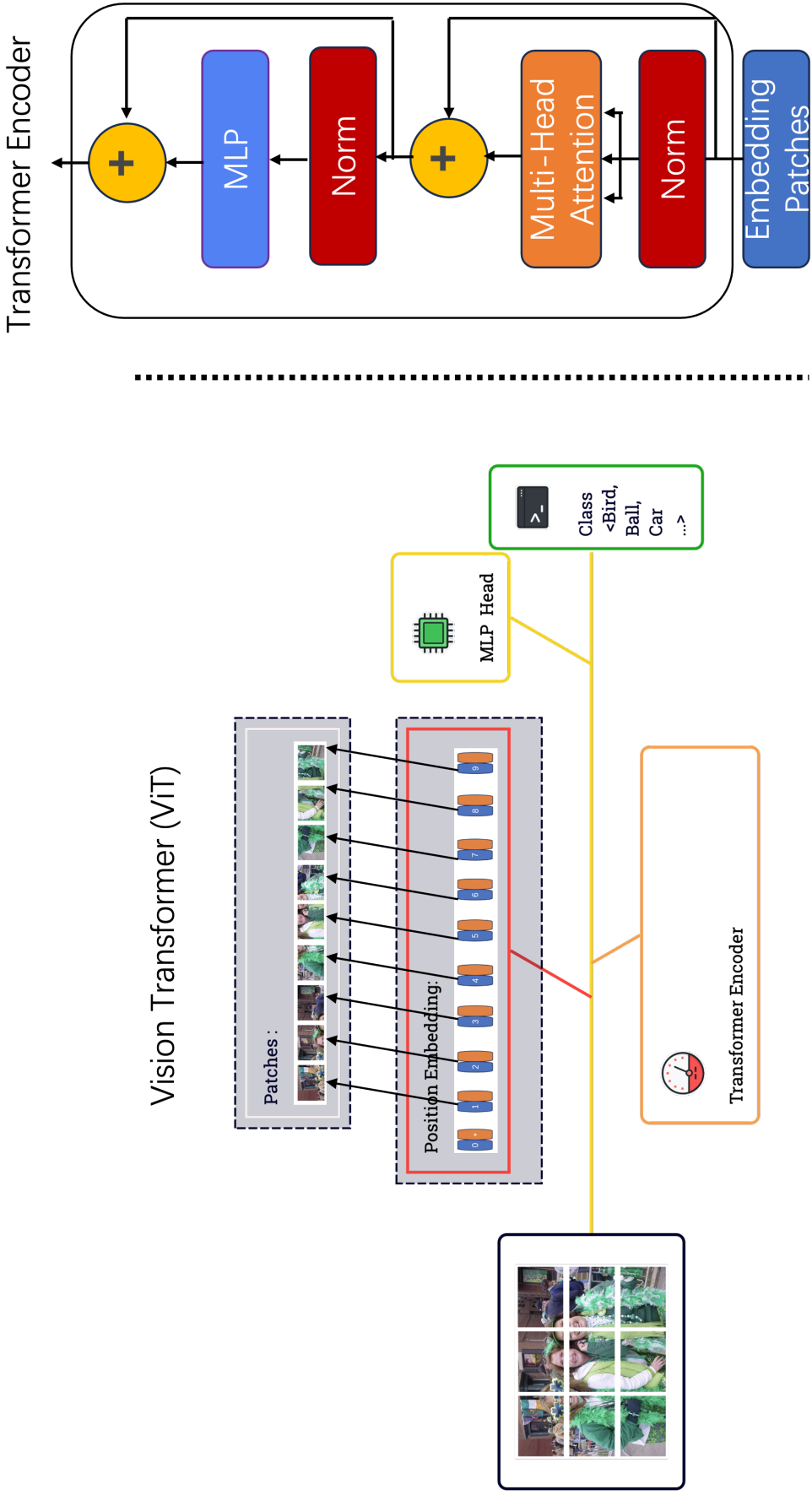


Figure 5.2: Overview of ViT model [3]. The ViT model splits an image into fixed-size patches, linearly embeds each of them, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder.



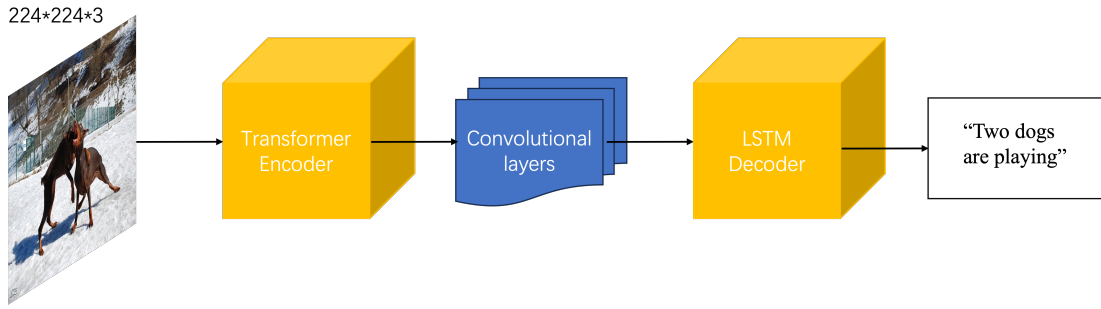


Figure 5.3: Overview of the proposed model. To balance the performance and cost, we apply convolutional layers for feature map dimension reduction while maintaining the global information and the relationships between different regions.

substantial for the LSTM model. Generally speaking, a larger feature map provides richer spatial contextual information, and the model better understands the relationship between the target and its surroundings. However, real-time detection is crucial for a mobile-oriented model, so keeping the performance while reducing computation costs with a smaller feature map is essential. In this chapter, we compared the performance and cost of image captioning models with the previous three ViT models as an encoder.

As mentioned before, the number of parameters in an LSTM model depends on its input, hidden, and output sizes. If the input size is halved while the other data sizes remain the same, the weight matrix from the input layer to the hidden layer will have half as many rows with the same number of columns that define the hidden size. With limited computational resources, the massive size of parameters will cause out-of-memory.

In addition, while accuracy is vital in image captioning, speed should also be considered, especially for mobile real-time applications. To find the optimum model, we fine-tuned the model with large feature maps by inserting convolutional layers for feature dimension reduction to maintain the balance of performance and cost of the two base versions of ViT encoder, as shown in Figure 5.3. By maintaining accuracy and achieving more stability with the reduced feature dimension, we expect the processing time to decrease.

To compare the performance of different encoders, we keep using the single-layer Long Short-Term Memory (LSTM) network as a sequence generation decoder in our

---

model for image captioning. The LSTM takes image features as input to generate captions. The forget, input and output gates help process sequential data and learn long-range dependencies. The LSTM is composed of a hidden state and three gate units: forget, input, and output gate. The forget gate determines which previous hidden-state information to retain, the input gate decides which new input information to incorporate, and the output gate controls how the current input generates a new hidden state.

As mentioned, we use pre-trained Vision Transformer models to extract image features, represented as vectors, and feed the dimension-reduced feature map into the LSTM as input sequences. Like previous research, we use cross-entropy loss and optimize with Adam. We randomly sample data, compare generated sequences with ground truth, and compute loss. We also use learning rate decay to enhance training stability.

During the decoding phase, we use a sampling strategy to generate image captions by sampling words from the output distribution at each time step. The word with the highest probability is chosen as the prediction for diverse and creative caption generation at each time step. During training, we tuned the hyperparameters, setting the LSTM's hidden dimension to 512 and using the Adam optimizer with an initial learning rate of 0.001 to optimize performance in the image captioning task.

## 5.4 Dataset and Experiment Settings

The main goal of an image captioning model is to achieve less distance between the predicted sentence and the GT. We utilized the Flickr 8K dataset [56] sourced from Flickr. The dataset encompasses 8,000 samples, with five human-annotated captions for each image. We employed a standard division into training, validation, and testing subsets, comprising 6,000 images for training and 1,000 for validation and testing, respectively.

Unlike the Darknet we used in Chapter 4, the ViT model pays more attention to global contextual understanding. We applied the pre-trained ViT as an encoder of our model to maintain such global information.

We set several experiments with the three ViT versions as encoders in the experiment to compare the performance and computational cost. For the goal of real-time detection, we measured the prediction time on both GPU and CPU machines.

We applied corresponding pre-processes to meet the different requirements of different encoders. We also embedded labels to transform textual captions into numerical representations. Our model is based on the Vision Transformer and Long Short-Term Memory network.

LSTMs are well-suited for sequential data processing and essential for generating fluent and contextually relevant sentences. Their recurrent nature allows them to capture dependencies and relationships within textual data, which is crucial for formulating coherent captions that accurately describe the visual content.

The combination of ViT and LSTM offers a synergistic approach. The ViT contributes strong visual feature extraction capabilities, enabling the model to comprehend images effectively. The LSTM, in turn, refines these features by sequentially generating captions, ensuring that the generated text aligns with the visual context and syntactic structure. To gauge the effectiveness of our model, we compare its performance against VGG16-LSTM using appropriate statistical tests.

All experiments were conducted within a computer environment running Ubuntu 20.04. The hardware configuration included an AMD Ryzen 9-3900X CPU with 32GB of RAM and a GTX 3090 GPU boasting 24GB of memory. Pytorch served as the chosen deep learning framework. Adhering to prior research practices, a maximum caption length of 20 words was enforced for both datasets. A carefully curated vocabulary was constructed by excluding words occurring fewer than five times. Various machine translation metrics, namely BLEU, METEOR, ROUGE, and CIDEr, were employed for evaluation.

BLEU assesses the degree of n-gram overlap between generated captions and ground truth references. The METEOR evaluation metric gauges semantic propositional image caption quality by computing a weighted harmonic mean of single-word recall and precision. ROUGE, encompassing various variants like ROUGE-L and ROUGE-

---

N, evaluates the match between generated word sequences and reference descriptions. Our experiments primarily employ ROUGE-L, which defines the longest common subsequence as the longest identical fragment within generated and ground-truth sentences. CIDEr, a consensus-based automatic caption evaluation metric, treats sentences as documents and employs TF-IDF to assign word weights.

## 5.5 Empirical Analysis

Firstly, we provide a table with the performance of the three related version models and the predicting time on CPU and GPU, as shown in Table [5.2](#).

For comparison, we also calculated the number of operators of our proposed models shown in Table [5.3](#).

We provide a detailed overview of the quantitative evaluation metrics used to assess our model’s performance, as shown in Table [5.4](#). These metrics include BLEU, METEOR, ROUGE, and CIDEr. They measure how well our model captures visual understanding, generalization, and human-like captioning.

Table 5.2: The cost and performance of our model with different versions of ViT as an encoder on the testing set for Flickr 8K. B@1 and B@4 denote BLEU-1 and BLEU-4 scores. “No.” denotes the number of learning epochs when the models received the best BLEU-4 score on the evaluation data.

Model	Predicting time (on CPU)	Predicting time (on GPU)	B@1	B@4	Loss	No.
ViT-Tiny-38016	2.9s	0.02s	81.7	42.5	2.67	15
ViT-Small-76032	3.2s	0.04s	81.4	40.7	2.76	17
ViT-Base-224-38016	3.1s	0.2s	82.7	43.1	2.63	17
ViT-Base-224-8192	2.7s	0.1s	81.9	42.6	2.65	15
ViT-Base-224-4096	2.6s	0.08s	82.7	42.2	2.67	15

Table 5.3: Number of operators of our models and predicting time on CPU and GPU machine

Model	FLOPs	Predicting time (CPU)	Predicting time (GPU)
VGG+LSTM-4096	23.9G	2.6s	0.2s
Darknet + SACM + LSTM-10816	66.3G	3.2s	0.2s
ViT-tiny + LSTM-38016	36.6G	2.9s	0.02s
ViT-small + LSTM-76032	67.6G	3.2s	0.04s
ViT-base + LSTM-38016	52.1G	3.1s	0.2s
ViT-base + LSTM-8192	27.6G	2.7s	0.1s
ViT-base + LSTM-4096	24.3G	2.6s	0.08s

Table 5.4: Comparisons of our proposed ViT-LSTM with convolutional feature reduction and some SOTA methods on Flickr 8K dataset. B@1, B@4, M, R, C, and P denote BLEU@1, BLEU@4, METEOR, ROUGE-L, CIDEr, and the model sizes.

Method	B@1	B@4	M	R	C	P
Neuraltalk2 [5]	57.9	16.0	-	-	-	31 M
D-CNN [132]	49.5	20.1	42.5	-	-	-
VGG16-LSTM [133]	62.6	28.7	-	-	-	138 M
Hard-attention [32]	66.0	31.4	24.8	50.3	68.9	149 M
Neural Baby Talk [134]	66.4	32.6	26.2	52.5	84.5	37.7 M
m-RNN [135]	66.9	32.8	25.5	51.1	75.8	180 M
SCST [109]	67.5	33.8	25.8	51.6	76.0	-
Vis-to-Lang [95]	72.9	30.7	27.9	-	54.3	157 M
ResNet with Attention [136]	55.6	33.5	-	-	-	-
AoANet [137]	67.4	33.5	26.7	52.7	84.7	115 M
CNN-Bi-GRU [138]	65.6	39.4	-	-	-	-
ViT-LSTM-8192 (ours)	<b>82.9</b>	42.6	28.4	63.1	103.0	105M
ViT-LSTM-4096 (ours)	81.7	42.2	26.9	63.2	101.6	95M
CATANIC [139]	78.8	<b>46.7</b>	-	63.8	<b>136.5</b>	300 M

We explore the influence of different feature dimension settings on the model’s performance. We investigate the effects of varying feature dimensions as 4,096 (the same as the baseline model) and 8192 (2 times larger than the baseline model) on the model’s performance.

In addition, we also discuss the practical implications of our model’s performance, particularly in the context of wearable device-oriented models. For this purpose, we compare the parameter size of our model with other SOAT models.

As we can see from the table, our model shows competitive performance. Specifically, it outperforms the original CNN+RNN-based methods, indicating that our critical designs are effective for image captioning tasks. In addition, our model’s performance

---

is also better than many models with larger scale. This suggests that further research on the encoder–decoder architecture could inspire new efforts in this area.

Our model outperforms most of the previous models on the Flickr 8K dataset across all metrics in both single and ensemble configurations. Our model outperforms other models by BLEU-1 and METEOR. This is mainly because the ViT encoder pre-trained with a large dataset performs better on image classification. However, the CATANIC model has a slightly higher score in BLEU-4 and CIDER-D despite having a parameter size almost three times larger than ours, with differences of 3% and 3.5%, respectively.

## 5.6 Conclusion and Possible Future Work

In this paper, we propose a ViT-LSTM model for solving image captioning tasks to address the challenge of long-range dependencies. The ViT model pre-trained with the ImageNet 21k dataset can capture global context, enabling the following LSTM to generate captions that reflect local and global visual cues.

Moreover, we employ convolutional layers for feature map dimension reduction. Convolutional layers preserve spatial relationships and local patterns while reducing dimensionality. This helps the model understand local details and relationships between regions. In this way, better performance and less computational cost could be balanced.

Our proposed model can automatically generate descriptive captions in plain English for images. Experiments on the Flickr 8K dataset show the robustness of our ViT-LSTM model in terms of parameter scale and multiple metrics such as BLEU scores, METEOR, ROUGE, and CIDEr. Convolutional layers can reduce the dimension of the final feature map from the encoder to maintain the generated caption quality while speeding up the prediction process. The experiment results proved that our model could address the challenge of balancing performance and cost.

It is also essential to acknowledge the limitations of our model. As we know, the cross-entropy loss calculates the distance between the prediction distribution and the target. It is a good choice for prediction with only one correct ground truth. However,



image captioning has five or even more ground truth and several specific metrics. It would be interesting to research how to train a model to fit the metrics better.

# Chapter 6

## Conclusion

So far, we have seen that image captioning has emerged as a dynamic and promising field at the intersection of computer vision and natural language processing. The intricacies of image captioning, delving into the challenges the visually impaired face in comprehending their surroundings and the significance of leveraging machine learning to facilitate a more inclusive and accessible world, one is that the mobile device-oriented models require real-time prediction. Moreover, training a deep and large-scale neural network with high performance in a single GPU machine is computationally expensive. Even loading a pre-trained large-scale image captioning model in a single lab-level GPU sometimes runs out of memory. In addition, training an image captioning model in a single GPU device takes several days. During testing time, the higher total number of parameter costs require a longer time for prediction. This is why we research balancing performance and cost of image captioning.

In conclusion, our research aims to determine the model's effectiveness for real-world applications, particularly in providing visually impaired individuals with immediate, contextually relevant information about their surroundings. In this dissertation, we first proposed a novel Darknet-based image captioning model. Through several theoretical analyses and empirical studies with image captioning, we have shown that such models achieve comparable performance to the large-scale image captioning models. In the following paragraphs, we conclude the dissertation in a chapter-wise manner.

Chapter 2 summarized a literature review of existing image captioning models. We provided some preliminary knowledge about our proposed models in this dissertation. The literature review undertaken in this dissertation has served as the foundation upon which our research journey was built. Through an exhaustive examination of existing studies, we gained insights into the historical progression, significant milestones, and the evolving landscape of image captioning. The review provided a critical perspective on the challenges, trends, and state-of-the-art methodologies shaping the field.

In Chapter 3, we first introduced a novel end-to-end image captioning model architecture that combines a Darknet-based feature extractor with an LSTM-based caption generator. Unlike existing models that rely on pre-trained CNNs as intermediaries, our model allows for a direct path from raw images to generated captions, simplifying the overall process. The end-to-end image captioning model utilizes carefully designed feature extractors and caption generators to enhance caption quality. Empirical research supports the model’s outstanding performance, and its low parameter requirements and efficiency make it well-suited for various practical applications. Our model is designed with a low computational and time cost by convolutional feature dimension reduction, making it highly suitable for resource-constrained environments.

Chapter 4 aims to strike a harmonious balance between performance and computational cost. To achieve this, this chapter introduces an innovative approach—utilizing a pre-trained deep learning model initially designed for object detection to encode the input image. By leveraging this pre-trained model, features representing various objects within the image can be efficiently extracted in a single pass. Furthermore, we propose incorporating a size-adjustable convolutional module (SACM) as an intermediary step before decoding these features into coherent sentences.

The experimental results demonstrate the effectiveness of our approach. With the appropriately configured SACM, our model achieves remarkable performance on standard image captioning benchmarks. Leveraging a pre-trained object detection model and a size-adjustable convolutional module, our method demonstrates outstanding results on benchmark datasets while reducing the computational overhead substantially

---

compared to existing approaches.

Lastly, in Chapter 5, we introduced a novel image captioning model that combines a vision transformer encoder and LSTM decoder, emphasizing its unique approach and real-time applicability while providing insights into its performance compared to established vision models. This approach represents an innovative departure from existing methods, which typically involve encoding and decoding stages. Importantly, this research marks the first utilization of the vision transformer method in image caption generation. Furthermore, we applied the convolutional layer for feature dimension reduction to emphasize the practicality of real-time image caption generation.

# References

- [1] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, “Collecting image annotations using amazon’s mechanical turk,” in *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon’s Mechanical Turk*, 2010, pp. 139–147.
- [2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [4] P. Jay, “Understanding and implementing architectures of resnet and resnext for state-of-the-art image classification: From microsoft to facebook,” 2018. [Online]. Available: <https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-classification>
- [5] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.
- [6] C. Wang, H. Yang, C. Bartz, and C. Meinel, “Image captioning with deep bidirectional lstms,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 988–997.
- [7] J. Wang, W. Wang, L. Wang, Z. Wang, D. D. Feng, and T. Tan, “Learning visual relationship and context-aware attention for image captioning,” *Pattern Recognition*, vol. 98, p. 107075, 2020.
- [8] Y. Zheng, Y. Li, and S. Wang, “Intention oriented image captions with guiding objects,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8395–8404.
- [9] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikingler-Cinbis, F. Keller, A. Muscat, and B. Plank, “Automatic description generation from images: A survey of models, datasets, and evaluation measures,” *Journal of Artificial Intelligence Research*, vol. 55, pp. 409–442, 2016.
- [10] A. Karpathy, “Connecting images and natural language,” Ph.D. dissertation, Stanford University, 2016.

- 
- [11] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Gray scale and rotation invariant texture classification with local binary patterns,” in *Computer Vision-ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part I 6*. Springer, 2000, pp. 404–420.
- [12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.
- [13] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [15] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” *arXiv preprint arXiv:1404.2188*, 2014.
- [16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [19] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 111–118.
- [20] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.
- [21] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, “Ask the locals: multi-way local pooling for image recognition,” in *2011 international conference on computer vision*. IEEE, 2011, pp. 2651–2658.
- [22] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, “Learning mid-level features for recognition,” in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2559–2566.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
-

- [24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [25] M. A. Covington, “Building natural language generation systems,” *Language*, vol. 77, no. 3, pp. 611–612, 2001.
- [26] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, “Extensions of recurrent neural network language model,” in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 5528–5531.
- [27] A. Graves and A. Graves, “Long short-term memory,” *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.
- [28] Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” *arXiv preprint arXiv:1508.01991*, 2015.
- [29] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based lstm for aspect-level sentiment classification,” in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.
- [30] C. Zhou, C. Sun, Z. Liu, and F. Lau, “A c-lstm neural network for text classification,” *arXiv preprint arXiv:1511.08630*, 2015.
- [31] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.
- [32] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.
- [33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [34] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [35] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional image generation with pixelcnn decoders,” *Advances in neural information processing systems*, vol. 29, 2016.
- [36] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, “A convolutional encoder model for neural machine translation,” *arXiv preprint arXiv:1611.02344*, 2016.
- [37] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *International conference on machine learning*. PMLR, 2017, pp. 1243–1252.

- 
- [38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [39] M. Ilse, J. Tomczak, and M. Welling, “Attention-based deep multiple instance learning,” in *International conference on machine learning*. PMLR, 2018, pp. 2127–2136.
- [40] W. Wang and J. Shen, “Deep visual attention prediction,” *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2017.
- [41] P. Barros, G. I. Parisi, C. Weber, and S. Wermter, “Emotion-modulated attention improves expression recognition: A deep learning model,” *Neurocomputing*, vol. 253, pp. 104–114, 2017.
- [42] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
- [43] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.
- [44] H. R. Tavakoli and J. Laaksonen, “Prominent object detection and recognition: A saliency-based pipeline,” *CoRR*, 2017.
- [45] P. Cao, Z. Yang, L. Sun, Y. Liang, M. Q. Yang, and R. Guan, “Image captioning with bidirectional semantic attention-based guiding of long short-term memory,” *Neural Processing Letters*, vol. 50, pp. 103–119, 2019.
- [46] A. Tariq and H. Foroosh, “Feature-independent context estimation for automatic image annotation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1958–1965.
- [47] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, “Every picture tells a story: Generating sentences from images,” in *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV 11*. Springer, 2010, pp. 15–29.
- [48] S. Li, G. Kulkarni, T. Berg, A. Berg, and Y. Choi, “Composing simple image descriptions using web-scale n-grams,” in *Proceedings of the fifteenth conference on computational natural language learning*, 2011, pp. 220–228.
- [49] G. K. V. P. S. Dhar, S. Li, Y. C. A. C. B. Tamara, and L. Berg, “Baby talk: Understanding and generating simple image descriptions,” 2013.
- [50] D. Elliott and F. Keller, “Image description using visual dependency representations,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1292–1302.
-



- [51] A. Aker and R. Gaizauskas, “Generating image descriptions using dependency relational patterns,” in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 1250–1258.
- [52] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi, “Collective generation of natural image descriptions,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 359–368.
- [53] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, “Treetalk: Composition and compression of trees for image descriptions,” *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351–362, 2014.
- [54] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. Berg, T. Berg, and H. Daumé III, “Midge: Generating image descriptions from computer vision detections,” in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 747–756.
- [55] V. Ordonez, G. Kulkarni, and T. Berg, “Im2text: Describing images using 1 million captioned photographs,” *Advances in neural information processing systems*, vol. 24, 2011.
- [56] M. Hodosh, P. Young, and J. Hockenmaier, “Framing image description as a ranking task: Data, models and evaluation metrics,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.
- [57] C. Sun, C. Gan, and R. Nevatia, “Automatic concept discovery from parallel text and visual corpora,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2596–2604.
- [58] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik, “Improving image-sentence embeddings using large weakly annotated photo collections,” in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part IV 13*. Springer, 2014, pp. 529–545.
- [59] R. Kiros, R. Salakhutdinov, and R. S. Zemel, “Unifying visual-semantic embeddings with multimodal neural language models,” *arXiv preprint arXiv:1411.2539*, 2014.
- [60] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, “Boosting image captioning with attributes,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4894–4902.
- [61] A. Mnih and G. Hinton, “Three new graphical models for statistical language modelling,” in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 641–648.
- [62] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, “Exploring the limits of language modeling,” *arXiv preprint arXiv:1602.02410*, 2016.

- 
- [63] R. Kiros, R. Salakhutdinov, and R. Zemel, “Multimodal neural language models,” in *International conference on machine learning*. PMLR, 2014, pp. 595–603.
- [64] A. Karpathy, A. Joulin, and L. F. Fei-Fei, “Deep fragment embeddings for bidirectional image sentence mapping,” *Advances in neural information processing systems*, vol. 27, 2014.
- [65] M.-C. De Marneffe, B. MacCartney, C. D. Manning *et al.*, “Generating typed dependency parses from phrase structure parses.” in *Lrec*, vol. 6, 2006, pp. 449–454.
- [66] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, “Deep captioning with multimodal recurrent neural networks (m-rnn),” *arXiv preprint arXiv:1412.6632*, 2014.
- [67] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, “Devise: A deep visual-semantic embedding model,” *Advances in neural information processing systems*, vol. 26, 2013.
- [68] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, “Grounded compositional semantics for finding and describing images with sentences,” *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014.
- [69] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, “Explain images with multimodal recurrent neural networks,” *arXiv preprint arXiv:1410.1090*, 2014.
- [70] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [71] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [72] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [73] C. Gan, T. Yang, and B. Gong, “Learning attributes equals multi-source domain generalization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 87–97.
- [74] X. Chen and C. Lawrence Zitnick, “Mind’s eye: A recurrent visual representation for image caption generation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2422–2431.
- [75] J. Johnson, A. Karpathy, and L. Fei-Fei, “Densecap: Fully convolutional localization networks for dense captioning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4565–4574.
-

- [76] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, “Draw: A recurrent neural network for image generation,” in *International conference on machine learning*. PMLR, 2015, pp. 1462–1471.
- [77] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, “Spatial transformer networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [78] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.
- [79] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [80] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.
- [81] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, “Guiding the long-short term memory model for image caption generation,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2407–2415.
- [82] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, “Generation and comprehension of unambiguous object descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.
- [83] K. van Deemter, I. van der Sluis, and A. Gatt, “Building a semantically transparent corpus for the generation of referring expressions.” in *Proceedings of the Fourth International Natural Language Generation Conference*, 2006, pp. 130–132.
- [84] J. Viethen and R. Dale, “The use of spatial relations in referring expression generation,” in *Proceedings of the Fifth International Natural Language Generation Conference*, 2008, pp. 59–67.
- [85] M. Mitchell, K. van Deemter, and E. Reiter, “Natural reference to objects in a visual domain,” in *Proceedings of the 6th international natural language generation conference*, 2010.
- [86] M. Mitchell, K. Van Deemter, and E. Reiter, “Generating expressions that refer to visible objects,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1174–1184.
- [87] N. FitzGerald, Y. Artzi, and L. Zettlemoyer, “Learning distributions over logical forms for referring expression generation,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1914–1925.
- [88] C. J. Tomlin, J. Lygeros, and S. S. Sastry, “A game theoretic approach to controller design for hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.

- 
- [89] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, “Referitgame: Referring to objects in photographs of natural scenes,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.
- [90] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, “From captions to visual concepts and back,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.
- [91] O. Maron and T. Lozano-Pérez, “A framework for multiple-instance learning,” *Advances in neural information processing systems*, vol. 10, 1997.
- [92] A. Berger, S. A. Della Pietra, and V. J. Della Pietra, “A maximum entropy approach to natural language processing,” *Computational linguistics*, vol. 22, no. 1, pp. 39–71, 1996.
- [93] F. J. Och, “Minimum error rate training in statistical machine translation,” in *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, 2003, pp. 160–167.
- [94] J. Lu, C. Xiong, D. Parikh, and R. Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375–383.
- [95] X. Li, A. Yuan, and X. Lu, “Vision-to-language tasks based on attributes and attention mechanism,” *IEEE transactions on cybernetics*, vol. 51, no. 2, pp. 913–926, 2019.
- [96] Y. Luo, J. Ji, X. Sun, L. Cao, Y. Wu, F. Huang, C.-W. Lin, and R. Ji, “Dual-level collaborative transformer for image captioning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 3, 2021, pp. 2286–2293.
- [97] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, “Unified vision-language pre-training for image captioning and vqa,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 041–13 049.
- [98] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, J. Gao *et al.*, “Vision-language pre-training: Basics, recent advances, and future trends,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 14, no. 3–4, pp. 163–352, 2022.
- [99] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som *et al.*, “Image as a foreign language: Beit pretraining for all vision and vision-language tasks,” *arXiv preprint arXiv:2208.10442*, 2022.
- [100] S. Takada, R. Togo, T. Ogawa, and M. Haseyama, “Generation of viewed image captions from human brain activity via unsupervised text latent space,” in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 2521–2525.
-

- [101] I. Laina, C. Rupprecht, and N. Navab, “Towards unsupervised image captioning with shared multimodal embeddings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7414–7424.
- [102] Y. Zhou, W. Tao, and W. Zhang, “Triple sequence generative adversarial nets for unsupervised image captioning,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7598–7602.
- [103] Y. Feng, L. Ma, W. Liu, and J. Luo, “Unsupervised image captioning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4125–4134.
- [104] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” *Advances in neural information processing systems*, vol. 12, 1999.
- [105] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, “Deep reinforcement learning-based image captioning with embedding reward,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 290–298.
- [106] V. Konda and J. Tsitsiklis, “Actor-critic algorithms,” *Advances in neural information processing systems*, vol. 12, 1999.
- [107] Z. Ren, H. Jin, Z. Lin, C. Fang, and A. Yuille, “Multi-instance visual-semantic embedding,” *arXiv preprint arXiv:1512.06963*, 2015.
- [108] —, “Joint image-text representation by gaussian visual-semantic embedding,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 207–211.
- [109] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7008–7024.
- [110] L. Zhang, F. Sung, F. Liu, T. Xiang, S. Gong, Y. Yang, and T. M. Hospedales, “Actor-critic sequence training for image captioning,” *arXiv preprint arXiv:1706.09601*, 2017.
- [111] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: Lessons learned from the 2015 mscoco image captioning challenge,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 652–663, 2016.
- [112] R. Staniūtė and D. Šešok, “A systematic literature review on image captioning,” *Applied Sciences*, vol. 9, no. 10, p. 2024, 2019.
- [113] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [114] A. C. G. Jocher and J. Qiu, “Yolo by ultralytics,” <https://github.com/ultralytics/ultralytics>, 2023.

- 
- [115] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
- [116] J. Terven and D. Cordova-Esparza, “A comprehensive review of yolo: From yolov1 to yolov8 and beyond,” *arXiv preprint arXiv:2304.00501*, 2023.
- [117] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [118] Q. Wu, C. Shen, L. Liu, A. Dick, and A. Van Den Hengel, “What value do explicit high level concepts have in vision to language problems?” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 203–212.
- [119] Q. Wu, C. Shen, P. Wang, A. Dick, and A. Van Den Hengel, “Image captioning and visual question answering based on attributes and external knowledge,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1367–1381, 2017.
- [120] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
- [121] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2641–2649.
- [122] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [123] H. A. Almuzaini and A. M. Azmi, “Impact of stemming and word embedding on deep learning-based arabic text categorization,” *IEEE Access*, vol. 8, pp. 127 913–127 928, 2020.
- [124] M. Xin, H. Zhang, D. Yuan, and M. Sun, “Learning discriminative action and context representations for action recognition in still images,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 757–762.
- [125] D. Forsyth, “Object detection with discriminatively trained part-based models,” *Computer*, vol. 47, no. 02, pp. 6–7, 2014.
- [126] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
- [127] Pytorch, “Word embeddings: Encoding lexical semantics,” [https://pytorch.org/tutorials/beginner/nlp/word\\_embeddings\\_tutorial.html](https://pytorch.org/tutorials/beginner/nlp/word_embeddings_tutorial.html), 2023.
-

- [128] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*.
- [129] A. Agarwal and A. Lavie, “Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output,” in *Proceedings of the Third Workshop on Statistical Machine Translation*, 2008, pp. 115–118.
- [130] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out*, 2004, pp. 74–81.
- [131] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.
- [132] M. Bhalekar and M. Bedekar, “D-cnn: A new model for generating image captions with text extraction using deep learning for visually challenged individuals,” *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8366–8373, 2022.
- [133] S. Srivastava, H. Sharma, and P. Dixit, “Image captioning based on deep convolutional neural networks and lstm,” in *2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*. IEEE, 2022, pp. 1–4.
- [134] J. Lu, J. Yang, D. Batra, and D. Parikh, “Neural baby talk,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7219–7228.
- [135] A. Graves, “Generating sequences with recurrent neural networks,” *arXiv preprint arXiv:1308.0850*, 2013.
- [136] A. Sethi, A. Jain, and C. Dhiman, “Image caption generator in hindi using attention,” in *Advanced Production and Industrial Engineering*. IOS Press, 2022, pp. 101–107.
- [137] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, “Attention on attention for image captioning,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4634–4643.
- [138] R. Solomon, M. Abebe *et al.*, “Amharic language image captions generation using hybridized attention-based deep neural networks,” *Applied Computational Intelligence and Soft Computing*, vol. 2023, 2023.
- [139] T. Zhang, T. Zhang, Y. Zhuo, and F. Ma, “Catanic: Automatic generation model of image captions based on multiple attention mechanism,” 2023.
- [140] J. Aneja, A. Deshpande, and A. G. Schwing, “Convolutional image captioning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5561–5570.

- 
- [141] J. Gu, G. Wang, J. Cai, and T. Chen, “An empirical study of language cnn for image captioning,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1222–1231.
- [142] T. Yao, Y. Pan, Y. Li, and T. Mei, “Exploring visual relationship for image captioning,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 684–699.
- [143] X. Yang, K. Tang, H. Zhang, and J. Cai, “Auto-encoding scene graphs for image captioning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 685–10 694.
- [144] Y. Pan, T. Yao, Y. Li, and T. Mei, “X-linear attention networks for image captioning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 971–10 980.
- [145] X. Zhang, X. Sun, Y. Luo, J. Ji, Y. Zhou, Y. Wu, F. Huang, and R. Ji, “Rstnet: Captioning with adaptive attention on visual and non-visual words,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 465–15 474.
- [146] J. Ji, Y. Luo, X. Sun, F. Chen, G. Luo, Y. Wu, Y. Gao, and R. Ji, “Improving image captioning by leveraging intra-and inter-layer global representation in transformer network,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 2, 2021, pp. 1655–1663.
- [147] Y. Wang, J. Xu, and Y. Sun, “End-to-end transformer based model for image captioning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2585–2594.
- [148] J. C. Hu, R. Cavicchioli, and A. Capotondi, “Expansionnet v2: Block static expansion in fast end to end training for image captioning,” *arXiv preprint arXiv:2208.06551*, 2022.
- [149] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” *arXiv preprint arXiv:2301.12597*, 2023.
- [150] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, “Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 318–23 340.
- [151] C. Li, H. Xu, J. Tian, W. Wang, M. Yan, B. Bi, J. Ye, H. Chen, G. Xu, Z. Cao *et al.*, “mplug: Effective and efficient vision-language learning by cross-modal skip-connections,” *arXiv preprint arXiv:2205.12005*, 2022.
- [152] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, “Meshed-memory transformer for image captioning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 578–10 587.
-



- [153] Z. Fang, J. Wang, X. Hu, L. Liang, Z. Gan, L. Wang, Y. Yang, and Z. Liu, “Injecting semantic concepts into end-to-end image captioning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 009–18 019.
- [154] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, “Image captioning: Transforming objects into words,” *Advances in neural information processing systems*, vol. 32, 2019.
- [155] Y. Li, Y. Pan, T. Yao, J. Chen, and T. Mei, “Scheduled sampling in vision-language pretraining with decoupled encoder-decoder network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8518–8526.
- [156] Y. Li, Y. Pan, T. Yao, and T. Mei, “Comprehending and ordering semantics for image captioning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 990–17 999.
- [157] J. Luo, Y. Li, Y. Pan, T. Yao, H. Chao, and T. Mei, “Coco-bert: Improving video-language pre-training with contrastive cross-modal matching and denoising,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5600–5608.
- [158] Y. Pan, Y. Li, J. Luo, J. Xu, T. Yao, and T. Mei, “Auto-captions on gif: A large-scale video-sentence dataset for vision-language pre-training,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 7070–7074.
- [159] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565.
- [160] L. Wu, M. Xu, L. Sang, T. Yao, and T. Mei, “Noise augmented double-stream graph convolutional networks for image captioning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3118–3127, 2020.
- [161] T. Yao, Y. Pan, Y. Li, and T. Mei, “Hierarchy parsing for image captioning,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2621–2629.
- [162] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, and T. Zhang, “Recurrent fusion network for image captioning,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 499–515.