Generating Image Captions in Amharic Language Using Hybridized Attention-Based Deep Neural Networks

Rodas Solomon W/Mariam

A Thesis Submitted to the department of Computer Science and Engineering,
School of Electrical Engineering and Computing

Office of Graduate Studies
Adama Science and Technology University

June, 2022

Adama, Ethiopia

Generating Image Captions in Amharic Language Using Hybridized Attention-Based Deep Neural Networks

Rodas Solomon W/Mariam

Advisor: Mesfin Abebe (Ph.D.)

A Thesis Submitted to the department of Computer Science and Engineering
School of Electrical Engineering and Computing

Office of Graduate Studies
Adama Science and Technology University

June, 2022

Adama, Ethiopia

DECLARATION

I declare that this Master's Thesis entitled "**Generating Image Captions in Amharic Language Using Hybridized Attention-Based Deep Neural Networks**" is my own work and has not been submitted to any university for similar purpose. The references used in this thesis are duly recognized by proper citations.

Rodas Solomon	_____	_____
Name of student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have closely advised the student while developing this thesis and read the draft thesis entitled "**Generating Image Captions in Amharic Language Using Hybridized Attention-Based Deep Neural Networks**" prepared under my guidance by Rodas Solomon. Therefore, I recommend the submission of the thesis to the department for further review and evaluation.

Mesfin Abebe (Ph.D.)	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-advisor	Signature	Date

APPROVAL PAGE

I hereby certify that the recommendation and suggestion given by the thesis review committee are appropriately incorporated into the final thesis entitled "**Generating Image Captions in Amharic Language Using Hybridized Attention-Based Deep Neural Networks**" by **Rodas Solomon**.

Mesfin Abebe (Ph.D.)	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-advisor	Signature	Date

APPROVAL PAGE OF BOARD OF REVIEWERS

We, the undersigned, members of the Board of Reviewers of the thesis open defence by **Rodas Solomon** have read and evaluated his thesis entitled "**Generating Image Captions in Amharic Language Using Hybridized Attention-Based Deep Neural Networks**" and assessed the understanding of thesis is accepted and we recommend the implementation of the thesis.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

First of all, I thank Almighty God for helping me reach this great opportunity in peace and health. Next, I would like to thank my dear family for their unwavering support, morale, prayer, and financial support.

I would like to express my deepest gratitude to my advisor, Mesfin Abebe (Ph.D.), who has been by my side to help me achieve this great opportunity. His persistent constructive opinions have given me a good study understanding.

A special thanks to Yared Tekalign and members for the translation work they provided at an affordable price and on time for this thesis. I also want to thank all my friends who stayed this long process with me, always offering support and love. Also, I extend many thanks to staff members in ASTU for their kind support during my research study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF CODE SNIPPET

LIST OF ACRONYMS

1G, 2G	1 Gram, 2 Gram
3G, 4G	3 Gram, 4 Gram
3D	Three-Dimensional
AM	Attention Mechanism
AMT	Amazon Mechanical Turk
API	Application Programming Interface
Bi-GRU	Bidirectional Gated Recurrent Unit
Bi-LSTM	Bidirectional Long Short-Term Memory
BLEU	Bilingual Evaluation Understudy
BRNN	Bidirectional Recurrent Neural Network
C1 and C2	Caption 1 and Caption 2
CNN	Convolutional Neural Networks
CNTK	Microsoft Cognitive Toolkit
CPU	Central Processing Unit
DL	Deep learning
DNN	Deep Neural Network
DSEN	Dense Semantic Embedding Network
EE-LSTM	Element Embedding-Long Short Term Memory
EXPER	Experiment
GAN	Generative Adversarial Network
GB	Gigabyte
GloVe	Global Vectors
GPU	Graphics Processing unit
GRU	Gated Recurrent Unit

HES	Human Evaluation Study
IDE	Integrated Development Environment
JPEG	Joint Photographic Experts Group
LDA	Latent Dirichlet Allocation
LSTM	Long Short-Term Memory
METEOR	Metric for Evaluation of Translation with Explicit Ordering
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSCOCO	Microsoft Common Objects in Context
MT	Machine Translation
NLP	Natural Language Processing
NMT	Neural Machine Translation Systems
NN	Neural Network
RAM	Random Access Memory
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RGB	Red-Green-Blue
RQ	Research Question
SC-NLM	Structure-Content Neural Language Model
TB	Terabyte
TPU	Tensor Processing Unit
VGGNet	Visual Geometry Group network
Vs-code	Visual-Studio Code

ABSTRACT

More than thirty million Ethiopian people speak Amharic, and Amharic is a family of Semitic languages. This research aims to generate image captions using the Amharic language based on the deep learning approach. Generating captions in the Amharic is used for different application tasks, primarily to help people with visual impairments to understand the content image. Although experiments with captions in English and different languages have been carried out in the past years, generating captions using the Amharic has not been widely studied. Recently, image captions have moved by encoder-decoder deep learning approaches. Generally, this approach uses previously trained CNN models to take out visual features of the picture and represent a specific dimension through the encoder. And passes these features to a decoder (such as LSTM or GRU) to generate captions. To improve approaches that use encoder-decoder various experiments have been carried out over the past few years using different methods. The AM and bidirectional language decoder are one of these methods to play an key role in producing better captions. Bridging the modality gap between the visual and textual features is one of the significant challenges in these previous methods to *enhance the semantically correct caption. In this case, the gap between these two features maximizes the models dose not predict proximity words to define the picture content.. To address this issue, this study proposed by hybridized attention-based DNN for generating Amharic cations. The proposed consists of an Inception-V3 CNN encoder to get the image features, Visual attention help to contrate essential areas of the image, and Bi-GRU with an AM as a decoder for language generation. Bi-GRU can learn long-term relationships between vision and textual features effectively compared to Bi-LSTM. For the dataset for the proposed model, we translated the English version of the Flickr8k dataset into Amharic and performed experiments. As the result of the hybridized attention-based approach shows, we get better results on 1G-BLEU, 2G-BLEU, 3G-BLEU, and 4G-BLEU, which are 60.6, 50.1, 43.7, and 38.8, respectively. In addition, the proposed model achieved better results in experiments using BNATURE and Flickr8k (English) datasets. And the result demonstrated the significance of the proposed approach to compare with the baseline models. It is 0.21% and 0.21% higher than CNN-Bi-GRU and Bag-LSTM on the 4G-BLEU matrix.*

Keywords : *Amharic Image Caption, Encoder-decoder, Bi-GRU, Attention Mechanism*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Artificial intelligence applications are now widely used, and one of their main contributions is to simplify human activities and also the ability to perform various complex tasks. Computer vision and NLP are one of the fields in the artificial intelligence, each of which is integrated into many areas of our life and so far, scientists have been conducting various research in these areas. Image captioning task is one of application in these fields that consisting generate concise descriptions for a given picture in the natural language that can capture the shown things (Masotti et al, 2018). And it is also the basic issue in artificial intelligence associated with CV and NLP (Vinyals et al., 2015).

Recent research shows that, although there is a significant improvement in English language image captioning, the study of captioning using the Amharic language is very limited or no study has been done in this area. However, generating a caption from an image is a difficult task to understand. For this reason, it needs a cognition of both the picture and language (textual) domains, which have two distinct feature representations.

As deep learning algorithms become more popular, the process of solving complex problems also increases. The most popular deep learning algorithms are RNNs, CNNs, LSTMs, GANs, Stacked Auto-Encoders, etc. In recently, (Herdade et al., 2019) (Mary et al., 2019) researches have shown that the image caption approach has shifted to encoder-decoder model. The encoder-decoder framework approach is initially prepared for MT (Bai et al., 2018). The researchers have formulated a model for caption tasks based on this MT idea. Overall, this framework works as an input picture encoded into intermediate denotation of the information included within the image and afterward decoded into a descriptive text ordering (Herdade et al., 2019). This approach uses pre-train CNN models through the encoder to obtain visual elements and turn them into a specified length vector. And the decoder LSTM or GRU NNs most utilized for language generation. The problem with this encoder-decoder method is incompetent to squeeze all the essential information of a source into a fixed length vector (Bahdanau et al., 2015). And again, the unidirectional LSTM decoder problem is restricted to having only past information, thus yield poor outcome for long sequences (Agughalam et

al., 2020). As mentioned here (Zakir et al., 2018), unidirectional LSTM is unable to produce contextually well-formed descriptions. In general, most recent methods train visual and textual separately, this leads to a semantic correctness gap between the image and generated text. Those are the gap or problems that we need to address in this study.

In this paper, we investigate to address those problems by using hybridized attention-based DNNs for generating Amharic captions. Our model consists of a pre-train CNN model, Bi-GRU, and an AM for generating an Amharic description from images. Pre-train CNN that offer advantages in extracting the important features from entire the image and Bi-GRU networks allow for better predicting based on past and future data. The advantage of the Bi-GRU with AM allows generating words by focusing only on the necessary information in the Bi-GRU. Now through the AM, we can include all essential information of Bi-GRU into a specified size vector as context.

In this (Zhou et al., 2016) relation classification task, they showed successful work using Bi-LSTM with AM. This concept we used for image caption in this research. We used the Flickr8K dataset to implement our model. By using Google translation, we translate the Flickr8K caption's part into Amharic, and some translations edit manually to describe it more accurately. Finally, For the training again testing of the model, utilized this dataset. And also evaluate by using the most common evaluation metric for captions.

1.2. Motivation of the Study

Applications based on the local language are more accessible to that country than applications that use a foreign language. The official working language of Ethiopia is the Amharic language, which is spoken by nearly 30 million people and is the second-largest told Semitic language behind Arabic (Wondwossen et al., 2014), (Fikre et al., 2020). The motivation to do this study is the current DNNs algorithms, such as pre-train CNNs, Bi-LSTMs, Bi-GRUs, and AM, are recently successful in different sectors. We use these algorithms, and the AM methods applied to this widely spoken Amharic language can be trained to generate image captions.

The machine trained in this language looks at the given image and then produces a description in Amharic text that accurately describes the image. But in this case, the difficulty is to understand the content of the images given to the machine and generating appropriate captions. Solving this problem and generate accurate image captions will remain a key area

of research for researchers. And various methods and experiments have been performed by different researchers to address this problem.

1.3. Statement of the Problem

Most existing attention-based image captions do not combine information from both visual and linguistic content. These AMs use models generate image captioning by using only one of them, either using the image or the text. Therefore, the problem with these models expands the gap between the visual and textual relevant features selection while generating caption (word prediction). This leads to inadequate semantic correctness of the result image caption. To address these challenges and generate a meaningful caption, in this study, a hybridized attention-based model is present.

1.4. Research Questions

This study attempt will be to answer the following research questions.

RQ1. What are the effective DNNs for hybridized attention-based encoder-decoder model to generate captions?

RQ2. What are the constraints that effectively improve the hybridized attention-based DNN image captions?

RQ3. What will be the outcome of the attention-based Bi-GRU decoder regarding improving the qualities of the captions?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of the study is to develop a model that can generating image caption in Amharic language using hybridized attention-based DNNs.

1.5.2. Specific Objectives

The specific objectives are mentioned in the following lines to address achieving the general objective of the study:

- Acquisition and preparation of datasets for Amharic captions.
- To design hybridized an attention-based encoder-decoder DNN.
- To train hybridized an attention-based DNNs with the prepared dataset.

- The trained model is tested by a test set.
- Evaluates the performance of the model.

1.6. Significance of the Study

Image captions are important tasks and practical applications for a variety of image and language-based applications. The image caption is primarily used for the visually impaired (blind people). Since blind people cannot view the image, it allows them to understand the image content using an application that convert caption into speech. On the other hand, for people working in photo and video editing applications, this task allows them to generate image captions efficiently. Finally, caption allows programmers working on search algorithms to search images from a database using only image content. These mentioned above significance of the study, specifically described as follows.

- Assistance for visual impairment (vision loss) : - Visual impairment defines as loss of eyesight (vision), whether it can't see totally or partial eyesight loss (slightly invisible). So, the image captions help people with visual impairments to understand the image content by transforming text into speech applications.
- A recommendation in editing software's :- The captions task automatically accelerates the nearest descriptions method for digitalized editing application products.
- Image searching : - It used to search images from the database by image context. Most image searching retrieve images using textual query primary depend on metadata of images. This type of search can't search just by looking at the given image. So, it is required to search images only if the images with metadata similar to the textual query.

1.7. Delimitation Scope

1.7.1. Scope of the Study

The study concentrates on producing Amharic captions to the given image by using hybridized attention-based DNNs. This DNN is an encoder-decoder framework model and trained and tested with the translated into Amharic dataset.

1.7.2. Limitation of the Study

One of the main challenges with generating image captions in Amharic is the lack of a well-organized dataset in the Amharic language. We translated the English dataset into Amharic to train and test for this study. There were different obstacles during the translation process. For this reason, most English sentences represent foreign cultures, so some words become meaningless when translated into Amharic. Therefore, this challenge and grammatical errors also negatively affect the implementation of the proposed model.

1.8. Organization of the Thesis

The rest section of this thesis is structured as follows:

Chapter Two : Describe an outline of DNNs literature and corresponding works concerning the previous and DNN approaches for an image caption and AM. This chapter also explains the Amharic language, the overview of ML, DL, and different deep learning models.

Chapter-Three : Describe study methods such as dataset collected technique, dataset pre-processing methods, tools for designing, and the evaluation approaches.

Chapter Four : Describe the architecture of the proposed model in detail. CNN image encoder, visual attention, Bi-GRU with AM and its mathematical and pseudocode for proposed model are discussed.

Chapter Five : Describe the proposed model implementation with the setting of a working configuration, implantation techniques for dataset processing, model implementation, described using snip code and visualization graph for the model summary.

Chapter Six : Present the expected proposed model key results, compare the new outcome with the latest baseline models, and describe the significance of the results.
Chapter Seven : The concluding part explains the recaps of the entire work, a summary of the result, and gives directions to the potential of work in the future.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

In recent years, a massive amount of data that generated by both humans and machines. According to (Jayagopal et al., 2021), approximately eighty percent of data captured today is unstructured and gathered from a different source. To collect (selecting) or process (for analysis, sequence) this vast amount of data in human capacity is an impossible and complex task. In this case, AI is involved in addressing this complex issue. AIs are the capacity of a machine's system to mimic human behavior. The power of AI contributes significantly is far better than that of humans at logically and arithmetically correct collecting, processing massive amounts of data. So, using this AI advantage, various AI-enabled systems or applications have been developed for different services over the years.

MLs are a subset of AI that concentrates on getting machines to decisions by feeding them data. The different kinds of ML are as follows:

Supervised learning : the first ML is a learning in which we train that machine using well-labelled data. Labelled data indicates that the output is already known. So the model requires mapping each input data to the output label. Most algorithm utilizes this principle for different tasks such as regression, random forest, classification (SVM, Naive-Bayes), others.

Unsupervised learning : is another type of ML that differs from supervised learning, where we do not need to supervise the model. The model learns from the data and discovers patterns and elements in the data, and returns the output. It employs unlabelled data to train the model, which means there is no set output variable. It performs more complex tasks compere to the supervised approach. K-means clustering algorithms utilize unsupervised learning.

Semi-supervised learning : It utilizes a fusion of supervised and unsupervised learning methods. Its use incorporates a short piece of labeled data with an enormous portion of unlabelled data to train models.

Reinforcement learning : the last classification of ML algorithm is that the agent collaborates with its present circumstances (environment) by trial and error and using feedback from its action and experience. These types of ML algorithms are important when

designing a system. However, ML makes it difficult to analyses or process large amounts of data. Thus, DL designs to address this issue.

DL is a kind of ML that operates the concept of neural networks to solve complex problems. It executes with the use of a NN. DL algorithms have numerous hidden layers, and each one layer is an interconnected node. Each hidden layer can pull elements (features) and transfer them into the following layer. This extraction feature moves from the input layer to the output layer in NN, and this computational progress is known as forward propagation. The reverse process is called backpropagation.. Backpropagation is a process to optimize the weights (parameters) in NN, which NNs learn. DL performed by artificial DNN has accomplished great success lately in numerous essential areas (Fan et al., 2021).

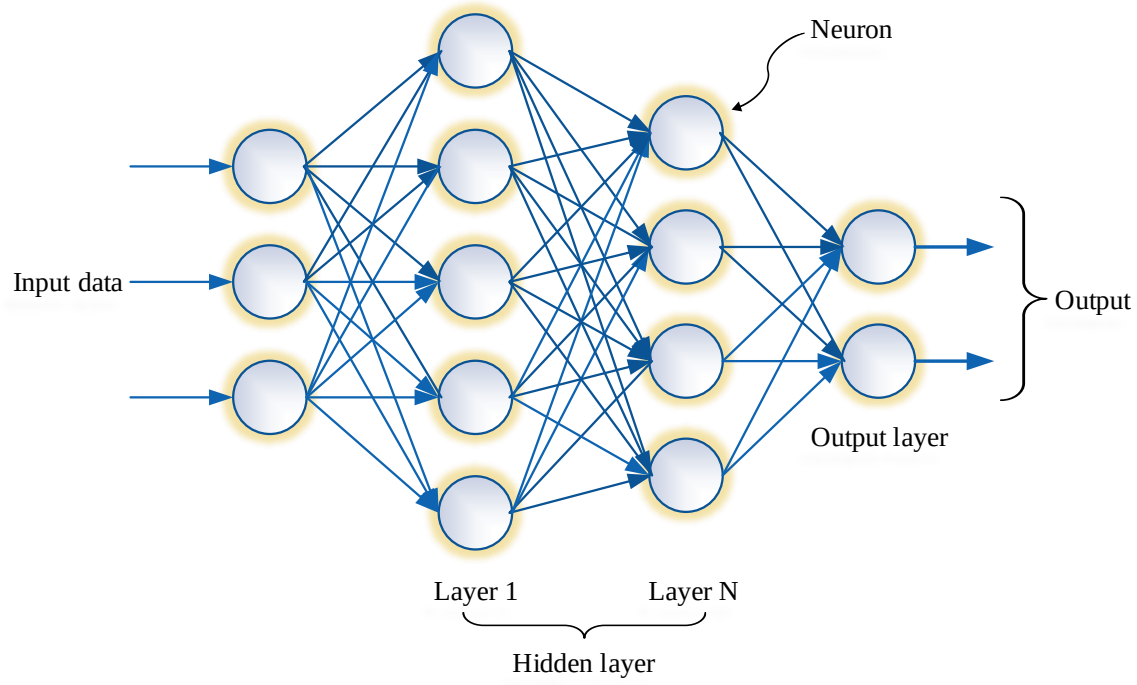


Figure 2.1 A DNN architecture with N hidden (multiple hidden) layers

: (Source)

ANN is a group of collaborative nodes layer called artificial neural, which is created by the perception in a neural layer. In the DL algorithm, there are different types of NN to solve specific complex problems. The most common NNs are**:**

CNN : used primarily in CV and picture categorization. It contained three layers of input, an output, and multiple hidden (Hussain et al., 2019). CNN can detect features and patterns within an image (such as object detection or recognition).

RNN : another type of NN that uses in NLP and computer speech recognition applications. It uses sequential data and time-series data.

Image caption, which objective generates meaningful sentence descriptions for an image, is a challenging task that requires an understanding of both the AI sub-categories (CV and NPL). It not only needs to identify the entities and features from the provided picture but again requires verbalizing them with natural language in the correct order (Yuan et al., 2021). CV, as a subfield of AI that trains the computer to understand the visual world (image or video). CV has become increasingly popular in various areas of research in recent years. And also effective in solving complex problems in different fields. Another hand is the role of the NPL. NPL is a bough of AI that concerns the design and implementation of systems and algorithms able to interact via human language (Lauriola et al., 2021). Therefore, image captioning is a difficult task that involves two different AI subfields into a single task.

2.2. Overview of Amharic Language

Ethiopia is a nation found in the horn of Africa, in the east of the continent, with more than eighty nations and nationalities. Amharic language speaks by more than half of the residents and is the official working language of the FDRE (Bekele et al., 2020) (Fikre et al., 2020). Outside an Ethiopia, the Amharic language speaks in some other nations, especially Eritrea, Canada, the USA, and Sweden (Kesito et al., 2018). It is written using a writing system called Fidel or abugida, adapted from the one used for the now-extinct Ge'ez language (Bekele et al., 2020). Currently, the Amharic alphabet has thirty-three main core characters, each occurring in seven different forms or orders. We listed each one on this Figure 2.2.

So, in general, The Amharic category includes a morphologically rich language but is less resourced (available resources for scientific research) within the Semitic language family (Seyoum et al., 2019). And also, it is one of the least researched and presents a challenge to the area of NLP.

1st	2nd	3rd	4th	5th	6th	7th
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ha	hu	hi	hā	he	hə	ho
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
la	lu	li	lā	le	lə	lo
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
ḥa	ḥu	ḥi	ḥā	ḥe	ḥə	ḥo
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ma	mu	mi	mā	me	mə	mo
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
śa	śu	śi	śā	śe	śə	śo
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ra	ru	ri	rā	re	rə	ro
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
sa	su	si	sā	se	sə	so
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
ḳa	ḳu	ḳi	ḳā	ḳe	ḳə	ḳo
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ba	bu	bi	bā	be	bə	bo
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ta	tu	ti	tā	te	tə	to
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ḫa	ḫu	ḫi	ḫā	ḫe	ḫə	ḫo
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
na	nu	ni	nā	ne	nə	no
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ʿa	ʿu	ʿi	ʿā	ʿe	ʿə	ʿo
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ka	ku	ki	kā	ke	kə	ko
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
wa	wu	wi	wā	we	wə	wo
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ʾa	ʾu	ʾi	ʾā	ʾe	ʾə	ʾo
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
za	zu	zi	zā	ze	zə	zo
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ya	yu	yi	yā	ye	yə	yo
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
da	du	di	dā	de	də	do
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ

ga	gu	gi	gā	ge	gə	go
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ṭa	ṭu	ṭi	ṭā	ṭe	ṭə	ṭo
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
pa	pu	pi	pā	pe	pə	po
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ṣa	ṣu	ṣi	ṣā	ṣe	ṣə	ṣo
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ḍa	ḍu	ḍi	ḍā	ḍe	ḍə	ḍo
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
fa	fu	fi	fā	fe	fə	fo
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
pa	pu	pi	pā	pe	pə	po

Figure 2.2 Amharic core characters

(Source)

2.3. Background: Deep Neural Networks

DNNs are nowadays widely employed for multiple AI applications, including CV, speech recognition, and robotics (Sze et al., 2020). Recently, image caption models using DL algorithms in different techniques have performed successful state-of-the-art results. Following CNN's success, it dedicated the concentration of investigators from different occupations or areas. The fields of CV and NPL, CNN, and other DL models were due to a significant capacity for their outstanding outcomes in many challenging issues (Martínez et al., 2019). Researchers use different DL algorithms to solve complex problems. In our case, generate image descriptions problem. CNN, RNN, LSTM are the most common DNNs algorithms.

2.3.1. Convolutional Neural Networks

In the field of DL, CNNs have dominated the CV and other different areas in recent years. The CNNs have demonstrated excellent performance in many CV and ML problems (Hussain et al., 2019). The basic CNN comprises three layers: namely, input, output, and multiple hidden. The CNN hidden layers generally consist of convolutional, pooling, normalization, and fully connected (Hussain et al., 2019).

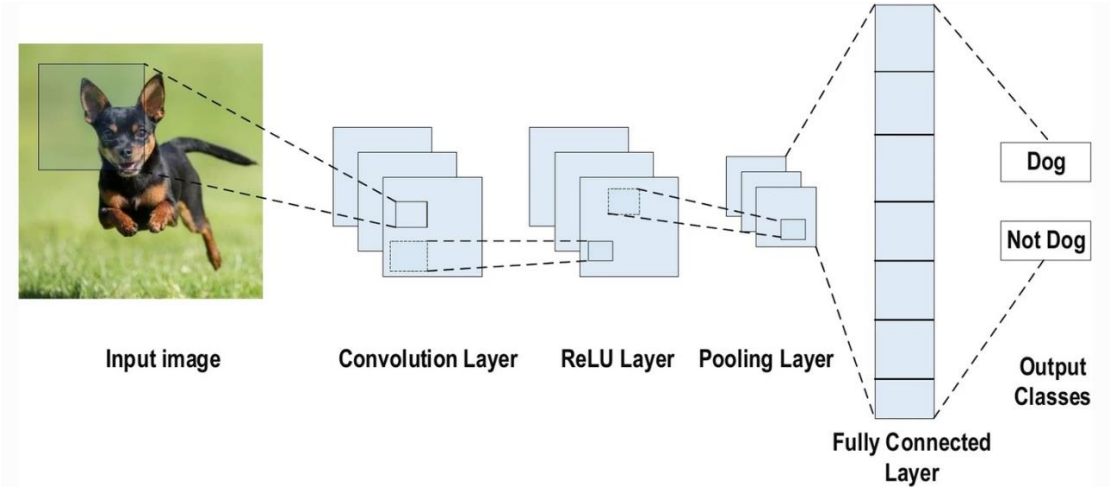


Figure 2.3 Typical CNN architecture for image classification

(Source)

In the first layer (input layer), it stores the input data after data is loaded. This input layer (or denoted by x) describes in three dimensions: width, height (m), and depth (m) or $m \times m \times r$, where r is depth and equal to 3 in an RGB image (Alzubaidi et al. 2021).

The next layer is the hidden layer which includes convolution layers, pooling layers, and activation functions. Convolution layers use to for extracting the features of the image. In the convolution networks, several kernels (filters) are available, denoted by k. The input layer convoluted with a similar dimension ($n \times n \times q$), n needs to be less than m, while q is either equivalent to or less than r. Additionally, the kernel put group weight (w^k), bias (b^k) shared all over the input space, and for generating k feature map H^kas output with a size of $(m - n + 1) \times (m - n + 1)$. The dot product between input and weight values forms the convolution layer calculation as Eq.2.1 (Alzubaidi et al. 2021) Finally, by applying the activation function to the output of the convolution layer.

$$H^k = f(w^k * x + b^k) \qquad 2.1$$

The next layer of the hidden layer directs as the pooling layer or 'max pooling'. It is used to decrease the dimensionality of the feature map but keeps essential information. This leads to reducing the computational complexity, controlling the overfitting issue. Max pooling takes the minimum elements from each sub-region of the feature map. For example, the max-pooling operation for changing a 4x4 feature map output into a 2x2 output with stride 2 describe in figure.

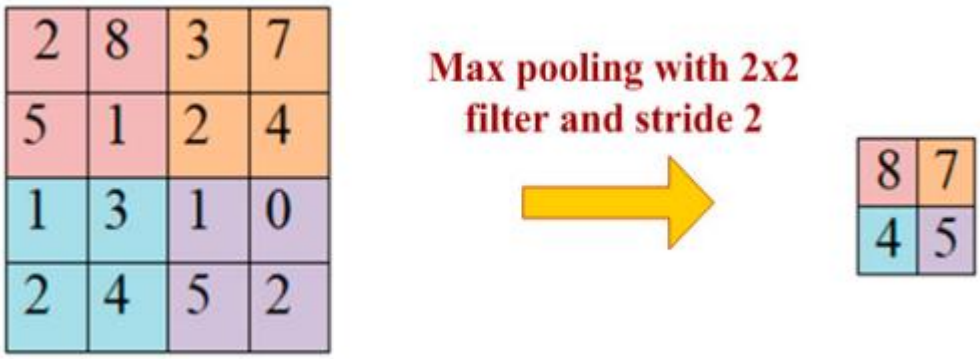


Figure 2.4 Max pooling with filter 2 × 2 and stride size [26]

After each convolution layer in CNN, apply the nonlinear activation function. The sigmoid function, SoftMax function, Tanh function and ReLU function are most typical activation function in DL models. ReLU activation function is the most used, and it involves any negative inputs within the kernelled image to zero.

The last CNN architecture layer is called the classification layer (fully connected layer). The output from the final convolution or pooling layer is input for a fully connected layer, which is flattened (1-dimensional array). This layer computes predicted categories by recognizing the input picture, which integrates all the elements learned by earlier layers. (Jana et al., 2020).

Finally, the last outcome of the fully connected designates the last CNN output. The method to calculate the loss (predicted error) is called loss function. This error shows the difference between the actual output and the predicted one. The gradient descent algorithm utilizes to minimize the training error and repeatedly modifies the network parameters through every training epoch, (Alzubaidi et al., 2021). The mathematical expression of this process is shown in Eq.2.2 (Alzubaidi et al. 2021).

$$w_{ij^t} = w_{ij^{t-1}} - \Delta w_{ij^t}, \qquad \Delta w_{ij^t} = \eta * \frac{\partial E}{\partial w_{ij}} \tag{2.2}$$

The last weight in the present training epoch is marked by w_{ij^t}, while the weight in the initial $(t-1)$ training epoch is represented $w_{ij^{t-1}}$. The learning rate is η and the prediction error is E (Alzubaidi et al., 2021).

2.3.2. Recurrent Neural Networks

RNNs represent a specific form of the DL models, which are often used with sequential data (time-series data). It can train sequentially generate by processing actual orderings of information one phase at a time and predicting what reaches next (Graves et al., 2013). RNNs have 2 infusions: present and recent history. Both are essential when RNNs make a decision. It assumes the present intake and outcome that it has retained from the previous input for making the decision.

During the activity of RNNs, the data follows a loop mechanism, which outcomes in enormous updates to NN weight. These huge changes make RNN fall in preserving the gradient called vanishing gradient decent. Generally, RNNs are typically challenging to train due to the well-known gradient vanishing and hard to learn long-term patterns (Li et al., 2018). In Eq.2.3, we demonstrate the calculation for the hidden variables, while Eq.2.4 demonstrates the outcome (output) variable.

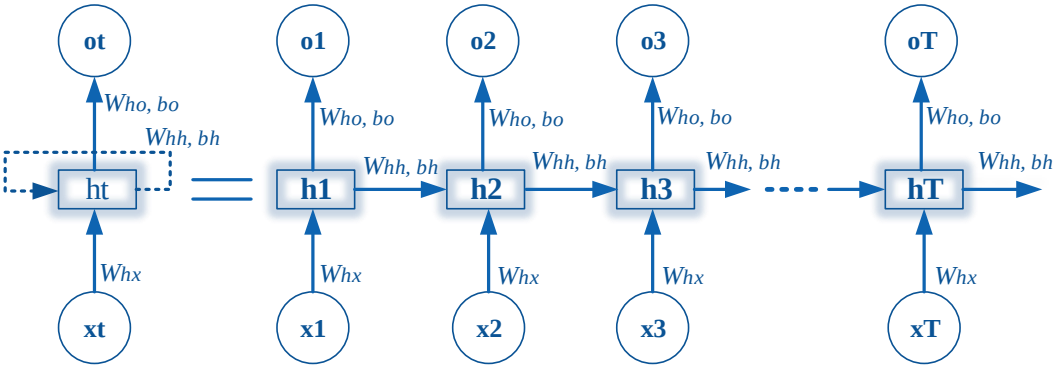


Figure 2.5 A standard RNN architecture

$$h_t = \phi_h(X_t W_{hx} + h_{t-1} W_{hh} + b_h) \quad 2.3$$

$$O_t = \phi_o(h_t W_{oh} + h_{t-1} + b_o) \quad 2.4$$

Where, $X_t \in \mathbb{R}^{n\,x\,d}$ and $h_t \in \mathbb{R}^{n\,x\,h}$ denote the input at time step t and the hidden state, where n is number of samples, d is the number of inputs of individual example and h is the number of hidden units (Schmidt et al., 2019). W_{hx} defines the weight matrix between hidden and hidden recurrent, hidden-state to hidden-state matrix W_{hh}, b_h and b_{ho} a bias parameter, and ϕ represents activation. Since h_t recursively includes h_{t-1} and this procedure ensues for

every time step the RNN contains traces of all hidden states that preceded h_{t-1} as well as h_{t-1} itself (Schmidt et al., 2019).

2.3.3. Long Short-Term Memory

LSTMs are a sort category of RNNs that can understand long-term relationships. It was first presented by Sepp Hochreiter and Jurgen Schmidhuber in 1997 and is broad benefits until now (Yang et al., 2020). The major cause for the advent of LSTM is to address the problem of Long-term dependence and vanishing gradient in RNN by introducing new components, namely memory cells, and gate units. LSTMs have 3 separate entrances (gates) such as forget gate, output gate, and input gate that regulate information flow in LSTM cell. The forget and input entrances address the issue of gradient disappearance and gradient explosion so that long-term data can capture (Yang et al., 2020). The memory of LSTMs is similar to computer memory. Finally, LSTM compared with ordinary RNN, LSTM gives a more reasonable performance in long series input and better memorization of past state.

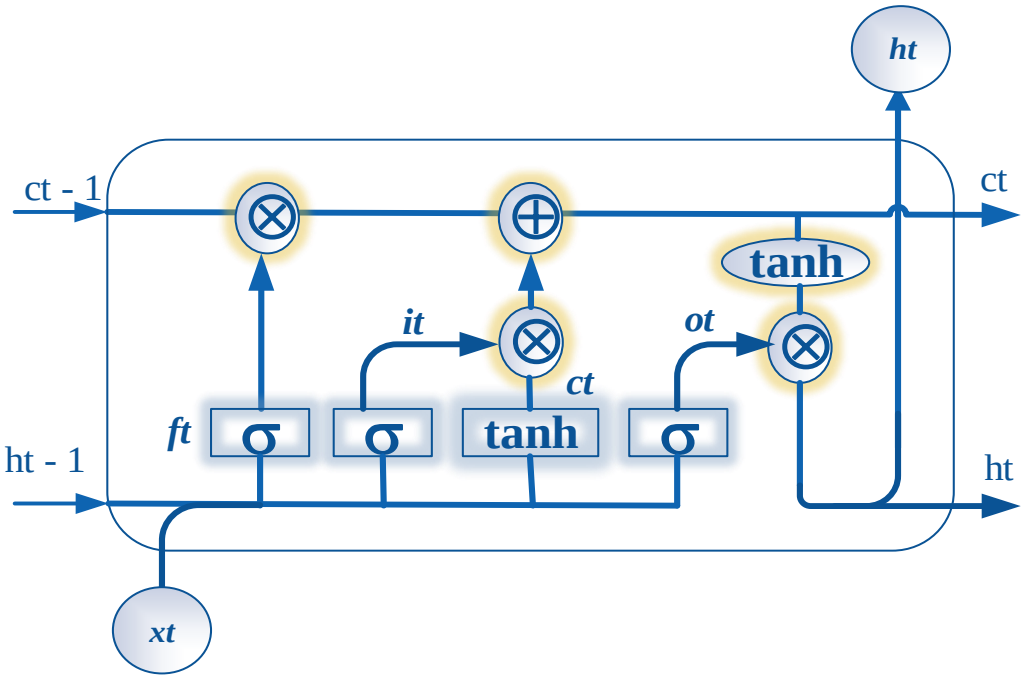


Figure 2.6 LSTM structure

where x_t : input of outer at time step t.

- h_{t-1}, h_t: state of hidden at times $(t-1)$ or t
- C_{t-1}, c_t: state of cell at time step $t-1$ or t.

- f_t: the result of the forget gate
- i_t: the result of the input gate
- $\tilde{\boldsymbol{c}}_t$: gate of cell
- o_t: gate of output, state.
- $W_{xi}, W_{xf}, W_{xo}, W_{hi}, W_{hf}, W_{ho}$ as weight matrices while b_i, b_f, b_o are their respective biases.

The above notations show in (Aden, 2021) to demonstrate the work of the LSTM network.

Every period the LSTMs take to input the X_t, the four gates (f_t, i_t, c_t) and o_t is updated as follows: The computations for these gates utilize the sigmoid activation operation σ to convert the outcome ∈ (0, 1) individually results in a vector with entries ∈ (0, 1) (Schmidt et al., 2019).

$$o_t = \sigma(X_t W_{xo} + h_{t-1} W_{ho} + b_o) \quad 2.5$$

$$i_t = \sigma(X_t W_{xi} + h_{t-1} W_{hi} + b_i) \quad 2.6$$

$$f_t = \sigma\left(X_t W_{xf} + h_{t-1} W_{hf} + b_f\right) \quad 2.7$$

The equations Eq.2.5, Eq.2.6, and Eq.2.7 are presented to show the LSTM gate's work show in (Schmidt et al, 2019).

Next, we need a candidate memory cell ($\tilde{\boldsymbol{c}}_t$) which has an identical computation as the earlier noted gates but instead uses a tanh activation function (Schmidt 2019). The respective computation is shown in Eq.2.8. The Eq.2.8 shows in (Schmidt et al., 2019).

$$\tilde{\boldsymbol{c}}_t = \tanh(X_t W_{xc} + h_{t-1} W_{hc} + b_c) \quad 2.8$$

The old memory content C_{t-1}, which combines with the presented gates, handles how considerably the old memory content keeps to obtain the new memory content $\mathbf{c}_t$. This is demonstrated in Eq.2.9 where ⊙ represents element-wise multiplication (Schmidt et al., 2019).

$$\mathbf{c}_t = f_t \odot C_{t-1} i_t \odot \tilde{\boldsymbol{c}}_t \quad 2.9$$

The final stage is the calculation for the states of hidden h_t and it can be modified utilizing the outcomes from the output gate and the cell state at the current time step as follows: Eq.2.10 (Aden, 2021)

$$h_t = o_t \odot \tanh(\tilde{c}_t) \qquad 2.10$$

Bi-LSTMs are an extension of LSTM that can better memorization of the future state. This NN contains LSTM units that perform in both directions to integrate past and future context data (Jang et al., 2020). Unlike the LSTM neural, the Bi-LSTM neural uses the two hidden states aggregated input information in two directions (forward and backward) to maintain (capture) information from both future and past. It uses text categorization. A structure of the Bi-LSTMs network shows in the Figure 2.7. Let X_t as input at time t and, the hidden state output by forward LSTM be $\vec{h}_t$ and the output by backward LSTM be $\overleftarrow{h}_t$, $\oplus$ demote element-wise sum, the hidden state output h_t calculation show in Eq.2.11

$$h_t = \left[\vec{h}_t \oplus \overleftarrow{h}_t\right] \qquad 2.11$$

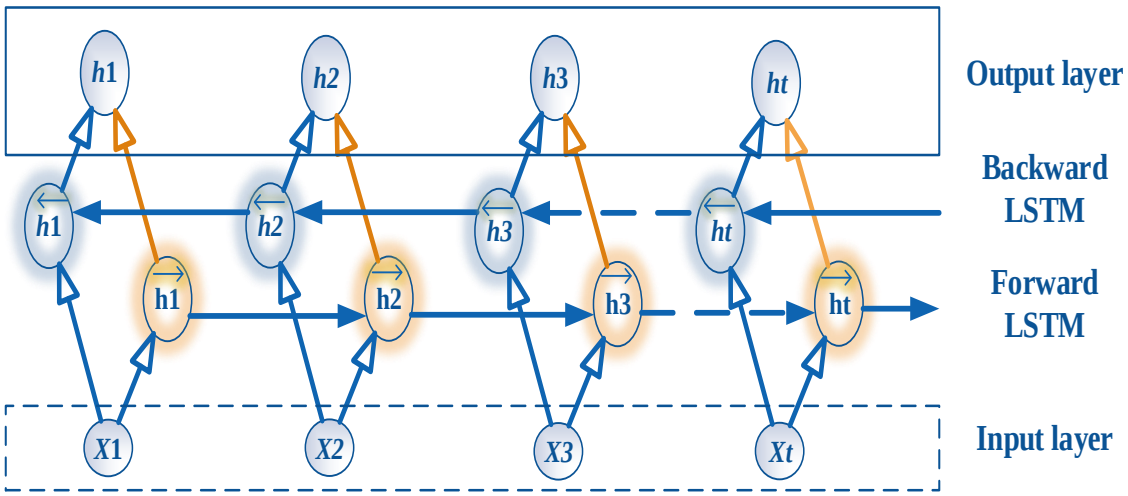


Figure 2.7 Bidirectional Long Short-Term Memory (Bi-LSTM)

2.3.4. Gated Recurrent Unit

GRUs are the particular variety of RNNs that aims to address the vanishing gradient issue of RNNs. To solve this problem, GRU uses two entrances; the update gate and the reset gate. GRU's update gate desires how much data from the earlier units must be forwarded (Humaira et al., 2021). The reset entrance (gate) operates by the model to decide how considerably data

from the last unit to omit. The computations for these gates are shown in Eq.2.12 and Eq.2.13.

Current memory content is utilized to keep the appropriate information from the earlier units (Humaira et al., 2021). And the final step is the last memory (final memory) at the present (current) unit time phase. It is a vector utilized to hold the last data for the present unit and pass it to the following layer (Humaira et al., 2021). The current memory content and final memory at the current unit from the calculation shown in Eq.2.14 and Eq.2.15.

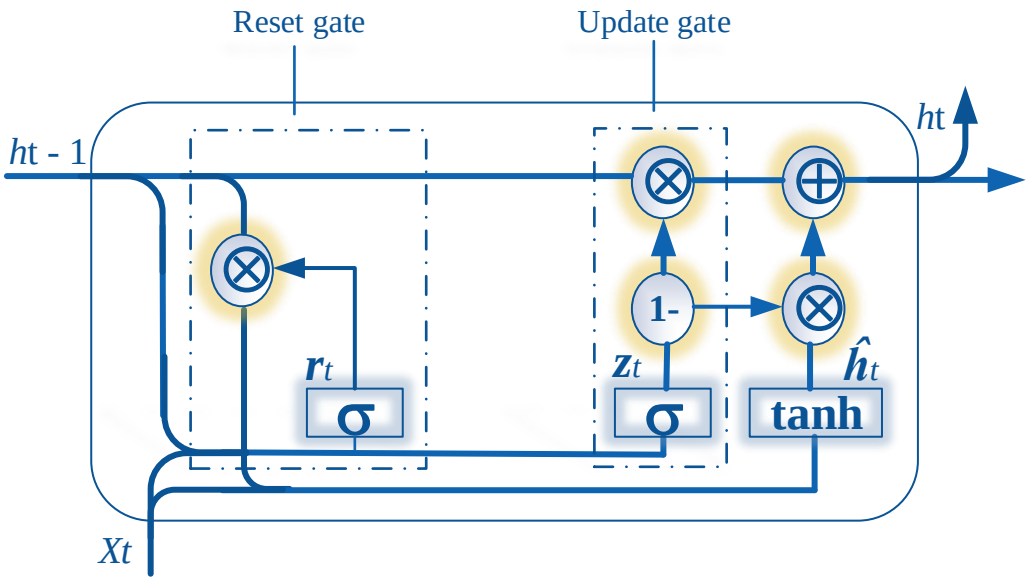


Figure 2.8 The architecture of GRU cell

where x_t: external input at time step t.

- h_{t-1}: Information from previous unit.
- $\tilde{h}_t$: Current memory content.
- h_t: A memory of the final at the current unit.
- z_t: The output of the update gate at the current timestamp.
- r_t: The output of the reset gate at the current timestamp
- W, W_z and W_r: is weight at current unit, weight matrix at update gate and weight matrix at reset gate.

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \qquad 2.12$$

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \quad 2.13$$

$$\tilde{h}_t = tanh(W.[r_t * h_{t-1}, x_t]) \quad 2.14$$

$$h_t = (1 - z_t) * h_{t-1} + z_t \tilde{h}_t \quad 2.15$$

The above equations demonstrate the work GRU's gates show in (Humaira et al. 2021).

The traditional unidirectional GRU is unable to take the pass and the feature information at the same time. The reverse sequence or future information is an important task that uses sequence data. The Bi-GRU NN is a variant of the unidirectional GRU in that the input series passes via a forward and a backward NN, and then the results of the two are merged in the same output layer (Li et al., 2020). In Eq.2.16, 2.17 and 2.18 adopted from (Lv et al., 2020), we can see the computation for the state of hidden at the moment t (h_t) will be acquired by the combination of hidden states $\vec{h}_t$ and $\overleftarrow{h}_t$ generated by the forward and the backward GRU at the time t. let x_t as external input at time step t, forward and backward GRU unit donated by $\overrightarrow{GRU}$ and $\overleftarrow{GRU}$, and h_{h-1} and h_{h+1} denoted as information from the previous and future unit.

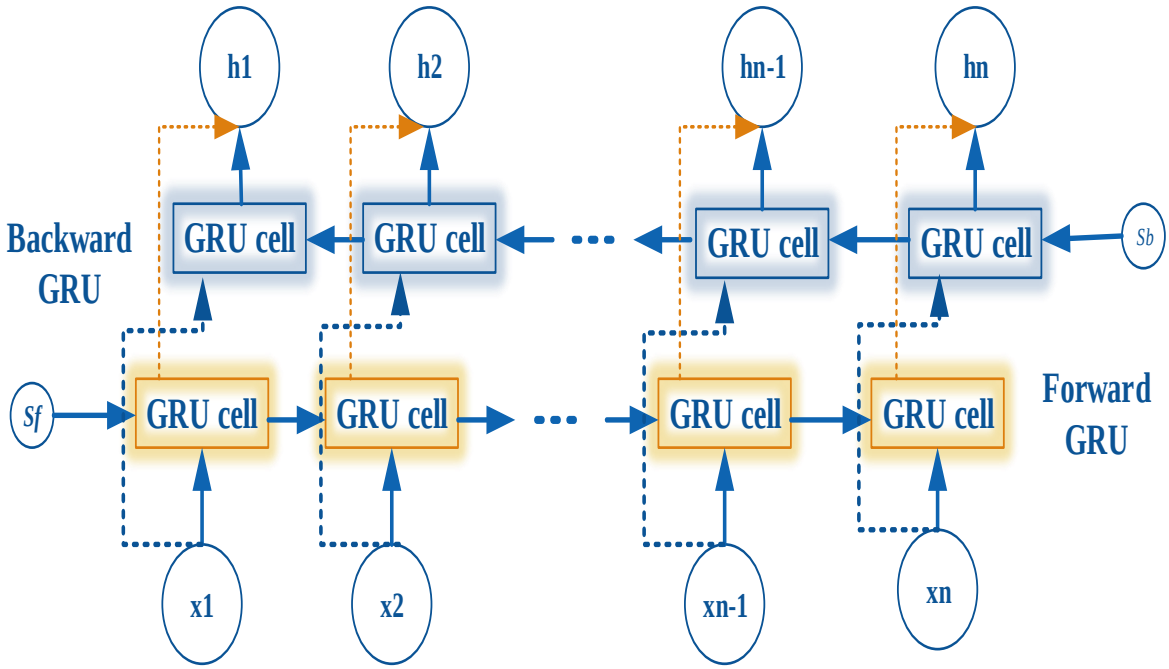


Figure 2.9 The architecture of the Bidirectional GRU

$$\vec{h}_t = \overrightarrow{GRU}(h_{t-1}, x_t) \quad 2.16$$

$$\overleftarrow{h}_t = \overleftarrow{GRU}(h_{t+1}, x_t) \quad 2.17$$

$$h_t = concatenate(\vec{h}_t, \overleftarrow{h}_t) \quad 2.18$$

These equations and notations show in (Lv et al. 2020).

2.4. Attention Mechanism

The recent trend in DL models uses AMs. It is a significant concept in the DL sub-fields, such as NPL or CV. Attention is motivated by the biological procedures of humans that watch to concentrate on specific components when processing extensive amounts of information (Niu et al., 2021). The AM in DL is also an attempt to implement same as in human biological procedures by attending to a few relevant things when processing the data. AMs or their variants are widely used in various application domains.

AM was presented by (Bahdanau et al., 2015), to enhance NMT by choosy concentrate on elements of the source sentence during translation (Niu et al., 2021). The previous approaches of the sequence-to-sequence models have drawbacks to understanding long input or complex sentences and selecting words more relevant when generating or translating sentences. AMs address those issues by helping memorize long input sentences and selecting all the relevant information in the input sentence in NMT. Based on this point, different variations of AMs have become an increasingly common ingredient of deep learning models. In the image captioning tasks, various types of AMs were applied. And those mechanisms are utilized to provide more concentration to the appropriate position in the image or textual data during generating captions to have better description quality.

2.5. Previous Image Captioning Approaches

This section reviews the major types of previous captions method. They are retrieval-based and template-based.

2.5.1. Retrieval-based Approaches

The retrieval strategy employs a searching method to encounter proper picture descriptions (Liu et al., 2019). In retrieval-based approaches, to generate an appropriate caption for the

given image. First, discover the visually identical pictures (similar image content) with their descriptions from the image-caption database and these descriptions are called candidate descriptions. Finally, the descriptions for the given pictures specified from these captions pool (a set of existing captions).

(Kuznetsova et al., 2012) follows retrieval-based image captioning. This method begins retrieving relevant candidate descriptive phrases from the database. Then generate a coherent caption based on the text descriptions of the retrieved similar image sets. (Ordonez et al., 2011) A proposed retrieve and transmit captions from a dataset to a query image using global picture representation. Later on, (Kuznetsova et al., 2013) presented an image caption generalization method to reduce the information gap between image and text compared to previously retrieval-based image caption approaches. In general, The retrieval-based approach needed extensive data to cover every likely query picture (Agughalam et al., 2020).

2.5.2. Template-based Approaches

Template-based methods have specified templates with some empty slots to caption generation (Zakir et al., 2018). In these approaches: first, the object, attributes, action are detected by using detection algorithms, and then the blank space in the template are filled. A predefined fixed number of spaces in a template fill with acquired entity attributes and relations to captions. (Hutchison et al., 2010) proposed a triplet of scene components (object, action, scene) to load the template space for captions. These triplets provide a general idea of what the image is to generate a caption. (Yang et al., 2011) proposes a quadruplet template that includes (Nouns-Verbs-Scenes-Prepositions) for generating image caption. (Li et al., 2011) used a picture identification method that extracted vision data (objects, attributes, and spatial relationships). Then, encodes it as a group of triplets ((adj1, obj1), prep, (adj2, obj2)).

Template-based approaches can generate grammatically correct descriptions compared to retrieval-based methods. However, this approach is rigid and cannot generate variable length descriptions (Martínez et al., 2019)(Zakir et al., 2018). Therefore, the outcome captions lack naturality compared with human written sentences.

2.6. DNNs for Image Captioning

The previous image caption techniques and CNN success in image classification tasks have fuelled the researchers that used DNN for image caption tasks. DNNs methods still influence previous image caption methods. (Kiros et al., 2014) uses a deep CNN as an encoder for extracting picture features, and these features project into the LSTM hidden states. LSTM uses a multimodal space for leaking a joint image-text embedding. The authors introduce a new NN language model as a decoder called SC-NLM. It used the combination of content vector and structure to word generations.

(Vinyals et al., 2015) propose encoder-decoder approaches are the same method as (Kiros et al., 2014). The authors use pre-trained CNN (Inception-V1) for image features, and these features and word embedding pass into the LSTM decoder for generating image captions. The LSTM decoder uses the initial state to include image information and then the next words predicted. This step continues until it obtains the end of the token of the sentence. (Kiros et al., 2014) and (Vinyals et al., 2015) use the encoder-decoder approach, these approaches performed better compared to Template-based and retrieval-based. Researchers have used various enhancement methods to improve the encoder-decoder NN framework and generate semantically richer sentence captions. Those are: adopting the spatial AM and utilizing semantic embeddings improve the semantic richness and bridge the gap between picture and textual information modalities of captions (Agughalam et al., 2020).

(Zhang et al., 2020) uses a semantic embedding approach to bridge the modality gap between pictures and textual data. This approach contains two parts: the first part is the combination CNN-LSTM image caption model was utilized to predict the picture description for the global picture and all the object parts. The authors use R-CNN to detect the object regions. Then all descriptions transform into semantic words that contain detailed element information. In the second role, the authors introduce the EE-LSTM language model, which are embeds both visual and semantic element features to generate image captions.

(Peng et al., 2019) uses extracted scene information from the image and textual data to generate more semantically precise descriptions. This approach is as follows: the first step is image representation. The authors used pre-train ResNet (Sangeetha et al., 2006) and Places365 CNN for global and scene feature extraction and trained by MLP to predict image scene. To extract scene information from the text corpus using LDA. Secondly, the next step

is scene factors which are the image scene and sentence scene vector fuse into double LSTM for generating captions. Double LSTM also helps describe the content picture with a better accurate and good semantic understanding of the image effectively (Peng et al., 2019). Their approach improves the semantic richness of caption with scene factor.

Later on, (Wang et al., 2018) and (X. Xiao et al., 2019) utilized bidirectional LSTM methods to take benefit of both pass and future context data in the series during predicting words. In their methods, they use a pre-trained CNN model extracting image features and those features passed into the deep Bi-LSTM to use in both directions (forward and backward). Based on their proposed results, which means (multimodal Bi-LSTM and Bi-LSTM with incorporated into the DSEN), the Bi-LSTM architecture improves generating longer word sequences. And it's a high capacity for knowing long-term vision and text by taking both the past and future context.

Over the years, various methods have been presented to handle (solve) the issue of captions. Xu et al., 2015 was the first to present the AM in the image captions tasks and also first initialized the hidden state of language LSTM with visual feature vector (Qin et al., 2019). For image captioning tasks, this AM weights CNN features on the salient regions or parts of the image based on a relevance score. Then these features are directly computed into an LSTM decoder with word embeddings. Those weighted features dynamically update at each time step during generating description. (Xiao et al., 2019) added the AM with two separate LSTM for generating image captions. The first LSTM with visual attention is utilized to capture appropriate information from the global picture features and context words, and the second LSTM uses for the language generation module. The authors employed visual attention as it utilized a visual sentinel.

(Lu et al., 2017) also adopted an adaptive encoder-decoder framework that generates the next word by automatically deciding when to look(sentinel get (Lu et al., 2017)) and where to look (spatial attention (Lu et al., 2017)). The authors extend the LSTM decoder to store both long and short visual and textual information called visual sentinel. (Cao et al., 2019) utilized AM to high-level semantic (summary description of the pictures) to improve the image captioning. In their approach, AM deploys on salient semantic element selection rather than focusing on the entire image vector while generating an image description. The authors use VggNet-16 to obtain global image features. And these features are fed into the semantic attention model with word embedding. lastly, to generate the captions of an image, those

text-related image features of attention enter into the bidirectional LSTM model. The semantic AM has significantly improved the relationship between image and text. Finally, their outcomes demonstrate that the implementation of attention further enhances image captioning. The significance of visual and textual attention for captions is to reduce the modality gap between the text and picture elements (Agughalam et al., 2021).

2.7. Related Works

In the previous section, the early image captioning approaches mean not using deep learning techniques (template-based and retrieval-based methods) and using DNN based approaches were discussed for generating image captioning. Now let's look at the corresponding work and drawbacks of the captions based on DNN.

Image captions with semantic embedding have demonstrated remarkable outcomes to produce complex image captions. (Zhang et al., 2020) utilizes semantic representation of images to enhance image caption. The authors argue global image features representations are not adequate to describe all the important elements in the images. They used R-CNN to depict important elements in the image. R-CNN to detect object regions and predict image descriptions. So in the first part, the whole image descriptions predict by CNN-LSTM combined with regional captions, and then all captions are segmented into semantic words (semantic features). Finally, they introduce new element embedding LSTM models that use CNN features and semantic features to predict image descriptions. Their approaches produced detailed image captions that used semantic embedding. However, the semantic element information is statically represented without considering relevant image features.

(Peng et al., 2019) attempted an image semantic understanding model by combining scene factors. The authors use two pre-trained CNN (ResNet and Places365) to extract the global features of the image and the features of the deep scene. LDA models use to train and extract scene information from large text volumes. Finally, both visual and textual extracted are fused into double LSTM to perform better in the semantic caption. This approach has some limitations: since the model trains in unidirectional LSTM always contain only pass information and do not include feature information. And also the feature repetition during word predicts (Agughalam et al., 2021).

(Pedersoli et al., 2017) presents an area attention-based encoder-decoder model that associates parts of the picture (beads on CNN activation grids, object proposal, and spatial

transformers) with the words of a description. (Chen et al., 2017) presented a combination of the spatial and channel-wise AM over a 3D CNN features map for generating image caption. Unlikely (Pedersoli et al., 2017) the authors used features extracted from different channels and multiple layers instead of only uses from spatial locations. Channel-wise attention deploys as the strategy of picking semantic characteristics on the request in the context of a sentence. Their result is an improved image caption. And also a good understanding of what spatial and channel-wise attention looks like in a CNN network that involves captions generations. However, their approach focuses on selecting essential regions of the picture for the language model, but they largely ignore the relevant part of sentences while generating the descriptions. And again, the LSTM language model limited retain future inputs information during model training. (Cao et al., 2019) proposed generating image caption by using the Bi-LSTM with an AM. This AM employed as semantic guidance is infusion into each unit of the Bi-LSTM.

The aforementioned studies and most of the image captions tasks are based on the English language. English, as morphologically poor language, usually follows a set word ordering (subject-verb-object) (Avramidis et al., 2008). The deep learning models made in the English language give better results but have a weaker outcome on morphology-rich languages (German, Bangla, Amharic, and others).

(Jishan et al., 2020) tried image captioning in the Bangla language. The authors followed the hybrid encoder-decoder architecture to generate image captioning. This encoder-decoder framework has three components: the first is the CNN encoder for extracting image features, the second is LSTM and BRNN as language decoders to generating captions. Also, the authors contributed creation of a new dataset. This dataset is called BNLIT and contains 8700 images with one annotation. And these images are collected representing the culture of Bangladesh, and the description is in the Bangla language.

(Faruk et al., 2020) proposed an encoder-decoder model for caption in the Bangla language. The authors used Inception-V3 as a CNN encoder to extract the visual elements from images, and through the language decoder, they used Bi-GRU to generate image text. The authors argue GRU has more efficiency and better results compared to LSTM. Also, they proposed a new dataset called BNATURE. This dataset is configured based on flicker8k and contains 8000 images with five Bengali captions.

In general, those methods (Faruk et al., 2020), (Jishan et al., 2020) gave good results with BLEU and METEOR scores. However, their method limitation is the image feature extraction directly passes into the language model without attending to some relevant features of the image.

In these previous studies, different image caption approaches present. The recent studies show that the image caption approach using Bi-GRU, BI-LSTM and an AM has improved the accuracy of image captioning. For this study, we follow the studies that use Bi-LSTM, Bi-GRU and an AM. Bi-LSTM can outline long-range visual and text relations from forwarding and backward directions (Wang et al., 2016). AMs enable the model to attend to the essential elements of the visual-textual while generating image captioning. The main problem is that these Bi-GRU and AM presented studies used only textual or visual content to generate image descriptions. In this proposed work, words are generated depending on the content of the required information of the visual-textual. Therefore, to address this dependency problem, we have combined or concatenated the textual and visual attention features and fed into a Bi-GRU with an AM to generate image caption words that know both their content. In the table below, we have summarized the gaps and methods used by the previous ones.

2.8. Summary of Related Work

The following tables show earlier works on image captions.

Table 2.1 Related work summary

Authors	***Title***	***Methods Used***	***Gap***
(Zhang et al., 2020)	Image captioning via semantic element embedding	Semantic embeddings approach	– The semantic elements information is statically represented without considering relevant image features.
(Peng et al., 2019)	Image caption model of double LSTM with scene factors	Double LSTM with scene information	– Unidirectional LSTM is limited to capturing only the past information, not considering the future information. – Feature redundancy during words generated.
(Pedersoli et al., 2017)	Areas of Attention for Image Captioning	Attention-based CNN-GRU encoder-decoder approach	– The AM does not consider textual features while generating captions. – Unidirectional GRU limited to retaining future information.

(Chen et al., 2017)	SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning	Attention-based CNN-LSTM encoder-decoder approach	– The language model lacks textual attention while word generating. – And it is limited to retaining future contextual information.
(Cao et al., 2019)	Image Captioning with Bidirectional Semantic Attention-Based Guiding of Long Short-Term Memory	CNN and Bi-LSTM with AM model	– The visual element obtains from the dense layer for an AM. So, this attention is limited to looking at different regions of the picture.
(Faruk et al., 2020)	Image to Bengali caption generation using deep CNN and bidirectional gated recurrent unit	CNN-Bi-GRU encoder-decoder approach	– Lack of visual AM and text related image feature during word generating.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

This chapter describes the procedures from the dataset to evaluation measures used in the proposed work to address the study question. Let's look at the overall procedure of the Amharic image captions proposed model.

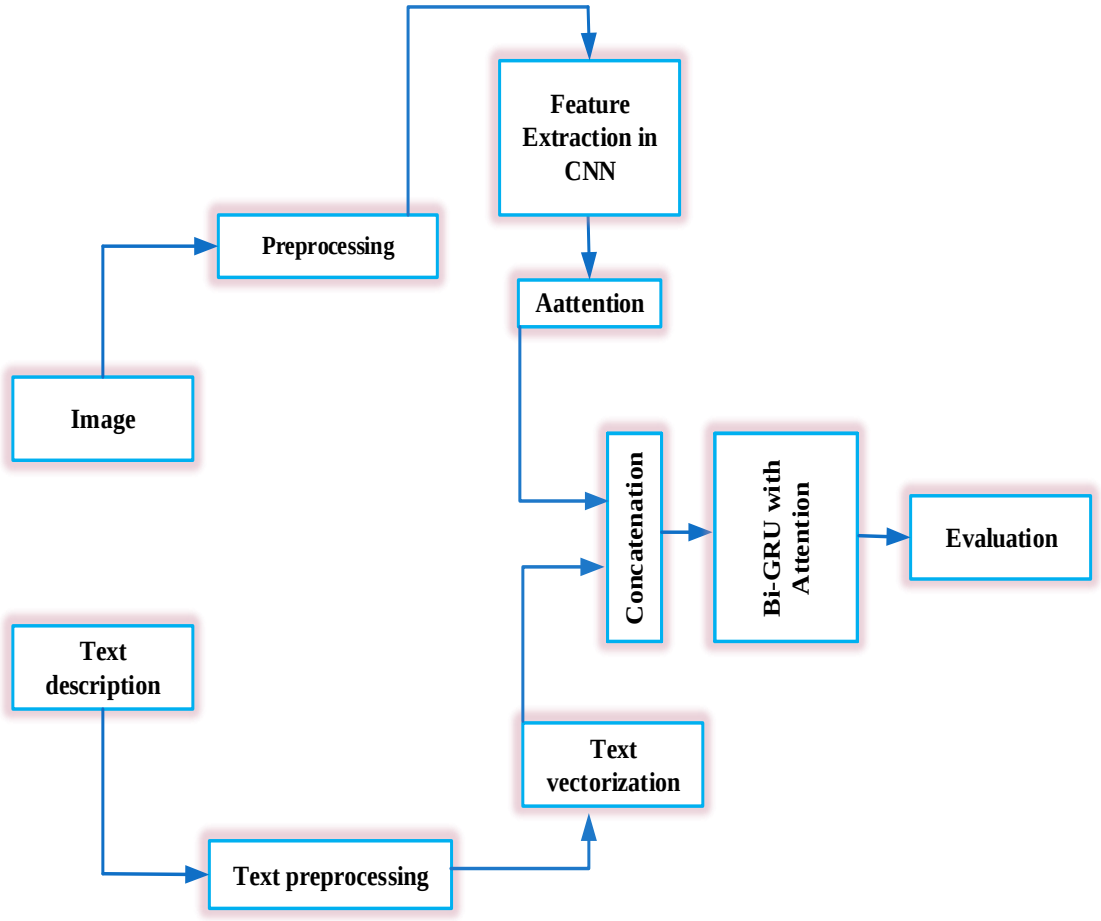


Figure 3.1 Shows the procedure of Amharic image captioning

3.2. Dataset

Dataset is the top most important part to implement both machine learning or deep learning area. For our study, we used well-known Flickr8k (containing 8,000 images) dataset to train this proposed architecture. The official split, it sets 6,000 (75%) pictures for train, 1,000

(12.5%) pictures for validation, and 1,000 (12.5%) pictures for test (Chen et al., 2017). Each image is by five English sentences described (annotated). These sentences were translated into Amharic language using Google translator with the help of manual labor (a person who can write and speak Amharic fluently) to produce semantically correct translated Amharic descriptions. It took up to three months to complete the translation process. The sample translated Flickr8k dataset of the images with corresponding descriptions viewed in Appendix A.

Table 3.1 Show the translated Flickr8k dataset

Image	Text description
	– **የባህር ዳርቻው ላይ በርከት ያሉ ሰዎች ቁልቁል ተቀምጠዋል ።** *A couple of several people sitting on a ledge overlooking the beach .* – **የተወሰኑ ሰዎች በባህር ዳርቻው ላይ ግድግዳ ላይ ተቀምጠዋል ።** *A group of people sit on a wall at the beach .* – **በአሥራዎቹ ዕድሜ ውስጥ የሚገኙ ወጣቶች አንድ የባህር ዳርቻ አጠገብ ግድግዳ ላይ ተቀምጠዋል ።** *A group of teens sit on a wall by a beach .* – **በባህር ዳርቻው ያሉ ብዙ ሰዎች ።** *Crowd of people at the beach .* – **ብዙ ወጣቶች በተጨናነቀ የባህር ዳርቻ ግድግዳ ላይ ተቀምጠዋል።** *Several young people sitting on a rail above a crowded beach .*
	– **ጥቁር እና ነጭ ውሻ በነጭ አጥር በተከበበ ሳር የተሸፈነ የአትክልት ስፍራ ውስጥ እየሮጠ ነው ።።** *A black and white dog is running in a grassy garden surrounded by a white fence.* – **ጥቁርና ነጭ ውሻ በሣሩ ውስጥ እየሮጠ ነው ።።** *A black and white dog is running through the grass.* – **በሣሩ ውስጥ አንድ የቦስተን ቴሪየር እየሮጠ ።።** *A Boston terrier is running in the grass.* – **አንድ ቦስተን ቴሪየር በነጭ አጥር ፊት ለፊት ለምለም ሣር ላይ እየሮጠ ነው ።።** *A Boston Terrier is running on lush green grass in front of a white fence.* – **ውሻ በእንጨት አጥር አጠገብ ባለው አረንጓዴ ሣር ላይ ይሮጣል ።** *A dog runs on the green grass near a wooden fence.*

- **አንድ ወንድና ሕፃን በውሃ ላይ በቢጫ ጀልባ ውስጥ ናቸው ።**
  A man and a baby are in a yellow kayak on water .
- ***ሰማያዊ ጃኬት የለበሰ አንድ ወንድና ትንሽ ልጅ በቢጫ ታንኳ እየነዱ ናቸው ።***
  A man and a little boy in blue life jackets are rowing a yellow canoe .
- **ረጋ ባለ ውሃ ውስጥ አንድ ወንድ እና ልጅ በጀልባ ።።**
  A man and child kayak through gentle waters.
- **አንድ ወንድና ትንሽ ልጅ በቢጫ ጀልባ ተሳፈሩ ።።**
  A man and young boy ride in a yellow kayak .
- ***ሰው እና ልጅ በቢጫ ካያክ ጀልባ ተሳፈሩ ።***
  Man and child in yellow kayak .

There are two reasons why we chose the flickr8k dataset to train and test the proposed model. The primary cause is the availability and speed of our GPU. The dataset can train with a low-power GPU and is easy to analyse contrasted to other datasets (like Flicker30k and MSCOCO). Secondly, the reason is the description collected by a human from an AMT worker. These descriptions were gathered on AMT with precise instructions to verbalize the main action shown in the picture (Bernardi et al., 2017).

Table 3.2 Summary of the datasets used for this study

No.	***Dataset's name***	***No. of images***	***Training set***	***Testing set***	***No. of sentences***
1	Flickr8k (Amharic)	8,000	6,400	1,600	40,000

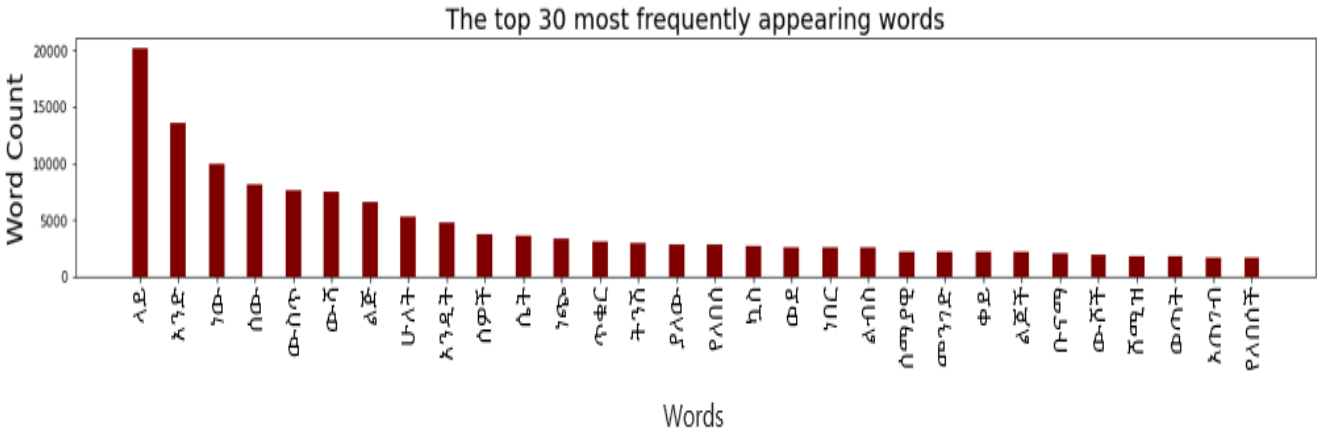


Figure 3.2 Visualization of the top 30 occurring words

3.3. Data Pre-processing Technique

Data pre-processing is a fundamental phase in ML and DL algorithms to prepare raw data into a suitable machine-reasonable format. The dataset in the Amharic image captioning process requires pre-processing techniques to make a suitable machine-reasonable format. Let's look at the image and text pre-processing strategies.

3.3.1. Text Pre-processing

The data in the text description dataset is a human-readable format. For the ML model to comprehend this data, it should convert it into digital format. For doing this, there are different pre-processing methods used to make it a more suitable format.

In the text description dataset, remove each unnecessary word or character before directly using it for the proposed model. To do this, we execute some essential data cleaning, such as lower-casing all the words to avoid duplication of words, removing special characters or non-alphanumeric characters (like '+' '%' '$' '#' '@', etc.), removing words that contain numbers (like 'Hello123', etc.). In addition, we have manually adjusted some of the inappropriate words and symbols that disturb the translated statement, see the

Table 3.3 In general, this data cleaning (filtering) performs on caption datasets about 40,000 (8000 * 5) sentences.

After the data cleaning performs, we distinguish the beginning and the ending of each caption sequence (like word tokens '< start >' and '< end >'). These indicate the start and stop prediction sequence for the model. We also apply basic tokenization and vectorization in this text description dataset. Tokenization splits the caption or sentences into smaller levels (units) such as word, character, or sentence levels. For this work, the captions were breakdown into word-level tokens. Vectorization is a method of transforming text data into numerical data.

Table 3.3 Types of dataset errors that are manually modified

Captions	**Correction captions**	**Remarks**
በ3ቱ	በሶስቱ	Words that contain a number
የApple or የ Motocross	የአፕል or የሞተር ክሮስ	Letters with other non-transliterated words

በ ፈገግታ or ቲ ሸርት	በፈገግታ or ቲሸርት	Single space with letters and words
ሁለት ሠራተኞች	ሁለት ሠራተኞች	Double spaces between words
ቀይ፣ or ነበር።	ቀይ or ነበር	Unnecessary symptoms

3.3.2. Image Processing

We have about 8,000 images through our image dataset. To train the proposed model, we need to reduce the size of an image in the dataset by the same size. For this reason, to reduce the model parameter numbers. In this case, the python script resizes all pictures of the dataset and the size of pixels is prepared by (299 width, 299 hight, 3 colour channels) to accept the proposed model. And other image processing techniques, like a grayscale histogram and data augmentation, are applied before the image datasets feed into the model.

3.4. Feature Extraction

On the other hand, image features extraction is used for the encoder. We employ a pre-train CNN model to obtain visual features from given images instead of a training model from scratch. Training the entire convolutional networks from scratch needs a large dataset and high computing power. So, we use the pre-train Inception-V3 CNN model for this research. For this reason, it has a lower parameter number as compared to other pre-train CNN models (VGG-16, ResNet-50). is a CNN that trains on millions of pictures in the ImageNet database (Seshadri et al., 2020). This pre-train model originally trained for classification tasks, and it can classify 1000 different classes of images. This thesis uses the Inception-V3 not to classify tasks is to get bottleneck features or extract the information of the image. This feature extraction is performed by remove the last SoftMax layer at the end of the Inception-V3 model and get 2048 features from each picture. And it accepts the image size of 299 * 299 * 3. The below table shows the components of the Inception-V3 model.

TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Conv	3×3/2	299×299×3
Conv	3×3/1	149×149×32
Conv padded	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Conv	3×3/1	73×73×64
Conv	3×3/2	71×71×80
Conv	3×3/1	35×35×192
3 × Inception	Module 1	35×35×288
5 × Inception	Module 2	17×17×768
2 × Inception	Module 3	8×8×1280
Pool	8 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
SoftMax	Classifier	1 × 1 × 1000

Figure 3.3 Inception-V3 model architecture

(Source)

3.5. Development Tools

This section describes the various development tools utilized to execute the proposed model from design to implementation.

3.5.1. Design Tools

Design tools are essential for analysing design concepts, presenting, and developing. Microsoft Visio is software that allows users to draw a variety of diagrams. We used Visio version 16 for this study.

3.5.2. Hardware tools

We use hardware tools to implement this study appropriately. Hardware tools needed for this study list in the table below.

Table 3.4 Implementation tools of hardware employed for this study

No.	*Name of Device*	*Used in the study*
1	Hard Disk	To store datasets.
2	GPU	The process of training large datasets on the CPU takes time, so we used GPU for this study to speed up the training and computation.
3	RAM	It is used for GPU to accelerate the train operation.

3.5.3. Software Development Framework Tools

For executing the study work, distinct application packages and programming language uses. These are mentioned below with their description.

***TensorFlow*:** it is a mathematical computation library for training and building ML and DL algorithms models with the easy use of high-level APIs. It also can be run on desktop computers or cloud platform.

Keras : is simply an interface that allows easy access and customizes the ML frameworks such as TensorFlow, CNTK, and Theano. These frameworks are called backends. It is more user-friendly and easier to build a model. It also can be run on CPU and GPU.

Anaconda : is a free and open-source distribution of R and Python programming languages that focuses on providing everything for data science and ML applications with more than one hundred fifty packages included with Python's core languages.

Python : is a widespread general-purpose programming language that can utilize for different applications. For this study, we used the latest version of Python (Python 3.7.).

Mendeley Desktop : is a free web and desktop version used to reference managing applications that create and manage a research library. We used the Mendeley Desktop version 1.18.9 for this study.

Microsoft office packages : It uses to write the study document in a proper format.

3.6. Baseline Works

This proposed model compares various image caption models which use the same encoder-decoder approach based on different benchmark datasets. Both CNN-Bi-GRU (Faruk et al., 2020) and Bag-LSTM (Cao et al., 2019) take as the baseline for the experiment.

CNN-Bi-GRU is an encoder-decoder based which generates captions in the Bangladeshi language. In this model, pre-train Inception-V3 is operated as an image encoder and Bi-GRU as language decoder to generate captions. Using this model Bi-GRU decoder, they got better results with the newly introduced dataset.

Bag-LSTM This model consists of three components. The first Pre-train Inception-V3 CNN to extract image feature. The second semantic attention model uses to produce picture features associated with visual data. The last and third are Bi-gLSTM, which process both forward and backward contexts. Bag-LSTM performed better by different evaluations using different benchmark datasets.

These models choose to demonstrate the viability of the proposed model. Both are different model approaches, and CNN-Bi-GRU uses the BNATURE dataset (similar to the Flickr8k dataset) as a benchmark. Bag-LSTM used the Flickr8k dataset. The objective of these selected models uses to demonstrate how the proposed model enhanced image caption using those datasets in different language structures.

3.7. Evaluation Method

In this work, we employ the two most typical evaluation metrics used for measures in MT or image captions to evaluate the performance of our model. Those are BLEU and Human evaluations metrics. Let's see them in detail:

3.7.1. BLEU Metric

It is a precision-based metric which utilized to estimate the quality of machine-generated text. BLEU matric is scope lies from 0 to 100, denoting perfect mismatch and perfect match, respectively, and again has the benefit of comparing n-grams of the predicted outcome with the n-grams of the actual data (Phukan et al., 2020). If the BLEU score is toward 100, it implies that the model performed better. The computation is as follows: first, the B_P factor calculated by this equation Eq.3.1 is to handle short sentences. Finally, the final BLEU score can be calculated by this equation Eq.3.2.

$$\mathrm{B}_P = \begin{cases} 1, \mathrm{c} > r \\ e^{\left(1-\frac{r}{\mathrm{c}}\right)}, \quad \mathrm{c} \leq \mathrm{r} \end{cases} \tag{3.1}$$

Where r is the length of the reference corpus, and candidate translation length is given by c (Wo et al., 2016)

$$\mathrm{BLEU} = \mathrm{B}_P . exp\left(\sum_{\mathrm{n}=0}^{\mathrm{N}} w_{n \log P_n}\right) \tag{3.2}$$

The equations Eq.3.1 and Eq.3.2 (Wo et al., 2016).

where w_n are positive weights summing to one, and the n-gram precision p_n is calculated using n-grams with a maximum length of N (Wo et al., 2016). Also, we employ a greedy search method to calculate these n-gram BLEU scores. The greedy search for a word with the highest probability selects the flowing word in the sequence (Humaira et al., 2021).

Hence, these evaluations indicate that the highest BLEU score was obtained, which means the generated captions are nearer to human judgment.

3.7.2. Human Evaluation Study

In this section, the Twenty-four volunteer who can write and speak Amharic fluently asks to evaluate whether the resulting image description accurately describes the content of the given image. These volunteers make a point: first, look at the image shown to them and then evaluate the picture description. Overall, this question has two components. Firstly, the sentence describes the image content correctly or is related to the provided picture. Second, the generated texts are grammatically accurate or are fluent. Considering the components of these evaluations, the volunteers fill out a form provided. And finally, the average score calculates.

CHAPTER FOUR

4. PROPOSED HYBRIDIZED ATTENTION-BASED CAPTIONS IN AMHARIC LANGUAGE

4.1. Chapter Overview

This chapter describes the proposed solution to the problem of the caption. The fundamental solution of the proposed model is improving the narrow correlation gap between visual and textual feature representation to generate semantically correct captions. This section again explains the all-inclusive proposed key of architecture with necessary diagrammatic representation, mathematical expressions, and algorithms for a well-defined address to the problem.

4.2. Proposed Model Architecture

Attention-based Bi-GRU encoder-decoder framework is the proposed model for effectively generating image caption in the Amharic language by reducing the gap between image-text features representation. The proposed model also introduced Bi-GRU with AM (additive attention) language model decoder by extending (Faruk et al., 2020) work. And it is used to generate realistic captions by considering both visual and textual relevant features.

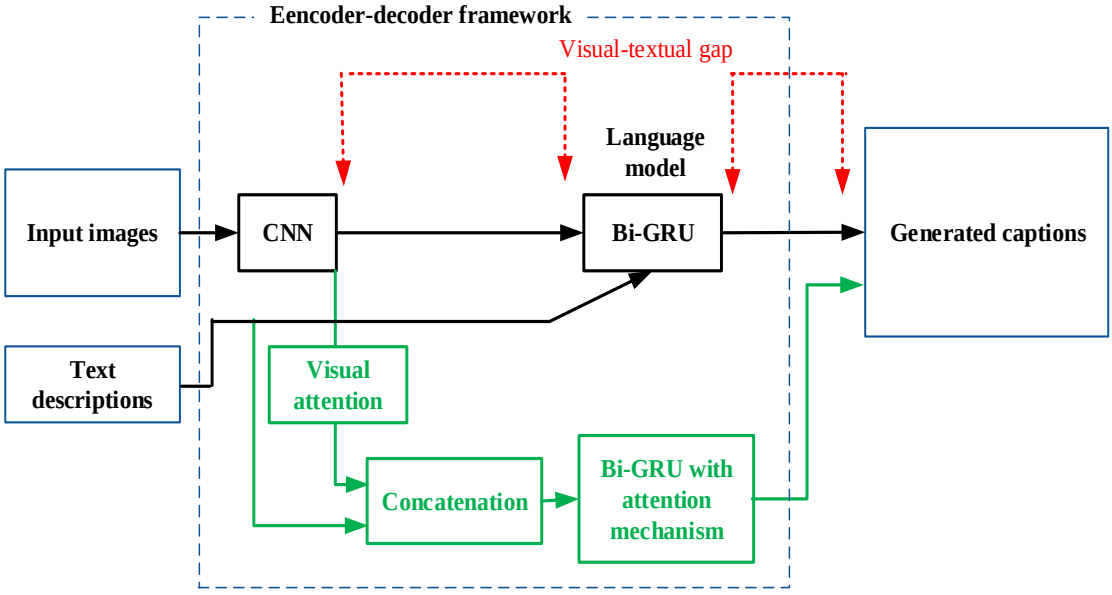


Figure 4.1 Show the gap between the baseline and the proposed solution

The above figure shows the bold black line, bold green line, and bold red dots representing the pipeline of the baseline model, proposed model, and limitation or gap between the two models. This baseline encoder-decoder model has two separate gaps addressed by this proposed model. The first is the gap between the encoder-decoder framework. The encoder feature extraction goes directly to the Bi-GRU, preventing the decoder from capturing only the essential image information. Secondly, baseline model is limited to selecting the required visual-textual related features during the captions generation. This proposed model addresses this limitation by adding an additive mechanism to the language decoder framework during image captions generate that the decoder focus on the important visual-textual related feature information. Let's look at the detailed proposed model architecture diagram.

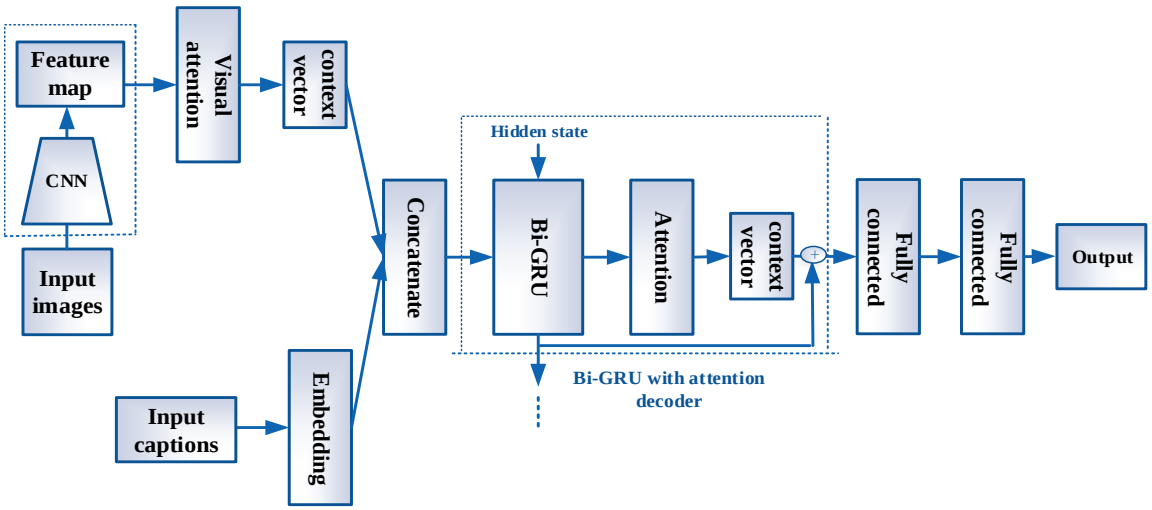


Figure 4.2 Proposed model architecture

4.3. Word Embedding (Feature Representation)

In NLP, word embedding uses for representing the words or the whole document vocabulary in numerical form. Before implementing word embedding, we entirely cleaned some noise-containing elements from the description sentences. At the end of the clearing process, each sentence transforms into a word-level piece. This process is called tokenization, and each token is a vocabulary with an assigned unique integer value or each token map to some number. The last pre-process is a padding that uses the input sentence to make a similar shape. Once the pre-processing is complete, the input sentence moves to the next word embedding layer.

In the input sentence consisting of T words $S = \{x_1, x_2, \dots x_T\}$, every word in x_i is converted into a feature vector e_i inspired by matrix-vector product (Zhou et al., 2016).

$$e_i = W^{wrd} v^i \qquad 4.1$$

The matrix W^{wrd} is a parameter to be learned, and it is an element of $\mathbb{R}^{d^w}|V|$, where V is a vocabulary size, and d^w is word embedding of size (hyperparameter), v^i is the size of a vector $|V|$. Finally, the given sentence $S = \{x_1, x_2, \dots x_T\}$ convert into real feature vector $Emb_s = \{e_1, e_2, \dots e_T\}$.

4.4. CNN Image Encoder

The encoder is responsible for integrating image features into a language decoder, and these features are extracts using pre-train CNN. For the image encoder model, employ the Inception-V3 pre-train model to take out the local image features. By removing the last two layers, we obtain a global average pooling layer of 8 * 8 * 2048. Inception-V3 global average pooling layer 2048 feature vector takes an 8 * 8 feature map and by using a flatten layer to convert 8 * 8 metrics into a 64 * 2048 vector of features. The final extractor outcome produces an L feature vector, each of which is a D dimensional representation correlated with a portion of the picture.

$$A = [a_1, a_2, \dots a_L], \qquad a_i \in \mathbb{R}^{D\,(2048)}$$

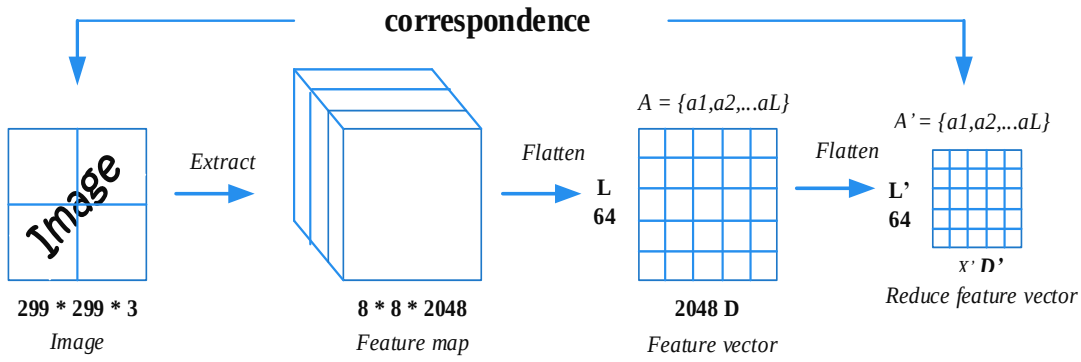


Figure 4.3 Encoder feature extraction

Represent the CNN visual feature A at each of the L grid locations. Finally, the extracted feature vector output passes into a dense layer with a ReLU activation function. This layer has an dimension unit dim (hyper-parameter) that uses to reduce the input dimension the same as the dimension of word embedding Emb_s. The new reduces feature vector, which

outcome of the CNN image encoder utilizes for the AM to decide which section of the picture is better relevant in producing the following word in captions generation.

4.5. Visual Attention Mechanism

This visual attention permits the image caption to concentrate only on certain essential parts of the picture rather than the overall image content. We follow (Xu et al., 2015) to implement the visual attention function. First, calculating attention score at each time t and location i in the image based on the hidden state h_{t-1} and the output of encoder (set of a_i visual feature) by passing it via a non-linear activation function (tanh). Then apply the SoftMax activation function to produce the attention distribution over the picture parts. For each part i, the AM generates a positive weight α_i, which can interpret as an essential to provide to the place i in combining the feature vector a_i.

$$\alpha_{ti} = \text{SoftMax}\left({W_a}^T tanh(W_a a_i + W_h h_{t-1})\right) \quad 4.2$$

Where ${W_a}^T \in \mathbb{R}^L$, W_a, W_h denotes the size of each vector, and those are learnable parameters. $\alpha \in \mathbb{R}^L$ is attention weight (attention distribution) over features in A.

Finally, we can compute the context vector C_t based on the attention distribution. It is necessary to combine visual representation specific to each word at a given time step t.

$$C_t = \sum_{i=0}^{L} \alpha_{ti} a_i \quad 4.3$$

After visual attention, we combine the context vector output and the text feature representation output. It reduces the contact gap between the two. The result of a concatenation Con_x shows in the following equation.

$$Con_x = [Emb_s \oplus C_t] \quad 4.4$$

Where, $\oplus$ represents the element-wise sum to integrate the context vector and word embedding output.

4.6. Bi-GRU with AM Language Decoder

The language decoder aims to generate captions for the given pictures, the descriptive sentence that correctly describes what see in the picture. The initial step of this decoder accepts the outcome of concatenating layer as input and passes it to the Bi-GRU network layer. The Bi-GRU network computed the backward and forward hidden sequence represented as $\vec{h}$ and $\overleftarrow{h}$. To demonstrate the computation, we take the forward part as a sample. Bidirectional GRU networks operate as follows:

$$z_t = \sigma(W_z.[h_{t-1}, Con_{xt}] + b_z) \quad 4.5$$

$$r_r = \sigma(W_r.[h_{t-1}, Con_{xt}] + b_r) \quad 4.6$$

$$\tilde{h} = \sigma(W.[r_t * h_{t-1}, Con_{xt}] + b) \quad 4.7$$

$$h_t = (1 - z_t) * h_{t-1} + z_t \tilde{h}_t \quad 4.8$$

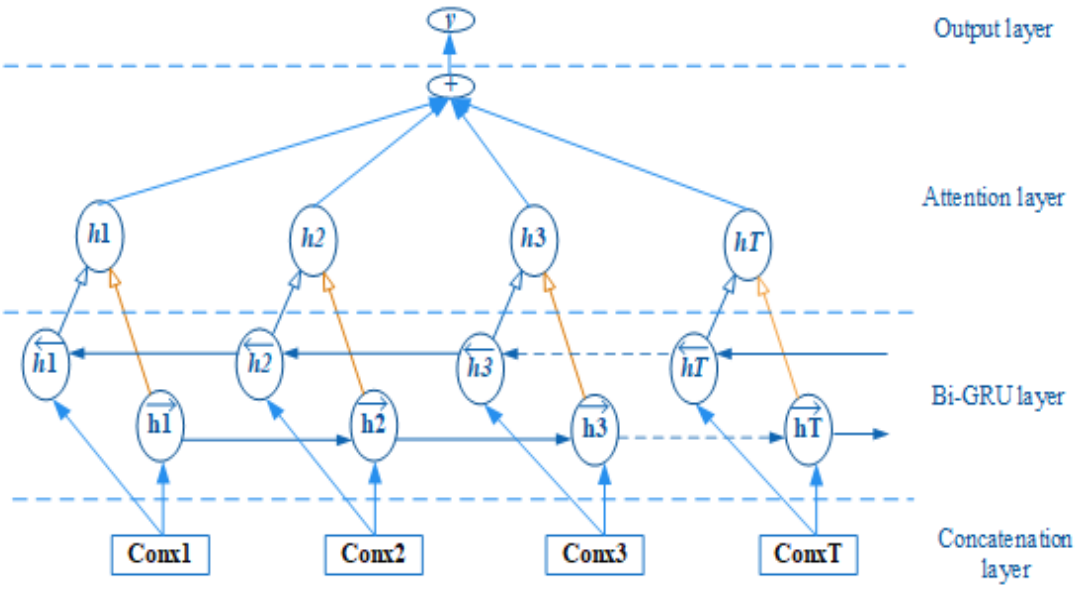


Figure 4.4 Bidirectional GRU with AM

Where, z_t and zr_t indicate the update and reset get output at current timestep t, respectively, h_{t-1}, $\tilde{h}$, and h_t are reparented information from the previous unit, current memory content, and final memory at the current unit. Con_{xt} denotes the concatenation of visual attention context and text feature at timestep t. σ is the ReLU activation function. W, W_z, and W_r are

represented at the current unit, weigh matrix at update get and reset gate. b, b_z, and b_r are bise terms corresponding with gates.

In this paper, we follow the Bi-GRU or Bi-LSTM model that can explore information both from the past and future. The model consists of right and left layers ordering context, which pass separately. The following equation Eq.4.9 demonstrates the output of the j^{th} word: where, $\oplus$ denoted concatenate (element wise sum to merge the forward $\overrightarrow{h_\iota}$ and backward $\overleftarrow{h_\iota}$ pass result).

$$h_j = \left[\overrightarrow{h_\iota} \oplus \overleftarrow{h_\iota}\right] \tag{4.9}$$

In this manner, the hidden state annotation consists of T_x words $h_j = \{h_1, h_2, \ldots h_{T_x}\}$. It contains the summary of information from the two forward and backward directions.

4.6.1. Addictive Attention

In the Bi-GRU with attention decoder, we use addictive attention which is the same as visual attention called Bahdanau-style attention. This mechanism the first introduced in this MT work. Now a day, its variants are widely used in various application domains. In the proposed attention-based language decoder, addictive attention calculates using two steps. First, an operation yields a matching score (alignment scores) e_{tj} between the current hidden h_t state (information from the current time step) and previous hidden state h_j (information from previous time step) by using additive projection (Yin et al., 2018). Unlike other methods (Yucong et al. 2021), attention operates over the Bi-GRU unit in both directions hidden state with output of Bi-GRU rather than encoder features output.

$$e_{tj} = V_a{}^T tanh\left(W_a . h_t + U_a . h_j\right) \tag{4.10}$$

Where, V_a, W_a, and $U_a \in \mathbb{R}^d$ learned attention parameter, d is the dimensionality of the hidden state. Secondly, calculate the attention context vector (final score) C_j for hidden state h_j based on the output of alignment scores. To calculate the final score:

$$C_j = \sum_{j=1}^{T_x} e_{tj} h_j \tag{4.11}$$

Finally, the addictive attention multidimensional output is converted to 1-D dimensional using flatten layer. This layer performs as a transition for the final two dense layer output. The first dense layer uses the length parameter, and the second one or final output dense layer uses a probability of vocabulary or size of lexicon to produce output descriptions.

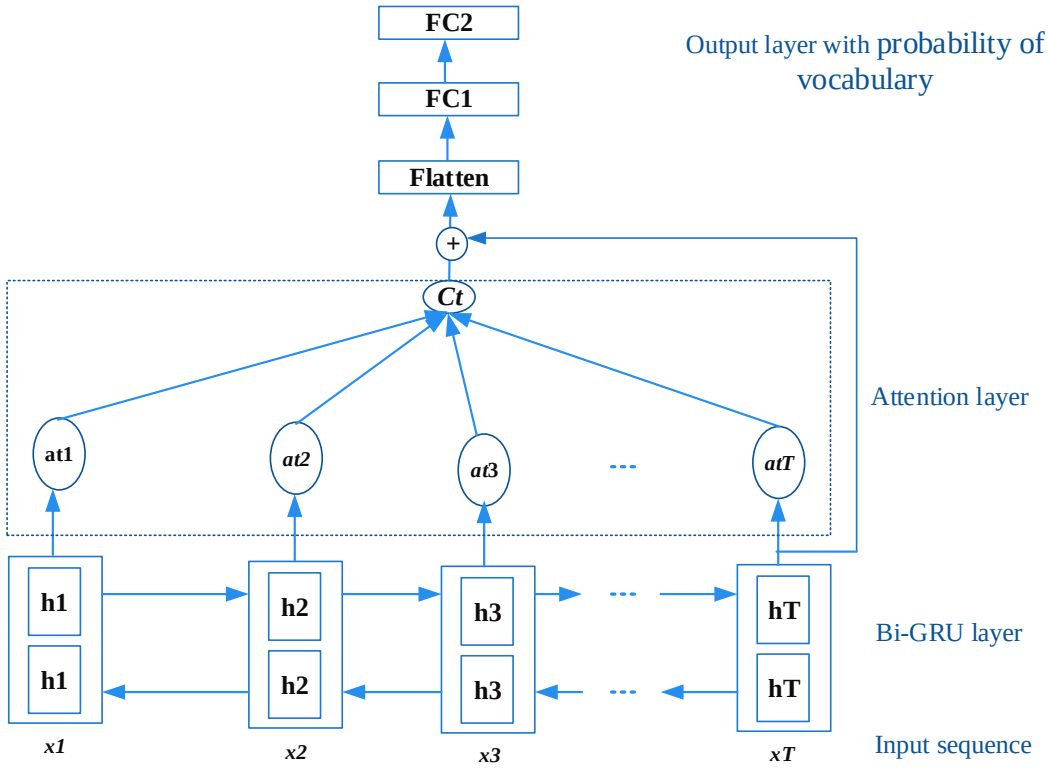


Figure 4.5 Bi-GRU with AM language decoder model

The decoder model captions generate using probability distribution $p(p_i \mid t_i, I)$ for the image. Where $p_i \in \mathbb{R}^{L\times D}$ is the caption generated by the model, $t_i \in \mathbb{R}^{L\times D}$ is the L length actual captions embedding matrix. I denoted by image feature. We use the supervised learning method for the model learning approach inspired by (Vaswani et al., 2017). Given the training set (target ground-truth sequence), the model would be trained by minimizing the SCE as the loss function, which is the same as CE for the multi-classification model. But the only difference between the two is how truth labels (ground-truth sequences) are defined. In SCE, truth labels are integer encoded rather than a one-hot vector. The SoftMax activation function applies to the probabilities before the SCE Loss computation. The objective is to make the probability distribution model output as close as possible to the desired result (integer encoded ground-truth output value). Finally, through probabilistic SoftMax operation outputs the following word in the sequence. The SoftMax greedily

chooses the word with the highest chance. The greedy search is employed to produce the image captions and compute evaluation scores.

4.7. Training Pseudocode

In this section we have shown the overall algorithms for generating image caption for Amharic language using hybridized attention-based deep neural networks. The section proposed model is highlighted.

Algorithm 1: Hybridized attention-based encoder-decoder framework.

Input: I a collection of pictures and C relating collection of captions during the training process.

Output: caption C' for test pictures I' during generate captions for the test pictures.

1. *Each of the captions c_i in I image does the text pre-processing.*
2. *And choose 5,000 repetitive words from C.*
3. *The most frequently selected words are sent to the tokenization T and generate a T_i set of tokens, where $i = 1,2, \dots n$.*
4. *Each token T_i pass to the word embedding Emb_s.*
5. *The word embedding Emb_s coverts token T_i into numerical vector indices where $V_i = 1,2, \dots n$.*
6. *Extract features f_i where $i = 1,2, \dots n$ from image I_i utilizing Inception-V3 network.*
7. *The extract image features f_i pass via the encoder E.*
8. *Combined the encoder output features e_i and hidden state h_i through the single layer NN and the result scores* S.
9. *The SoftMax function is applied on scores* S *to yield the attention weights A_{tt}.*
10. *Give attention weights to the different pixels in the picture called context vector $Conv$.*
11. *Combine context vector $Conv$ and word embedding vector Emb_s through the concatenation con_x.*
12. ***Learn the weighs of Bi-GRU network by feeding the concatenate vector con_x.***

13. ***Merge the output of forward h_f and backward hidden state h_b via the concatenation con_{x_1}***
14. ***Apply AM A_{tt_2} on final merge output vector con_{x_1} and $Bi-GRU$* output result.**
15. ***Combine the attention output A_{tt_2} with the output of Bi-GRU layer to generate final probability model output O.***

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

This section discussed the proposed model architecture implementation. We also discussed the configuration of the working environment and hybridized attention-based deep neural networks implantation.

5.2. Working Environment

We describe the tools we have utilized to develop and implement the proposed model. Laptop: A laptop is used to do model design work.

Hardware and Software configuration

- Processor : Intel® Core™i5
- RAM and hard disk: 8GB and 1TB
- Graphics : Intel ® Graphics 3000
- System type : x64-based processor
- Operating system : Windows 10
- Programming language : python, kears and TensorFlow

Desktop and Google Collaboratory are used for developing or training a proposed model.

Hardware and software configuration

- Operating system : Windows 10
- Processor : Intel® Core™ i5
- Graphics : Intel ® HD Graphics 4600
- GPU : GeForce RTX 2070 Super 6 GB RAM
- RAM and hard disk : 8GB and 500GB
- System Type : x64-based processor
- Programming language : python, Keras and TensorFlow

Google Collaboratory (colab) is a free Jupyter notebook environment for a maximum of 12 hours from the Google cloud server. The hardware specification of the colab:

- GPU : Nvidia k80 /T4
- RAM : 12GB

5.3. Environment setup

Application software

Anaconda : is a free and open-source distribution of R and Python programming languages that focuses on providing everything for data science and ML applications with more than one hundred fifty packages included with Python's core languages. The proposed model anaconda application version 2.1. 1 with 64-bit support utilized.

Jupyter Notebook : is a web-based application that contributes a front-end to many different languages and interactive shells such as IPython. The Jupyter notebook with a 6.4.5 version uses to execute the proposed model.

Colab : is a cloud service based on Jupiter notebook, which can improve the computing period by operating the GPU and RAM. We can train our model for 12 hours using colab GPUs.

5.4. Dataset Processing

In this area, we use the Flick8k dataset translated from English to Amharic to train and test our models. Again, we used different language structures of the dataset, like the Flick8k English version dataset and the BNATURE (Bangladesh language) dataset. The next figure shows the code snip of importing the dataset and reading the images into a separate variable.

```
INPUT_PATH = "../content/drive/MyDrive/demo/"
IMAGE_PATH = INPUT_PATH+'Images/'
CAPTIONS_FILE = INPUT_PATH+'amharic_captions.txt'
```

Snippet code 5.1 Snipes code loading the dataset

5.5. Captions Pre-processing and Tokenization

In this phase, the captions or text data pre-processes to clean unwanted characters or non-alphanumeric characters. Also, we set the top word count equal to 5000 and "<unk>" for handling out-of-vocabulary. At the end of the cleaning process, each sentence converts into small chanks (tokens) words, and by using text_to_sequence, we convert each token of the sentence corpus into a sequence of integers.

```
Word_top_cnt = 5000
S_chars = '!"#$%&()*+.,-/:;=?@[\]^_`{|}~ '
W_toks = text.Tokenizer ( num_words = Word_top_cnt,
                                                    oov_token = "<unk>",
                                                    filters = S_chars)
W_toks. fit_on_texts(annotations)
seqs_train = W_toks.texts_to_sequences(annotations)
```

Snippet code 5.2 Snipped code of caption pre-processes

5.5.1. Image Processing

In image processing, we first read the image data from non-volatile storage. Then covert the compressed format (JPEG) to a 3D uint8 tensor (3D matrix) by using the tf.io.decoder_jpg() function available in the TensorFlow library and resize images into the shape (299, 299) as required by the network model. Finally, by using inception_v3.preprocess_input() function prepares the images for the Inception-V3 CNN model. This function allows normalized (scale) input pixels between -1 and 1. Load the pre-trained Inception-V3, and the pre-processing of the images snipped code provided to the proposed model is as follows:

```
I_model_V3 = InceptionV3(include top = False, weights = 'imagenet')
I_input = I_model_V3.input
H_hidden_lar = I_model_V3.layers [-1] .output
I_features_ext = tf.keras.Model (I_input, H_hidden_lays)
```

Snippet code 5.3 Loading pre-train Inception-V3 model

We extract the image features by removing the last layer (`I_model_V3.layers[-1].output`) of the Inception-V3 model.

```
def I_prep(i_path, shape = (299, 299)):
    im_ = tf.io.read_file(i_path)
    im_ = tf.image.decode_jpeg(im_,channels = 3)
    im_ = tf.image.resize(im_,shape)
    im_ = inception_v3.preprocess_input(im_)
    return im_, i_path
```

Snippet code 5.4 Define image processing

5.6. Train and Test Data

To train and test the proposed model, we use an 80:20 split ratio for the training and testing sets of subgroups. The following figure shows a code sniping for creating the train-test split based on an 80:20 ratio and a random state equal to 42.

```
image_train, image_test, caption_train, caption_test = train_test_split (
    im_all_vector, cap_vector, test_size = 0.2, random_state = 42)
```

Snippet code 5.5 Train and test dataset split

After the dataset splits by a given ratio, next, merge both images and captions to create a train and test dataset using tf.data.Dataset.from_tensor_slices(). The snippet code is below batch size = 32 and buffer size = 1000.

```
def generate_dataset(images_data, captions_data,
                     S_BATCH_ = 32, S_BUFFER_ = 1000):
    data_sets = tf.data.Dataset.from_tensor_slices((
        data_images,  data_caption))
    data_sets = data_sets.shuffle(S_BUFFER_)

    data_sets = data_sets.map(lambda item1, item2: tf.numpy_function(
          function_map,[item1, item2], [tf.float32,  tf.int32]),
          num_parallel_calls = tf.data.experimental.AUTOTUNE).
          batch(S_BUFFER_)
    data_sets = data_sets.prefetch(buffer_size = tf.data.
                              experimental.AUTOTUNE)
    return data_sets

train_data_sets = generate_dataset(image_train,caption_train)
test_data_sets = generate_dataset(image_test,caption_test)
```

Snippet code 5.6 Creating dataset

The shape of each image and caption in the created dataset should be (32, 64, 2048) and (32, 33).

5.7. Model Building

5.7.1. CNN Image Encoder

We utilize the Keras model subclassing API to construct the Encoder-decoder framework for the proposed model. Model subclassing API tends to be more flexible and fully customizable than functional or sequential API.

CNN image encoder is a single fully connected layer followed by a ReLU activation function. The extracted image features by Inception-V3 are input to the fully connected layer to get the output vector. This output vector is the last hidden state of the CNN encoder that connects to the decoder model. The code snip of the CNN encoder shows below Snippet code 5.7. Embedding dimension = 256, and dropout = 0.2.

```
class I_Encoder( tf.keras.Model):
    def __init__(self, dim_embedding_):
        super(I_Encoder, self).__init__()
        self.ds = tf.keras.layers.Dense(dim_embedding_)
        self.drop_out = tf.keras.layers.Dropout(0.2)

    def call(self,  x):
        x = self.drop_out(x)
        x = self.ds(x)
        x = tf.nn.relu(x)
        return x
```

Snippet code 5.7 Define the CNN encoder

5.7.2. Bi-GRU with AM Decoder

In the decoder, we first calculated the visual AM. Visual attention application is based on encoder output (feature) the shape (32, 64, 256) and expands hidden state shape (32, 1, 256). The final AM result is to yield a context vector shape (64, 256), which assign weights to the different pixel in the image. The next step is to merge the context vector and word embedding layer. Once combined, the *enc_emb* shape (64, 1, 512) and passed to the Bi-GRU layer.

```
# hidden shape
sh_hidden_ta = tf.expand_dims(hidden, 1)
scorel = self.Vattn(tf.nn.tanh( self.Uattn( features) +
                                self.Wattn(sh_hidden_ta)))
attWs = tf.nn.softmax( scorel, axis = 1)
context_vx = attW * features
context_vx = tf.reduce_sum(context_vx, axis = 1)
dec_emb = self.embedding( x)
enc_emb = tf.concat([tf.expand_dims (context_vx, 1),
                     dec_emb], axis=-1)
```

Snippet code 5.8 Implementation of visual AM.

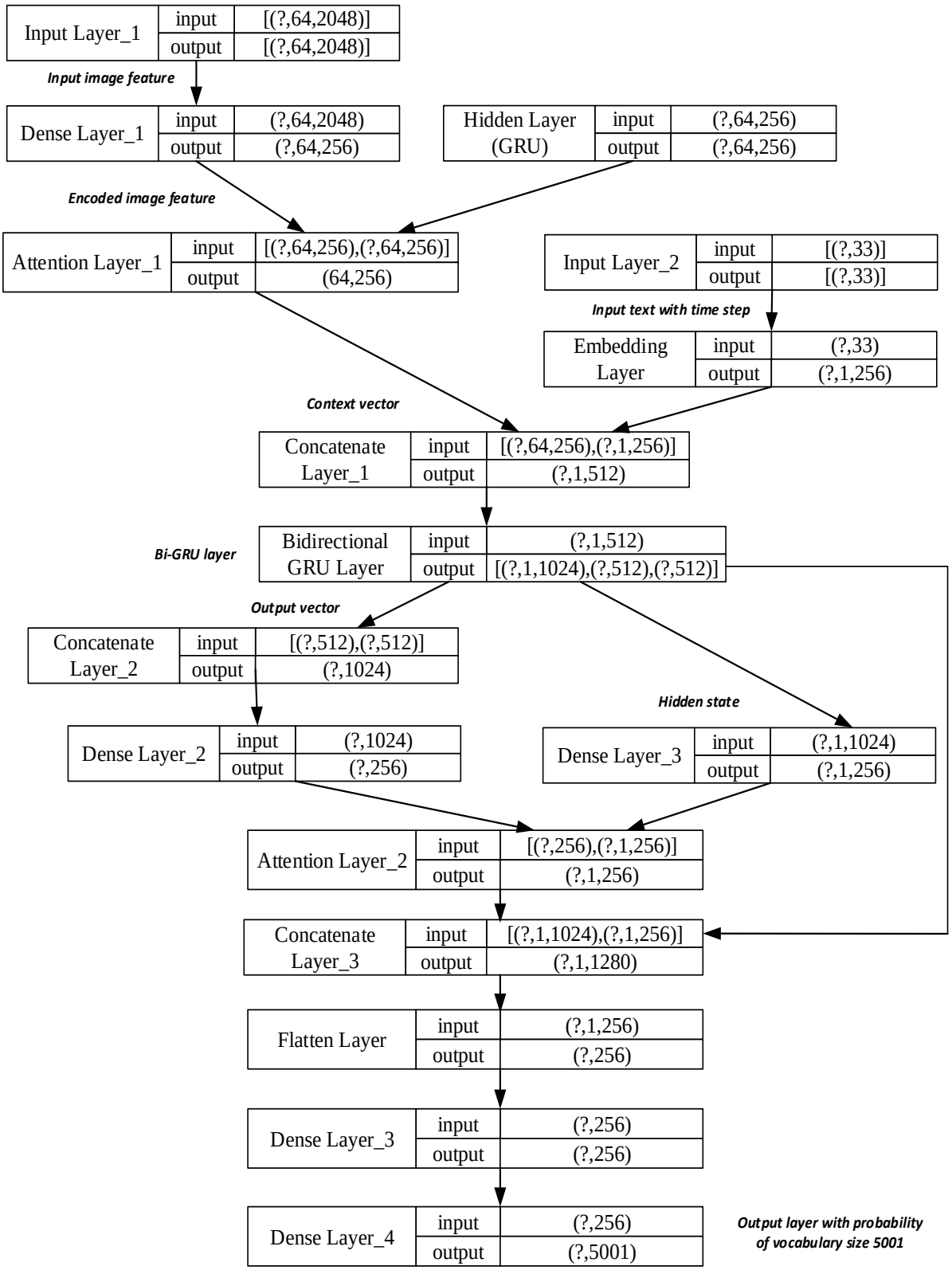
Bi-GRU networks are a sort of RNNs capable of learning sequence data in both directions (The written information is well explained in Chapter 2 and 4). The Bi-GRU forward and backward layers are merged and sent to the AM with the final Bi-GRU output. Finally, the AM result is again fused to the Bi-GRU output and sent to the final dense layers. The overall implementation of the decoder code snippet shows in detail in Appendix C. Now, let's look at the Bi-GRU and AM code snippet separately below.

```
# Bidirectional GRU
self.bigru = tf.keras.layers.Bidirectional(
     tf.keras.layers.GRU
    (self._units, return_sequences = True,
     return_state = True,
     dropout = 0.2, recurrent_dropout = 0.2,
     recurrent_initializer = tf.keras.initializers.
     glorot_uniform (seed = 26),
     kernel_regularizer = tf.keras.regularizers.l2 (0.001),
     recurrent_regularizer = tf.keras.regularizers.l2 (0.1)))
```

Snippet code 5.9 Define Bi-GRU network layer

```
dec_emb = self.embedding(x)
# Combine Word embedding with a context vector
enc_emb = tf.concat([tf.expand_dims(context_vx, 1),
                     dec_emb], axis = -1)
(bigru_outputs, forward_h, backward_h) = self.bigru(enc_emb)
# Merged forward with a backward hidden layer
state_h = self.concat([forward_h, backward_h])
bigru_outputs = self.w_1(bigru_outputs)
state_h = self.w_1(state_h)
# AdditiveAttention
attn_out = self.attention([bigru_outputs,state_h])
# Combine Bi-GRU with attention
decoder_concat_output = self.concat([bigru_outputs,attn_out])
output = self.flatten(decoder_concat_output)
```

Snippet code 5.10 Implementation of Bi-GRU with an AM



Snippet code 5.11 Visualization graph of proposed model

Where, ? denote the batch size.

CHAPTER SIX

6. RESULTS AND DISCUSSIONS

6.1. Overview

In the chapter, we discuss the result and discussion proposed model. We also discuss experimental findings in this study. This thesis generally trie applies the AM in two separate CNN encoders and a Bi-GRU decoder to generate a more relevant image caption in the Amharic language. Also, we demonstrate the results of the two experiment steps we have taken to achieve this. And finally, we show the result by comparing the proposed model with the baseline model we chose to compare.

6.2. Hybridized Attention-based Model

This hybridized attention-based image captioning model requires producing appropriate statements for the given picture. For this study, we conducted two separate experiments. The first experiment performs on a CNN-Bi-GRU encoder-decoder model. The second one we proposed was a hybridized attention-based CNN-Bi-GRU model. This model differs from the first experiment in that it generates an image caption by extending the AM on both the encoder and the decoder. These experiments perform using the new Flick8k dataset translated into the Amharic language, and it contains 8,000 images with five corresponding descriptions. Again, these experiments evaluate by the BLEU scores with the same training epoch and model hyperparameters setting.

6.2.1. EXPER 1: CNN Encoder and Bi-GRU Decoder Model (Basic Model)

This experiment performs outside of the AM, and it uses to compare how much this mechanism affects the model performance. In the basic CNN-Bi-GRU encoder-decoder model experiment: the first step is to extract image features using the pre-train Inception-V3 model. In the next step, these features pass into the CNN encoder. Second, the CNN encoder output and word embedding layer are combined and sent to the Bi-GRU layer. The final model output probability is based on this Bi-GRU. The hyperparameter set we use for the EXPER is listed in the Table 6.1 below:

Table 6.1 hyperparameters setup

Hyperparameter	**Values**
Batch size	32
Embedding dimension	256
Dense layer	256
Hidden layer (unit)	512
Dropout for encoder and decoder	0.2 (for dense layer) and 0.2 (for GRU cell), 0.3 (for dense layer)
Recurrent dropout	0.2
Recurrent activation	ReLU
Kernel regularize and Recurrent regularize	L2 (0.001) and L2 (0.1)
Batch normalization	-
Optimizer	Adam
Epoch	35

By using these hyperparameters, train the model and evaluates using the Flickr8k dataset in 1,000 test image predictions. And, also we apply the 1G-BLEU, 2G-BLEU, 3G-BLEU, and 4G-BLEU scores for our evaluation result based on the greedy search method. We present the results in the Table 6.2 below:

Table 6.2 Basic model BLEU score translated Amharic Flickr8k dataset.

Dataset	Model	1G-BLEU	2G-BLEU	3G-BLEU	4G-BLEU
Translated Flickr8k	CNN-Bi-GRU	46.4	38.2	33.0	28.5

We have shown the loss and the accuracy plot in the diagram below.

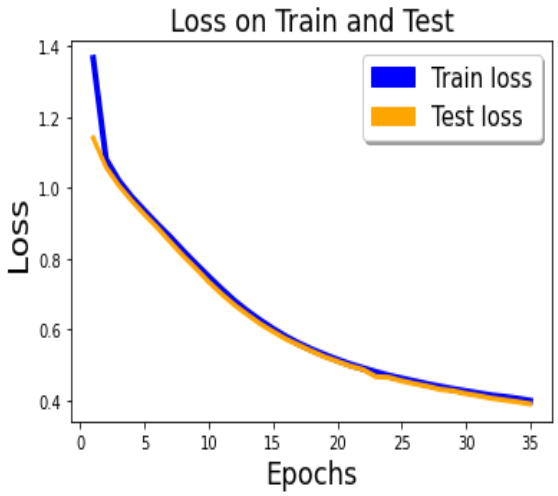


Figure 6.1 Model performance on train and test loss Vs epoch

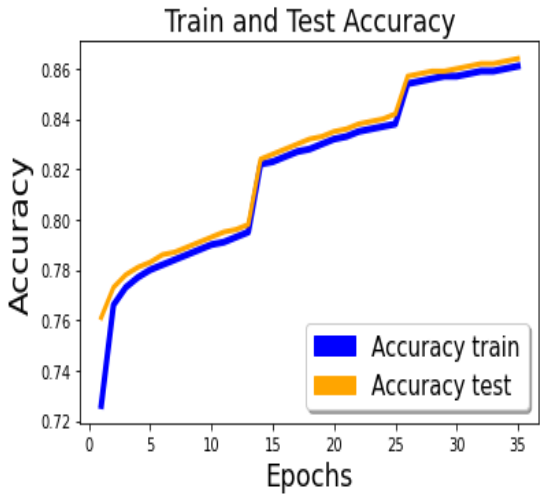


Figure 6.2 Model performance on train and test accuracy Vs epoch

The diagram above shows that the CNN encoder and Bi-GRU decoder model trains up to 35 epochs, the loss of the train has decreased from 1.365 to 0.398, and test loss lowered from 1.140 to 0.390. Also, the training accuracy has improved from 0.726 to 0.861 and the testing accuracy enhanced from 0.765 to 0.864. To raise the performance of the CNN bidirectional GRU model is the aim. In the following experiment sections, the AM added is present.

6.2.2. EXPER 2: Hybridized Attention-based CNN-Bi-GRU Model

In this experiment, the AM performs both a CNN encoder and a Bi-GRU decoder. The first AM (visual attention) locates between the CNN and the Bi-GRU, which contributes to the Bi-GRU by focusing only on the feature of the images and selecting the relevant image features. Extend the AM on the existing Bi-GRU decoder minimizes the vision and textual gap while generating the nearest to actual descriptions. And, Combining the AM result with the Bi-GRU output to produce a better and more reliable image caption. This experiment is similar to the first test with epochs and hyperparameters. From the outcomes indicated in Table 6.3, the performance of this attention-based hybridized model show higher BLEU scores compared to the previous experiment result, thus having an overall better performance.

Table 6.3 Hybridized model BLEU score translated Amharic Flickr8k dataset.

Dataset	Model	1G-BLEU	2G-BLEU	3G-BLEU	4G-BLEU
Translated Flickr8k	Hybridized model	61.6	50.1	43.7	38.8

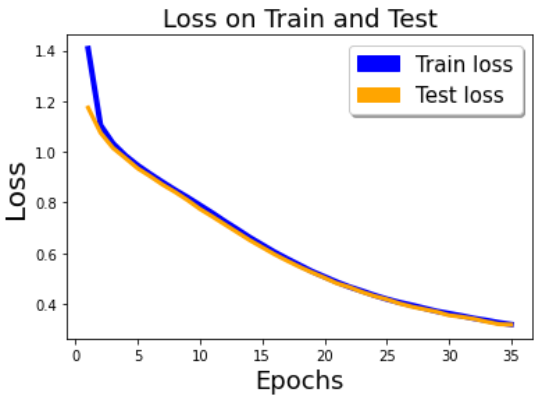


Figure 6.3 Performance of the model on the train and test loss against epoch

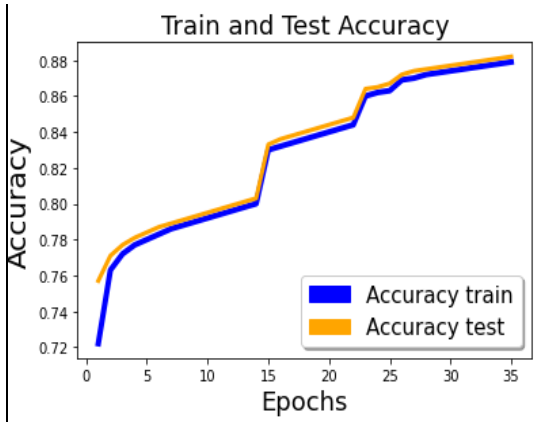


Figure 6.4 Performance of the model on the train and test accuracy against epoch

As the diagram above shows, we have plotted the training loss versus test loss and train accuracy versus test accuracy separately. As the outcome of experiment 2 shows, the accuracy got 0.87 for the train and 0.88 for the test set. And we observed approximately 0.319 loss in the training set, and the test loss is 0.318.

6.3. Experiment Result

In this thesis, two experiments conduct. The first one CNN-Bi-GRU uses as a basic model. This basic model comes from the baseline model that we chose to compare our model. The second hybridized model (hybridized attention-based CNN-Bi-GRU model), this proposed model significantly improved each BLUE score and human evaluation by introducing an AM with the Bi-GRU decoder.

6.3.1. BLEU Score

From the experiment, the result demonstrated in Figure 6.3 and Figure 6.4 The capacity of the hybridized model improved in comparison with the basic model. For example, when comparing both 4G-BLEU scores, the hybridized model increased by 13%. It indicates more

descriptive image captions produce. Thus, visual attention and Bi-GRU with AM have made a significant contribution to image caption by attending to the appropriate part of the picture and selecting words that are relevant to the image caption. When generating the next word, the Bi-GRU with an attention decoder can use to produce words that are more suitable to the image.

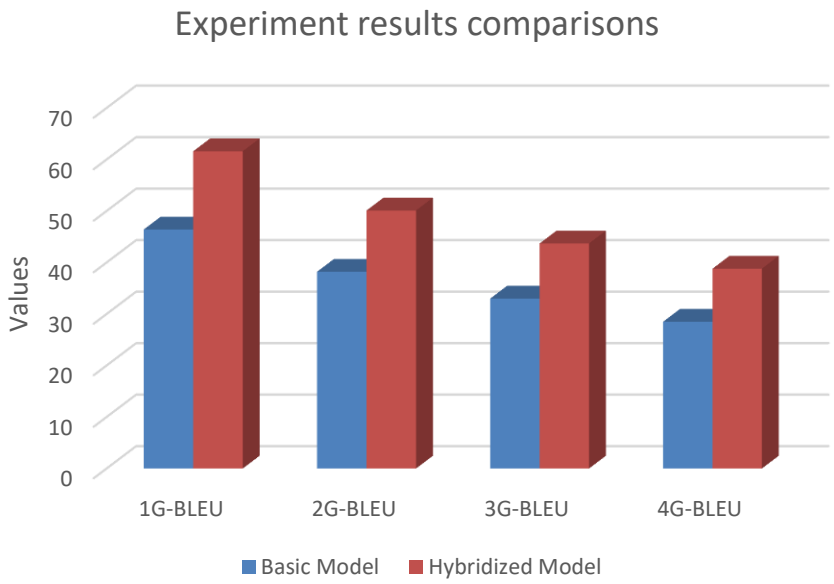
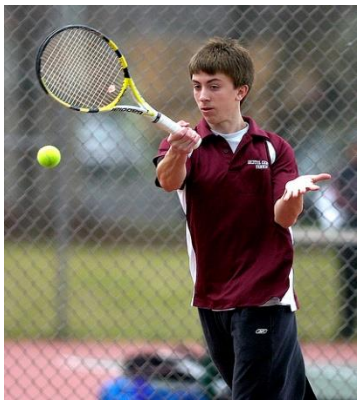


Figure 6.5 Experiment results comparisons

To contrast the results of the two models, let's look at some of the results of image captions as follows:

Table 6.4 Examples of models generated captions results with BLEU

Images	Captions	4G-BLUE
Example 1: 	**Reference:** ውሻ በአፉ ኪስ ይዞ ወደ ካሜራ ይሮጣል	
	CNN-Bi-GRU: ሰማያዊ ኪስ ይዞ የሚሮጥ ውሻ	8.9
	Hybridized model: ሰማያዊ ኪስ ይዞ ነጭ ውሻ በሣር የተሸፈነ ሜዳ ላይ እየሮጠ ነው	5.7

Example 2: 	**Reference:** ወንድና ሴት ፈገግ እያሉ ነው	
	CNN-Bi-GRU: ሁለት ሰዎች ይስቃሉ	0
	Hybridized model: አንድ ወንድና ሴት ይስቃሉ	13.1
Example 3: 	**Reference:** መጎናጸፍ ያለበት ቡናማ ውሻ ቀይ ኳስ እያሳደደ ነው	
	CNN-Bi-GRU: ቀይ ኳስ እያሳደደ ነው	36.8
	Hybridized model: ቡናማ ውሻ ቀይ ኳስ በሳር ላይ እያሳደደ ነው	71.6
Example 4: 	**Reference:** ሰማያዊ የዋና ልብስ የለበሰች አንዲት ሴት	
	CNN-Bi-GRU: አንዲት ሴት አረንጓዴ ውኃ ውስጥ ይራመዳል	22.1
	Hybridized model: ሰማያዊ የዋና ልብስ የለበሰች አንዲት ሴት በውሃ ውስጥ እየዋኘች ነው	51.7
Example 5: 	**Reference:** አንድ ልጅ በቴኒስ ውድድር የቴኒስ ኳስ መታ	
	CNN-Bi-GRU: አንድ ልጅ በአደባባይ የቴኒስ ኳስ መታ	32.4
	Hybridized model: አንድ ልጅ በቴኒስ ውድድር የቴኒስ ኳስ ይመታዋል	80.9

Overall, From the newly translated Amharic dataset, both models made good word predictions. However, the hybridized model can produce a more descriptive caption. BLEU score evaluation to indicate the similarity between the generated model captions (candidate texts) and the actual captions (texts in reference). Accordingly, in Table 6.4, the hybridized model scored better in BLEU scores. Sometimes the candidate text has high BLEU scores, but this score does not mean that the image is well described in semantic correctness. To illustrate this, we present some examples below:

Table 6.5 Examples from translated Flickr8k dataset

Images	Captions	4G-BLUE
Example 1: 	**Reference:** አንድ ቡናማ ውሻ ወደ ውቅያኖሱ እየሮጠ ነው	
	CNN-Bi-GRU: ቡናማ ውሻ ወደ ውሃ ውስጥ ዘለለ	28.5
	Hybridized mode: ቡናማና ነጭ ውሻ ወደ ውቅያኖስ ይሮጣል	18.7
Example 2: 	**Reference:** ሁለት ሰዎች በአንድ ውድድር ላይ ካራቴ እያከናወኑ ነው	
	CNN-Bi-GRU: ሁለት ወንዶች ልጆች መድረክ ላይ ካራቴ እያከናወኑ ነው	36.5
	Hybridized mode: ሁለት ሴቶች እርስ በርሳቸው ይጋጫሉ	10.9

6.3.2. Human Evaluation Study Results

We have compared the basic and hybridized models by employing human evaluation metrics to evaluate the generate Amharic captions or descriptions of the provided image that accurately describe the given image content. This review presents the user with six images and two corresponding descriptions. Users evaluate these descriptions and then give points to the sentences that accurately describe the image content. Finally, the average score calculates to determine the best model based on the results. The complete questionnaire for this shows at ANC. As an example, let's look at the questionnaire form as follows:

Human Evaluation Study

Instructions

In this task, you need to evaluate the description quality for the provided pictures.

Please read the picture and the descriptions precisely and give the point for each description by considering the sentence describes the image content correctly and grammatically accurately each sentence. **Question :** Which image description best describes the image?

Answer by indicating to **A or B.**

No.	Pictures	Candidate Descriptions/Captions	Point
1		**C1:** አንድ ሰው ፀሐይ ስትጠልቅ በባንዲራ አጠገብ በጀልባ ላይ ተቀምጧል	○ A
		C2: አንድ ሰው ፀሐይ ስትጠልቅ በመርከብ ላይ ተቀምጧል	○ B

Figure 6.6 Demonstrate the sample questionnaire for HES

In this questionnaire, volunteers mark the radio button that produces the best image description from C1 and C2. The first C1 refers to the basic model, while the second one or C2 is the result of a hybridized model. Twenty-four volunteers participated in this questionnaire, and six different pictures were presented with their model caption results. The first C1 caption selected a 38% average of all users, and the second C2 chose an average of 62%. The results of all six questions show in the following graph.

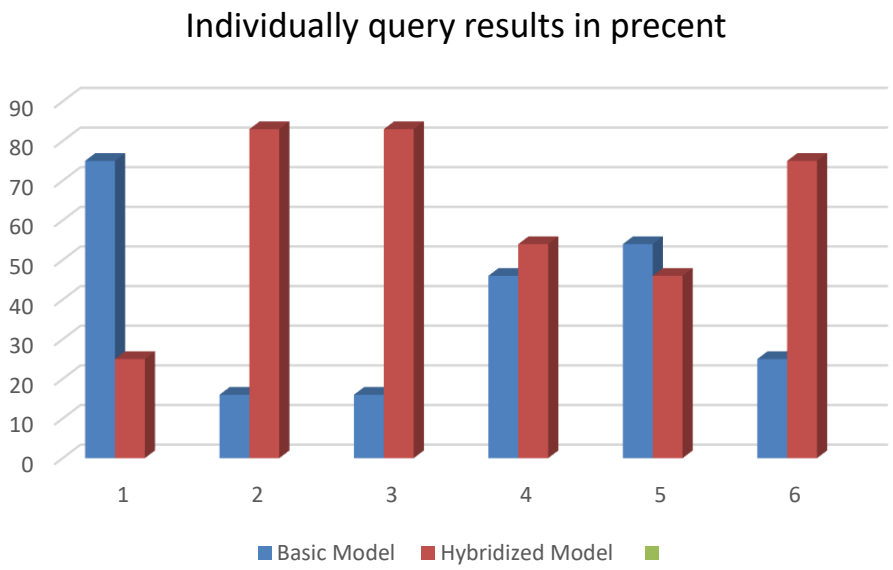


Figure 6.7 Each query results in precent

The two models' average results show in percent in the following figure:

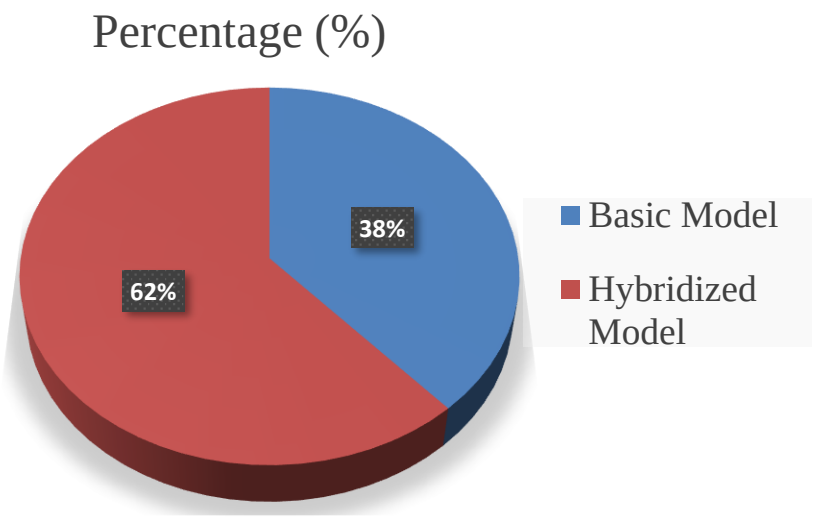


Figure 6.8 Shows the percentage of model results

The results of Figure 6.8 demonstrate that conform to the human evaluation matrix: the hybridized model averagely boosts by 24 % compared to the basic model. This hybridized model (proposed model) can produce more acceptable descriptions with semantically correct captions by comparison with the basic model.

The comparative experiment results on the BNATURE and Flickr8k English dataset presents as follows:

Table 6.6 BNATURE and Flickr8k datasets performance on the proposed model

Dataset	Model	1G-BLEU	2G-BLEU	3G-BLEU	4G-BLEU
BNATURE	CNN-Bi-GRU	42.6	27.9	23.6	16.4
	Hybridized model	**63.1**	**53.3**	**47.4**	**42.7**
Flickr8k English	Bag-LSTM	59.7	41.4	28.1	18.2
	Hybridized model	**65.6**	**53.2**	**45.3**	**39.4**

The results of Table 6.6 show that we have experimented with datasets with two different language structures. And we compared these two papers CNN-Bi-GRU (Faruk et al., 2020) and Bag-LSTM (Cao et al., 2019) as a baseline to find out how much space we have or the potential of our model. Our proposed model performed better on all BLEU scores on both dataset varieties. As the BNATURE dataset shows in this diagram: Our training accuracy got 0.925 and for testing reached 0.928. We observed approximately 0.186 loss in the train, and for the testing got 0.181. The results of the English Flickr8k dataset show from this diagram our training accuracy reached 0.882 and for testing got 0.885. We attended about the training loss of 0.303, and test loss reached about 0.295.

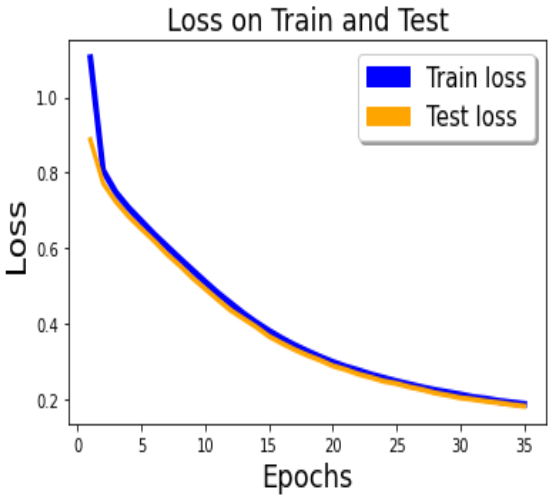


Figure 6.9 Performance of the model on the train and test loss against epoch for the BNATURE dataset

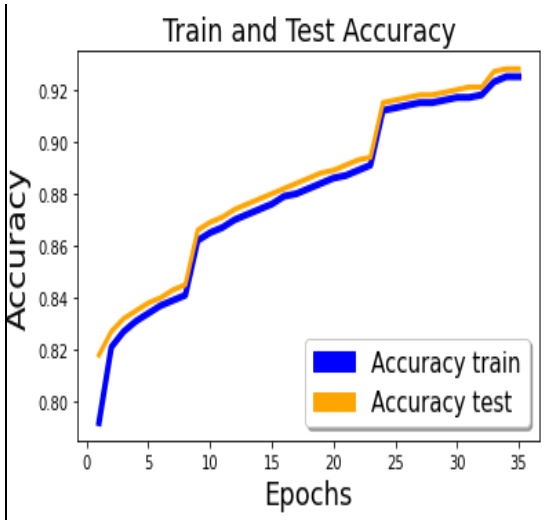


Figure 6.10 Performance of the model on the train and test accuracy against epoch for the BNATURE dataset

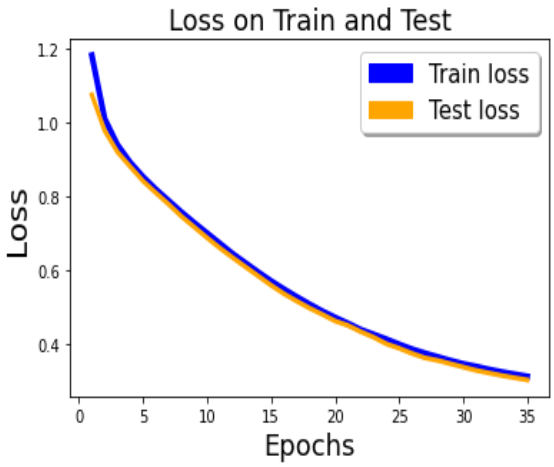


Figure 6.11 Model performance on train and test loss Vs epoch for Flickr8K English

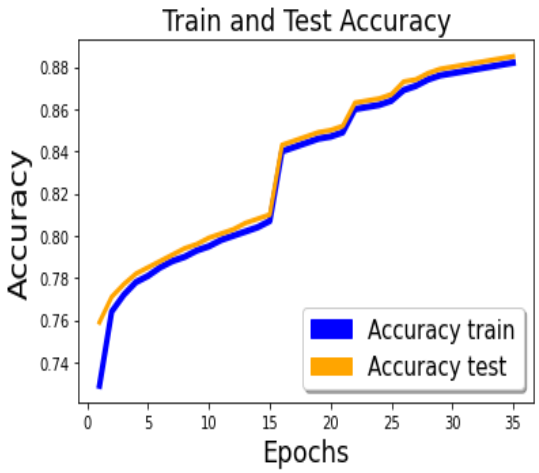


Figure 6.12 Model performance on train and test loss Vs epoch for Flickr8K English

Let's take observe the output captions using English Flickr8k dataset produced by the proposed model:

Table 6.7 Examples of captions generated by English Flickr8k dataset

Images	Captions	4G-BLUE
Example 1: 	**Reference:** two dogs fight on the grass	
	Hybridized model: two dogs are fighting on the grass	59.7
Example 2: 	**Reference:** a very small boy carries a bicycle tire on his shoulder	
	Hybridized model: a boy carries a bicycle tire on his shoulder	71.2

Developing a likely image caption in Bengali was made possible using the Flickr8k arrangement in the Bangladeshi language dataset (BNATURE). Also, achievable English language image descriptions using the Flickr8k datasets, which are used in the second baseline model and most commonly used for image captioning and other tasks. In general, datasets similar to the Flickr8k dataset negatively affect the capacity of models.

In this study, one of the primary problems we encountered with the dataset was its small size. Because of the small amount of this translated Flickr8k dataset was a big problem (challenging) to train our DNN model. This small size dataset has made the model suffer from an overfitting problem. Due to this, we applied two effective techniques such as dropout, and regularization to address the overfitting problem. Also, we monitored the training and testing loss function values. Both train and test values must decrease during the process of training. Otherwise, if the testing set increases, the training process will halt and re-train by a new hyperparameters configuration.

The second problem we faced was the preparation problem associated with the new translated Amharic dataset. The dataset prepares if there are sentences that do not follow the grammatical rules (sentence structure) or if the words are not evenly distributed for the training and testing set. It leads to reducing the performance of the model. Due to this, we tried to manually revise the translated sentences in the dataset to prevent grammatical errors.

However, despite these problems, the proposed model still performs competitive performance on those datasets. We first compared the sematic attention-based Bi-LSTM (Bag-LSTM model) using English Flickr8k dataset, and our hybridized model achieved better results at BLEU scores. It is 6%, 12%, 17%, and 21% higher than Bag-LSTM model on 1G-BLEU, 2G-BLEU, 3G-BLEU, and 4G-BLEU. Again, the Bengali language bidirectional GRU-based captions model compares to the hybridized model using the BNATURE dataset. Therefore, our hybridized model is 1G-BLEU, 2G-BLEU, 3G-BLEU, and 4G-BLEU increased by 20%, 25%, 24%, and 21%.

6.4. Results Discussion on Hybridized Model

Our experiment conducted outcome shows using the translated Amharic dataset, the hybridized model obtained better results in qualitative and quantitative.

The first experiment had a CNN-Bi-GRU encoder-decoder approach. The pre-train Inception-V3 CNN designates image features, and Bi-GRU is an employed language decoder for representing words for the generate captions.. This model introduces the Bi-GRU model. It is a sequence processing model consisting of two GRU layers used to process information in forwarding and backward directions with efficient performance. As an initial point, this model is effective in the translated Amharic language dataset. However, the outcomes of the predictions shown in Table 6.4 when evaluating the qualitative results did not yield acceptable results. This model is based on the overall image information when generating an image caption. It leads to the fact that the resulting caption does not contain all the essential details of the image.

The second experiment differs from the first it added an AM. Visual attention is between the encoder-decoder, which focuses on the essential areas of the image features and provides those features to the language decoder. Next up is Bi-GRU follows an AM that obtains visual attention integrated with word embedding inputs. Based on this, the final result determines by combining the Bi-GRU output with the extended AM output. This hybrid

effect played an essential role in generating the most relevant image captions for the given image. The results of the hybridized model caption in Table 6.4 show the quality of the predicted words that accurately describe the image content compared to the first CNN-Bi-GRU. Quantitative results indicate: that it improves by 10%, which is higher than the CNN-Bi-GRU model on 4G-BLEU. It demonstrates that caption by the hybridized model is more equivalent to the actual captions.

Another thing we have noticed is that our hybridized model has brought significant improvements compared to other bidirectional and attention-based models. It enhances by 21%, which is higher than the CNN-Bi-GRU model on 4G-BLEU. Also, it improves by 21%, which is higher than the Bag-LSTM model on 4G-BLEU.

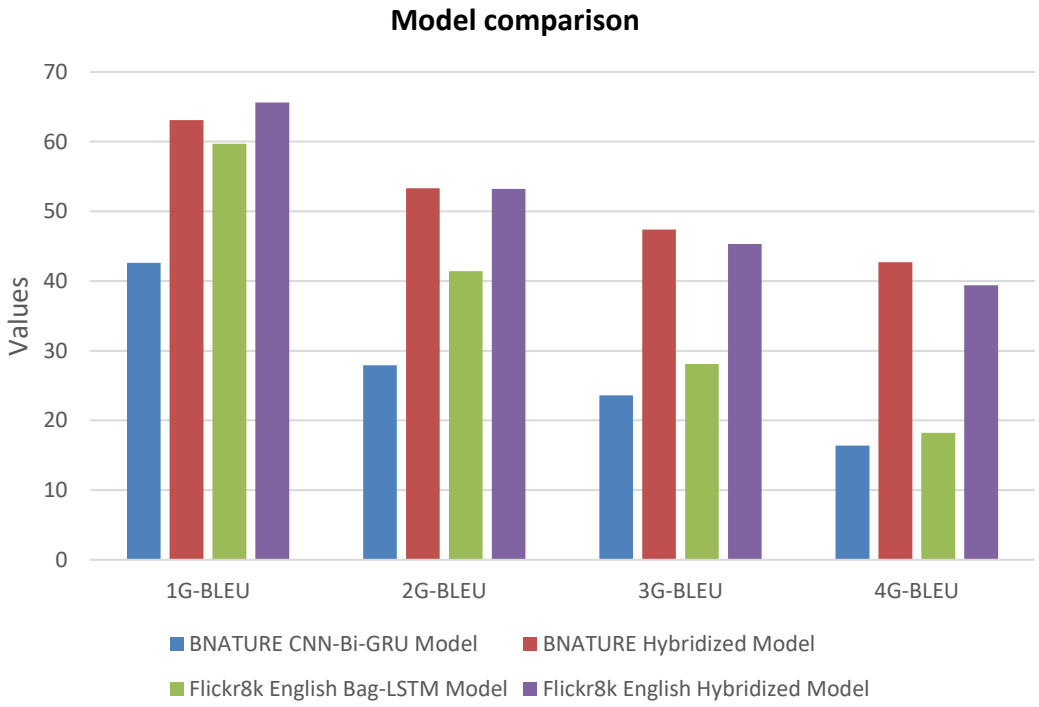


Figure 6.13 The proposed model comparison across different datasets

6.4.1. Research Question Discussion

The research question raises in the first chapter. According to their order in this section, those questions answer as follows:

Answer for question 1 : Inception-v3 and Bi-GRU DNNs are successfully identified for the proposed model encoder-decoder framework.

Inception-v3 is the third version of inception CNN developed by Google. This network has already trained over one million images from the ImageNet. Also, it is more efficient compared to similar pre-train CNN models such as AlexNet, VGGNet, or ResNet. The principal reason for this is that the Inception-v3 parameter reduces by almost half of the other DNN models.

Bi-GRU is another type of Bi-LSTM. Bi-GRU networks have a simple configuration and better speed compared to Bi-LSTM. The cause for this, it only uses two gates, the update and reset, so the calculation speed is maximum. Image descriptions generate employing the above-mentioned usable DNNs. Via the encoder, Inception-v3 was used to obtain visual features. And the decoder part or Bi-GRU is used to train sequence information (textual data) to predict words.

Answer for question 2 : To improve the encoder-decoder image caption framework: the visual attention and Bi-GRU configured with an AM identifies better for the constraints. Visual attention provides better image representation by capturing only the essential features of the image. Another Bi-GRU collaboration with the AM reduces the gap between the two visual and textual features. Also, it generates a caption that captures the semantics of the sentence for the image.

Answer for question 3 : Experimental results show that attention-based Bi-GRU or hybridized model achieved better image captioning. Bi-GRU learns only the essential features of the image with the corresponding text features. Next, the AM focuses on Bi-GRU output and hidden state. Finally, combining the AM result and the Bi-GRU outcome is used to predict relevant words that describe more suitable image content. To show the efficiency of the attention-based Bi-GRU decoder, we conduct separate experiments with the AM and without involving the AM. If we take the 4G-BLEU score for comparison, the attention-based Bi-GRU decoder increased the 4G-BLEU score from 28.5 to 38.8, which 0.10% increased performance.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

Amharic image caption provides significant benefits for a variety of Amharic language-based applications. In the case of captioning, the important thing is to understand the two distinct domains of image and text to generate suitable grammatical and meaningful captions that accurately describe the content of a given image. Merging these two domains with different features is a difficult task in generating an image caption. Despite this challenge, various methods have been used to yield a better image caption in recent years. One of these methods is the encoder-decoder approach adopted from MT, and it made a significant contribution to the image captioning tasks. The encoder-decoder image descriptions have achieved remarkable results by applying various enhancement techniques. However, these approaches primarily establish either the image or the text elements. So that there is a widening of the modality gap between visual and textual features, which becomes the generate captions will be some content distortion in the grammatical and semantic correctness.

The CNN-Bi-GRU approach uses the encoder through Inception-V3 CNN, to obtain picture features, and the Bi-GRU decoder utilizes for generating captions. In this methodology, the captions generate based on the overall visual elements or the whole picture information, which leads to some details of the image not describes, and the grammatically correct descriptions will not generate. Another approach is the semantic attention model utilized. Semantic attention aims to provide a semantic guide for the LSTM decoder by producing visual features closely related to textual information. Semantic and visual attention-based encoder-decoder models generate a caption that is either dependent on the parts of the image or text, so they are limited to producing words that are better related to the visual content. This study proposed to address these issues by hybridized attention-based NNs.

In this work, we proposed hybridized attention-based CNN-Bi-GRU model for Amharic image captioning. The model approach is an encoder-decoder framework that consists of three main parts to produce a well-defined and accurate description of a visual correlation that clearly describes the image. The first is the image encoder, which uses to learn the

extracting image features. We used the pre-train Inception-V3 CNN model to extract image features by pulling out the SoftMax layer, which is the last layer of Inception-V3. The second part is visual attention, and it applies to these extracted features. Visual attention uses to concentrate only on the essential areas of images. The last third is the language decoder or Bi-GRU with an AM. Bi-GRU is another type of LSTM that learns two-way (bidirectional) long-term dependency between sequential information efficiency. The Bi-GRU input is a mixture of context vector (AM output) and textual features from word embedding. It helps to reduce the gap between visual and textual attributes. We have extended attention to Bi-GRU to produce more grammatical and semantical image captions. This strategy allows the model to select words that nicely describe the image by focusing on Bi-GRU output and hidden state. Finally, incorporating the Bi-GRU and the extended AM outputs, which uses to generate the final image captions. Thus, the experiment showed that Bi-GRU with attention language decoder helps the proposed model enhance the image captions in Amharic using the translated Flickr8k dataset.

Compare the proposed model with baseline models on different benchmarks dataset across the BLUE score evaluation matrix. Compared with the bassline work CNN-Bi-GRU, and Bag-LSTM. The experiment result shows the effectiveness of the proposed model. Using the BNATURE dataset, the proposed model improves the baseline work CNN-Bi-GRU. Its 1G-BLEU, 2G-BLEU, 3G-BLEU, and 4G-BLEU increased by 20%, 25%, 24%, and 21%. Also, the proposed model got a better BLEU score using the English Flickr8k dataset. It is 6%, 12%, 17%, and 21% higher than the Bag-LSTM model on 1G-BLEU, 2G-BLEU, 3G-BLEU, and 4G-BLEU. This work concludes that extended visual attention and Bi-GRU integrated with AM decoder are constraints to generating image caption does improve the semantics of the generated descriptions.

7.2. Future Work

This work is an initial point for producing a better image description in Amharic. The Flickr8k dataset (English) translated into Amharic is the main contributor to this proposed model. There were various challenges during the translation of the caption into Amharic, and the images represent a foreign culture. Some of these were words that did not have Amharic meaning, Grammatical errors, and the lack of suitable words to describe the image. These negatively impact the potential of the proposed model.

In future work, we plan to make our dataset like the Flickr8k English dataset by collecting images that represent different cultures of our country from various sources and preparing five Amharic annotation sentences for each image. Additionally, we intend to apply the proposed model in the Amharic language for another task, such as video activity recognition.

SPECIAL ACKNOWLEDGMENTS

This research project is funded by Adama Science and Technology University (ASTU) under the grant number:

ASTU/SM-R/383/21

Adama, Ethiopia

References

Agughalam, Davis Munachimso. 2020. "Bidirectional LSTM Approach to Image Captioning with Scene Features Davis Munachimso Agughalam Supervisors : Dr Paul Stynes."

Agughalam, Davis Munachimso. n.d. "Bidirectional LSTM Approach to Image Captioning with Scene Features Davis Munachimso Agughalam Supervisors : Dr Paul Stynes."

Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. 2021. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*. Vol. 8. Springer International Publishing.

Avramidis, Eleftherios, and Philipp Koehn. 2008. "Enriching Morphologically Poor Languages for Statistical Machine Translation." *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference* (June):763–70.

Bahdanau, Dzmitry, Kyung Hyun Cho, and Yoshua Bengio. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate." *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* 1–15.

Bai, Shuang, and Shan An. 2018. "A Survey on Automatic Image Caption Generation." *Neurocomputing* 311:291–304. doi: 10.1016/j.neucom.2018.05.080.

Bekele, A. A. 2020. "Automatic Generation of Amharic Math Word Problem and Equation." *Journal of Computer and Communications* 59–77. doi: 10.4236/jcc.2020.88006.

Bernardi, Raffaella, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2017. "Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures." *IJCAI International Joint Conference on*

Artificial Intelligence **0:4970–74.**

Biadgling, Aden Darge. 2021. "Developing Deep RNN-Based Client Resource Utilization Prediction Model for an Improved Cloud Resource Provisioning."

Cao, Pengfei, Zhongyi Yang, Liang Sun, Yanchun Liang, Mary Qu Yang, and Renchu Guan. 2019. "Image Captioning with Bidirectional Semantic Attention-Based Guiding of Long Short-Term Memory." ***Neural Processing Letters*** **50(1):103–19. doi: 10.1007/s11063-018-09973-5.**

Chen, Long, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat Seng Chua. 2017. "SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning." ***Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*** **2017-Janua:6298–6306. doi: 10.1109/CVPR.2017.667.**

Fan, Feng-Lei, Jinjun Xiong, Mengzhou Li, and Ge Wang. 2021. "On Interpretability of Artificial Neural Networks: A Survey." ***IEEE Transactions on Radiation and Plasma Medical Sciences*** **5(6):741–60. doi: 10.1109/trpms.2021.3066428.**

Faruk, Al Momin, Hasan Al Faraby, Md Muzahidul Azad, Md Riduyan Fedous, and Md Kishor Morol. 2020. "Image to Bengali Caption Generation Using Deep CNN and Bidirectional Gated Recurrent Unit." ***ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings*** **19–21. doi: 10.1109/ICCIT51783.2020.9392697.**

Fikre, Turegn, and Addis Ababa. 2020. "Effect of Preprocessing on Long Short Term Memory Based Sentiment Analysis for Amharic Language."

Graves, Alex. 2013. "Generating Sequences With Recurrent Neural Networks." 1–43.

Herdade, Simao, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. "Image Captioning: Transforming Objects into Words." ***ArXiv*** **(NeurIPS):1–11.**

Humaira, Mayeesha, Shimul Paul, Md Abidur Rahman Khan Jim, Amit Saha Ami, and Faisal Muhammad Shah. 2021. "A Hybridized Deep Learning Method for Bengali Image Captioning." ***International Journal of Advanced***

Computer Science and Applications 12(2):698–707. doi: 10.14569/IJACSA.2021.0120287.

Hussain, Mahbub, Jordan J. Bird, and Diego R. Faria. 2019. "A Study on CNN Transfer Learning for Image Classification." *Advances in Intelligent Systems and Computing* 840(June):191–202. doi: 10.1007/978-3-319-97982-3_16.

Hutchison, David, and John C. Mitchell. 1973. *Lecture Notes in Computer Science (ECCV2010)*. Vol. 9.

Jana, Ranjan, Siddhartha Bhattacharyya, and Swagatam Das. 2020. "Handwritten Digit Recognition Using Convolutional Neural Networks." *Deep Learning: Research and Applications* 51–68. doi: 10.1515/9783110670905-003.

Jang, Beakcheol, Myeonghwi Kim, Gaspard Harerimana, Sang Ug Kang, and Jong Wook Kim. 2020. "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism." *Applied Sciences (Switzerland)* 10(17). doi: 10.3390/app10175841.

Jayagopal, Vellingiri, and Basser K. K. 2021. "Data Management and Big Data Analytics." *Research Anthology on Big Data Analytics, Architectures, and Applications* 1614–33. doi: 10.4018/978-1-6684-3662-2.ch078.

Jishan, Asifuzzaman. n.d. "Bangla Language Textual Image Description by Hybrid Neural Network Model." doi: 10.11591/ijeecs.v21.i2.pp757-767.

Kesito, Abeto Alemu. 2018. "Character Recognition of Bilingual Amharic-Latin Printed Documents." (November).

Kiros, Ryan, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models." 1–13.

Kuznetsova, Polina, Vicente Ordonez, Alexander Berg, Tamara Berg, and Yejin Choi. 2013. "Generalizing Image Captions for Image-Text Parallel Corpus." *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference* 2:790–96.

Kuznetsova, Polina, Vicente Ordonez, Alexander C. Berg, Tamara L. Berg, and

Yejin Choi. 2012. "Collective Generation of Natural Image Descriptions." ***50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*** **1(July):359–68.**

Lauriola, Ivano, Alberto Lavelli, and Fabio Aiolli. 2021. "An Introduction to Deep Learning in Natural Language Processing: Models, Techniques, and Tools." ***Neurocomputing*** **(xxxx). doi: 10.1016/j.neucom.2021.05.103.**

Li, Pengpeng, An Luo, Jiping Liu, Yong Wang, Jun Zhu, Yue Deng, and Junjie Zhang. 2020. "Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation." ***ISPRS International Journal of Geo-Information*** **9(11). doi: 10.3390/ijgi9110635.**

Li, Shuai, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. 2018. "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN." ***Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*** **5457–66. doi: 10.1109/CVPR.2018.00572.**

Li, Siming, Girish Kulkarni, Tamara L. Berg, Alexander C. Berg, and Yejin Choi. 2011. "Composing Simple Image Descriptions Using Web-Scale N-Grams." ***CoNLL 2011 - Fifteenth Conference on Computational Natural Language Learning, Proceedings of the Conference*** **(June):220–28.**

Liu, Xiaoxiao, Qingyang Xu, and Ning Wang. 2019. "A Survey on Deep Neural Network-Based Image Captioning." ***Visual Computer*** **35(3):445–70. doi: 10.1007/s00371-018-1566-y.**

Lu, Jiasen, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. "Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning." ***Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*** **2017-Janua:3242–50. doi: 10.1109/CVPR.2017.345.**

Lv, Pin, Song Liu, Wenbing Yu, Shuquan Zheng, and Jing Lv. 2020. "EGA-STLF: A Hybrid Short-Term Load Forecasting Model." ***IEEE Access*** **8:31742–52. doi: 10.1109/ACCESS.2020.2973350.**

Martínez, Mario Gómez, Anna Bosch Rué, and Casas Roma Jordi. 2019. "Deep Learning for Image Captioning An Encoder-Decoder Architecture with Soft Attention."

Masotti, Caterina, Danilo Croce, and Roberto Basili. 2018. "Deep Learning for Automatic Image Captioning in Poor Training Conditions." ***Italian Journal of Computational Linguistics*** **4(1):43–55. doi: 10.4000/ijcol.538.**

Niu, Zhaoyang, Guoqiang Zhong, and Hui Yu. 2021. "Neurocomputing." ***Neurocomputing*** **452:48–62. doi: 10.1016/j.neucom.2021.03.091.**

Ordonez, Vicente, Girish Kulkarni, and Tamara L. Berg. 2011. "Im2Text: Describing Images Using 1 Million Captioned Photographs." ***Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*** **1–9.**

Pedersoli, Marco, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. 2017. "Areas of Attention for Image Captioning." ***Proceedings of the IEEE International Conference on Computer Vision*** **2017-Octob(ii):1251–59. doi: 10.1109/ICCV.2017.140.**

Peng, Yuqing, Xuan Liu, Weihua Wang, Xiaosong Zhao, and Ming Wei. 2019. "Image Caption Model of Double LSTM with Scene Factors." ***Image and Vision Computing*** **86:38–44. doi: 10.1016/j.imavis.2019.03.003.**

Phukan, Borneel Bikash, and Amiya Ranjan Panda. 2020. "An Efficient Technique for Image Captioning Using Deep Neural Network." ***ArXiv*****.**

Qin, Yu, Jiajun Du, Yonghua Zhang, and Hongtao Lu. 2019. "Look Back and Predict Forward in Image Captioning." ***Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*** **2019-June:8359–67. doi: 10.1109/CVPR.2019.00856.**

R, Sreela S., and Sumam Mary Idicula. 2019. "The Significance of Self-Attention over LSTM in Image Captioning." ***International Journal of Applied Engineering Research*** **14(16):3500–3502.**

Sangeetha, V., and K. J. Rajendr. Prasad. 2006. "Deep Residual Learning for Image Recognition." ***Indian Journal of Chemistry - Section B Organic and***

Medicinal Chemistry **45(8):1951–54. doi: 10.1002/chin.200650130.**

Schmidt, Robin M. 2019. "Recurrent Neural Networks (RNNs): A Gentle Introduction and Overview." (1):1–16.

Seshadri, Madhavan, Malavika Srikanth, and Mikhail Belov. 2020. "Image to Language Understanding: Captioning Approach." ***ArXiv*****.**

Seyoum, Binyam Ephrem, Yusuke Miyao, and Baye Yimam Mekonnen. 2019. "Universal Dependencies for Amharic." ***LREC 2018 - 11th International Conference on Language Resources and Evaluation*** **(May):2216–22.**

Sze, Vivienne, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. 2020. "Efficient Processing of Deep Neural Networks." ***Synthesis Lectures on Computer Architecture*** **15(2):1–341. doi: 10.2200/s01004ed1v01y202004cac050.**

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." ***Advances in Neural Information Processing Systems*** **2017-Decem(Nips):5999–6009.**

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. "Show and Tell: A Neural Image Caption Generator." ***Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*** **07-12-June:3156–64. doi: 10.1109/CVPR.2015.7298935.**

Wang, Cheng, Haojin Yang, Christian Bartz, and Christoph Meinel. 2016. "Image Captioning with Deep Bidirectional LSTMs." ***MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*** **988–97. doi: 10.1145/2964284.2964299.**

Wang, Cheng, Haojin Yang, and Christoph Meinel. 2018. "Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning." ***ACM Transactions on Multimedia Computing, Communications and Applications*** **14(2s). doi: 10.1145/3115432.**

Wo, Krzysztof, and Krzysztof Marasek. 2016. "Enhanced Bilingual Evaluation Understudy."

Wondwossen Mulugeta. 2014. "A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts." ***HiLCoE Journal of Computer***

Science and Technology **2(2):80–87.**

Xiao, Fen, Xue Gong, Yiming Zhang, Yanqing Shen, Jun Li, and Xieping Gao. 2019. "DAA: Dual LSTMs with Adaptive Attention for Image Captioning." ***Neurocomputing*** **364:322–29. doi: 10.1016/j.neucom.2019.06.085.**

Xiao, Xinyu, Lingfeng Wang, Kun Ding, Shiming Xiang, and Chunhong Pan. 2019. "Dense Semantic Embedding Network for Image Captioning." ***Pattern Recognition*** **90:285–96. doi: 10.1016/j.patcog.2019.01.028.**

Xu, Kelvin, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." ***32nd International Conference on Machine Learning, ICML 2015*** **3:2048–57.**

Yang, Shudong, Xueying Yu, and Ying Zhou. 2020. "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example." ***Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*** **98–101. doi: 10.1109/IWECAI50956.2020.00027.**

Yang, Yezhou, Ching Lik Teo, Hal Daumé, and Yiannis Aloimonos. 2011. "Corpus-Guided Sentence Generation of Natural Images." ***EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*** **444–54.**

Yin, Wenpeng, and Hinrich Schütze. 2018. "Attentive Convolution: Equipping CNNs with RNN-Style Attention Mechanisms." ***Transactions of the Association for Computational Linguistics*** **6:687–702. doi: 10.1162/tacl_a_00249.**

Yuan, Chenxi, Yang Bai, and Chun Yuan. 2021. "Bridge the Gap: High-Level Semantic Planning for Image Captioning." 3157–67. doi: 10.18653/v1/2020.coling-main.281.

Yucong, Qiao, and Ma Li. 2021. "Image Caption Based on Bigru and Attention Hybrid Model." ***ACM International Conference Proceeding Series*** **128–36. doi: 10.1145/3488933.3488978.**

Zakir Hossain, M. D., Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2018. "A Comprehensive Survey of Deep Learning for Image Captioning." ***ArXiv*** **0(0).**

Zhang, Xiaodan, Shengfeng He, Xinhang Song, Rynson W. H. Lau, Jianbin Jiao, and Qixiang Ye. 2020. "Image Captioning via Semantic Element Embedding." ***Neurocomputing*** **395(xxxx):212–21. doi: 10.1016/j.neucom.2018.02.112.**

Zhou, Peng, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification." ***54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*** **207–12. doi: 10.18653/v1/p16-2034.**

APPENDIXES

Appendix A: Translated Dataset

The translated captions sample we used for Amharic captioning is available here: https://drive.google.com/file/

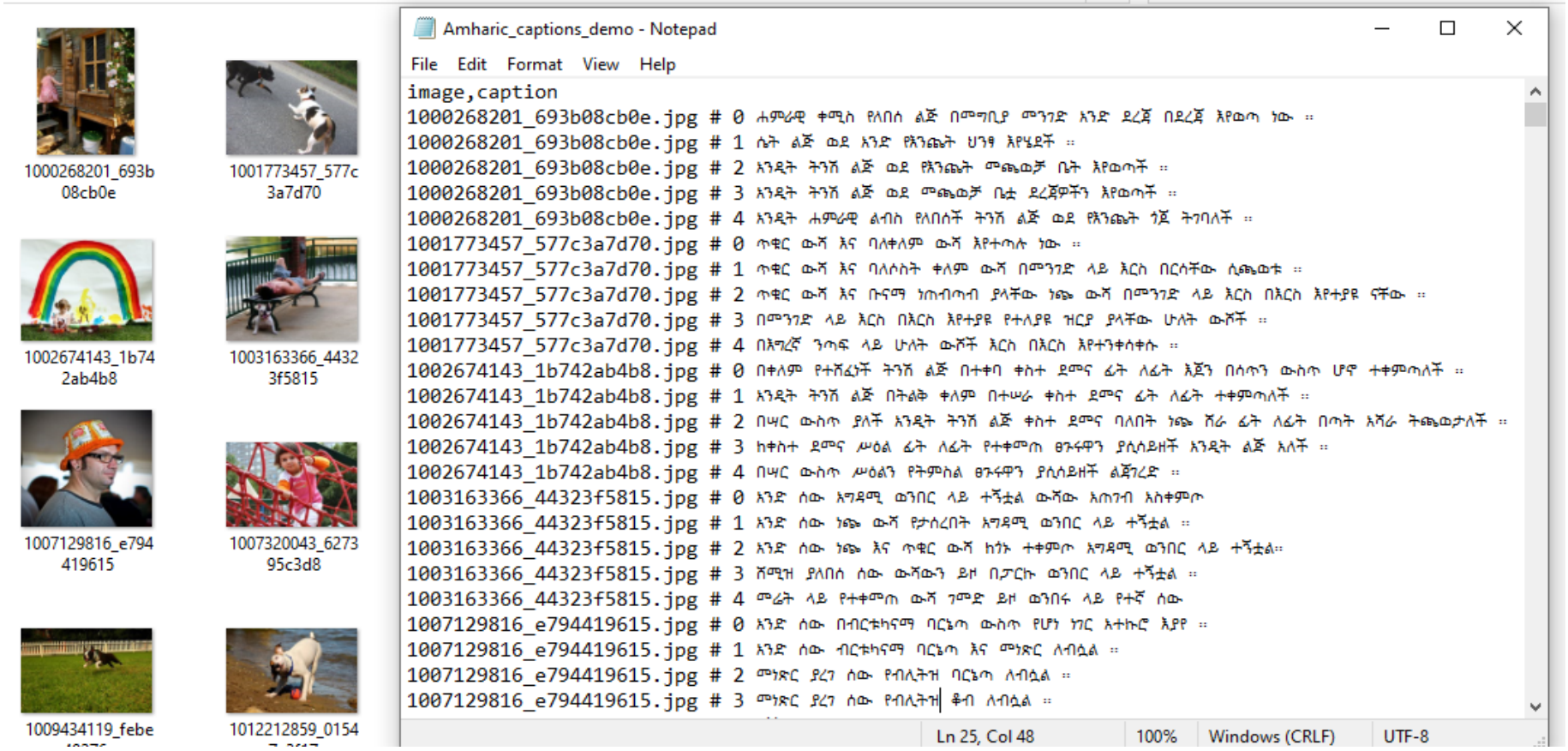


Figure 0.1 Sample translated caption dataset

Appendix B: Samples of Visual Attention Plot for Generated Words

Actual Caption: ነጭ ውሻ በበረዶው ውስጥ አንድ ነገር በአፉ ይይዛል

Prediction Caption: አንድ ውሻ በአፉ ውስጥ እንጨት የሚመስል ነገር ይዞ በበረዶው ውስጥ እየሮጠ ነው

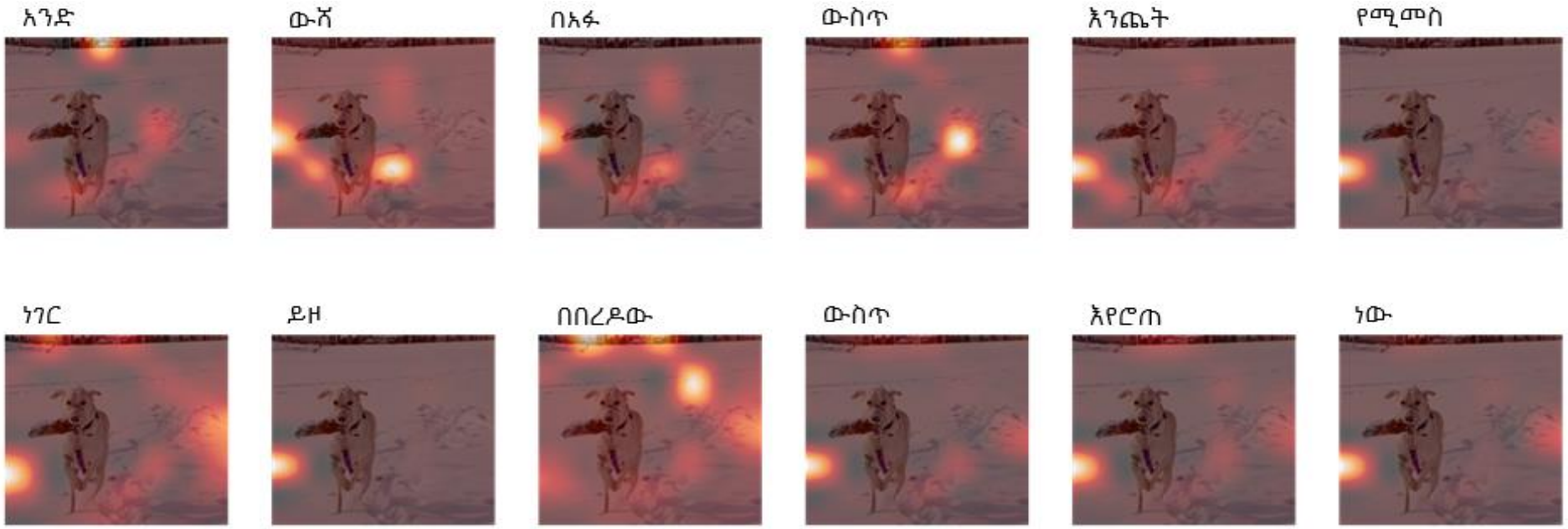


Figure 0.2 Visual attention plot 1

Actual Caption: አንድ ሰው በዐለት ወይም በተራራ አናት ላይ ቆሞ ፀሐይ ስትጠልቅ እየተመለከተ ነው

Prediction Caption: አንድ ሰው በተራራ አናት ላይ ቆሞ ፀሐይ ስትጠልቅ እየተመለከተ ነው

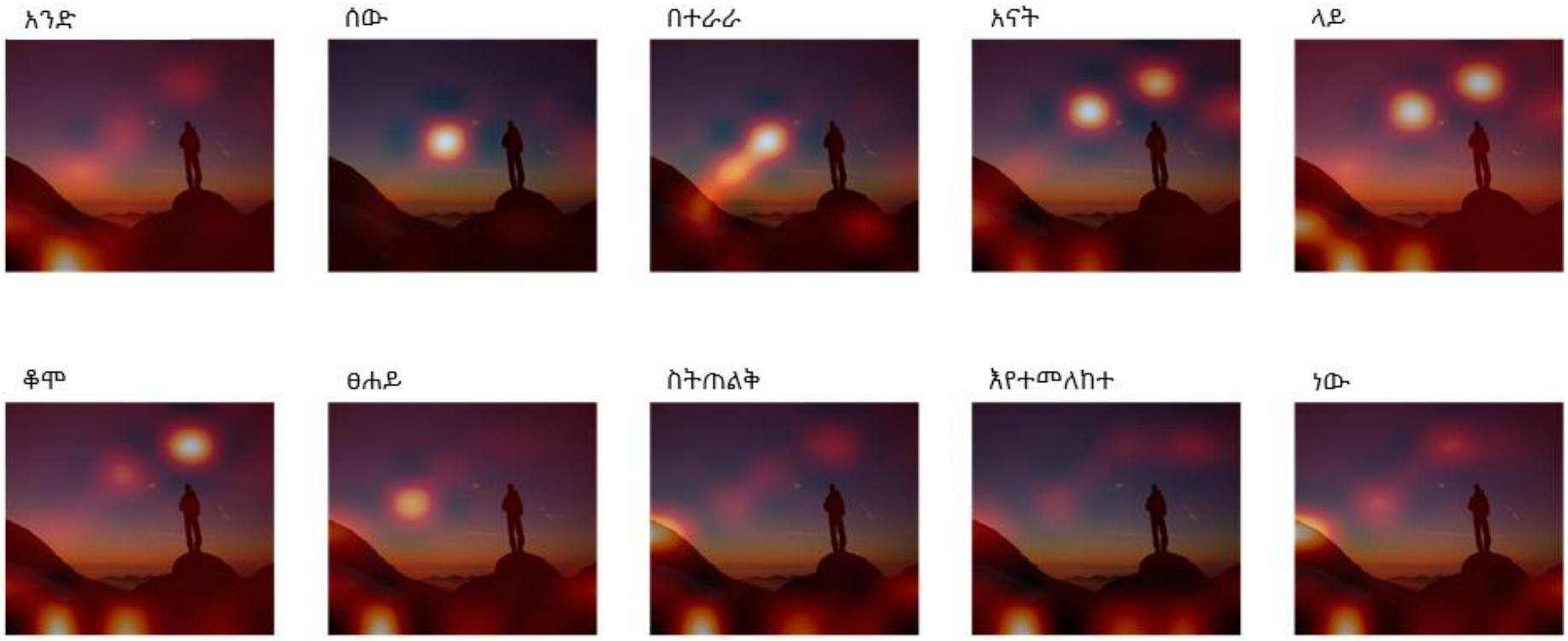


Figure 0.3 visual attention plot 2

	**Actual Caption:** a medium sized child jumps off of a dusty bank over a creek **Prediction Caption:** a boy jumps off of a dusty bank over a creek

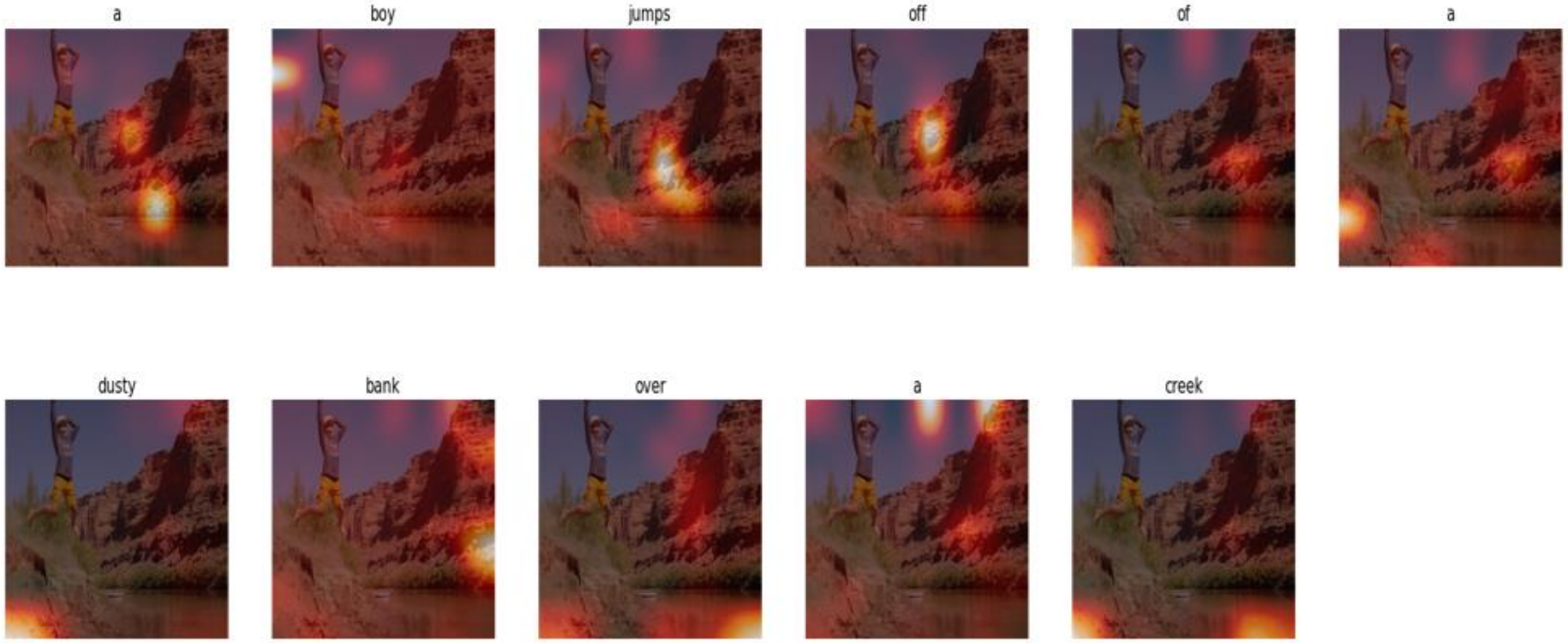


Figure 0.4 Visual attention plot 3

Appendix C: Questioner Form for Human Evaluation Study

Human Evaluation Study

Instructions

In this task, you need to evaluate the description quality for the provided pictures.

Please read the picture and the descriptions precisely and give the point for each description by considering the sentence describes the image content correctly and grammatically accurately each sentence. **Question :** Which image description best describes the image?

Answer by indicating to **A or B.**

No.	Pictures	Candidate Descriptions/Captions	Point
1		**C1:** አንድ ሰው ፀሐይ ስትጠልቅ በባንዲራ አጠገብ በጀልባ ላይ ተቀምጧል	○ A
		C2: አንድ ሰው ፀሐይ ስትጠልቅ በመርከብ ላይ ተቀምጧል	○ B
2		**C1:** ሁለት ሰዎች ይስቃሉ	○ A
		C2: አንድ ወንድና ሴት ይስቃሉ	○ B

3		**C1:** ሁለት ወንዶች ልጆች መድረክ ላይ ካራቴ እያከናወኑ ነው	A
		C2: ሁለት ሴቶች እርስ በርሳቸው ይጋጫሉ	B
4		**C1:** አንድ ወንድና አንዲት ሴት ልጆች በመንገድ ላይ ይሄዳሉ	A
		C2: አንድ ወንድና አንዲት ሴት በባህር ዳር ባዶ በሆነ የባሕር ዳርቻ ላይ ይራመዳሉ	B
5		**C1:** ሁለት ሴቶች መንገድ ተሻግረው ይሄዳሉ	A
		C2: ቀሚስ የለበሱ ሁለት ሴቶች መንገድ ላይ ይራመዳሉ	B
6		**C1:** ቡናማ ውሻ ወደ ውሃ ውስጥ ዘለለ	A
		C2: ቡናማና ነጭ ውሻ ወደ ውቅያኖስ ይሮጣል	B

Appendix D: Define Bi-GRU with AM decoder

```
class Decoder(tf.keras.Model):
 def __init__(self, embedding_dim, units, vocab_size):
   super(Decoder, self).__init__()
   self.units = units
   self.w_1 = tf.keras.layers.Dense(units,activation='relu')
   self.w_2 = tf.keras.layers.Dense(units,activation='relu')
   self.embedding = tf.keras.layers.Embedding(vocab_size,
                                              embedding_dim)
   # Bidirectional GRU
   self.bigru = tf.keras.layers.Bidirectional(
       tf.keras.layers.GRU
       (self.units,return_sequences=True,
        return_state=True,
        dropout=0.2,recurrent_dropout=0.2,
        recurrent_initializer=tf.keras.initializers.
        glorot_uniform(seed=26),
        kernel_regularizer=tf.keras.regularizers.l2(0.001),
        recurrent_regularizer=tf.keras.regularizers.l2(0.1)))

   self.fc1 = tf.keras.layers.Dense(self.units)

   self.dropout = tf.keras.layers.Dropout(0.3,
                                          noise_shape=None, seed=None)
   self.batchnormalization = tf.keras.layers.
   BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001,
                      center=True, scale=True, beta_initializer='zeros',
                      gamma_initializer='ones', moving_mean_initializer='zeros',
                      moving_variance_initializer='ones', beta_regularizer=None,
                      gamma_regularizer=None, beta_constraint=None,
                      gamma_constraint=None)
```

```

   self.fc2 = tf.keras.layers.Dense(vocab_size)

   # Implementing Attention Mechanism
   self.Uattn = tf.keras.layers.Dense(units)
   self.Wattn = tf.keras.layers.Dense(units)
   self.Vattn = tf.keras.layers.Dense(1)
   self.attention = tf.keras.layers.AdditiveAttention(self.
                                                      units,
                                                      causal = True)
   self.concat = tf.keras.layers.Concatenate()
   self.flatten = tf.keras.layers.Flatten()
 def call(self, x, features, hidden):
```

```
    hidden_with_time_axis = tf.expand_dims(hidden, 1)
    score = self.Vattn(tf.nn.tanh(self.Uattn(features) +
                                  self.Wattn(hidden_with_time_axis)))
    attention_weights = tf.nn.softmax(score, axis=1)
    context_vector = attention_weights * features
    context_vector = tf.reduce_sum(context_vector, axis=1)

    dec_emb = self.embedding(x)
    # Combine Word embedding with a context vector
    enc_emb = tf.concat([tf.expand_dims(context_vector, 1),
                         dec_emb], axis=-1)
    (bigru_outputs, forward_h, backward_h) = self.bigru(enc_emb)
    # Merged forward with a backward hidden layer
    state_h = self.concat([forward_h, backward_h])
    bigru_outputs = self.w_1(bigru_outputs)
    state_h = self.w_1(state_h)
    # AdditiveAttention
    attn_out = self.attention([bigru_outputs,state_h])
    # Combine Bi-GRU with attention
    decoder_concat_output = self.concat([bigru_outputs,attn_out])
    output = self.flatten(decoder_concat_output)

    x = self.fc1(output)
    x= self.dropout(x)
    x= self.batchnormalization(x)
    x = self.fc2(x)
    return x, state_h, attention_weights
def init_state(self, batch_size):
    return tf.zeros((batch_size, self.units))
```