Attention Based Image Caption Generation

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

AYUSH KUMAR GUPTA

2K21/CSE/08

Under the supervision of

PROF. ANIL SINGH PARIHAR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May, 2023

M.Tech (Computer Science & Engineering)   Ayush Kumar Gupta   2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Ayush Kumar Gupta, Roll No. 2K21/CSE/08 student of M. Tech (Computer Science and Engineering), hereby declare that the Project Dissertation titled "**Attention Based Image Caption Generation**" which is being submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi, in partial fulfilment of requirements for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: New Delhi	**AYUSH KUMAR GUPTA**
Date:	(2K21/CSE/08)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I, hereby certify that the Project titled "**Attention Based Image Caption Generation",** which is submitted by Ayush Kumar Gupta, Roll No. 2K21/CSE/08, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is a record of the project work carried out by the student under my supervision. The work is based on research papers by Xu et al [1] and Liu et al. [2]. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: New Delhi **PROF. ANIL SINGH PARIHAR**

Date: **SUPERVISOR**

ABSTRACT

The task of generating comprehensive and elaborate descriptions for images, commonly referred to as image captioning, presents a formidable challenge. This involves the amalgamation of computer vision and natural language processing techniques to establish a connection between visual data and textual comprehension. The fundamental goal of image captioning is to develop models and algorithms capable of comprehending the information conveyed by an image, thereby generating captions that effectively and coherently portray the visual content of the image in a manner akin to human-like interpretation. The concept of deep learning is introduced as a potential solution for image captioning, with a specific emphasis on the utilization of convolutional neural networks (CNNs) to extract salient visual features and recurrent neural networks (RNNs) to generate descriptive captions. This approach highlights the integration of CNNs and RNNs within the framework of deep learning, enabling the fusion of visual and textual understanding to facilitate the image captioning process. Image captioning is far more challenging than tasks like object identification and image categorization. Usually, two pipelines are used in the process: the first pipeline performs the computer vision task, while the second pipeline covers the natural language processing task. Deep learning approaches can manage the aforementioned pipelines and can create captions for images that are more robust. For visually impaired people, image captioning is immensely helpful. Image captioning makes things more accessible and entertaining for users, and it may be utilized to improve intelligent systems in a variety of ways. This research proposes a attention based image captioning method based upon the encoder decoder architecture. The proposed methodology firstly extract the image features. The image features is passed to the attention layer which applies attention to the different region of images. Later, the decoder layer receives the attention vector and context vector to produce the caption.

ACKNOWLEDGEMENT

I am extremely grateful to my project guide, Prof. Anil Singh Parihar, Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to him for the extensive support and encouragement he provided. I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my project work. They helped me throughout by giving new ideas, providing necessary information and pushing me forward to complete the project work.

AYUSH KUMAR GUPTA
(2K21/CSE/08)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Abbreviations	Full Form
CNN	Convolutional Neural Network
DL	Deep Learning
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
VGG	Visual Geometry Group Network
GoogLeNet	Inception v1
ResNet	Residual Network
RGB	Red, Green, and Blue
R-CNN	Region-based Convolutional Neural Network
NLP	Natural Language Processing
OSCAR	Object-Semantics Aligned Pre-training for Vision-Language Tasks
BERT	Bidirectional Encoder Representations from Transformers
MLP	Multilayer Perceptron
Swin-T	Swin Transformer
FLOP	Floating Point Operations
ResNeXt	Residual Network with Extensive Aggregation
RegNet	Regularized Neural Network
MS-COCO	Microsoft Common Objects in Context
NoCaps	No Capitalization
CC12M	Creative Commons 12 Million
MSRCG	Microsoft Research Concept Graph

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the era of information, a plethora of sources provide access to images, such as the internet, television, and various other mediums. However, human beings possess the remarkable ability to interpret visual stimuli even in the absence of explicit explanations. In contrast, for machines, generating accurate descriptions necessitates not only the identification of objects within an image but also the comprehension of their interrelationships, attributes, and associated activities. Achieving this level of understanding requires a profound contextual understanding of the image for machines. Image captioning involves the generation of a descriptive narrative to accompany an input image. Hence, it can be regarded as a problem encompassing the entire sequence-to-sequence process. Image captioning plays a crucial role in aiding individuals who are blind or visually impaired by providing them with valuable insights into the content of images displayed on websites and other platforms. Furthermore, in the realm of e-commerce applications, image captioning proves to be beneficial for describing the visual content of products and facilitating their organization based on relevant categories. Image captioning holds significant potential for various multi-modal applications, such as cross-modal image retrieval, multi-modal event extraction, video captioning, and visual question answering. These diverse domains can greatly benefit from the capabilities offered by image captioning techniques.

An image caption is generated by leveraging the fusion of computer vision and natural language processing techniques, enabling the creation of a descriptive caption for every incoming image. A lot of research was done in previous years to create an image caption. The process of classifying images and extracting their properties has seen significant advancement thanks to Deep Learning (DL)-based models. When it comes to designs based on convolutional neural networks (CNNs), the widely recognized ImageNet dataset is commonly employed as a benchmark for evaluation. This influential advancement in deep learning has also stimulated research endeavors aimed at modeling temporal data. Presently, numerous studies are underway to enhance our understanding of image semantics and the intricate relationships existing among objects within an image.

Initially, the researcher proposed [3] a methodology based on recurrent neural networks (RNNs) to discern image features and generate corresponding captions. Two pipelines make up the model: one for capturing visual features and the other for language generation. Firstly, the input image is passed through CNN for feature extraction. The RNN is used to generate the titles from the CNN input feature vector instead of the language generation model. Another study [4] employs an LSTM [5] network to generate language because RNNs struggle to grasp dependencies in lengthy sequences, and as a result, this model outperforms others on test datasets. Several studies have been conducted to upgrade and expand over these baseline language generation and CNN models.

1.2 Problem Statement

The challenge of automatically creating semantically relevant and detailed captions for photos requires the development of precise and coherent algorithms. Despite recent improvements in deep learning-based methods, this field still faces a number of difficulties. The management of complex visual sceneries with numerous objects, the handling of ambiguity and context understanding, and the achievement of human-level language creation are some of these difficulties. Furthermore, bias in datasets and generalisation across many image categories continue to be major issues. For image captioning systems to develop their capabilities and improve their useful applications in fields like accessibility, content retrieval, and social media enrichment, these obstacles must be overcome. This project main objective is to create an attention-based system for producing pertinent captions for photographs.

1.3 General Concepts Involved

1.3.1 Convolutional Neural Networks (CNNs)

Humans can outperform in a variety of tasks, like object recognition, image segmentation and classification. CNNs have caused a paradigm change in the field of computer vision. This remarkable achievement has revolutionized the landscape of computer vision research. CNNs is created specifically to process and analyse visual input by taking use of spatial relationships seen in images. The fundamental benefit of CNNs is their automatic learning of representations of images, which is made possible by a series of convolutional, pooling, and non-linear layers. These layers can capture high-level semantic information, such as object shapes and structures, and low-level visual data, such as edges and textures.

1.3.1.1 Input Layer

The input layer of a CNN is in charge of ingesting input data, which is typically a picture or a series of pictures. The input layer of applications that use images accepts the image's pixel values as input. The dimensions of the input layer depend on the size of the input image, which is often given as height, width, and number of channels (for instance, RGB channels). The input layer acts as the picture data's entry point, making it easier to transmit the input image to later layers for the purposes of feature extraction and learning. Pre-processing methods like normalisation and scaling can be used in the input layer to ensure an ideal input data format for later processing steps, which will optimise the data representation within the CNN. These challenges have witnessed the utilization of deep CNN architectures such as AlexNet [6], VGGNet [7], GoogLeNet [8], and ResNet [9], which have exhibited exceptional performance and garnered significant attention within the research community. With the development of video-based CNNs [10] and 3D CNNs [11], which enable powerful video interpretation and action recognition, CNNs have also advanced beyond static picture analysis. Fig 1. shows the abstract architecture of CNN.

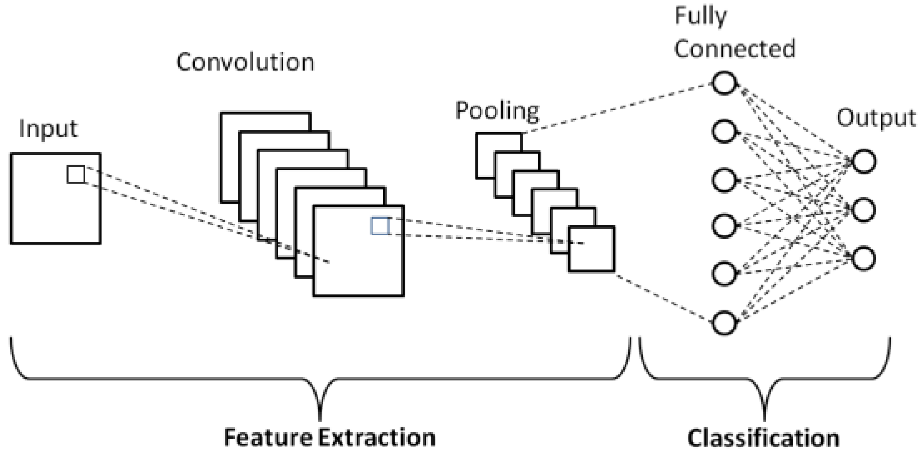


Figure 1. CNN abstract architecture *[45]*

1.3.1.2 Convolutional Layer

A major part of a CNN, for feature extraction convolution layers plays a very important role. This layer is made up of a number of trainable filters, sometimes known as kernels, that convolutionally navigate the input data using a sliding window method. The filters identify local relationships and trends in the input, allowing the network to gather significant characteristics. Convolutional layers make use of common weights to find patterns that are similar across diverse input areas. The extracted features are spatially arranged to from the various features at the convolution layer. The effectiveness of CNNs in a variety of computer vision tasks and applications is greatly enhanced by this hierarchical feature extraction approach.

1.3.1.3 Pooling Layer

This layer follows the convolutional layer in CNNs, is crucial. The task aim of this layer is to down sample feature maps acquired from the prior layer, hence reducing their spatial dimensions. Mostly max pooling is used, this chooses the highest value present in a particular region and discarding all other values. Pooling assists in lowering the network's computational complexity while preserving the most important properties by lowering the spatial resolution. Additionally, by capturing the existence of significant features independent of their specific location in the input, pooling helps achieve translation invariance. Improved robustness, generalisation, and efficiency are results of this down sampling process in CNN designs.

1.3.1.4 Fully Connected Layer

After one or more pooling layers, CNN architecture often contains a fully connected layer, also referred as dense layer. The final layer of the network performs high-level categorization and reasoning using the features that were gathered from earlier layers. Each neuron in the completely connected layer forms connections with every neuron in the layer above, making it easier to analyse the learnt features thoroughly. Leveraging weighted connections and activation functions, this layer assigns the acquired features to specific output classes or predictions. The fully connected layer equips the CNN to reach firm conclusions and generate precise predictions by utilising the hierarchical characteristics obtained from earlier convolutional and pooling layers.

1.3.1.5 Output Layer

In a CNN, output layer is placed after the fully connected layer. This layer utilizes the computations and learned features from the preceding layers to generate the desired output or predictions. It is essential to the network's final phase since it allows CNN to produce the desired output using the data gathered from across the whole network. The output layer's configuration is determined by the current task. For problems involving classification, the output layer typically comprises of neurons corresponding to various classes, with each neuron expressing the likelihood or confidence score that the input belongs to a specific class. Commonly used to obtain normalised class probabilities is SoftMax activation. A single neuron or a number of neurons representing various regression goals may make up the output layer in regression tasks.

1.3.2 Long Short Term Memory

The LSTM architecture, a type of RNNs, successfully captures long-term dependencies in sequential input data while addressing the issue of disappearing gradients. This is achieved through the utilization of memory cells and gating mechanisms. Essential parts like the forget gate, output gate, input gate and memory cell make up an LSTM unit. The following equations regulate each gate in an LSTM unit, establishing its functionality and behaviour:

1. Input Gate (i) :

$$i(t) = \sigma(W_i \cdot [h(t-1), x(t)] + b_i) \tag{1.1}$$

In this equation, the notations are as follows:

- $i(t)$ represents the input gate activation at time step t
- σ represents the input gate activation at time step t
- σ denotes the sigmoid activation function.
- W_i represents the weight matrix associated with the input gate.
- $h(t-1)$ represents the hidden state from the previous time step.
- $x(t)$ represents the input at time step.
- $[h(t-1), x(t)]$ denotes the concatenation of the hidden state and input.

- b_i represents the bias term associated with the input gate.

The calculation of the input gate activation in an LSTM network, determining the extent to which new input data should be allowed to update the memory cell, is governed by the provided equation. To regulate the flow of information, the sigmoid activation function is applied, ensuring that gate activation remains within range of 0 to 1. This mechanism guarantees controlled and precise handling of input data within the LSTM network.

1. Forget Gate (f) :

$$f(t) = \sigma\left(W_f \cdot [h(t-1), x(t)] + b_f\right) \tag{1.2}$$

In this equation, the notations are as follows:

- $f(t)$ represents the forget gate activation at time step t.
- σ denotes the sigmoid activation function.
- W_f represents the weight matrix associated with the forget gate.
- $h(t-1)$ represents the hidden state from the previous time step.
- $x(t)$ represents the input at time step t.
- $[h(t-1), x(t)]$ denotes the concatenation of the hidden state and input.
- b_f represents the bias term associated with the forget gate.

The computation of the forget gate activation in an LSTM network, governing the extent to which the previous memory cell state should be retained or discarded, follows the provided equation. The information flow is regulated by the sigmoid activation function, which guarantees that the gate activation remains within the range of 0 to 1. This mechanism allows precise control over the retention or removal of information from the memory cell state within the LSTM network.

2. Output Gate (o):

$$o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o) \tag{1.3}$$

In this equation, the notations are as follows:

- $o(t)$ represents the forget gate activation at time step t.
- σ denotes the sigmoid activation function.
- W_o represents the weight matrix associated with the forget gate.
- $h(t-1)$ represents the hidden state from the previous time step.
- $x(t)$ represents the input at time step t.
- $[h(t-1), x(t)]$ denotes the concatenation of the hidden state and input.
- b_o represents the bias term associated with the output gate.

The calculation of the output gate activation in an LSTM network, determining the portion of the memory cell state to be revealed as the network's output, is performed using the provided equation. The information flow is regulated by the sigmoid activation function, ensuring that the gate activation remains within the range of 0 to 1. This mechanism facilitates precise control over the exposure of the memory cell state as the final output of the LSTM network.

3. Memory Cell (c):

$$c(t) = f(t) \cdot c(t-1) + i(t) \cdot tanh(W_c \cdot [h(t-1), x(t)] + b_c) \qquad (1.4)$$

In this equation, the notations are as follows:

- $c(t)$ represents the memory cell state at time step t.
- $f(t)$ represents the forget gate activation at time step t.
- $i(t)$ represents the input gate activation at time step t.
- $tanh$ denotes the hyperbolic tangent activation function.
- W_c represents the weight matrix associated with the memory cell.
- $h(t-1)$ represents the hidden state from the previous time step.
- $x(t)$ represents the input at time step t.
- $[h(t-1), x(t)]$ denotes the concatenation of the hidden state and input.
- b_c represents the bias term associated with the output gate.

The updated memory cell state in an LSTM network is determined by this equation. The amount of the prior memory cell state that is kept depends on how the forget gate is activated, whereas the amount of fresh input data that is assimilated depends on how the input gate is activated. The application of the hyperbolic tangent activation function results in the nonlinear transformation of weighted sum of input and hidden state. This transformation empowers the LSTM to discern intricate patterns and features within the data. The LSTM can recognize and capture subtle patterns and complex characteristics by utilizing the hyperbolic tangent activation function, which improves the efficiency of its information analysis and interpretation.

4. Hidden State (h):

$$h(t) = o(t) \cdot tanh\big(c(t)\big) \tag{1.5}$$

In this equation, the notations are as follows:

- $h(t)$ represents the hidden state at time step t.
- $o(t)$ represents the output gate activation at time step t.
- $tanh$ denotes the hyperbolic tangent activation function.
- $c(t)$ represents the memory cell state at time step t.

The LSTM network utilizes encoded data, represented by the hidden state, to identify and retain relevant dependencies and patterns in sequential input data. By employing the memory cell state and output gate activation, this equation calculates the updated hidden state within LSTM network. Memory cell state is subjected to a non-linear change by the hyperbolic tangent activation function, which captures the pertinent data for current time step. Simultaneously, output gate activation governs the extent to which memory cell state is accessible as the network's output. This mechanism ensures effective information processing and controlled output generation within the LSTM network.

1.3.3 Attention Mechanism

When creating captions, a model can use attention mechanism to focus on various regions of image. This accomplished by dynamically evaluating the significance of various image characteristics during the captioning process. The attention mechanism enhances the alignment between visual and textual information, producing captions that are more correct and pertinent to the context by focusing on pertinent image regions. The initial step in the attention mechanism's operation is decoding the image into a series of features. Following that, a probability distribution over every potential region in the image is created using this sequence. The next word in the caption is then generated using the attributes of the region with the highest likelihood. Until the caption is finished or maximum length of caption is targeted or end of token is generated, this process is repeated. Attention mechanism is used to enhance the efficiency of image captioning models has been demonstrated. The attention process can also provide captions that are more elaborative and thorough. The attention mechanism is a formidable method that has the potential to completely change the industry of image captioning. The attention mechanism can assist models in producing more precise, contextually relevant, and descriptive captions by efficiently integrating visual context. The "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" research paper by Xu et al. [1] is widely recognized as a significant milestone in the field of image captioning using attention mechanisms. This paper not only introduced the concept of attention mechanisms in image captioning but also demonstrated their effectiveness in generating more accurate and descriptive captions by focusing on relevant image regions.

CHAPTER 2

LITERATURE REVIEW

In recent times, there has been a notable surge in research interest and activity within the domain of image captioning. This increased attention has led to a proliferation of research projects focused on exploring various aspects of this field. CNNs, among other deep learning architectures, have been extensively studied in the context of their application to the extraction of visual features. The use of RNNs, in particular LSTM networks, for producing captions in picture captioning tasks has also received substantial study attention. Attention techniques have been frequently used to improve the alignment between visual and textual information, enabling models to concentrate on pertinent image regions during caption production. Researchers have also looked into using reinforcement learning and reinforcement learning-based techniques to enhance caption quality. The previous research in the area of image captioning that uses various encoder decoder models is examined in this chapter.

2.1 Related Work

2.1.1 Methods based on templates and retrieval that make use of neural networks

Deep neural networks is used generate captions for images. By incorporating retrieval-based methods, we can address challenges related to embedding and ranking, thereby facilitating the utilization of researchers' proposed deep models for image captioning as a multi-modal input. Socher et al. [12] introduced a dependency tree recursive neural network, which represents phrases or sentences as compositional vectors in order to facilitate description retrieval. The obtained multimodal feature is mapped into a shared space with the aid of a max-margin. Due to innovation and adaptability of new models, deep neural networks ultimately improve performance captioning of picture approaches. The drawbacks of phrases constructed using template-based and retrieval techniques persisted.

2.1.2 Multimodal learning based Image Captioning

The application of methods like retrieval-based and template-based approaches encountered several restrictions in the step of producing sentences for image captioning. No matter how sophisticated the neural networks used to label the photos are, they do not rely on existing descriptions or make any assumptions about sentence organization. These techniques could result in better, more expressive, flexible, and well-structured phrases. The use of multi-model neural networks relies solely on learning to generate captions for images. Fig 2. depicts a typical setup for multimodal learning-based picture captioning techniques.

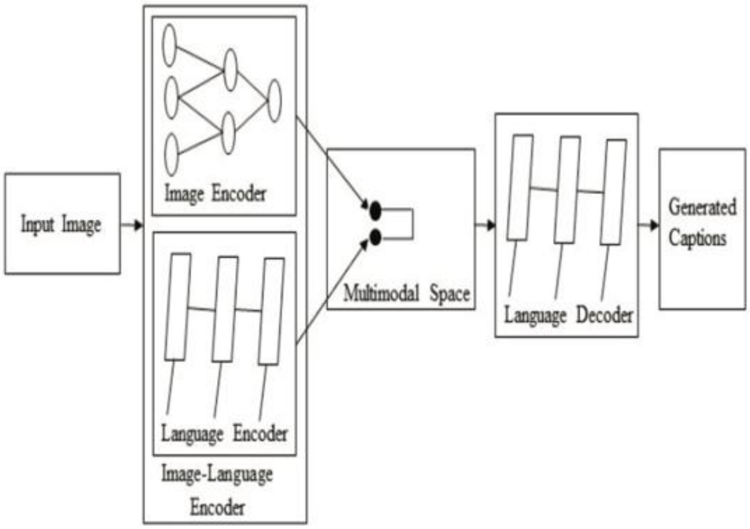


Figure 2. Model abstract architecture of methods based on templates and retrieval that make use of neural networks *[48]*

Kiros et al. [13] present a neural language model designed for generating captions for images, utilizing a log-bilinear language approach. On the other hand, Mao et al. [3] apply an RNN language model in multimodal scenarios to directly estimate the probability of obtaining a word given an image and the previously generated words, resulting in caption generation. Their method uses a deep CNN to extract picture features while modelling distribution of words based on both image data and contextual words using an RNN and a multimodal component. Output, input, and recurrent layers make up the majority of the RNN language model.

2.1.3 Encoder-Decoder Framework based Image Captioning

In their investigation on image captioning, Kiros et al. [13] employed an encoder and decoder framework to describe natural scenes. This approach combines a multi-modal sentence generation model with an image-text embedding model to produce a description for a given query image that is generated step-by-step, much like a language translation. They used LSTM to encrypt textual input in order to encode textual data. To encrypt visual data, a deep CNN was used. The textual data is stored by LSTM hidden states, which are used to extend the embedding space after the visual content has been optimised by a pair-wise ranking loss. A structural content neural language that is conditioned on the feature vector of the backdrop word is used to decode image features in this embedding space. This allows for generation of caption, word by word. The approach was inspired by neural machine translation, where Vinyals et al. [4] used a deep CNN [7] as an encoder for image encoding and an LSTM [14] as a decoder for generating descriptions from image features. Fig. 3. shows the abstract architecture of encoder decoder based models. Typically, the encoder part of the model involves a visual encoder, which processes visual data, while the decoder part contains a language model responsible for generating textual descriptions. However, the specific architecture of the language model and visual encoder can vary depending on the chosen model architectures and approaches.

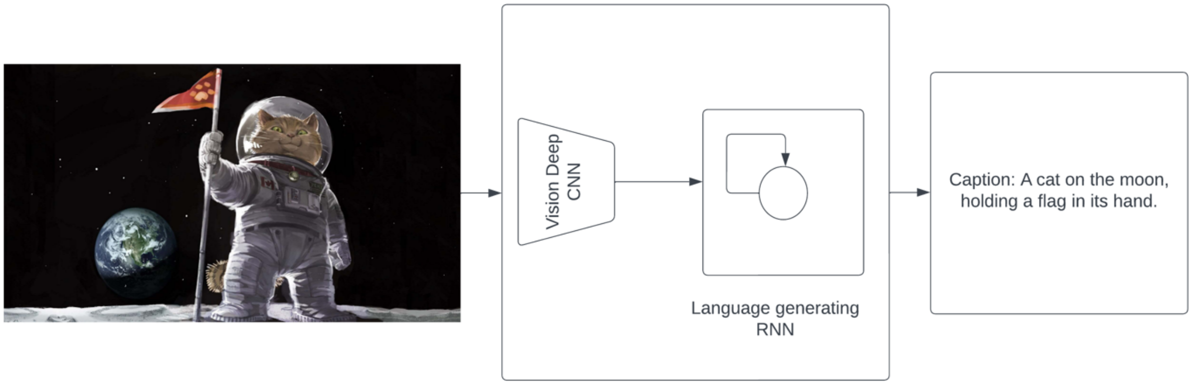


Figure 3. A common approach for picture captioning is the encoder-decoder architecture.

2.1.4 Image Captioning with Attention Mechanism

Attention mechanisms can be incorporated at different stages within image captioning models. To simplify the discussion, we will consider the encoder-decoder model as the fundamental architecture. The following subsections will explore different attention mechanisms and their respective areas of application.

2.1.4.1 Attention Over Grid of CNN Features

In response to the limitations associated with global representations, numerous subsequent methods have opted to enhance the level of detail in visual encoding [15], [1], [16]. Dai et al. [17] exemplify this by substituting 1D global feature vectors with 2D activation maps, effectively integrating spatial structure directly into the language model. A sizable segment picture captioning researchers has also embraced additive attention technique, taking inspiration from the field of machine translation. This approach provides dynamic encoding of visual information over time to image captioning structures, enhancing flexibility and enabling finer-grained control.

In the original proposal presented by Bahdanau et al. [18], referred to as additive attention, attention weights are computed using a single-layer feed-forward neural network incorporating a hyperbolic tangent non-linearity. The following equation is used to determine the additive attention score between h_i and x_j when two sets of vectors, $\{x_1, \dots, x_n\}$ and $\{h_1, \dots, h_m\}$, are provided:

$$f_{\text{att}}(h_i, x_j) = W_3^{\top} \tanh(W_1 h_i + W_2 x_j) \tag{2.1}$$

Where w_1 and w_2 are weights matrices, and w_3 is a weight vector used to carry out a linear combination. Subsequently, a SoftMax function is employed to derive a probability distribution, $P(x_j | h_i)$, which signifies the degree of relevance between the element encoded by h_i and x_j.

Xu et al. [1] introduced a ground-breaking approach that employs additive attention on the spatial output grid of a convolutional layer. By choosing a subset of the properties related to each word, this strategy enables the model to selectively pay attention to specific grid elements.

Specifically, their method retrieves the activation values from the ultimate convolutional layer of a VGG network [19]. The weight allocated to each piece on the grid is then decided via an additive attention method, indicating its proportional importance in creating the next word. To enhance the encoder-decoder framework, Yang et al. [20] introduced a recurrent review network.

By paying close attention to the encoder's hidden states, this network completes a preset amount of review iterations. It produces a "thought vector" after each iteration, which is subsequently used by the decoder's attention mechanism. Chen et al. [21] pioneered the implementation of channel-wise attention on convolutional activations, alongside the traditional application of spatial attention. They further conducted experiments involving multiple convolutional layers to harness the benefits of multi-level features. Likewise, Jiang et al. [22] proposed the utilization of multiple CNNs to leverage the complementary information captured by each network. These networks were subsequently fused together using a recurrent process to enhance the representations they collectively encode. Several research studies have integrated saliency information, which refers to the elements in a scene that capture human attention, into the process of generating captions. This incorporation of stimulus-based attention using saliency information aims to enhance the generation of captions. The concept was initially investigated by Sugano and Bulling [23], who employed human eye fixations as a basis for image captioning. They accomplished this by integrating normalized fixation histograms as an input into the soft-attention module of the existing [1] approach. Additionally, they gave the fixated or unfixated portions of the attended image different weights.

2.1.4.2 Attention Over Visual Regions

In distinction to saliency-based methodologies [23], Anderson et al. [24] introduced an alternative solution that combines both bottom-up and top-down mechanisms for attention. Their method proposed a bottom-up channel using object detection technique which is in charge of producing most probable regions to focus on in an image. Subsequently, these regions undergo a top-down mechanism to determine their respective weights during the word prediction process. To realize the bottom-up pathway, the Faster R-CNN [25] framework is utilized for object detection, generating pooled feature vectors for each proposed region. This allows the model to capture pertinent visual information at a granular level, focusing on regions. The pre-training methodology of this approach is a key component. During the pre-training phase, an auxiliary training loss is incorporated, which involves predicting attribute classes in addition to object classes, utilizing the Visual Genome dataset [26]. The model's ability to learn more reliable feature representations is improved by this additional training objective, which gives the model the ability to focus on a wide range of detections that includes both surrounding regions and prominent objects. As a result, the model becomes proficient in capturing abundant and densely-packed visual information, thereby yielding enhanced performance.

2.1.4.3 Self-Attention Encoding

Self-attention refers to an attention mechanism that establishes connections between every element within a given set. This mechanism facilitates the computation of an improved representation for the set of elements by leveraging residual connections. First introduced by Vaswani et al. [27] as a solution for machine translation and language comprehension tasks, this concept laid the foundation for the emergence of the Transformer architecture and its diverse adaptations. These advancements have had a profound impact in the field of NLP and subsequently extended their influence to the domain of Computer Vision.

To analyse three sets of vectors: a set of n_q query vectors Q, a set of key vectors K, and a set of value vectors V, each set comprising n_k elements, self-attention uses scaled dot-product mechanism, and multiplication-based attention operator. Using similarity distribution formed between the key vectors and query, this operator calculates weighted sum of values of vectors.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

The scaling factor, d_k, is used to scale the dot product in the self-attention mechanism. In self-attention, the linear projections of same set of input items yield query vectors, key vectors, and value vectors. The remarkable achievements of the Transformer architecture serve as evidence that leveraging self-attention leads to superior performance in comparison to attentive RNNs.

Huang et al. [28] proposed an extension to the attention operator that incorporates a gating mechanism guided by the context. This mechanism determines the weighting of the final attended information. Their method concatenates the inquiries with the output of the self-attention action. A gate vector and an information vector are then calculated and multiplied together. This mechanism was implemented in their encoder to augment the visual features. Consequently, subsequent models, including [29], have adopted this approach. Pan et al. [30] proposed the application of bilinear pooling techniques to further enhance the representational capacity of the attended feature output. This method, which produces set of improved region-level and image-level characteristics, distinguishes itself by integrating higher-order interactions to encode properties at the region level. By incorporating the bilinear pooling mechanism, the encoding of region-level features is strengthened through the consideration of pairwise interactions. As a result, both the region-level and image-level representations are significantly improved. Li et al. [31] proposed OSCAR, an architecture inspired by BERT, that leverages object tags as anchor points to enable semantic alignment between images and text. To do this, they pre-trained extensively huge dataset made up of 6.5 million image and text pairs. During the pre-training process, Li et al. [31] employed a masked token loss, akin to BERT's mask language loss, along with a contrastive loss. The contrastive loss was used to differentiate between aligned word-tag-region triples and non-aligned ones. Expanding on the OSCAR framework, Zhang et al. [32] proposed VinVL, which introduced a novel object detector capable of extracting enhanced visual features. Additionally, changes were made to the VinVL model's vision-and-language pre-training objectives. Following a similar approach, Hu et al. [33] further improved the VinVL model by increasing its size and leveraging larger-scale noisy data for pre-training.

2.1.4.4 Attention In Language Models

When it comes to improving language models' performance in captioning images, attention mechanisms are crucial. They enable precise fusion of visual and textual information, leading to improved quality and coherence of the generated captions. The additive attention mechanism, introduced by Xu et al. [1], incorporates the previous hidden state to guide the attention process over the visual features X. A Multilayer Perceptron (MLP) that creates the expected output word is fed a context vector that is computed by the system. Context modelling might use some improvement, Ge et al. [34] utilize a bidirectional LSTM with an auxiliary module. This module estimates the hidden state of the LSTM in reverse direction. The two sentences obtained from the bidirectional LSTM are combined with grid visual data as part of a cross-modal attention mechanism. This combination of textual and visual data makes it possible to produce a finished caption that is thorough and well-written.

Anderson et al. [24] introduced a specialized approach that involves two distinct layers dedicated to visual attention and language modelling. An LSTM is used as the first layer's top-down visual attention model. It contains elements from the mean-pooled image as well as the previously created word and concealed state. The present hidden state computes a probability distribution across the regions of the image via additive attention mechanism. Following first LSTM layer, attended image feature vector sent to second layer of LSTM, where it is combined with hidden state of first layer. The language modelling effort is made easier by the merging process, which creates a probability distribution over the vocabulary. Huang et al. [35] proposed an adaptive attention time mechanism, enabling the decoder to take a variable number of attention steps for each word generated. A confidence network linked with the second-layer of LSTM which determines degree of attention variability. The model can dynamically change the attention mechanism depending on the environment and unique needs of the generated word thanks to this confidence network, which assesses the requirement and duration of attention for each word. This adaptive approach enhances the flexibility and precision of the attention mechanism, ultimately leading to improved language generation capabilities.

CHAPTER 3

PROPOSED METHOD

3.1 THEORY OF THE PROPOSED METHOD

The proposed model in this study is based on the CNN and attention mechanisms. The suggested model makes advantage of the newly put forth CNN technique by Liu et al. [2], which performs significantly better than other methods for classifying images on the ImageNet dataset.

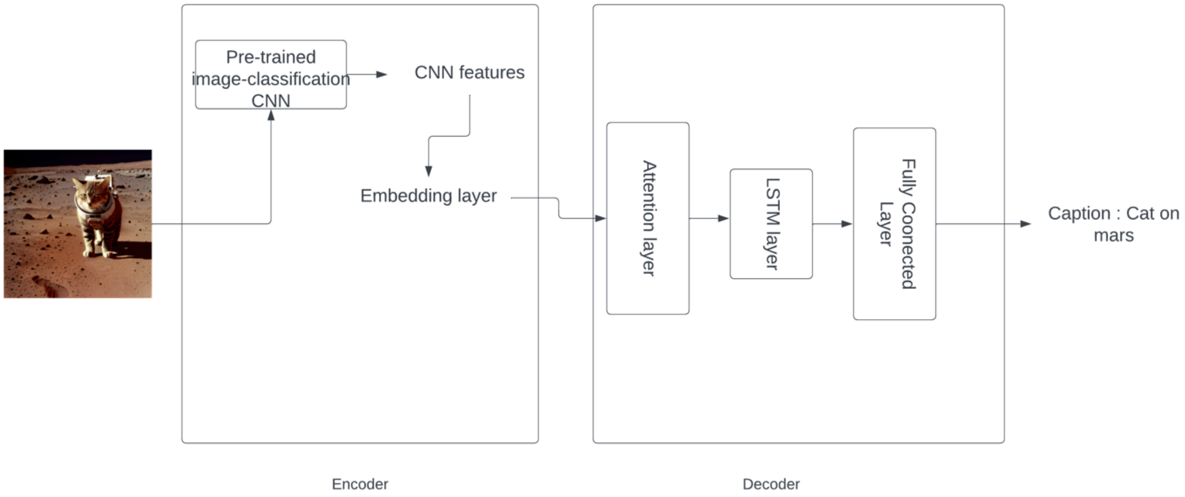


Figure 4. Model abstract architecture

3.1.1 Encoder Block

CNNs are frequently used as the encoder in image captioning, with the task of processing the input image. Its main purpose is to successfully capture the image's visual content by extracting the image's high-level characteristics. Layers of the CNN gradually examine the image at various scales while learning to identify patterns, objects, and their spatial relationships. These characteristics compactly represent the image's relevant visual information. It is the responsibility of the encoder to convert the pixels from input to feature vector, which is subsequently sent to the decoder. The encoder effectively encodes the image, allowing the decoder to provide precise and contextually appropriate captions depending on the visual content. The subsequent subsections will provide a detailed explanation of this specific block.

3.1.1.1 Pre-Trained Image-Classification CNN Block

The fundamental objective of this module is to extract visual attributes from incoming images. Through the utilization of multiple convolutional layers, the CNN block learns to discern significant patterns and shapes within the image. The subsequent layers receive these extracted features for further processing and interpretation. The CNN block can interpret the visual content of an image and extract pertinent information necessary for creating correct captions by acquiring hierarchical representations. The inclusion of convolutional operations within the CNN block significantly enhances the effectiveness and robustness of image captioning models. The final layer of the CNN is removed in this model to extract picture characteristics which is typically a classification layer. The CNN architecture employed is ConvNeXt-Xlarge-384-22k-1k [2], known for delivering superior results in image classification tasks.

3.1.1.1.1 ConvNeXt Model

The authors commence their research by utilizing a standard ResNet model, such as ResNet50, as the baseline architecture. Subsequently, they incrementally enhance the architecture to construct a hierarchical vision Transformer model, such as Swin-T [36]. Fig. 5 illustrates the evolutionary process of ConvNeXt, highlighting the design progression from the initial ResNet model to the final Swin-T model.

Swin Transformers adopt ConvNets as their foundation and employ a multi-stage framework with varying feature map resolutions. There are two key considerations in their design:

1. Stage compute ratio: The proportion of computational resources allocated to each stage.
2. "Stem cell" structure: The initial stage that serves as the foundation for subsequent stages.

The "res4" stage, a resource-intensive component of the original ResNet architecture, was created especially to enable operations like object detection. Swin-T employs a comparable philosophy but adds a slightly altered stage compute ratio of 1:1:3:1. Similarly, ConvNeXt modifies the amount of blocks inside each stage, taking a cue from Swin-T. For instance, the original ResNet-50 configuration of (3, 4, 6, 3) is modified to (3, 3, 9, s3) in ConvNeXt.

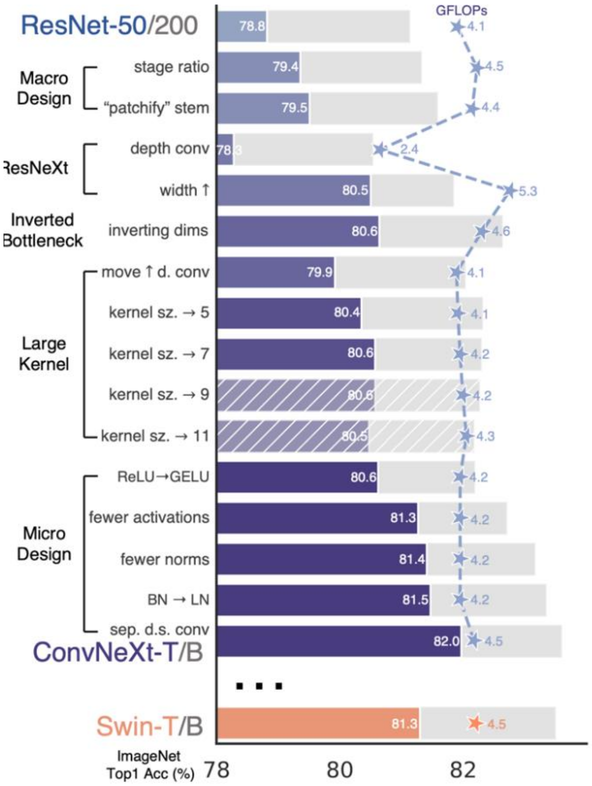


Figure 5. The design Journey of ConvNeXt

Compared to a standard ResNet, ResNeXt offers an improved trade-off between floating point operations (FLOPs) and accuracy. The key feature of ResNeXt is the utilization of grouped convolution, which involves dividing the convolutional filters into distinct groups. In essence, the guiding principle behind ResNeXt [37] is to "increase the number of groups while expanding the width." By employing this approach, ResNeXt achieves enhanced performance by effectively increasing model capacity without a significant increase in computational complexity. This strategy allows for improved representation learning and better utilization of network parameters, resulting in improved accuracy while maintaining a reasonable FLOPs efficiency.

An essential component found in the Transformer block is a phenomenon known as a "inverted bottleneck," which is characterised by a concealed MLP block width that is four times wider than the input dimension. Although this design decision increases the depthwise convolution layer's floating-point operations (FLOPs), it actually lowers the ConvNeXt network's overall FLOPs to 4.6G. This reduction is primarily attributed to a significant decrease in FLOPs achieved by the 1×1 convolution layer within the down sampling residual blocks. Intriguingly, this FLOPs reduction also leads to a marginal improvement in performance, elevating it from 80.5% to 80.6%.

ConvNeXt exhibits remarkable performance when compared to robust ConvNet baselines such as RegNet [38] and EfficientNet [39], particularly in terms of the trade-off between accuracy and computational requirements, as well as inference throughputs on the ImageNet-1K dataset. ConvNeXt also performs better than Swin Transformers without the need for specialised modules like shifting windows or relative position bias. Furthermore, ConvNeXt achieves a higher throughput of 774.7 images/s compared to Swin Transformers' throughput of 757.9 images/s. Despite the prevailing belief that transformers outperform ConvNets on larger scales due to their reduced inductive biases, this notion is challenged by the exceptional accuracy of ConvNeXt-Xl. It achieves an accuracy of 87.8% on the more extensive ImageNet-22K dataset, surpassing the performance of Swin Transformers.

3.1.2 Decoder Block

The extracted features are then sent through the embedding layer after being extracted by the CNN. The embedding layer's primary duty is to give the decoder block the appropriate input.

3.1.2.1 Attention layer

The utilization of the attention layer in this model draws inspiration from the approach introduced by Xu et al. [1]. The attention mechanism gives ability model to choose focus on different sections of the image during caption generation process by simulating the operation of the human visual system. When integrated into the LSTM network, this attention layer facilitates the alignment of the generated words with the corresponding visual content at each time step, enhancing the model's ability to generate contextually relevant captions. Through its ability to attend to diverse regions within an image, the model achieved notable advancements in localization accuracy and the overall quality of generated captions. The subsequent sections will provide a detailed explanation of the various components comprising the attention mechanism.

1. Inputs: The input to model is RGB a channel image, contains three channel i.e. red green and blue.
2. Visual Feature Extraction: A pre-trained CNN, such as ConvNeXt, performs convolutional operations on the input picture. These techniques allow for the effective capture of the image's important information by performing feature extraction.
3. Encoding: The visual features that were obtained undergo a linear transformation, which creates an encoded representation specifically designed to meet the needs of the LSTM network. This stage makes sure that visual data is seamlessly included into the sequential processing architecture of the LSTM.
4. Attention Calculation: During each time step, the attention layer calculates the weights assigned to various regions of the image. In order to do this calculation, the encoded visual attributes are compared to the LSTM network's previous hidden state. The generated attention weights are used to indicate the importance or relevance of various image regions while coming up with the current phrase for the caption.
5. Context Vector: The attention weights play a crucial role in the creation of the context vector, which is formed by combining the encoded visual information's weighted sum. Selective and focused information from the image that is determined to be relevant based on the attention weights is successfully captured by this context vector. The model acquires more contextual knowledge by adding this context vector, which helps it generate the next word in the caption.
6. Integration with LSTM: After being created, the context vector is joined with the preceding word's embedding and supplied as input to the LSTM network. The LSTM processes this input, leading to the generation of the subsequent hidden state. This following concealed state then serves as the foundation for computing the attention weights during the following time step.

7. Caption Generation: The sequential, word-by-word creation of the caption is made easier by the LSTM network. The attention mechanism is utilized by the network to focus on different parts of the image during each stage. This process continues until the model generates the end-of-sentence token or reaches the predetermined maximum length for the caption.

3.1.2.2 LSTM Layer

Detailed explanation about LSTM is already covered in section 1.3.2. The LSTM is trained using a sequence-to-sequence approach, where a feature, denoted as a_t, is sampled from location s_t at time t, and then provided as input to the LSTM for word generation. This procedure is iterated K times, resulting in the generation of an image caption composed of K words.

$$e_{ti} = f_{att}(a_i, h_{t-1}) \tag{3.1}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{K=1}^{L} exp(e_{tk})} \tag{3.2}$$

The model generates the final caption by selecting the word with the highest probability from the vocabulary and adding it to the output caption. This process of caption creation continues until either the end token is generated or the maximum length for the sentence is reached.

CHAPTER 4

DATASETS

The performance of deep learning models must be improved, which calls for a large amount of data. Image captioning datasets encompass images captured from diverse perspectives, with multiple captions associated with each image. To ensure comprehensive model training, it is advantageous to provide multiple ground truth captions. When selecting a dataset for the model, it is crucial to consider various dataset characteristics, including the average caption length and vocabulary size, to ensure appropriate coverage and representation. Researchers often leverage commonly employed datasets as a baseline for benchmarking their models against established standards in the field. These datasets encompass a considerable number of photographs spanning diverse domains, with each image being associated with multiple captions. Table 1 presents an overview of the prevalent datasets utilized in image captioning, along with pertinent details such as the distribution of caption lengths.

1.1 Benchmark Datasets

The MS-COCO Dataset [40] comprises a vast collection of photographs capturing everyday objects in their natural environments, showcasing intricate scenarios from various facets of ordinary life. The dataset encompasses approximately 1.5 million images, with approximately 82,000 images earmarked for training purposes and an additional 40,000 images reserved for validation. Significantly, each image in the training and validation sets is accompanied by five distinct captions written by humans. It is worth mentioning that the test image captions remain inaccessible to the general public. Given its diverse and complex nature, this dataset poses substantial challenges for image captioning tasks.

Flickr30k [41] data set the automatic picture captioning and context-sensitive language interpretation tasks are the focus of this dataset. The dataset contains 158K captions that were contributed by people, together with roughly 31K photographs that were downloaded from the Ficker website. The dataset serves as a standard for sentence-based picture descriptions, and it is frequently used to research the connection between visual media (images) and verbal expressions (image descriptions).

The NoCaps Dataset [42] encompasses a collection of 15,100 photos sourced from the validation and test sets of Open photos. Each photo within the dataset is associated with 11 manually crafted captions. The validation and test groups of the dataset, each comprising 4,500 and 10,600 images, respectively, are the two primary groups. The collection of photos can be categorized into three subgroups: in-domain, near-domain, and out-of-domain images, depending on their similarity to the MS-COCO dataset. The categorization is determined by how closely the photos resemble the dataset. This dataset provides a diverse range of images, allowing for comprehensive evaluation across different domains and levels of similarity to MS-COCO. The 12 million image text pairs in the CC12M Dataset [43] were created specifically for the purpose of pretraining vision and language. The size of this dataset exceeds that of the CC3M cite Sharma 2018 conceptual dataset by nearly 4 times. 20.2 to 16.3 words make up the average caption. This dataset includes JPEG photos with a minimum size of 400 pixels. Word repetition is limited to 0.2 percent at most.

MSRCG [44], a Microsoft Research Concept Graph: This dataset, which has more than 2 million photos with comprehensive semantic annotations and natural language captions, is frequently used for testing and refining image captioning algorithms. Visual Genome dataset [26], which has over 108,000 images with strong semantic and visual annotations, is frequently used for computer vision applications like image captioning.

Table 1. Several benchmark datasets for the visual question-answering job and picture captioning

SNo.	Dataset	Year	Total Samples / Caption per Image
1	MSCOCO	2014	1.5M / 5
2	Flickr30k	2015	30K / 5
3	Flickr8k	2015	8K / 5
4	NoCaps	2019	166K / 11
5	CC12M	2021	12M / 11
6	Visual Genome	2017	2 million / 1 to 10

CHAPTER 5
METHODOLOGY

5.1 Training Metrics

The training of our model is conducted on a Google Colab platform using an Intel Core i7 CPU @3.20 GHz. The experiment is performed using the PyTorch and TensorFlow deep learning frameworks. To process the data, the extracted features are saved in .npy files and subsequently passed to encoder for further analysis. The decoder takes in the encoder output, uses the start token as its input, and initializes the hidden state to zero. This input is processed by the decoder, which also generates predictions and a new hidden state. Afterwards, the model is revised using the decoder's hidden state, and the loss is computed based on the predictions. The teacher-forcing technique is employed to determine the subsequent input for the decoder. Teacher forcing entails feeding the decoder the target word as input. The computation of the gradients, their application to the optimizer, and backpropagation constitute the final phase.

The training loop and the evaluate function are similar, however the evaluate function varies in that it does not use instructor forcing. Contrarily, the decoder utilizes the hidden state, encoder output, and previous predictions as inputs during each time step. The prediction process comes to an end when the model produces the end token. The attention weights for each time step are also saved for future use.

5.2 Performance Metrics

By comparing the generated captions to the actual captions, the Bilingual Evaluation Understudy (BLUE) scores, which measure the effectiveness of attention-based image captioning model, were computed. Quality of machine-generated image captions is frequently assessed using the BLUE score. It was first created for machine translation, but it has been modified for evaluations of image captions. The BLUE score gauges how closely a produced caption resembles reference captions offered by human reviewers. The occurrence of n-grams, which are consecutive sequences of n words, is compared to determine how well the generated and reference captions align. It penalises inconsistencies like excess or missing words and rewards exact matches with higher scores. The resulting score, which ranges from 0 to 1, is a representation of how well the generated and reference captions align. Higher values denote a stronger alignment. The performance and developments of image captioning algorithms can be assessed and tracked by researchers and practitioners using the useful quantitative indicator known as the BLUE score. They can evaluate the calibre of generated captions and make wise decisions to improve their methods in the area of picture captioning by using the BLUE score. The BLUE score formula is show below

$$BLUE = BP \times exp\,(\textstyle\sum_{n=1}^{N} w_n log\, pression_n) \qquad (5.1)$$

Where

- The BP (Brevity Penalty) is a penalty term incorporated into the BLUE score calculation to address the potential discrepancy in length between the obtained caption and the reference captions. It takes into account the difference in length to ensure a fair evaluation and comparison of the captions.
- N represents the maximum order of n-gram considered when calculating the BLUE score.
- w_n represents the weights assigned to each n-gram order.
- $presion_n$ is the precision of the generated caption for the n-gram order.

The model undergoes training for 20 epochs using a batch size of 64. Figure 6 illustrates the loss curve across epochs.

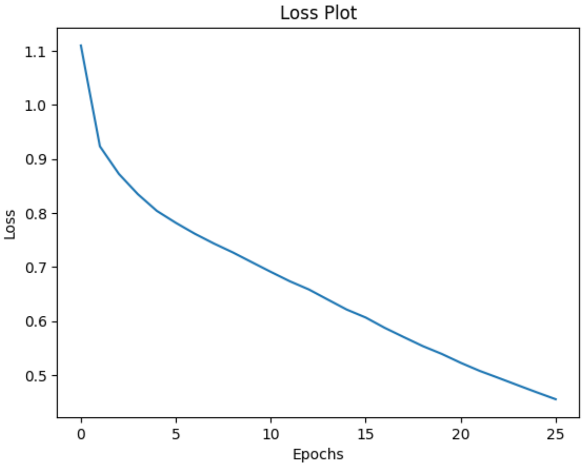
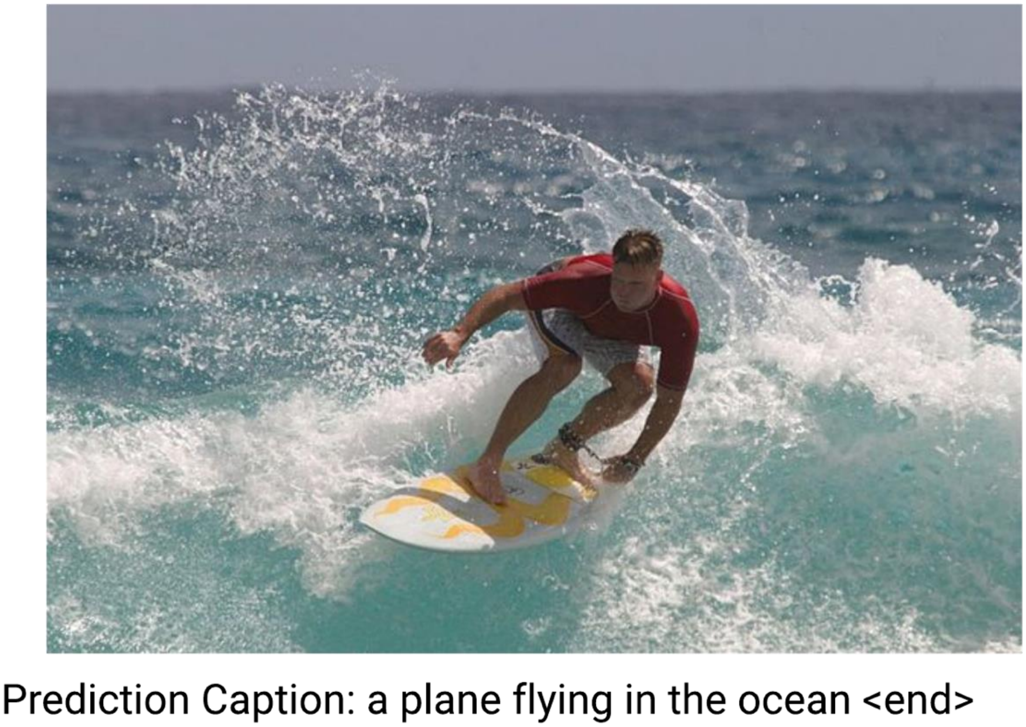


Figure 6. Loss curve with respect to number of epochs

5.3 Qualitative and Quantitative Evaluation

In comparison to other cutting-edge methods in the field, we thoroughly assessed our suggested methodology. The evaluation was conducted using well-known datasets, with an emphasis on the MS-COCO dataset. To evaluate the performance of our model, we utilized the BLUE score as a metric to measure the similarity between the generated captions and the ground truth captions. Due to the high-dimensional nature of the extracted image features from ConvNeXt, truncation becomes necessary. Regrettably, this truncation process has negatively impacted the performance of our model. Additionally, as ConvNeXt is specifically designed for RGB channel images, the inclusion of an extra channel for grayscale images is required as input. Due to these inherent limitations, our model exhibits a relatively lower performance compared to the current state-of-the-art models. The results of the experiment can be observed in Figure 7 and Figure 8.



Prediction Caption: a plane flying in the ocean <end>

Figure 7. Input image for caption generation with generated captions *[46]*

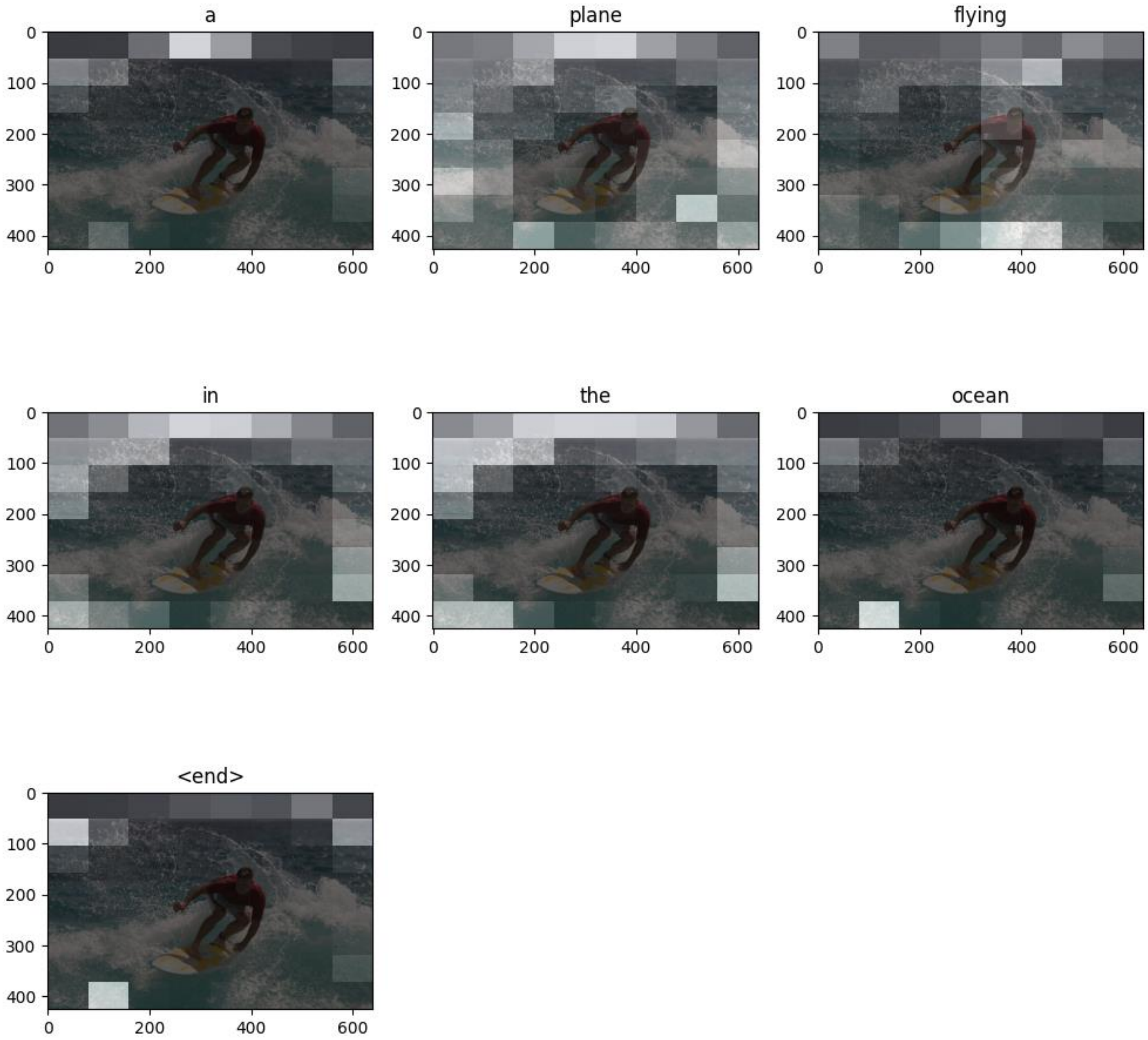


Figure 8. Model caption generation at different region of image

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In the domain of Image captioning, researchers have explored and developed various models, many of which employ the encoder-decoder architecture. In our research project, we have devised an image captioning model that integrates ConvNeXt and an attention mechanism. Our method makes use of ConvNeXt to extract useful information from the input image, allowing the creation of insightful captions. Nevertheless, the extracted features from the ConvNeXt block exhibit a surplus of intricate details, which may hinder the captioning process. To mitigate this challenge, we integrated an attention mechanism into our model. This approach helps the model to focus its attention on the most pertinent areas of the photos, improving the calibre of the captions that are created. We conducted analyses using the Blue score metric on well-known datasets like MS-COCO and Flicker to gauge the effectiveness of our technique. Although our model exhibits potential, it is important to acknowledge its limitations, which hinder its performance when compared to state-of-the-art models. It is worthwhile to investigate alternate architectures for both the encoder and decoder blocks in order to get around these restrictions and boost the model's effectiveness. By carefully considering and implementing these architectural advancements, we can potentially enhance the overall performance of our model.

6.2 Future Work

The potential of this research extends to numerous attention-related possibilities. Attention can also be employed in language modelling. Additionally, a bi-directional attention mechanism can be implemented, calculating attention in both directions: from the image to the caption and from the caption to the image. Moreover, an Attention over Attention approach can be applied while maintaining the proposed encoder architecture. Furthermore, utilizing pre-trained word embeddings from a vast corpus can yield significant benefits. Lastly, training the model on multiple datasets can enhance the generation of more robust captions.

Bibliography

[1] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, PMLR, 2015, pp. 2048-2057.

[2] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[3] J. Mao, W. Xu, Y. Yang, J. Wang and A. L. Yuille, "Explain images with multimodal recurrent neural networks," *arXiv preprint arXiv:1410.1090,* 2014.

[4] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[5] A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks,* no. Springer, pp. 37-45, 2012.

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, pp. 84-90, 2017.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[11] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics,* vol. 2, no. MIT Press, pp. 207-218, 2014.

[13] R. a. S. R. a. Z. R. Kiros, "Multimodal neural language models," in *International conference on machine learning*, 2014.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[15] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7008-7024.

[16] J. Lu, C. Xiong, D. Parikh and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375-383.

[17] B. Dai, D. Ye and D. Lin, "Rethinking the form of latent states in image captioning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 282-298.

[18] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[19] K. Simonyan, "Very deep convolutional networks for large-scale image recognition, arXiv," *Preprint.,* pp. arXiv-1409, 2015.

[20] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen and R. R. Salakhutdinov, "Review networks for caption generation," *Advances in neural information processing systems,* vol. 29, 2016.

[21] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5659-5667.

[22] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu and T. Zhang, "Recurrent fusion network for image captioning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 499-515.

[23] Y. Sugano and A. Bulling, "Seeing with humans: Gaze-assisted neural image captioning," *arXiv preprint arXiv:1608.05203,* 2016.

[24] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[25] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.

[26] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. Shamma, M. Bernstein and F.-F. Li, "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations," *International Journal of Computer Vision},* vol. 123, 2017.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[28] L. Huang, W. Wang, J. Chen and X.-Y. Wei, "Attention on attention for image captioning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[29] F. Liu, X. Ren, X. Wu, S. Ge, W. Fan, Y. Zou and X. Sun, "Prophet attention: Predicting attention with future attention," *Advances in Neural Information Processing Systems,* vol. 33, pp. 1865-1876, 2020.

[30] Y. Pan, T. Yao, Y. Li and T. Mei, "X-linear attention networks for image captioning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[31] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei and a. others., "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Computer Vision--ECCV 2020: 16th European Conference, Glasgow, UK, August 23--28, 2020, Proceedings, Part XXX 16*, 2020.

[32] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[33] X. Hu, Z. Gan, J. Wang, Z. Yang, Z. Liu, Y. Lu and L. Wang, "Scaling up vision-language pre-training for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[34] H. Ge, Z. Yan, K. Zhang, M. Zhao and L. Sun, "Exploring overall contextual information for image captioning in human-like cognitive style," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[35] L. Huang, W. Wang, Y. Xia and J. Chen, "Adaptively aligned image captioning via adaptive attention time," *Advances in neural information processing systems,* vol. 32, 2019.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[37] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[38] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[39] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, 2014.

[41] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[42] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee and P. Anderson, "nocaps: novel object captioning at scale," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[43] S. Changpinyo, P. Sharma, N. Ding and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[44] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *Computer Vision--ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, 2010.

[45] S. Balaji, "Binary Image classifier CNN using TensorFlow," medium.com, [Online]. Available: https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697.

[46] S. a. Hawaii, "File:Surfing in Hawaii.jpg," [Online]. Available: https://commons.wikimedia.org/wiki/File:Surfing_in_Hawaii.jpg.

[47] S. Chen and Q. Zhao, "Boosted attention: Leveraging human attention for image captioning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[48] H. Sharma, M. Agrahari, S. K. Singh, M. Firoj and R. K. Mishra, "Image captioning: a comprehensive survey," in *2020 International Conference on Power Electronics \& IoT Applications in Renewable Energy and its Control (PARC)*, 2020.