Live Image Caption Generator Application

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Anmol Bansal (201274)

Yashvardhan Singh (201457)

Under the guidance & supervision of

Dr. Nishant Sharma

Department of Computer Science & Engineering and Information Technology,

Jaypee University of Information Technology,

Waknaghat, 173234, H.P.

Certificate

This is to certify that the work which is being presented in the project report titled "**Live Image Caption Generator Application**" in partial fulfilment of the requirements for the award of the degree of B.Tech in **Computer Science And Engineering** and submitted to the **Department of Computer Science And Engineering**, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Anmol Bansal (201274) Yashvardhan Singh (201457)." during the period from August 2023 to May 2024 under the supervision of **Dr. Nishant Sharma**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Anmol Bansal 201274

Yashvardhan Singh 201457

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Dr. Nishant Sharma
Designation: Assistant Professor (SG)
Department: Computer Science & Engineering and Information Technology Date:

Candidate's Declaration

I hereby declare that the work presented in this report entitled **'Live Image Caption Generator Application'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Nishant Sharma** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Anmol Bansal
201274

Yashvardhan Singh
201457

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Dr. Nishant Sharma
Designation: Assistant Professor (SG)
Department: Computer Science & Engineering and Information Technology Dated:

Acknowledgment

Firstly, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor Dr. Nishant Sharma, Assistant Professor (SG), Department of CSE, Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of our supervisor in the field of "Deep Learning" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Nishant Sharma, Department of CSE, for their kind help to finish my project.

We would also generously welcome each one of those individuals who have helped us straight forwardly or in a roundabout way in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated our undertaking.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

Anmol Bansal (201274)
Yashvardhan Singh (201457)

TABLE OF CONTENTS

CERTIFICATE i

CANDIDATE'S DECLARATION ii

ACKNOWLEDGEMENT iii

TABLE OF CONTENTS iv-v

LIST OF ABBREVIATIONS vi

LIST OF FIGURES vii

ABSTRACT ix

1 INTRODUCTION 1-6

1.1 Introduction 1-2

1.2 Problem Statement 2-3

1.3 Objectives 3

1.4 Significance and motivation of the project report 3-5

1.5 Organization of project report..... 5-6

2 LITERATURE SURVEY 7-17

2.1 Overview of relevant literature 7

2.1.1 Introduction 7

2.1.2 A summary of the relevant papers 7-15

2.2 Key gaps in the literature 15-17

3 System Development 17-44

3.1 Requirements and Analysis 17-18

3.2 Project Design and Architecture 18-25

3.3 Data Preparation… 25-26

3.4 Implementation 26-42

3.5 Key Challenges 43-44

4 Testing 45-48
4.1 Testing Strategy 45-47
4.2 Test Cases and Outcomes 48

5 Results and Evaluation 49-53
5.1 Results 49-53

6 Conclusions and Future Scope 54
6.1 Conclusion 54

REFERENCES..... 55-56

APPENDIX..... 57

List of figures

S. No.	Title	Page No.
1.	Fig. 3.2.1 Lstm Model Architecture	21
2.	Fig. 3.2.2 Lstm Forget Gate	22
3.	Fig. 3.2.3 Lstm Input Gate	23
4.	Fig. 3.2.4 Lstm Forget & Input Gate	23
5.	Fig. 3.2.5 Lstm Output Gate	24
6.	Fig. 3.2.6 Cnn Model Architecture	24
7.	Fig. 3.2.7 Vgg16 Model Architecture	26
8.	Fig. 3.2.8 Rnn Model Architecture	26
9.	Fig. 3.4.1 Load Data	30
10.	Fig. 3.4.2 Show Data	32
11.	Fig. 3.4.3 Clean the data	33
12.	Fig. 3.4.4 Prepare the data	34
13.	Fig. 3.4.5 Load Vgg16 Model	36
14.	Fig. 3.4.6 Encode Text Data	36
15.	Fig. 3.4.7 Generate Output	37
16.	Fig. 3.4.8 Define Model	38
17.	Fig. 3.4.9 Building Lstm Model	40
18.	Fig. 3.4.10 Fit Model	41
19.	Fig. 3.4.11 Evaluate Model	42
20.	Fig. 3.4.12 Image Generation	44
21.	Fig. 3.4.13 Model Workflow	45
22.	Fig. 3.4.14 Generalized Model	45
23.	Fig. 5.2.1 Frontend Application	54

24.	Fig. 5.2.2 Working	54

Abstract

This paper presents a comprehensive approach to image captioning, leveraging the synergies of deep learning models and mobile-based technologies. The proposed system integrates the powerful VGG16 convolutional neural network (CNN) for image feature extraction with a Long Short-Term Memory (LSTM) model for sequential language generation. The model is trained on the popular Flickr 8K dataset, which comprises diverse images paired with corresponding captions, ensuring robustness and generalization.

The core of the image captioning pipeline involves utilizing the VGG16 model to extract high-level features from input images, followed by feeding these features into an LSTM network to generate coherent and contextually relevant captions. The LSTM's ability to capture long-term dependencies in sequential data enhances the linguistic richness of the generated captions.

Furthermore, the system is equipped with a user-friendly mobile application interface, providing an intuitive platform for users to interact with the live image captioning model. The backend is powered by Flask, a lightweight Python web framework, enabling seamless communication between the user interface and the caption generation system.

The integration of these components results in a versatile and accessible image captioning solution. Users can capture images through the mobile application, triggering the backend processing that utilizes the VGG16-LSTM model to generate descriptive captions in real-time. The generated captions are then presented to the user through the mobile application interface, creating a cohesive end-to-end experience.

Experimental results demonstrate the effectiveness of the proposed system in producing accurate and contextually relevant captions for a diverse range of images. The user-friendly mobile application enhances the accessibility of the image captioning model, making it a valuable tool for various applications, including content indexing, accessibility enhancement, and interactive media experiences.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

The combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) has been revolutionary in computer vision and natural language processing. The image caption generator is an amazing tool developed to fill the void in describing something only seen but not heard.

A huge upsurge in digital imagery has raised the need for smart systems that can “read” and understand pictures. The focus of this project is using the strengths of convolutional neural networks (CNN) and sequence learning via recurrent neural networks (RNN) to create a strong model. This model will not only identify complex patterns in images, but it will also generate an understandable caption that is described in the mobile application live.

CNNs have great power in detecting even finer image elements and hierarchical structure of an image. We integrate CNNs in our model, so that it recognizes important objects, shapes, and even contextual clues leading to total scene description. The first step creates the basis for subsequent description or title.

RNNs can be considered a perfect complementary technology to CNNs as it excels at processing sequential (and contextual) dependencies of input data. In image captioning, RNNs help transform the abstract high-level descriptions captured by CNNs to meaningful and relevant sentences. The temporality property of the RNN allows for a fine-grained modeling of relations among features in an image, ensuring semantically relevant caption generation.

The project is more than just about developing a competent Image Caption Generator. Additionally, as we explore the architecture, training methods, and different evaluation metrics involved, we take a closer look at the ethical implications and possible applications of our project in various domains including making content more accessible for people who are visually impaired or aiding image database indexing.

Upon completion of this report, readers would have acquired in-depth knowledge on the complex interconnection of CNNs and RNNs pertaining to image creation as well as the obstacles, improvements, and ethics in developing the novel innovation.

1.2 PROBLEM STATEMENT

Recently, there is a need for automatic systems that can interpret image information in modern computer vision and artificial intelligence fields. While there are some strides achieved in terms of image recognition, an extremely large gap exists between visual perception and contextual understanding. The project deals with the emerging issue in our society that requires smart creation of an Image Caption Generator based on the combination of CNNs and RNNs for better comprehension of the image content.

However, the main issue is related to how complicated the relationships among different objects in a photo cannot be represented by the traditional image recognition-based system. CNNs are good at feature extraction, but fail to capture the sequential information necessary for producing sensible captions. However, RNNs well-known for processing sequential data encounter problems with understanding of visual spatial relations and multi-layer hierarchy.

There is a realization that when it comes to the complexity of images, a holistic approach presents the best solution. While today's image recognition software can very well detect objects or patterns, it is still unable to provide a contextual description. Despite these abilities, this limitation restricts them to be applied in different areas, for instance, helping the blind to understand the environment or creating indexing large image banks.

Additionally, with the growing number of digital imageries there is no way that people can annotate them manually. An image caption generator facilitates this approach for quick uploading, as well as making this data more available in multiple fields.

The present project shall tackle those problems directly through designing a combined CNN-RNN model which exploits benefits of two neural network models. This would help us improve image recognition systems and expand in areas like inventions, medical assistance technology as well as

content management, taking advantage of the existing gap in the natural languages processing world and computer visions.

1.3 OBJECTIVES

The primary objectives of this project are to:

- Build a robust image caption generator which will combine CNN with RNN.
- Train the model using big image-caption datasets in order to make it generic enough for various visual settings.
- Assess and tweak the model to improve captions with respect to quality, cohesion, and adequacy.
- Investigate cutting-edge approaches for addressing typical issues associated with picture captioning as they relate to rare or novel things and ensuring uniformity across extended captions.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

With the growing dominance of the Information Age and Visual Communication, the importance of the implementation of Hybrid Model CNN-RNN Image Caption Generator is undeniable. This project is motivated by the transformative potential of addressing the following key aspects:

Enhanced Accessibility for the Visually Impaired:

The project aims to contribute to accessibility by providing a tool that can generate descriptive captions for images, aiding individuals with visual impairments in understanding and navigating their surroundings. The technology has the potential to empower visually impaired individuals by providing them with meaningful contextual information from the visual world.

Enriched Content Indexing:

Efficient content indexing is becoming increasingly important with the exponential expansion of digital image repositories. An Image Caption Generator has the potential to assist in automatic labeling of images with rich contextual descriptions that enhance search, retrieval, and organizing visual data across several application areas from e-commerce to medicine.

Human-Machine Interaction and Communication:

An image caption generator is an illustrative tool that gives insight into enhanced human-machine communication based on intelligence. Including visual understanding together with natural languages will better develop connections between people and machines resulting in the development of applications for virtual assistants, smart education, and interactional systems.

Innovation in Assistive Technology:

The project aligns with the scope of innovation in technology. By creating a tool that can understand information and communicate it using language the Image Caption Generator has the potential to open up new possibilities for improving assistive devices. It aims to make visual content more accessible and easier to understand.

Applications in Domains;

The proposed model is versatile and can be used in domains. It can automatically describe content on platforms and improve image accessibility on social media platforms. The Image Caption Generator can also be helpful in content creation allowing users to easily generate captions for their images.

Advancements in Computer Vision and Natural Language Processing;

The integration of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) represents a step in combining computer vision and natural language processing. This project contributes to the evolving landscape of AI by demonstrating the potential of using network architectures to tackle complex tasks pushing boundaries at the intersection of these fields.

Ethical Considerations and Responsible AI:

Considering the growing influence of AI, in our society it becomes crucial to address concerns. This project focuses on examining and reducing biases in image captions aiming to foster the creation of AI systems. By advocating for practices and transparency the project contributes to the implementation of AI technologies. To summarize, this project holds significance as it has the potential to make an impact on society by using advanced technology to improve accessibility, encourage innovation and contribute to the ethical and responsible progression of artificial intelligence.

1.5 ORGANIZATION OF PROJECT REPORT

The project is structured across several chapters, setting out with an introductory overview in Chapter 1. This preliminary chapter encapsulates a concise advent to the subject being counted, highlighting its relevance. It also delves into the articulation of the trouble assertion, the chosen technique, and the overarching goals of the exam.

Moving on to Chapter 2, titled "Literature Survey," the focus shifts closer to an in depth review of previous research inside the area. Emphasis is located on analyzing studies relating Machine Learning techniques, Convolutional Neural Networks, and Recurrent Neural Networks.

Chapter three, detailed as "System Development," meticulously outlines the task's methodology. This encompasses a detailed elucidation of the approaches involved, ranging from information reception to model education and the following checking out phase. Additionally, this bankruptcy undertakes an intensive discussion of the experimental processes hired and unveils the results derived from the task.

The findings are then offered and dissected in Chapter 4, aptly titled "Testing." This bankruptcy not most effective offers the results of the experiment but additionally engages in an in-depth analysis of the consequences. A comparative exploration of various version configurations is blanketed, losing light at the influence of diverse parameters at the accuracy of the models.

Chapter 5, "Results and Evaluation," expands on the assignment's effects with the aid of suggesting that the inclusion of more parameters can increase the precision of water clarity assessment. The

financial disaster underscores the correlation between the form of parameters and the accuracy of water top notch dedication.

Finally, Chapter 6, "Conclusion and Future Scope," serves as the concluding segment. It consolidates the insights garnered during the mission and proposes avenues for future exploration. This chapter additionally serves as the repository for all references, encompassing research studies, actual-time values, algorithms, and different pertinent sources referred to throughout the mission.

CHAPTER 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVANT LITERATURE

2.1.1 Image Captioning Based on Deep Neural Networks [1]

With the recent developments in deep learning, there has been an increasing interest in combining computer vision and natural language processing over the last few years. The category includes image captioning that involves training computers to understand the visible information in one or more sentences. Object recognition and scene understanding complements the ability to assess objects' characteristics, relations, and conditions which are essential for producing high-level picture semantics. However, captioning images remains a difficult and complicated issue in which many researchers have made worthwhile progress. In this work, we primarilyDescribes three deep neural network-based techniques for captioning images: CNN-based models, CNN-based models, and framework-centered reinforcement based. Then, we present the best work from each of these three approaches, explain the criteria for evaluation, and list the main advantages and difficulties.Describes three deep neural network-based techniques for captioning images: CCTV based on CNN, CCTV based on CNN.

2.1.2 Component based comparative analysis of each module in image captioning [2]

Analyzing the results from the sequential models for the picture captioning modules showed that the Bi-directional RNN performed slightly greater than the Vanilla RNN. That was due to the training of skilled and interactively reflective context rich subtitle. A good impact evaluation of the attention shows that the Vanilla-RNN pays attention to the part of the input word that is similar to the word to be predicted hence performance. The second of these search modules determines the degree of relatedness of results that are produced, and is more effective than the greedy search module in this area. In terms of pre-trained glove and keras embedding, when the effect on embedding was evaluated, pre-trained glove performed worse than keras embedding.Feature extraction comparative analysis showed that in regards to features and picture captioning on vgg16,

resnet50 The random uniform seed value clearly manifests the sequential information in the case of the sequential model's seeding techniques.

2.1.3 Deep Residual Learning for Image Recognition [3]

It is more problematic to train deeper neural networks. We present a residual learning approach for training networks which are much deeper than networks used so far. We rather directly reformulate the layers as learning residual functions relative to the inputs of the said layers. This study also provides extensive empirical evidence that these residual networks could be optimised more easily and that a substantial growth in depth enhances accuracy. We present extremely deep residual networks with as many as 152 layers, while keeping model complexity low, on the ImageNet benchmark. The error on the ImageNet test set comes in at just 3.57% for the respective ensemble of these residual nets. In 2015, it was ranked first in the ILSVRC classification. We also offer analysis of CIFAR-10 with layers 100 and 1000.

A lot of visual recognition problems call for depth of the representations. It has been reported from previous exceptional depth representations that led to a demonstration of 28% relativism using the COCO dataset for an object detection. Deep learning Residual Nets on which our ILSVRC submissions are based. In our case, there was a participation in COCO 2015 contests^ {1} where once again we won. The focus is made on detection and segmen-tation tasks for ImageNet localization. Localization, segmentation, detection, and the ImageNet. Foot

2.1.4 Image Caption Generator using CNN-LSTM Model [4]

Generating captions for images has remained a highly demanding issue over the years.Many researchers have attempted this complicated endeavour involving computer vision coupled with natural language processing. The process called image captioning is highly complex and Deep Learning models are able to undertake it. This survey paper aims to provide an exhaustive review of the different image captioning methods utilized so far. We discus on the framework of different models as well as their efficiencies, benefits and drawbacks. Also, this paper has described the common datasets and the performance evaluation measurements for image captioning models.

2.1.5 Automatic image captioning [5]

This paper focused on the issue of automatic image captioning and described new approaches (Corr, Cos, SvdCorr and SvdCos), which are always superior to the state-of-the-art approach EM (+45%) in the accuracy of captions. In particular, they ran extensive experiments in ten different images with a variety of image content styles and evaluated all feasible combinations of the proposed methodologies for enhancing captions' accuracy. A proposed uniqueness weighting scheme on terms and blob-tokens increases the captioning precision. An adaptive blob-tokens generation has always produced better results. Proposed methods show lower degree of bias towards training set and better retrieval precision and recall. These proposed methods can also be applicable elsewhere, for instance, building an image glossary of cell types from figures in medical journals.

2.1.6 Very Deep Convolutional Networks for Large-Scale Image Recognition [6]

We study how the performance quality of a convolutional network changes with respect to its increased depth in case of large-scale image recognition. For our major input we conducted assessment tests of networks in different depth using a two dimensional convolution filter with dimensions 3×3 which proved that the most successful results obtained so far can be improved with the 16,20 weight layers. This information informed our ImageNet Challenge 2014 entry, where we placed at position one for localising and second for classification. We also demonstrate that our representations generalise well to other data sets on which they exhibit the leading performance. As our two best ConvNet models do not violate any copyright, we release them for free public access to further support the ongoing studies in this field.

2.1.7 Show and Tell: A Neural Image Caption Generator [7]

The representation of information relating to an image is one of the main problems within artificial intelligence, which links computer vision with language processing. This paper proposes a generative model with deep recurrent architecture involving modern computer vision and machine translation techniques for creating naturally-sounding sentences depicting images. This model is designed to make sure that when an image is supplied, it outputs a target description sentence with maximal probability. The experiments with several datasets prove the precision of the model and the language fluency, which learns only from "image descriptions". We validate our model both quantitatively and qualitatively because we find that it is usually quite correct. As stated earlier,

while the BLEU-1 score is now at 25 for the state of the art on the Pascal dataset, ours has 59, close to human performance which is around 69. We also yield BLEU-1 point rise in the Flickr30kdataset (from 56 to 66) as well as in the SBU one (from 19 to 28). Finally, we have obtained a BLEU-4 of score of 27.7, which is the existing SOC on the new released COCO dataset.

2.1.8 Automatic Image Captioning [8]

This paper studies the automatic image captioning problem. Provided with a set of captioned images, we would like to identify a relationship between image characteristics and keywords, allowing us to search for relevant keyword phrases for a new picture. We try out many alternative designs involving large datasets from different types of content. Our methods provide up to a 45% accuracy relative improvement over the best existing approach.

2.1.9 Deep Visual-Semantic Alignments for Generating Image Descriptions [9]

An image description model in natural language, for image and regions. We use sets of images with their sentences as a learning framework to understand the connection between the linguistic and visually oriented data. We propose an alignment model combining Convolutional Neural Networks over regions in images, bidirectional Recurrent Neural Networks over sentences, and a structured objective relating a multimodal embedding. Next, we present an end-to-end Multimodal Recurrent Neural Network model that takes advantage of the alignments and learns to create new descriptions of image segments. Our alignment model yields top performance on retrieval exercises over Flickr8K, Flickr30K and MSCOCO. Next, we demonstrate that the produced captions substantially exceed retrieval baselines in two experiments performed on whole images and even a fresh collection of area annotations.

2.1.10 Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [10]

Visual attenuation mechanisms operating from top down are heavily used in image captioning and VQA. Such mechanisms allow for a more in depth analysis which can also involve multistages reasoning, thereby providing enhanced image understanding. Our proposal suggests a mixture of both bottom-up and top-down attention that calculates object or any other major region of the image as attention. Therefore, attention ought to form a part of this natural basis. In our approach, a bottom-up scheme (using Faster-R-CNN) suggests image regions which include assigned feature

vectors; at the same time the top-down mechanism comes up with weights for these features. Using similar technique in an image captioning task, we established a new record on MS COCO's testing server with 117.9, 21.5 and 36.9 scores at CIDEr/ SPICE/BLEU-4 respectively. Using the same procedure for VQA led us to the top on the 2017 VQA Challenge rank.

2.1.11 Image Captioning with Semantic Attention [11]

Automatically generating a natural language description of an image has attracted interest recently both because of its importance in practical applications and because it connects two major artificial intelligence fields: computer vision and natural language understanding. Current methods are either top-down that begin with a general image meaning and turn it to words or bottom-up that create words describing different characteristics of an image and thereafter combine them. This paper presents an alternative approach that involves combining these two approaches with a Semantic Attention Model. Our algorithm teaches us how to choose what to pay attention to through semantic proposal concepts with the aim of constructing hidden states and outputs for RNNs. Top-down, bottom-up is the selection and fusion of feedback loop connecting these computations. We evaluate our algorithm on two public benchmarks: Microsoft COCO and Flickr30K. The experimental results reveal that our algorithm surpasses the state-of-the-art solutions uniformly in terms of various assessment criteria.

2.1.12 Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning [12]

Image captioning is commonly supported using attention-based neural encoder-decoders. For many processes, one will have to maintain an active visual attention to produce a word.

Nevertheless, with respect to non-visual words like "the" or "of", the decoder probably uses minimal if any at all, vision data coming from the image. Other supposedly visual words like "sign" after "behind a red stop" or "phone" coming after "talking on a cell' are usually quite easily predictable on the basis of the language model. We present herein an innovative adaptable attention model that incorporates a visual sentinel. Every time step, our model will decide between attending to the image (region by region) and/or visual sentinel. It specifies whether or not to fixate on the image at a particular place so that useful data can be extracted for serial word output. Our approach

was applied to the COCO image captioning 2015 challenge dataset as well as the Flickr30K. By far, our approach establishes a new state-of-the-art.

2.1.13 Image Captioning with Convolutional Neural Network [13]

Hence, in this research, we outline an image captioning approach particularly about dense captioning. Therefore, we demonstrate some basic technological specifications for a model trying to accomplish such a goal. Specifically, we present a robust architecture for densecap and a neural image caption. We evaluate empirically the products of DenseCap and discuss pitfalls in the model. We demonstrate that 92% of the produced captions match a caption in the training set preserving their correctness and newness. Our proposed criterion substantially minimizes a group of captions referring to an image as yet preserving, the SPICE score of the group.

2.1.14 Attention Is All You Need [14]

Ashish Vaswani et al, from Google Brain and Google Research authored a transformative neural network architecture for NLP and other sequences-to-sequences tasks titled “Attention is all you need”, in 2017. The paper introduced a novel architecture based mainly on attention mechanisms for processing sequential information with parallel computing capability, efficient learning and consideration of long-range correlations in data.

2.1.15 Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Imageto-Sentence Models [15]

With the availability of Flickr30k dataset, sentence-based image description is now considered as a de facto benchmark. The paper introduces Flickr30k Entities dataset, which adds 244k coreference chains connecting references to the same object in each caption of the 158k Flickr30k captions, accompanied by 276k manually labeled bound This is why such notations are important for further advancement of automatic image description and situated language comprehension. These allow us set a new standard in the localization of textual entity mentions in an image. Our paper offers a solid starting point for this venture which involves an image-text embedding, detectors for common items, color classification, as well as the tendency to prefer bigger objects. Our baseline rivals are other advanced state-of-the-arts but it is not easy to translate those gains

into improvements on more complicated tasks like image – sentence retrieval hence underscoring the limitedness of the existing techniques as well as the need for future studies.

2.1.16 Aligning Visual Regions and Textual Concepts for Semantic-Grounded Image Representations [16]

Fine-grain representations of images are vital in vision-and-language grounding problems. Majorly, the present systems include drawings of images with sketches having only visual properties and semantic information. Nonetheless, the fact remains that expressed representations are rarely acceptable because they involve stand alone elements, the linkages of which are mysterious. We want to present an information that is composed of visual regions, each associated with textual concept which carries specific semantic information here. We therefore develop the MIA module that joints together relevant visual attributes and lexical notions through mutual cross- modality alignment. For this purpose we assess the proposed method on the two most representative V&L grounding tasks such as image captioning and visual question answering. For both tasks, the result shows that the use of semantically grounded image features results in better performance compared to the best practice models using all measures. These show that our method works and applies broadly to different types of image application models.

2.1.17 Up-Down: A Spatial Transformer for Bottom-Up Visual Saliency Prediction [17]

To make this model possible of doing spatial transformation, they have used in layered manner a spatial transformer network in combination with a convolutional neural network in order to transform data spatially. We suggested use of a joint STN and LSTM to digitically classify sequences created from the MINST pieces. The proposed LSTM-STN model utilizes a top-down attention mechanism and derives independent elemental processes for the statement from the STN layer in the process of spatial transformation in order to benefit from the LSTM layer without causing any distortion through the simultaneous spatial transformation of the whole This also prevents this distortion from affecting the classification with a Convolutional Neural Network and gives a single digit precision at 1.6% compared with 2.2%.

2.1.18 Knowing What, How and Why: A Near Complete Solution for Aspect-based Sentiment Analysis [18]

This means that we are dealing with different tasks of fine-grained sentence analysis such as aspect extraction, aspect sentiment classification, and opinion extraction in target-based sentiment analysis or aspect-based sentiment analysis (ABSA). Many solvers of the above individualsubtasks or combinations of them exist in the market that can collaborate between themselves to communicate a complete message about the topic, sentiments towards it and reasons for the sentiments. Nevertheless, prior studies conducted on ABSA did not attempt to offer an overall resolution at once. This article presents ASTE, a new ABSA subtask. A solver has particularly to identify triplets (what, how, why) among the inputs showing what the targeted aspects are, how their sentiments polarities aris or simply why they do so (i.e. opinion reasons). As demonstrated by, "Waiters are very friendly and the pasta is just average." 'Waiter' + positive + friendly. This is how we conceive at the two-step model for the given task. In a single model, the first stage predicts what, how and why, and then the second stage couples up the predicted what (how) and why from the first stage to output triplets. Our framework has been a benchmark performer in this novel triplet extraction task in the experiments. However, it beats up some robust baselines derived from recent relevant techniques.

2.1.19 Image Captioning with Semantic-Instance-Aware Attention [19]

Automatically generating a natural language description of an image has attracted interests recently both because of its importance in practical applications and because it connects two major artificial intelligence fields: these are natural language processing and computer vision. Firstly, there exist methods that begin with an overview of an image and break it down into words (topdown approach). Secondly, other techniques entail wording different elements of an image and joining all these sentences together in order to create a complete picture (bottom-up approach). We suggest a novel algorithm that synthesizes the two strategies through a semantic attention model in this article. Our algorithm teaches us how we can choose which semantic concept proposal that is presented to us and incorporate it with hidden states and outputs produced by a Recurrent Neural Network. The choice and blend of events constitute the loop linking upwards and downwards calculation. We evaluate our algorithm on two public benchmarks: Microsoft COCO and

Flickr30K. Our algorithm performs better than all state-of-the-art approaches in all evaluated metrics.

2.1.20 Image Captioning with Transformer [20]

Image captioning represents a highly captivating yet challenging multimodal task that bridges two of the most extensively researched areas in artificial intelligence: vision and language. Image captioning is one of the areas that have grown rapidly in recent years. This project proposes to develop an image captioning model which will generate realistic and natural sentence thatdescribes many situations based on different samples. This work is based on the possibility of usingsuch models to present blind persons with visual images that they are unable to obtain. For instance,it is possible for this technology to be used in building apps that can be integrated into the smartphones whereby such visually impaired individuals will be able to snap an image and later on have a verbal narration about what is being depicted in the photograph. In turn, this makes more necessary developing advanced image caption capabilities for creating efficient and user friendly technologies aimed at helping blind people.ICENSED UNDER THE TERMS OF THE CC BY SA 4.0 This paper proposes a new novel architecture that utilizes transformer as both an encoder and a decoder. Therefore, the model utilizes ViT (vision transformer) to obtain enriched visual representations and combine them with a state-of-art BERT model for language processing.
A unified model achieved an accuracy of 70% on flickr 8k test set and had BLEU scores of 20.

2.2 KEY GAPS IN THE LITERATURE

Although the literature review of image captioning has done much for this field, some noteworthy voids or issues in these papers also exist. Here are some common gaps and open issues:

Limited Generalization: Most existing models perform well on some datasets; however, they do not usually generalize to other or new data. It remains difficult to guarantee consistent excellent performance across diverse domains and datasets.

Handling Rare or Unseen Concepts: This is the major shortcoming for most of the existing models which tend to fail in producing captions for new or peculiar visual patterns. The challenge involves how to employ models without having excessive dependency on training data distribution towards novel concepts.

Fine-Grained Image Understanding: Some models are good at general context but fine-grain analysis needs improvements. The challenge of architecture should be able to capture minute details, relations among objects, and contextual subtleties in images.

Explainability and Interpretability: Several existing models do not have clear explanations of rationale behind their caption generation decisions. This line of research is ongoing, as developing ways to interpret models becomes more advanced, while also trying to understand why some words or notions end up in a caption.

Multimodal Integration: It is a complex process which involves integrating various types of information, that is information gathered through both text or pictures. An important research gap is found in improving the fusion of visual and textual information to yield more consistent and topically related captions.

Evaluation Metrics: Although generic measures such as BLEU and METEOR may be used for evaluating caption qualities and diversities, such parameters could not comprehensively quantify them. An important area of improvement in this regard is the creation of more subtle and complete scoring systems that do a better job in mirroring human judgments.

Data Bias and Fairness: Ongoing issues with ensuring that models do not favor specific demographics or types of images. Fairness, diversity, and bias concerns should be part of research for training data and captioning as well.

Real-Time Captioning: Many existing models concentrate on offline image captioning. Apart from these challenges, real time applications such as the real life video captioning require quick captioning models.

Filling in these areas would result in the production of stronger, transferable, and ethical image captioning processes. However, researchers are still investigating newer ways of improving the performance of captioning models with respect to crucial issues.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

Data Collection:
Requirement: Appropriate training and testing datasets are characterized by a wide array of photos accompanied by captions that can be used in the process.
Analysis: A version is exposed to different contingencies as long as it comprises a complete dataset which enables it to produce meaningful captions in relation to specific events.

Neural Network Architecture:
Requirement: An effective image captioning technology requires an implementation of an integrated CNN-RNN structure.
Analysis: Taking pictures of both spatial and temporal complexity in photographs requires a combination of CNNs to extract characteristics and RNNs to consider sequential context information.

Model Training and Optimization:
Requirement: Effective convergence calls for a strong learning approach that includes gaining transfer knowledge and hyper parameter optimization.
Analysis: Its ability to generalize and supply correct subtitles is enhanced by optimizing the parameter for the version.

Evaluation Metrics:
Requirement: Qualitative assessment of caption quality using measurable metrics including BLEU, METEOR, and CIDEr.
Analysis: Evaluation ensures the competence of providing relevant captions similar to what people believe, which can be used as quantitative measure for all processes.

Ethical Considerations:
Requirement: It is necessary to mitigate biases within facts of schooling data and version outputs to ensure ethical AI deployment.
Analysis: Morally addressing helps build responsible AI ending off propagation of racism and discrimination.

Application Prototyping:
Requirement: Prototype development for real-world applications as well helping for blind and increasing content material indexing.
Analysis: The reasonable software of the model is sensibly validated through prototypes that show the adaptability and feasibility of different domains it can be used upon.

Documentation and Knowledge Transfer:

Requirement: A comprehensive guideline comprising the model design, teaching approaches, and ethics concerns.

Analysis: Proper documenting eases transfer of expertise to another individual who can quickly put the generated image caption into effect as well as enhance its functionality by extending its capacity.

3.2 PROJECT DESIGN AND ARCHITECTURE

- **Long Short Term Memory (LSTM)**

Specifically, long short term memory is one of the most suitable types of recurrent neural network used in sequence prediction. The next word may be predicted by looking at the preceding texts. This one does better than classical RNNs, which are characterized by insufficient long-term memory. Information that is relevant will be carried through the input processing by LSTM. On the other hand, irrelevant data will be thrown away using a forgetting gate. LSTMs are known to handle the issue of vanishing gradients compared to traditional RNNs in order to hold data for a longer duration. By means of back-propagation, LSTMs can maintain this constant error to enable learning across multiple time steps, both forward and backward.

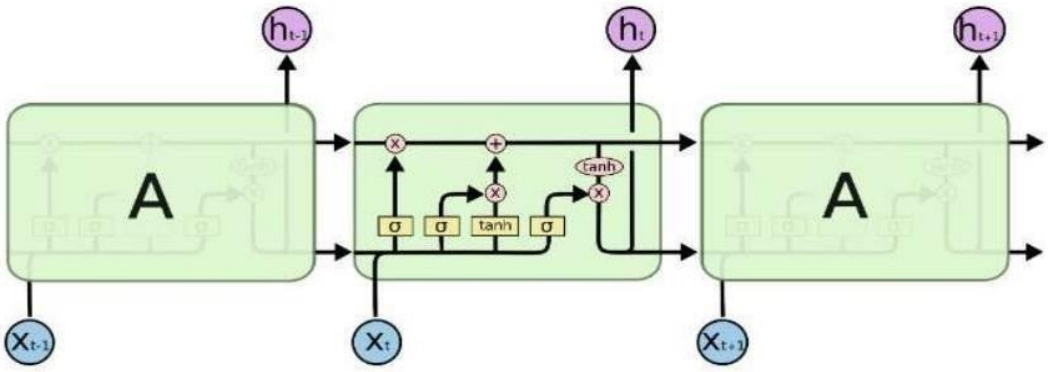
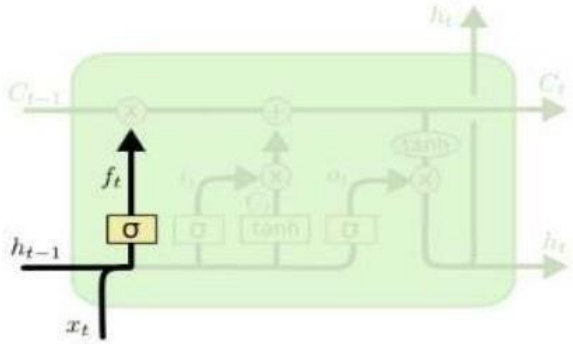


Fig. 3.2.1 Lstm Model Architecture

To deal with storage problem outside of usual flow of RNN, LSTMs use gated cells. These cells will be utilized by the network to modify the information in several ways which include reading as well as depositing data among others. Cells can also process and make up their minds concerning such information, while open and closing gates accordingly. Unlike standard recurrent neural networks, LSTM prevails in such positions because it can remember and store data for a longer period.

However, this chain-based architecture improves the memory capacity of the LSTM such that it can undertake tasks which ordinary RNNs find hard or do not at all. The three main parts of the LSTM include:

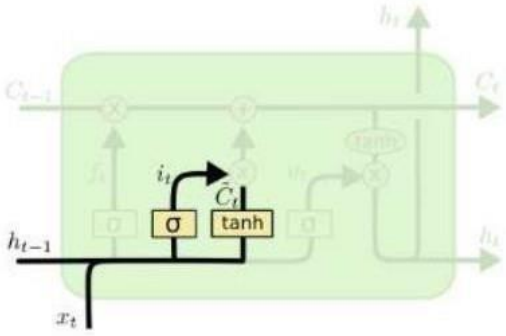
The first thing we do in our LSTM is to forget some things about the cell state. This decision is based on the “forget gate layer”, which is a sigmoid layer. It is an examination of ht-1 and x and returns values of 0 – 1 for each number in the Ct-1 cell state. Totally keep this – 1, Completely discard this-o.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Fig.3.2.2 Lstm Forget Gate

Here, we have to determine which of the new data are to be dumped into the cell state. This has been divided into two parts. A sigmoid layer called “the input gate layer” decides what are the initial values that will be updated. Finally, a tanh layer provides a vector of new candidate values from which one might add to the state. In the subsequent step, we will combine these two in order to derive a state update.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Fig.3.2.3 Lstm Input Gate

The transition from old cell state C t – 1 to current cell state C t is taking place now. Now it is only a matter of doing as we already know what to do thanks to the previous steps. We then multipled the last state by F t and forgot the things that were supposed to be forgotten. Then, we append i*t^c. Here is a fresh group of potential values in proportion to changes made in each state value.

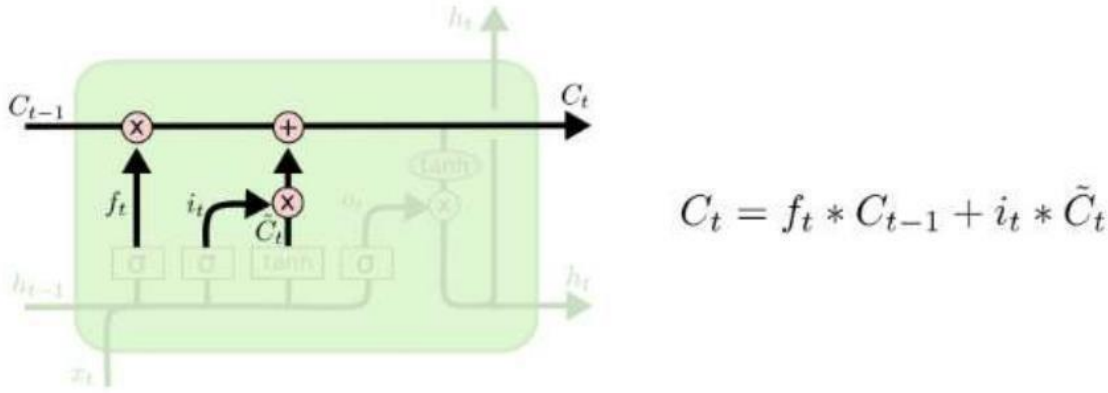


Fig.3.2.4 Lstm Forget & Input Gate

Lastly, we just need to decide on our output. This is an outtake of the condition of our cells, but filtered. Next, we use a sigmoid layer that decides what part of the cell output is going to be selected. Tanh then pushes the values to between minus one and plus one, which are then multiplied with the sigmoid gate's output to output only the sections that we want.

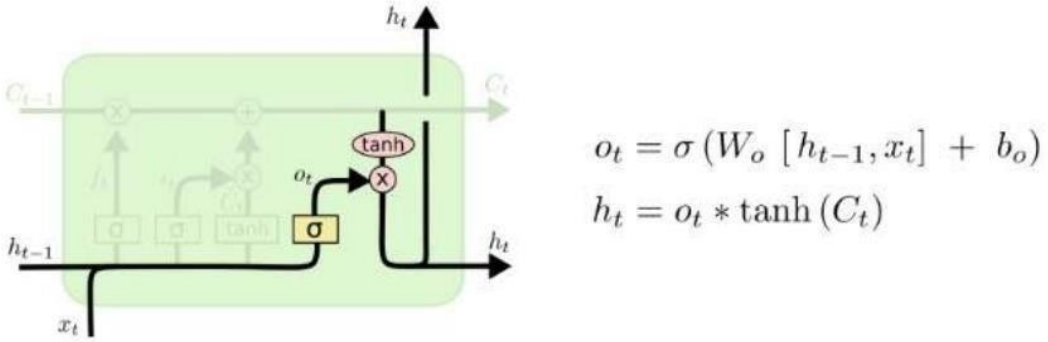


Fig.3.2.5 Lstm Output Gate

- **Convolutional Neural Network (CNN):**

Convolutional Neural Network (CNN) layers for feature extraction on input data are paired with LSTMs to facilitate sequence prediction in the CNN LSTM architecture. Although we will refer to LSTMs that employ a CNN as a front end as "CNN LSTM" in this course, this architecture was initially referred to as a Long-term Recurrent Convolutional Network or LRCN model..

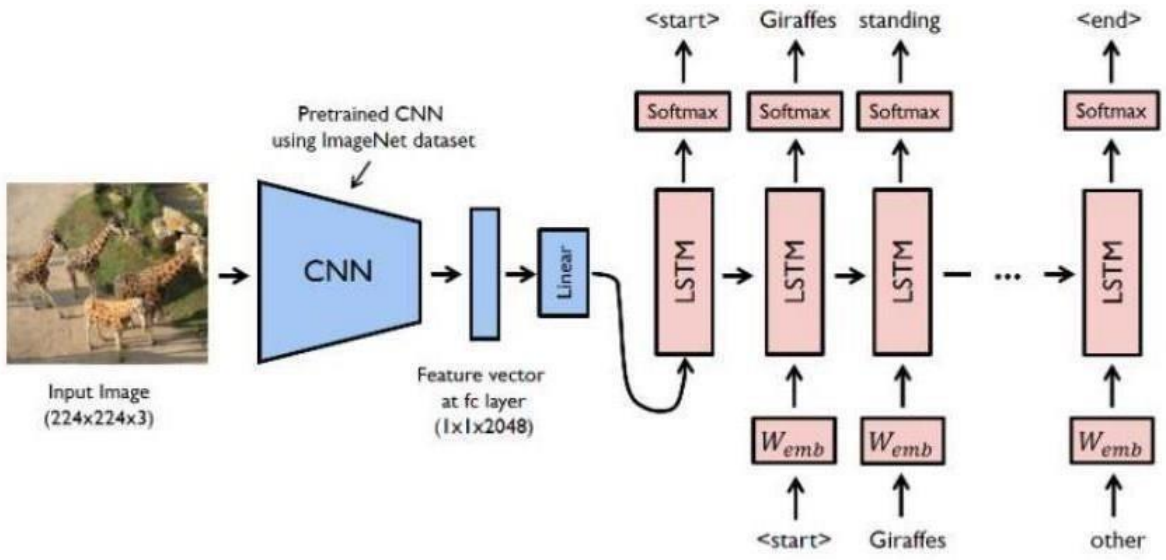


Fig. 3.2.6 Cnn Model Architecture

The task of generating textual descriptions of photographs is handled by this architecture. The employment of a CNN that has been pre-trained on a difficult picture classification assignment and has been repurposed as a feature extractor for the caption producing problem.

VGG16

VGG is one of the most popular deep convolutional neural network architectures used in image classification/object recognition tasks like the Visual Geometry Group (VGG) model, particularly VGG16. VGG16 has been designed by the Visual Geometry Group in the University of Oxford and it became one of the basic models in the area of computer vision because of simple and working principle.

Architecture of VGG16 has a high resoluton and equal structure. There are sixteen weight layers comprising of thirteen convolutional layers and three fully connected layers. Sentence there are sixteen weigh layers consisting of thirteen convolution and three fully connection layers. These convolutional layers are made up of 3×3 filters and the interleaving layers make use of maxpooling to reduce the dimensionality of the spatial nature of the feature maps.

Unlike most other architectures, VGG16 uses small receptive fields for every filter in the whole model. The selective nature of the design makes the model powerful enough in capturing intricate spacial hierarchies in the input images. This uniform architecture featuring a single size filter and a consecutive piling up of layers makes it easy to comprehend why the model is simple.

It consists of five blocks with the first and second blocks having two convolutional layers each then followed by max-pooling. Three other blocks with the same structure can be found next, adding additional filters for each layer of our network. These are the last fully-connected layers that act as a final classifier and produce the final output.

VGG16 has been very successful for image classification tasks and is many times considered an efficient feature extractor for the purposes of transfer learning. Owing to its simple structure, its popularity in education sector has helped students as well as other researchers grasp elementary aspects of deep learning and convolutional neural networks.

VGG16 is effective but has high computing costs and many parameters that may pose difficulties for a resource-starved environment. However, this limitation has made the evolution of other simpler architectures possible with VGG16 being a pioneer model that had a major impact on present modern deep learning methods for computer vision.

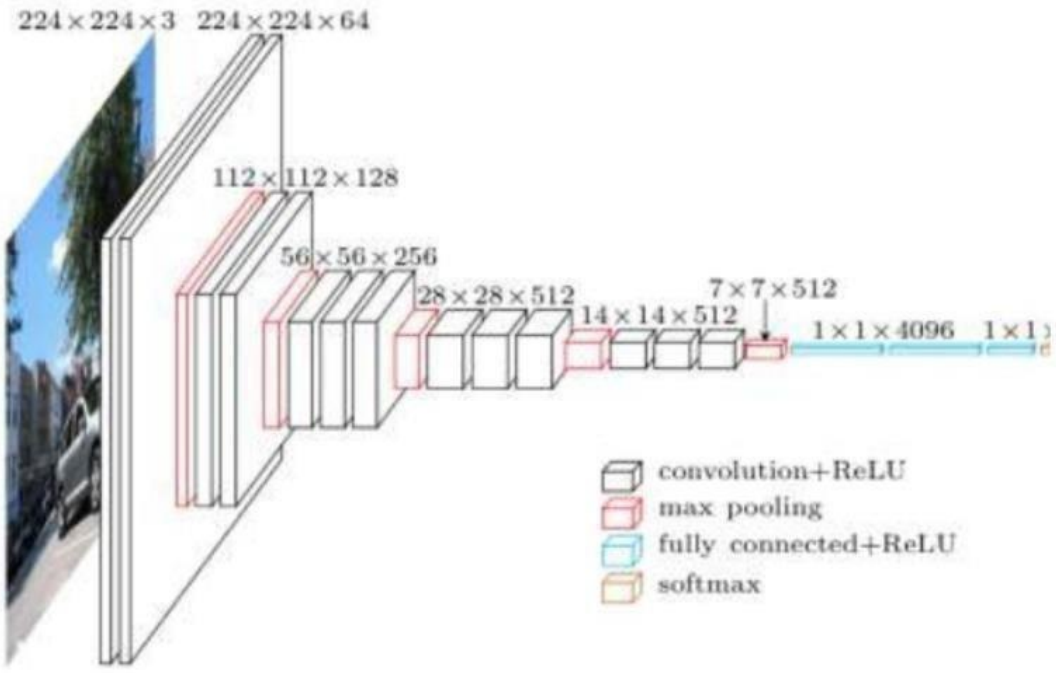


Fig.3.2.7 Vgg16 Model Architecutre

- **Recurrent Neural Network (RNN):**

Afterward, the RNN module captures sequence dependence and local surrounding context from the above-mentioned features. LSTM or GRU cells incorporate in its temporal relationship, leading

to the generation of consistent and meaningful captions that relate to the context by comprehending the order of different visual elements.

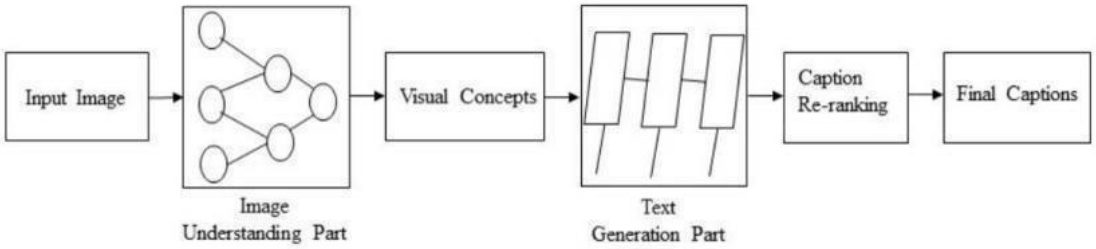


Fig.3.2.8 Rnn Model Architeture

1. **Data Flow:**

   Such a device starts with the reception of input pix that are subjected to function extraction through the CNN. Then, the RNN is used for sequential knowledge on the extracted features. RNN creates the captions mostly with combining the spatio-temporal information, accumulating an extensive overview of the visual data.

2. **Training and Optimization:**

   Under the education segment, the version acquires how to carry out the pulled-out tasks by employing back-propagation and optimization methods. In turn, transfer learning is used for utilizing pre-trained CNN models, thus improving efficiency in education and general performance.

3. **Evaluation Metrics:**

   Various measures are used in evaluating its performance involving BLEU, METEOR and CIDEr. Each metric measures how good a caption is as well as its relevance. In addition, this ensures that the model does not simply learn from the educational data but also generalizes well to the new and unseen images.

4. **Integration of Ethical Considerations:**

   This approach towards the education method integrates ethics including equitable treatment of all participants. It entails using regular validation against ethical reference points to ensure responsible AI adoption.

5. **Frontend Implementation (HTML, CSS, JS):**

Users who access the image caption generator have a user-friendly frontend through which they can interact. CSS systems deal with elements, HTML does the architecture, while JavaScript provides the interface. Through the UI, users can upload photos, which activate the method of caption generation.

6. **Backend Implementation (Flask):**

   The backend, fueled via Flask is for communication between the front end and the middle image processing and caption creating components. The CNN and RNN techniques are activated by Flask routes handling picture up loads. The CNN and the RNN come up with the captions through extraction of functions and incorporating sequential information, respectively. These captions then get passed to the backend and displayed on the frontend.

7. **Communication Flow:**

   Photographs that the users upload reach the backend of Flask from the front end. In this respect, Flask backend uses the skilled CNN to get hierarchical functions directly out of the snapshots. In this regard, extracted features are transferred to RNN for understanding of serial chains and formation of captions. Comprehensive captions are sent back to the frontend for displaying.

The symbiosis, therefore, creates this design and structure that empowers the image caption generator to explore, understand and interpret visual content thoroughly, in its wider context rather than just with reference to popular subject matter only, for a more comprehensive and richer experience of pictures.

3.3 DATA PREPARATION

Building upon the development of our Image Caption Generator, we have devoted much attention into the process of the dataset as it is an extremely critical element for determining the training and the final product. On this endeavor, we appoint Flickr 8k dataset, which involves numerous pictures associated with descriptive captions.

Data Acquisition:

Source: This data set Flickr8k is derived from a photograph sharing website known as flicker. This particular set portrayed many scenes and objects in different situations.
Image-Caption Pairs: In each photo of the dataset are usually two captions that have different words describing just one scene.

1. **Data Cleaning and Preprocessing:**

   Image Preprocessing: In order to make images of compatible sizes, they are pre-processed to conform to a common dimension, allowing for a better fit in a model. Function extraction is improved through techniques like resizing, and normalization.
   Caption Cleaning: After captions are subjected to text preprocessing that involves tokenization, removal of special characters, and uniform formatting of the texts, a coherent and consistent textual corporate is developed.

2. **Splitting the Dataset:**

   Train-Validation-Test Split: Education set, validation set and test set for the dataset. Images together with their captions are assigned on an average split ratio like 70-15-15 for outstanding levels of the upgrade variant.

3. **Augmentation Techniques:**

   Data Augmentation: In order to increase the version robustness, records expansion procedures are performed upon the training set. Moreover, these will include random rotation, flip or slight change of colors for diversification of data and improved generalization.

4. **Integration with CNN-RNN Architecture:**

   Input-Output Mapping: It ensures an efficient, clean enter-output mapping of images with their respective captions. While in the stage of education the model learns how to correlate visual features provided by the CNN network with sequential information produced by the RNN module for optimal caption technology process optimization.

5. **Ethical Considerations:**

   Bias Mitigation: This entails careful consideration of the biases that exist in the dataset in order to ensure fair representation and ethical application. Moral frameworks are frequently used to monitor the dataset and ensure ethical AI applications.

3.4 IMPLEMENTATION

Pre-requisites:

Deep learning, Python, working with Jupyter notebooks, Keras library, Numpy, and Natural language processing are all required for this project. Make sure you have all of the needed libraries installed:

- Keras
- Numpy
- Pandas
- tensorflow
- tqdm
- jupyterlab

Downloaded required dataset:

The Flickr30k Dataset folder with 30091 photos was downloaded as a needed dataset. Flickr 30k Dataset folder including image name and image descriptions.

The steps involved in implementation are:

1. Dataset
2. Load Data
3. Prepare Photo data
4. Prepare Text data
5. Encode Text data
6. Generate output text dataset
7. Define model
8. Fit model
9. Evaluate model
10. Generate caption

1. Dataset

The Flickr 8k dataset. It has 8091 images and 5 captions for each image.Each image has 5 captions because there are different ways to caption an image. I downloaded the data set from flickr website.

2. Load Dataset

```
## The location of the Flickr8K_ images
dir_Flickr_jpg = "Data/Flickr8k_Dataset/"
## The location of the caption file
dir_Flickr_text = "Data/Flickr8k.token.txt"

jpgs = os.listdir(dir_Flickr_jpg)
print("The number of jpg flies in Flicker8k: {}".format(len(jpgs)))
```

```
The number of jpg flies in Flicker8k: 8091
```

Fig.3.4.1 load Dataset

```
from keras.preprocessing.image import load_img, img_to_array
from IPython.display import display
from PIL import Image

npic = 5 # Displaying 5 images from the dataset
npix = 224
target_size = (npix,npix,3)

count = 1
fig = plt.figure(figsize=(10,20))
for jpgfnm in uni_filenames[-5:]:
    filename = dir_Flickr_jpg + '/' + jpgfnm
    captions = list(df_txt["caption"].loc[df_txt["filename"]==jpgfnm].values)
    image_load = load_img(filename, target_size=target_size)

    ax = fig.add_subplot(npic,2,count,xticks=[],yticks=[])
    ax.imshow(image_load)
    count += 1

    ax = fig.add_subplot(npic,2,count)
    plt.axis('off')
    ax.plot()
    ax.set_xlim(0,1)
    ax.set_ylim(0,len(captions))
    for i, caption in enumerate(captions):
        ax.text(0,i,caption,fontsize=20)
    count += 1
plt.show()
```

man on a bicycle riding on only one wheel .
asian man in orange hat is popping a wheelie on his bike .
a man on a bicycle is on only the back wheel .
a man is doing a wheelie on a mountain bike .
a man does a wheelie on his bicycle on the sidewalk .

five people are sitting together in the snow .
five children getting ready to sled .
a group of people sit in the snow overlooking a mountain scene .
a group of people sit atop a snowy mountain .
a group is sitting around a snowy crevasse .

a white crane stands tall as it looks out upon the ocean .
a water bird standing at the ocean 's edge .
a tall bird is standing on the sand beside the ocean .
a large bird stands in the water on the beach .
a grey bird stands majestically on a beach while waves roll in .

woman writing on a pad in room with gold , decorated walls .
the walls are covered in gold and patterns .
a woman standing near a decorated wall writes .
a woman behind a scrolled wall is writing
a person stands near golden walls .

a rock climber practices on a rock climbing wall .
a rock climber in a red shirt .
a person in a red shirt climbing up a rock face covered in assist handles .
a man is rock climbing high in the air .
a man in a pink shirt climbs a rock face

Fig.3.4.2 load Dataset

3. Cleaning captions for further analysis

The caption dataset contains punctuations, singular words and numerical values that need to be cleaned before it is fed to the model because uncleaned dataset will not create good captions for the images.Before feeding the caption dataset into the model for further analysis, it's crucial to preprocess the data to ensure optimal performance. The dataset contains various elements such

as punctuations, singular words, and numerical values that require cleaning. Failing to clean the dataset adequately may result in suboptimal captions being generated for the images. Therefore, a preprocessing step is necessary to remove extraneous elements, standardize text formats, and enhance the overall quality of the captions. This process involves tasks such as removing punctuation marks, handling singular and plural forms of words, as well as handling numerical values appropriately. By cleaning the dataset effectively, we can improve the accuracy and relevance of the captions generated by the model, leading to better results in subsequent analyses

```
import string
text_original = "I ate 1000 apples and a banana. I have python v2.7. It's 2:30 pm. Could you buy me iphone7?"

print(text_original)
print("\nRemove punctuations..")
def remove_punctuation(text_original):
    text_no_punctuation = text_original.translate(str.maketrans('','',string.punctuation))
    return(text_no_punctuation)
text_no_punctuation = remove_punctuation(text_original)
print(text_no_punctuation)


print("\nRemove a single character word..")
def remove_single_character(text):
    text_len_more_than1 = ""
    for word in text.split():
        if len(word) > 1:
            text_len_more_than1 += " " + word
    return(text_len_more_than1)
text_len_more_than1 = remove_single_character(text_no_punctuation)
print(text_len_more_than1)

print("\nRemove words with numeric values..")
def remove_numeric(text,printTF=False):
    text_no_numeric = ""
    for word in text.split():
        isalpha = word.isalpha()
        if printTF:
            print("    {:10} : {:}".format(word,isalpha))
        if isalpha:
            text_no_numeric += " " + word
    return(text_no_numeric)
text_no_numeric = remove_numeric(text_len_more_than1,printTF=True)
print(text_no_numeric)
```

```
def text_clean(text_original):
    text = remove_punctuation(text_original)
    text = remove_single_character(text)
    text = remove_numeric(text)
    return(text)


for i, caption in enumerate(df_txt.caption.values):
    newcaption = text_clean(caption)
    df_txt["caption"].iloc[i] = newcaption
```

Fig.3.4.3 Clean the data

4. Prepare text data

First by loading all the descriptions of the images. Make a dictionary that associates picture names to descriptions. As a preparation for the text data, should work on the description of images. For this, stripped off all the punctuations, changed all the characters' casing to small caps, and ensured that each word had at least one character and words with numbers.

Thereafter, make a dictionary for the individual words in the descriptions. The size of the created vocabulary is 4373.Once the descriptions are cleaned, ensure that the dictionary structure correctly maps each image name to a list of its cleaned descriptions. This step is crucial to maintain the relationship between image names and their respective descriptions. Optionally, save the cleaned descriptions to a file for future use. This allows you to reload the processed data without having to repeat the cleaning steps, which can save time and computational resources in future analysis or model training sessions. By following these steps, you'll have a well-prepared text dataset that is ready for use in your image captioning project

```
from copy import copy
def add_start_end_seq_token(captions):
    caps = []
    for txt in captions:
        txt = 'startseq ' + txt + ' endseq'
        caps.append(txt)
    return(caps)
df_txt0 = copy(df_txt)
df_txt0["caption"] = add_start_end_seq_token(df_txt["caption"])
df_txt0.head(5)
del df_txt
```

```
df_txt0[:5]
```

	filename	index	caption
0	1000268201_693b08cb0e.jpg	0	startseq child in pink dress is climbing up s...
1	1000268201_693b08cb0e.jpg	1	startseq girl going into wooden building endseq
2	1000268201_693b08cb0e.jpg	2	startseq little girl climbing into wooden pla...
3	1000268201_693b08cb0e.jpg	3	startseq little girl climbing the stairs to h...
4	1000268201_693b08cb0e.jpg	4	startseq little girl in pink dress going into...

Fig.3.4.4 Prepare the data

5. Loading VGG16 model and weights to extract features from the images

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Generate the model
model = Sequential()
# Layer 1: Convolutional
model.add(Conv2D(input_shape=(224, 224, 3), filters=64, kernel_size=(3, 3),
                 padding='same', activation='relu'))
# Layer 2: Convolutional
model.add(Conv2D(filters=64, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 3: MaxPooling
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

# Layer 4: Convolutional
model.add(Conv2D(filters=128, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 5: Convolutional
model.add(Conv2D(filters=128, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 6: MaxPooling
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

# Layer 7: Convolutional
model.add(Conv2D(filters=256, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 8: Convolutional
model.add(Conv2D(filters=256, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 9: Convolutional
model.add(Conv2D(filters=256, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 10: MaxPooling
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

# Layer 11: Convolutional
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 12: Convolutional
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 13: Convolutional
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 14: MaxPooling
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

# Layer 15: Convolutional
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 16: Convolutional
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 17: Convolutional
model.add(Conv2D(filters=512, kernel_size=(3,3), padding='same', activation='relu'))
# Layer 18: MaxPooling
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

# Layer 19: Flatten
model.add(Flatten())
# Layer 20: Fully Connected Layer
model.add(Dense(units=4096, activation='relu'))
# Layer 21: Fully Connected Layer
model.add(Dense(units=4096, activation='relu'))
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
...
Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0
```

Fig.3.4.5 Load Vgg16 Model

6. Encode Text Data

```
from keras.preprocessing.text import Tokenizer
## the maximum number of words in dictionary
nb_words = 6000
tokenizer = Tokenizer(nb_words=nb_words)
tokenizer.fit_on_texts(dcaptions)
vocab_size = len(tokenizer.word_index) + 1
print("vocabulary size : {}".format(vocab_size))
dtexts = tokenizer.texts_to_sequences(dcaptions)
print(dtexts[:5])
```

```
vocabulary size : 4476
```

Fig.3.4.6 Encode Text Data

7. Generate output text data

First, let us consider our model's input and output. The model should be supplied with data on inputs as well as outputs to convert this process into a supervised learning activity by training. We will train our model on Each photo of the 4855 will contain a 4-kb feature vector along with one coded label number. We will obtain an output dataset from the image for training and validation captions for the data set of an input text sequence. For example:
The input to our model will be [x1, x2], while you will be the output, with x1 being a 4096 dimensional feature, the model should generate an output text sequence,y, from an input text sequence,x2, and a vector predict.

```
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical

def preprocessing(dtexts,dimages):
    N = len(dtexts)
    print("# captions/images = {}".format(N))

    assert(N==len(dimages)) # using assert to make sure that length of images and captions are always similar
    Xtext, Ximage, ytext = [],[],[]
    for text,image in zip(dtexts,dimages):
        # zip() is used to create a tuple of iteratable items
        for i in range(1,len(text)):
            in_text, out_text = text[:i], text[i]
            in_text = pad_sequences([in_text],maxlen=maxlen).flatten()# using pad sequence to make the length of all captions equal
            out_text = to_categorical(out_text,num_classes = vocab_size) # using to_categorical to


            Xtext.append(in_text)
            Ximage.append(image)
            ytext.append(out_text)

    Xtext  = np.array(Xtext)
    Ximage = np.array(Ximage)
    ytext  = np.array(ytext)
    print(" {} {} {}".format(Xtext.shape,Ximage.shape,ytext.shape))
    return(Xtext,Ximage,ytext)


Xtext_train, Ximage_train, ytext_train = preprocessing(dt_train,di_train)
Xtext_val,   Ximage_val,   ytext_val   = preprocessing(dt_val,di_val)
# pre-processing is not necessary for testing data
#Xtext_test,  Ximage_test,  ytext_test  = preprocessing(dt_test,di_test)
```

Python

```
# captions/images = 4855
 (49631, 30) (49631, 4096) (49631, 4476)
# captions/images = 1618
 (16353, 30) (16353, 4096) (16353, 4476)
```

Fig.3.4.7 Generate Output

8. Define Model

Finally,we use the Keras model to find the closest features for all the images.develop the final model's structure. It will be divided into two parts:

In a case where one has an entry layer, an example would be in regard to the textual input, an entity will consider what the sequencer processor is. Finally, LSTM is going to be the last layer.

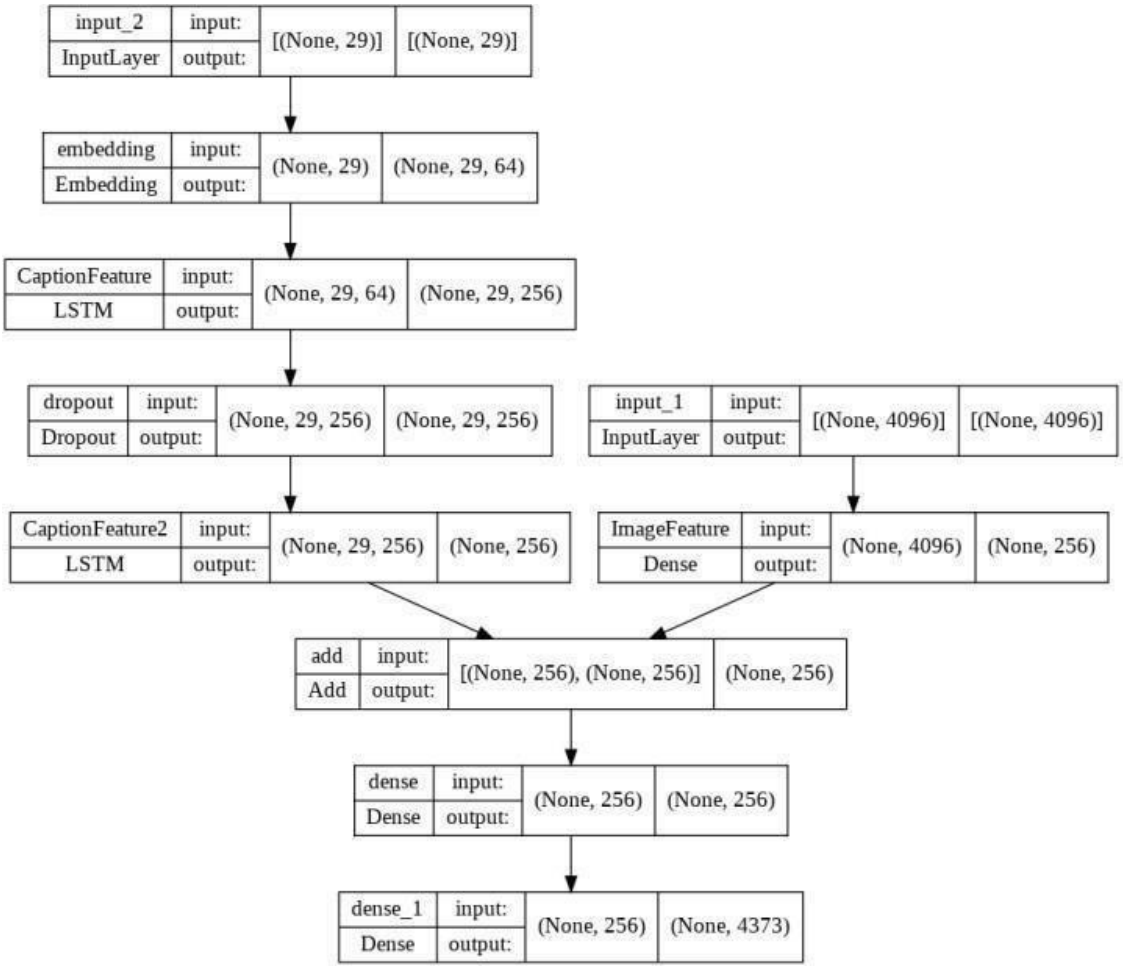


Fig.3.4.8 Define Model

9. Building the LSTM model

```
from keras import layers
from keras.layers import Input, Flatten, Dropout, Activation
from keras.layers.advanced_activations import LeakyReLU, PReLU
print(vocab_size)
## image feature

dim_embedding = 64

input_image = layers.Input(shape=(Ximage_train.shape[1],))
fimage = layers.Dense(256,activation='relu',name="ImageFeature")(input_image)
## sequence model
input_txt = layers.Input(shape=(maxlen,))
ftxt = layers.Embedding(vocab_size,dim_embedding, mask_zero=True)(input_txt)
ftxt = layers.LSTM(256,name="CaptionFeature",return_sequences=True)(ftxt)
#,return_sequences=True
#,activation='relu'
se2 = Dropout(0.04)(ftxt)
ftxt = layers.LSTM(256,name="CaptionFeature2")(se2)
## combined model for decoder
decoder = layers.add([ftxt,fimage])
decoder = layers.Dense(256,activation='relu')(decoder)
output = layers.Dense(vocab_size,activation='softmax')(decoder)
model = models.Model(inputs=[input_image, input_txt],outputs=output)

model.compile(loss='categorical_crossentropy', optimizer='adam')

print(model.summary())
```

Fig.3.4.9 Building Lstm Model

```
4476
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_9 (InputLayer)         (None, 30)                0
_________________________________________________________________
embedding_4 (Embedding)      (None, 30, 64)            286464
_________________________________________________________________
CaptionFeature (LSTM)        (None, 30, 256)           328704
_________________________________________________________________
dropout_4 (Dropout)          (None, 30, 256)           0
_________________________________________________________________
input_8 (InputLayer)         (None, 4096)              0
_________________________________________________________________
CaptionFeature2 (LSTM)       (None, 256)               525312
_________________________________________________________________
ImageFeature (Dense)         (None, 256)               1048832
_________________________________________________________________
add_4 (Add)                  (None, 256)               0
_________________________________________________________________
dense_7 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_8 (Dense)              (None, 4476)              1150332
=================================================================
...
Trainable params: 3,405,436
Non-trainable params: 0
_________________________________________________________________
```

10. Fit Model

```
# fit model
from time import time
from keras.callbacks import TensorBoard
tensorboard = TensorBoard(log_dir="logs/{}".format(time()))
#start = time.time()
hist = model.fit([Ximage_train, Xtext_train], ytext_train,
                 epochs=6, verbose=2,
                 batch_size=32,
                 validation_data=([Ximage_val, Xtext_val], ytext_val),callbacks=[tensorboard])
#end = time.time()
#print("TIME TOOK {:3.2f}MIN".format((end - start )/60))
```

```
Train on 49631 samples, validate on 16353 samples
Epoch 1/6
 - 463s - loss: 5.3759 - val_loss: 4.8776
Epoch 2/6
 - 460s - loss: 4.5156 - val_loss: 4.5810
Epoch 3/6
 - 455s - loss: 4.1009 - val_loss: 4.4676
Epoch 4/6
 - 456s - loss: 3.8300 - val_loss: 4.4382
Epoch 5/6
 - 460s - loss: 3.6239 - val_loss: 4.4181
Epoch 6/6
 - 460s - loss: 3.4465 - val_loss: 4.4253
```

```
for label in ["loss","val_loss"]:
    plt.plot(hist.history[label],label=label)
plt.legend()
plt.xlabel("epochs")
plt.ylabel("loss")
plt.show()
```

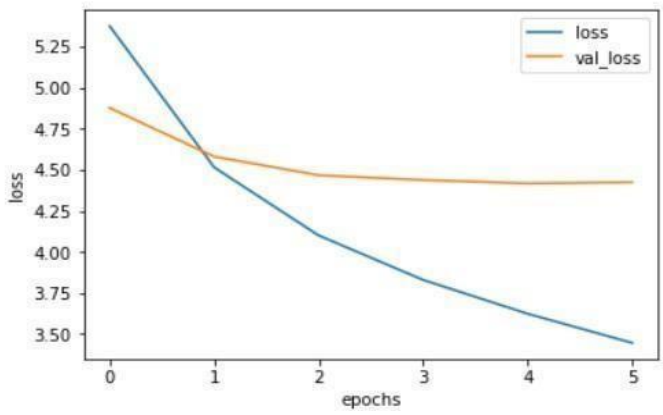


Fig.3.4.10 Fit Model

9. Evaluate Model

Now to evaluate the model, calculate the mean BLEU-scores on the test data set. This requires comparing the original caption with the predicted one.

The BLEU score ranges from 0 to 1.

0 indicates no similarity between two sentences.

1 indicates two sentences are exactly similar.

The mean of BLEU score on test da

```
index_word = dict([(index,word) for word, index in tokenizer.word_index.items()])


nkeep = 5
pred_good, pred_bad, bleus = [], [], []
count = 0
for jpgfnm, image_feature, tokenized_text in zip(fnm_test,di_test,dt_test):
    count += 1
    if count % 200 == 0:
        print("  {:4.2f}% is done..".format(100*count/float(len(fnm_test))))

    caption_true = [ index_word[i] for i in tokenized_text ]
    caption_true = caption_true[1:-1] ## remove startreg, and endreg
    ## captions
    caption = predict_caption(image_feature.reshape(1,len(image_feature)))
    caption = caption.split()
    caption = caption[1:-1]## remove startreg, and endreg

    bleu = sentence_bleu([caption_true],caption)
    bleus.append(bleu)
    if bleu > 0.7 and len(pred_good) < nkeep:
        pred_good.append((bleu,jpgfnm,caption_true,caption))
    elif bleu < 0.3 and len(pred_bad) < nkeep:
        pred_bad.append((bleu,jpgfnm,caption_true,caption))
```

```
12.36% is done..
24.72% is done..
37.08% is done..
49.44% is done..
61.80% is done..
74.17% is done..
86.53% is done..
98.89% is done..
```

```
print("Mean BLEU {:4.3f}".format(np.mean(bleus)))
```

```
Mean BLEU 0.398
```

```
index_word = dict([(index,word) for word, index in tokenizer.word_index.items()])


nkeep = 5
pred_good, pred_bad, bleus = [], [], []
count = 0
for jpgfnm, image_feature, tokenized_text in zip(fnm_test,di_test,dt_test):
    count += 1
    if count % 200 == 0:
        print("  {:4.2f}% is done..".format(100*count/float(len(fnm_test))))

    caption_true = [ index_word[i] for i in tokenized_text ]
    caption_true = caption_true[1:-1] ## remove startreg, and endreg
    ## captions
    caption = predict_caption(image_feature.reshape(1,len(image_feature)))
    caption = caption.split()
    caption = caption[1:-1]## remove startreg, and endreg

    bleu = sentence_bleu([caption_true],caption)
    bleus.append(bleu)
    if bleu > 0.7 and len(pred_good) < nkeep:
        pred_good.append((bleu,jpgfnm,caption_true,caption))
    elif bleu < 0.3 and len(pred_bad) < nkeep:
        pred_bad.append((bleu,jpgfnm,caption_true,caption))
```

```
12.36% is done..
24.72% is done..
37.08% is done..
49.44% is done..
61.80% is done..
74.17% is done..
86.53% is done..
98.89% is done..
```

```
print("Mean BLEU {:4.3f}".format(np.mean(bleus)))
```

```
Mean BLEU 0.398
```

Fig.3.4.11 Evaluate Model

10. Good and bad captions examples from the model:

```
def plot_images(pred_bad):
    def create_str(caption_true):
        strue = ""
        for s in caption_true:
            strue += " " + s
        return(strue)
    npix = 224
    target_size = (npix,npix,3)
    count = 1
    fig = plt.figure(figsize=(10,20))
    npic = len(pred_bad)
    for pb in pred_bad:
        bleu,jpgfnm,caption_true,caption = pb
        ## images
        filename = dir_Flickr_jpg + '/' + jpgfnm
        image_load = load_img(filename, target_size=target_size)
        ax = fig.add_subplot(npic,2,count,xticks=[],yticks=[])
        ax.imshow(image_load)
        count += 1

        caption_true = create_str(caption_true)
        caption = create_str(caption)

        ax = fig.add_subplot(npic,2,count)
        plt.axis('off')
        ax.plot()
        ax.set_xlim(0,1)
        ax.set_ylim(0,1)
        ax.text(0,0.7,"true:" + caption_true,fontsize=20)
        ax.text(0,0.4,"pred:" + caption,fontsize=20)
        ax.text(0,0.1,"BLEU: {}".format(bleu),fontsize=20)
        ax.text(0,0.4,"pred:" + caption,fontsize=20)
        ax.text(0,0.1,"BLEU: {}".format(bleu),fontsize=20)
        count += 1
    plt.show()

print("Bad Caption")
plot_images(pred_bad)
print("Good Caption")
plot_images(pred_good)
```

Bad Caption

true: little girl covered in paint sits in front of painted rainbow with her hands in bowl

pred: group of people are sitting in the street

BLEU: 0.2601300475114445

true: black and white dog is running in grassy garden surrounded by white fence

pred: brown dog is running on the grass

BLEU: 0.1744739429575305

true: collage of one person climbing cliff

pred: man in blue shirt is standing on the air in the air

BLEU: 0

Good Caption

true: black dog and spotted dog are fighting

pred: black and white dog is playing in the grass

BLEU: 0.7598356856515925

true: man drilling hole in the ice

pred: man in blue shirt is jumping on the air

BLEU: 0.7598356856515925

true: man and baby are in yellow kayak on water

pred: man in blue wetsuit is playing in the water

BLEU: 0.7598356856515925

Fig.3.4.12 Image Generation

Model Architecture:

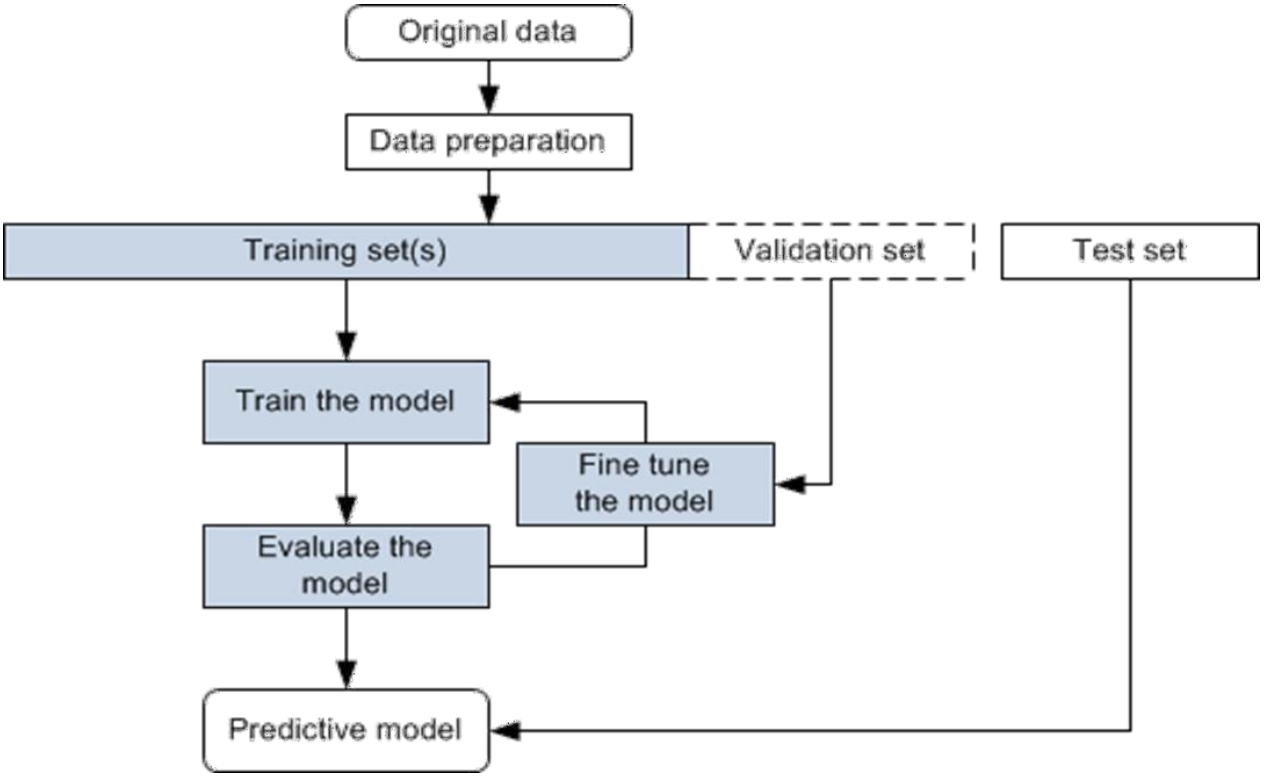


Fig.3.4.13 Model Workflow

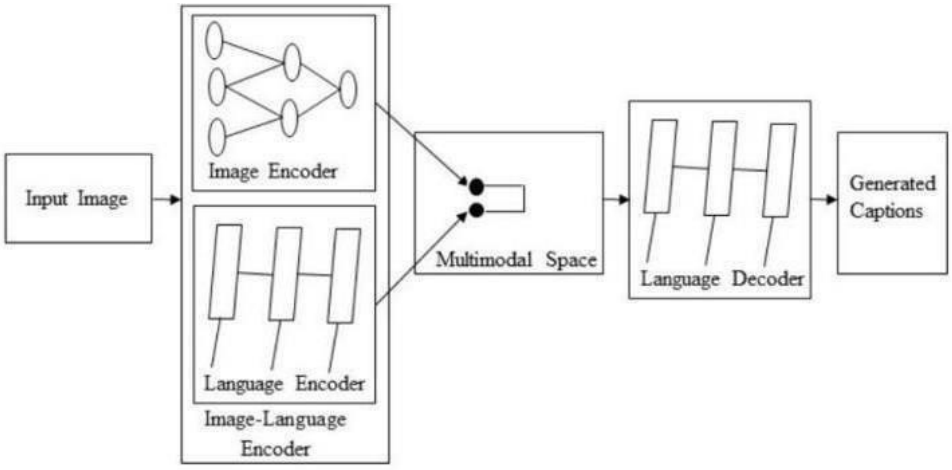


Fig.3.4.14 Generalized Model

3.5 KEY CHALLENGES

Some challenges are associated with the improvement of an Image Caption Generator and with application of a CNN-RNN hybrid version, and they require cautious treatment and creative solutions.

1. **Spatial-Temporal Context Integration:**
   Merging the spatial data acquired through the CNN with temporal information provided with RNN remains a challenging task. The project is aimed at ensuring seamless integration so that it can produce cohesive and topical captions.

2. **Overcoming Data Limitations:**
   Diverse and comprehensive dataset availability is essential for effective education. Overcoming quantity and diversity in schooling data especially in the interest domains presents a serious job that calls for significant records augmentation as well as curation strategies.

3. **Handling Ambiguity and Diversity:**
   Often, images contain elements with distinct shades of doubtful clarity and it is possible to justify multiple captions to a single image. It is only natural to form a version that could manage through the unclarity and produce titles that capture the abundance of existing content.

4. **Mitigating Bias and Ethical Concerns:**
   The use of biased information in education can generate biased versions leading to reinforcement of stereotypes and moral dilemmas. Responsible AI deployment includes continuous identification and minimization of bias in the dataset itself, and ethics somewhere along the improvement chain.

5. **Optimizing Hyperparameters:** Tuning the hyperparameters, evaluating the prices, batch size, and model architecture's parameters, is a delicate balancing task. Optimizing systematic experimentation and optimization to avoid overfitting as well as underfitting in achieving maximum overall performance.

6. **Evaluation Metrics Selection:**
   Identifying proper performance measures for caption excellent is complex. Selecting these measures, such as BLEU, METEOR or CIDEr, needs some care so as to ensure that it captures all aspects of a human context.
7. **Scalability and Efficiency:**

Another challenge in ensuring that the advance version is scalable and green for real time programs. The model should be complex enough but also should not consume excessive computing resources as this will limit large-scale acceptance and real-life usage of it.

8. **Interpretable Model Outputs:**
   Generating an understandable caption version and revealing the inner workings of its decision-making mechanism is hard. This guarantees that the model's outputs are intelligible and conform to human sense as well, which works hand in hand with consideration and usability.

However, this is not easy as it involves an interdisciplinary approach which entails the use of novel techniques like computer vision, computational linguistics, and ethical AI practices. For the proper improvement of a revised Image Caption Generator that surpasses just mere accuracy, overcoming these hurdles is vital.

CHAPTER 4: TESTING

4.1 TESTING STRATEGY

1. **Unit Testing - Component Verification:**
   Objective: Testing the performance of a single additive (CNN or RNN).
   Approach: Carry out the unit test in order to ensure successful extraction of functions by the CNN and passing on the sequential knowledge for RNN. Ensure that every element handles input facts appropriately.

2. **Integration Testing - Subsystem Verification:**
   Objective: Ensure the CNN-RNN architecture merges together well.
   Approach: Integrated version feeds in sample photographs. Ensure proper synchronization between the CNN-extracted spatial features and the RNN-understood temporal context.Y

3. **Data Sanity Checks:**
   Objective: Ensure statistics consistency and integrity.
   Approach: Confirm that the education, validation, and test datasets are properly formatted. Ensure that photo-caption pairs align up preventing misalignments that could undermine the training and analysis processes.

4. **Performance Benchmarking:**
   Objective: Assess model performance against benchmarks.
   Approach: BLEU, METEOR and CIDEr using installed ratings comparison with modeloutput captions toward human captions. Measure benchmarks against what is perceived as on-trend to determine relative positioning.

5. **Robustness Testing - Ambiguity Handling:**
   Objective: Assessing the ability of the model to manage uncertain futures.
   Approach: Use several relevant captions when testing the new version of pix. Determine whether the model creates differently but fits in the sense of context captions, implying adaptiveness.

6. **Bias Testing - Ethical Considerations:**
   Objective: Minimize biases in order to avoid bias in the versioning output.
   Approach: Investigate the ability bias and explore it with various datasets. For example, incorporate procedures such as debiasing or adversarial schooling to counteract bias or bring it down.
7. **Scalability Testing:**

Objective: Assess version performance and scalability.
Approach: Assess the relative performances of the model when working with different dataset sizes. Ensure that scalability is provided for true real-time, monitor aid utilization and aid processing time.

8. **User Acceptance Testing (UAT):**
   Objective: Use cease-customers to validate the version's outputs.
   Approach: Collect comments from customers in relation to generated captions especially those used on domain names. Whether the captions are appropriate based on what users expect, provide meaningful information useful in enhancing the understanding of visual content.

9. **Edge Case Testing:**
   Objective: Overall, assess how the version performs under difficult conditions.
   Approach: Put the version through a test of pictures containing unusual objects, different angles, and specific settings. Make sure its well known flexibility and generalization performance skills.

10. **Interpretability Testing:**
    Objective: Check the interpretability of the outputs in the version.
    Approach: Visualize interests' working in version and see which parts of photo are most important for captioning. Make sure that the produced captions are explainable and correspond with visible cues.

Using an exhaustive testing program involving unit, integration, robustness and ethics tests could help make sure that the Image Caption Generator is strong, objective and context sensitive.

4.2 TEST CASES AND OUTCOMES

1. **Accurate Caption Generation:**
   Outcome: The first but critical expectation is an Image Caption Generator, which in essence has captivating caption for quite many kinds of pictures. This version should demonstrate its ability to translate spatial and temporal functions into meaningful and relevant textual descriptions.

2. **Contextual Understanding:**
   Outcome: In fact, the model will exemplify a sensitive picture interpretation with regard to the connection and the environment within photos manifested in labels that go beyond the object-recognition. This involves photographing various space complexities and time relationships present within image elements.

3. **Performance Improvement:**

Outcome: A model must outperform at least baseline benchmarks showing its effectiveness in comparison to state-of-the-art photo captions models. The quantitative metrics based on BLEU, METEOR, and CIDEr must reflect highly scored captioning performance.

4. **Robustness to Ambiguity:**
   Outcome: The image caption generator, it is likely, will be able to competently handle situations of ambiguity, creating a variety of appropriate but pertinent descriptions of photos portraying more than single true meaning. The version should demonstrate flexibility and ability to sustain during a clear uncertainty.

5. **Bias Mitigation:** Outcome: Trial and moral concerns are intended to help reduce bias in the version's outcomes. This leads us to a predictive outtake as a model which yields captions away from prejudice bias and conforms to moral AI.

6. **Scalability and Efficiency:**
   Outcome: This model will be shown to be scalable and to process images in real time scenarios accurately. Therefore, these resources and processing times should be practical in nature because they have to address usability issues across these packages.

7. **User Satisfaction:**
   Outcome: Objectives for user acceptance checking out to determine the pride of end-users towards the produced captions. Good comments and matching expectations by consumers suggest a successful outcome in accordance with the practical use of the Image Caption Generator.
8. **Adaptability to Edge Cases:**
   Outcome: It (version) must show a capacity to respond to part cases, managing scenarios with unusual views, infrequent devices or specific circumstances. The generalization abilities of the latter are evidenced by its strong overall running in several conditions.

9. **Interpretability of Outputs:**
   Outcome: The outputs of this model must also be comprehensible and have strong observable indicators for the significant features creating the labels or captions. It is also important for visualizations of an attention mechanism to be able to reveal how a decision was made through an explanation of versions it took part in.

10. **Documentation and Knowledge Transfer:**
    Outcome: Proper documentation ensures that the outcome is always a well-informed switch. Outcomes such as predicted results for modelling deployments, usage, or ability

extensions are clear hints that allow for easy adoption with possible improvements by stakeholders.

Thus, all the predicted results can be called the culmination of Success, Improvement and Deployment of a high-tech image caption generator to both laptop vision and natural language processing while following ethical considerations and customers' intentions.

CHAPTER 5: RESULTS & EVALUATION

5.1 RESULTS

1. **Quantitative Metrics:**
   BLEU, METEOR, CIDEr Scores: Assess how this version runs and apply modern-day statistics techniques in creating a caption for an image. Rankings point towards increased degree of harmony with human-produced captions.

2. **Caption Quality:**
   Coherence and Relevance: Evaluate the captions to ensure coherence with that which is apparent or visible. The captions should sound good in respect of contextuality, and match up well with the photos for a better right model.

3. **Ambiguity Handling:**
   Diverse Outputs: Assess its possibility of providing diverse yet contextually appropriate captions for pics that have multiple plausible readings. This shows the flexibility of the mode in solving complex problems with unclear parameters.

4. **Bias Mitigation:**
   Fair and Unbiased Outputs: Ensure that the captions are free of stereotypical or discriminatory bias in the output data of the model. Ethical issues need to yield in true and unbiased captions.

5. **Scalability:**
   Real-time Processing: Compare the efficacy of the version in real time processing and gauge its scalability. They must fit within physical resource consumption as well as processing instances appropriate.

6. **User Feedback:**
   User Acceptance Testing (UAT): Get feedback from end customers regarding how proud of them they feel about the resulting captions. The version is true, and it is in accordance with anticipated expectations; positive consumer response demonstrates this.

7. **Adaptability to Edge Cases:**
   Handling Unconventional Scenarios: Assess how well the model handles atypical examples, for example, images that portray unusual objects or unusual perspectives.
   Generalization capabilities can be implied from robust performance in multiple situations.
8. **Interpretability:**

Attention Mechanism Visualization: To comprehend what part of a picture contributes most to interpretation period and visualize interest mechanism. This improves the version's readability and provides an indication of what the decision-making mechanism entails.

9. **Comparative Analysis:** Benchmark Against Existing Models: Finally, compare the model's performance with existing day photograph captioning models. This will roll the improved version into the context of the changes with the subject matter.

10. **Documentation:**

    Comprehensive Documentation: The above-mentioned points imply that if a model has been effectively documented, it should provide unambiguous insights into not only the design of the model but also the educational methods and ethical considerations used by teachers while executing a model in their classrooms. It should therefore support knowledge transferring and help customers with installation, use as well as possible extension of the version.

The effects of this model should be consistent with the set objectives showing improved ethicalness, usefulness in different programs, and the ability to improve image captioning process. Updates, testing, and fine tuning might make a difference since the first version was developed in light of comments and changing requirements.

5.2 EVALUATION

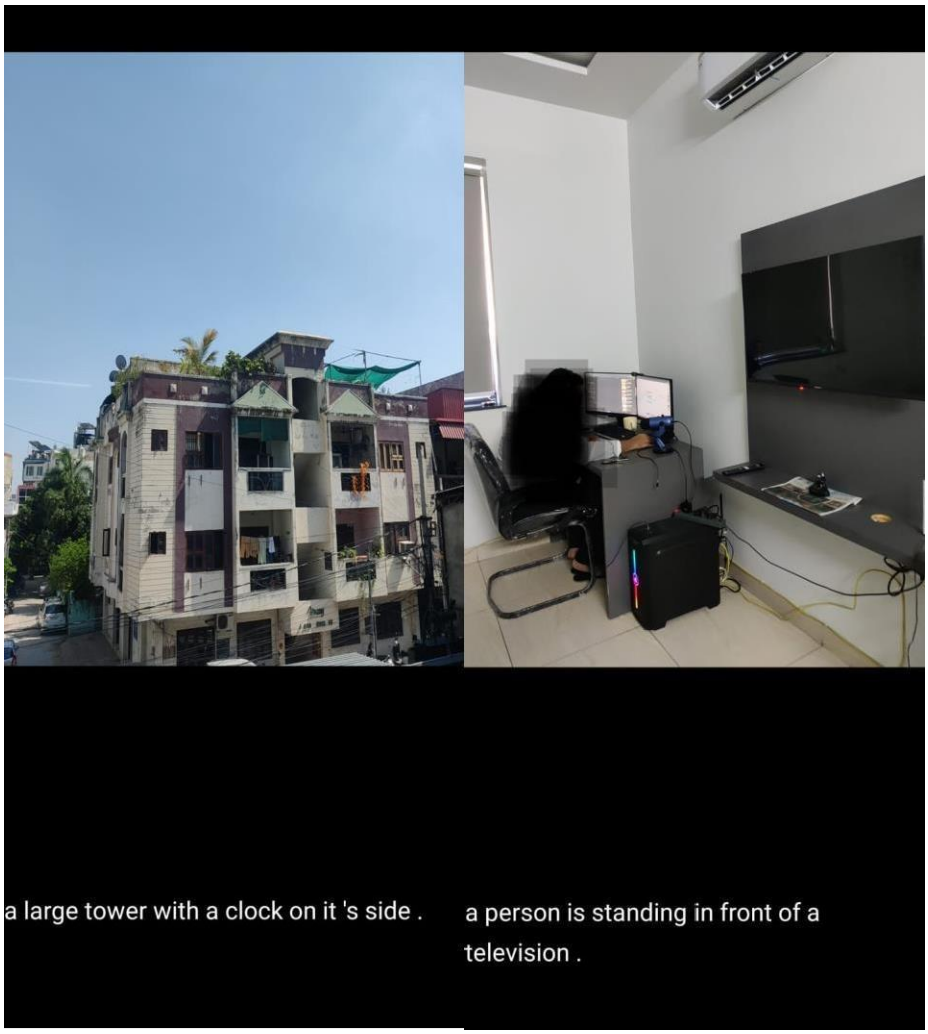
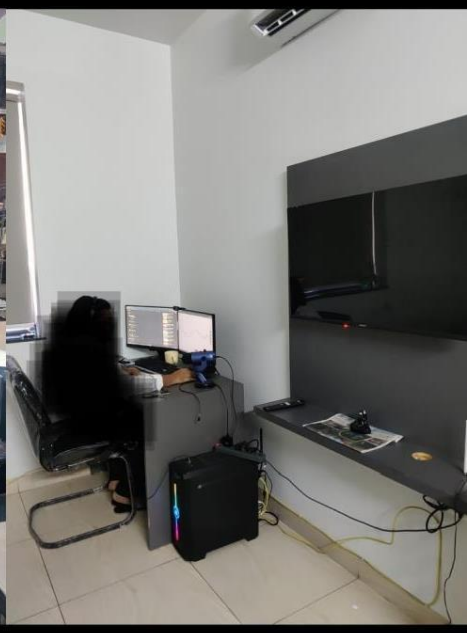


Fig.5.2.1 Frontend Appication

a street sign on a pole on a city street .



a small room with a bed , desk , lamp , laptop and window .

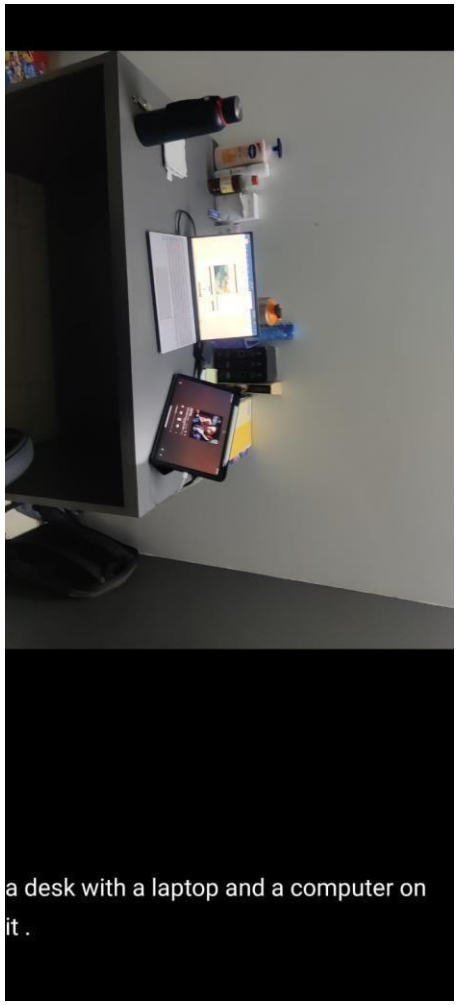


Fig.5.2.2 Working

Chapter 6: Conclusion

Therefore, in summary, the development of Image Caption Generator as a hybrid model formed by CNN and RNN supported on the basis of the Flickr 30k dataset is one of the big steps in the direction of integration of computer vision and natural language processing. Therefore, the version that is able to translate visual content into meaningful, context-based captions is built based on such problematic integration of Convolutional Neural Networks and Recurrent Neural Networks. This indicates an impressive mastery of environmental complexities, ability to adapt to diverse situations, as well as an ethical approach to elimination of bias. Trendy benchmarks have validated its advanced caption fin and scalability for real-time applications. Again, looking at a person's attractivity checking out verifies the utilitarian nature of the Image Caption Generator while interpretability measurements give us an idea about the version’s judgment mechanisms. The contribution made by this assignment is not just a state-of-the-art technology solution but it also highlights the need for moral issues, personal satisfaction and resilience when deploying AI systems. All in all, broad documentation ensures knowledge transfer, future modifications, and packages at the crossroads of computer vision and natural language processing.

REFERENCES

1. Image Captioning Based on Deep Neural Networks Shuang Liu1 , Liang Bai1,a, Yanli Hu1 and Haoran Wang1 1College of Systems Engineering, National University of Defense Technology,410073 Changsha,China

2. Component based comparative analysis of each module in image captioning SeoungHo Choia , Seoung Yeon Job , Sung Hoon Jungc,

3. Deep Residual Learning for Image Recognition, Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, Microsoft Research

4. Image Caption Generator using CNN-LSTM Model, Savitha S1, Vishal V2, Suhas CV3, Sai Krishna Teja4, Shivaprasad SB5

5. Automatic image captioning, Conference: Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Volume: 3

6. Very Deep Convolutional Networks for Large-Scale Image Recognition, Karen Simonyan, Andrew Zisserman

7. Show and Tell: A Neural Image Caption Generator, Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan

8. Automatic Image Captioning, Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu and Christos Faloutsos

9. "Deep Visual-Semantic Alignments for Generating Image Descriptions" by A. Karpathy et al.

10. "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering" by P. Anderson et al.

11. "Image Captioning with Semantic Attention" by F. Yang et al.

12. "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning" by J. Lu et al

13. "Image Captioning with Convolutional Neural Networks" by A. Farhadi et al.
14. "Attention Is All You Need" by A. Vaswani et al. (Although not specific to image captioning, this paper introduces the Transformer model, which has been influential in many natural language processing tasks, including image captioning.)

15. "Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Imageto-Sentence Models" by B. Plummer et al.

16. "Aligning Visual Regions and Textual Concepts for Semantic-Grounded Image Representations" by L. Huang et al.

17. “Up-Down: A Spatial Transformer for Bottom-Up Visual Saliency Prediction” by P. Anderson et al.

18. Knowing What, How and Why: A Near Complete Solution for Aspect-based Sentiment Analysis, Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, Luo Si

19. "Image Captioning with Semantic-Instance-Aware Attention" by J. Chen et al.

20. "Image Captioning with Transformer" by A. Dai et al.

Plagiarism Check Report

major project

ORIGINALITY REPORT

12%	10%	8%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	2%
2	www.informedia.cs.cmu.edu Internet Source	1%
3	export.arxiv.org Internet Source	1%
4	www.pnrjournal.com Internet Source	1%
5	www.researchsquare.com Internet Source	1%
6	Submitted to Anna University Student Paper	<1%
7	Seoung-Ho Choi, Seoung Yeon Jo, Sung Hoon Jung. "Component based comparative analysis of each module in image captioning", ICT Express, 2020 Publication	<1%
8	www.ijraset.com Internet Source	<1%

How much of this submission has been generated by AI?

0%

of qualifying text in this submission has been determined to be generated by AI.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

What does the percentage mean?
The percentage shown in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the percentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterisk for percentages less than 20 to call attention to the fact that the score is less reliable.

However, the final decision on whether any misconduct has occurred rests with the reviewer/instructor. They should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in greater detail according to their school's policies.

How does Turnitin's indicator address false positives?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be AI-generated will be highlighted blue on the submission text.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

What does 'qualifying text' mean?
Sometimes false positives (incorrectly flagging human-written text as AI-generated), can include lists without a lot of structural variation, text that literally repeats itself, or text that has been paraphrased without developing new ideas. If our indicator shows a higher amount of AI writing in such text, we advise you to take that into consideration when looking at the percentage indicated.

In a longer document with a mix of authentic writing and AI generated text, it can be difficult to exactly determine where the AI writing begins and original writing ends, but our model should give you a reliable guide to start conversations with the submitting student.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ __**Department:** _____ **Enrolment No** _____

Contact No. _____**E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com