MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade

Project Report

on

"Image Caption Generator"

Submitted By:

Aashutosh Savita (0901AM211001)

Pratima Jadon (0901AM211039)

Faculty Mentor:

Dr. Rajni Ranjan Singh , Coordinator of Center for AI

CENTER FOR ARTIFICIAL INTELLIGENCE
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Aashutosh Savita (0901AM211001)** & **Pratima Jadon(0901AM211039)** has submitted the project report titled **Image Caption Generator** under the mentorship of **Dr. Rajni Ranjan Singh**, in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in **Artificial Intelligence & Machine Learning** from Madhav Institute of Technology and Science, Gwalior.

23/11/23

Dr. R. R. Singh
Faculty Mentor
Coordinator
Center for Artificial Intelligence

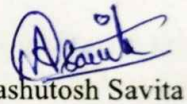
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

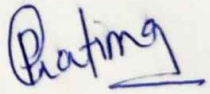
DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfillment of requirement for the award of the degree of Bachelor of Technology in **Artificial Intelligence & Machine Learning** at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Dr. R. R. Singh, Coordinator, Center for AI.** I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.



Aashutosh Savita

0901AM211001



Pratima Jadon

0901AM211039

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Dr. R.R Singh**, **Coordinator, Center for Artificial Intelligence,** for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Aashutosh Savita

0901AM211001

Pratima Jadon

0901AM211039

ABSTRACT

In this project, we use CNN and LSTM to identify the caption of the image. As the deep learning techniques are growing, huge datasets and computer power are helpful to build models that can generate captions for an image. This is what we are going to implement in this Python based project where we will use deep learning techniques like CNN and RNN. Image caption generator is a process which involves natural language processing and computer vision concepts to recognize the context of an image and present it in English. In this survey paper, we carefully follow some of the core concepts of image captioning and its common approaches. We discuss Keras library, numpy and jupyter notebooks for the making of this project.We also discuss about flickr_dataset and CNN used for image classification.

KEYWORDS: *generate captions, deep learning techniques, concepts of image captioning.*

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

PROJECT OVERVIEW

1.1 INTRODUCTION

Every day we come across a large number of images from various sources such as the internet, news articles, document diagrams and advertisements. These sources contain images that the viewer must interpret for themselves. Most images have no description, but humans can largely understand them without their detailed captions. However, a machine must interpret some form of caption if humans require automatic captions from it.

Captions are important for many reasons. Captions for any image on the Internet can lead to faster and more descriptive image searching and indexing.

Since researchers have been working on object recognition in images, it has become clear that simply stating the names of recognized objects does not make as good an impression as a full, human-like description. As long as machines do not think, speak and behave like humans, descriptions in natural language will remain a challenge to be solved.

Image labeling is used in various fields such as biomedicine, commerce, web search and military, etc. Application. Social media such as Instagram, Facebook, etc. can automatically generate captions from images.

1.2 MOTIVATION

A crucial task in the fields of computer vision and natural language processing is creating captions for photographs. A machine that can mimic a human's capacity to describe a visual is a tremendous advancement in artificial intelligence. Capturing the relationships between things in the image and expressing them in a normal language (like English) is the main problem of this task.Computer systems have traditionally generated text descriptions for photographs by utilizing pre-defined templates. Nevertheless, this method falls short of offering the necessary diversity to provide text descriptions that are rich in vocabulary. With neural networks' enhanced efficiency, this weakness has been minimized.

1.3 IMAGE CAPTIONING

Process :- Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions. Image captioning is a popular research area of Artificial Intelligence (AI) that deals with image understanding and a language description for that image. Image understanding needs to detect and recognize objects. It also needs to understand scene type or location, object properties and their interactions. Generating well-formed sentences requires both syntactic and semantic understanding of the language .Understanding an image largely depends on obtaining image features. Therefore, it can be applied to many areas, including biomedicine, commerce, the military, education, digital libraries, and web searching. Social media platforms such as Facebook and Twitter can directly generate descriptions from images. The descriptions can include where we are (e.g., beach, cafe), what we wear and importantly what we are doing there.

Techniques :- The techniques used for this purpose can be broadly divided into two categories: (1) Traditional machine learning based techniques and (2) Deep machine learning based techniques. In traditional machine learning, hand crafted features such as Local Binary Patterns (LBP) , Scale-Invariant Feature Transform (SIFT) , the Histogram of Oriented Gradients (HOG)], and a combination of such features are widely used. In these techniques, features are extracted from input data. They are then passed to a classifier such as Support Vector Machines (SVM) in order to classify an object. Since hand crafted features are task specific, extracting features from a large and diverse set of data is not feasible. Moreover, real world data such as images and video are complex and have different semantic interpretations.

On the other hand, in deep machine learning based techniques, features are learned automatically from training data and they can handle a large and diverse set of images and videos.

For example, Convolutional Neural Networks (CNN) are widely used for feature learning, and a classifier such as Softmax is used for classification. CNN is generally followed by Recurrent Neural Networks (RNN) or Long Short-Term Memory Networks (LSTM) in order to generate

captions.Deep learning algorithms can handle complexities and challenges of image captioning quite well.

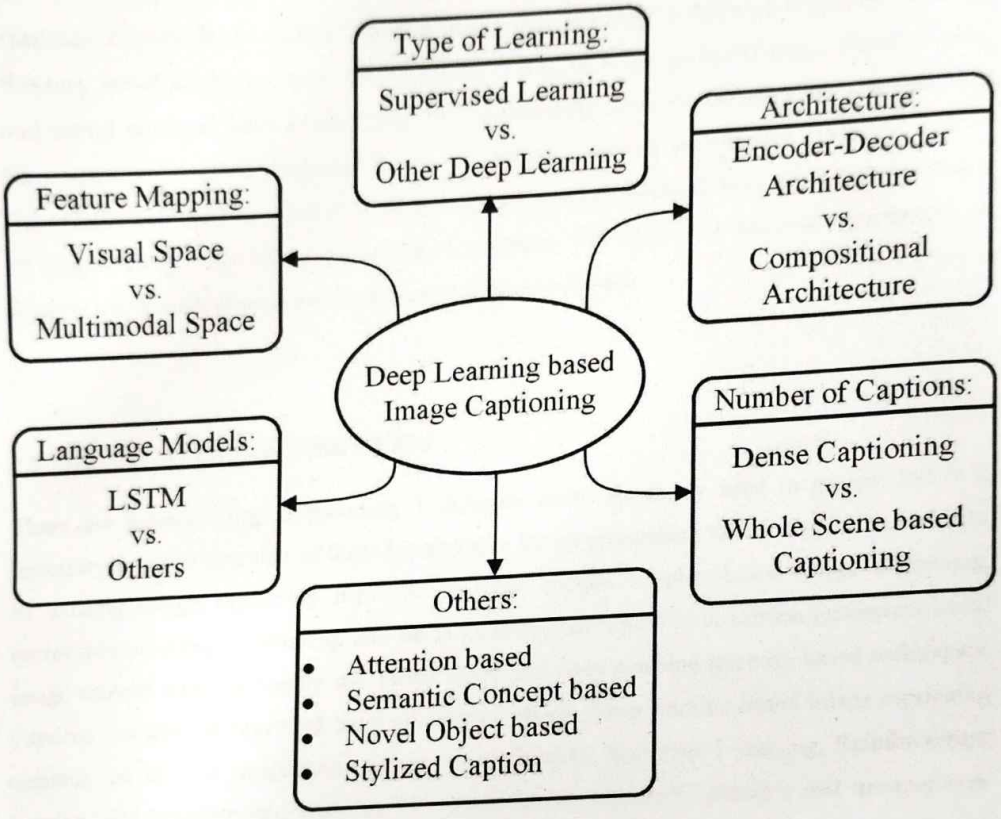


Figure.1.1 An overall taxonomy of deep learning-based image captioning.

CHAPTER 2

LITERATURE REVIEW

Recently, image captioning has received a lot of interest, particularly in the field of natural language. Context-based natural language description of images is desperately needed. Although this may sound idealistic, recent advancements in natural language processing, computer vision, and neural networks have made it possible to accurately describe images, that is, to represent their visually grounded meaning. To do so, we are utilizing cutting-edge methods such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and relevant datasets of images along with human-perceived descriptions. We show that retrieval experiments on datasets like Flicker provide results from our alignment model.

2.1 IMAGE CAPTIONING METHODS

There are various Image Captioning Techniques some are rarely used in present but it is necessary to take a overview of those technologies before proceeding ahead. The main categories of existing image captioning methods and they include template-based image captioning, retrieval-based image captioning, and novel caption generation.Novel caption generation-based image caption methods mostly use visual space and deep machine learning based techniques. Captions can also be generated from multimodal space. Deep learning-based image captioning methods can also be categorized on learning techniques: Supervised learning, Reinforcement learning, and Unsupervised learning. We group the reinforcement learning and unsupervised learning into Other Deep Learning. Usually captions are generated for a whole scene in the image. However, captions can also be generated for different regions of an image (Dense captioning). Image captioning methods can use either simple Encoder-Decoder architecture or Compositional architecture. There are methods that use attention mechanism, semantic concept, and different styles in image descriptions. Some methods can also generate description for unseen objects. We group them into one category as "Others". Most of the image captioning methods use LSTM as language model. However, there are a number of methods that use other language models such as CNN and RNN. Therefore, we include a language model-based category as "LSTM vs. Others.

2.2 DEEP LEARNING BASED IMAGE CAPTIONING METHODS

We draw an overall taxonomy in Figure 1 for deep learning-based image captioning methods. We discuss their similarities and dissimilarities by grouping them into visual space vs. multimodal space, dense captioning vs. captions for the whole scene, Supervised learning vs. Other deep learning, Encoder-Decoder architecture vs. Compositional architecture, and one „Others" group that contains Attention-Based, Semantic Concept-Based, Stylized captions, and Novel Object-Based captioning. We also create a category named LSTM vs. Others.

A brief overview of the deep learning-based image captioning methods is shown in table. It contains the name of the image captioning methods, the type of deep neural networks used to encode image information, and the language models used in describing the information. In the final column, we give a category label to each captioning technique based on the taxonomy in Figure 1.

2.3 SUPERVISED LEARNING VS. OTHER DEEP LEARNING

In supervised learning, training data come with desired output called label. Unsupervised learning, on the other hand, deals with unlabeled techniques. Reinforcement learning is another type of machine learning approach where the aims of an agent are to discover data and/or labels through exploration and a reward signal. A number of image captioning methods use reinforcement learning and GAN based approaches. These methods sit in the category of "Other Deep Learning". beled data. Generative Adversarial Networks (GANs) are a type of unsupervised learning.

2.4 DENSE CAPTIONING VS. CAPTIONS FOR THE WHOLE SCENE

In dense captioning, captions are generated for each region of the scene. Other methods generate captions for the whole scene.

2.4.1. DENSE CAPTIONING

The previous image captioning methods can generate only one caption for the whole image. They use different regions of the image to obtain information of various objects. However, these methods do not generate region wise captions. Johnson proposed an image captioning method called DenseCap. This method localizes all the salient regions of an image and then it generates descriptions for those regions.

A typical method of this category has the following steps:

(1) Region proposals are generated for the different regions of the given image.

(2) CNN is used to obtain the region-based image features.

(3)The outputs of Step 2 are used by a language model to generate captions for every region.

A block diagram of a typical dense captioning method is given in Figure 4.

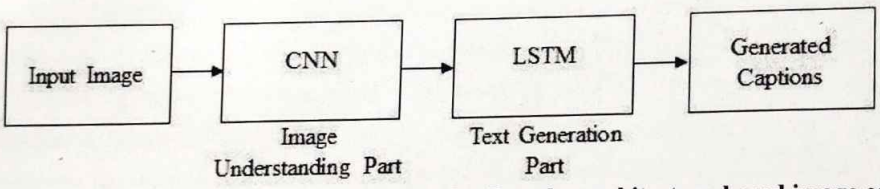


Figure.2.4. A block diagram of simple Encoder-Decoder architecture-based image captioning.

2.4.2 CAPTIONS FOR THE WHOLE SCENE

Encoder-Decoder architecture, Compositional architecture, attention-based, semantic concept-based, stylized captions, Novel object-based image captioning, and other deep learning networks-based image captioning methods generate single or multiple captions for the whole scene.

2.5 LSTM VS. OTHERS

Image captioning intersects computer vision and natural language processing (NLP) research. NLP tasks, in general, can be formulated as a sequence to sequence learning. Several neural language models such as neural probabilistic language model , log-bilinear models , skip-gram models , and recurrent neural networks (RNNs) have been proposed for learning sequence to sequence tasks. RNNs have widely been used in various sequence learning tasks. However, traditional RNNs suffer from vanishing and exploding gradient problems and cannot adequately handle long-term temporal dependencies.

LSTM networks are a type of RNN that has special units in addition to standard units. LSTM units use a memory cell that can maintain information in memory for long periods of time. In recent years, LSTM based models have dominantly been used in sequence to sequence learning tasks. Another network, Gated Recurrent Unit (GRU) has a similar structure to LSTM but it does not use separate memory cells and uses fewer gates to control the flow of information.

However, LSTMs ignore the underlying hierarchical structure of a sentence. They also require significant storage due to long-term dependencies through a memory cell. In contrast, CNNs can learn the internal hierarchical structure of the sentences and they are faster in processing than LSTMs. Therefore, recently, convolutional architectures are used in other sequence to sequence tasks, e.g., conditional image generation and machine translation. Inspired by the above success of CNNs in sequence learning tasks, Gu proposed a CNN language model-based image captioning method. This method uses a language-CNN for statistical language modelling. However, the method cannot model the dynamic temporal behaviour of the language model only using a language-CNN. It combines a recurrent network with the language-

CNN to model the temporal dependencies properly.

.They use a feed- forward network without any recurrent function.

The architecture of the method has four components: (i) input embedding layer (ii) image embedding layer (iii) convolutional module, and (iv) output embedding layer. It also uses an attention mechanism to leverage spatial image features.

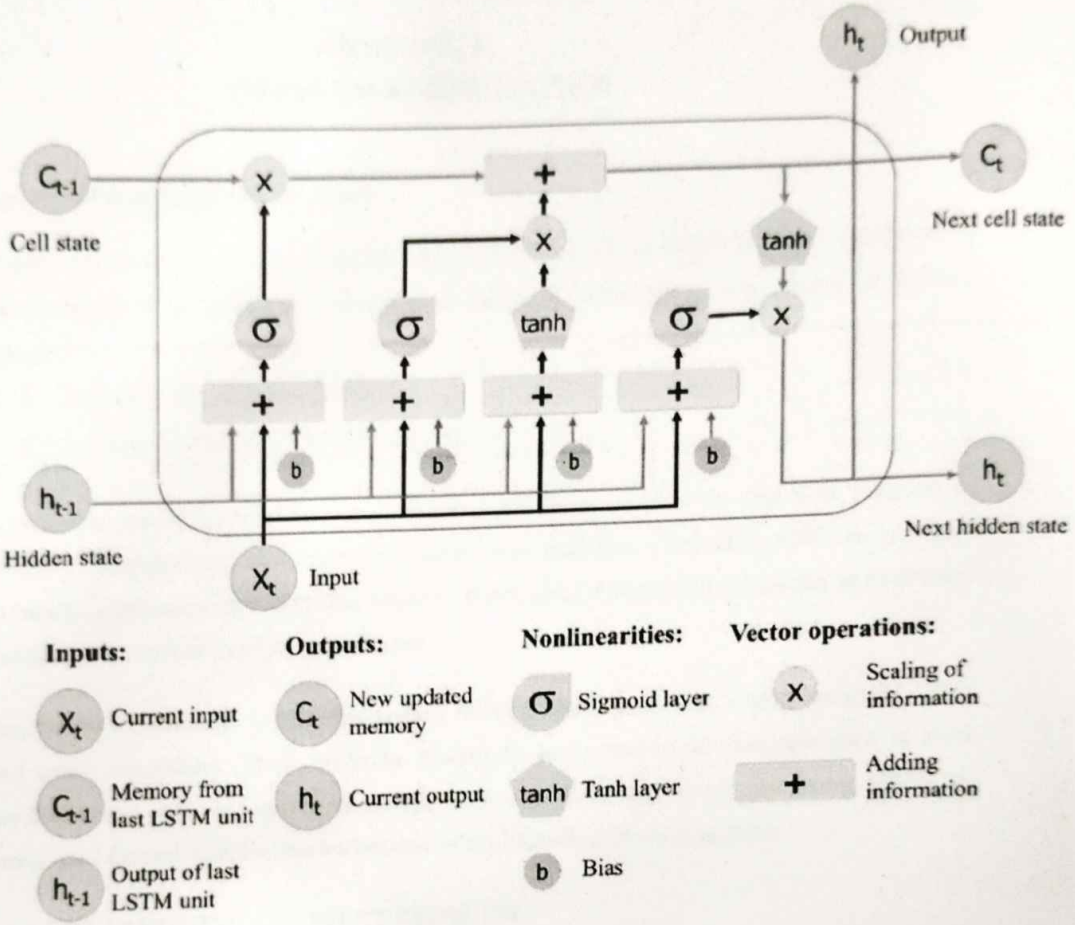


Figure 2.5 The structure of the Long Short-Term Memory (LSTM) neural network.

CHAPTER 3
PROBLEM FORMULATION

3.1 PROBLEM IDENTIFICATION

Despite the successes of many systems based on the Recurrent Neural Networks (RNN) many issues remain to be addressed. Among those issues the following two are prominent for most systems.

1. The Vanishing Gradient Problem.
2. Training an RNN is a very difficult task.

A recurrent neural network is a deep learning algorithm designed to deal with a variety of complex computer tasks such as object classification and speech detection. RNNs are designed to handle a sequence of events that occur in succession, with the understanding of each event based on information from previous events.

Ideally, we would prefer to have the deepest RNNs so they could have a longer memory period and better capabilities. These could be applied for many real-world use-cases such as stock prediction and enhanced speech detection. However, while they sound promising, RNNs are rarely used for real-world scenarios because of the vanishing gradient problem.

3.1.1 THE VANISHING GRADIENT PROBLEM

This is one of the most significant challenges for RNNs performance. In practice, the architecture of RNNs restricts its long-term memory capabilities, which are limited to only remembering a few sequences at a time. Consequently, the memory of RNNs is only useful for shorter sequences and short time-periods.

Vanishing Gradient problem arises while training an Artificial Neural Network. This mainly occurs when the network parameters and hyperparameters are not properly set. The vanishing gradient problem restricts the memory capabilities of traditional RNNs—adding too many time-steps increases the chance of facing a gradient problem and losing information when you use backpropagation.

CHAPTER 4

PROPOSED WORK

The main aim of this project is to get a little bit of knowledge of deep learning techniques. We use two techniques mainly CNN and LSTM for image classification.

So, to make our image caption generator model, we will be merging these architectures. It is also called a CNN-RNN model.

- CNN is used for extracting features from the image. We will use the pre-trained model Xception.
- LSTM will use the information from CNN to help generate a description of the image.

4.1 CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.

Convolutional Neural networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images.

It scans images from left to right and top to bottom to pull out important features from the image and combines the feature to classify images. It can handle the images that have been translated, rotated, scaled and changes in perspective.

4.2 LONG SHORT TERM MEMORY

LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information.

LSTMs are designed to overcome the vanishing gradient problem and allow them to retain information for longer periods compared to traditional RNNs. LSTMs can maintain a constant error, which allows them to continue learning over numerous time-steps and backpropagate through time and layers.

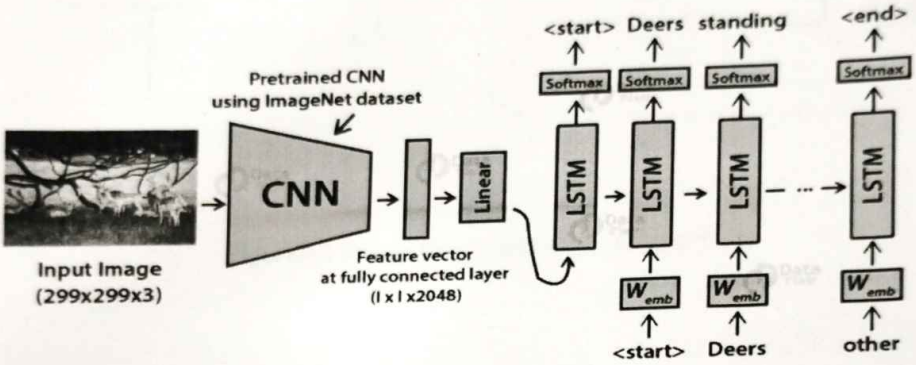


Figure. 4.1. Model, Image Caption Generator

LSTMs use gated cells to store information outside the regular flow of the RNN. With these cells, the network can manipulate the information in many ways, including storing information in the cells and reading from them. The cells are individually capable of making decisions regarding the information and can execute these decisions by opening or closing the gates. The ability to retain information for a long period of time gives LSTM the edge over traditional RNNs in these tasks.

The chain-like architecture of LSTM allows it to contain information for longer time periods, solving challenging tasks that traditional RNNs struggle to or simply cannot solve.

The three major parts of the LSTM include:

Forget gate—removes information that is no longer necessary for the completion of the task. This step is essential to optimizing the performance of the network.

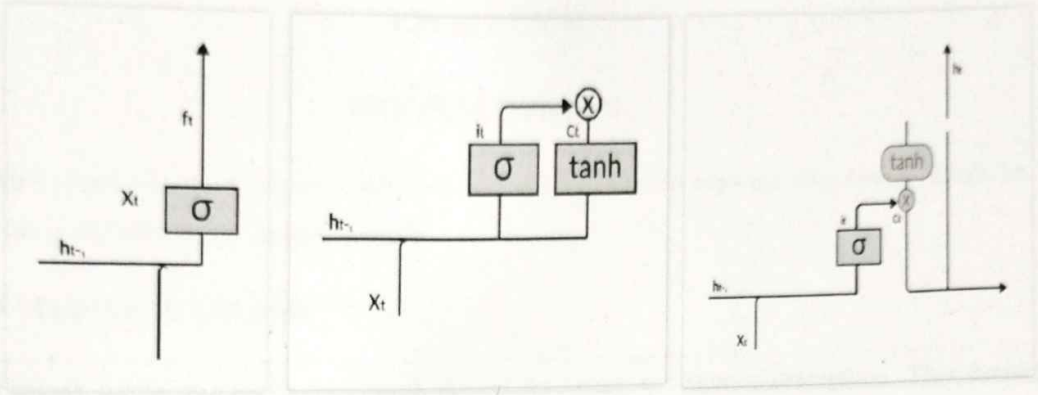


Figure.4.2 Forget Gate,Input Gate, Output Gate

Input gate—responsible for adding information to the cells

Output gate—selects and outputs necessary information

The CNN LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. This architecture was originally referred to as a Long-term Recurrent Convolutional Network or LRCN model, although we will use the more generic name "CNN LSTM" to refer to LSTMs that use a CNN as a front end in this lesson.

This architecture is used for the task of generating textual descriptions of images. Key is the use of a CNN that is pre-trained on a challenging image classification task that is re-purposed as a feature extractor for the caption generating problem.

CHAPTER 5

SYSTEM DESIGN

This project requires a dataset which have both images and their captions. The dataset should be able to train the image captioning model.

5.1 FLICKR 8K DATASET

Flickr8k dataset is a public benchmark dataset for image to sentence description. This dataset consists of 8096 images with five captions for each image. These images are extracted from diverse groups in Flickr website. Each caption provides a clear description of entities and events present in the image. The dataset depicts a variety of events and scenarios and doesn"t include images containing well-known people and places which makes the dataset more generic. The dataset has 6000 images in training dataset, 1000 images in development dataset and 1000 images in test dataset. Features of the dataset making it suitable for this project are: • Multiple captions mapped for a single image makes the model generic and avoids overfitting of the model.

- Diverse category of training images can make the image captioning model to work for multiple categories of images and hence can make the model more robust.

5.2 IMAGE DATA PREPARATION

The image should be converted to suitable features so that they can be trained into a deep learning model. Feature extraction is a mandatory step to train any image in deep learning model. The features are extracted using Convolutional Neural Network (CNN) with the Visual Geometry Group (VGG-16) model. This model also won ImageNet Large Scale Visual Recognition Challenge in 2015 to classify the images into one among the 1000 classes given in the challenge. Hence, this model is ideal to use for this project as image captioning requires identification of images.

In VGG-16, there are 16 weight layers in the network and the deeper number of layers help in better feature extraction from images. The VGG-16 network uses 3*3 convolutional layers making its architecture simple and uses max pooling layer in between to reduce volume size of

the image. The last layer of the image which predicts the classification is removed and the internal representation of image just before classification is returned as feature. The dimension of the input image should be 224*224 and this model extracts features of the image and returns a 1-dimensional 4096 element vector.

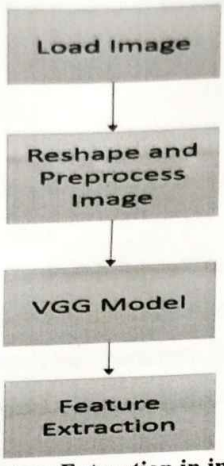


Figure 5.1: Feature Extraction in images using VGG

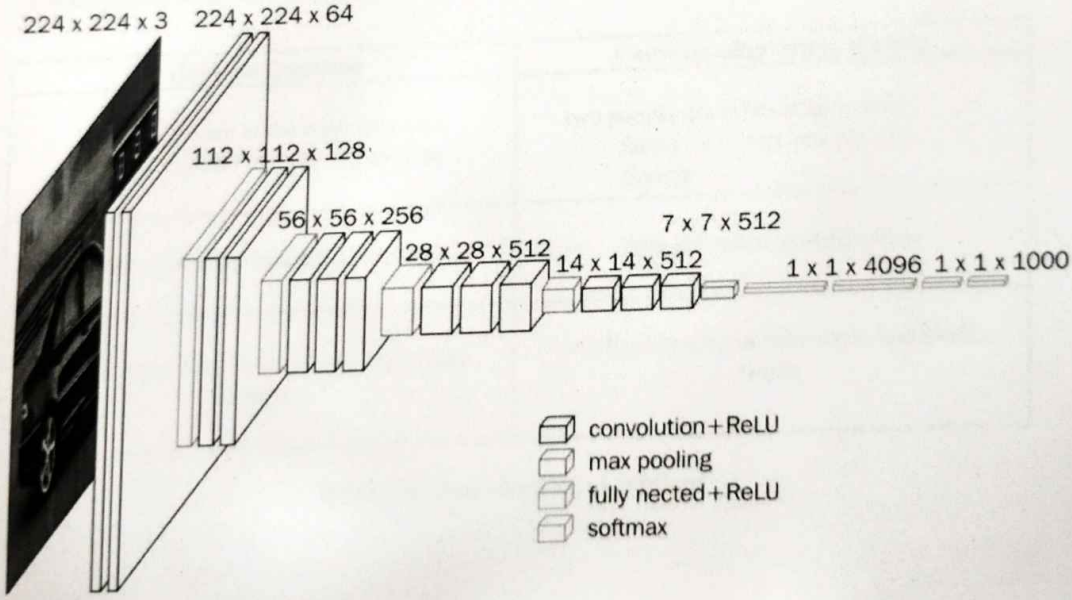


Figure 5.2 Architecture of VGG 16 model

5.3 CAPTION DATA PREPARATION

Flickr8k dataset contains multiple descriptions described for a single image. In the data preparation phase, each image id is taken as key and its corresponding captions are stored as values in a dictionary.

5.3.1 DATA CLEANING

In order to make the text dataset work in machine learning or deep learning models, raw text should be converted to a usable format.

The following text cleaning steps are done before using it for the project:

• Removal of punctuations. • Removal of numbers. • Removal of single length words.

• Conversion of uppercase to lowercase characters. Stop words are not removed from the text data as it will hinder the generation of a grammatically complete caption which is needed for this project.

Table 1 shows samples of captions after data cleaning.

Original Captions	Captions after Data cleaning
Two people are at the edge of a lake, facing the water and the city skyline.	two people are at the edge of lake facing the water and the city skyline
A little girl rides in a child 's swing.	little girl rides in child swing
Two boys posing in blue shirts and khaki shorts.	two boys posing in blue shirts and khaki shorts

Table 5.1: Data cleaning of captions

CHAPTER 6
IMPLEMENTATION

6.1 PRE-REQUISITES

This project requires good knowledge of Deep learning, Python, working on Jupyter notebooks, Keras library, Numpy, and Natural language processing.

Make sure you have installed all the following necessary libraries:

- pip install tensorflow
- keras
- numpy
- tqdm
- jupyterlab

6.2 PROJECT FILE STRUCTURE

Downloaded from dataset:

- **Flicker8k_Dataset** – Dataset folder which contains 8091 images.
- **Flickr_8k_text** – Dataset folder which contains text files and captions of images. The below files will be created by us while making the project.

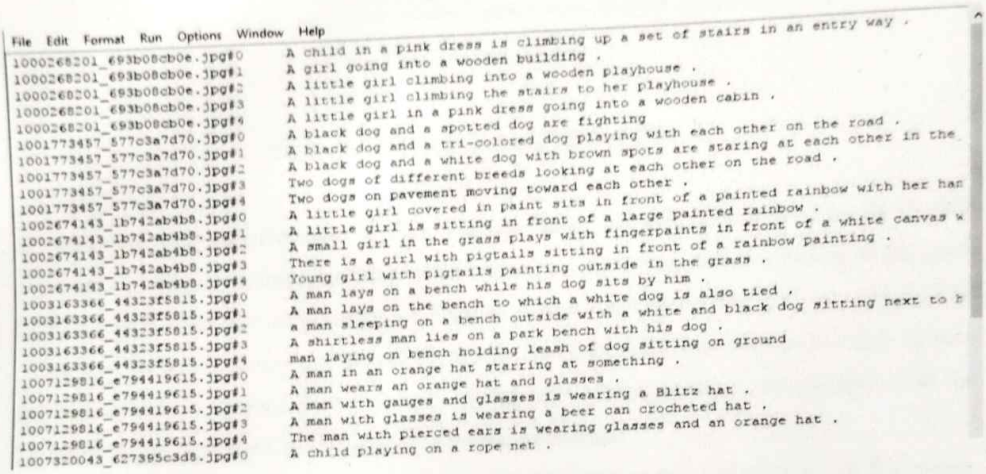
- **Models** – It will contain our trained models.
- **Descriptions.txt** – This text file contains all image names and their captions after preprocessing.
- **Features.p** – Pickle object that contains an image and their feature vector extracted from the Xception pre-trained CNN model.
- **Tokenizer.p** – Contains tokens mapped with an index value.
- **Model.png** – Visual representation of dimensions of our project.
- **Testing_caption_generator.py** – Python file for generating a caption of any image.
- **Training_caption_generator.ipynb** – Jupyter notebook in which we train and build our image caption generator.

6.3 BUILDING THE PYTHON BASED PROJECT

Let"s start by initializing the jupyter notebook server by typing jupyter lab in the console of your project folder. It will open up the interactive Python notebook where you can run your code. Create a Python3 notebook and name it **training_caption_generator.ipynb.**

6.3.1 GETTING AND PERFORMING DATA CLEANING

The main text file which contains all image captions is **Flickr8k.token** in our **Flickr_8k_text** folder.



```
File  Edit  Format  Run  Options  Window  Help
1000268201_693b08cb0e.jpg#0     A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg#1     A girl going into a wooden building .
1000268201_693b08cb0e.jpg#2     A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg#3     A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg#4     A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg#0     A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg#1     A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg#2     A black dog and a white dog with brown spots are staring at each other in the
1001773457_577c3a7d70.jpg#3     Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg#4     Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg#0     A little girl covered in paint sits in front of a painted rainbow with her han
1002674143_1b742ab4b8.jpg#1     A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg#2     A small girl in the grass plays with fingerpaints in front of a white canvas w
1002674143_1b742ab4b8.jpg#3     There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg#4     Young girl with pigtails painting outside in the grass .
1003163366_44323f5815.jpg#0     A man lays on a bench while his dog sits by him .
1003163366_44323f5815.jpg#1     A man lays on the bench to which a white dog is also tied .
1003163366_44323f5815.jpg#2     a man sleeping on a bench outside with a white and black dog sitting next to h
1003163366_44323f5815.jpg#3     A shirtless man lies on a park bench with his dog .
1003163366_44323f5815.jpg#4     man laying on bench holding leash of dog sitting on ground
1007129816_e794419615.jpg#0     A man in an orange hat starring at something .
1007129816_e794419615.jpg#1     A man wears an orange hat and glasses .
1007129816_e794419615.jpg#2     A man with gauges and glasses is wearing a Blitz hat .
1007129816_e794419615.jpg#3     A man with glasses is wearing a beer can crocheted hat .
1007129816_e794419615.jpg#4     The man with pierced ears is wearing glasses and an orange hat .
1007320043_627395c3d8.jpg#0     A child playing on a rope net .
```

Figure.6.1. Flicker DataSet text format

The format of our file is image and caption separated by a new line ("\n").

Each image has 5 captions and we can see that #(0 to 5)number is assigned for each caption. We will define 5 functions:

- **load_doc(filename)** – For loading the document file and reading the contents inside the file into a string.
- **all_img_captions(filename)** – This function will create a **descriptions** dictionary that maps images with a list of 5 captions. The descriptions dictionary will look something like the Figure.

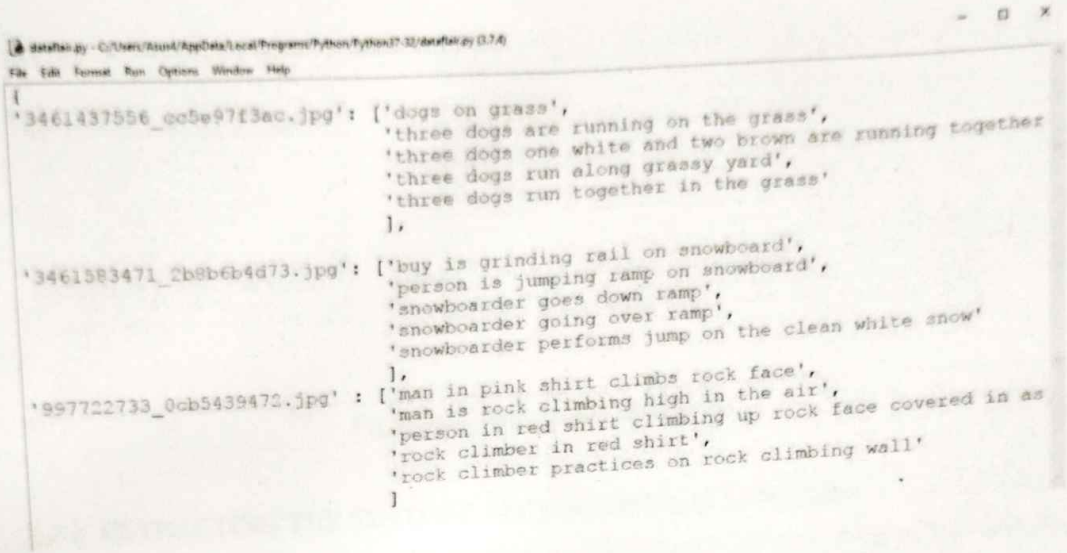


Figure.6.2. Flicker Dataset Python File

- **cleaning_text(descriptions)** – This function takes all descriptions and performs data cleaning. This is an important step when we work with textual data, according to our goal, we decide what type of cleaning we want to perform on the text. In our case, we will be removing punctuations, converting all text to lowercase and removing words that contain numbers.So, a caption like "A man riding on a three-wheeled wheelchair" will be transformed into "man riding on three wheeled wheelchair"
- **text_vocabulary(descriptions)** – This is a simple function that will separate all the unique words and create the vocabulary from all the descriptions.
- **save_descriptions(descriptions, filename)** – This function will create a list of all the descriptions that have been preprocessed and store them into a file. We will create a descriptions.txt file to store all the captions. It will look something like this:

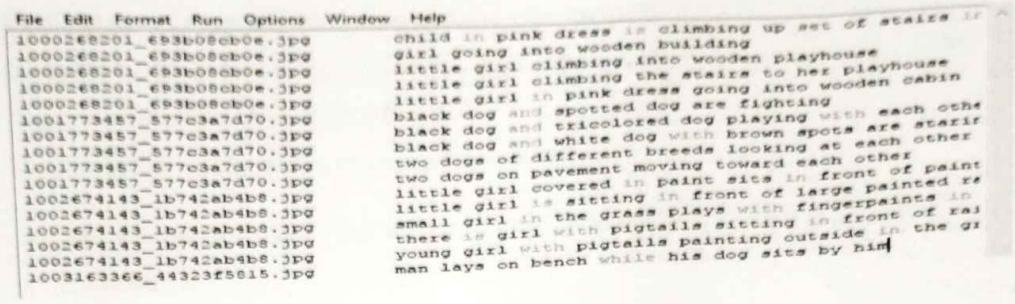


Figure.6.3. Description of Images

6.3.2 EXTRACTING THE FEATURE VECTOR FROM ALL IMAGES

This technique is also called transfer learning, we don"t have to do everything on our own, we use the pre-trained model that have been already trained on large datasets and extract the features from these models and use them for our tasks. We are using the Xception model which has been trained on imagenet dataset that had 1000 different classes to classify. We can directly import this model from the keras.applications . Make sure you are connected to the internet as the weights get automatically downloaded. Since the Xception model was originally built for imagenet, we will do little changes for integrating with our model. One thing to notice is that the Xception model takes 299*299*3 image size as input. We will remove the last classification layer and get the 2048 feature vector.

```
model = Xception( include_top=False, pooling="avg" )
```

The function **extract_features()** will extract features for all images and we will map image names with their respective feature array. Then we will dump the features dictionary into a "features.p" pickle file.

This process can take a lot of time depending on your system. I am using an Nvidia 1050 GPU for training purpose so it took me around 7 minutes for performing this task. However, if you are using CPU then this process might take 1-2 hours. You can comment out the code and directly load the features from our pickle file.

6.3.3 LOADING DATASET FOR TRAINING THE MODEL

In our **Flickr_8k_test** folder, we have **Flickr_8k.trainImages.txt** file that contains a list of 6000 image names that we will use for training.

For loading the training dataset, we need more functions:

- **load_photos(filename)** – This will load the text file in a string and will return the list of image names.

- **load_clean_descriptions(filename, photos)** – This function will create a dictionary that contains captions for each photo from the list of photos. We also append the <start> and <end> identifier for each caption. We need this so that our LSTM model can identify the starting and ending of the caption.

- **load_features(photos)** – This function will give us the dictionary for image names and their feature vector which we have previously extracted from the Xception model.

6.3.4 TOKENIZING THE VOCABULARY

Computers don"t understand English words, for computers, we will have to represent them with numbers. So, we will map each word of the vocabulary with a unique index value. Keras library provides us with the tokenizer function that we will use to create tokens from our vocabulary and save them to a **"tokenizer.p"** pickle file.

Our vocabulary contains 7577 words. We calculate the maximum length of the descriptions. This is important for deciding the model structure parameters. Max_length of description is 32.

6.3.5 Create Data generator

Let us first see how the input and output of our model will look like. To make this task into a supervised learning task, we have to provide input and output to the model for training. We have to train our model on 6000 images and each image will contain 2048 length feature vector and caption is also represented as numbers. This amount of data for 6000 images is not possible to hold into memory so we will be using a generator method that will yield batches.

The generator will yield the input and output sequence.

For example:

The input to our model is [x1, x2] and the output will be y, where x1 is the 2048 feature vector of that image, x2 is the input text sequence and y is the output text sequence that the model has to predict.

x1(feature vector)	x2(Text sequence)	y(word to predict)
feature	start,	two
feature	start, two	dogs
feature	start, two, dogs	drink
feature	start, two, dogs, drink	water
feature	start, two, dogs, drink, water	end

Table 6.1. Word Prediction Generation Step By Step

6.3.6 Defining the CNN-RNN model

To define the structure of the model, we will be using the Keras Model from Functional API. It will consist of three major parts:

- **Feature Extractor** – The feature extracted from the image has a size of 2048, with a dense layer, we will reduce the dimensions to 256 nodes.
- **Sequence Processor** – An embedding layer will handle the textual input, followed by the LSTM layer.
- **Decoder** – By merging the output from the above two layers, we will process by the dense layer to make the final prediction. The final layer will contain the number of nodes equal to our vocabulary size.

Visual representation of the final model is given in the figure

6.3.7 Training the model

To train the model, we will be using the 6000 training images by generating the input and output sequences in batches and fitting them to the model using model.fit_generator() method. We also save the model to our models folder. This will take some time depending on your system capability.

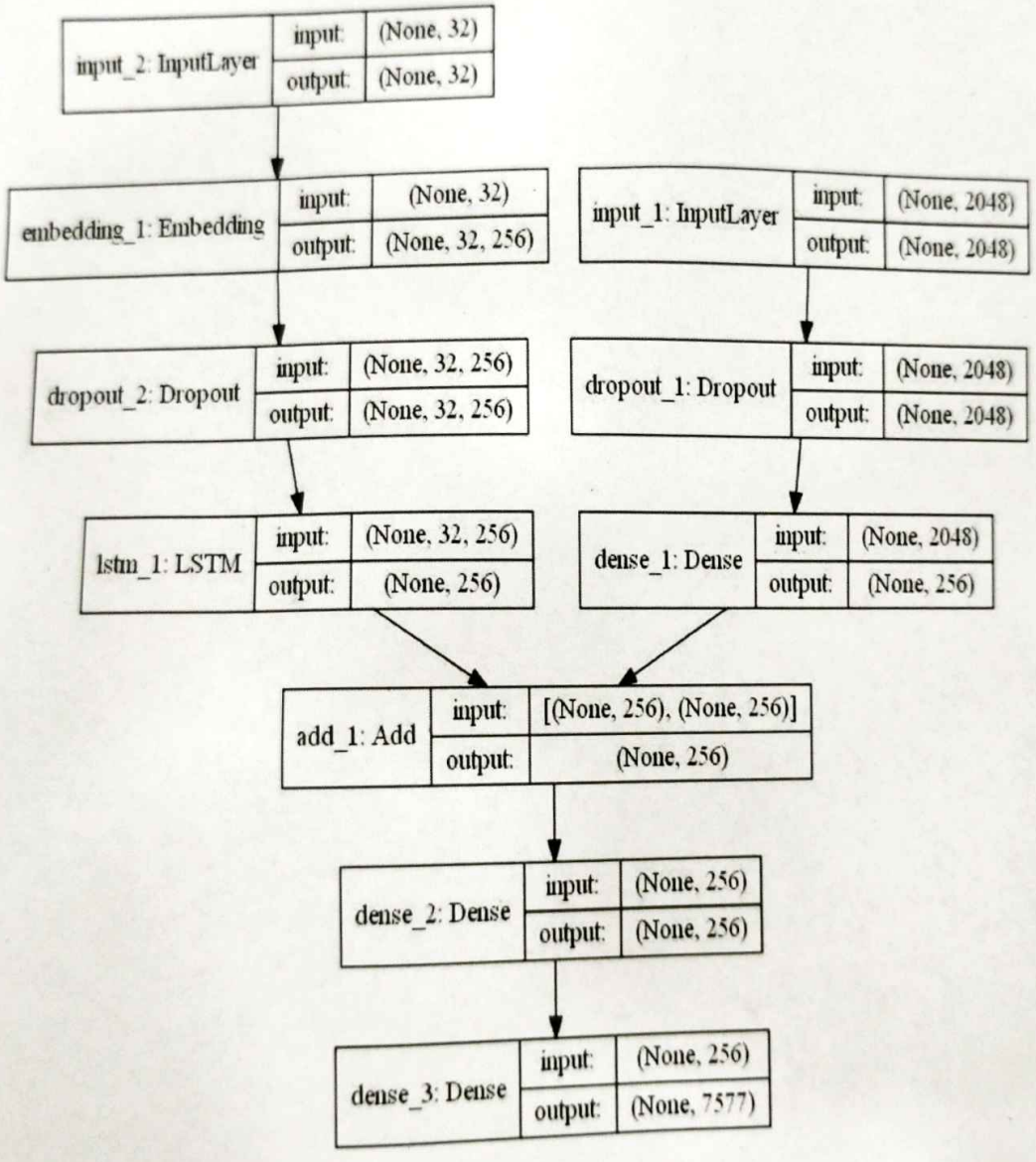


Figure.6.4. Final Model Structure

6.3.8 Testing the model

The model has been trained, now, we will make a separate file testing_caption_generator.py which will load the model and generate predictions. The predictions contain the max length of index values so we will use the same tokenizer.p pickle file to get the words from their index values.

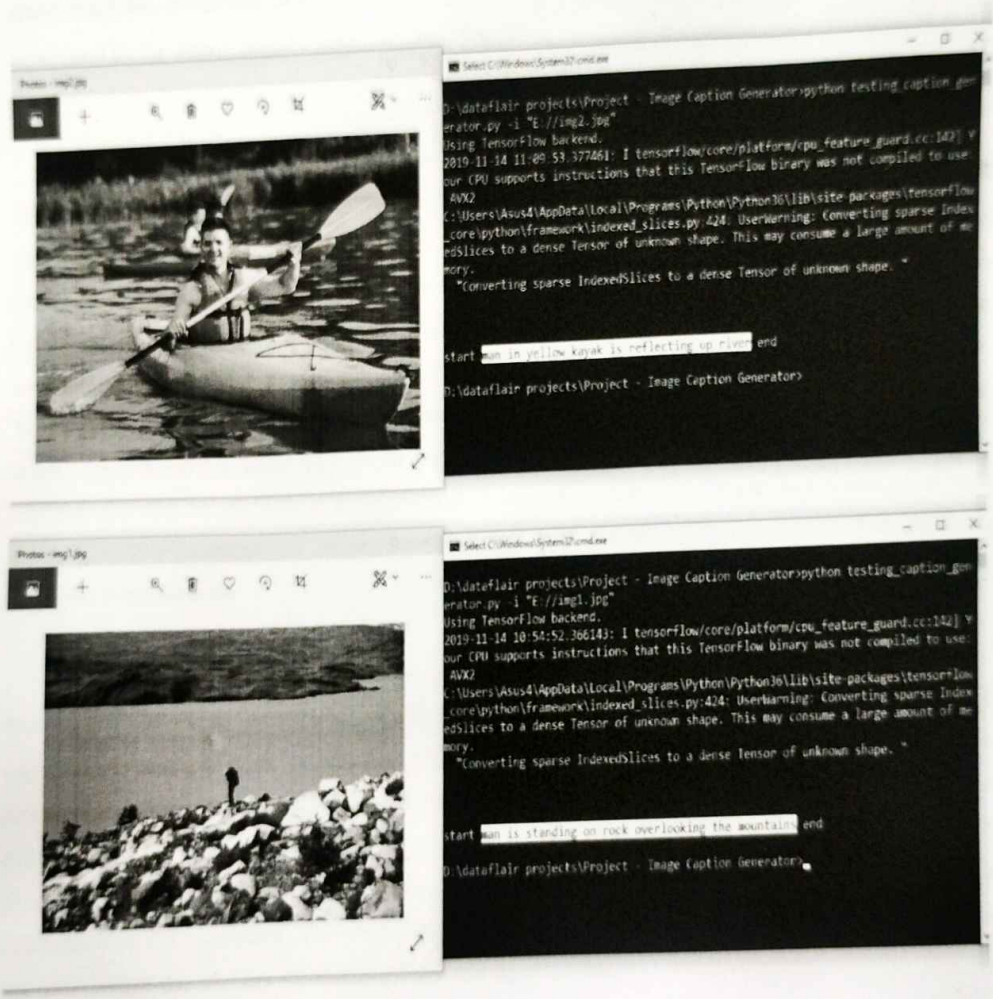


Figure.6.5. Output Caption of Given Image

CHAPTER 7

CONCLUSION, LIMITATION AND FUTURE SCOPE

In this chapter we have thrown some light on the conclusion of our project. We have also underlined the limitation of our methodology. There is a huge possibility in this field, as we have discussed in the future scope section of this chapter.

7.1 CONCLUSION

We have examined deep learning approaches for picture captioning in this study. A taxonomy of image captioning methods has been provided, along with a general block diagram of the main categories and a summary of their benefits and drawbacks. We talked about the advantages and disadvantages of various assessment criteria and datasets. The experimental outcomes are also briefly summarized. We gave a quick overview of possible lines of inquiry for this field of study. While deep learning-based image captioning techniques have advanced significantly in recent years, there is still a long way to go until a reliable technique that can provide high-quality captions for almost all images is developed. Automatic image captioning will continue to be a topic of active research for some time to come with the introduction of new deep learning network architectures.

We have used Flickr_8k dataset which includes nearly 8096 images, and the corresponding captions are also stored in the text file. Although deep learning -based image captioning methods have achieved a remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for sometime. The scope of image-captioning is very vast in the future as the users are increasing day by day on social media and most of them would post photos. So this project will help them to a greater extent.

7.2 LIMITATIONS

The neural image caption generator gives a useful framework for learning to map from images to human-level image captions. By training on large numbers of image-caption pairs, the model learns to capture relevant semantic information from visual features.

However, with a static image, embedding our caption generator will focus on features of our images useful for image classification and not necessarily features useful for caption generation. To improve the amount of task-relevant information contained in each feature, we can train the image embedding model (the VGG-16 network used to encode features) as a piece of the caption generation model, allowing us to fine-tune the image encoder to better fit the role of generating captions.

Also, if we actually look closely at the captions generated, we notice that they are rather mundane and commonplace. Take this possible image-caption pair for instance:

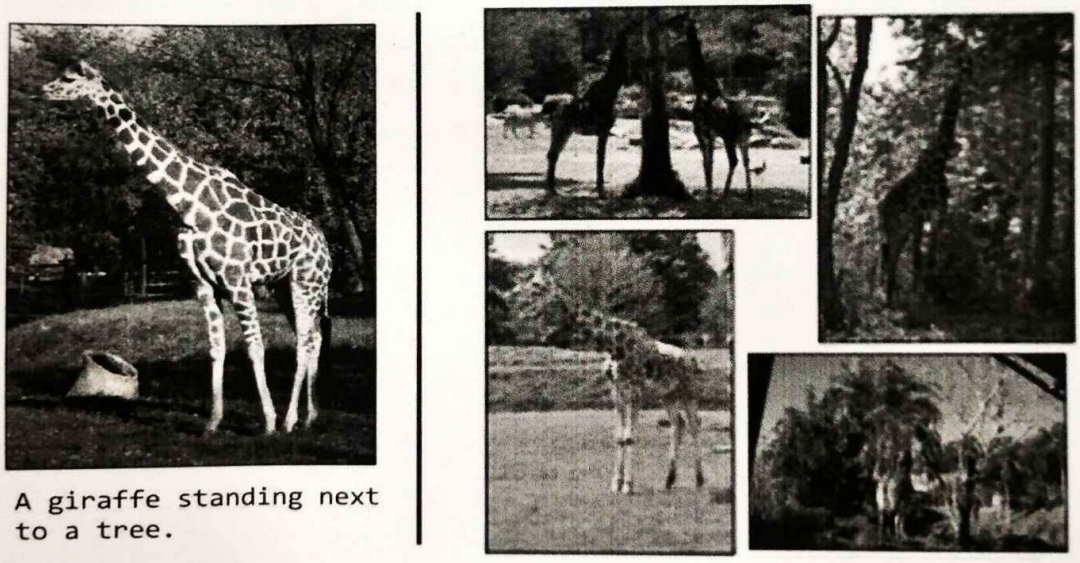


Figure.8.1. The above picture depicts clear limitation of the model because it rely most on the training dataset

This is most certainly a "giraffe standing next to a tree." However, if we look at other pictures, we will likely notice that it generates a caption of "a giraffe next to a tree" for any picture with a giraffe because giraffes in the training set often appear near trees.

7.3 FUTURE SCOPE

The exponential rise of photographs on the internet and in social media has made image captioning a significant issue in recent years. This study covers the different approaches and methodologies employed in the research, as well as discussing the various picture retrieval studies conducted in the past. Given the difficulties in feature extraction and similarity calculation in images, this field has a vast amount of untapped potential for future research.The similarity calculation used by current image retrieval systems makes use of various aspects, including color, tags, histograms, and more. Since these approaches don't rely on the image's context, findings cannot be entirely correct. Therefore, a thorough investigation into picture retrieval employing the image context, such as image captioning, will help to resolve this issue in the future. By training it on new picture captioning datasets, this project can be improved in the future to better identify classes with lesser precision.

This methodology can also be combined with previous image retrieval methods such as histogram, shapes, etc. and can be checked if the image retrieval results get better.

REFERENCES

[1] Abhaya Agarwal and Alon Lavie. 2008. Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with

human rankings of machine translation output. In Proceedings of the ThirdWorkshop on Statistical Machine Translation.

Association for Computational Linguistics, 115–118.

[2] Ahmet Aker and Robert Gaizauskas. 2010. Generating image descriptions using dependency relational patterns. In

Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational

Linguistics, 1250–1258.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image

caption evaluation. In European Conference on Computer Vision. Springer, 382–398.

[4] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition. 5561–5570.

[5] https://www.kaggle.com/code/shweta2407/vgg16-and-lstm-image-caption-generator

[6] https://www.researchgate.net/figure/The-structure-of-the-Long-Short-Term-Memory-LSTM-neural-network-Reproduced-from-Yan_fig8_3342685