D.K.T.E. Society’s Textile and Engineering Institute, Ichalkaranji.

(An Autonomous Institute)

Department of Computer Science and Engineering

2020-2021

DKTE

Promoting Excellence in
Teaching, Learning & Research

Project Report On

IMAGE CAPTION GENERATOR

Under the Guidance Of

Prof. K.S.Kadam

Submitted By:

1. AKKOLE PRASHANT POPAT	17UCS12001XX
2. FARAS SOHEL FIROJ	17UCS12059XX
3. TELI GURUPUTRA CHANNAPA	17UCS12061XX
4. TAHSILDAR MOINAHAMD MUJEEB	17UCS12060XX
5. SHINDE KRISHNA DNYANDEV	17UCS12055XX

Prof. K.S.Kadam	Prof.(Dr).D.V.Kodavade	Prof.(Dr).P.V.Kadole
(Project Guide)	**(HOD)**	**(Director)**

DKTE

Promoting Excellence in
Teaching, Learning & Research

YEAR 2020-2021

DEPARTMENT OF Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitles “Image Caption Generator “is record of project work carried out in this college by

AKKOLE PRASHANT POPAT	(17UCS12001XX),
FARAS SOHEL FIROJ	(17UCS12059XX),
TELI GURUPUTRA CHANNAPA	(17UCS12061XX),
TAHSILDAR MOINAHAMD MUJEEB	(17UCS12060XX),
SHINDE KRISHNA DNYANDEV	(17UCS12055XX)

In the partial fulfilment of the requirement for degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE of SHIVAJI UNIVERSITY, KOLHAPUR. This project report is a record of their own work carried out under my supervision and guidance during the academic year 2020-2021.

Prof. K.S.Kadam
[Project Guide]

Prof. (Dr.) D. V. Kodavade
[**Head of the Department]**

Prof. (Dr.) P.V. KADOLE
[Director]

DECLARATION

We hereby declare that the project work report entitled “Image Caption Generator” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji. An Autonomous Institute affiliated to Shivaji University, Kolhapur is in partial fulfilment of degree B. Tech (C.S.E). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy/Cyber/IPR Act amended from time to time.

Name	**Roll No**	**Signature**
1. AKKOLE PRASHANT POPAT	17UCS12001XX	
2. FARAS SOHEL FIROJ	17UCS12059XX	
3. TELI GURUPUTRA CHANNAPA	17UCS12061XX	
4. TAHSILDAR MOINAHAMD MUJEEB	17UCS12060XX	
5. SHINDE KRISHNA DNYANDEV	17UCS12055XX	

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our project coordinator
Prof. K.S.Kadam for his guidance, invaluable support and encouragement throughout the project. We would also like to thank our Head of the Department for providing us this opportunity to work on a project.

This project would have been impossible without our project guide and we want to extend sincere thanks to him for his guidance and constant supervision

as well as for providing necessary information regarding the project.

We would also like to express our gratitude towards our parents and our college for their kind cooperation and encouragement which helped us in completion of this project. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

Thank you,

AKKOLE PRASHANT POPAT	17UCS12001XX
FARAS SOHEL FIROJ	17UCS12059XX
TELI GURUPUTRA CHANNAPA	17UCS12061XX
TAHSILDAR MOINAHAMD MUJEEB	17UCS12060XX
SHINDE KRISHNA DNYANDEV	17UCS12055XX

ABSTRACT

Automatically describing the content of images using natural language is a fundamental and challenging task. With the advancement in computing power along with the availability of huge datasets, building models that can generate captions for an image has become possible. On the other hand, humans are able to easily describe the environments they are in.

Given a picture, it’s natural for a person to explain an immense number of details about this image with a fast glance. Although great development has been made in computer vision, tasks such as recognizing an object, action classification, image classification, attribute classification and scene recognition are possible but it is a relatively new task to let a computer describe an image that is forwarded to it in the form of a human-like sentence.

So, to make our image caption generator model, we will be merging CNN-RNN architectures. Feature extraction from images is done using CNN. The information received from CNN is then used by LSTM for generating a description of the image.

INDEX

Introduction

Caption generation is an artificial intelligence problem when a detailed sentence is found for a given image. It includes dual methods from computer vision to understanding the content of an image and converting a language model from the field of natural language processing into the perception of images into words in the correct order. Image caption includes recommendations for modifying applications, use in virtual assistants, for image indexing, visually impaired, social media and many other natural language processing applications. Recently, intensive teaching methods have yielded cutting-edge results on examples of this problem. Intensive learning models have proven to be able to achieve optimal results in the field of caption generation problems.

A single end-to-end model can be defined to estimate a given photo caption, rather than requiring a pipeline of complex data preparation or specially designed models. We developed a model using the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model and created a working model of the image caption generator by implementing CNN with LSTM.

CNN acts as an encoder to extract attributes from images and LSTM as an encoder to create words that describe images. The model consists of three parts. The first part is the CNN used to understand the content of the image. Image perception Computer vision "What are things?", "Where are things?" And "How do objects interact?", For example, CNN should identify "Teddy Bears", "Tables" and their relative positions in the image. The second part is the RNN used to construct a given sentence with a visual attribute. For example, RNN should generate a sequence of probabilities for two words, "teddy bear, table". The third part is used to form a sentence by searching for a combination of possibilities. Therefore, our system helps the user to get a detailed title for a given input image.

a. Problem definition

To develop a system for users, which can automatically generate the description of an image with the use of CNN along with LSTM {Long Short time memory}**.**

b. Aim and objective of the Project

Aim:

The goal of image captioning is to automatically generate descriptions for a given image, i.e., to capture the relationship between the objects present in the image, generate natural language expressions, and judge the quality of the generated descriptions.

The objectives of the project are given below:

- To detect the objects, present in the image.
- To generate the caption based on the objects detected from the image.
- To develop a user interface to get the description of an image.

c. Scope and limitation of Project

Scope:

Intelligent monitoring:

Intelligent monitoring enables the machine to identify and determine the behavior of people or vehicles in the captured scene and generate alarms under appropriate conditions to prompt the user to react to emergencies and prevent unnecessary accidents. For example, in channel monitoring, it collects the fairway operations and illegal activities, monitors the conditions of the fairway, and promptly discovers the conditions of the waterway operations, traffic conditions, illegal sand mining, and the use of navigation channels. Then report the situation to the command center for scheduling and stop illegal activities in a timely manner. Image captioning can be applied to this aspect. Through the image captioning methods, the machine can understand the scenes it captures, so that it can respond to specific situations or notify users in a timely manner based on human settings.

Human-computer interaction:

With the advancements of science and technology and the need for the development of human life, robots have been used in more and more industries. Auto-pilot robots can intelligently avoid obstacles, change lanes and pedestrians based on the road conditions according to the surrounding driving environment they observe. In addition to safe and efficient driving, it is also possible to perform operations such as automatic parking. Freeing the driver’s eyes and hands greatly simplifies people’s lives and reduces safety risks. If the machine wants to do the work better, it must interact with humans better. The machine can tell humans what it sees, and humans then perform appropriate processing based on machine feedback. To accomplish these tasks, we must rely on the automatic creation of image descriptions.

Limitations:

Richness of image semantics:

The current study can describe the image content to a certain extent, but it is not sensitive to the number of objects contained in the image. For example, the model often cannot accurately describe the objects with terms such as “two” or “group”. Besides, the selection of focus points in complex scenes are different. For individuals, it is easy to understand the important content of the film and capture the information of interest. But for the machine, this will not be easy. The current image description automatic generation technology can describe pictures with simple scenes more comprehensively, but if the picture contains complex scenes and numerous object and object relationships, the machine often cannot grasp the important content in the image well. More attention will be paid to some minor information. This situation often affects the final result of the image description, sometimes even misinterpreting the original meaning of the image content.

Inconsistent objects during training and testing:

From the current study, during the training process, the input to the network at each time step is a real word vector or a mixture of real words and images, and the output of the network is the predicted word. However, in the test process, the network inputs at each time step are the output word vector in the vocabulary of the training dataset. The existing training process relies heavily on the selection of data sets. Once a given image contains novel objects, the approach taken is to select the closest object from the data set instead of the true object. In this way, there are inconsistencies in the training and the testing process when the new objects are created. Such discrepancies may lead to and may be due to the cumulative error pattern in text descriptions that are completely inconsistent with the image content, resulting in incorrect description results.

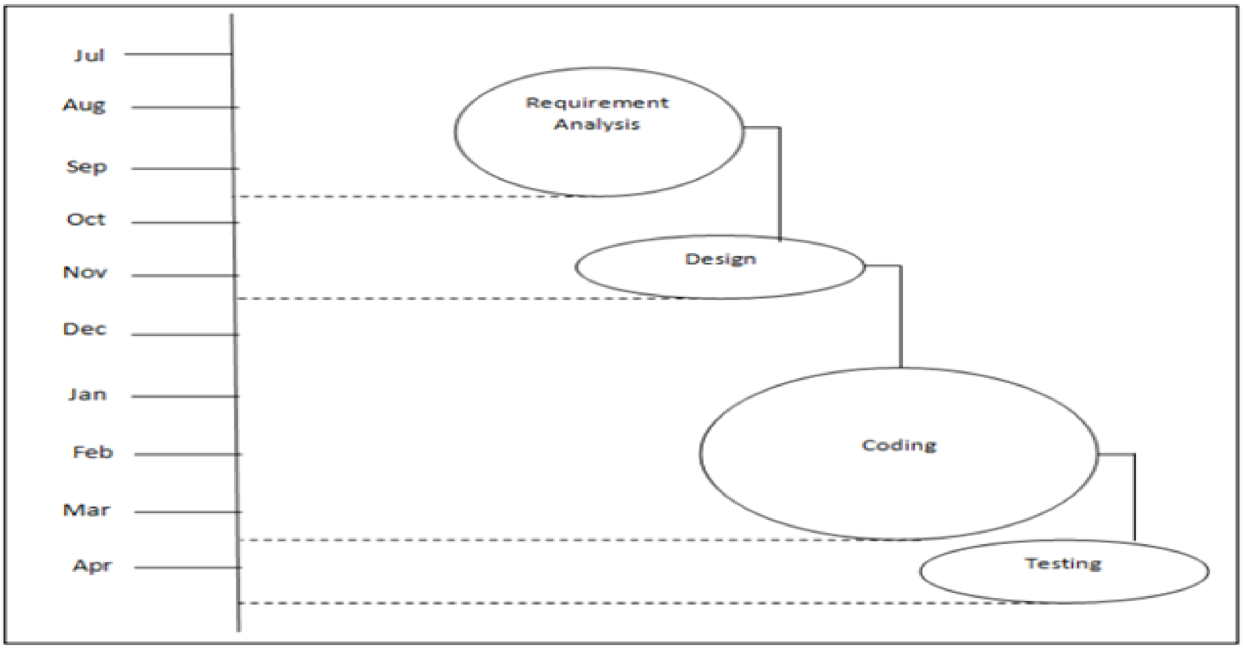
Cross-language text description of images:

The existing image captioning method based on deep learning or machine learning requires a lot of identified training models.. In practical applications, it is required that a text description of a plurality of languages can be provided for the image to meet the needs of different native language users. At present, there are many training samples described in English and Chinese texts, but there are few mark-ups in other language text descriptions. If the textual description of each language in the image is carried out, manual marking will require a lot of manpower and time. Therefore, how to implement cross language text.

d. Timeline of the Project

We have used the classic life cycle paradigm also called "Waterfall Model". For software engineering which is a sequential approach to software development that Starts at the system level and develops through analysis, design, coding, testing and maintenance. We had completed software requirement analysis by the mid of September 2020 which encompasses both system and software requirement gathering. By the end of December 2020, we had completed project planning and design.

On the basis of design prepared in the previous stage by the end of March 2021 we completed the coding stage. After completion of coding stage, the important part in the software development which is testing phase carried out in Third week of April 2021.Various criteria of testing were taken into account which includes unit testing, integration testing, validation testing and system testing. First, each and every module of the project was tested under the unit testing. After the unit testing, integration testing was carried out by integrating all modules tested in unit testing. After unit testing the module prepared was cross checked with the design.



Background study and Literature overview

Literature Survey

Literature Overview:

It is worth mentioning that intuition and development of image captioning technology is not new. Multiple corporations, research communities and open-source communities have features. Since we are not classifying features but only constantly striving to achieve higher performance and resilience in the technology. The race for better performance has grown rapidly since the advent of deep neural networks. Researchers have developed many neural networks with better performance and features than their predecessors.

Krizhevsky et al. developed a neural network powered by GPU training procedures. The model was successful in significantly reducing the input amplitude for output (none, 1000) output without model overfitting. It uses 5 convolution layers backed by Maxpooling layers and proper implementation of dropout regularization which enhances its performance. Karpathy and FeiFei were the first to process image datasets and their corresponding sentence descriptions to enable computers to generate image descriptions. It introduces a Multimodal Recurrent Neural Network (m-RNN) that uses the co-linear arrangement of features in order to carry out the task. Vinyals et al. developed a production model that includes RNN that promotes machine translation and computer vision for image Caption by confirming the good probability of the sentence being generated to accurately describe the target image. Xu et al. developed an attention-based model capable of automatically learning and interpreting the image. The model was trained using the standard backpropagation technique and was able to identify the object entities in the image as well as produce an accurate caption.

APPROACH

In this work, neural framework is proposed for generating captions from images which are basically derived from probability theory. By using a powerful mathematical model, it is possible to achieve better results, which maximizes the probability of the correct translation for both inference and training.

A. Convolutional Neural Network (CNN):

The convolutional networks are currently used in visual recognition. There are number of convolutional layers in CNN. After these convolutional layers, next layers are fully connected layers as in multilayer neural network [14]. CNN is designed to take advantage of the 2D structure of the input image. This target is accomplished with the help of number of local connections and tied weights along with various pooling techniques which result in translation invariant features. The main advantages of using CNN are ease of training and possessing less parameters as compared to other networks with equal number of hidden states. For this work, we are using Visual Group Geometry(VGG) network, which is Deep CNN for large scale image recognition [15]. It is available in 16 layers as well as 19 layers. The classification error results for both 16 and 19 layers are almost same for validation set as well as test set, which is around 7.4% and 7.3%. This model gives the features of images which are used in further process of caption generation.

B. **Long Short-Term Memory (LSTM)**:

The transitory dynamics in a set of things are modelled by using a recurrent neural network [17]. It is very difficult for ordinary RNN to acquire long term dynamics as they get vanished and exploding weights or gradients [9]. The memory cell is main block of LSTM. It stores the present value for long period of time. Gates are there for controlling update time of state of cell. The number of connections between memory cell and gates represent variants

C. GENERATION OF SENTENCE WITH LSTM:

The process of sentence generation in neural network is taken from principle of encoder decoder in modelling of network and machine translation [1], [11]–[13], [16]. In this modelling, variable sequence of words in natural language is mapped to distributed vector by using encoder. Then, a new sequence of words is generated by using decoder in natural target language depending on mapped vectors. In training process, the aim is to maximize chances of perfect translation such that the sentence is in natural source language. Applying this principle while generating the captions, the target is to maximize the amount of the image caption generated given an image.

Images and sentences are encoded as fixed-length vectors before using them as inputs to LSTM. First of all for each images, CNN features are computed and then they are mapped to the embedding matrix. A new sequence is generated by concatenating sequence of words and an image in a sentence. In this new sequence, image is considered as beginning symbol of sequence and the sequence of words is treated as the remaining part of new sequence. This new sequence is used as an input to the LSTM network for training purpose by iterating the recurrence connection for l from 1 to Li . The transfer matrix which is linear in nature for image features, word embedding matrix and some arguments of LSTM are parameters of neural model. The image caption model has three sub models, first one is image model which repeats the image feature vector 28 times having dimension 28 x 4096 here 28 represents the maximum number of words in a caption. The second one is language model consisting of single LSTM unit and outputs the matrix having dimension 28 x 256, 256 is the output size of LSTM unit and the final model merge these two vectors and pass it to another LSTM unit having output dimension 28 x 915. For training we pass same encoded text vector as target vector but while testing we just encode" sol" to feature vector along with test image feature vector and we get matrix of dimension 28 x 915 and we decode that matrix into sequence words.

Requirement Analysis

FUNCTIONAL REQUIREMENTS:

Functional requirements are featuring that the system will need in order to deliver or operate. In the case of this project, it is important to gather some of the requirements needed to achieve the goals already set. A usage case analysis with client (user) story is implemented, which results in the following functional and non-functional requirements being summarized. Functional requirements are collected from the user story developed from the minutes collected during meetings with the customer and are described here.

Functional Requirements:

1. The dataset should contain a considerable amount of data to train the model.
2. The system should have a browsing files function.
3. The system should have uploading functions for sample images to the server of the system.
4. The system should be able to detect objects from the uploaded image.
5. The system should return an error status page if something goes wrong, the system should be able to generate multiple captions for the image.
6. The system should be able to generate the best caption based on the ranking.

Non-Functional Requirement:

1) Object detection model should give maximum accuracy.
    a) Detection of objects should be done in proper time and more accurately.
    b) All objects should be detected within the image.
2) The system should give maximum accuracy while generating sentences.
    a) On the basis of object detection, the system should generate the sentences accurately.
    b) The system should not be confused while generating sentences.
3) Caption generation should be done with maximum accuracy.
    a) The caption should be grammatically correct.
    b) The caption should be meaningful.

REQUIREMENT ANALYSIS

NO.	Requirement	Essential/Desirable	Description of the Requirement	Remarks
RS1	The dataset should contain a considerable amount of data to train the model.	Essential	To train the model collective data required	Dataset consist of considerable amount of data to train model.
RS2	The system should have a browsing files function.	Essential	There should be an interface for browsing files	The image file is uploaded.
RS3	The system should have a uploading functions for sample images to server of system	Essential	There should be an interface for browsing files and uploading images.	The files are uploaded to Server

RS4	The system should be able to detect objects from the uploaded image.	Desirable	System should be able to identify type of uploading files type [.png, .jpeg, .jpg] correct is built by the system	The objects are detected from the image.
RS5	The system should return an error status page if something goes wrong.	Essential	Server will respond to errors in system operation.	The error condition is returned.
RS6	The system should be able to generate multiple captions for the image.	Essential	This will help in choosing best captions according to their rankings	Multiple captions are generated for the image.
RS7	The system should be able to generate the best caption based on the ranking.	Essential	Server will respond to a success page with expected output.	The best caption is created for the image.

SOFTWARE AND HARDWARE REQUIREMENTS

Software requirements:

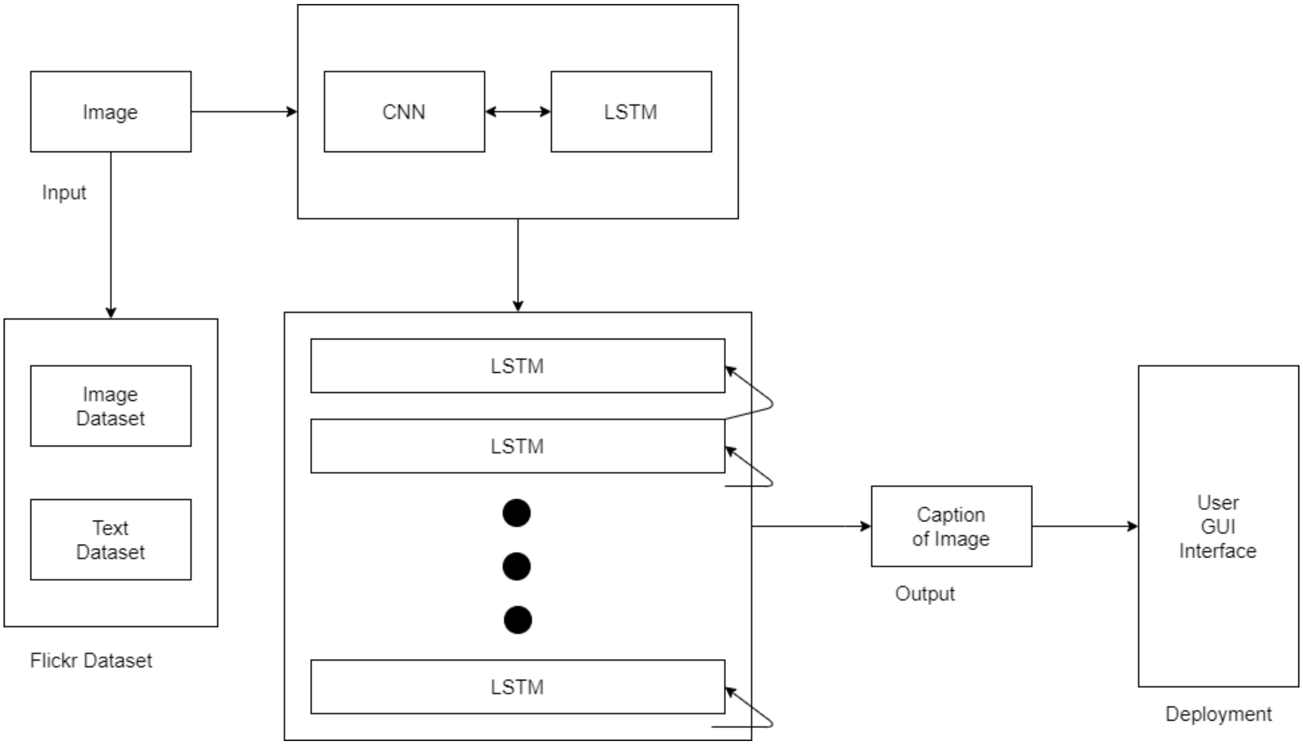
- Windows 10 or Ubuntu 18.4
- Web Browser
- Nginx or apache server
- Python3 with Flask Web Library
- Flicker8k Dataset
- DL Libraries {Anaconda Framework}

Hardware requirement:

- Processor: Intel Corei3 or later
- RAM: Minimum 4GB
- Secondary storage: Minimum 15GB
- Graphics card: NVidia or Google colab

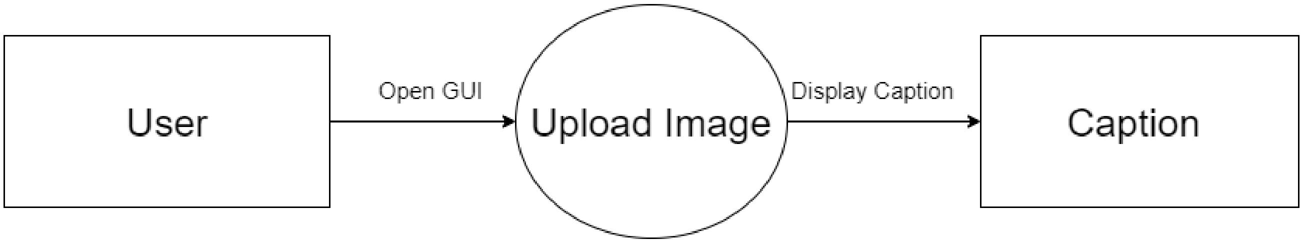
System Design

SYSTEM ARCHITECTURE:

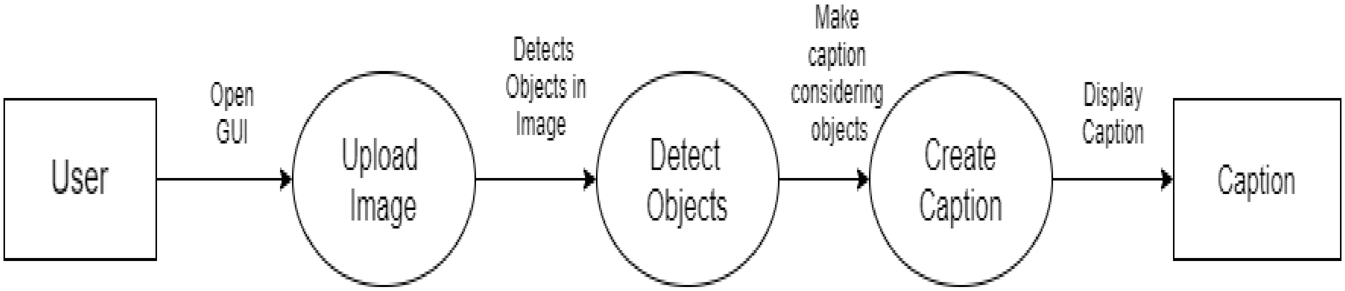


DATA FLOW DIAGRAM:

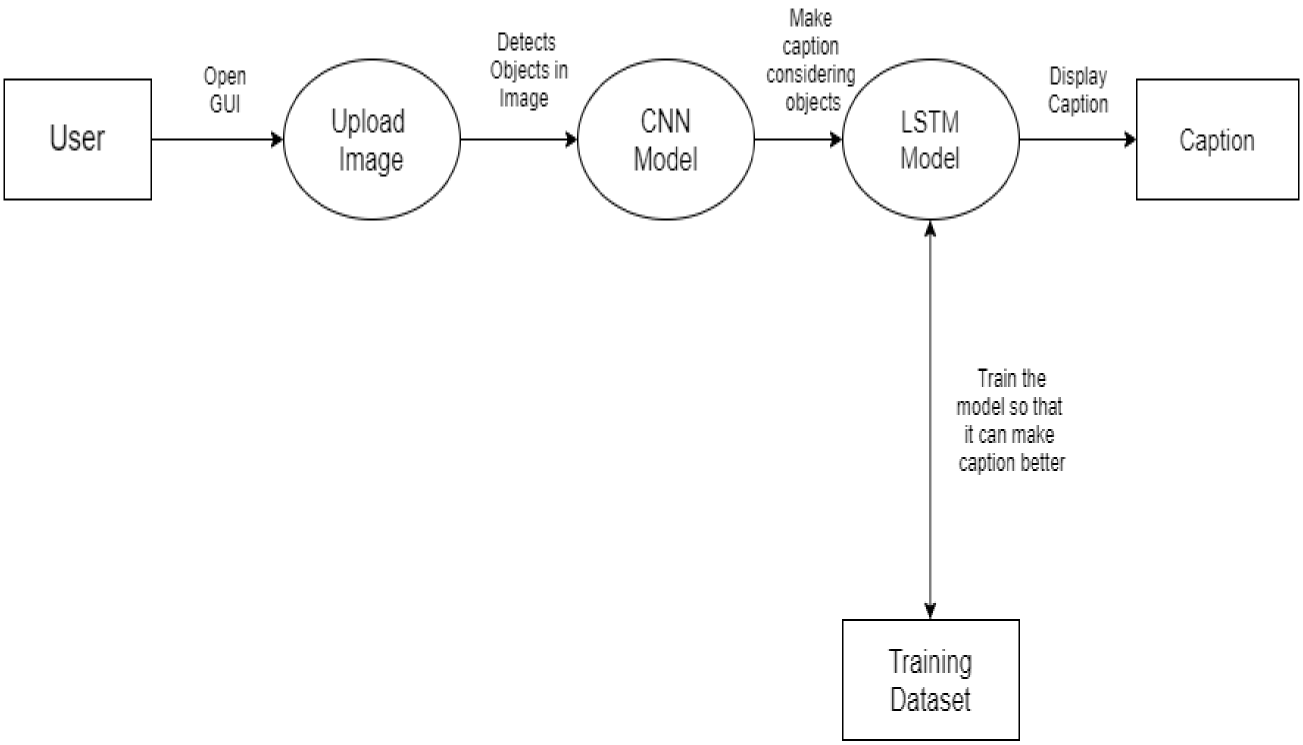
DFD level 0:



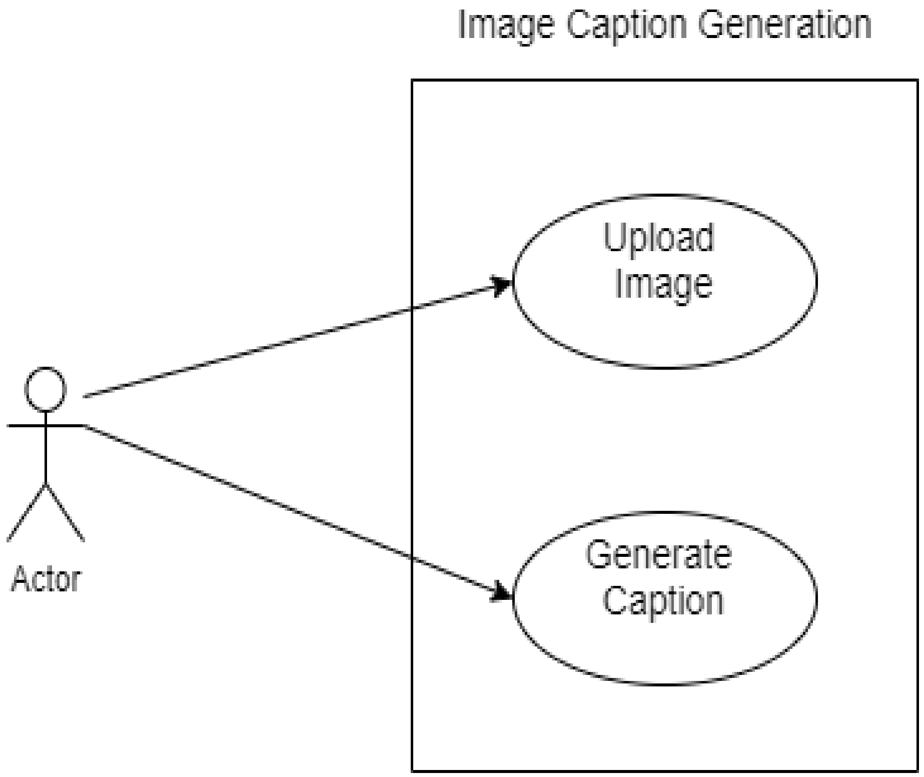
DFD level 1:



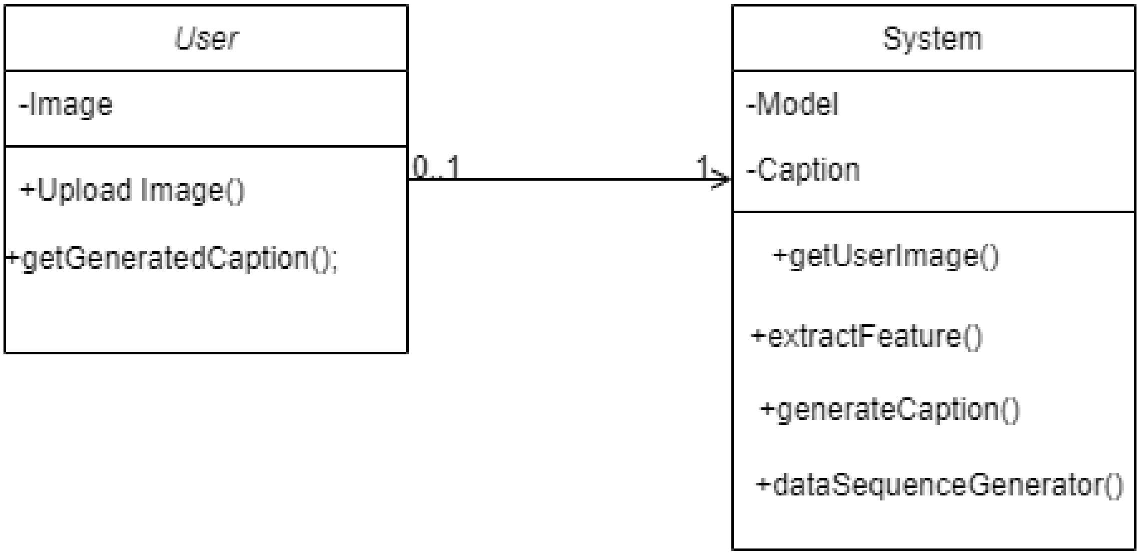
DFD level 2:


User
Open GUI
Upload Image
Detects Objects in Image
CNN Model
Make caption considering objects
LSTM Model
Display Caption
Caption
Train the model so that it can make caption better
Training Dataset

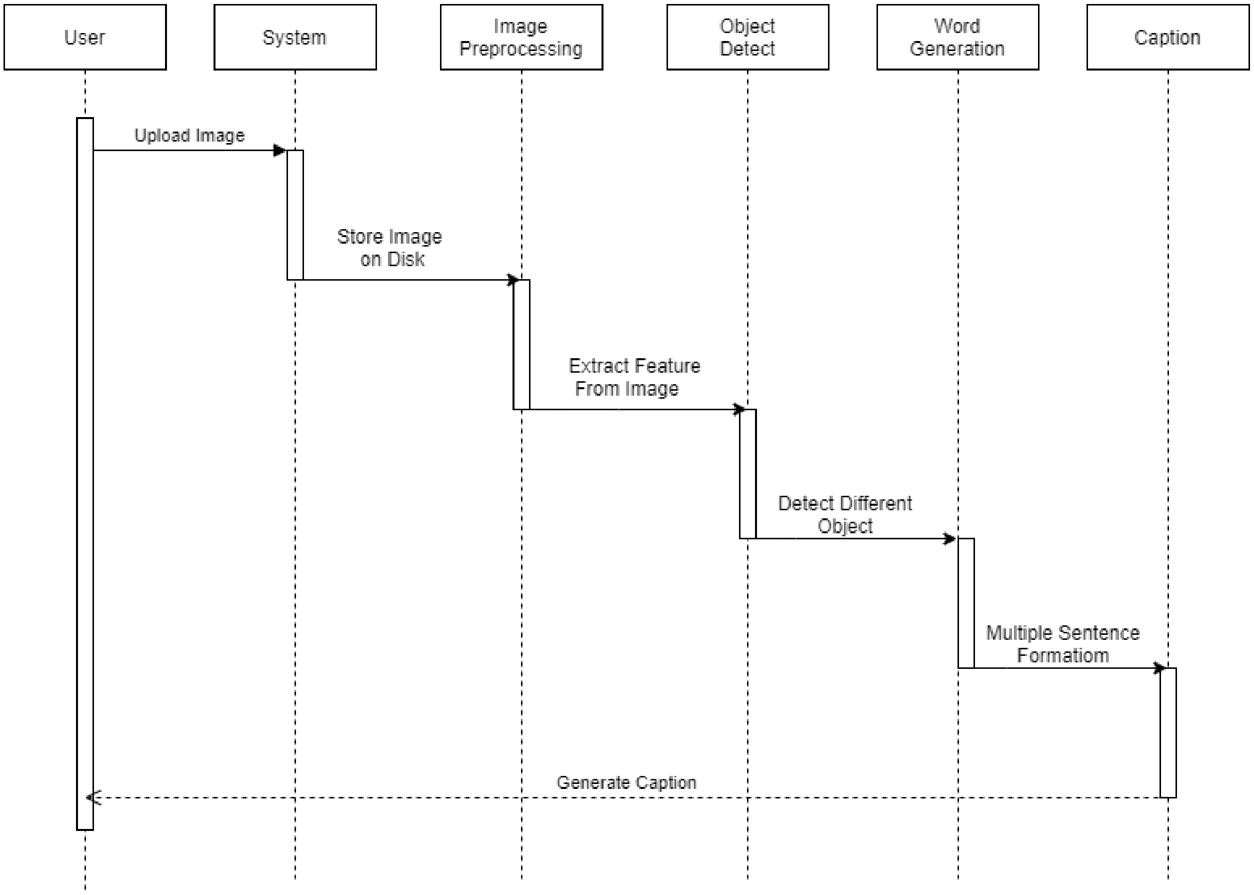
USE CASE DIAGRAM:



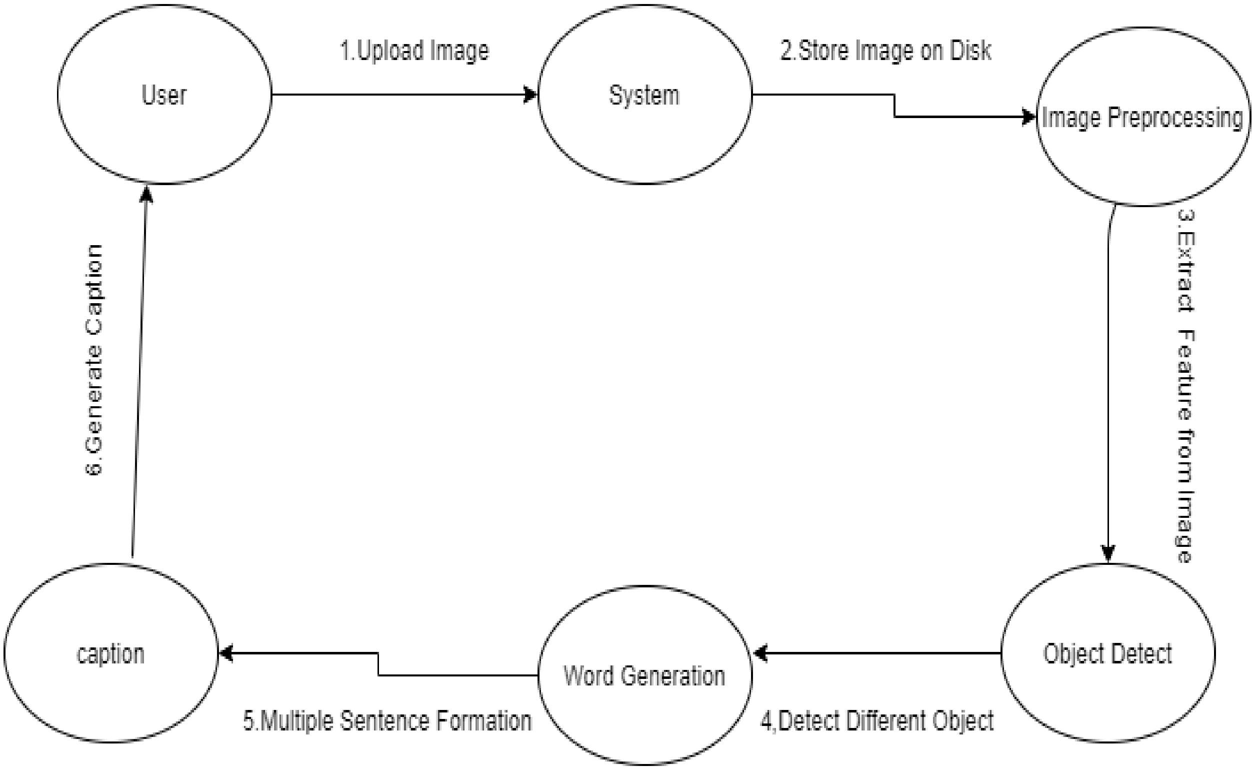
CLASS DIAGRAM:



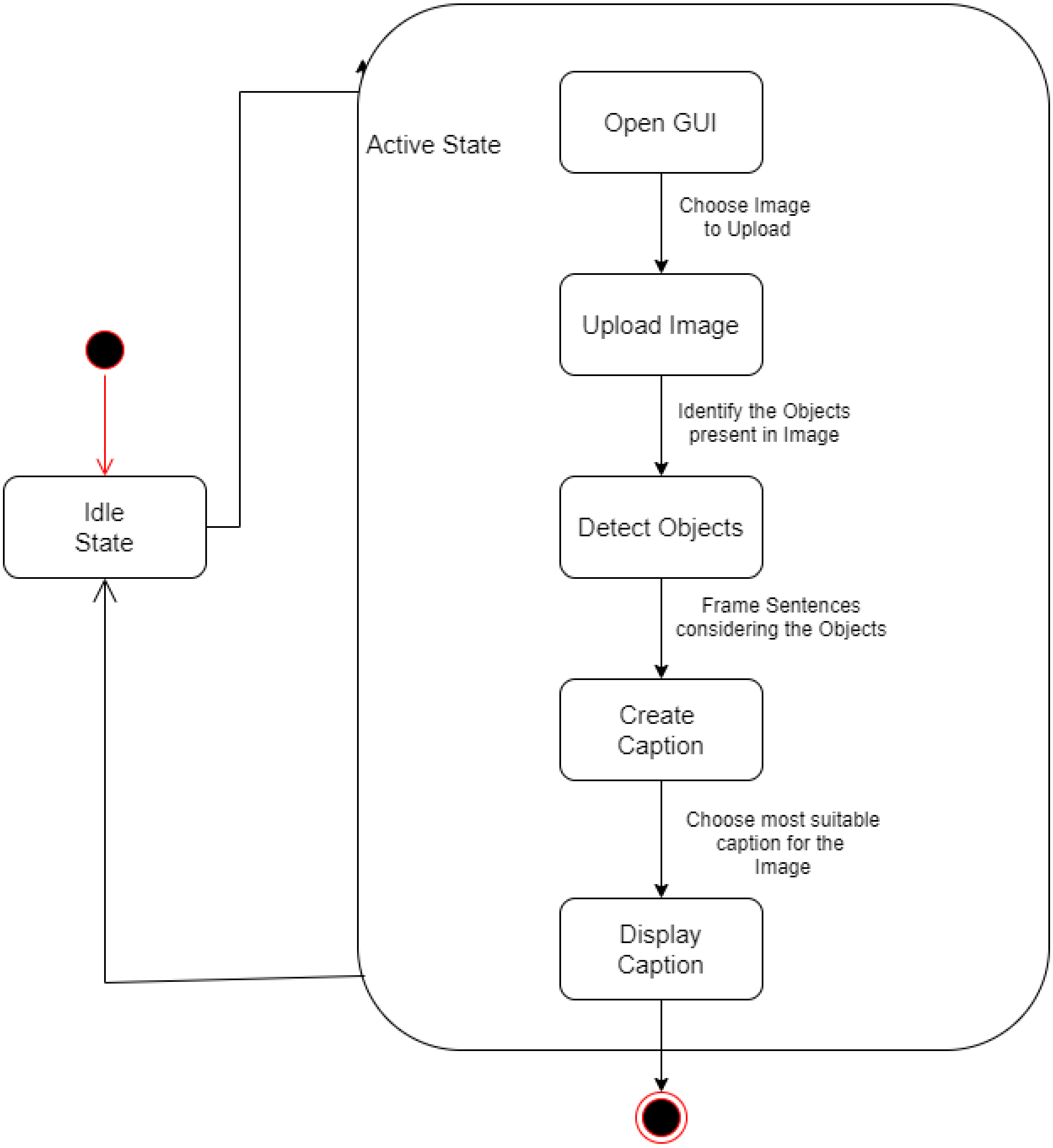
SEQUENCE DIAGRAM:



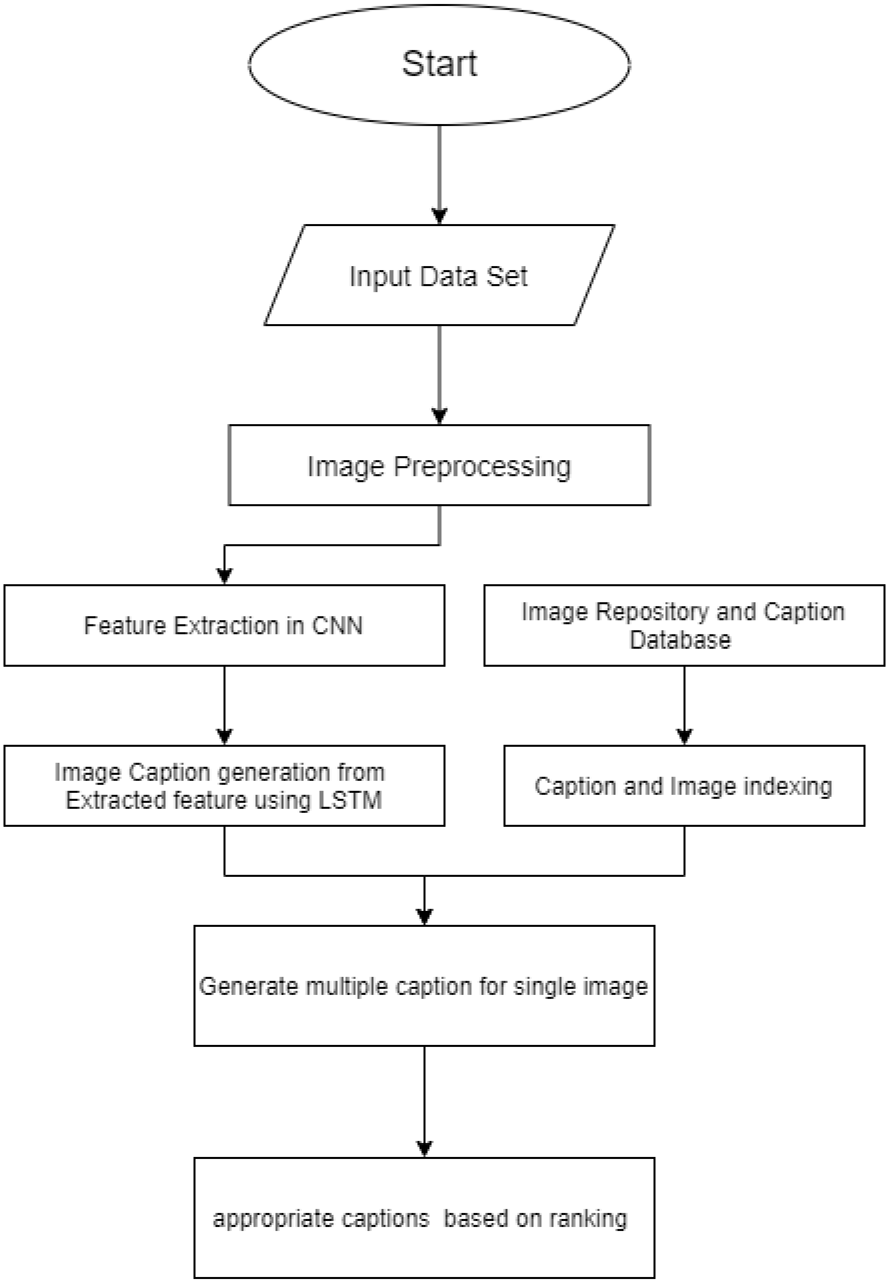
Collaboration Diagram:



State chart Diagram:



ACTIVITY DIAGRAM:


Start
Input Data Set
Image Preprocessing
Feature Extraction in CNN
Image Repository and Caption Database
Image Caption generation from Extracted feature using LSTM
Caption and Image indexing
Generate multiple caption for single image
appropriate captions based on ranking

Implementation

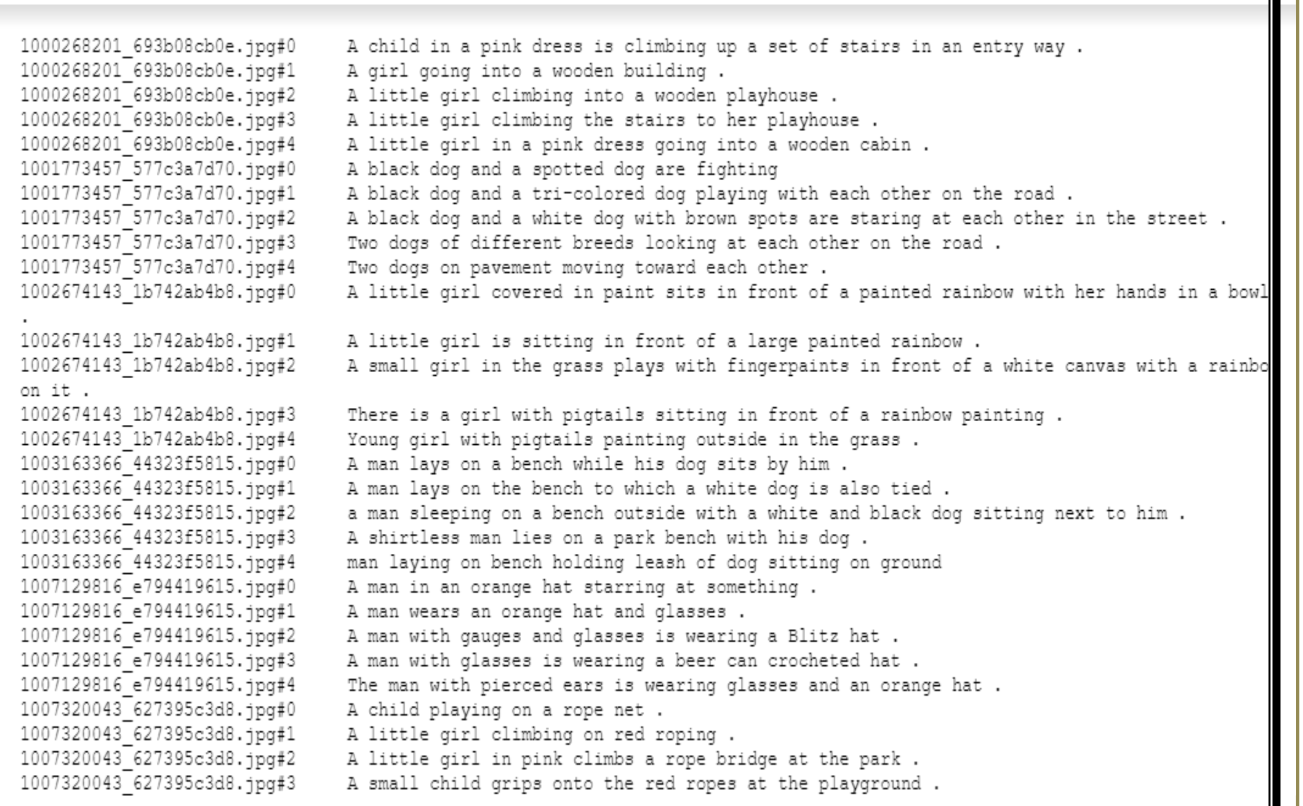
1. Data Collection

There are several open-source datasets available for this issue, including Flickr 8k (with 8k images), Flickr 30k (with 30k images), MS COCO (with 180k images) and so on.

We have used Flickr 8k dataset for this project, which contains 8092 images with 5 captions each. These images are divided as follows:

Training Images - 6000
Dev Images - 1092
Test Image - 1000

The "Flickr8k.token.txt" file contains image id for each image with 5 captions for each image. The file contents are as follows:



```
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 street .
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 hands in a bowl
.
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 with a rainbo
on it .
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 him .
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 .
1007320043_627395c3d8.jpg#1     A little girl climbing on red roping .
1007320043_627395c3d8.jpg#2     A little girl in pink climbs a rope bridge at the park .
1007320043_627395c3d8.jpg#3     A small child grips onto the red ropes at the playground .
```

2. Data Cleaning

When we deal with text, we generally perform some basic cleaning like lower-casing all the words (otherwise “hello” and “Hello” will be regarded as two separate words), removing special tokens (like ‘%’, ‘$’, ‘#’, etc.), removes words that contain numbers (such as ‘hey199’, etc.). Create a vocabulary of all the unique words present across all the 8092*5 (i.e., 40460) image captions (corpus) in the data set.

This means we have 8763 unique words across all the 40460 image captions. We write all these captions and the ids of their pictures in a file and save them to disk. However, if we think about it, many of these words will occur very few times, say 1, 2 or 3 times. Since we are creating a predictive model, we would not like to have all the words present in our vocabulary but the words which are more likely to occur or which are common. This helps the model become more robust to outliers and make less mistakes.

3. Loading the Training set

The text file "Flickr_8k.trainImages.txt" contains the names of the images that belong to the training set. Thus, we isolated 6000 training images from the list. Now, we will load the description of these images from the file into the Python dictionary (saved on the hard disk). However, when we load them, we add two tokens to each caption as follows (importance explained later):
'Start' -> This is the start caption token, which is added at the beginning of each caption.
'Stop' -> This is the end caption token that is added at the end of each caption.

4. Data Preprocessing - Images

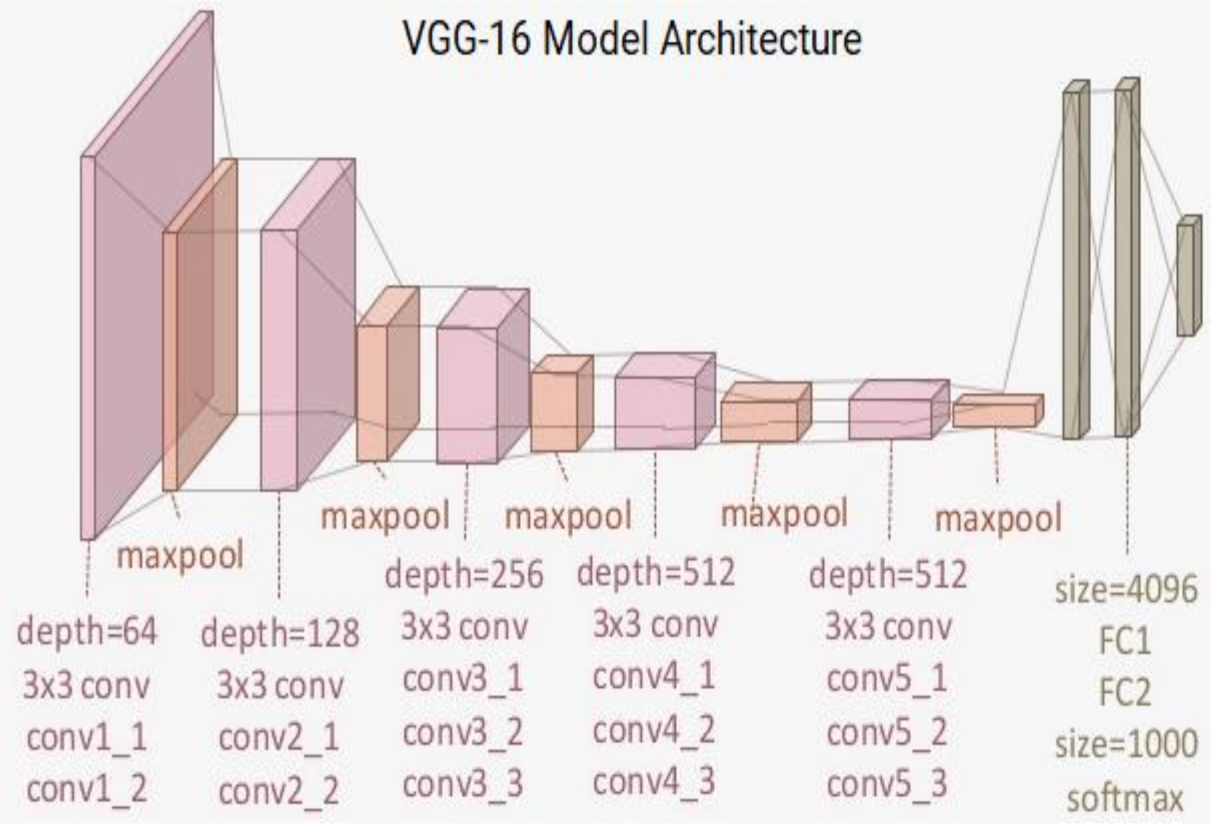
Images are nothing more than input (x) in our model. As you already know the model must give any input vector.

We need to convert each image to a fixed-size vector that can be fed into the neural network. We use a pre-trained model to understand the content of photographs. There are so many models to choose from. In this case, we will use the Oxford Visual Geometry Group or VGG model that won the ImageNet competition in 2014. You can pre-compute "photo features" and save them to a file using a pre-trained model. As a description of the photos given in the dataset we can load these features later and feed them on our model. This is no different than driving a photo through a full VGG model; We had already done that once.

This is an optimization that will make training our models faster and consume less memory.

We can load the VGG model in Keras using the VGG class. Since this is the model used to estimate the classification of the photo, we will remove the last layer from the loaded model. We are not interested in categorizing images, but we are interested in the internal representation of photos before classification takes place. These are the "features" taken from the model photo

Keras also provides tools to resize the loaded image to the desired size for the model (e.g., 3 channel 224 x 224-pixel image). Image Features 1 Dimensional 4,096 Element Vectors.



5. Data Preprocessing — Captions

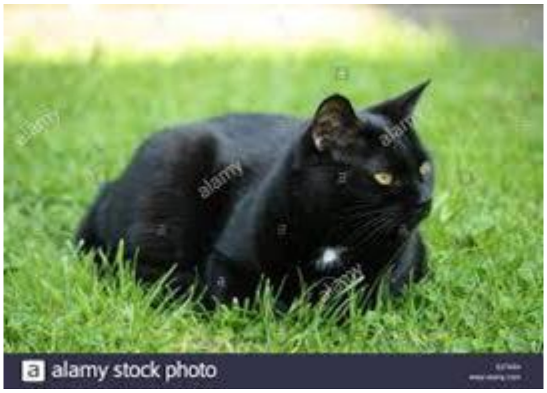
We should note that the caption is something we want to follow. So, during the training period, the caption is the target variable (Y) that the learner learns to evaluate the model.

But if you look at the picture, the whole caption is not generated at once. We evaluate the word-by-word headings. In this way, we must encode each word into a vector of a certain size. However, this part will appear later when we look at the model design, but for now we will create two Python dictionaries, one word to index and the other to index to word. We denote each particular word in the vocabulary by an integer. The maximum length of any caption is 37.

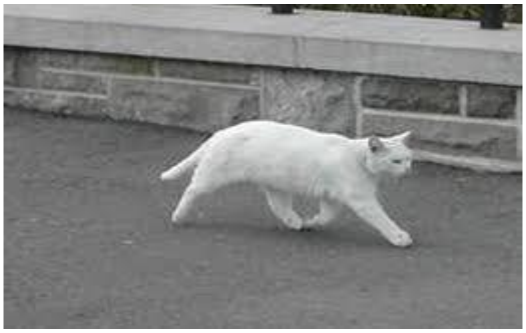
6. Data Preparation using Generator Function

This is one of the important stages of the case study. Here we will understand how to make the data in a way that is convenient to give as input to the in-depth learning model.

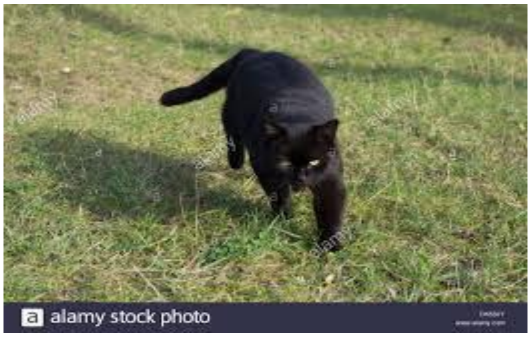
Consider that we have 3 images and their 3 related caption as follows :



(Train image 1) Caption -> The black cat sat on grass



(Train image 2) Caption -> The white cat is walking on road



(Test image) Caption -> The black cat is walking on grass

We use the first two images and their captions to train the model and the third image to test our model. First we need to convert both the images to their corresponding feature vector as discussed above. Let "Image_1" and "Image_2" be the feature vectors of the first two images respectively. Secondly, let's build the vocabulary for the first two (train) captions by adding the two tokens "start" and "stop" in both of them.

Caption_1 -> "start the black cat sat on grass stop"

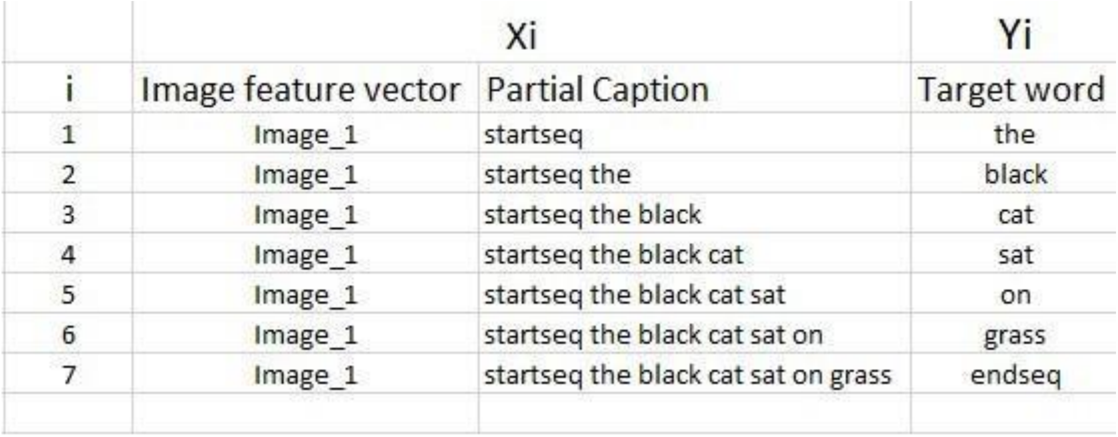
Caption_2 -> "start the white cat is walking on road stop"

vocab = {black, cat, stop, grass, is, on, road, sat, start, the, walking, white}

Give an index to each word in the vocabulary:

black -1, cat -2, stop -3, grass -4, is -5, on -6, road -7, sat -8, start -9, the -10, walking -11, white -12

To frame it as a supervised learning problem where we have a set of data points D = {Xi, Yi}, where Xi is the feature vector of data point 'i' and Yi is the corresponding target variable.



	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq

Data points corresponding to one image and its caption

Take the first image vector Image_1 and its corresponding caption "start the black cat sat on grass stop". Image vector is the input and the caption is what we need to predict. But the way we predict the caption is as follows:

For the first time, we provide the image vector and the first word as input and try to predict the second word,

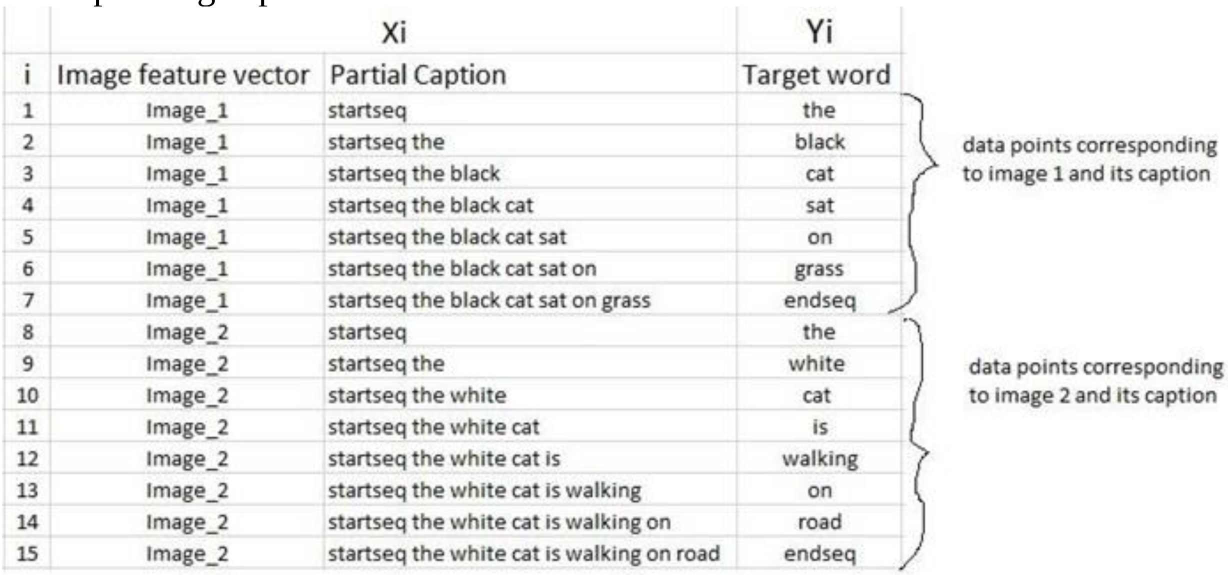
i.e.:

Input = Image_1 + 'start'; Output = 'the'

Then we provide image vector and the first two words as input and try

to predict the third word, i.e.:
Input = Image_1 + ‘start the’; Output = ‘cat’
And so on…

Thus, we can summarize the data matrix for one image and its corresponding caption as follows:



	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq
8	Image_2	startseq	the
9	Image_2	startseq the	white
10	Image_2	startseq the white	cat
11	Image_2	startseq the white cat	is
12	Image_2	startseq the white cat is	walking
13	Image_2	startseq the white cat is walking	on
14	Image_2	startseq the white cat is walking on	road
15	Image_2	startseq the white cat is walking on road	endseq

data points corresponding to image 1 and its caption

data points corresponding to image 2 and its caption

Data Matrix for both the images and captions

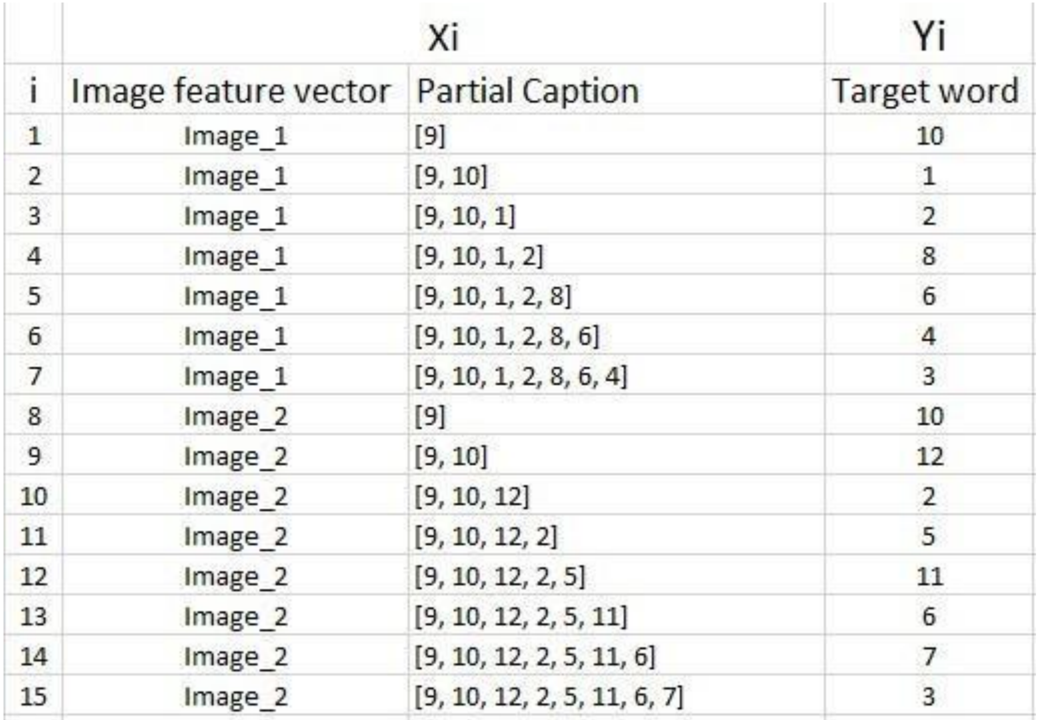
It must be noted that one image + caption is not a single data point but multiple data points depending on the length of the caption.

We must now understand that in every data point, it’s not just the image which goes as input to the system, but also, a partial caption which helps to predict the next word in the sequence.

Since we are processing sequences, we will employ a Recurrent Neural Network to read these partial captions.

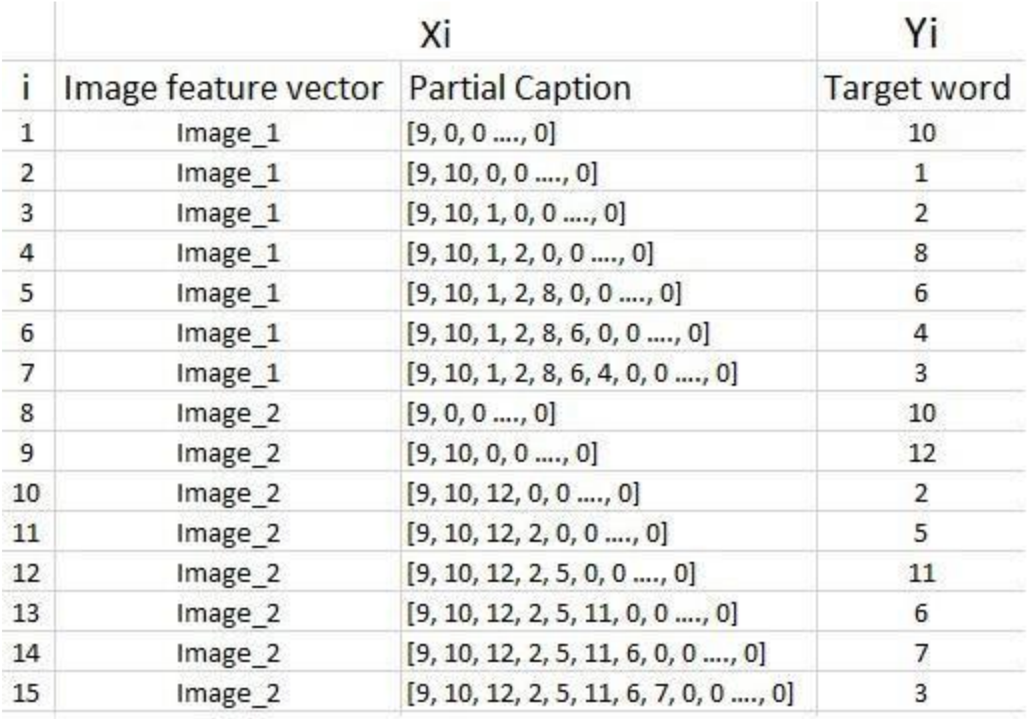
However, we have already discussed that we are not going to pass the actual English text of the caption, rather we are going to pass the sequence of indices where each index represents a unique word.

Since we have already created an index for each word, let’s now replace the words with their indices and understand how the data matrix will look like:



	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9]	10
2	Image_1	[9, 10]	1
3	Image_1	[9, 10, 1]	2
4	Image_1	[9, 10, 1, 2]	8
5	Image_1	[9, 10, 1, 2, 8]	6
6	Image_1	[9, 10, 1, 2, 8, 6]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4]	3
8	Image_2	[9]	10
9	Image_2	[9, 10]	12
10	Image_2	[9, 10, 12]	2
11	Image_2	[9, 10, 12, 2]	5
12	Image_2	[9, 10, 12, 2, 5]	11
13	Image_2	[9, 10, 12, 2, 5, 11]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7]	3

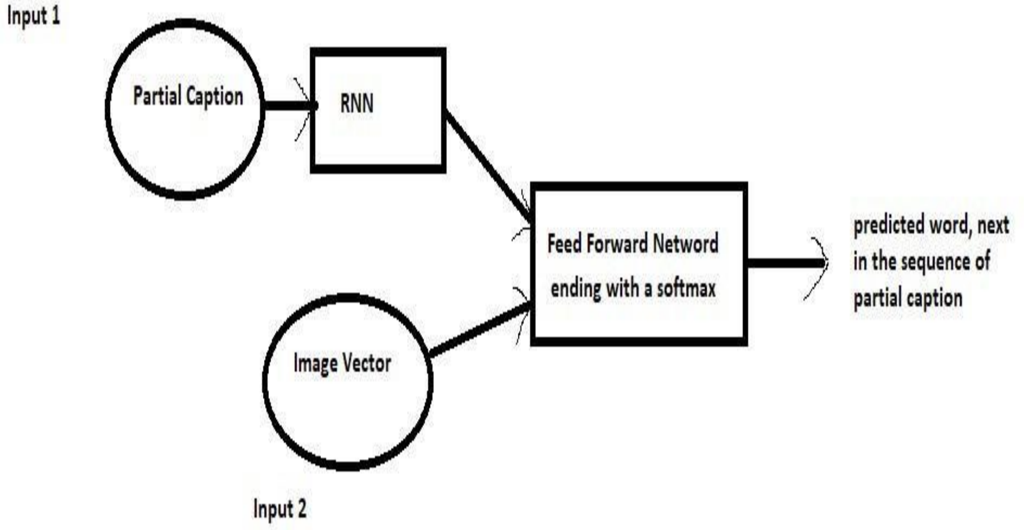
Since we would be doing batch processing, we need to make sure that each sequence is of equal length. Hence we need to append 0’s (zero padding) at the end of each sequence. We had calculated the maximum length of a caption, which is 37. So we will append those many numbers of zeros which will lead to every sequence having a length of 37.



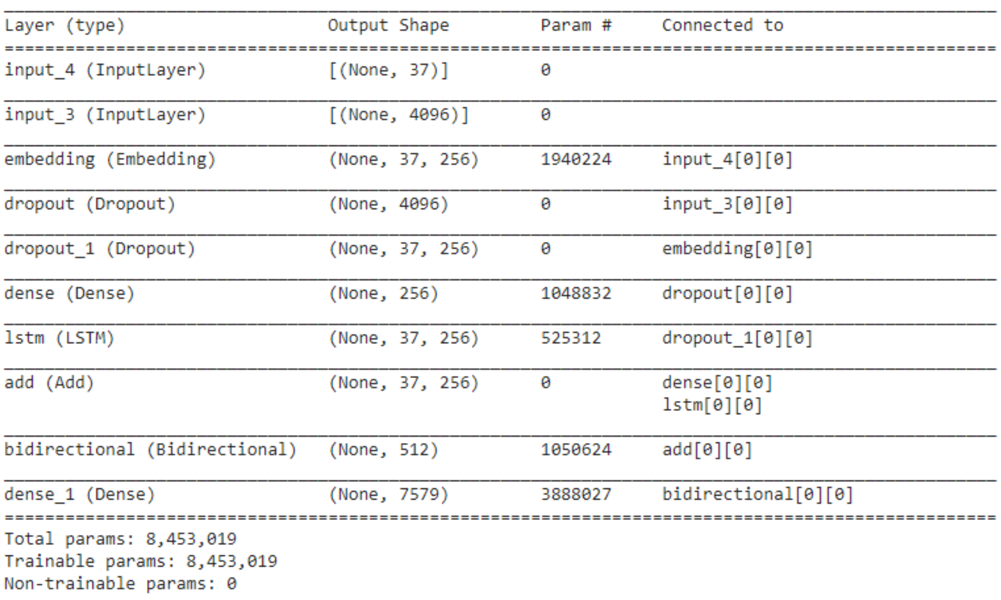
	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9, 0, 0 ...., 0]	10
2	Image_1	[9, 10, 0, 0 ...., 0]	1
3	Image_1	[9, 10, 1, 0, 0 ...., 0]	2
4	Image_1	[9, 10, 1, 2, 0, 0 ...., 0]	8
5	Image_1	[9, 10, 1, 2, 8, 0, 0 ...., 0]	6
6	Image_1	[9, 10, 1, 2, 8, 6, 0, 0 ...., 0]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4, 0, 0 ...., 0]	3
8	Image_2	[9, 0, 0 ...., 0]	10
9	Image_2	[9, 10, 0, 0 ...., 0]	12
10	Image_2	[9, 10, 12, 0, 0 ...., 0]	2
11	Image_2	[9, 10, 12, 2, 0, 0 ...., 0]	5
12	Image_2	[9, 10, 12, 2, 5, 0, 0 ...., 0]	11
13	Image_2	[9, 10, 12, 2, 5, 11, 0, 0 ...., 0]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6, 0, 0 ...., 0]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7, 0, 0 ...., 0]	3

7. Model Architecture

The input consists of two parts, the image vector and the partial caption, we cannot use the Sequential API provided by the keras Library. For this reason, we use functional APIs that allow us to create integrated models. First, let's look at a concise structure of high-level sub-modules,



High level architecture



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_4 (InputLayer)            [(None, 37)]         0
__________________________________________________________________________________________________
input_3 (InputLayer)            [(None, 4096)]       0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 37, 256)      1940224     input_4[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 4096)         0           input_3[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 37, 256)      0           embedding[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 256)          1048832     dropout[0][0]
__________________________________________________________________________________________________
lstm (LSTM)                     (None, 37, 256)      525312      dropout_1[0][0]
__________________________________________________________________________________________________
add (Add)                       (None, 37, 256)      0           dense[0][0]
                                                                 lstm[0][0]
__________________________________________________________________________________________________
bidirectional (Bidirectional)   (None, 512)          1050624     add[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 7579)         3888027     bidirectional[0][0]
==================================================================================================
Total params: 8,453,019
Trainable params: 8,453,019
Non-trainable params: 0
```

Model Summary

The LSTM (long-term short-term memory) layer is nothing more than a repetitive neural network dedicated to processing sequence inputs (in our case partial caption). The weight of the model is updated by the backpropagation algorithm and the model learns to output a word, image feature vector and partial caption. So, in a nutshell, we have:

Input_1 -> Partial Caption
Input_2 -> Image feature vector
Output -> An appropriate word, next in the sequence of partial caption provided in the input_1 (or in probability terms we say conditioned on image vector and the partial caption)

Testing

Test case No	Test Case	Input	Expected Output	Actual Output	Status
01	Upload Image	Image	Image Upload	Image Uploaded	Pass
02	Object Detection from Image	Image	Object detected from image	Object detected from image	Pass
03	Recognize action performed in image	Image	Action recognized from image	Action recognized from image	Pass
04	Sentence formation	Five descriptions for image	Sentence formed	Sentence formed	Pass
05	Sentence Should be grammatically correct	Five descriptions for image	Grammatically correct sentence should be formed	Sentence is grammatically correct for 60% images	Pass
06	Sentence Formation	Single description for image	Single description formation	Single description formed	Pass

Performance Analysis

The primary focus while training the data is to ensure that the value of the loss function does not increase rapidly or remain stagnant with each epoch but display a gradual decrease. In order to monitor the performance, we specially monitor the training loss.

A) Training Loss:

We have trained the model using 6000 image data with 20 epochs. For each epoch, we noted the corresponding loss value and it was found to reduce with each epoch. The plot for model loss with each epoch is given below:

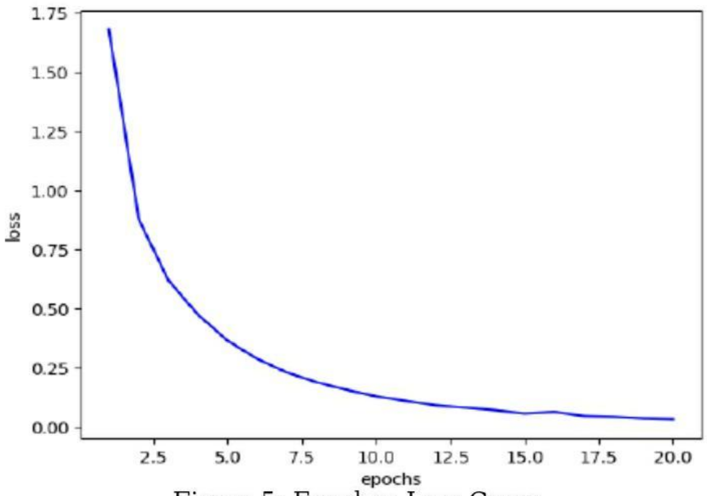


Figure 5: Epoch vs Loss Curve

B) BLEU Performance:

Following the model training we determine the accuracy of the model to generate the respective caption. As discussed in the section, we are using the BLEU metric to determine the accuracy of word generation. We are testing the text generation procedure using 4gram scores. Each gram corresponds to different weights. Following are the observations made on performing BLEU evaluation.

N-gram	**WEIGHTS**	**SCORE**
BLEU-1	1.0,1.0,1.0,1.0	79.0190
BLEU-2	0.5,0.5,0,0	52.8925
BLEU-3	0.3,0.3,0.3,0	37.4354
BLEU-4	0.25,0.25,0.25,0.25	18.3087

APPLICATIONS

- Self-driving cars - Automatic driving is one of the biggest challenges and if we can properly caption the scene around the car, it will boost the self-driving system.
- CCTV cameras are everywhere today, but in addition to watching the world, if we can create relevant captions, we can sound the alarm as soon as malicious activity occurs. This may help reduce some crime or accidents.
- It helps to improve Google image search in the same way as Google search. The image can be captioned first and then searched based on its sentence.
- Picasa: Using facial recognition to identify your friends and yourself in a group picture.
- Google Photos: Categorize your photos into mountains, seas, etc.

ETHICS

Declaration of Ethics:

As A Computer Science & Engineering Student, I believe it is Unethical To,

1. Surf the internet for personal interest and non-class related purposes during classes

2. Make a copy of software for personal or commercial use

3. Make a copy of software for a friend

4. Loan CDs of software to friends

5. Download pirated software from the internet

6. Distribute pirated software from the internet

7. Buy software with a single user license and then install it on multiple Computers

8. Share a pirated copy of software

9. Install a pirated copy of software

Cost Estimation

No.	Equipment	Price
1	Computer System	Rs.40000/-
2	Gradient Paper space (Cloud GPU)	Rs.3000/-
3	Internet Connection	Rs.1500/-
	Total=	44500/-

REFERENCES

1. Jyoti Aneja, Aditya Deshpande, Alexander Schwing, Convolutional Image Captioning, [Online] Available: https://arxiv.org/pdf/1711.09151.pdf

2. Andrej Karpathy, Li Fei-Fei, Deep VisualSemantic Alignments for Generating Image Descriptions, [Online] Available: https://cs.stanford.edu/people/karpathy/cvpr2015.pdf

3. https://stackoverflow.com/questions/tagged/face-captioning

4. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database