

**A PROJECT REPORT  
ON**

**“ASSISTIVE VISION: IMAGE CAPTION GENERATOR”**

**SUBMITTED TO  
SHIVAJI UNIVERSITY, KOLHAPUR**

**IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE  
AWARD OF DEGREE**

**BACHELOR OF TECHNOLOGY IN INFORMATION  
TECHNOLOGY AND ENGINEERING**

**SUBMITTED BY**

<b>Ms. Mrunali Bhojraj Lipte</b>	<b>17UIT11027XX</b>
<b>Ms. Anjali Shahaji Patil</b>	<b>17UIT11042XX</b>
<b>Ms. Mrunali Rajendra Patil</b>	<b>17UIT11044XX</b>
<b>Ms. Snehal Maruti Powar</b>	<b>17UIT51048XX</b>
<b>Mr. Ruturaj Amar Mudholkar</b>	<b>17UIT72005XX</b>

**UNDER THE GUIDANCE OF**

**Prof. A. S. SHELAR**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
AND ENGINEERING**

**D.K.T.E. SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE,  
ICHALKARANJI**

**(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)**

**Accredited with 'A+' Grade by NAAC, An ISO 9001: 2015 Certified YEAR 2020-  
2021**

**YEAR 2020-2021**

**D.K.T.E. SOCIETY'S  
TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI**

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

Accredited with 'A+' Grade by NAAC, An ISO 9001: 2015 Certified

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**AND ENGINEERING**



**CERTIFICATE**

**This is to certify that, project work entitled  
“ASSISTIVE VISION: IMAGE CAPTION GENERATOR”**

**is a bonafide record of project work carried out by**

<b>Ms. Mrunali Bhojraj Lipte</b>	<b>17UIT11027XX</b>
<b>Ms. Anjali Shahaji Patil</b>	<b>17UIT11042XX</b>
<b>Ms. Mrunali Rajendra Patil</b>	<b>17UIT11044XX</b>
<b>Ms. Snehal Maruti Powar</b>	<b>17UIT51048XX</b>
<b>Mr. Ruturaj Amar Mudholkar</b>	<b>17UIT72005XX</b>

**In the partial fulfillment of award of degree, Bachelor of Technology in Computer Science and Engineering prescribed by Shivaji University, Kolhapur for the academic year 2020-2021**

**Prof. A.S. SHELAR**  
**[Project Guide]**

**Prof. ( Dr.) D.V.KODAVADE**  
**(H.O.D. C.S.E.)**

**Prof.(Dr.) P.V.KADOLE**  
**(DIRECTOR)**

**EXAMINER**

# DECLARATION

We hereby declare that, the project work report entitled “**ASSISTIVE VISION: IMAGE CAPTION GENERATOR**” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.TECH.(IT). 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.

PRN	Name	Signature
17UIT11027XX	Ms. Mrunali Bhojraj Lipte	
17UIT11042XX	Ms. Anjali Shahaji Patil	
17UIT11044XX	Ms. Mrunali Rajendra Patil	
17UIT51048XX	Ms. Snehal Maruti Powar	
17UIT72005XX	Mr. Ruturaj Amar Mudholkar	

# ACKNOWLEDGMENT

With great pleasure we wish to express our deep sense of gratitude to prof. A. S. Shelar for his valuable guidelines, support and encouragement in completion of project report.

Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavde for his co-operation in preparing this project report.

We feel gratified to record our cordial thank to other staff members of Information Technology department for their support, help and assistance which they extended as and when required.

Thank you,

Ms. Mrunali Bhojraj Lipte	17UIT11027XX
Ms. Anjali Shahaji Patil	17UIT11042XX
Ms. Mrunali Rajendra Patil	17UIT11044XX
Ms. Snehal Maruti Powar	17UIT51048XX
Ms. Ruturaj Amar Mudholkar	17UIT72005XX

# ABSTRACT

In the past few years, the problem of generating descriptive sentences automatically for images has garnered a rising interest in natural language processing and computer vision research. Image captioning is a fundamental task which requires semantic understanding of images and the ability of generating description sentences with proper and correct structure. In this study, the authors propose a hybrid system employing the use of multilayer Convolutional Neural Network (CNN) to generate vocabulary describing the images and a Long Short-Term Memory (LSTM) to accurately structure meaningful sentences using the generated keywords. The convolutional neural network compares the target image to a large dataset of training images, then generates an accurate description using the trained captions

Image caption generator aims to automatically generate a sentence description for an image. The model will take an image as input and generate an English sentence as output, describing the contents of the image. The capturing mechanism involves a tedious task that collaborates both image processing and computer vision. The model is based on a deep recurrent architecture that combines recent advances in computer vision and machine translation and that can be used to generate natural sentence describing an image.

# INDEX

## **1. INTRODUCTION**

## **2. PROBLEM DESCRIPTION**

- a. Problem definition
- b. Aim and objective of the project
- c. Scope and limitation of the project
- d. Timeline of the project

## **3. BACKGROUND STUDY AND LITERATURE REVIEW**

- a. Literature overview
- b. Investigation of current project and related work

## **4. REQUIREMENT ANALYSIS**

- a. System requirement
- b. Functional requirement
- c. Analysis

## **5. SYSTEM DESIGN**

- a. Architectural Design
- b. Data Design
  - 1. Data flow diagram
  - 2. Sequence diagram
  - 3. Activity diagram
  - 4. Deployment diagram

## **6. IMPLEMENTATION**

- a. Detailed Description of Method
- b. Methodology

**7. INTEGRATION AND TESTING**

**8. PERFORMANCE ANALYSIS**

**9. APPLICATION**

**10. INSTALLATION GUIDE AND USER MANUAL**

**11. ETHICS**

**12. CONCLUSION**

**13. REFERENCE**

# INTRODUCTION

An Image caption Generator may be popular research area of computer science that deals with image understanding and a language description for that image. Generating well formed sentences requires both syntactic and the semantic understanding of the language. Having the ability to explain the content of the picture using accurately formed sentences may be a very challenging task, but it could even have a good impact, by helping the visually impaired people for better understand the content of images.

This task is significantly harder compared to the image classification or object recognition tasks that are well researched.

The biggest challenge is most definitely having the ability to form an outline that has got to capture not only the objects contained in an image, but also express how these objects relate to each other.



**a. Problem definition:**

Design and Develop a system, which generate the description of an image using Convolutional Neural Network (CNN)

**b. Aim and objectives of the Project:**

- In this project, we'll take a glance at a noteworthy multi modal topic where it'll combine both image and text processing to create a useful Deep Learning application, Image Captioning.
- Image Captioning refers to the method of generating textual description from a picture – supported the objects and actions within the image.
- To involves computer vision and language processing concepts to acknowledge the context of a picture and describe them in a natural language.
- To provide an assistive vision to the blind people.
- To address the complexity of image captioning and decoding it with accuracy.

**c. Scope and limitation of the Project:****➤ Scope**

The “**ASSISTIVE VISION: IMAGE CAPTION GENERATOR**” gives the caption as a output.

The system has wide scope in various fields like Image Search Tools, Guidance Devices, Self-Driving Cars, Artificial intelligence.

**➤ Limitations**

The main goal is to create a web application which covers all the functions of image description and provides a great interface for Digital assistant to the user.

A Digital assistant help the user to give answer to his questions which might be given in speech form as a command.

#### **d. TimeLine of Project**

In this project, we were used classic life cycle paradigm also called Water Fall Model. In the software engineering which is the sequential approach to the software development which begins at the system level and proceed through analysis, design, coding, testing and maintenance. We had completed software requirement analysis by the mid of October 27 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 coding stage.

After completion of coding stage, the important part in the software development which is testing phase carried out in first 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 module tested in unit testing. After unit testing the module prepared was cross checked with the design.

<b>TOPIC</b>	<b>START DATE</b>	<b>END DATE</b>
Domain Selection	17 Aug 2020	20 Aug 2020
Domain Finalization	23 Aug 2020	27 Aug 2020
Selection of Problem Statement	4 Sept 2020	15 Sept 2020
Finalization of Problem Statement	22 Sept 2020	23 Sept 2020
Study on Research Paper	25 Sept 2020	1 Oct 2020
Documentation of Synopsis	3 Oct 2020	9 Oct 2020
Requirement Analysis	13 Oct 2020	17 Oct 2020
System Requirement	1 Nov 2020	4 Nov 2020
Module Identification	7 Nov 2020	20 Nov 2020
System Architecture	22 Nov 2020	25 Nov 2020
Implementation 25%	1 Dec 2020	15 Dec 2020
Testing 25%	18 Dec 2020	30 Dec 2020
Implementation 50%	19 Feb 2021	25 March 2021
Testing 50%	1 April 2021	16 April 2021
Implementation 75%	1 May 2021	5 May 2021
Testing 75%	5 May 2021	7 May 2021
Implementation 100%	8 May 2021	12 May 2021
Testing 100%	12 May 2021	13 May 2021
Report Making	1 Jan 2021	12 May 2021

# **BACKGROUND STUDY AND LITERATURE OVERREVIEW**

## 2. Literature Overview

### a. Literature Overview:

There are no Extra Required for image Classification in CNN. All you need is and Python compiling IDE. The software which is User friendly and Easy to use is SPYDER3. You should have the required packages installed some of the Modules/Packages like Keras, TensorFlow

We found some of the literature papers based on image caption generator that uses some technologies and devices that aims to this project.

Some of the literatures are as follows-

- Baby Talk: For understanding and generating the straightforward image description. (Girish Kulkarni, Visruth Premraj, Vicente Ordenz) (2011)
- Oriol Vinyals: Alexander Toshev, Samy Bengio, Dumitru Erhan), Show and Tell. An outline must capture not only the objects contained in a picture, but it also must express that how these objects associated with one another and also their attributes as well as the activities that they're involved in. (2015)
- A Real-Time Image Caption Generator: Created a first mobile device-based application to introduce possible use cases to perform well in real time and maintain to object high quality caption (Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra, Kumar Bansode) (2017)

**b. Investigation of current Project and Related work:**

Deep neural networks have considerable success in varied tasks in text, speech and image domains. Till today, there are lots of techniques used for recognizing emotion from text. Variations of Recurrent Neural Networks, such as Long Short-Term Memory (LSTM) and Bidirectional LSTM have been effective in modeling sequential information.

This work presents a model, which is a neural network that can automatically view an image and generate appropriate captions in natural language like English. The model is trained to produce the sentence or description from given image. The descriptions or captions obtained from the model are categorized into: Description without errors, Description somewhat related to image.

# **REQUIREMENT ANALYSIS**

## **a. System Requirements**

### **1. Software and Hardware requirements:**

- **Hardware Requirements**

- ✓ Processor (Intel Core i7-10700T 10th GEN)
- ✓ PCIe x4 NVME SSD (For faster performance)
- ✓ GPU - Quadro 2000 (CUDA enabled, capability - 2.1)
- ✓ RAM 16 GB (Intel Core i7-10700T 10<sup>th</sup> GEN)

- **Software requirements:**

- ✓ Windows 10 64-bit
- ✓ x64-based processor

- **Application or Tools and Technologies**

- ✓ Python 3.9.0
- ✓ Libraries - TensorFlow, Keras
- ✓ Python compiling IDE



## b. Functional Requirements

- Data Pre-processing

For making the information in much usable format Data Pre-processing includes the functions as tokenization which is that the task of chopping up data into pieces, called tokens and at the identical time discard certain characters, like punctuation. Then words are reduced to a root by removing unnecessary characters, usually a suffix called as stemming. Then we are going to remove stop words, means frequent words like “the”, “is”, etc. that don’t have specific semantic.

- Designing model

After pre-processing of data we will move towards classification. The system is passed to the RNN classifier, which includes dependent activation. It will reduce the complexity of increasing parameters. It memorizes the output of each previous layer & gives as input to next hidden layer. For long term dependencies we are dealing with LSTM’s, which control the flow and mixing of inputs as per trained weights. So, it will give us most control ability and thus better results

- Import libraries and dataset

Importing required libraries like TensorFlow, keras, matplotlib. Also require Loading and cleaning the text dataset, generated the vocabulary and Loading the image dataset. So, it will give us Cleaned text dataset, mapped text to image.

- Feature vector extraction

For this feature extraction we use Input (Human Understandable) Image. Then it covert the image into an encoding so that machine can understand pattern in it. So, it will give us Encoded Image.

- Tokenizing and glove vector embedding

We use Input as words in vocabulary to tokenizing all words in vocabulary using keras tokenizer and storing the relations between words in our vocabulary using an embedding matrix. So, it will give us Tokenized word

- Defining and training the model.

We use Input Untrained model and it will define the model using keras model from functional API and Train the by decreasing batch size and increasing number of epochs. So, it will give us Trained model

- Predicting the output.

As we give input as Image and it model will ready to predict caption for any given image.

## c. Analysis

Analyses of different methods are done in this project. Their procedures and drawbacks are explained as follows.

### 1.Convolutional Neural Network:

Convolutional Neural networks are specialized deep neural networks which processes the information that has input shape such as a 2D matrix. CNN works well with pictures and can be easily represented as a 2D matrix. Image classification and identification are often easily done using CNN. It can determine whether a picture could be a bird, a plane or Superman, etc. Important features of an image can be extracted by scanning the image from left to right and top to bottom and finally the features are combined together to classify images. It can deal with the pictures that have been transformed, rotated and changes in perspective.

### 2. Long Short-Term Memory

LSTM are variety of RNN (recurrent neural network) which similar temperament for sequence prediction problems. We will predict what the next words and will be on the basis of previous text. It's shown itself effective from the standard RNN by overcoming the constraints of RNN. LSTM can do relevant information throughout the processing, it discards the non-relevant information.

### 3.Data Exploration

For the image caption generator, we've used the Flickr\_8K dataset. There are other big datasets like Flickr\_30K and MSCOCO dataset but it can take weeks for systems having only CPU support just to train the network, so we used a little Flickr8k dataset. Employing a huge dataset helps in developing a much better model

# **SYSTEM DESIGN**

## 4. System Design

### a. Architecture Diagram of System: -

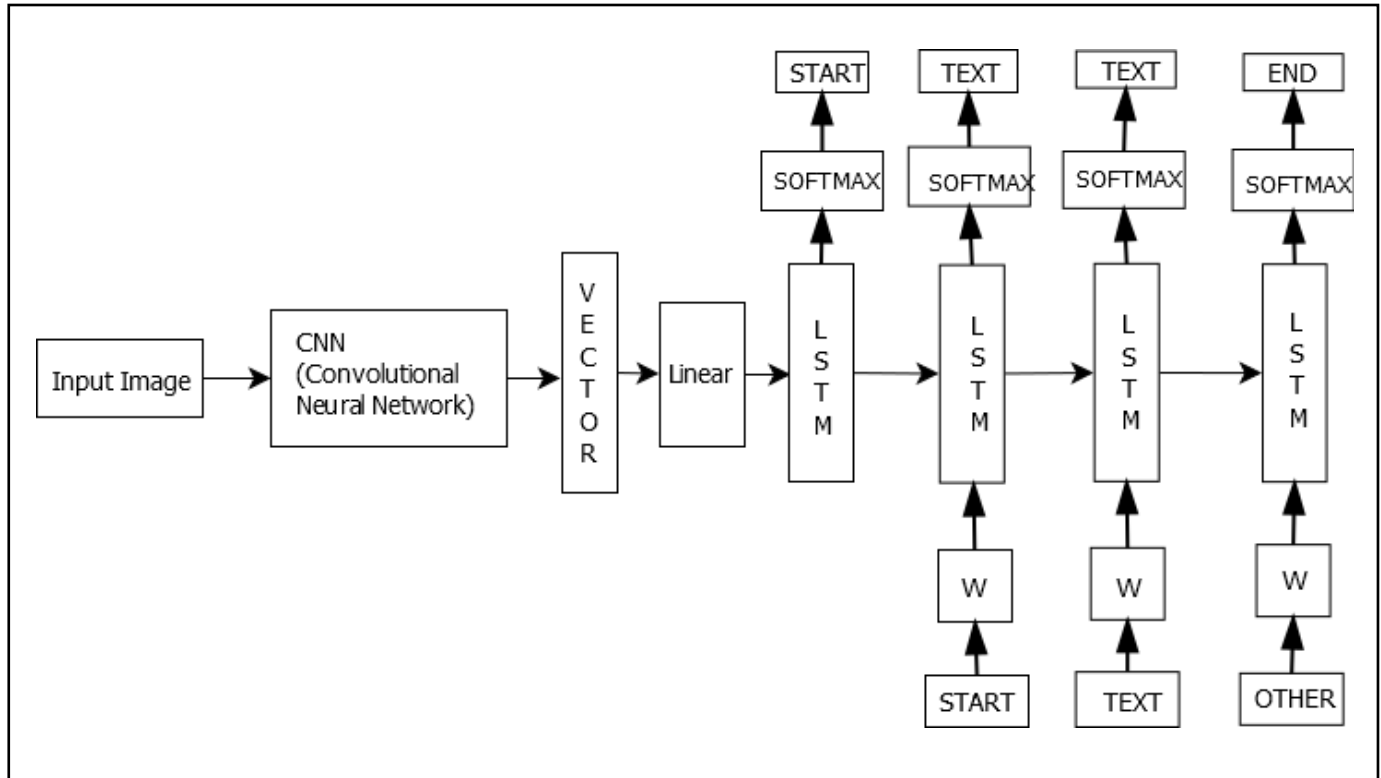


Fig: Architecture Diagram of System

This Architecture Diagram elaborate overall structure of system. The main objectives of system.

#### Components in Architecture diagram

In Architecture Diagram, A four modules of system are presented with required component. An input image, CNN, vector, LSTM etc. components are required.

## Flow Chart Diagram

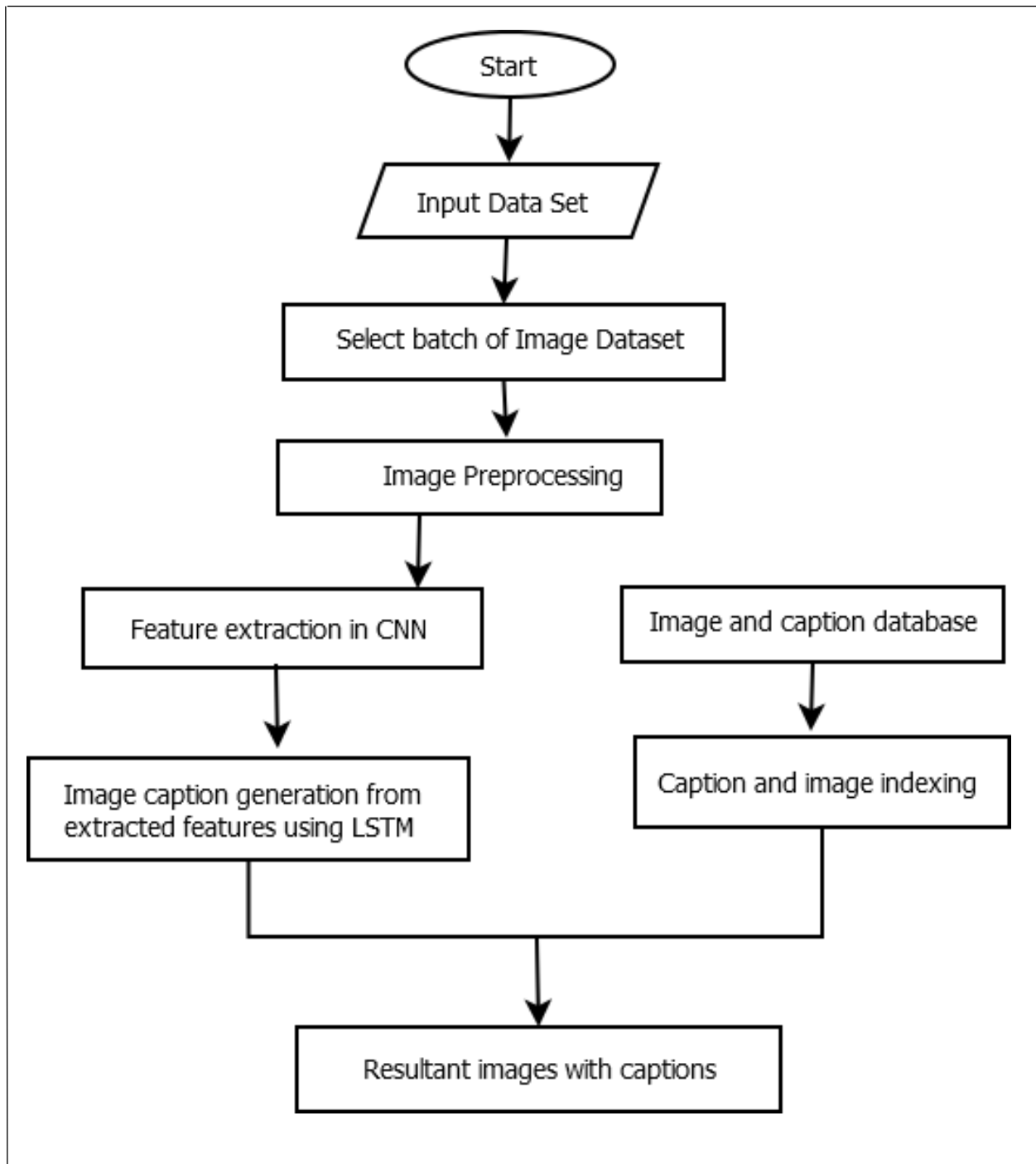


Fig: Flow chart Diagram

## Use Case Diagram

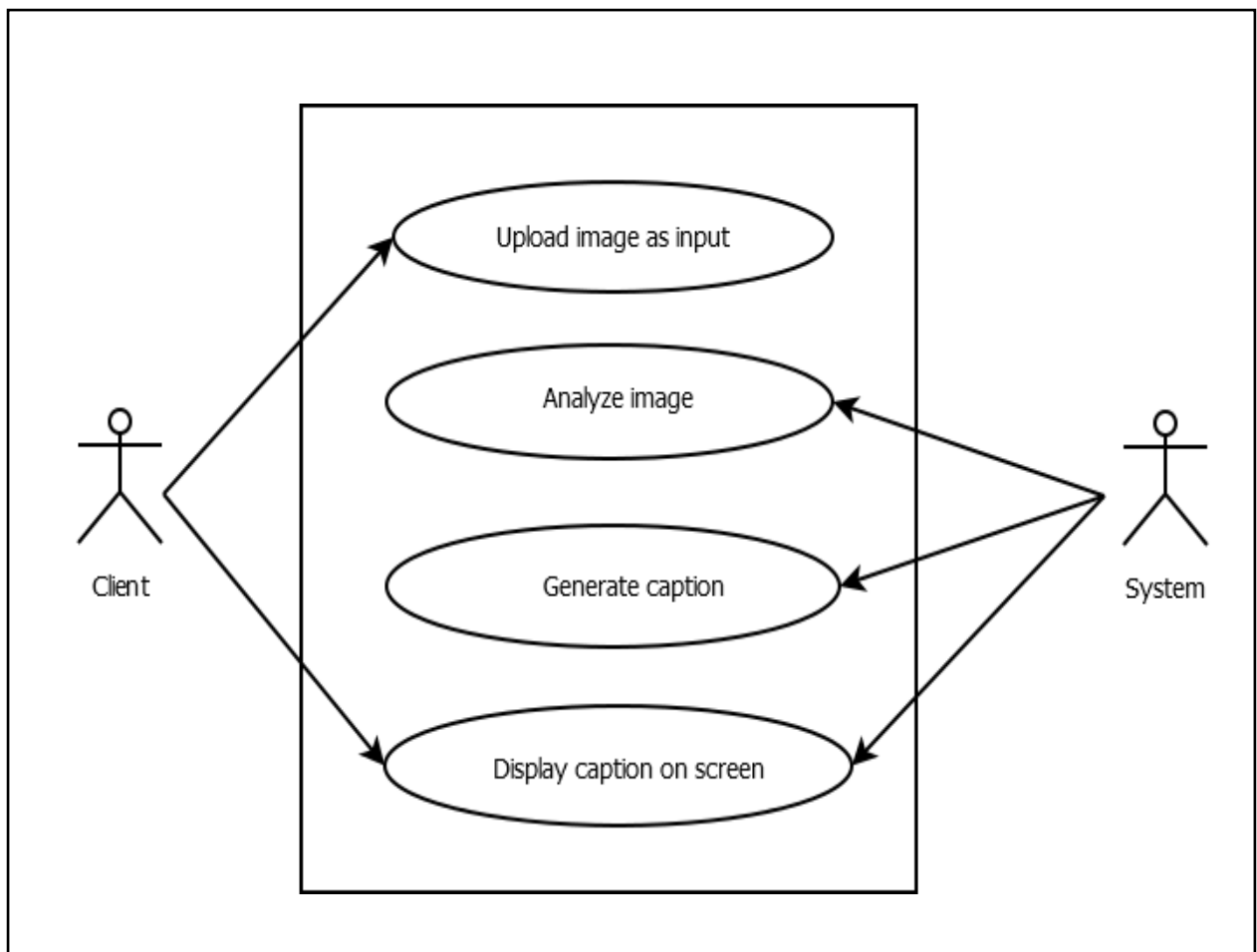


Fig: Use Case Diagram

### Components in Use Case Diagram:

As we see in above diagram our main functionalities are shown. In diagram one main user is there and another one is our machine learning model.

## b. Data Design

### a. Data Flow Diagram

#### DFD level 0

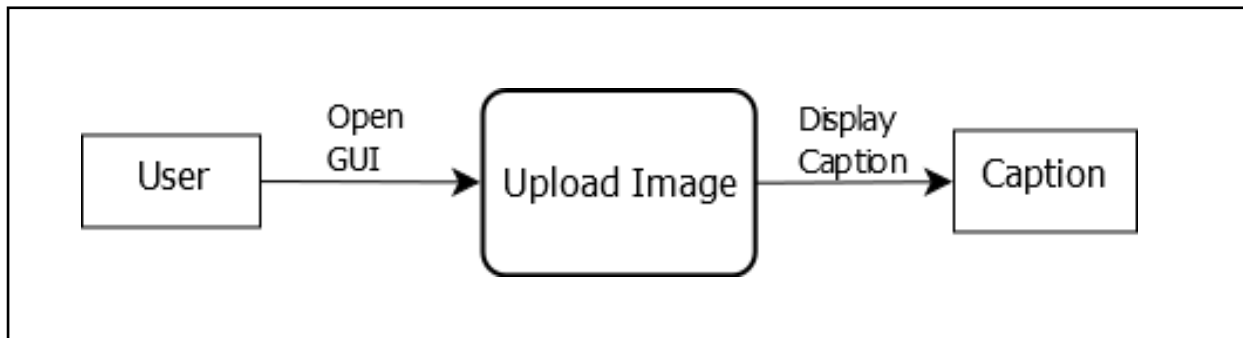


Fig: Diagram of DFD Level 0

As shown in diagram first the user gives the image and transfer that data which is input to system which then generate caption.

#### DFD level 1

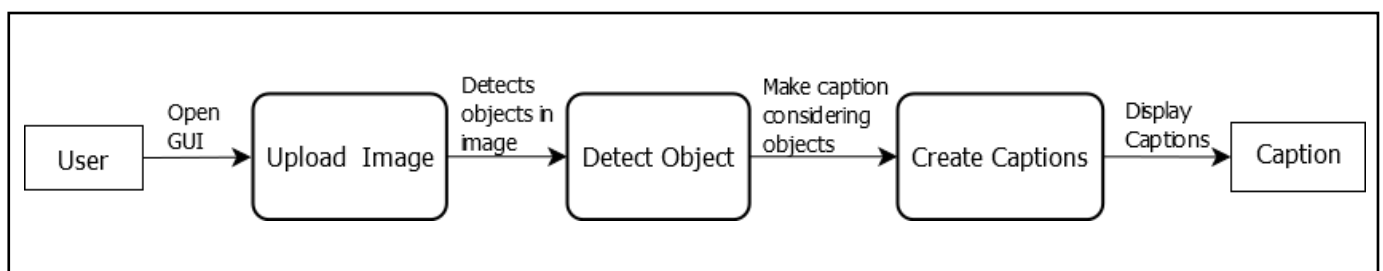


Fig: Diagram of DFD level 1

As shown in diagram the specific functionality of each module is given which performs their respective task



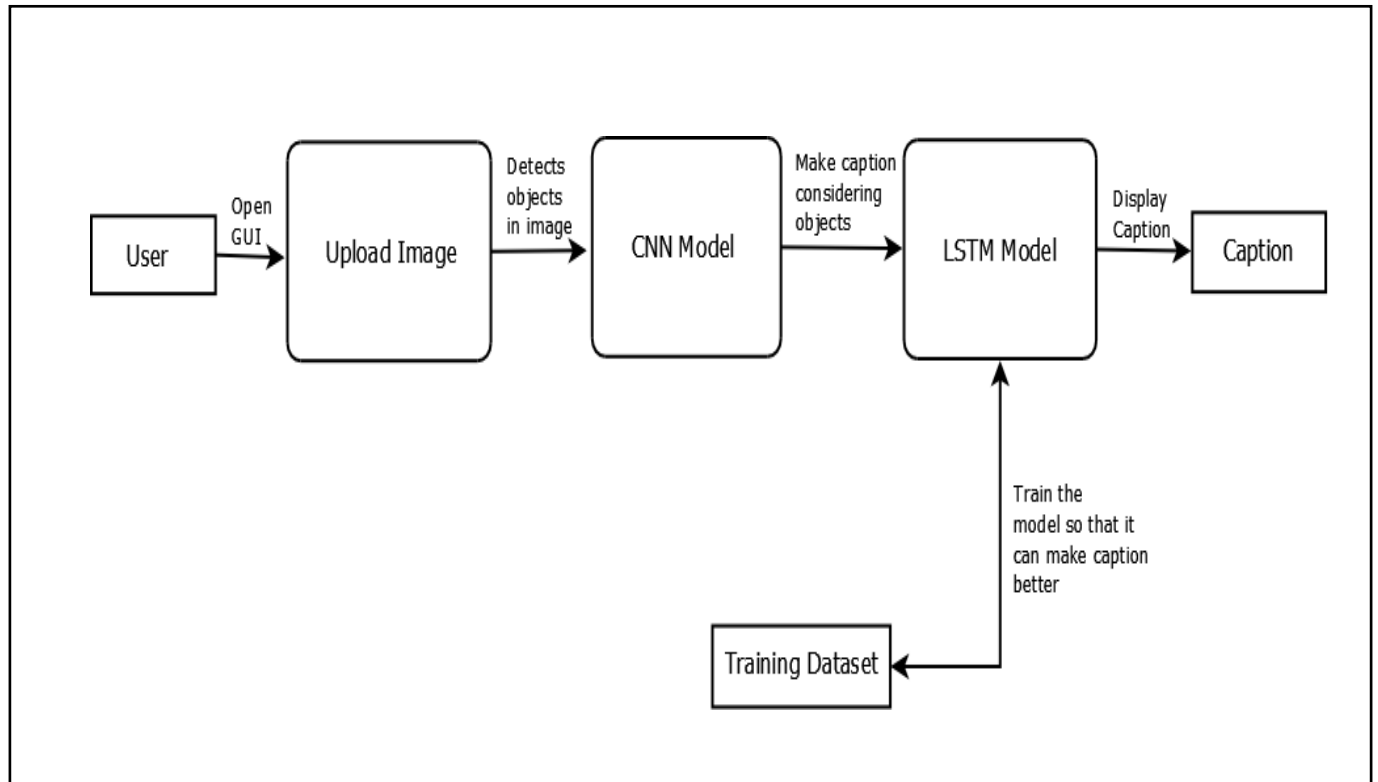
**DFD level 2**

Fig: Diagram of DFD level 2

As shown in diagram the specific functionality of each module is given which performs their respective task

### c. Sequence diagram

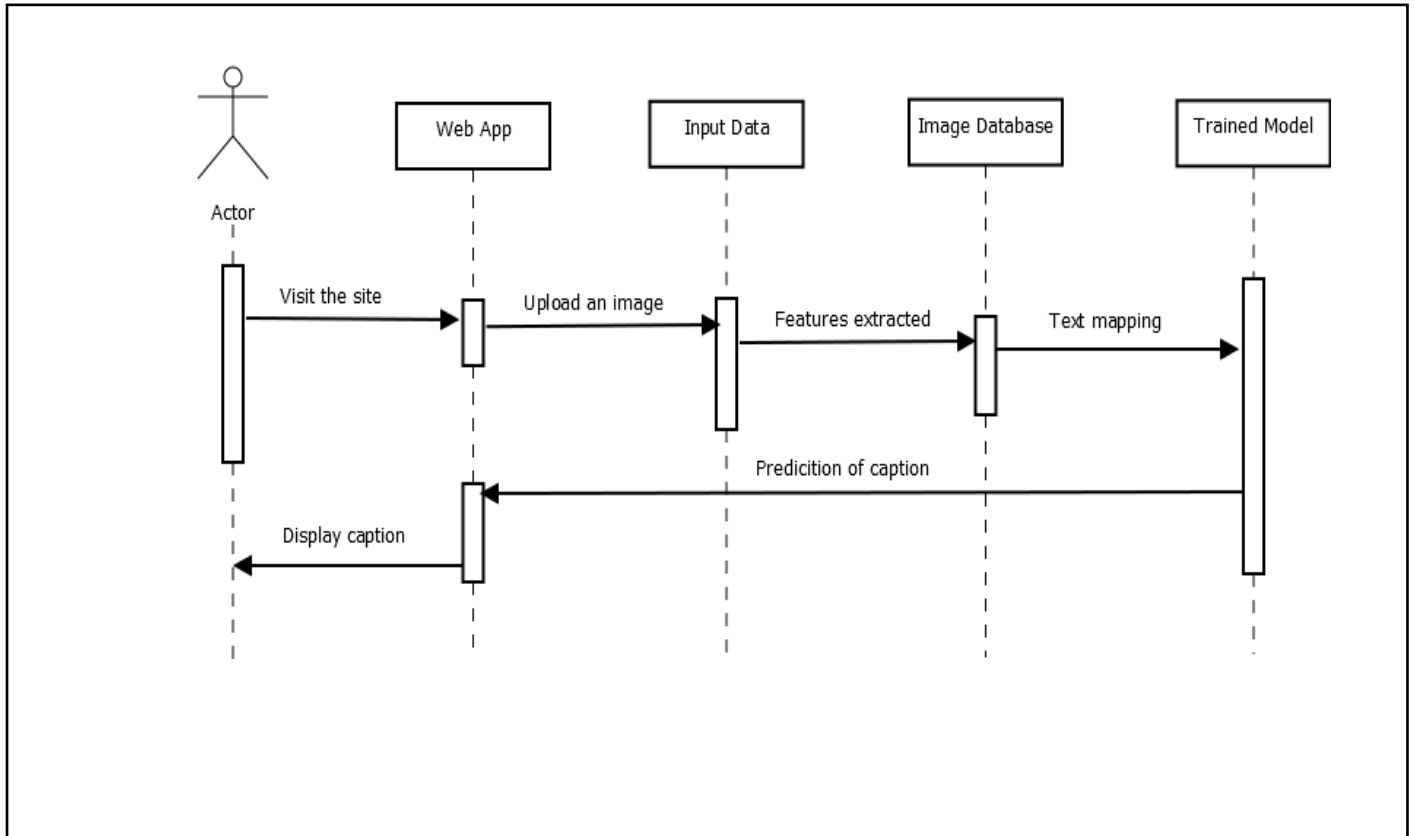


Fig: Sequence Diagram

#### Components in Sequence diagram:

As we can see in diagram the sequence of several activities is shown, which are carried out in specific manner.

# **IMPLEMENTATION**

## **a. Detailed Description of Methods**

The implementation of the model was done using the Python SciPy environment. Keras 2.0 was used to implement the deep learning model because of the presence of the VGG net which was used for the object identification. Tensorflow library is installed as a backend for the Keras framework for creating and training deep neural networks. TensorFlow could be deep learning library developed by Google. It provides heterogeneous platform for execution of algorithms i.e.; it may be run on low power devices like mobile additionally as large scale distributed system containing thousands of GPUs.

The neural network was trained on the Nvidia GeForce 1050 graphics processing unit which has 640 Cuda cores. In order to define structure of our network TensorFlow uses graph definition. Once graph is defined it can be executed on any supported devices. The photo features can be pre-computed using the pretrained model and also saved. These features are then loaded and them into our model because the interpretation of a given photo within the dataset to scale back the redundancy of running each photo through the network anytime, we would like to check a replacement language model configuration.

# Implementation-

## 1.Model Building

Our model will have 3 major steps:

1. Processing the sequence from the text.
2. Extracting the feature vector from the image.
3. Decoding the output using SoftMax by concatenating the layers.

## 2. Training the model

- To train the model, we will be using the training images.
- Our dataset has images and captions so we will create a function that can train the data in batches.
- 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 model's folder. This will take some time depending on our system capability.

## 3.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.

## **b.Methodology**

### **1. Preprocessing**

#### **A. Data Analysis**

Analysis of text involve:

a] Lexical Analysis- It will include identifying and analyzing the structure of words. Lexicon of a language means the gathering of words and phrases in a very language. Lexical analysis is dividing the full chunk of txt into paragraphs, sentences, and words

b] Syntactic Analysis- It includes the analysis of words within the sentence for grammar and arranging the words during a manner that shows the connection among the words. The sentence as “The school goes to boy” is get rejected by English syntactic analyzer

c] Semantic Analysis- It draws the precise meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It’s done by mapping syntactic structures and objects within the task domain.

#### **B. Cleaning-**

Cleaning involves removing URLs, Hashtags, Reserved words, emojis, and stop words.

## 2. LSTM Module

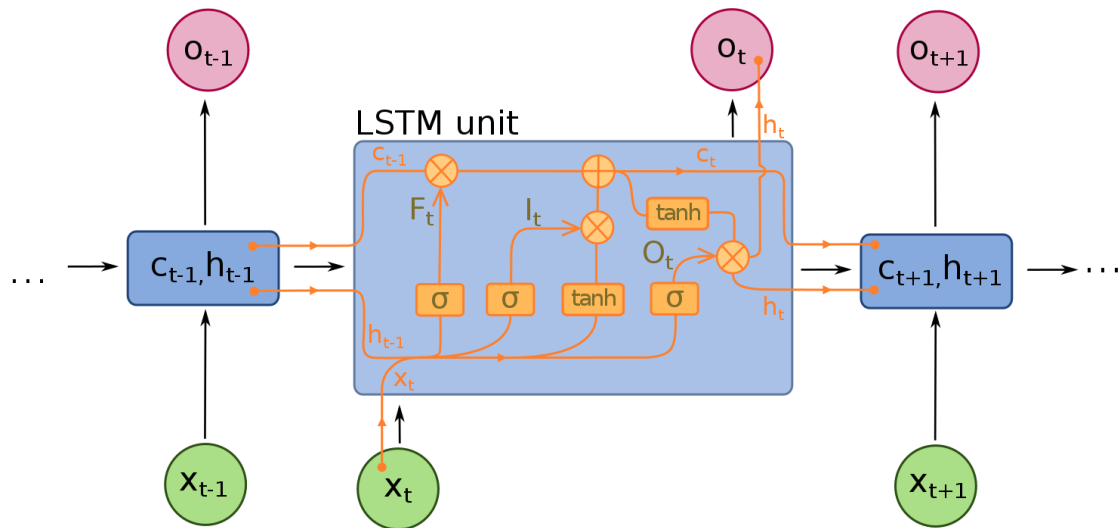
A recurrent neural network (RNN) could be a family of artificial neural networks which is specialized within the processing of sequential data. In contrast with traditional neural networks, RNNs are designed to accommodate sequential data by sharing their internal weights processing the sequence. For this purpose, the computation graph of RNNs includes cycles, representing the influence of the previous information on this one.

Long Short-Term Memory networks are a modified version of recurrent neural networks, which makes it easier to recollect past data in memory. The problem of RNN i.e vanishing gradient is resolved here. LSTM is similar temperament to classify process and predict statistic given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present.

**1.Input gate**—It discovers that which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0, 1. and tanh function gives weightage to the values which are passed deciding their level of importance ranging from -1 to 1.

**2.Forget gate**—It will discover that what details can be discarded from the block. It's decided by the sigmoid function. it looks at the previous state ( $h_{t-1}$ ) and therefore the content input ( $X_t$ ) and outputs variety between 0 (omit this) and 1 (keep this) for each number within the cell state  $C_t-1$ .

**3. Output gate**—the input and also the memory of the block is employed to determine the output. The sigmoid function will decide which values to let through 0 and 1. tanh would give weightage to the values which are passed deciding their level of importance ranging from -1 to 1 and multiplied with output of the Sigmoid.







# **INTEGRATION AND TESTING**

# INTEGRATION AND TESTING

## Unit test cases generation and its testing reports

Testcase	Input	Expected output	Status
1.		The small dogs play in the snow	pass
2.		Children are playing with football on grass	pass

# **PERFORMANCE ANALYSIS**

## Performance analysis:

### 1. Encoding-

After preprocessing we need to convert text into numerical vector. So machine learning model will understand it.

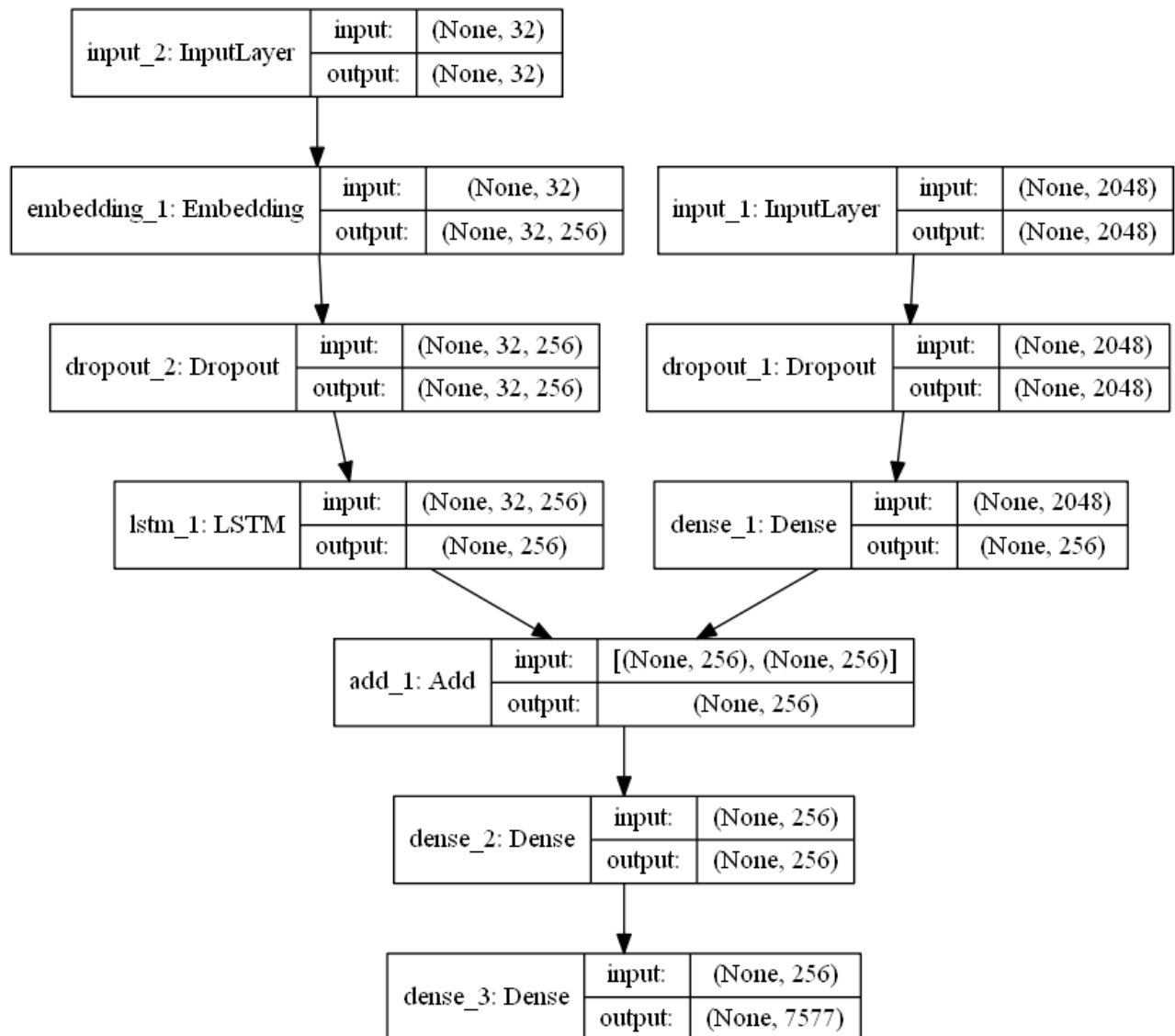
The input to the model are  $[x_1, x_2]$  and the output can be  $y$ , where the  $x_1$  is the 2048 feature vector of that picture,  $x_2$  is the input text sequence and  $y$  is the only output text sequence which the model will have to predict.

$x_1$ (feature vector)	$x_2$ (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

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

- **Feature Extractor:** The feature extracted from the picture has a size of 2048 and with the dense layer, we can reduce the dimensions to 256 nodes.
- **Sequence Processor:** An embedding layer can handle the textual input and then followed by the LSTM layer.
- **Decoder:** On merging the output from the above two layers, we can process the dense layer to make the final prediction. The last layer would include the number of nodes equal to the vocabulary size.

The visual representation of the final model will be given below –



# **APPLICATION**

# Applications:

## 1. Image Search Tools: -

Caption can be generated from image at first and search can be performed on the basis of caption that is generated from image.

## 2. Guidance Devices: -

Caption can be generated from image and converted into voice; these can be provided for blind people by guiding them without any support.

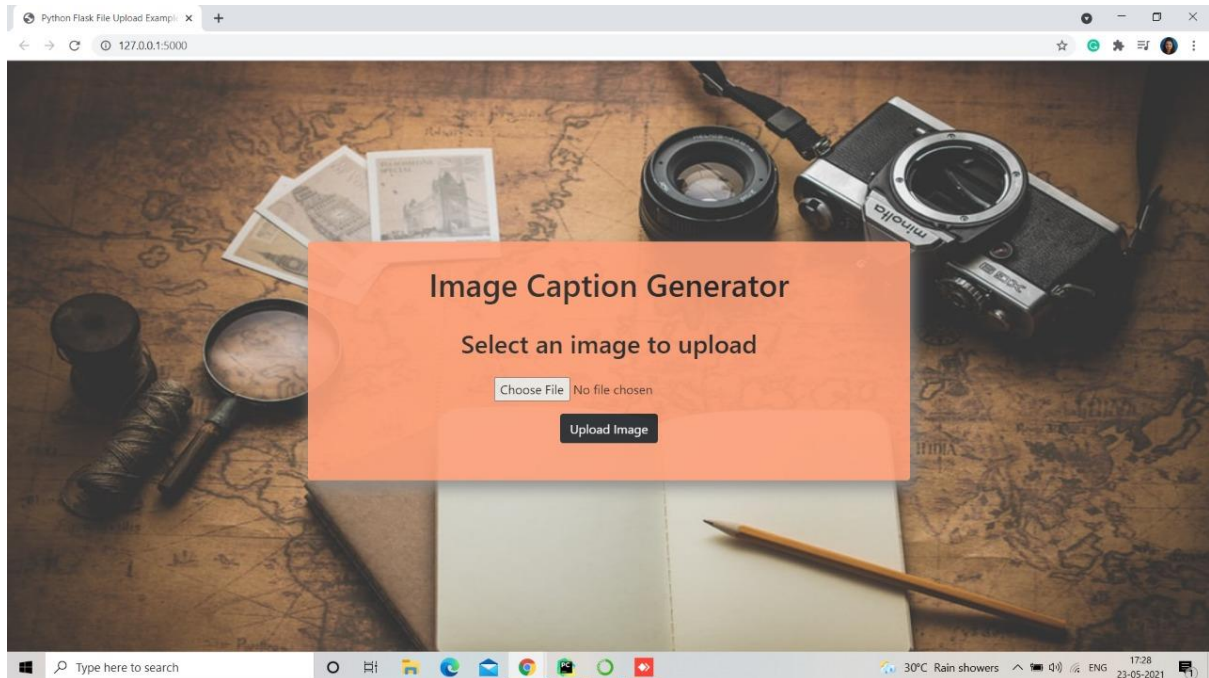
## 3. Self-Driving Cars: -

All self-drive cars are using image/video processing with neural network to attain their goal in order to boost self-driving.

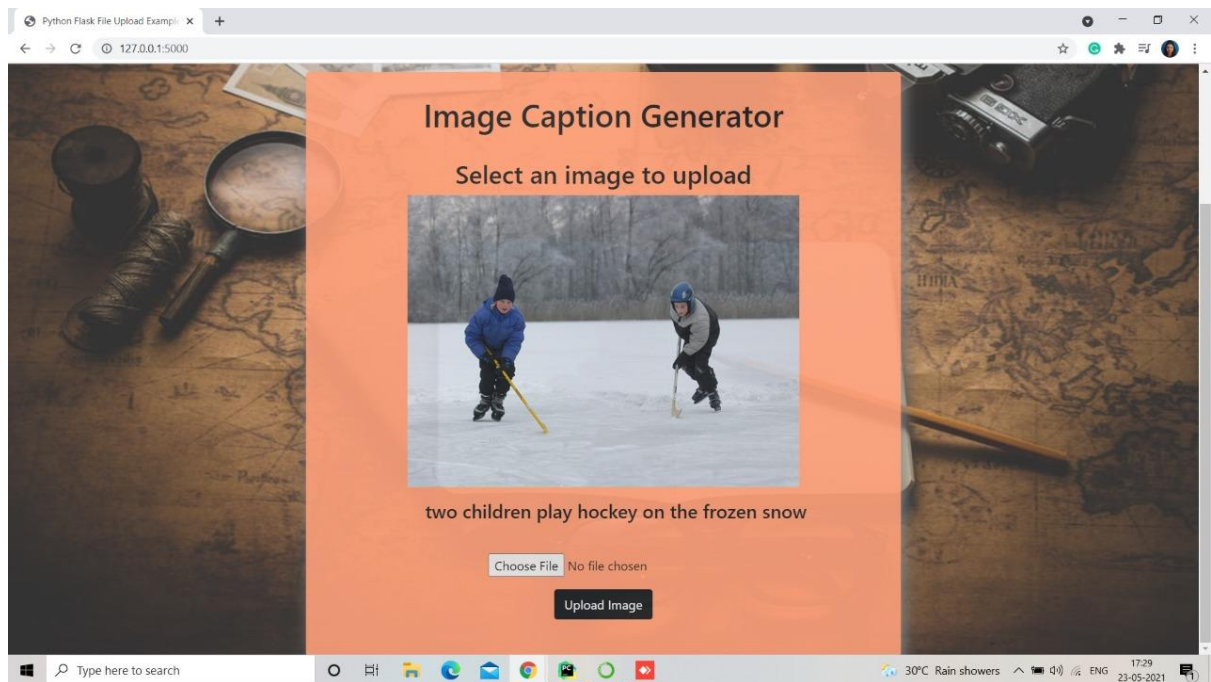
- Some of the applications of the project are as follows:
  - 1) Social Media Platforms like Facebook can infer directly from the image, where you are (beach, cafe etc.), what you wear (color) and more importantly what you're doing also (in a way).
  - 2) Digital Object Identification
  - 3) Used within the heavy vehicles and also in light vehicles

## • Website

### Upload Image



### Display Image





# **Installation Guide and User Manual**

## **Installation: -**

### **1) First install all required dataset.**

### **2) Install jupyter Notebook**

Use the following installation steps:

1. Download the Anaconda software. The recommended version for download Anaconda is latest Python 3 version (currently Python 3.5).
2. Install the latest version of Anaconda which downloaded, following the instructions on the download page.
3. Congratulations, you have installed Jupyter Notebook. To run the notebook:

### **3) Then install the Libraries.**

Make sure you could have already installed all the following required libraries:

- pip install TensorFlow
- keras
- pillow
- NumPy
- tqdm
- jupyterlab

#### 4) Project File Structure

Downloaded from the dataset:

- Flickr8k\_Dataset: Dataset folder that contains 8091 pictures.
- Flickr\_8k\_text: Dataset folder that contains text files and captions of the images.

The below files would be created while making the project.

- Models: It can contain the trained models.
- Descriptions.txt: This text file will include all image names and their captions after preprocessing.
- Features.p: The pickle object that contains a picture and their feature vector can be extracted from the Xception pre-trained CNN model.
- Tokenizer.p: It contains tokens which mapped with an index value.
- Model.png: The visual representation of the dimensions of the project.
- Testing\_caption\_generator.py: The Python file for generating the caption of every image.
- Training\_caption\_generator.ipynb: The Jupyter notebook within we would train and build the image caption generator.

#### 5) steps to install TensorFlow

1. Download and install [Anaconda](#) with latest version.
2. On the Windows Start the menu and open an Anaconda Command Prompt. On macOS or Linux open a terminal window and also use the bash shell on macOS or Linux.
3. Choose a name for the TensorFlow environment, like “tf”.
4. To install the current release of GPU TensorFlow on the Linux or Windows use following command:

```
Conda create -n tf-gpu tensorflow-gpu
```

```
Conda activate tf-gpu
```

TensorFlow is now installed and also ready to use.

**6) Steps we followed in coding:**

- a. First, we import all the necessary packages
- b. Getting and performing data cleaning
- c. Extracting the feature vector from all images
- d. Loading dataset for Training the model
- e. Tokenizing the vocabulary
- f. Create Data generator
- g. Defining the CNN-RNN model
- h. Training the model
- i. Testing the model

# 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

## CONCLUSION

This work presents a model, which is a neural network that can automatically view an image and generate appropriate captions in natural language like English. The model is trained to produce the sentence or description from given image. The descriptions or captions obtained from the model are categorized as follows

- Description without errors
- Description with minor errors
- Description somewhat related to image
- Description unrelated to image

The categories in results are due to neighborhood of some particular words, i.e., for word like car it's neighborhood words like vehicle, van, cab etc. are also generated which might be incorrect. After so much of experiments, it is conclusive that use of larger datasets increases performance of the model. The larger dataset will increase accuracy as well as reduce losses. Also, it will be interesting that how unsupervised data for both images as well as text can be used for improving the image caption generation approaches.

# REFERENCES



## REFERENCES

### RESEARCH PAPERS: -

- [1] Kulkarni G, Premraj V, Dhar S, Li S, Choi Y, Berg AC, Berg TL (2011) Baby Talk: Understanding and Generating Image Descriptions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (20-25 June 2011)
- [2] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan
- [3] Show and Tell: A Neural Image Caption Generator (IEEE-2015)
- [4] Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra and Kumar Bansode, Camera2Caption: A Real-Time Image Caption Generator (International Conference on Computational Intelligence in Data science-2017)
- [5] N. Komal Kumar, D. Vigneswari, A. Mohan, K. Laxman, J. Yuvraj, Detection and recognition of Objects in Image Caption generator system: A deep learning approach (IEEE-2019)

### Courses related to technology: -

- [1][https://www.coursera.org/account/accomplishments/records/D767ZAC7DGQB?utm\\_source=link&utm\\_medium=certificate&utm\\_content=cert\\_image&utm\\_campaign=sharing\\_cta&utm\\_product=project](https://www.coursera.org/account/accomplishments/records/D767ZAC7DGQB?utm_source=link&utm_medium=certificate&utm_content=cert_image&utm_campaign=sharing_cta&utm_product=project)
- [2][https://www.coursera.org/account/accomplishments/records/VGBLCDVSBTB4?utm\\_source=link&utm\\_medium=certificate&utm\\_content=cert\\_image&utm\\_campaign=sharing\\_cta&utm\\_product=project](https://www.coursera.org/account/accomplishments/records/VGBLCDVSBTB4?utm_source=link&utm_medium=certificate&utm_content=cert_image&utm_campaign=sharing_cta&utm_product=project)