



Dissertation on
“Image Caption Generator”

Submitted in partial fulfilment of the requirements for the award of degree of

**Bachelor of Technology
in
Computer Science & Engineering**

UE18CS390A – Capstone Project Phase - 1

Submitted by:

**Shrivatsa G Hegde
Vinayak P Hegde
Ashwin Krishna P
Aman Prasad**

**PES1201801904
PES1201801562
PES1201801465
PES1201800271**

Under the guidance of

Prof. Nagegowda K S
Associate Professor
PES University

January - May 2021

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

‘Image Caption Generator’

is a bonafide work carried out by

**Shrivatsa G Hegde
Vinayak P Hegde
Ashwin Krishna P
Aman Prasad**

**PES1201801904
PES1201801562
PES1201801465
PES1201800271**

in partial fulfilment for the completion of sixth semester Capstone Project Phase - 1 (UE18CS390A) in the Program of Study-Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2021 – May. 2021. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6th semester academic requirements in respect of project work.

Signature
Prof. Nagegowda K S
Associate Professor

Signature
Dr. Shylaja S S
Chairperson

Signature
Dr. B K Keshavan
Dean of Faculty

External Viva

Name of the Examiners

Signature with Date

1. _____

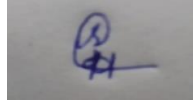
2. _____

DECLARATION

We hereby declare that the Capstone Project Phase - 1 entitled “**Image Caption Generator Using CNN-RNN & NLP**” has been carried out by us under the guidance of Prof. Nagegowda K S, Associate Professor, and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2021. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

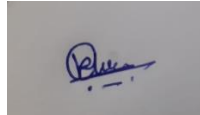
PES1201801904

Shrivatsa G Hegde



PES1201801562

Vinayak P Hegde



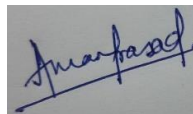
PES1201801465

Ashwin Krishna P



PES1201800271

Aman Prasad



ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Nagegowda K S, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE18CS390A - Capstone Project Phase – 1.

I am grateful to the project coordinators, Prof. Sunitha R and Prof. Silviya Nancy J for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

In the recent past, the problem of automatically describing images has gained lot of scope and interest. Researchers in the field of computer vision and image processing have continuously been trying to efficiently describe images. Due to recent advancement in the field of Deep Learning, this task has become possible.

The main difficulty in image captioning task is that, the description should not only contain the objects present in the picture, but also explain how those objects are related to one other. The description should be able to properly describe the environment. The task of training a machine to efficiently describe the environment is challenging and also has numerous applications from aiding the blind to monitoring security services.

This project mainly aims to develop a model which when given an image, identifies the objects, describes the relation between them and generates structured and grammatically correct captions.

TABLE OF CONTENT

Chapter No.	Title	Page No.
1.	INTRODUCTION	2
2.	PROBLEM STATEMENT	4
3.	LITERATURE SURVEY	6
	3.1 Learning CNN-LSTM Architectures for Image Caption Generation	6
	3.1.1 Hypothesis	6
	3.1.2 Model	6
	3.1.3 Dataset	7
	3.1.4 Alternate Models	7
	3.1.5 Conclusions	7
	3.2 Image Captioning - A Deep Learning Approach	8
	3.2.1 Hypothesis	8
	3.2.2 Model	8
	3.2.3 Alternate Model	9
	3.2.4 Limitation	9
	3.2.5 Conclusions	9
	3.3 Camera2Caption: A Real-Time Image Caption Generator	9
	3.3.1 Hypothesis	9
	3.3.1 Model	10
	3.3.3 Alternate Model	10
	3.3.4 Conclusions	10
	3.4 Show and Tell: A Neural Image Caption Generator	11
	3.4.1 Introduction	11
	3.4.2 Related work	11
	3.4.3 Model	11
	3.4.4 Alternate Model	11
	3.4.5 Experiments	12

3.4.6	Conclusions	12
4.	SYSTEM REQUIREMENTS SPECIFICATION	13
5.	HIGH LEVEL DESIGN	19
6	SYSTEM DESIGN	28
7.	IMPLEMENTATION (Walkthrough of initial steps)	30
8.	CONCLUSION OF CAPSTONE PROJECT PHASE-1	33
9.	PLAN OF WORK FOR CAPSTONE PROJECT PHASE-2	34
	REFERENCE/BIBLIOGRAPHY	35
	APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS	36
	APPENDIX B USER MANUAL (OPTIONAL)	37

LIST OF FIGURES

Figure No.	Title	Page No.
Figure1	Flowchart showing functionality of the system	17
Figure 2	State diagram	24
Figure 3	User Interface diagram	25
Figure 4	External interface diagram showing overall view of the working of the system	26
Figure 5	External interface diagram showing overall view of the working of the system with example image	27
Figure 7	Training stage	29
Figure 8	Prediction stage	29
Figure 9	LSTM decoder	29

LIST OF TABLES

TableNo.	Title	Page No.
----------	-------	----------

CHAPTER 1

INTRODUCTION

Image caption Generator is a well-known subject in the domain of Deep Learning which focusses on description of an image based upon the objects contained in it. This task is a combination of image-scene understanding, Extraction of features and translating the visual representations into a natural language. Generation of well-formed and logically correct description requires proper understanding of the language in which the captions are described.

In this task, the main difficulty to describe the image is that, the model should not only search and find out the images that are present in the image but to correctly describe the way in which objects are related to one another. The image is then described in English leading to an additional requirement of language model along with visual understanding. In our model, combined network of CNNs and RNNs are used. CNNs are used for image processing whereas RNNs are used for caption generation. In RNNs, LSTMs are specifically used which are of great help for generating the correct sequence of words.

Image captioning has a variety of applications such as security systems, robotic vision and helping the visually impaired. Looking forward to the future applications of automated image captioning, it is very clear that this field is very promising and complex at the same time. However, achieving high accuracy for caption generation models is a major concern.

The below diagram can give a basic idea about the model and how text is generated for a given input image.

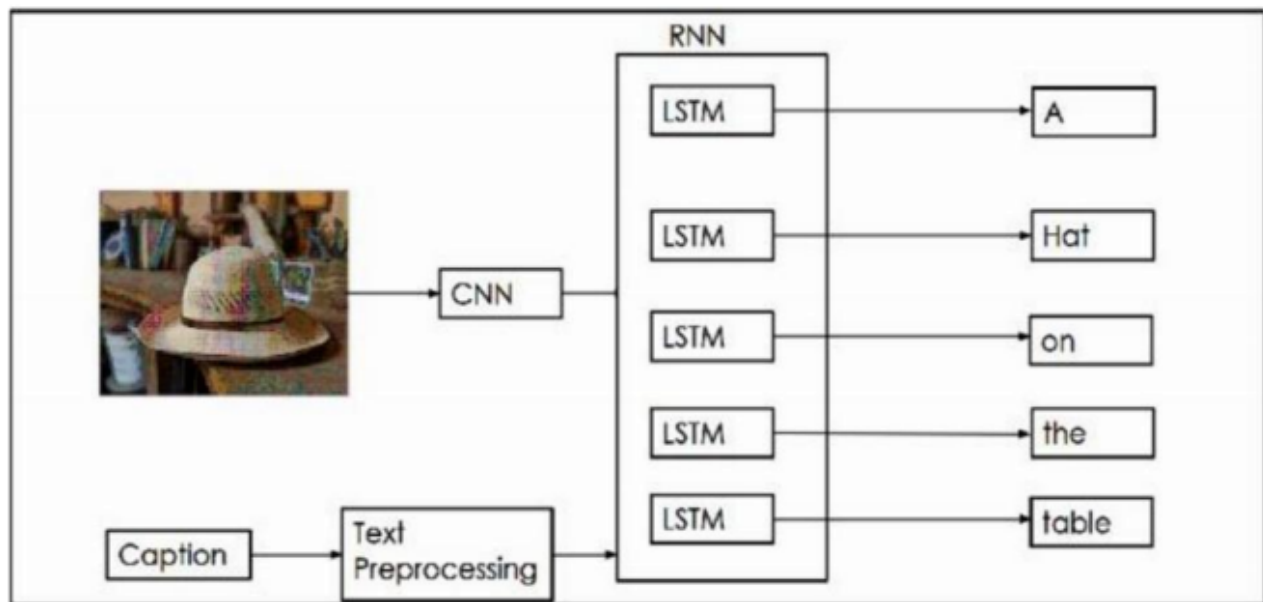


Figure 1

CHAPTER 2

PROBLEM STATEMENT

We humans can easily recognise and describe various images which we see every day without any difficulty, but this same task is very difficult for a machine to do if it is not trained to do so. There has been lot of developments in the field of image processing and computer vision in recent past and researchers have tried to build a model which on giving an image as input identifies various objects present in the image, relates them with each other and successfully describes the image. This research activity completed gained new direction after recent developments in deep learning . CNN which perform very well in any image related problems have helped us achieve this. Along with CNN . Combining CNN and RNN provides a great way to process image data and generate descriptions.

Due to increase in compute power available , training and tuning combined model of CNN and RNN has become easy. Once a machine is trained to automatically describe the image, it would help visually-aided people to recognise things in their surroundings, help children to learn language in better way and also can be used for security monitoring purposes .

Initially the image is passed on to a pre trained CNN model and then passed on to LSTM model to get appropriate set of words as descriptions. Implementing a model which not only outputs the objects present in the image but gives a meaningful and detailed explanation about the image is the main task of this project.

Feature extraction from images is done using CNN. We have used the pre-trained model Exception. The information received from CNN is then used by LSTM for generating a description of the image. However, sentences that are generated using these approaches are usually generic descriptions of the visual content and background information is ignored. Such generic descriptions do not satisfy in emergent situations as they, essentially replicate the information present in the images and detailed

descriptions regarding events and entities present in the images are not provided, which is imperative to understanding emergent situations.

CHAPTER 3

LITERATURE SURVEY

3.1 Learning CNN-LSTM Architectures for Image Caption Generation^[1]

3.1.1 Hypothesis

In recent times technologies are developing in a very faster rate and every day new inventions are happening around the world in various fields. Some of the advancements in the field of computer vision and statistical machine translation are revolutionary and these advancements are leading many other successful approaches that will redefine the entire field of focus. The framework of the model discussed by the authors here is a kind of continuation or a moderation of latest victorious approaches. Whenever researchers thought of a caption generating model, their intention was to produce sentences that are more eloquent so that any one could understand the environment, objects and the context of the image easily. To achieve this, the authors came up with an encoder-decoder approach. One RNN as an encoder and one more RNN to decode the earlier results in required target language.

RNNs have many merits because of using it, they can operate over sentences of sequence of words of varying length and the end to end results are very significant in machine translation

3.1.2 Model

The model is based on CNN-LSTM architecture.

The initial task is understanding the image, which is also called as semantic representation and that is decoded with the help of a LSTM network. All LSTMs share many same parameters. Images are vectorized and these vectorized portrayal of the image is given into the network as input. There will be a 'start' token to indicate the start of the sentence. All predictions are done by LSTMs and they produce image descriptions by using hidden states produced earlier. The authors have proposed the use of a CNN to map images to vectors of fixed length for representation of images. Also here we can see the use of 'GoNet' architecture which achieved great execution at a conference in 2014.

The authors have considered this model as 'Best' for generating descriptions for set of images selected

for training and validation purpose.

- Give a picture as an input to CNN
- Generates an encoding
- Forward it to LSTM for the job of decoding
- During this process, words are emitted.
- Feed currently guessed sequence of words again to the LSTM
- It predicts the upcoming word.

3.1.3 Dataset

The measurement and evaluation of success of the architecture is done on MSCOCO dataset which consists around 80000 training images, 40504 images for validation and 40775 test images. All the images are accompanied by minimum of five captions of different length. So totally the dataset consists of 160000 labelled images.

3.1.4 Alternate Models

Top-down approach:

CNNs for the job of encoding and replaced feedforward networks with RNNs, particularly the LSTMs. The common aspect of similar works is that the images have been represented as the top layer of a bigger CNN, thereby producing different models and architectures that are flexible for trainable and for which end-to-end training is possible.

Bottom-up:

Another technique to the same problem is nothing but making it simpler by dividing the problem into 2 smaller problems. Initially train a CNN and RNN which can learn to map pictures and parts of descriptions. Then train one more RNN which will learn to combine these i/p's from different image fragments found within real image for producing appropriate caption.

3.1.5 Conclusions

In the above discussed paper the authors have analyzed different models that were proposed for the

implementation of image caption generation process, and made improvements on them and provided detailed explanation on how a possibly best model can be built. The main factor for success here is the accuracy of the model and the authors have also looked at the effect of adding some more layers to the LSTM decoder. By observing some of the validation metrics, main observation was that, adding additional layers can make improvements in all performance metrics like BLEU-4, METEOR and CIDEr.

Also at the same time, adjoining extra layers beyond 2nd layer was not much effective because the model faced problem of overfitting.

3.2 Image Captioning - A Deep Learning Approach[2]

3.2.1 Hypothesis

- 1) In this paper the authors have proposed a model consisting of Convolutional Neural Network with many layers and a Long Short Term Memory (LSTM)
- 2) The model is evaluated and efficiency of the model is shown using the Flickr8K dataset.
- 3) Flickr 8K –It is the dataset which is being used and it contains around 8000 images.

3.2.2 Model:

The Python SciPy environment was used in the implementation of the model. To implement the deep learning model Keras 2.0 was used .

The model consists of three steps:

A. Image Feature Extraction

VGG 16 model was used to extract features of the images.

B. Sequence processor

In this step, text input is handled. The network is then connected to a LSTM .

C. Decoder

This step merges the input from the both the above phases and then feeds them into a 256

neuron layer. The output is passed on to final layer and here prediction of the next word is produced.

3.2.3 Alternate Models

Krizhevsky et al. made use of non-saturating neurons and GPU implementation of the convolution function to design a network.

The performance of this model is measured using the BLEU standard metric. Results have shown that compared to standard models the model proposed in this paper has performed better in terms of captioning the images in performance evaluation

3.2.4 Limitations

In this paper, concept of deep learning has been used to solve the problem and keras and tensorflow are used for implementation purpose.

3.2.5 Conclusion

In this paper, concept of deep learning has been used to solve the problem and keras and tensorflow are used for implementation purpose.

3.3 Camera2Caption: A Real-Time Image Caption Generator[3]

3.3.1 Hypothesis

It is tough for a computer to describe objects in the image with their relationships , but it is very useful in many fields. For example it could help the visually impaired people to understand their surrounding there by it can be used as an assistant.

Along with identifying the objects in images , it should also express the relationship between them in natural language.

That is why the process of caption generation for images is considered as the challenging task.

The whole purpose of the captioning is the ability to copy human to process the image data into natural language. That why it is the most attractive problem in Artificial Intelligence.

3.3.2 Model

In our project of the model we are going with the approach similar to the Show Tell by introducing an encoder and decoder architectural design.

The encoder is the pre-trained Inception V4 CNN (Convolutional Neural Network) model which is designed by the Google and the decoder is the Deep RNN (Recurrent Neural Network) with Long Short Term Memory Cells.

3.3.3 Alternate Models

Earlier, people are using templates for image caption generator instead of probabilistic generative models in natural language.

Farhadi et al in 2010, use triplets of (object , action, scene) along with the pre defined templates for generating captions.

The deep NN models were used with language model, which integrates the images and captions in the same vector hyper space.

So, our basic approach is, feeding images to CNN to get the extraction vector, which then goes to the RNN (which is LSTM) for caption generation.

3.3.4 Conclusion

We are first creating the system and deploying in mobile first to get all the possible use-cases for our model.

And we comparing our model with some other models based on image captioning to get positives and negatives.

After each testing , we be working on the negatives.

3.4 Show and Tell: A Neural Image Caption Generator, by Google[4]

3.4.1 Introduction

- The captions generated by the model in this paper must capture the objects and show us how objects that are present in the image are related to each other.
- The model that is described in this paper is based on an image classification CNN and then followed by word generating RNN.
- Captions are generated in English for the given input image

3.4.2 Related Work

- Deep CNNs are utilised for image classification along with RNNs for caption generation.

3.4.3 Model

- Neural networks and various other frameworks are used to generate image. This is the major difficulty in designing training and tuning of RNNs ..
- In order to overcome this difficulty, LSTMs are used for better sequence generation.

Multiple approaches are used to describe an image,.

Sampling process is used where we sample the word which is at the beginning of the sentence, later related set is given, further sampling the second word, this is done until sampling of last token is done.

BeamSearch is another approach where n best sentences are used till some particular time say t for generation of sentences having size $n+1$. The best n sentences are kept finally.

3.4.4 Alternate Model

- **A Farhadi** uses detections on various scene elements in the image, it is then converted to text
- **S. Li** outputs caption by making use of clauses obtained from objects that are detected

- The model in the paper is a result of great development in the field of sequence generation. Instead of beginning with a sentence, an image is provided which is processed by a deep CNNs (Convolution Neural Network).

Closest work to the proposed model in the paper:

- **R. Kiros** has used a forward feeding neural network which is used to predict words based upon the input image and pre-generated words.
- **J. Mao** has used recurrent neural networks for the same task as above.

3.4.5 Experiments

- Extensive experiments are done as to check the effectiveness of the model using various metrics, architectures and datasets.
- The challenges of training the models had to do with over-fitting. The weights of the CNN component was initialized a pretrained mode and overfitting was avoided.
- Transfer learning was done on two datasets : Flickr30k & Flickr8k.
- The two datasets are similar and have same owner. Upon training the Flickr30k, results were 4 BLEU points better.
- More data gives better results as the models are data driven.

Evaluating the descriptions provided by NIC by the help of human evaluation is performed. It is concluded that performance of NIC as compared to the reference system is more superior but very bad when checked upon with reality. One of the conclusion is that BLEU is not a very accurate metric

Datasets used:

MSCOCO, SBU, Pascal VOC 2008, Flickr8k, Flickr30k,

3.4.6 Conclusion

NIC (name of model used in the paperwork) uses neural networks which generates description in English upon providing an image as an input. It is based on CNNs that does image processing, followed by RNNs for generation of sentences. It can be concluded that performance increases as the size of dataset increases. The work done by the authors of this paper is very extensive and displays great application of deep learning concepts.

CHAPTER 4

PROJECT REQUIREMENTS SPECIFICATION

4.1 Introduction

This document lays out the project plan for the design and development of image caption generator, a CNN-RNN based combined model to generate appropriate captions for the images.

4.1.1 Project Scope

We humans can easily recognise and describe various images which we see everyday without any difficulty, but this same task is very difficult for a machine to do if it is not trained to do so. There has been lot of developments in the field of image processing and computer vision in recent past and researchers have tried to build a model which on giving an image as input identifies various objects present in the image, relates them with each other and successfully describes the image. This research activity completed gained new direction after recent developments in deep learning . CNN which perform very well in any image related problems have helped us achieve this. Along with CNN . Combining CNN and RNN provides a great way to process image data and generate descriptions.

Due to increase in compute power available , training and tuning combined model of CNN and RNN has become easy. Once a machine is trained to automatically describe the image, it would help visually-aided people to recognise things in their surrounding, help children to learn language in better way and also can be used for security monitoring purposes .

Initially the image is passed on to a pre trained CNN model and then passed on to LSTM model to get appropriate set of words as descriptions. Implementing a model which not only outputs the objects present in the image but gives a meaningful and detailed explanation about the image is the main task of this project.

4.2 Product Perspective

The product's primary feature will be generation of captions for the input images. This has a vast number of applications, with the help of our product we can help the blind people to get them information about the surroundings. Security and safety can be improved by analysis of the images for alarming the concerned authorities. Our product will also be a great help in automated cars by learning about the environment.

Our product tends to mimic the human brain just as a human being describes a particular image.

4.2.1 Product Features

UI: The user interacts by giving the image as input.

Image Description Model: A trained image description model takes an image from user and outputs the description of the image.

4.2.2 Operating Environment

- **Hardware Platform:** Intel Core i5 7th Gen 8GB DDR4 or higher. (Recommended: NVIDIA GeForce MX110 (2GB DDR3 dedicated) or higher.)
- **Operating System :** Ubuntu 16.04 or higher / Windows 10.
- Python 3.x

4.2.3 General Constraints, Assumptions and Dependencies

- High computation power is required
- End user cannot expect 100% accurate captions Input images will be of good resolution
- Image should be clear
- Creation of dataset upon which training is to be performed should include only those image which have no legal complications ex: copyright, no permission by the owner of picture, defense classified images etc.

4.2.3 Risks

Obtaining a large amount of high quality images could be difficult.

High variations in image quality and dimensions may lead to poor caption generation.

4.3 Functional Requirements

We will train our model on desired dataset. We will initially use a part of the dataset to validate and test the model. To finalize our model, we will test it against images from other datasets and check the accuracy.

The sequence of operations starts with the user uploading the image. Once this is done the CNN and RNN perform their function and produce meaningful and structured captions.

If user provides the input image then it will generate the caption , else if the input is not proper then the model will ask user to give correct input image.

4.4 External Interface Requirements

4.4.1 User Interfaces

The User Interface is very simple and convenient. The user interacts by inserting the image.

If user provides the image as input then it will generate the caption , else if the input is not proper then the model will give "Invalid file format" error message and ask user to provide correct input image.

User can upload any image for which he expects a description/caption. Then our well-trained image description model takes an image and outputs the description of the picture.

Generated captions may or may not be 100% accurate because accuracy of generated caption depends on the grade of the image, and clarity of objects present in the picture.

Flow chart showing the functionality of the model:-

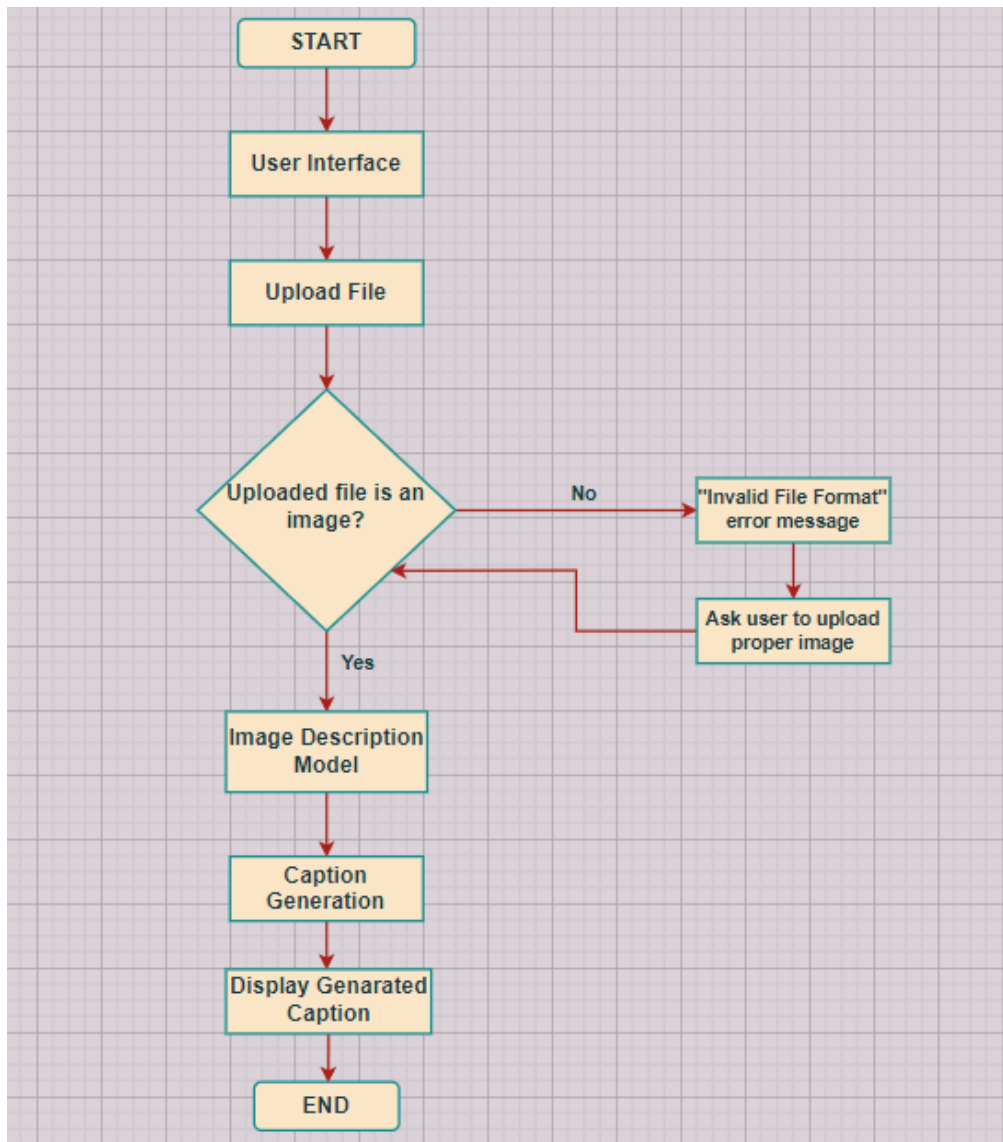


Figure 2

4.4.2 Hardware Requirements

Intel Core i5 7th Gen 8GB DDR4 or higher. (Recommended: NVIDIA GeForce MX110 (2GB DDR3 dedicated) or higher.).

4.4.3 Software Requirements

Python 3 x:

Description: High level language used for developing software, machine learning, etc.

Version: 3.x

Databases: None

Operating Systems: Windows/Linux

Tools and libraries: Keras, tensor-flow, pandas, numpy, etc.

Source: <https://www.python.org/downloads/>

4.4.4 Communication Interfaces

As our main goal is to build the model which generates appropriate captions for the images, we restrict the efforts for user interface and web hoisting. So no specific communication interfaces listed as of now.

4.5 Non-Functional Requirements

4.5.1 Performance Requirement

The captions generated by the product will be reliable. Although, accuracy can be increased by using large datasets but still there may be errors in the generated captions.

4.5.2 Safety Requirements

Creation of dataset upon which training is to be performed should include only those image which have no legal complications ex: copyright, no permission by the owner of picture, defense classified images etc.

4.5.3 Security Requirements

None as of now.

4.6 Other Requirements

Appendix A: Definitions, Acronyms and Abbreviations

- UI : User Interface

Appendix B: References

- [Python documentation](#)
- [Keras library documentation](#)
- [Tensorflow website](#)

CHAPTER 5

High Level Design

5.1 Introduction

Producing descriptions for pictures or images is an area which is in limelight recently and it is also becoming well-liked by the people and researchers in the domain of Deep Learning which handles description of an image based upon the objects contained in it. This task is a combination of image-scene understanding, Extraction of features and translating the visual representations into a natural language. Generation of well-formed and logically correct descriptions is possible only with good understanding of the target language in terms of syntactic knowledge and semantic knowledge.

The main challenge here is generating a description which does not only looks for the contents in an image, rather it must also correctly describe the way in which objects are related to one another. The descriptions of the image has to be expressive and those descriptions should be in a human understandable language like English, leading to an additional requirement of language model along with visual understanding. In our model, combined network of CNNs and RNNs are used. CNNs are used for image processing whereas RNNs are used for caption generation. In RNNs , LSTMs are specifically used which are of great help for generating the correct sequence of words.

5.2 Current System

Image caption generator has shown great development in the past few years. It is clearly seen that the model is data driven and performance is enhanced with large datasets. In the current scenario, training the model with large datasets requires powerful computational power. The model gives correct descriptions of the input images but accuracy is a major concern which can be increased in the future through new technologies/methods.

When a new product is out there in the market, the expectations always stays high. Similarly when someone introduces a model to generate descriptions for images, people expect it to be hundred percent accurate so that it could detect all the contents and objects in the image and tell exactly what the image is all about. But when it comes to the technical level, achieving such high accuracy in this attempt is not at all easy and feasible at this stage.

Today there are only very few successful models for generation of descriptions for images, **CaptionBot** by Microsoft is one such successful image captioning model.

CaptionBot and DrawingBot by Microsoft are two models whose functionalities are exactly opposite. CaptionBot for image → caption and DrawingBot for caption → image .

Merits of CaptionBot:

- Easy to use
- Simple User Interface
- Impressive speed of processing
- Able to detect and distinguish objects in some complex images

Demerits of CaptionBot:

- Accuracy is not excellent
- No guarantee of correct description
- Some captions were totally opposite to the context, and the model was trending in social medias because of many people making fun of it. (A car from rear view described as a suitcase, skull as banana and vice-versa, rat described as a cat and many such examples!)

But one should always understand that to generate accurate caption, the objects in the image should be clear, distinguishable and picture should be of good quality. Also model should be trained over huge dataset which is a complex process because of the large computational power needed. Also there are billions of objects that exist on the universe, so training the model in this case always has its own limitations.

5.3 Design Considerations

5.3.1 Design Goals

Our main goal is to automatically generate captions for the images given as input by the user. Before the latest victorious approaches and evolution of Deep NNs the task was considered to be extremely difficult by many researchers and developers who were experts in this field.

Generating descriptions is a fundamental task and has wide range of applications in aiding the blind, enhancing performance of self-driving cars.

The model described is dependent on 2 factors – one is CNN and the other is RNN .

CNN has proven to be the first choice for many problems that includes images and data from the images as inputs.

RNNs are best suited for any kind of work with letters, words, sentences & paragraphs.

5.3.2 Architecture Choices

Basically the approach here is an encoder-decoder approach. From the knowledge that we gained during literature survey, we can specify some of the major architectural choices and decisions.

- 1) CNN to understand the image, that means; finding the objects that are present in the image and getting knowledge about the surrounding and relationships.
- 2) Outcome of the above step is used by the image description model to produce proper sentences and meaningful captions.

In more simple words, an encoder and one more RNN to decode the earlier results in required target language.

RNNs have many merits because of using it, they can operate over sentences of sequence of words of varying length and the end to end results are very significant in machine translation.

5.3.3 Constraints, Assumptions and Dependencies

Design Constraints:

Achieving high accuracy requires training the model with more number of images adding up to computational and time constraint.

Creation of dataset upon which training is to be performed should include only those image which have no legal complications

Assumption:

End user cannot expect 100% accurate caption.

Dependencies:

Tools – Python, Tensorflow, keras, Jupyter, git

Computers with Faster RAM is required as the data is large

Performance Requirements:

The captions generated by the product will be reliable. Although, accuracy can be increased by using large datasets but still there may be errors in the generated captions.

Safety Requirements:

Creation of dataset upon which training is to be performed should include only those image which have no legal complications ex: copyright, no permission by the owner of picture, defense classified images etc.

Hardware Requirements:

Intel Core i5 7th Gen 8GB DDR4 or higher. (Recommended: NVIDIA GeForce MX110 (2GB DDR3 dedicated) or higher.).

Software Requirements:

Python 3 x:

Description: High level language used for developing software, machine learning, etc.

Version: 3.x

Databases: None

Operating Systems: Windows/Linux

Tools and libraries: Keras, tensor-flow, pandas, numpy, google colab etc.

Source: <https://www.python.org/downloads/>

5.4 High Level System Design

Flowchart showing functionality of the system:

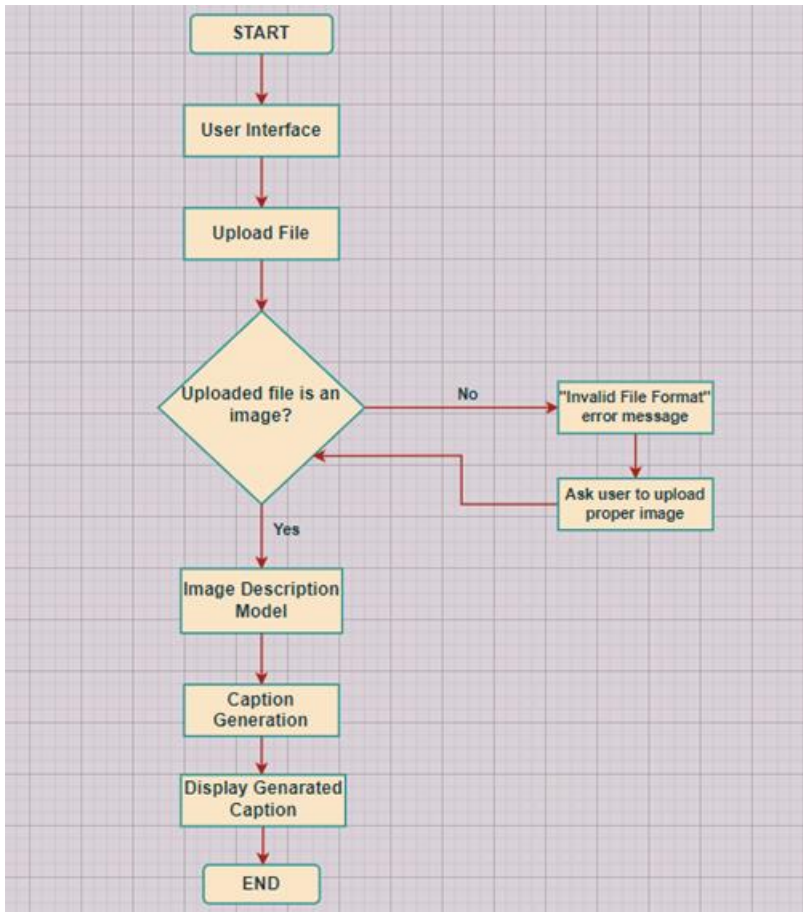


Figure 2

5.5 Design Description

5.5.1 Reusability Considerations

NumPy – To do any array based or matrix based calculation and data analysis then its better to use it.

Pandas – Can be used for data cleaning and analysis.

Matplotlib – helpful in developing any plots, graph etc.

Keras – Interface that allows to easily access and customize many ANN related tools.

5.6 State Diagram

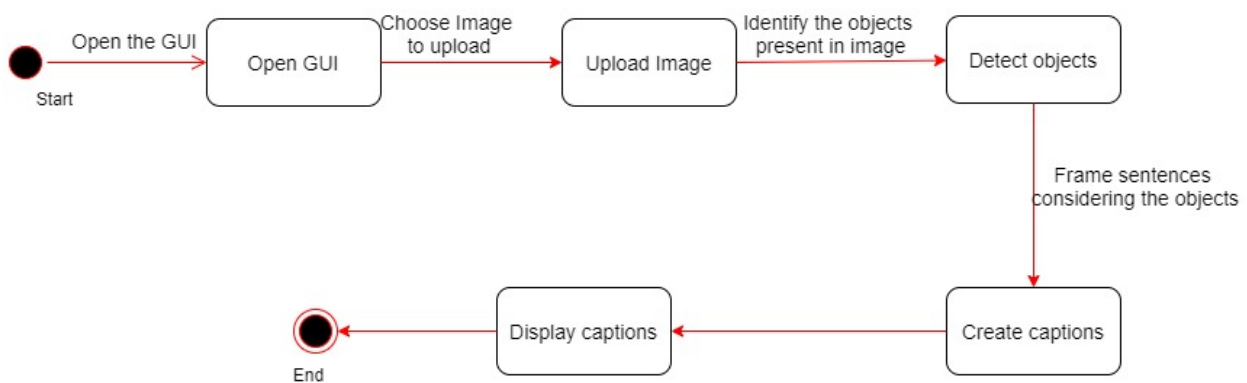


Figure 3

5.7 User Interface Diagrams

User interface:

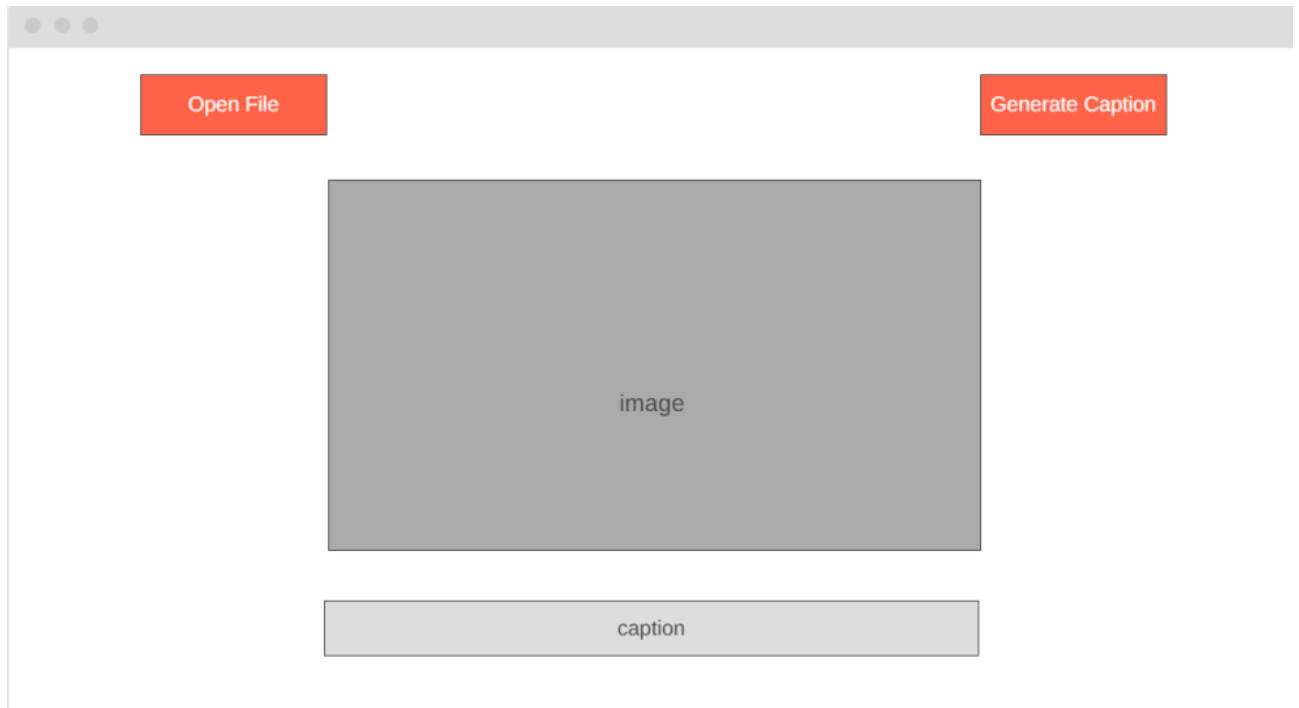


Figure 4

5.8 External Interfaces

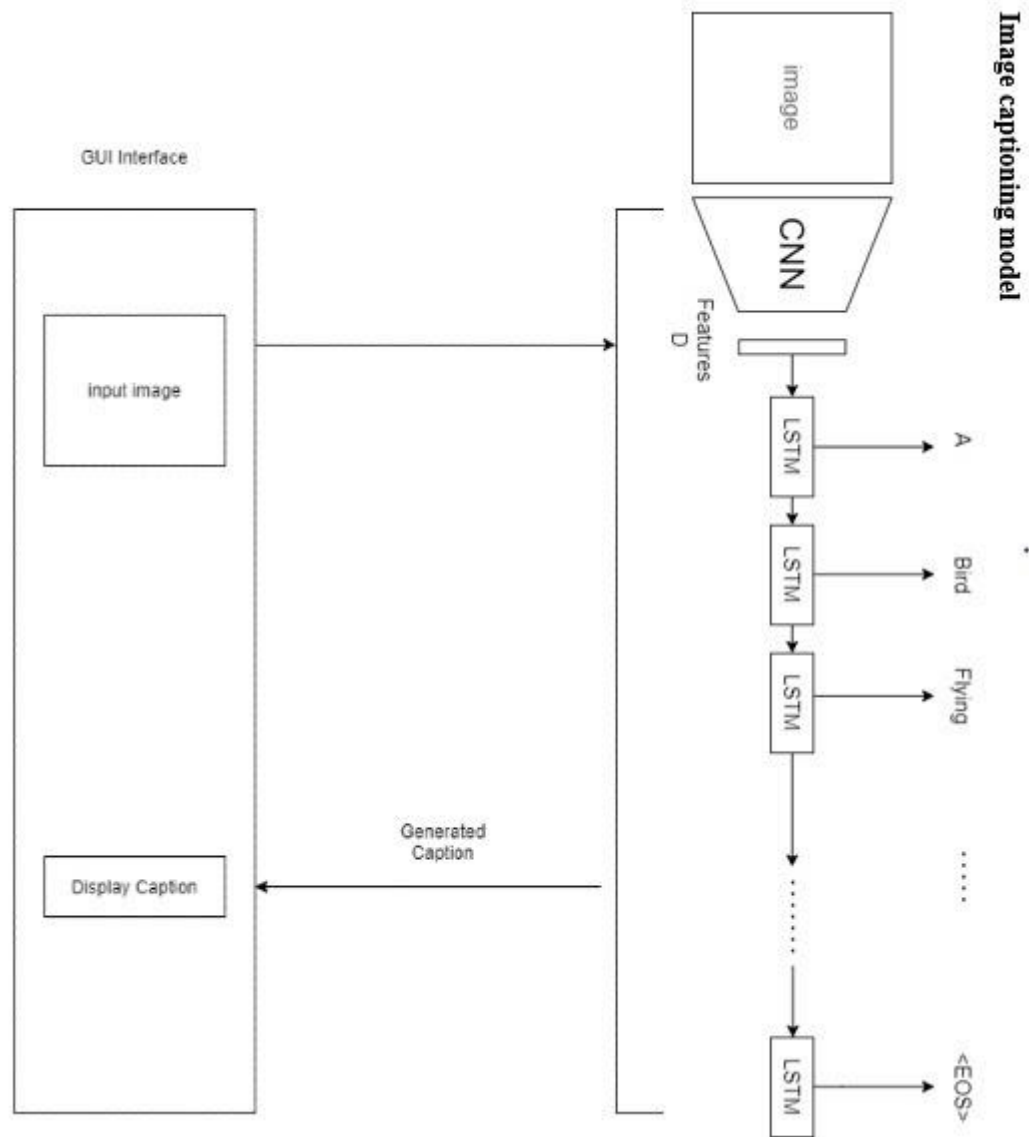


Figure 5

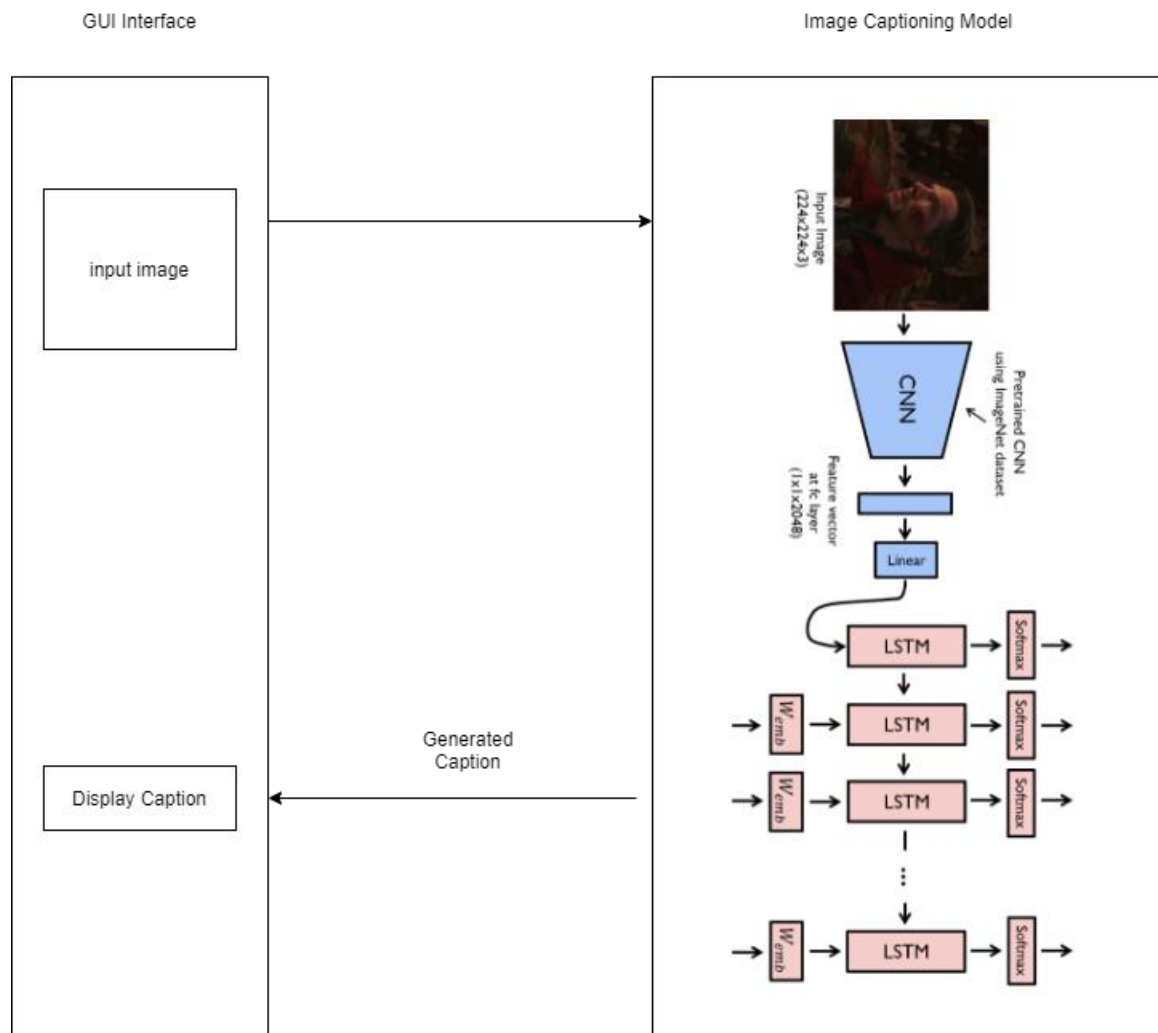


Figure 6

5.9 Help

The product presents a UI where the user inputs an image and gets the desired output. For better understanding of the inbuilt system, user can refer to the following links:

- Tensorflow website
- Keras , pandas library documentation
- Python documentation.

CHAPTER 6

SYSTEM DESIGN

We are using the flickr's 8K dataset for our Image caption generator. A pre defined/trained model shall be used to get the whole content which is present in the all images. The dataset contains various descriptions of every image and minimal cleaning is required for the descriptive texts.

Moving on to the next step, cleaning of the description text shall be performed. The descriptions are already tokenized and can be worked with smoothly.

Our model can be described mainly in 3 parts:-

image Feature Extractor –It is a pre-trained model on the Image-Net dataset

Sequence/sequential Processor - It is a word interpreting layer which handles the text i/o, followed by LSTM's.

Decoder-This will merge the vectors from 2 input models by add(+)tion operation.

Further, we shall fit it on the training dataset. At last of the run, We will use the best model with high accuracy to train our final model.

We load the training dataset for preparing the tokenizer for encoding the generated words as input sequences to our model.

Once the model is fit, we shall evaluate results of its predictions on the holdout test dataset. Then loading the image to describe and extract the features, prediction of features is done and used as an input to the existing model. At the end, we generate a description of the image.

Training Stage:

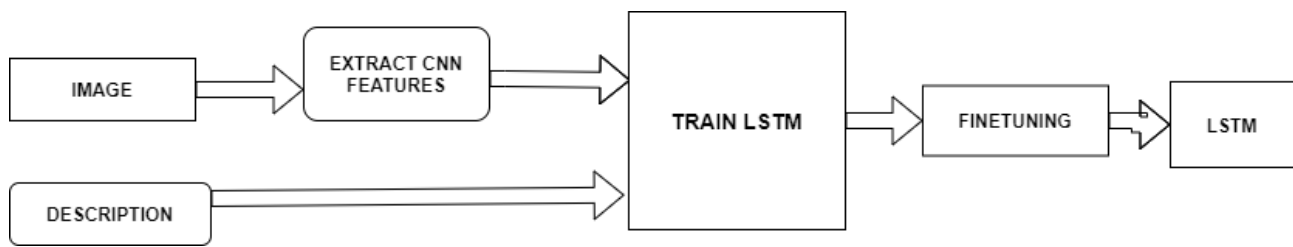


Figure 7

Prediction stage:

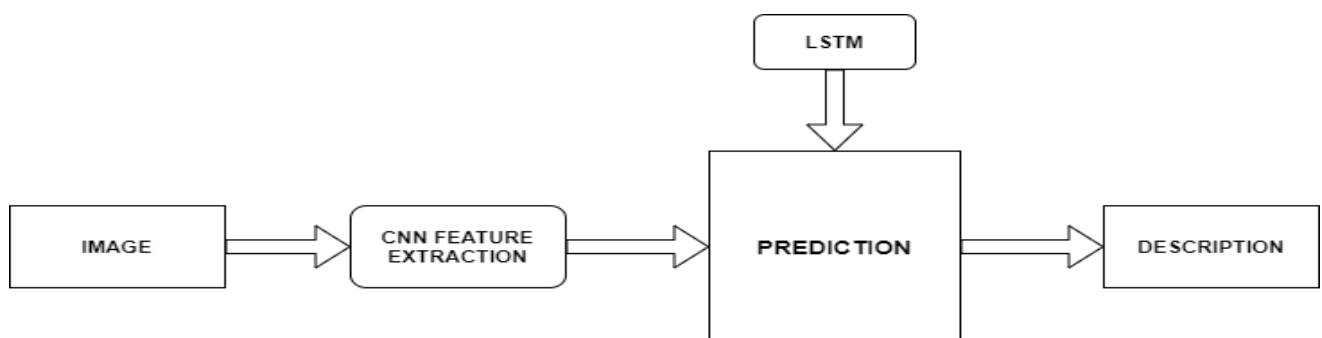


Figure 8

LSTM decoder:

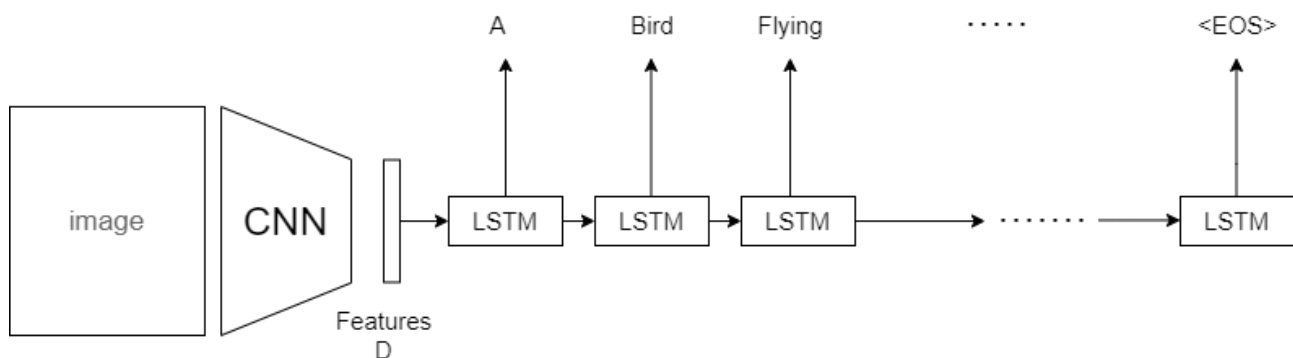


Figure 9

CHAPTER 7

IMPLEMENTATION (Walkthrough)

> image_caption_generator

Name	Date modified	Type	Size
.git	15-05-2021 16:47	File folder	
.ipynb_checkpoints	15-05-2021 16:51	File folder	
Flicker8k_images	09-05-2021 23:46	File folder	
Flicker8k_text	09-05-2021 23:46	File folder	
.gitignore	15-05-2021 16:41	Text Document	1 KB
descriptions	15-05-2021 17:03	Text Document	3,072 KB
features	15-05-2021 18:33	P File	65,477 KB
image_caption_generator	15-05-2021 18:53	IPYNB File	25 KB
tokenizer	15-05-2021 18:52	P File	344 KB

```
dataset_text = r"C:\Users\Shrivatsa Hegde\Desktop\2022\image_caption_generator\Flicker8k_text"
dataset_images = r"C:\Users\Shrivatsa Hegde\Desktop\2022\image_caption_generator\Flicker8k_images"

#data cleaning
vocabulary = data_cleaning(dataset_text)
print(" vocabulary is created from all the descriptions")
print("descriptions.txt is created...")
print("Every image is mapped with a list of 5 captions..")
print("Step 1 completed")
```

```
Length of descriptions = 8092
Length of vocabulary = 8763
vocabulary is created from all the descriptions
descriptions.txt is created...
Every image is mapped with a list of 5 captions..
Step 1 completed
```

Getting and performing data cleaning :

The main text file which contains all image captions is Flickr8k.token in our Flickr_8k_text folder. 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 6 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.

cleaning_text(descriptions) – This function takes all descriptions and performs data cleaning. we will be removing punctuations, converting all text to lowercase and removing words that contain numbers.

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.

data_cleaning(dataset_text) - get the image data and caption data, cleans it, dump it into files.

```
features = extract_features(dataset_images)
dump(features, open("features.p", "wb"))


features = load(open("features.p", "rb"))
print("Extracting the feature vector from all images ")
print(" the features dictionary is dumped into “features.p” pickle file.")
print("Step 2 completed ...")
```

100%  8091/8091 [1:24:18<00:00, 1.60it/s]

Extracting the feature vector from all images
the features dictionary is dumped into “features.p” pickle file.
Step 2 completed ...

```
features = extract_features(dataset_images)
dump(features, open("features.p", "wb"))

features = load(open("features.p", "rb"))
print("Extracting the feature vector from all images ")
print(" the features dictionary is dumped into “features.p” pickle file.")
print("Step 2 completed ...")
```

40%  3204/8091 [33:10<58:17, 1.40it/s]

```
filename = dataset_text + "/" + "Flickr_8k.trainImages.txt"
train_imgs = load_photos(filename)
train_descriptions = load_clean_descriptions("descriptions.txt", train_imgs)
train_features = load_features(train_imgs)
print("dictionary for image names and their feature vector is obtained")
print("step 3 completed... ")
```

```
dictionary for image names and their feature vector is obtained
step 3 completed...
```

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.

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.

```
tokenizer = create_tokenizer(train_descriptions)
dump(tokenizer, open('tokenizer.p', 'wb'))
vocab_size = len(tokenizer.word_index) + 1
vocab_size
print("each word is given an index and stored into tokenizer.p")
print("step 4 completed ...")
```

```
each word is given an index and stored into tokenizer.p
step 4 completed ...
```

CHAPTER 8

CONCLUSION OF CAPSTONE PROJECT PHASE-1

Initially we defined a clear problem statement, found the scope of the project and major tasks to be done. We prepared a timeline Gantt Chart and planned our project by dividing it into smaller tasks. We completed the literature survey, gone through many research papers and studied about recent advancements in the area of caption generation, and among various available methods we selected the most suitable model and architecture for our project. We learnt about the working of Convolutional Neural Networks, Recurrent Neural Networks and LSTM and how we could use these in different stages of our project. We prepared Project Requirement Specification document and High level design document.

We have planned about the implementation of the project to build a working model which we will be doing in the second phase of the project and we intend to achieve maximum accuracy in the process of Image Caption Generation.

CHAPTER 9

PLAN OF WORK FOR CAPSTONE PROJECT PHASE-2

- 1) Prepare Low level Design Document.
- 2) Inputs to the technology, platforms used for the project.
- 3) First of all we going to import all the needed packages
- 4) Getting and performing data cleaning
- 5) feature extraction from the images using CNN
- 6) data set loading for the purpose of training
- 7) Tokenizing the vocabulary
- 8) Create Data generator
- 9) Defining our CNN(encoder)-RNN(decoder) model
- 10) Backtesting the strategies
- 11) Coding up the strategies
- 12) Training the model
- 13) Testing the model
- 14) Demo of the final and working model
- 15) Hard and soft copy of the report

REFERENCE / BIBLIOGRAPHY

- [1] Moses Soh, “Learning CNN-LSTM Architectures for Image Caption Generation”, Department of Computer Science, Stanford University.
- [2] Lakshminarasimhan Srinivasan, Dinesh Sreekanthan, Amutha A L, “Image Captioning - A Deep Learning Approach”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7239-7242
- [3] Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra and Nand Kumar Bansode, “Camera2Caption: A Real-Time Image Caption Generator”, Department of Computer Engineering Army Institute of Technology, Pune, India
- [4] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, “Show and Tell: A Neural Image Caption Generator”, Google
- [5] MD. ZAKIR HOSSAIN, FERDOUS SOHEL, MOHD FAIRUZ SHIRATUDDIN, HAMID LAGA "A Comprehensive Survey of Deep Learning for Image Captioning" - Murdoch University, Australia
- [6] captionbot.ai by Microsoft

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

UI : User Interface

FF : Feed Forward (networks)

APPENDIX B USER MANUAL (OPTIONAL)

Some resources and documentations for the understanding of planned development of the model:

- [Python documentation](#)
- [Keras library documentation](#)
- [Tensorflow website](#)
- [CaptionBot.ai product webpage](#)

THANK YOU