Image Caption Generation Using CNN And LSTM

Project Work-I Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BY

KOLICHINA SAI CHARAN LAHIRI	**198X1A0573**
MANAM ANITHA LAKSHMI	**198X1A0585**
POKALA PREM KUMAR	**198X1A05B5**
SHAIK ALTAF	**198X1A05C9**
MANDLA YASWANTH KUMAR	**198X1A0586**

Under the Guidance of

MR. S.L.V.V.D. SARMA
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY

ACCREDITED BY NAAC WITH 'A' GRADE

(APPROVED BY AICTE, AFFILIATED TO JNTUK, KAKINADA)

NH-16, CHOWDAVARAM, GUNTUR–522019

2022–2023

KALLAM HARANADHAREDDY INISTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"IMAGE CAPTION GENERATION USING CNN AND LSTM"** being submitted by

KOLICHINA SAI CHARAN LAHIRI	**198X1A0573**
MANAM ANITHA LAKSHMI	**198X1A0585**
POKALA PREM KUMAR	**198X1A05B5**
SHAIK ALTAF	**198X1A05C9**
MANDLA YASWANTH KUMAR	**198X1A0586**

in the partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University,Kakinada is a record of bonafide work carried out under our guidance and supervision.

SUPERVISOR **HEAD OF THE DEPARTMENT**

Mr. S.L.V.V.D. SARMA
Assistant Professor

Mr. V. RAJIV JETSON
Professor&HOD

EXTERNAL EXAMINER

DECLARATION

I/We **KOLICHINA SAI CHARAN LAHIRI (198X1A0573), MANAM ANITHA LAKSHMI (198X1A0585), POKALA PREM KUMAR (198X1A05B5), SHAIK ALTAF (198X1A05C9), MANDLA YASWANTH KUMAR (198X1A0586)** hereby declare that the project work-I titled "**IMAGE CAPTION GENERATION USING CNN AND LSTM**" is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. This is a Project Work-I Report of bonafide work carried out by us and the result embodied in this project have not been reproduced or copied from any source. The result embodied in this project have not been submitted to any other university for the award of any other degree.

KOLICHINA SAI CHARAN LAHIRI	**198X1A0573**
MANAM ANITHA LAKSHMI	**198X1A0585**
POKALA PREM KUMAR	**198X1A05B5**
SHAIK ALTAF	**198X1A05C9**
MANDLA YASWANTH KUMAR	**198X1A0586**

ACKNOWLEDGEMENT

We profoundly grateful to express our deep sense of gratitude and respect towards our honorable chairman, **Sri KALLAM MOHAN REDDY**, Chairman of Kallam group for his precious support in the college.

We are thankful to **Dr. M. UMA SANKARA REDDY**, Director, KHIT, GUNTUR for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B.S.B REDDY,**Principal, KHIT, GUNTUR for his support during and until the completion of the project.

We are greatly indebted to **Mr. V. RAJIV JETSON** Professor & Head of the Department, Computer Science and Engineering, KHIT, GUNTUR for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college.

We are also thankful to our Project Coordinator **Mr. G. MANTHRU NAYAK** who helped us in each step of our project.

We extend our deep sense of gratitude to our Supervisor **Mr. S.L.V.V.D. SARMA**, Assistant Professor**,** and other Faculty Members & Support staff for their valuable suggestions,guidance and constructive ideas in each and every step,which was indeed of great help towards the successful completion of our project.

KOLICHINA SAI CHARAN LAHIRI	**198X1A0573**
MANAM ANITHA LAKSHMI	**198X1A0585**
POKALA PREM KUMAR	**198X1A05B5**
SHAIK ALTAF	**198X1A05C9**
MANDLA YASWANTH KUMAR	**198X1A0586**

ABSTRACT

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

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 INTRODUCTION

Every day, a large number of images are seen from various sources such as the news articles, document diagrams and advertisements. These sources contain images that viewers would have to interpret themselves. Most images do not have a description, but the human can largely understand them without their detailed captions. However, machine needs to interpret some form of image captions if humans need automatic image captions from it. Image captioning is important for many reasons. Captions for every image on the internet can lead to faster and descriptively accurate images searches and indexing. Ever since researchers started working on object recognition in images, it became clear that only providing the names of the objects recognized does not make such a good impression as a full human-like description. As long as machines do not think, talk, and behave like humans, natural language descriptions will remain a challenge to be solved.Image captioning has various applications in various fields such as bio-medicine, commerce, web searching and military etc. Social media like Instagram , Facebook etc can generate captions automatically from images.

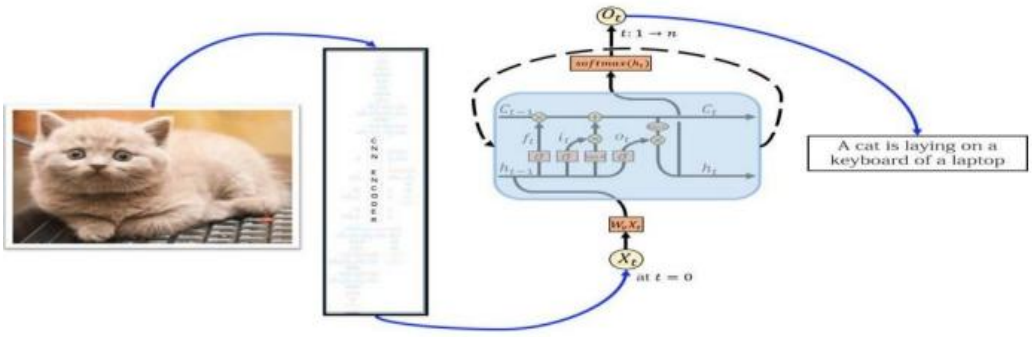


Figure 1.1: Model Architecture

1.2 MOTIVATION

Generating captions for images is a vital task relevant to the area of both Computer Vision and Natural Language Processing. Mimicking the human ability of providing descriptions for images by a machine is itself a remarkable step along the line of Artificial Intelligence. The main challenge of this task is to capture how objects relate to each other in the image and to express them in a natural language (like English).Traditionally, computer systems have been using pre- defined templates for generating text descriptions for images. However, this approach does not provide sufficient variety required for generating lexically rich text descriptions. This shortcoming has been suppressed with the increased efficiency of neural networks. Many state of art models use neural networks for generating captions by taking image as input and predicting next lexical unit in the output sentence

1.3 IMAGE CAPTIONING

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

1.4 OBJECTIVE OF IMAGE CAPTION GENERATION

The objective of our project is to develop a web based interface for users to get the description of the image and to make a classification system in order to differentiate images as per their description.It can also make the task easier which is complicated as they have to maintain and explore enormous amounts of data.

1.5 SCOPE OF IMAGE CAPTION GENERATION

Image captioning is a branch of artificial intelligence research that focuses on recognising natural scenes and explaining them in natural language. Image captioning could help with retrieval by allowing us to organise and request pictorial or image-based content in new ways.

1.6 PROBLEM STATEMENT

In existing system to generate a caption they used retrieval based and template based approach. By using these method the caption will be in fixed length and hard to generate captions for an image. Now in this project, CNN and LSTM are used to generate caption with high accuracy and efficiency.

CHAPTER 2

LITERATURE SURVEY

2.LITERATURE SURVEY

2.1 LITERATURE SURVEY

2.1.1 TEMPLATE-BASED APPROACHES

Title : Corpus-guided sentence generation of natural images

(Authors:Y. Yang, C. L. Teo, H. Daume, Y. Aloimono)

Template-based approaches have fixed templates with a number of blank slots to generate captions. In these approaches, different objects, attributes, actions are detected first and then the blank spaces in the templates are filled. For example, Farhadi et al. use a triplet of scene elements to fill the template slots for generating image captions.Template-based methods can generate grammatically correct captions.However, templates are predefined and cannot generate variable-length captions. Moreover, later on, parsing based language models have been introduced in image captioning which are more powerful than fixed template-based methods.

2.1.2 RETRIEVAL-BASED APPROACHES

Title : Every picture tells a story: Generating sentences from images

Authors:(A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hock enmaier, D. Forsyth)

Captions can be retrieved from visual space and multimodal space. In retrieval-based approaches, captions are retrieved from a set of existing captions. Retrieval based methods first find the visually similar images with their captions from the training data set. These captions are called candidate captions. The captions for the query image are selected from these captions pool. These methods produce general and syntactically correct captions. However, they cannot generate image specific and semantically correct captions.

2.1.3 RNN-CNN DEEP NEURAL NETWORK ALGORITHM

Title : Neural Caption Generation for News Images in School of Engineering and Applied Science

Authors: (Vishwash Batra, Yulan He, George Vogiatzis)

In this paper, they proposed a model for news images and caption generation for various news articles. The proposed system was trained using images, descriptions and captions. The idea of adding long description was an advancement which shows an great improvement in the results. The method is different than other traditional methodologies. The input along with image was a long description which increases the chances of correct caption generation. The more the vocabulary better the results. They used RNN and CNN deep neural network algorithm. Example: If the image has a building in it. It is difficult to predict if its a school or a society. The long description along with an image helps to predict the correct caption.

2.1.4 DENSE CAPTIONING

Title : DenseCap: Fully Convolutional Localization Networks for Dense Captioning

Authors : (Justin Johnson, Andrej Karpathy,Li Fei-Fei)

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

A typical method of this category has the following steps:

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

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

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

S.NO	AUTHORS	TITLE	ADVANTAGES	DISAVANTAGES
1	A.Farhadi,M. Hejrati,M.A Sadeghi,P.Young,ForsythC .Rashtchian	Every picture tells a story: Generating sentences from images (2010)	This model can alsoprovide images which are best described by a given sentence	It can not provideaccurate captions
2	Y.Yang,C.L. Teo,Daume Y.Aloimono	Corpus-guided sentence generation of natural images (2011)	This model produces sentencesthat are both relevant and readabale	It cannot generate variable length captions
3	Shanshan zhao,lixiang li,haipeng peng,zihanyang,Jiaxuan zhang	Image caption generation via unified retrieval and generation based method (2020)	This method achieves better performance in bothquantitative and qualitative evaluation	It does not show highaccuracy
4	Vishwash Batra, Yulan He,George Vogiatzis	Neural caption Generation for News Images in applied science (2018)	This model is betterthan encoder- decoder models	It cannot work on largerdatasets
5	Justin Johnson,A. Karpathy,LiFei-Fei	Fully Convolutional Localization Networks for Dense Captioning (2015)	This method generate captions for regions present in the image.	It does not show highaccuracy

Table 2.1 : Literature Survey Works

2.2 EXISTING SYSTEM

Existing System of Image captioning include Retrieval based image captioning, template based image captioning.In retrieval-based approaches,captions are retrieved from a set of existing captions.In this approach,we will provide a query image and retrieval based method first find the visually similar images with their captions from the training data set. The caption generated in this approach can be one of those retrieved sentences or combination of those retrieved sentences.Another approach used here is template based image captioning.In this method,we have a fixed templates having a number of fixed blank slots to generate captions.The different features,surroundings of the objects are detected firstly and then these features are used to fill in the blanks of those fixed templates. This makes the template relevant to the image provided.The existing model was not done on deep learning. The proposed model achieved 90% accuracy, higher than the existing model which gives 50% and 60% accuracy

2.3 DRAWBACKS OF EXISTING SYSTEM:

1. It is not much efficient than our proposed system.
2. It is hard to generate caption for an image.
3. It is a time taking process.

2.4 FEASIBILITY STUDY:

The main objective of feasibility study is to test the Technical and Operational for adding new modules and debugging old running system. We have

1. Technical Feasibility
2. Operational Feasibility
3. Economic Feasibility

2.4.1 TECHNICAL FEASIBILITY:

The first step is that the organization has to decide that what technologies are suitable to develop and considering the existing system. Here in this application used technology is PYTHON. These are all open source softwares.

2.4.2 OPERATIONAL FEASIBILITY:

Not only must an application make Technical sense, it must also operational sense. Very often you will lead to improve the existing operation, maintenance and support Infrastructure to support the new application that you intended to develop. We can say after studying the proposed system that the project looks operationally Feasible, as there is a measure of how will a proposed system solve the problem and it satisfy the requirements. our project does not need extra training. As the application has been built constructing on the easy way to use the GUI.

2.4.3 ECONOMIC FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour in to research and development of the system is limited.

CHAPTER 3

PROPOSED SYSTEM

3.PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

In our proposed system,we mainly use two deep learning techniques those are CNN and LSTM.Which are mainly used for Image classification.The Proposed Methodology employs a Convolutional Neural Network to extract information from an image.Long short term memory (LSTM) generates visual description using information from convolutional neural networks The main goal of this proposed system is to generate real time captions for each input image This model has been given a variety of images to test whether the model is producing the correct output/caption for images or not.

3.2 METHODOLOGY OF IMAGE CAPTION GENERATOR

The proposed project consists of a new set of deep learning algorithms which are listed in a step wise process.

1. **Step 1:** Dataset is collected from various source of internet.Here,we have used a dataset called flickr8k which is collected from kaggle.This dataset contains 8000 images for which each image has 5 captions.
2. **Step 2:** Data preprocessing is done in this step includes Data cleaning,Data reduction,Image data preparation For instance,punctuations,digits,single length words are removed from the text dataset.
3. **Step 3:** Rendering of two models which have been selected i.e,CNN and LSTM will be done in this step.Firstly,CNN takes image as input and extract features such as background,objects in the image.
4. **Step 4:** A pre trained model called xception is used to train LSTM which will generate captions.
5. **Step 5:** Features which were extracted by CNN are given to LSTM.This LSTM will Generate captions for the given image
6. **Step 6:** A caption will be generated for the given image.

3.3 ADVANTAGES OF IMAGE CAPTION GENERATOR

1.Convolutional Neural Networks (CNN) automatically detects the important features without any human supervision.

2.CNN is more efficient than previously proposed approaches

3.LSTM'S retain information for longer periods compared to traditional RNN'S

3.4 APPROACHES USED IN IMAGE CAPTION GENERATOR

3.4.1 DEEP LEARNING

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.In deep learning, a computer model learns to perform classification tasks directly from images,text,or sound.Deep learning models can achieve accuracy, sometimes exceeding human-level performance.Models are trained using a large set of labeled data and and neural network architectures that contain many layers.

3.4.2 NEURAL NETWORKS

Neural networks are artificial systems that were inspired by biological neural networks. These systems learn to perform tasks by being exposed to various datasets and examples without any task-specific rules. The idea is that the system generates identifying characteristics from the data they have been passed without being programmed with a pre-programmed understanding of these datasets. Neural networks are based on computational models for threshold logic. Threshold logic is a combination of algorithms and mathematics. Neural networks are based either on the study of the brain or on the application of neural networks to artificial intelligence. Components of a typical neural network involve neurons, connections which are known as synapses, weights, biases, propagation function, and a learning rule.

3.4.3 CNN

CNN stands for Convolutional Neural Networks.It is a deep learning algorithm which takes image as an input.CNN scans images from left to right and top to bottom to pull out important features from the image and combines the features to classify images.Pre-processing required in convolutional neural networks is much lower as compared to other classification algorithms.CNN is used to extract features from the image.A pre-trained model called Xception is used for this.CNN can be used in the fields of image recognition,image classification, and natural language processing.

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.While in primitive methods filters are hand-engineered,with enough training, CNN have the ability to learn these filters/characteristics.

There are 3 Layers in CNN

1. Convolutional Layer
2. Pooling Layer
3. Fully Connected Layer

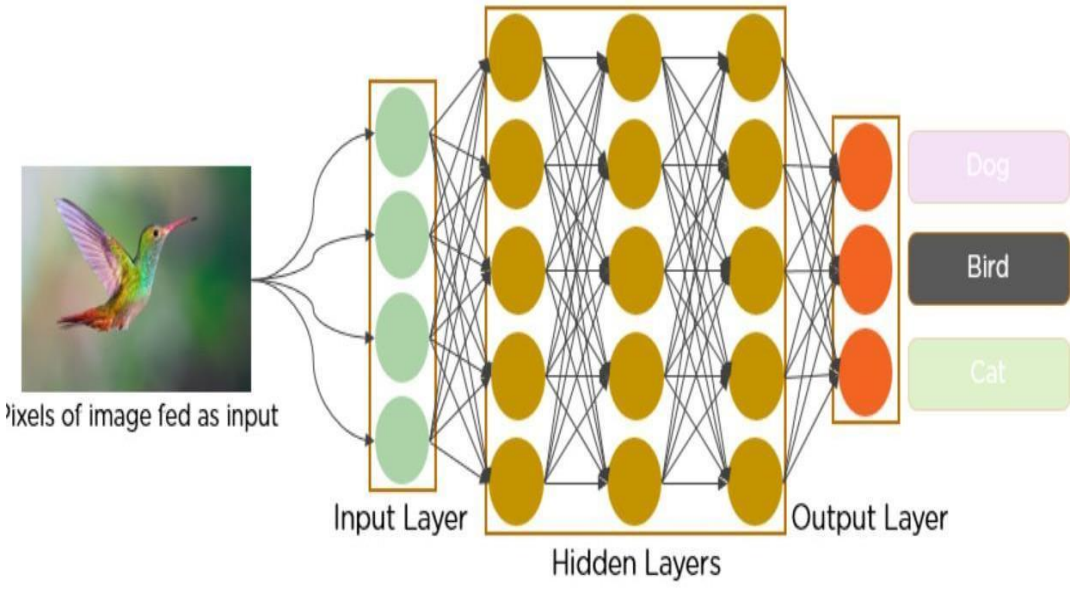


Figure 3.1: Layers of CNN

The architecture of a Conv Net is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area. The CNNs are inspired by visual system of human brain. The idea behind the CNNs thus is to make the computers capable of viewing the world as humans view it. This way CNNs can be used in the fields of image recognition and analysis, image classification, and natural language processing. CNN is a type of deep neural networks which contain the convolutional, max pooling, and activation layers. The convolutional layer, considered as a main layer of a CNN, performs the operation called “convolution” that gives CNN its name. Kernels in the convolutional layer are applied to the layer inputs. All the outputs of the convolutional layers are convolved as a feature map. In this study, the Rectified Linear Unit (ReLU) has been used in the activation function with a convolutional layer which is helpful to increase the non-linearity in input image, as the images are fundamentally nonlinear in nature. Thus, CNN with ReLU activation function is easier and faster. Since the ReLU is zero for all negative inputs, it can be defined as

$z=\max(0, I)$

RELU function implies that z is 0 for all negative values and non-zero for all positive values.

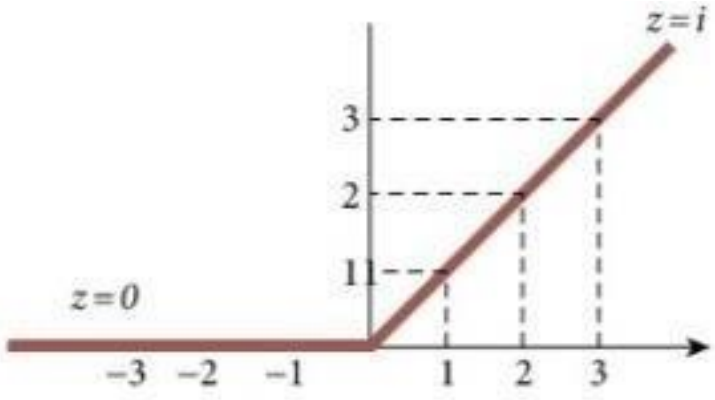


Figure 3.2: ReLU Function

The pooling layer is an important building block of CNN. Pooling can be the max, average, and sum in the CNN model. In this study, max pooling has been used because others may not identify the sharp features easily as compared to max pooling. The dropout layer has also been used, which drops the neurons during the training chosen at random to reduce the overfitting problem. Towards the last stage of the CNN used in the study, there is a flatting layer to convert the output of convolutional layers into a single-dimensional feature vector. After flattening, the vector data is given as an input to the next layers of the CNN called fully connected layers or dense layers. The main functionality of dense layers is to take flattened output results from the convolution and pooling layers and as input and classify the image to a specific class label.

3.4.3.1 LAYERS OF CNN CONVOLUTIONAL LAYER

The convolutional layer is the key component of convolutional neural networks, and is always at least their first layer. Its purpose is to detect the presence of a set of features in the images received as input. This is done by convolution filtering: the principle is to “drag” a window representing the feature on the image, and to calculate the convolution product between the feature and each portion of the scanned image. A feature is then seen as a filter: the two terms are equivalent in this context. The convolutional layer thus receives several images as input, and calculates the convolution of each of them with each filter. The filters correspond exactly to the features we want to find in the images. We get for each pair (image, filter) a feature map**,** which tells us where the features are in the image: the higher the value, the more the corresponding place in the image resembles the feature.

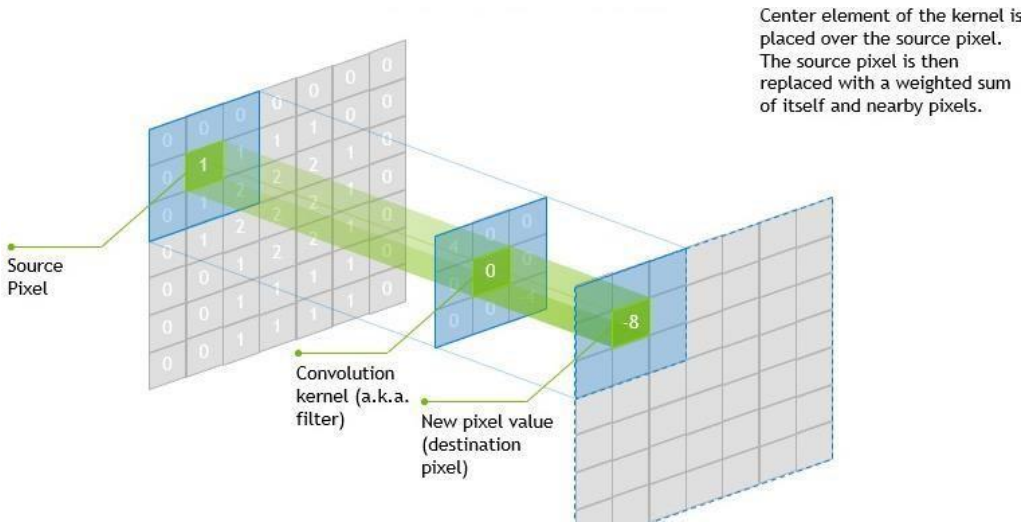


Figure 3.3 : Convolutional Layer

Unlike traditional methods, features are not pre-defined according to a particular formalism (for example SIFT), but learned by the network during the training phase! Filter kernels refer to the convolution layer weights. They are initialized and then updated by back propagation using gradient descent.

3.4.3.2 DROPOUT LAYER

The term "dropout" is used for a technique which drops out some nodes of the network. Dropping out can be seen as temporarily deactivating or ignoring neurons of the network. This technique is applied in the training phase to reduce overfitting effects. Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

As a neural network learns, neuron weights settle into their context within the network. Weights of neurons are tuned for specific features providing some specialization. Neighbouring neurons become to rely on this specialization, which if taken too far can result in a fragile model too specialized to the training data. This reliant on context for a neuron during training is referred to complex co-adaptations. Dropout is easily implemented by randomly selecting nodes to be dropped-out with a given probability. (e.g. 20%) each weight update cycle. This is how Dropout is implemented in Keras. Dropout is only used during the training of a model and is not used when evaluating the skill of the model.

In machine learning, regularization is way to prevent over-fitting. Regularization reduces over-fitting by adding a penalty to the loss function. By adding this penalty, the model is trained such that it does not learn interdependent set of features weights. Those of you who know Logistic Regression might be familiar with L1 (Laplacian) and L2 (Gaussian) penalties. Dropout is an approach to regularization in neural networks which helps reducing interdependent learning amongst the neurons.

3.4.3.3 FEATURES

1. Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

2. Dropout roughly doubles the number of iterations required to converge. However, training time for each epoch is less.

3. 3.With H hidden units, each of which can be dropped, we have 2^H possible models. In testing phase, the entire network is considered and each activation is reduced by a factor p.

3.4.3.4 DENSE LAYER

We cannot pass output of convolutional layer directly to the dense layer because output of convolutional layer is in multi-dimensional shape and dense layer requires input in single-dimensional shape i.e.,1-D array. Each Layer in the Neural Network contains neurons, which compute the weighted average of its input and this weighted average is passed through a non-linear function, called as an "activation function". Result of this activation function is treated as output of that neuron. In similar way, the process is carried out for all neurons of all layers. The output of the last layer will be considered as output for that image. In same way, we will pass all the images as to convolutional layer and then to the Neural Network, which will produce corresponding outputs for those images. summary() .method will display the architecture of the model.

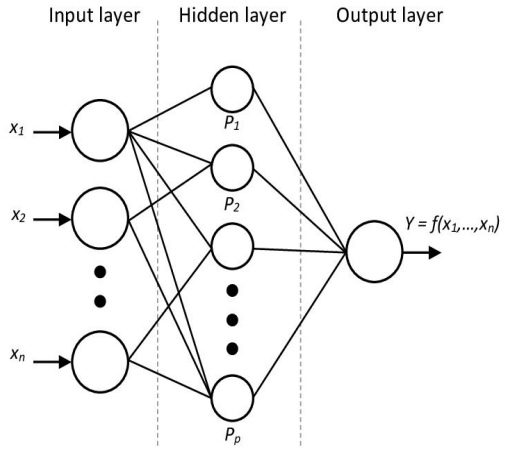


Figure 3.4 : Dense Layer

3.4.4 LSTM

LSTM stands for long short-term memory, it is a type of RNN(Recurrent Neural Networks.LSTM is capable of working with sequence prediction problems. It is used for word prediction purposes.In LSTM based on previous text, we can predict what the next word will be. It is the same as google search where our system will show the next word based on our previous text.LSTM can carry out relevant information throughout the process with a forget gate and discards non-relevant information.LSTM will use the information that is extracted from CNN to generate a description of the image.

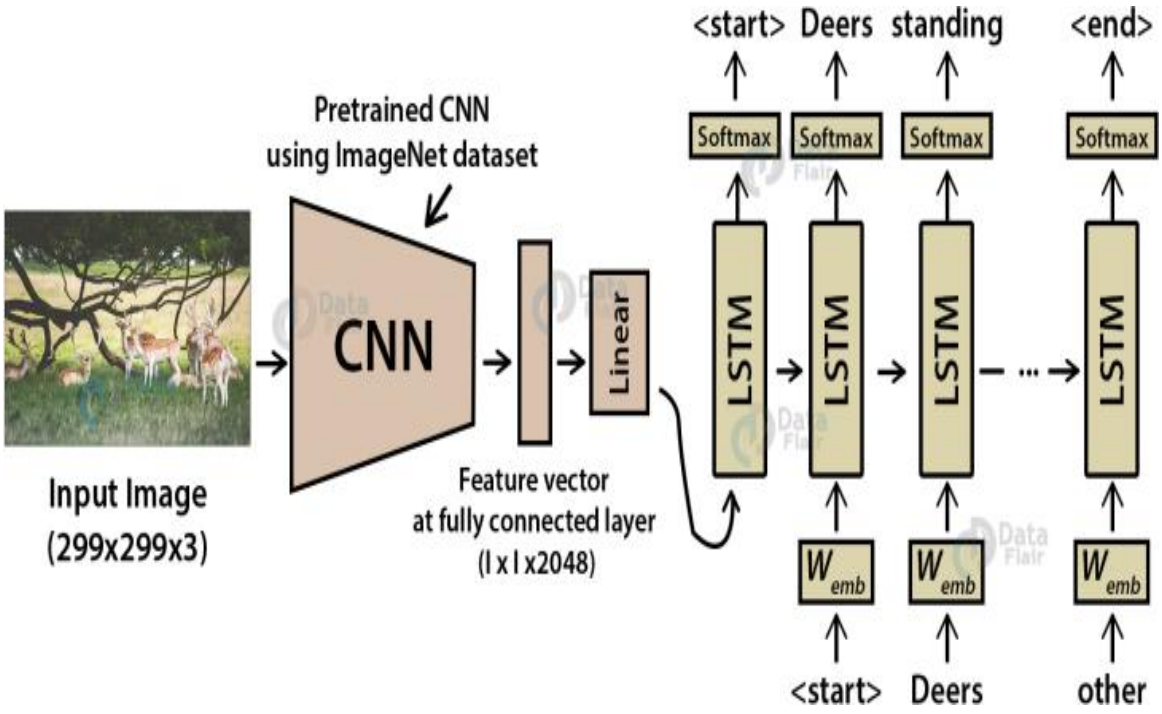


Figure 3.5: Model Image Caption Generator

LSTMs use gated cells to store information. With these cells, the network can manipulate the information in many ways, including storing information in the cells and reading from them. The cells are individually capable of making decisions regarding the information and can execute these decisions by opening or closing the gates. The ability to retain information for a long period of time gives LSTM the edge over traditional RNNs in these tasks. The chain-like architecture of LSTM allows it to contain information for longer time periods, solving challenging tasks that traditional RNNs struggle to or simply cannot solve. The three major parts of the LSTM include:

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

Input gate - Responsible for adding information to the cells

Output gate - Selects and outputs necessary information

The CNN LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. This architecture was originally referred to as a Long-term Recurrent Convolutional Network or LRCN model, although we will use the more generic name "CNN LSTM" to refer to LSTMs that use a CNN as a front end in this lesson. This architecture is used for the task of generating textual descriptions of images. Key is the use of a CNN that is pre-trained on a challenging image classification task that is re-purposed as a feature extractor for the caption-generating problem.

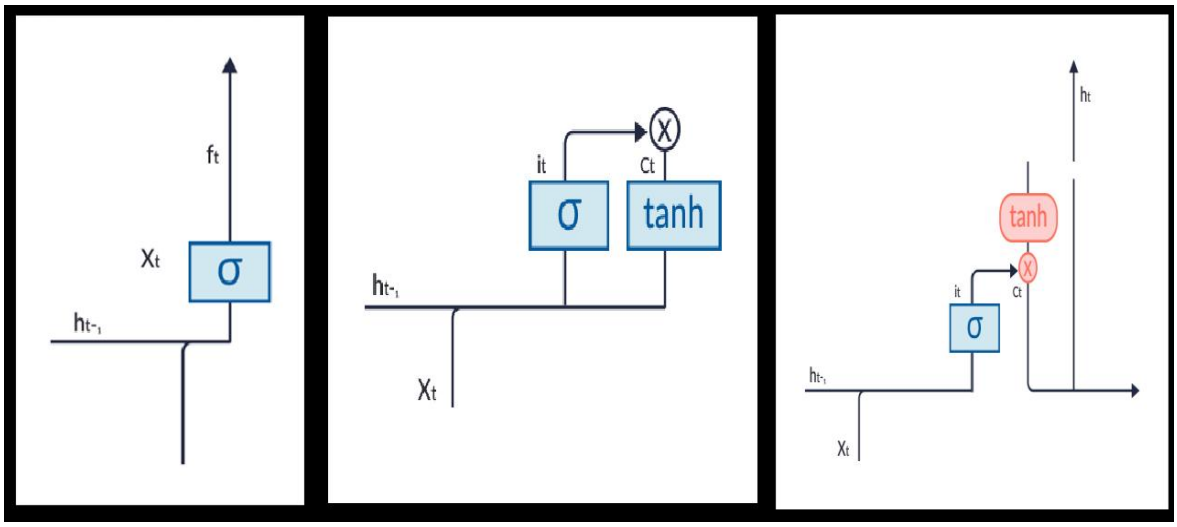


Figure 3.6: Forget gate

CHAPTER 4

REQUIREMENT SPECIFICATION

4.SOFTWARE REQUIREMENT SPECIFICATION

4.1 HARDWWARE REQUIREMENTS

RAM: 4GB

Processor: Core i5

Hard drive: 50 GB of HDD space

4.2 SOFTWARE REQUIREMENTS:

Operating system: Windows 10

Programming Language: PYTHON

Tool: Google Colab

4.3 TECHNOLOGIES USED:

- Python
- Deep Learning

4.4 FUNCTIONAL REQUIREMENTS:

The Functional Requirements will describe the features and functionality of the system. Functional Requirements record the operation that must be done. Functional Requirements are based for non-functional requirements. Outputs from computer systems are required primarily to communicate the results of processing to user. They are also used to provide a permanent copy of the result for later consultation. It also depends upon the type of software, expected users and the type of system where the software is used.The various types of outputs in general are:

- External outputs, whose destination is outside the organization.
- Internal outputs whose destination is within organization.
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.

4.5 NON-FUNCTIONAL REQUIREMENTS:

In systems engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The nonfunctional requirements canbe considered as quality attributes of a system.

Reliability: Users have to trust the system, even after using it for a long time. Your goal should be a long MTBF (mean time between failures). Create a requirement that data created in the system will be retained for number of years without the data being changed by the system.

Performance: The way in which the system meets its performance target is for it to be specified clearly and explicitly. The system itself might not need anything specifically for its basic operation, but the complete system along with components connected may have some Performance requirements.

Scalability: The ability for a business or technology is to accept increase volume without the impacting the system.

Usability: Usability determines how difficult it is to learn use the system. Prioritize the important functions of the system based on usage patterns.

Availability: It means for how long the system is available for its users and for how long the system will be operational.

Error Detection: Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors are always likely to occur, these types of errors can be discovered by using validations to check the input data.

Data Validation:.Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors.

CHAPTER 5

SYSTEM DESIGN

5.SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

This is how our proposed system architecture will look like

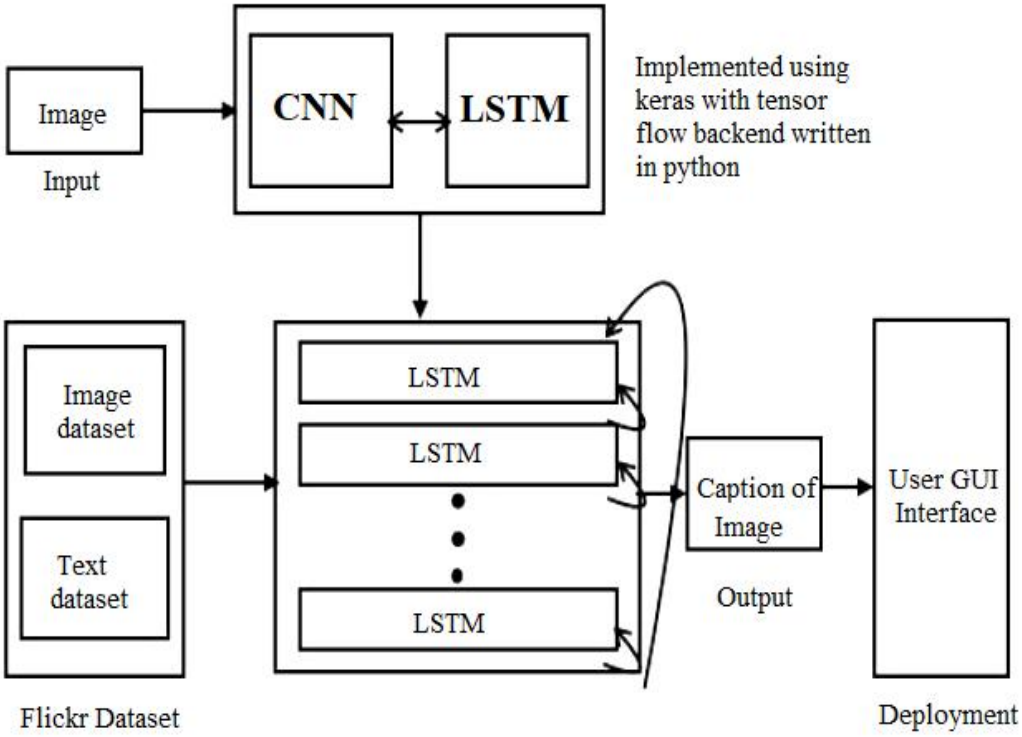


Figure 5.1: System Architecture of Image Caption Generator

In this system architecture,firstly we upload an image as input this image is forwarded to CNN in which it extracts the features such as background,scene,objects in the image using convolutional layer and pooling layer then these features are sent to LSTM by using fully connected layer.Now,the dataset which contains Image Dataset and text dataset is preprocessed to form an training model.This training model is used to train LSTM for which it generates captions.This generated captions are diplayed on screen using GUI Interface.This whole project is implemented using keras with tensorflow backend written in python.

5.2 DATA FLOW DIAGRAM(DFD)

A Data flow diagram shows the way,information flows through a system.It includes data inputs and outputs,data stores,and the various sub-processes the data moves through.Data flow diagrams visually represent systems and processes that would be hard to describe in just words.Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.Data flow diagrams provide us the basic overview of the whole Image caption generator system or process being analysed or modelled.Here we have shown the DFD's of our system.

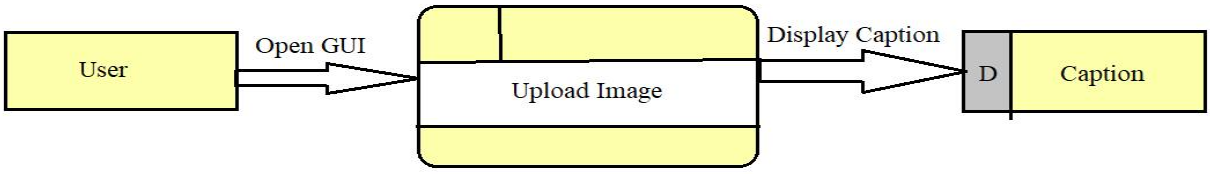


Figure 5.2: DFD Diagram Level 0

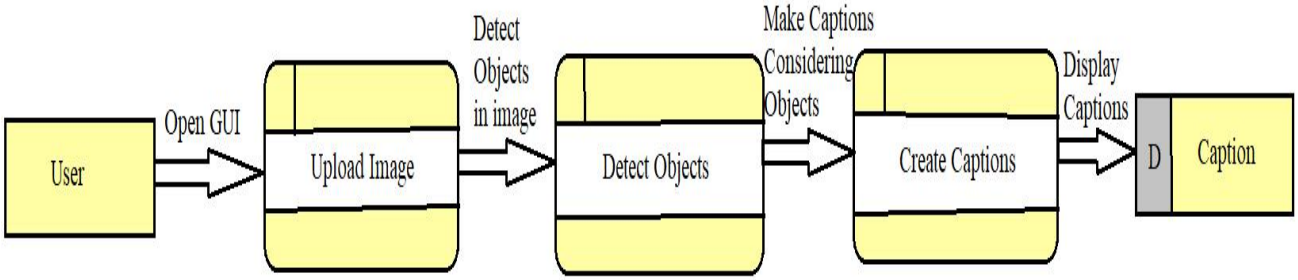


Figure 5.3: DFD Diagram Level 1

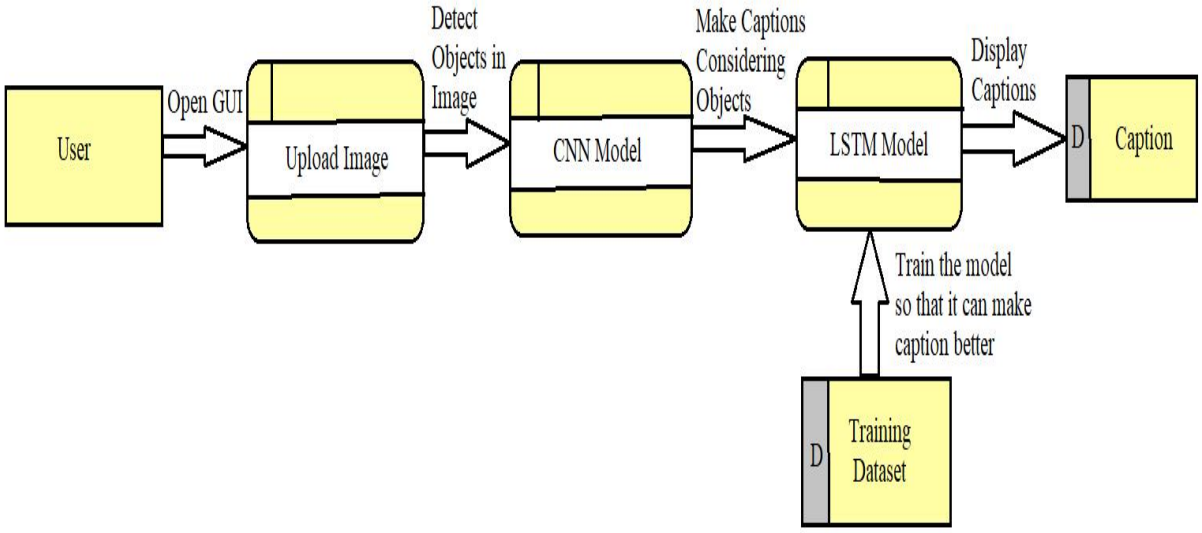


Figure 5.4: DFD Diagram Level 2

5.3 FLICKR8K DATASET

Flickr8k dataset is a public benchmark dataset for image to sentence description. This dataset consists of 8000 images with five captions for each image. These images are extracted from diverse groups in Flickr website. Each caption provides a clear description of entities and events present in the image. The dataset depicts a variety of events and scenarios and doesn‟t include images 37 containing well-known people and places which makes the dataset more generic. The dataset has 6000 images in training dataset, 1000 images in development dataset and 1000 images in test dataset. Features of the dataset making it suitable for this project are:

- Multiple captions mapped for a single image makes the model generic and avoids overfitting of the model.
- Diverse category of training images can make the image captioning model to work for multiple categories of images and hence can make the model more robust.

5.4 MODULE DESCRIPTION

The modules listed in this project are as follows:

1.System

1.1 Dataset Creation:

The dataset consists of 8000 images with five captions for each image.Each caption provides a clear description of events present in the image.

1.2 Pre-processing:

Pre-processing involves Image data preparation,caption data preparation,data cleaning and reshaping of images to appropriate format to train our model.

1.3 Training:

The pre-processed training dataset is used to train our model using LSTM algorithm and generates suitable captions for the image

1.4 Caption Generation:

The results of our model is to generate a suitable caption for an image.

2.User

2.1 Upload Image:

The user has to upload an image for which the caption has to be generated.

2.2 View results:

The generated caption for the image is viewed by the user.

5.5 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general purpose modeling language in the field of object-oriented software engineering.The goal is for UML to become a common language for creating models of object oriented computer software. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system,as well as for business modeling and other non-software systems.The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects

5.5.1 USECASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

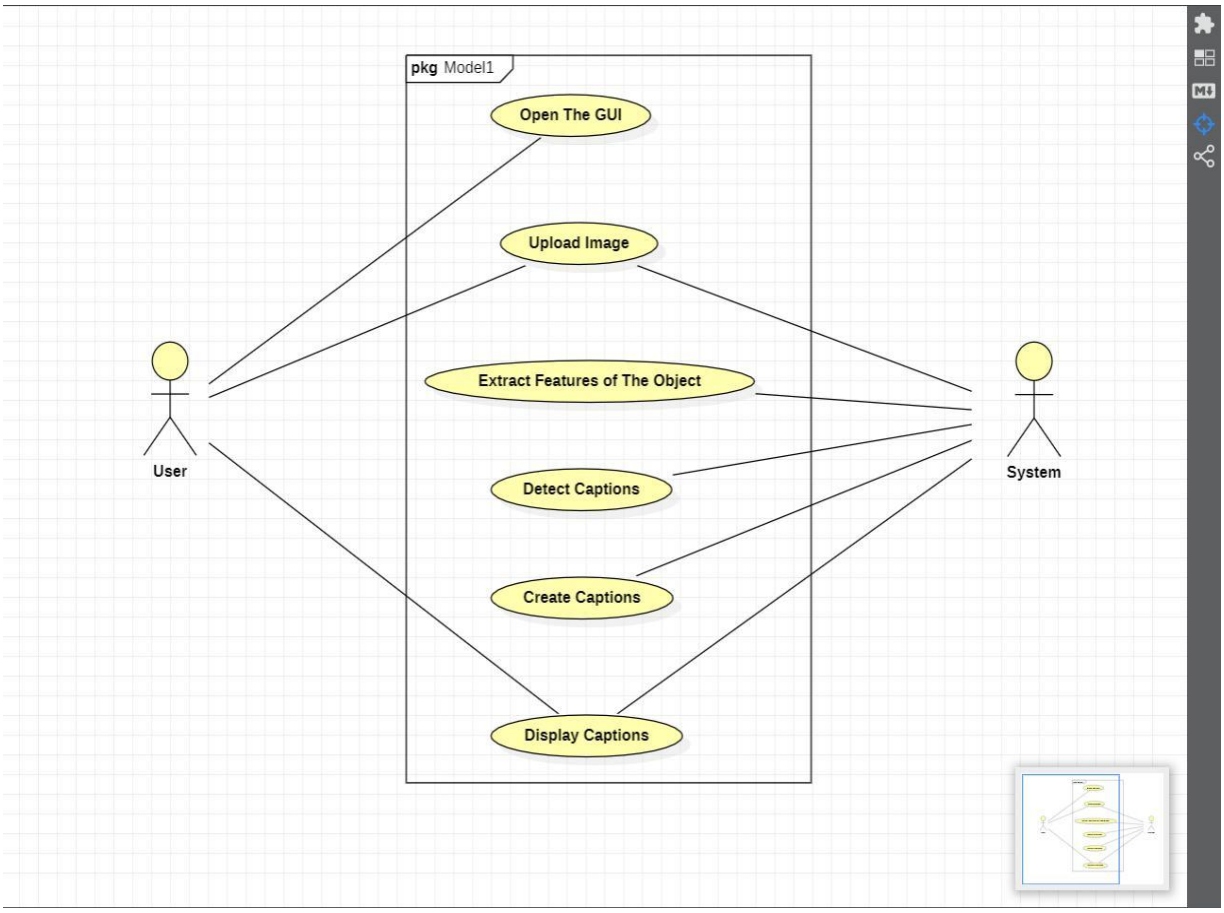


Figure 5.5: Usecase Diagram of Image Caption Generator

DESCRIPTION

Here we have two actors user and system.Firstly,user Opens the GUI and uploads image.System takes this image and extract features using CNN and sends this features to LSTM Which detect captions from this features,create captions and displays these captions to the user.

5.5.2 SEQUENCE DIAGRAM

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

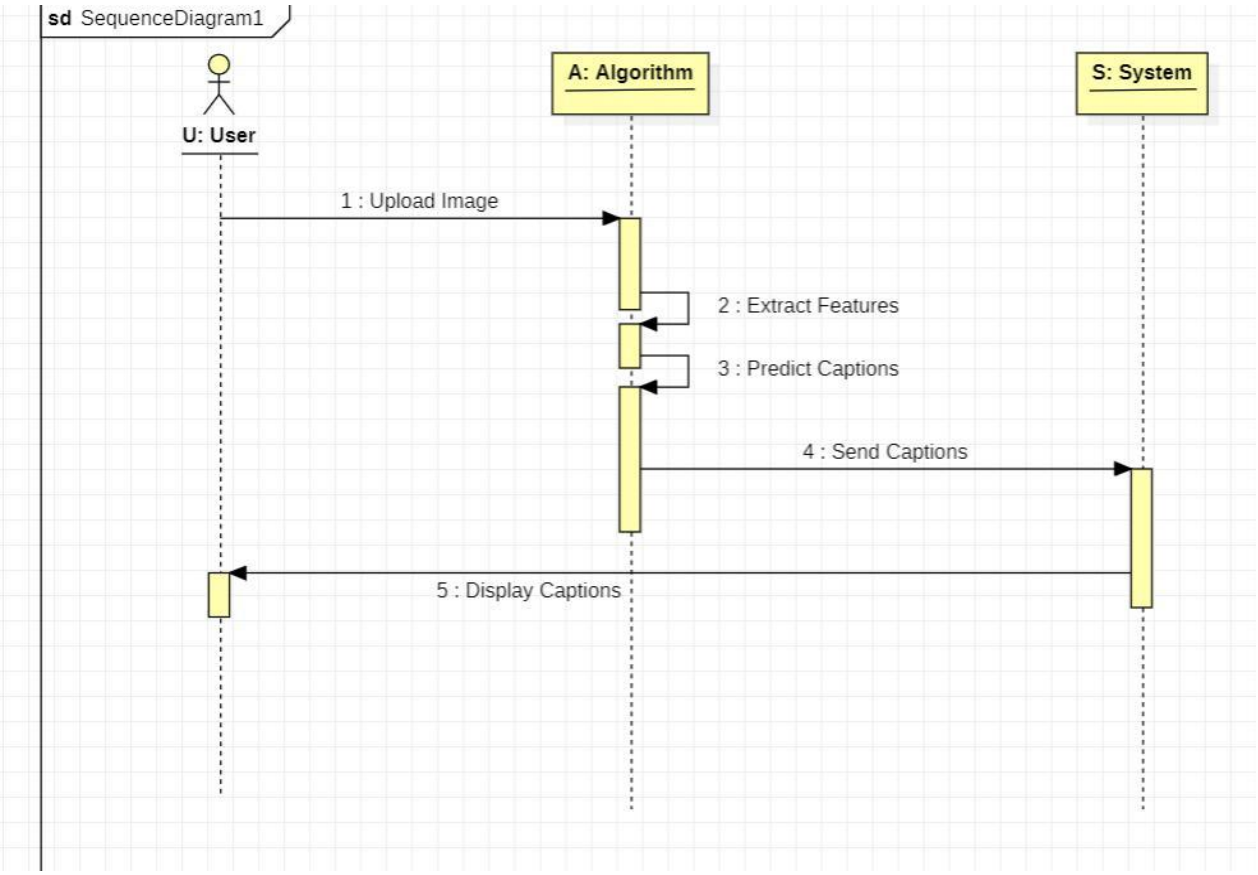


Figure 5.6: Sequence Diagram of Image Caption Generator

DESCRIPTION

In this sequence diagram we have an actor and objects like algorithm and system.Firstly,user uploads the image then algorithm takes this message and extract features of the image using CNN,then algorithm sends this features to predict captions using LSTM.The created captions are sent to the system and from system,captions are displayed to the user.This diagram mainly focuses on time ordering of messages.

5.5.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.Class diagram describes the attributes and operations of a class and also the constraints imposed on the system.The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

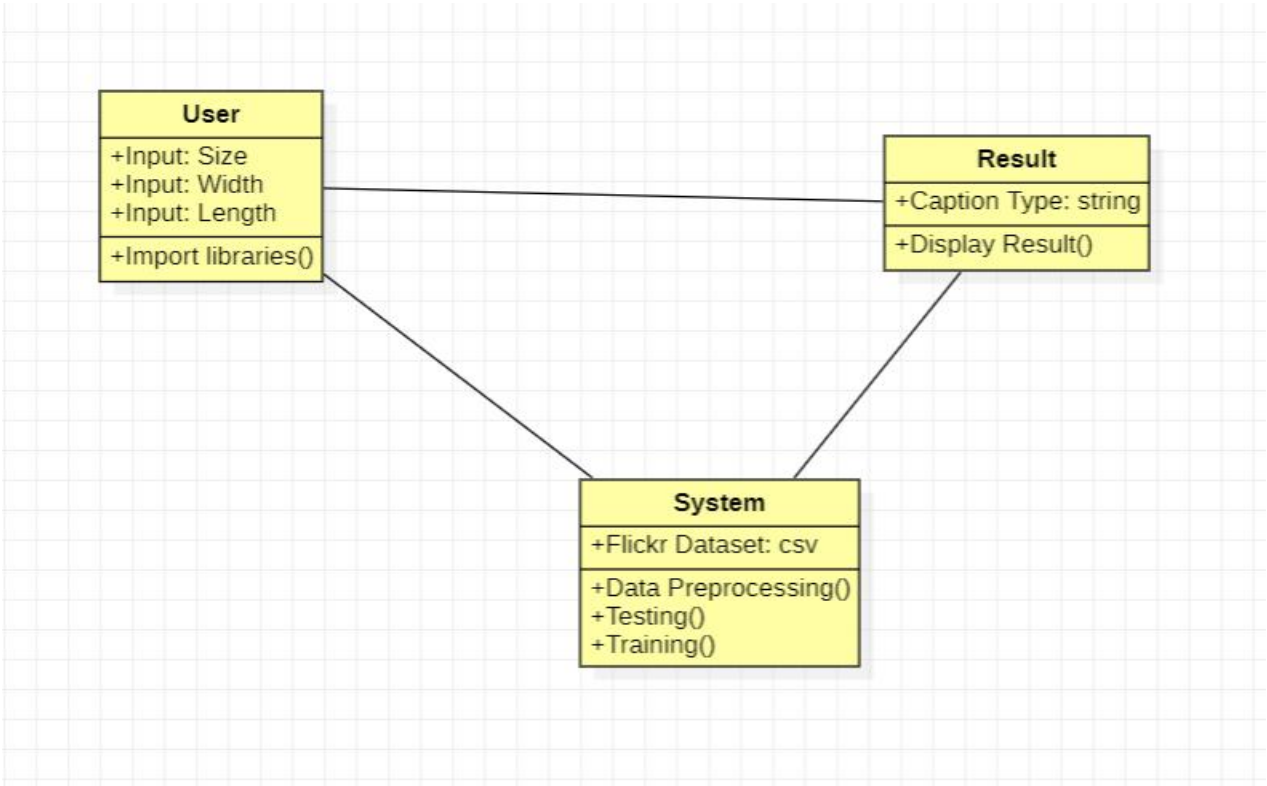


Figure 5.7: Class Diagram of Image Caption Generator

DESCRIPTION

In the below class diagram, system class will have three operations including Data preprocessing,training the data and testing data. Data tested in testing phase is used to predict the captions for given image.

5.5.4 STATECHART DIAGRAM

State chart diagram is a behavioural diagram in UML.It represents the behaviour using finite state transitions.A state chart diagram is used to represent the condition of the system at finite instances of time. State chart diagrams are used to model the dynamic behaviour of the system.These are designed to understand the reaction of object to internal or external stimuli.

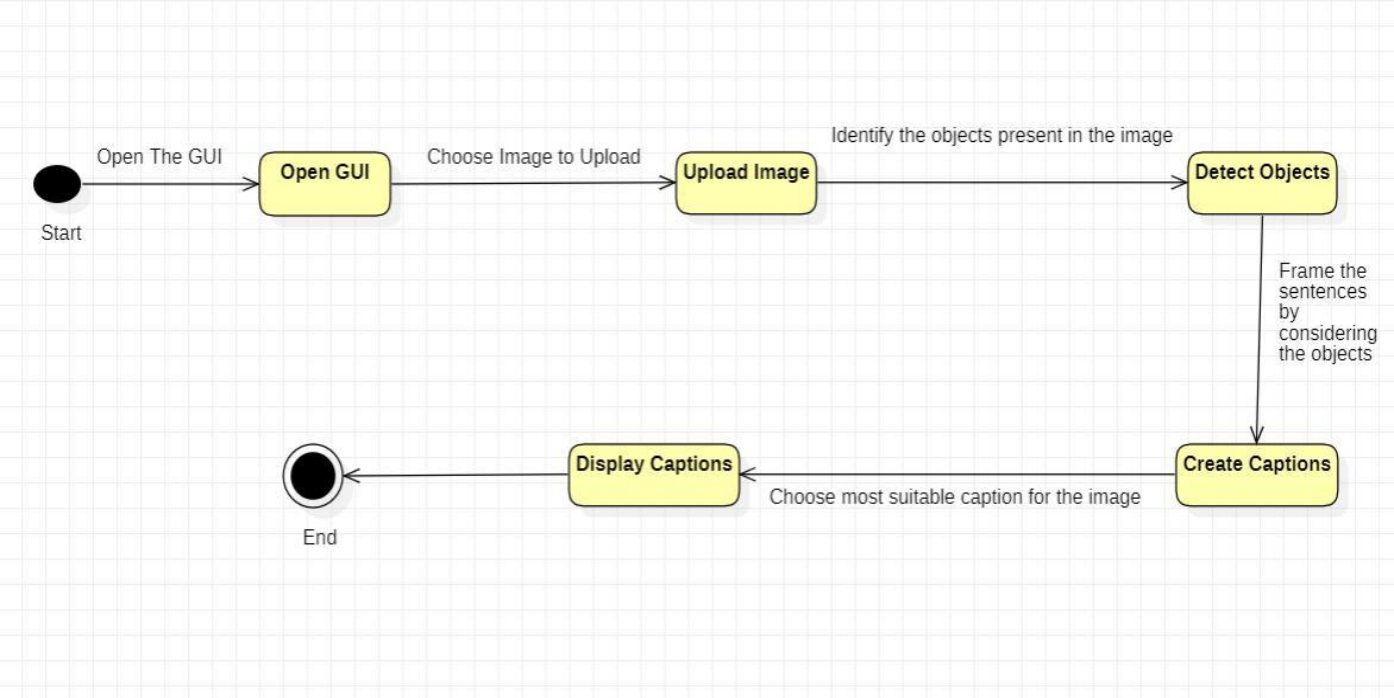


Figure 5.8: Statechart Diagram of Image Caption Generator

DESCRIPTION

The Figure 5.6 shows State Chart Diagram of the system.First user will browse the site. Then he will upload the image, CNN will identify the objects present in the image then LSTM will start preparing captions considering the objects present in the image using Training Dataset, which comprises of Image Data set and Text Data Set, after the training a suitable caption will be generated and displayed top the user.

CHAPTER 6

IMPLEMENTATION

6. IMPLEMENTATION

6.1 IMPLEMENTATION

This project requires good knowledge of Deep learning, Python, working on Jupyter notebooks,Keras library, Numpy, and Natural language processing. Make sure you have installed all the following necessary libraries:

6.1.1 PYTHON:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

6.1.2 PACKAGES IMPORTED:

NumPy : NumPy main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called axes. The number of axes is rank.

- Offers Mat lab capabilities within Python
- Fast array operations
- 2D arrays, multi-D arrays, linear algebra etc.

Tensorflow: TensorFlow is an open source framework developed by Google researchers to run machinelearning, deep learning and other statistical and predictive analytics workloads. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers.

Keras: Keras is a high-level, deep learning API developed by Google for implementing neural networks.It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras is relatively easy to learn and work with because it provides a python front-end with a high level of abstraction while having the option of multiple back-ends for computation purposes.

Pillow: Python Imaging Library (PIL) is the image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including PNG,JPEG,TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

Pickle: Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk . What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

cv2: cv2 is the module import name for open CV-python. OpenCV is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including object detection, face recognition, and tracking.

tqdm: tqdm is a library in Python which is used for creating Progress Bars. When ever we install Python library or installing software, we see the progress bar on screen, which estimates how long the process would take to complete this can be achieved by tqdm.

6.2 SAMPLE SOURCE CODE

```
from google.colab import drive
drive.mount("/content/drive/")
!pip install Keras-Preprocessing
import string
import numpy as np
from PIL import Image
import os
import tensorflow as tf
tf.get_logger().setLevel("ERROR")
from pickle import dump, load
import matplotlib.pyplot as plt
import cv2
import io
from tensorflow.keras.applications.xception import Xception
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Input, Dense, Dropout, Embedding, LSTM, concatenate
from tensorflow.keras.models import load_model, Model
from tqdm.notebook import tqdm
tqdm().pandas()
os.listdir('/content/drive/MyDrive/Project Dataset/Flickr8k_text')
dataset_images_path = '/content/drive/MyDrive/Project Dataset/Flicker8k_Dataset'
dataset_text_path = '/content/drive/MyDrive/Project Dataset/Flickr8k_text'
def load_txt(path):
    file = open(path, 'r')
    content = file.read()
    file.close()
    return content
```

```
# load_txt(dataset_text_path)
def make_dic(path):
    content = load_txt(path)
    lines = content.split('\n')
    dic = {}
    for line in lines[:-1]:
# print(line)
        img_name, caption = line.split('\t')
        img_name = img_name[:-2]
        if img_name not in dic.keys():
            dic[img_name] = [caption]
        else:
            dic[img_name].append(caption)
    return dic
def clean_caption(dic):
    filter_table = str.maketrans('', '', string.punctuation)
    for img_name, caption_list in dic.items():
        for i, caption in enumerate(caption_list):
            caption = caption.replace('-', ' ') # from "truck-driver" to "truck driver"
            word_list = caption.split()
            word_list = map(lambda x: x.lower().translate(filter_table), word_list)
            word_list = [word for word in word_list if len(word) > 1 and word.isalpha()]
           # remove numeric words and "a" character
            caption = ' '.join(word_list)
            dic[img_name][i] = caption
    return dic
def extract_vocabulary(dic):
    vocab = set()
    for k in dic.keys():
        [vocab.update(d.split()) for d in dic[k]]
return vocab
```

```
def save_dic_to_text(dic, path):
    lines = list()
    for img_name, caption_list in dic.items():
        for caption in caption_list:
            lines.append(img_name + '\t' + caption)
    data = '\n'.join(lines)
    file = open(path, 'w')
    file.write(data)
    file.close()
descriptions = make_dic(dataset_text_path + '/' + 'Flickr8k.token.txt')
cleaned_descriptions = clean_caption(descriptions)
vocabulary = extract_vocabulary(cleaned_descriptions)
save_dic_to_text(cleaned_descriptions, 'cleaned_descriptions.txt')
def extract_feature(path):
    model = Xception(include_top=False, pooling='avg')
    features = {}
    for img_name in tqdm(os.listdir(path)):
        img_path = path + '/' + img_name
        img = Image.open(img_path)
        img = img.resize((300, 300))
        img = np.array(img)
        img = np.expand_dims(img, axis=0)
        img = img / 127.5 - 1.0
        feature = model.predict(img, verbose=0)
        features[img_name] = feature
    return features
features = extract_feature(dataset_images_path)
print("Extracted features: ",len(features))
dump(features, open('features.p', 'wb'))
features = load(open('features.p', 'rb'))
```

```
def load_img_name_list(path):
    content = load_txt(path)
    img_name_list = content.split('\n')[:-1]
    return img_name_list
def load_cleaned_descriptions(path, img_name_list):
    content = load_txt(path)
    dic = {}
    for line in content.split('\n'):
        word_list = line.split()
        if len(word_list) < 1:
            continue
        img_name, caption_word_list = word_list[0], word_list[1:]
        if img_name in img_name_list:
            if img_name not in dic:
                dic[img_name] = []
            caption = '<start> ' + ' '.join(caption_word_list) + ' <end>'
            dic[img_name].append(caption)
    return dic
def load_features(img_name_list):
    all_features = load(open('features.p', 'rb'))
    features = { k: all_features[k] for k in img_name_list }
    return features
train_img_list = load_img_name_list(dataset_text_path + '/' + 'Flickr_8k.trainImages.txt')
train_descriptions = load_cleaned_descriptions('cleaned_descriptions.txt', train_img_list)
train_features = load_features(train_img_list)
def extract_caption(dic):
    all_captions = list()
    for k in dic.keys():
        [all_captions.append(caption) for caption in dic[k]]
    return all_captions
```

```
def create_tokenizer(dic):
    all_captions = extract_caption(dic)
    tokenizer = Tokenizer(oov_token='<oov>')
    tokenizer.fit_on_texts(all_captions)
    return tokenizer
tokenizer = create_tokenizer(train_descriptions)
dump(tokenizer, open('tokenizer.p', 'wb'))
vocab_size = len(tokenizer.word_index) + 1
def get_max_length_of_caption(dic):
    all_captions = extract_caption(dic)
    return max(len(d.split()) for d in all_captions)
max_length = get_max_length_of_caption(train_descriptions)
def data_generator(dic, features, tokenizer, max_length, vocab_size, batch_size):
    while True:
        n = 0
        X1, X2, y = list(), list(), list()
        for img_name, caption_list in dic.items():
            feature = features[img_name][0]
            for caption in caption_list:
                sequence = tokenizer.texts_to_sequences([caption])[0]
                for i in range(1, len(sequence)):
                    in_seq, out_seq = sequence[:i], sequence[i]
                    in_seq = pad_sequences([in_seq], maxlen=max_length)[0]
                    out_seq = to_categorical([out_seq], num_classes=vocab_size)[0]
                    X1.append(feature)
                    X2.append(in_seq)
                    y.append(out_seq)
                    n += 1
                    if n == batch_size:
                        X1, X2, y = np.array(X1), np.array(X2), np.array(y)
                        yield [X1, X2], y
```

```
                n = 0
                X1, X2, y = list(), list(), list()
def define_model(total_words, max_length):
    input_1 = Input(shape=(2048,))
    fe1 = Dropout(0.5)(input_1)
    fe2 = Dense(256, activation='relu')(fe1)

    input_2 = Input(shape=(max_length,))
    se1 = Embedding(total_words, 256, mask_zero=True)(input_2)
    se2 = Dropout(0.5)(se1)
    se3 = LSTM(256, activation='relu')(se2)

    decoder1 = concatenate([fe2, se3])
    decoder2 = Dense(256, activation='relu')(decoder1)
    outputs = Dense(total_words, activation='softmax')(decoder2)

    model = Model(inputs=[input_1, input_2], outputs=outputs)
    model.compile(loss='categorical_crossentropy', optimizer='adam')
    model.summary()

    return model
import random
def caption_viewer(gref, capt):
  with open("/content/cleaned_descriptions.txt",'r') as file:
    # description_f = file.read()
    img_name = gref.split("/")
    img_list = img_name[-1].split(".")
    # print(img_list)
    desc_list = list()
    captions_list = list()
    for line in file:
```

```
        if img_list[0] in line:
            desc_list.append(line)
        for every in desc_list:
            var = every.split(" ")
            var = var[1:]
            sentence = " ".join(var)
            captions_list.append(sentence)
        print("start\n"+random.choice(captions_list)+"end")
model_LSTM = define_model(vocab_size, max_length)
epochs = 18
batch_size = 60
steps = len(train_img_list) // batch_size
for i in tqdm(range(epochs)):
#print('Big Epoch ' + str(i+1))
    generator = data_generator(train_descriptions, train_features, tokenizer, max_length, vocab_size,
    batch_size)
    model_LSTM.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)
model_LSTM.save('models/model_LSTM.h5')
def test_extract_features(path, model):
    try:
        img = Image.open(path)
    except:
        print('Incorrect path or extension!')
    img = img.resize((300, 300))
    img = np.array(img)
    if img.shape[2] == 4:
        img = img[..., :3]
    img = np.expand_dims(img, axis=0)
    img = img / 127.5 - 1.0  # img = preprocess_input(img)
    feature = model.predict(img, verbose=0)
    return feature
```

```
def test_word_of_id(idx, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == idx:
            return word
    return None
def test_generate_caption(model, tokenizer, feature, max_length):
    in_text = 'start'
    for i in range(max_length):
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        sequence = pad_sequences([sequence], maxlen=max_length)
        prediction = model.predict([feature, sequence], verbose=0)
        prediction = np.argmax(prediction)
        word = test_word_of_id(prediction, tokenizer)
        if word is None:
            break
        in_text += ' ' + word
        if word == 'end':
            break
    return in_text
def test_load_doc(path):
    file = open(path, 'r')
    text = file.read()
    file.close()
    return text
def test_load_img_name_list(path):
    text = test_load_doc(path)
    img_name_list = text.split('\n')
    return img_name_list
test_max_length = 35
tokenizer = load(open('tokenizer.p', 'rb'))
model_LSTM = load_model('models/model_LSTM.h5')
```

```
xception = Xception(include_top=False, pooling='avg')
test_img_name_list = test_load_img_name_list(dataset_text_path + '/' + 'Flickr_8k.testImages.txt')
import random
import sys
from google.colab.patches import cv2_imshow
random_test_image_path = dataset_images_path + '/' + random.choice(test_img_name_list)
random_test_image_path = "/content/drive/MyDrive/Project Dataset/Flicker8k_Dataset/1007320043_
627395c3d8.jpg"
test_feature = test_extract_features(random_test_image_path, xception)
image = Image.open(random_test_image_path)
caption = test_generate_caption(model_LSTM, tokenizer, test_feature, max_length)
print(random_test_image_path)
caption_viewer(random_test_image_path, caption)
plt.imshow(image)
```

CHAPTER 7

TESTING

7. TESTING

7.1 TESTING INTRODUCTION:

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole. So, in a nutshell, that is a very basic overview of the general software development life cycle model. Now let's delve into some of the traditional and widely used variations.

7.1.1 UNIT TESTING:

Unit testing is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. it is concerned with functional correctness of the stand alone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects. Unit Testing Techniques:

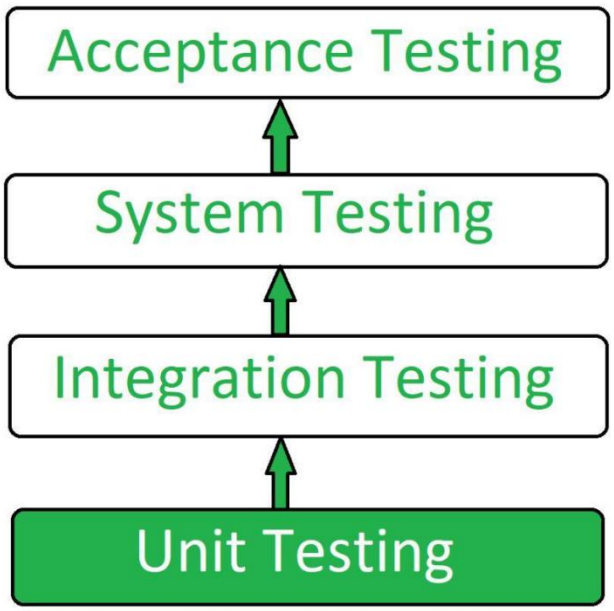


Figure 7.1: Phases of testing

Black Box Testing: Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester.

White Box Testing: White Box Testing is a testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security.

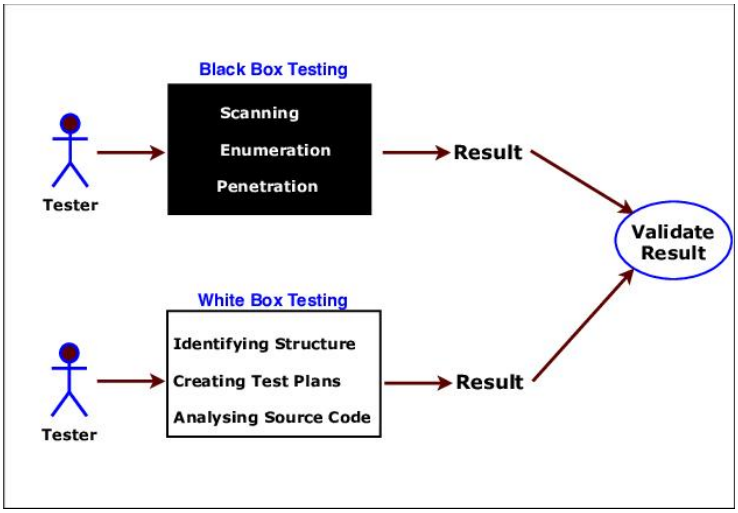


Figure 7.2: Unit testing techniques

7.1.2 INTEGRATION TESTING:

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test Stubs are used to assist in Integration Testing.

7.1.3 USER INTERFACE TESTING:

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI]. This usually means testing the visual elements to verify that they are functioning according to requirements - in terms of functionality and performance.

7.2 TEST RESULTS:

S.NO	INPUT	OUTPUT	RESULT
Test Case1 (UNIT TESTING)	The user gives the input in the form of a training with dataset	An output is predicted as the features are extracted	The result is that the dataset is trained. Therefore, testcase 1 is passed.
Test Case2 (INTEGRATION TESTING)	The user gives the input in the form of opening interface.	An output is predicted as the user opens the interface is successful.	The result is that the user opened the interface. Therefore, testcase 2 is passed.
Test Case3 (INTEGRATION TESTING)	The user gives the path of the image for which caption is to be generated.	An output is predicted with accurate caption of an image.	The result is that the user sees an accurate caption for the given image. Therefore, testcase 3 is passed.
Test Case4 (USERINTERFACE TESTING)	The user gives the input as path of image and tests whether it is producing caption or not	An output is predicted with accurate caption	The result is that the user sees caption for the image. Therefore, test case 4 is passed.

Table 7.1: Test Results

CHAPTER 8

OUTPUT SCREENS

8. OUTPUT SCREENS

Figure for successful training of dataset is shown below.

Fig 8.1: Training of dataset

User uploads the path of the image and the captions are predicted as below.

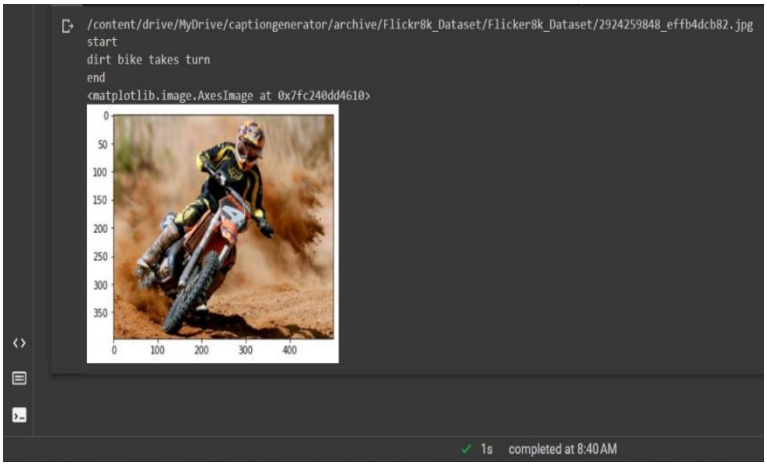


Fig 8.2: Caption for an image

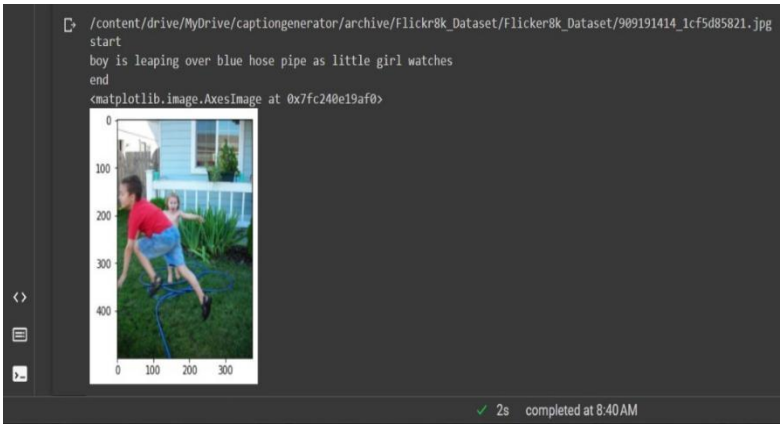


Fig 8.3: Caption for an image

Fig 8.4: Caption for an image

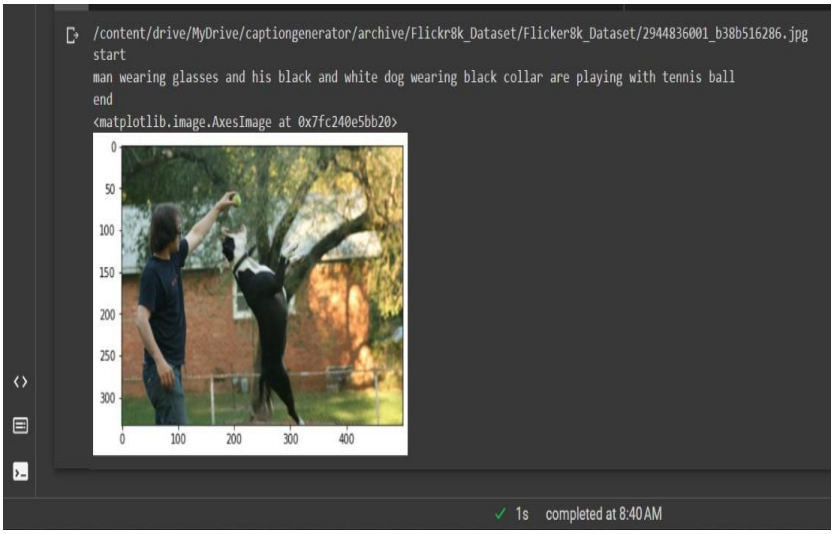


Fig 8.5: Caption for an image

CHAPTER 9

TIMELINE OF THE PROJECT

9. TIMELINE OF THE PROJECT

9.1 TIMELINE OF THE PROJECT

The proposed project has been carried out in various phases as follows:

1. **Understanding the problem:** The problem understanding of the project has taken around 30 days. Various research papers were studied to understand the problem of malicious URL detection.

2. **Literature Survey:** The survey for the literature for the project has taken around 60 days. All the problem related research papers have been summarized by detailed study.

3. **System Architecture and modules of the project:** After a detail understand of the problem and its requirements an appropriate Architecture for the proposed system has been done and module for this Proposed work has been prepared which has taken around 30 days.

4. **Design:** To prepare the design of this Proposed work is expected to take at least around 30 days.

5. **Project 1st External exam:** The External exam and viva will be held in November.

6. **Implementation:** Based on the design of this Project it is estimated to take around 30 days.

7. **Testing:** After completion of the Project as it need to be testing in many ways it is estimated to take around 30 days.

8. **Documentation:** On Basis of all the work that has been done in this proposed project it is estimated around 30 days for preparing document by representing all the necessary information.

S.No	ACTIVITY	TIME PERIOD
1	Understanding the Problem	July 2022
2	Literature survey	Aug-Sep, 2022
3	System Design	Oct-Nov 2022
4	Project 1st External exam (100 marks)	Nov 2022
5	Implementation	Jan 2023
6	Testing	Feb 2023
7	Documentation	March 2023

Table 9.1: Timeline of the project

CHAPTER 10

CONCLUSION

10. CONCLUSION

10.1 CONCLUSION

In this project, an overview of deep learning-based image captioning methods are given.This project has given a taxonomy of image captioning techniques, shown generic block diagram of the major groups and highlighted their pros and cons. different evaluation metrics and datasets with their strengths and weaknesses are discussed. A brief summary of experimental results is also given. This Project briefly outlines potential research directions in this area. Although deep learning-based image captioning methods have achieved a remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for some time.

This project have used Flickr_8k dataset which includes nearly 8000 images, and the corresponding captions are also stored in the text file. Although deep learning -based image captioning methods have achieved a remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for sometimes. The scope of image-captioning is very vast in the future as the users are increasing day by day on social media and most of them would post photos. So this project will help them to a greater extent.

10.2 FUTURE ENHANCEMENT

Future work Image captioning has become an important problem in recent days due to the exponential growth of images in social media and the internet. This report discusses the various research in image retrieval used in the past and it also highlights the various techniques and methodology used in the research. As feature extraction and similarity calculation in images are challenging in this domain, there is a tremendous scope of possible research in the future. Current image retrieval systems use similarity calculation by making use of features such as color, tags, IMAGE RETRIEVAL USING IMAGE CAPTIONING histogram, etc. There cannot be completely accurate results as these methodologies do not depend on the context of theimage. Hence, a complete research in image retrieval making use of context of the images suchas image captioning will facilitate to solve this problem in the future. This project can be further enhanced in future to improve the identification of classes which has a lower precision by training it with more image captioning datasets. This methodology can also be combined with previous image retrieval methods such as histogram, shapes, etc. and can be checked if the image retrieval results get better.

REFERENCES

[1] Shanshan zhao,lixiang li,haipeng peng,zihang yang, Jiaxuan zhang,image caption generation via unified retrieval and generation based method 2020

[2] Shuang Bai and Shan An. A Survey on Automatic Image Caption Generation Neurocomputing. ACM Computing Surveys 2018

[3] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazee Scheduled sampling for sequence prediction with recurrent neural networks. 2015

[4] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, D. Forsyth, Every picture tells a story: Generating sentences from images, in: European Conference on Computer Vision 2010

[5] Y. Yang, C. L. Teo, H. Daume, Y. Aloimono, Corpus-guided sentence generation of natural images, in: Proceedings of the Conference Empirical Methods in Natural Language Processing 2011

[6] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, A. Y. Ng, Grounded compositional semantics for finding and describing images with sentences 2014

[7] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson,Stephen Gould,and Lei Zhang.Bottom-up and top-down attention for image captioning 2017

[8] Vishwash Batra, Yulan He, George Vogiatzis (2018). Neural Caption Generation for News Images in School of Engineering and Applied Science, Aston University. 2018

[9] Justin Johnson, Andrej Karpathy,Li Fei-Fei.DenseCap: Fully Convolutional Localization Networks for Dense Captioning Department of Computer Science, Stanford University 2015

[10] Dataset Flickr8k https://www.kaggle.com/datasets/adityajn105/flickr8k

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 16th Mar 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 03

10.48047/IJIEMR/V12/ISSUE 03/16

Title **AN AUTOMATIC IMAGE CAPTION GENERATION APPRAOCH USING LSTM AND CNN**

Volume 12, ISSUE 03, Pages: 122-128

Paper Authors
K. SAI CHARAN LAHIRI, M. ANITHA LAKSHMI, P. PREM KUMAR,SK. ALTAF, M. YASWANTH KUMAR, SLVVD SARMA

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

AN AUTOMATIC IMAGE CAPTION GENERATION APPRAOCH USING LSTM AND CNN

[1]K. SAI CHARAN LAHIRI, [2]M. ANITHA LAKSHMI, [3]P. PREM KUMAR,[4]SK. ALTAF, [5]M. YASWANTH KUMAR, [6]SLVVD SARMA

[1,2,3,4,5] B. Tech Students, Dept of CSE, Kallam Haranadhareddy Institute of Technology, Chowdavaram, Guntur, A.P, India.

[6]Assistant Professor, Dept of CSE, Kallam Haranadhareddy Institute of Technology Chowdavaram, Guntur, A.P, India.

ABSTRACT: Automatic image caption generation is one of the frequent goals of computer vision. Image description generation models must solve a larger number of complex problems to have this task successfully solved. The objects in the image must be detected and recognized, after which a logical and syntactically correct textual description is generated. For that reason, description generation is a complex problem. It is an extremely important challenge for machine learning algorithms because it represents an impersonation of a complicated human ability to encapsulate huge amounts of highlighted visual pieces of information in descriptive language. As the deep learning techniques are growing, huge datasets and computer power are helpful to build models that can generate captions for an image. Hence in this work, an automatic image caption generation approach using LSTM and CNN is presented. In this project, deep learning techniques like CNN (Convolutional Neural Network) and LSTM (Long Short Term Memory) are used to identify the caption of the image. Image caption generator is a process which involves natural language processing and computer vision concepts to recognize the context of an image and present it in English. In this project, some of the core concepts of image captioning and its common approaches are followed. Keras library, numpy and jupyter notebooks are used for making of this project. This project also discusses about flickr_dataset and CNN used for image classification.

KEYWORDS: Automatic Image Caption Generation, Deep Learning (DL), Convolutional Neural Network) and Long Short Term Memory (LSTM).

I. INTRODUCTION

As we are living in the 21st century, image caption is one of the most needed tools these days. With the rise of users in internet the number of images and videos This data is usually unstructured and raw data which does not carry much information. Whether to extract and use the right data from the internet or to organize the files captioning plays a huge role. This application has a built-in function for producing captions for a specific image. Image captioning means automatic generation of a caption for an image [1].

As a recently emerged research area, it is attracting more and more attention. Nowadays, an image caption generator has become the need of the hour, be it for social media enthusiasts or visually impaired people. It can be used as a plug in in currently trending social media platforms to recommend suitable captions for people to attach to their post or can be used by visually impaired people to understand the image content on the web. Image captioning has various applications such as recommendations in editing applications, usage in virtual assistants, for image indexing, for visually impaired persons, for social media, and several other natural language processing applications [4].

To achieve the goal of image captioning, semantic information of images needs to be captured and expressed in natural languages. Connecting both research communities of computer vision and natural language processing, image captioning is a quite challenging task. Various approaches have been described to solve this problem. The number of digital

images increases rapidly; hence, categorizing these images and retrieving the relevant web images are a difficult process. For people to use numerous images effectively on the web, technologies must be able to explain image contents and must be capable of searching for data that users need. Moreover, images must be described with natural sentences based not only on the names of objects contained in an image, but also on their mutual relations.

Photo captions aim to describe objects, actions, and details found in an image using natural language. Most image caption research focuses on single-sentence captions, but the descriptive capabilities of this form are limited; one sentence can only describe in detail a small part of an image [2]. This task of automatically generating captions and describing the image is significantly harder than image classification and object recognition. The description of an image must involve not only the objects in the image, but also relation between the objects with their attributes and activities shown in images. Most of the work done in visual recognition previously has concentrated to label images with already fixed classes or categories leading to the large progress in this field. Eventually, vocabularies of visual concepts which are closed, makes a suitable and simple model for assumption.

Automatic caption generation for an image is one of the challenging problems in artificial intelligence. Image captioning models not only solve computer vision challenges of object recognition but also capture and express their relationships in natural language. This task is more complicated as compared to well-studied image classification and object recognition tasks, which have been the main focus in the computer vision community [5].

Recent advancements in language modeling and object recognition have made image captioning an essential research area in computer vision and natural language processing. Caption generation of an image has a great impact by helping visually impaired people to better understand the contents on the web [3]. Automatic caption generation is a tough undertaking that can aid visually challenged persons in understanding the content of web images. It may also have a significant impact on search engines and robots. This problem is substantially more difficult than image categorization or object recognition, both of which have been extensively researched.

Recently, deep learning methods have achieved state-of the-art results on examples of this problem. It has been demonstrated that deep learning models are able to achieve optimum results in the field of caption generation problems. Hence in this work, an automatic image caption generation approach using CNN and LSTM is presented in this work. The rest of the work s organized as follows: The section II demonstrates literature survey. The section III presents an automatic image caption generation approach using LSTM and CNN. The section IV evaluates the result analysis of presented approach. Finally the work is concluded in section V.

II. LITERATURE SURVEY

Xiangqing Shen, Bing Liu, Yong Zhou & Jiaqi Zhao et. al., [7] describes Remote sensing image caption generation via transformer and reinforcement learning. A new model using the Transformer to decode the image features to target sentences is presented. For making the Transformer more adaptive to the remote sensing image captioning task, we additionally employ dropout layers, residual connections, and adaptive feature fusion in the Transformer. Reinforcement

Learning is then applied to enhance the quality of the generated sentences. We demonstrate the validity of our proposed model on three remote sensing image captioning datasets. This model obtains all seven higher scores on the Sydney Dataset and Remote Sensing Image Caption Dataset (RSICD), four higher scores on UCM dataset, which indicates that the proposed methods perform better than the previous state of the art models in remote sensing image caption generation.

Songtao Ding, Shiru Qu, Yuling Xi, Arun Kumar Sangaiah, Shaohua Wan et. al., [8] describes Image caption generation with high-level image features. A novel image captioning model based on high-level image features is presented. They combine low-level information, such as image quality, with high-level features, such as motion classification and face recognition to detect attention regions of an image. We demonstrate that our attention model produces good performance in experiments on MSCOCO, Flickr 30K, PASCL and SBU datasets. This approach gives good performance on benchmark datasets.

Xinlei Chen, C. Lawrence Zitnick et. al., [9] describes learning a Recurrent Visual Representation for Image Caption Generation. A novel recurrent visual memory is presented that automatically learns to remember long-term visual concepts to aid in both sentence generation and visual feature reconstruction. Authors evaluated this approach on several tasks. These include sentence generation, sentence retrieval and image retrieval. State-of-the-art results are shown for the task of generating novel image descriptions. When compared to human generated captions, our automatically generated captions are preferred by humans over 19.8% of the time. Results are better than or comparable to state-of-the-art results on the image and sentence retrieval tasks for methods using similar visual features.

Philip Kinghorn, Li Zhang, Ling Shao et. al., [10] presents A region-based image caption generator with refined descriptions. A novel region-based deep learning architecture for image description generation is presented. It employs a regional object detector, recurrent neural network (RNN)-based attribute prediction, and an encoder–decoder language generator embedded with two RNNs to produce refined and detailed descriptions of a given image. Most importantly, the proposed system focuses on a local based approach to further improve upon existing holistic methods, which relates specifically to image regions of people and objects in an image. Evaluated with the IAPR TC-12 dataset, the proposed system shows impressive performance and outperforms state-of-the-art methods using various evaluation metrics

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio et. al., [11] describes Neural Image Caption Generation with Visual Attention. An attention based model is described that automatically learns to describe the content of images. Authors described how this model is trained in a deterministic manner using standard back-propagation techniques and stochastically by maximizing a variational lower bound. They also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. They validate the use of attention with state-of-theart performance on three benchmark datasets: Flickr9k, Flickr30k and MS COCO.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

III. AUTOMATIC IMAGE CAPTION GENERATION APPROACH

An automatic image caption generation approach using LSTM and CNN is presented in this section. The main objective of this project is to develop a web based interface for users to get the description of the image and to make a classification system in order to differentiate images as per their description. It can also make the task easier which is complicated as they have to maintain and explore enormous amounts of data. The fig. 1 shows the system architecture of automatic caption generation approach using CNN and LSTM.

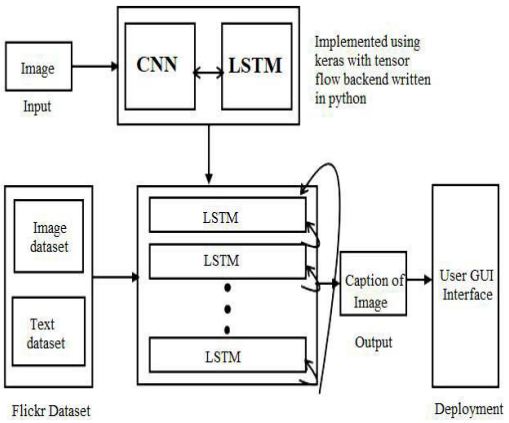


Fig. 1: System Architecture of Automatic Caption Generation Approach using CNN and LSTM

Here, a dataset called flickr8k which is collected from kaggle. Flickr8k dataset is a public benchmark dataset for image to sentence description. This dataset consists of 8000 images with five captions for each image. These images are extracted from diverse groups in Flickr website. Each caption provides a clear description of entities and events present in the image. The dataset depicts a variety of events and scenarios and doesn't include images 37 containing well known people and places which makes the dataset more generic. The dataset has 6000 images in training dataset, 1000 images in development dataset and 1000 images in test dataset. Features of the dataset making it suitable for this project are: Multiple captions mapped for a single image makes the model generic and avoids overfitting of the model. Diverse category of training images can make the image captioning model to work for multiple categories of images and hence can make the model more robust. Dataset is collected from various source of internet.

Data preprocessing is done in this step includes Data cleaning, Data reduction, Image data preparation. For instance, punctuations, digits, single length words are removed from the text dataset. Two deep learning models have been selected i.e, CNN and LSTM. Firstly, CNN takes image as input and extract features such as background, objects in the image.

CNN stands for Convolutional Neural Networks. It is a deep learning algorithm which takes image as an input.CNN scans images from left to right and top to bottom to pull out important features from the image and combines the features to classify images. Pre-processing required in convolutional neural networks is much lower as compared to other classification algorithms.

The architecture of a Conv Net is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area. The CNNs are inspired by visual system of human brain. The idea behind the CNNs thus is to make the computers capable of viewing the world as humans view it. This way CNNs

can be used in the fields of image recognition and analysis, image classification, and natural language processing.

CNN is a type of deep neural networks which contain the convolutional, max pooling, and activation layers. The convolutional layer, considered as a main layer of a CNN, performs the operation called "convolution" that gives CNN its name. Kernels in the convolutional layer are applied to the layer inputs. All the outputs of the convolutional layers are convolved as a feature map. In this study, the Rectified Linear Unit (ReLU) has been used in the activation function with a convolutional layer which is helpful to increase the non-linearity in input image, as the images are fundamentally nonlinear in nature.

The pooling layer is an important building block of CNN. Pooling can be the max, average, and sum in the CNN model. In this study, max pooling has been used because others may not identify the sharp features easily as compared to max pooling. The dropout layer has also been used, which drops the neurons during the training chosen at random to reduce the overfitting problem. CNN is used to extract features from the image. A pre-trained model called Xception is used for this.CNN can be used in the fields of image recognition, image classification, and natural language processing.

LSTM stands for long short-term memory, it is a type of RNN (Recurrent Neural Networks. LSTM is capable of working with sequence prediction problems. It is used for word prediction purposes. In LSTM based on previous text, one can predict what the next word will be. It is the same as google search where this system will show the next word based on our previous text. LSTM can carry out relevant information throughout the process with a forget gate and discards non-relevant information. LSTM will use the information that is extracted from CNN to generate a description of the image.

The CNN LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. This architecture was originally referred to as a Long-term Recurrent Convolutional Network or LRCN model, although we will use the more generic name "CNN LSTM" to refer to LSTMs that use a CNN as a front end in this lesson. This architecture is used for the task of generating textual descriptions of images. Key is the use of a CNN that is pre-trained on a challenging image classification task that is re-purposed as a feature extractor for the caption-generating problem.

A pre trained model called xception is used to train LSTM which will generate captions. Features which were extracted by CNN are given to LSTM. This LSTM will generate the captions for the given image. Once all these steps were implemented, a caption will be generated for the given image. The generated captions are displayed on screen using GUI (Graphical User Interface). This whole project is implemented using keras with tensorflow backend written in python.

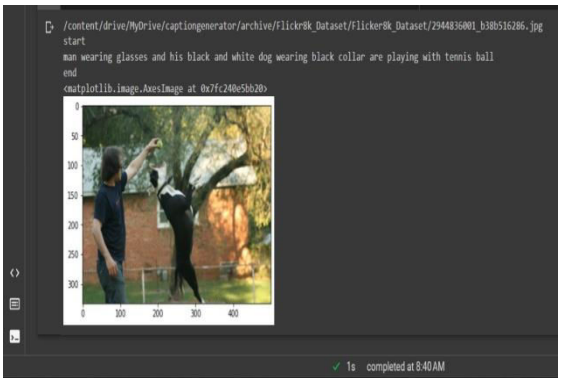
IV. RESULT ANALYSIS

In this section, an automatic image caption generation approach using LSTM and CNN is implemented using python. The result analysis of presented approach is evaluated here. In this analysis, flickr8k dataset is used. The CNN is used to extract the features and LSTM is used to generate the caption.

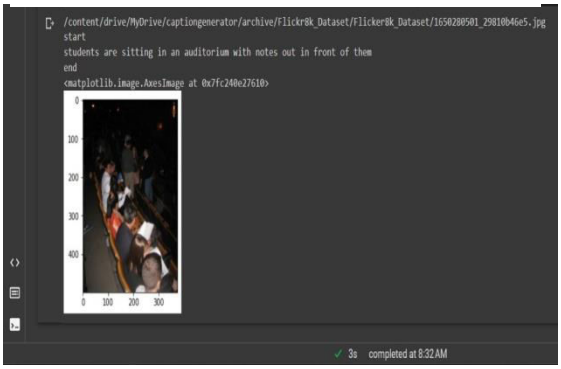
In this approach, firstly user upload an image as input this image is forwarded to CNN in which it extracts the features such as background, scene, objects in the image

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

using convolutional layer and pooling layer then these features are sent to LSTM by using fully connected layer. Now, the dataset which contains Image Dataset and text dataset is preprocessed to form an training model. This training model is used to train LSTM for which it generates captions. The user has to upload an image for which the caption has to be generated. The generated caption for the image is viewed by the user. The uploaded images and their captions are shown in following figures.

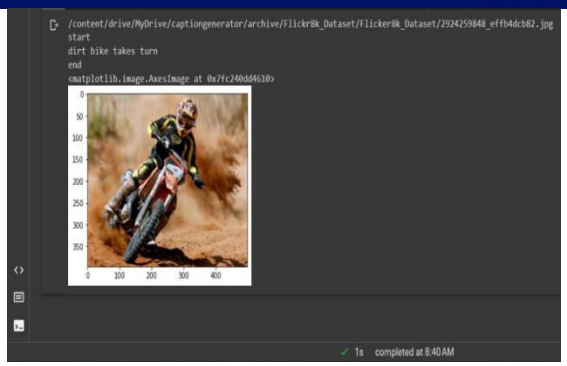


(a)



(b)

(c)



(d)

Fig. 2 (a), (b), (c) & (d): Uploaded imaged and their generated Captions

Hence this approach has generated the captions effectively and automatically for different images.

V. CONCLUSION

In this work, an automatic image caption generation approach LSTM and CNN is presented. This approach used Flickr_8k dataset which includes nearly 8000 images, and the corresponding captions are also stored in the text file. The deep learning models, Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) are employed in this analysis. The images are preprocessed and applied to CNN. CNN extracts the features like background, scene, objects in the image using convolutional layer and pooling layer. The extracted features are sent to LSTM by using fully connected layer. The LSTM generates the captions for the uploaded images. This approach has automatically generated the captions for different images. The generated captions are more accurate than state-of art approaches. The scope of image-captioning is very vast in the future as the users are increasing day by day on social media and most of them would post photos. So this project will help them to a greater extent.

VI. REFERENCES

[1] Peerzada Salman syeed, Dr.Mahmood Usman, "Image Caption Generator Using Deep Learning", Neuroquantology, October 2022, Volume 20, Issue 12, Page 2682-2691, Doi: 10.14704/Nq.2022.20.12.Nq77261

[2] Dr. P. Srinivasa Rao, Thipireddy Pavankumar, Raghu Mukkera, Gopu Hruthik Kiran, Velisala Hariprasad, "Image Caption Generation Using Deep Learning Technique", International Research Journal of Modernization in Engineering Technology and Science, Volume:04/Issue:06/June-2022, e-ISSN: 2582-5208

[3] Santosh Kumar Mishra, Rijul Dhir, Sriparna Saha and Pushpak Bhattacharyya, "A Hindi Image Caption Generation Framework Using Deep Learning", ACM Trans. Asian Low-Resour. Lang. Inf. Process., 2021,Vol. 20, No. 2, Article 32.

[4] Aishwarya Maroju, Sneha Sri Doma, Lahari Chandarlapati, "Image Caption Generating Deep Learning Model", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 10 Issue 09, September-2021

[5] Moksh Grover, Rajat Rathi Chinkit, Kanishk Garg, Ravinder Beniwal, "AI Optics: Object recognition and caption generation for Blinds using Deep Learning Methodologies", 2021 International Conference on Computing, Communication, and Intelligent Systems(ICCCIS), DOI: 10.1109/ICCCIS51004.2021.9397143

[6] Omkar Nitin Shinde, Rishikesh Gawde, Anurag Paradkar, "Social Media Image Caption Generation Using Deep Learning", International Journal of Engineering Development and Research, 2020, Volume 8, Issue 4, ISSN: 2321-9939

[7] Xiangqing Shen, Bing Liu, Yong Zhou & Jiaqi Zhao, "Remote sensing image caption generation via transformer and reinforcement learning", Multimedia Tools and Applications, volume 79, pages26661–26682 (2020), doi: 10.1007/s11042-020-09294-7

[8]Songtao Ding, Shiru Qu, Yuling Xi, Ar un Kumar Sangaiah, Shaohua Wan, "Image caption generation with high-level image features", Pattern Recognition Letters, Volume 123, 15 May 2019, Pages 89-95, Elsevier, doi: 10.1016/j.patrec.2019.03.021

[9] Xinlei Chen, C. Lawrence Zitnick, "Learning a Recurrent Visual Representation for Image Caption Generation", Computer Vision and Pattern Recognition (cs.CV), arXiv:1411.5654v1 , doi:10.48550/arxiv1411.5654

[10]Philip Kinghorn, Li Zhang, Ling Shao, "A region-based image caption generator with refined descriptions", Neurocomputing, Volume 272, 10 January 2018, Pages 416-424, Elsevier, Doi:10.1016/j.neucom.2017.07.014

[11] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio, "Neural Image Caption Generation with Visual Attention", Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.