DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,

SCHOOL OF ENGINEERING AND TECHNOLOGY,

SHARDA UNIVERSITY, GREATER NOIDA

IMAGE CAPTION GENERATOR

A project submitted

In partial fulfillment of the requirements for the degree of

Bachelor of Technology in Computer Science and Engineering

By

SUPRIYA KUMARI(2018009684)

JULU BASNET(2018015987)

MOHIT RATHORE(2018015523)

DIPANSHU (2018014442)

Supervised by:

Dr.Vivek Kumar Singh,Assoc. Prof (CSE)

March, 2022

CERTIFICATE

This is to certify that the report entitled **"Image Caption Generator"** submitted by Supriya kumari(2018009684),julu Basnet(2018015987),Mohit(2018015523), Dipanshu(2018014442) to Sharda University, towards the fulfillment of requirements of the degree of "**Bachelor of Technology**" is record of bonafide final year Project work carried out by him in the Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University. The results contained in this Project have not been submitted in part or full to any other University for award of any other Degree.

Signature of Supervisor

Name:

Designation:

Signature of Head of Department

Name:

(Office seal)

Place:

Date:

Signature of External Examiner

Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider ourselves very lucky and honored to have so many wonderful people lead us through in completion of this project.

First and foremost we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

My grateful thanks to **Dr. Vivek Kumar Singh** for his guidance in my project work.

Dr. Vivek Kumar Singh, who in spite of being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without his help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

Supriya Kumari(2018009684)
Julu Basnet(2018015987)
Mohit Rathore(2018015523)
Dipanshu(2018014442)

ABSTRACT

In this project, we desire to achieve accuracy upto 90%. Many researchers have done research activities in this field but the problem is sometimes, we may achieve wrong predictions so we can try to achieve the best results of the predictions. And even today, there is quite an absolute lack in this field. In-built applications that generate and provide a caption for those images. We have finished it with the help of deep neural network models. It is to generate a description for a particular image. It also generates syntactic and symmetric correct sentences. This technique allows in producing a suitable caption of an image. This analysis will use a convolution neural network (CNN) and long short term memory (LSTM) for image caption generators. The dataset used in this project is the Flickr 8k dataset. We are using the VGG-16 model which has 16 layers and the pre-trained model i.e vgg16 model for image classification through transfer learning. The goal is to improve the better caption for the provided image over the previous research. We can apply this technique in social media like instagram, facebook, suppose someone upload the picture then after getting uploaded it generates the particular caption based on that picture. We can achieve higher accuracy while minimizing and optimizing time and space complexity. There are different types of strategies for extracting the features from the images for generating sequence sentences. We can use caption generators for the images to enhance the image in proper descriptive form.

INDEX TERMS- Image captioning, Long Short Term Memory, VGG-16,Recurrent Neural Network,Convolution Neural Network.

Contents

List of Figures

List of Tables

Chapter 1: INTRODUCTION

Vision weakness or vision impairment, It means the restriction of activities and elements of the visual framework. As indicated by the WHO 285 million individuals are outwardly hindered around the world, including 39 million blind people. The life of visually impaired people is full of despair to a large extent and they are facing many difficulties in their daily life, without the sharpness of sight life is a very challenging task.

Technology is a new ray of hope in the lives of hopeless visually impaired people, by using it we can upgrade the lives of visually challenged humans. Al is at the fore in this era of technology for visually challenged people. One of the sciences which can solve this problem is known as Image Caption Generation (ICG).

AI can create a machine that can exactly convey an image like a normal human has significant applications in the field of robotic vision, business, and many more. There have been various efforts taken to get a solution to this problem including template-based solutions which use image classification i.e. Labeling objects from a proper arrangement of classes that can be embedded into a dummy layout sentence.A few introductory efforts to create more point-by-point picture descriptions have been made, for example by Farhadi and Kulkarni but these models generally depend on the hard-coded sentences & vision theories. Also, the objective of the majority of these works is to precisely depict the substance of a picture in a single sentence.

Much of the previous experiments have centered around gaining information and getting the sensation and object represented in the picture. But recently the center of attention is RNN.With the advancement in the field of technology,nowadays it is becoming one of the concerned subjects for the researchers to do research in the technology field where AI is one of the important research fields.

Many tasks related to AI based researches are still going on since the previous ages.Among the AI based research fields,image caption generator is also the one in which the most relevant captions are created on the basis of the uploaded input images.Basically,the motive behind using this application is to provide the assistance to the visually impaired person by creating some items which helps in the description of the scene around them by converting scene into text while crossing the road. It can be applicable in many possible scenarios like CCTV cameras,Google driving cars,Web development etc.

Image captioning is one of the important task of AI research in which captions are generated as per the provided input images with the help of CNN-LSTM architecture.Here,what actually happens is that as soon as the objects get detected,it can be represented in the context which reflects overall structure of the image.

If we say about humans,they can easily recognize,what's happening here and there within a very short period of the time but for machine,it's not possible within a short range of time because it performs the task stepwise and requires some algorithms to compute different values so it's becoming a challenging task as it requires a little bit more time to perform the activities.

For this task,datasets should be must.There are different datasets which can be used for the whole model but some of the datasets are heavy though they helps in creating a effective model,they takes a lot of time to train the model so here we are using only flickr 8k dataset which is easy to download and not much time is required to train the model.

Both the combination of CNN and LSTM forms a single architecture known as CNN-LSTM architecture.Flikr8k dataset is used for the testing and training phase.There are many versions of recurrent neural network but here we are using one of the most latest version of LSTM i.e.LSTM which results in the eradication of the vanishing gradient problem.CNN acts as encoder and LSTM acts as decoder so both the encoder and decoder performs every phases happening in the whole process of the application.

We are using the BLEU metric for evaluating the accuracy gained after testing of the dataset.The whole process of this model is based on the deep learning approach.Nowadays ,it is considered as one of the attracted approach which has a lot of attention towards the real-scenarios application.There has been a curiosity of the researchers since many times ago for the further improvement of this model using different datasets,techniques which can be applicable widely in the world .For this,the researchers are still working on it and will be continuously goes on until they get the best result.

Computer vision examines an image thinking about it as a 2D array. Hence, Image captioning is a language translation problem described by Venugopalan. Earlier, language translation was complicated and it included some different goals; however, the new work has shown that the undertaking would be able to be achieved in an efficient way using RNN. But, regular RNN suffers from removing these issues which were complex in the case of past applications. Using LSTM and GRU which contain internal processes and logic gates that keep information for a more extended time frame and pass only useful data, we can remove these issues. Abstracting pictures using natural languages is an elemental and challenging task.

This has a huge potential result on upcoming aspects. For instance, it could assist visually impaired individuals with a better understanding of the substance of pictures on the web.

Also,it could give more correct and conservative data of images in situations like picture sharing in social networks or video surveillance systems. This project completes the task using deep neural networks. Image captioning has various applications in various fields such as Bio-medicine, E-commerce, Web securities, military investigation, etc. Social media, for example, Facebook, Instagram, etc can generate titles from images automatically.

1.1. Motivation:

Image caption Generator helps especially for the blind person. As its use has been enhanced in different sectors like in the society also, we are having a kind of attention towards it. It can be created for the purpose of automatic driving i.e self driving cars in the research that has been in that field till now.

Above these scenarios have motivated us towards the direction in the proper way.

A product can be created for those who have vision problems which helps them to guide the time of crossing the road without the need of assistance from anybody. There won't be any need for the third person to hold their hands while crossing the road or during traveling. In this case, how it assists them, first whatever the scene present around them is converted into the text then, the text gets converted to voice. **"Nvidia"** launched an app for such a type of product in order to add the blind person .

It is found that it has been more attractive after the research gets done.

As we are motivated towards this, we are contributing this task into four team members.

1.2 Problem Definition:

The goal of this project is to detect and scan the image with the help of CNN and LSTM and generate the caption for the particular image. There are some difficult situations where we can apply neural networks for creating or thinking as humans.

The purpose of this project is to create applications that will help visually impaired people by creating an object which can provide the strong feeling of the sense of the surrounding, so that they can easily cross the road without the help of the second person.

These image captions can be read out loud to the visually impaired human being so that they have got an experience of what's happening around them as shown in fig 1.

Fig 1.visually impaired person

- "Nvidia" launched a product for the blind people. In this product, they are trying to explain the surroundings to a blind person or low vision as they could be dependent on sound and texts for description of a scene.
- Google Image search- we can try to make more effective use of it, automatic image caption can be done for any images. hence, search results would also be based on the object present in that image.
- Self driving cars are one of the biggest challenges in this field and being able to properly caption scenes around the car.
- Nowadays, Video surveillance cameras are available everywhere and 1-billion surveillance cameras will be watching around the world . If we'll try to generate relevant captions, we can set off alarms as quickly as malicious activity occurs somewhere. It is likely to help reduce crime and /or accidents.

1.3 Project Overview/ Requirement Specifications

1.3.1 Project Overview:

Describe the learning model used, the specification and implementation of the algorithms and, as a result, the web development application, we need to explore the latest approaches to the image captioning generator. To collect image information the usage of static object class libraries in the image and model use of statistical language models. There are two approaches in the field of applications. Firstly, to detect objects in the image using CNN and other is to caption the image using RNN based totally on LSTM. The interface of the model is developed using Flask Rest-API, a web framework of python. Primary use case of this project is to assist the visually impaired, to understand the surrounding environment and act to automatically do that. Our project pipeline, pipeline for deploying consists of the client and server .The client sends requests to the server in which the request consists of the image.So,after receiving the images,first step would be to store the images.Firstly,the images gets encoded into the resnet model.The saved image will go first into the resnet model and resnet model will give(1,2048) vector of encoding and this encoding goes into the train model which is image captioning model .As a result, model will generate a caption. Now,this caption can be returned from the server as a response.

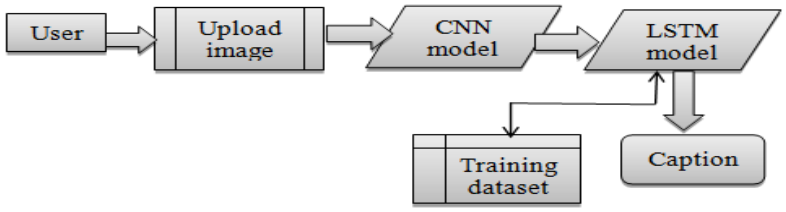


Fig 2.Visual representation of this approach

1.4 Hardware Specifications:

Table1. Hardware Specifications

Minimum Requirements	**Windows**
OperatingSystem	Windows10
Processor	Dual Core,Intel i3(8th gen)
RAM	2GB RAM
DISKSpace	The amount of disc Space Available Depends on the partition size and whether or not online help files are allowed. TheMathWorksinstallerwouldtellyouhow muchdiscvolumeyourpartitionneeds.
Graphics Adapter	8-bit graphics adapter and display(for256 simultaneous colors)
CD_ROMdrive	For Installation From CD

		Windows		
	Processor	**RAM**	**DISKSpace**	**GraphicsAdapter**
JUPYTER NOTEBOOK- Python (3.7)	Intel i3	1GB	1 GB for jupyter notebook, 5 GB for a typical installation	A 32-bit or 64-bit OpenGL capable graphics adapter is strongly-Recommended
Sublime text-3 (Python, html, css, javascript)	Intel i3	3.5GB	3.5GB	A 32-bit or 64- bit OpenGL capable graphics adapter

1.5 Software Specifications

Tensorflow
Keras
Flask
Nltk

Language

- **Python**

Chapter2: Literature Survey

2.1 Existing System:

2.1.1.“Detection and recognition of objects in Image Caption Generator”:

- To detect the object in the image and generate captions for a particular image and further enhance the existing image caption generator system.
- Input images are used for feature extraction and then output from the previous layer is fed as input to the current layer.
- Feed-forward neural network in RNN creates a few issues that it only works on the current input layer, it can’t handle sequential data and can’t memorize descriptive words as it consists of short memory.
- ICG model can be relevant in the upcoming date for more captions.

2.1.2. “Image Caption Generator via Visual Attentions and topic Model”:

- In this article, AIC-ICC is an image caption dataset published by AI Challenger 2017, and currently the biggest AIC-ICC dataset.
- AIC-ICC datasets contain 2,10,000 photos and 1.5 million descriptions.
- Each image has five different captions and 30,000 images are used for the training process of the model and the remaining datasets are used for testing and validation.
- To make the conversions between models from the visual context of the image to the text of the natural language.
- The information on the subject is extracted from the descriptive sentences of the image during the training phase and predicts the information on the subject of the image in the testing.

2.1.3. “Image Caption generation using CNN &LSTM”:

- CNN and LSTM model is used for the whole process where CNN extracts every information based on the particular image provided and LSTM uses every kind of information from the CNN then with the help of this information,it generates the visual description of the image.

- For the image caption generator, small datasets i.e. Flickr-8k datasets will be utilized. MSCOCO is the large dataset which helps in creating the models consisting of more accuracy but it takes a lot of time to train the network.

- The Inception V3 model consisting of parameters is used. Beam search algorithm which chooses various options for the input sequence. The BLEU metric is used for the evaluation.

2.1.4. "Image Caption Generator based on Deep Neural Networks":

- CNN, RNN and sentence generation methods are used for captioning the images. To get the content of the given image, CNN is used.

- LSTM method having few parameters helps to save the memory. GRU model is also applied which helps to attain equivalent results with few parameters and less time for training. In terms of model training speed, Gated Recurrent unit is 29.29% faster than LSTM for processing the same dataset.

- Using beam search, multiple sentences are generated. With the increase in beam size , the caption for each and every image becomes short.

- Here , they used CNN-15TM model for the clarification of the image .There is the problem of vanishing gradient so to overcome this problem .They used LSTM. LSTM is used to predict the texts which are mostly used in many fields like in mobile applications and in social media. Incase of social medias like in instagram suppose if anyone upload images then for getting uploaded it detects the objects first in the pictures and gives the suitable caption for that uploading image .The motive of using CNN-algorithm is to clarify each and every pixels of the image by means of similarities and differences

2.1.5. "Image Caption Generator Using CNN & LSTM":

- Based on deep learning neural networks. They also used machine learning techniques. Computer vision and machine translation based on deep learning models are used for describing the provided images as input and generating the captions based on those particular images.

Table2. Literature Survey:

S.No	Title	Author	Objective	Software/Hardware requirements	Advantages/ Disadvantage
1.	Image Caption generator using **CNN &LSTM**	UjwalaBhoga, G.sreeja, Md Arif	To create the caption eating platform using tensorflow library for producing **LSTM** model.	Flickr-8/MSCOCO datasetInceptionV3 **CNN LSTM** model	Due to the use of a large dataset it gives a more accurate model.The problem faced is it takes more time to train.
2.	Image-Captio n-Generator based on **Deep Neural Network**	Jianhui,Che n.WenqianD ong.Minche n Li.	To generate the suitable captions by decomposing into CNN & RNN	GoogleNet,Vgg CNN & LSTM ,RNN,GRU,AlexN et	Model generates syntactically correct and clear sentences based on the image
3.	Image Caption Generator using **CNN &LSTM**	Swarnim Tripathi,Rav i Sharma	To create the captions of image and shows the relationship b/w using CNN &LSTM	Flickr-8k dataset CNN,RNN, LSTM	This model has proposed that it gives the name of the objects and shows plain image classification
4.	Image Caption Generator	MeghaPanick er,Vikas Upadhya, Uriada, Mathur Gunjan seth	To obtain a bit of Competence in the deep learning plan.	Flickr 8k dataset CNN & RNN	This mode can be utilize in different kinds of filed(application fields)
5.	Image caption Deep learning approach	Lakshminar-a simhan, Srinivasan,Di nesh,Sreekant han.Amutha A.L	To generate image caption implementing deep learning approach	Flickr-8k & flickr-30k datasets	Network takes almost a week to train on GT 1050 4Gb and GTX GPUS

2.2 Proposed Work:

The main idea of the Proposed Solution is to eradicate almost all the existing issues & provide a better platform to explore with.There are several domains where the changes are implemented.These variations are accomplished by entering libraries such as Numpy & Pandas, etc. for applying in-built functionalities on datasets using these packages.Then the next step was to prepare GPU memory for the efficient use of training the objects.After these 2 important steps importing database images and updating the image database captions was a crucial task; on completing this we were ready with our initial set-up to work with.Now, we have Edited a few pictures and their captions from Database.Next, we have Cleansed the captions for further analysis.Another Mandatory Step was to Sort the top 50 names from refined data set to make searching process easier.After this we Loaded VGG16 model and weights to release features.After releasing features we Arranged the same images in the database.Then, we made caption tokens for further processing.We see that processing of captions and pictures is stated by need for a stand by model.The very next step was to Build LSTM model and provide it training for remembering the common patterns and recognising them.We also Adjusted the missing value by treating them with a default value.Generating the final model and finally measuring Performance Appraisal using BLEU scores.

This is how we achieved the desired goal without much hindrances.

We propose a CNN-LSTM model to generate the captions..The underlying principle is that photos are presented. as input first They are then recognised and a corresponding caption is provided depending on the image.Caption generation begins with tokenization, a technique that breaks down an object's glyphs into smaller components like words, symbols, and other elements.Tokens are created from the strings and then saved to a file. First and foremost, the data set is utilized for training, testing, and validation in data processing.Because certain data may be dummy data this technique is used to remove the dummy data and acquire only the original data (pure data).In the case of object detection, whatever the picture's objects will be identified.The LSTM model is utilized in this process. Things in the photo are identified when images are uploaded. LSTM is fed the extracted feature and then uses it to generate the words based on that feature. There is a third phase of this assignment, which is the generation of many phrases, in which the objects are first recognised for the generation of words. There is an output in the form of a phrase as a result of the addition of each word to the previously formed words.We provide a model that uses CNN-LSTM neural networks to automatically identify and describe the images in a given scene.

For object detection, a variety of pre-trained models are used, and the CNN-LSTM model is used to generate captions for the images. And Natural Language Processing toolkits(NLTK) are also used for data preprocessing vocabulary training.

Pre-trained models(VGG16 model) are used for object detection in Transfer Learning. The CNN is used to identify specific objects and concepts in images, while the RNN is used to feed the sequential data to itself. The encoder and decoder are typically referred to as CNN and RNN, respectively.

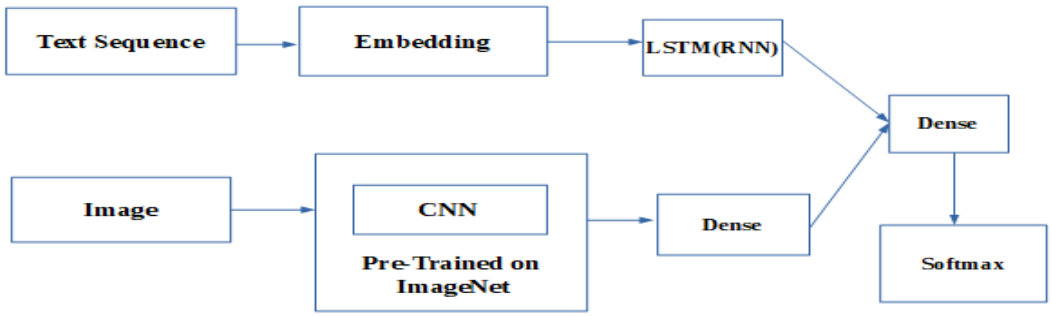


Fig3. RNN only handles text

CNN-LSTM model:

We'll use the Keras Model from API to define the model's structure. It consists of three major steps:

1)Image Feature Extraction:

The capabilities of the images from the Flickr 8k dataset are extracted from the usage of the Xception model ,VGG16 model because of the performance of the model in object detection. It is more accurate than the VGG16 model. We'll see about that later. The Xception is a convolutional neural network which consists of 36 layers,as this model configuration learns very quickly. A Dense layer is used to create a 2048 vector element illustration of the photo, which is then fed directly to the LSTM layer.

2)Sequence processor:

A sequence processor is a layer that acts as a word embedding layer for managing text input.The embedded layer contains rules for extracting the text's needed features as well as a mask to ignore padding values. For the last phase of picture captioning, the network is coupled to an LSTM.

3)Decoder:

In the last phase of the model, an additional operation is used to combine the input from the Image extractor and sequence processor stages, which is then delivered to a 256 neuron layer and eventually to a final output.

Dense layer that uses the textual content data processed inside the sequence processor phase to build a softmax prediction of the following phrase in the caption for the duration of the complete vocabulary. Understanding the flow of pictures and text requires an understanding of the network's structure.

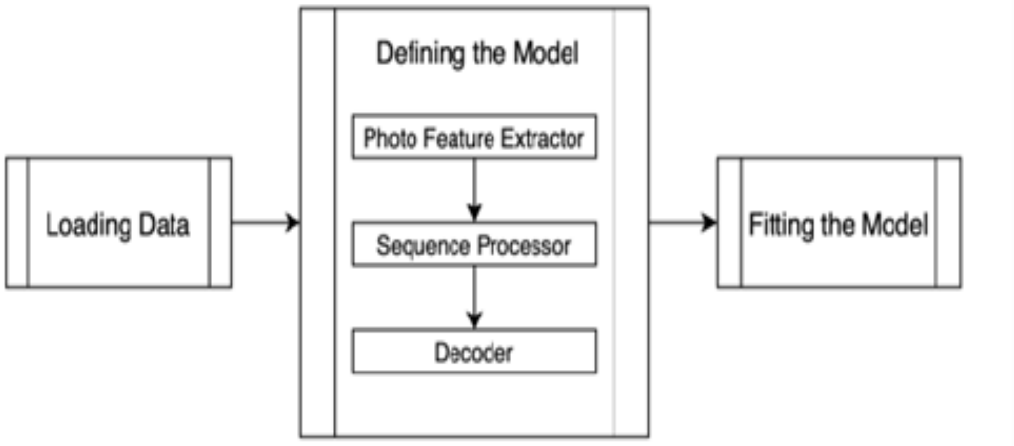


Fig 4 : Development of deep learning model

The data flow diagram displays the data flow throughout our proposed model, which begins with collecting and preprocessing images data collected from various sources to be in the desired input format. Following that, the text descriptions (captions) for each of the images are prepared, and the text and image data are combined to create a deep learning model using a combination of photo feature extractor, sequence processor, and decoder, which is then trained using progressive loading to ensure accuracy. The created model is evaluated or tested using test data that has been separated from the complete training set. Once the model has acquired the necessary level of accuracy, it may be used to generate descriptions for the images.

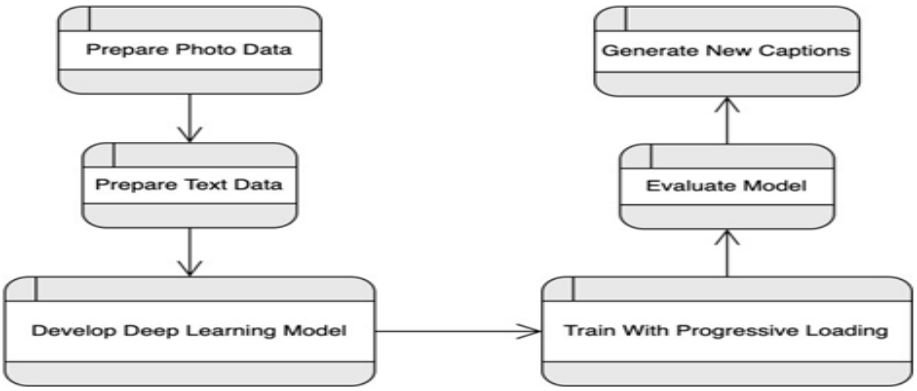


Fig 5. Data Flow Diagram

2.3 Feasibility Study

- It is a study which reveals whether a given project is feasible or not.
- It is conducted inorder to find the answers to the following questions:
  - Do we have required resources and technologies to build the project.j
  - Do we gain benefits from the projects?
  - Does it worth the investment done in the project?

It is of following types:

a)Technical and resource Feasibility

- In this feasibility,we check whether we have required technical resources like hardware and software requirements for the development of the overall project.
- This also analyzes technical skill and abilities of the technical team ,whether existing technology can be used or not,maintenance is not easy for the chosen technology or not.
- This project is python based which is the main technology requirement and other resources are the programming devices i.e laptop or computer and programming tool is jupyter notebook which is freely available and can be downloaded easily on the system.

b).Economical Feasibility

- This helps in analysis of the study,cost and benefits of the project.
- A detailed analysis is carried out to know what will be the cost of the project including hardware and software resources required,design and development cost of the project.
- It is also analyzed whether the project will be beneficial for the organization or not.
- The tools,hardware and software requirements for this project are free to use and open source software.As this project requires windows 10 OS,jupyter notebook,installment of various libraries which can be done easily with free of cost.There isn't any charge process for all this.

c).Time Feasibility:

The time duration for this project is provided as per the total time required for the complete process of this project.This project took around period of 9-10 months that had several deadlines as per the phases of the project given.The development of this overall project is well-planned and there won't be no delay so that it could result in timely completion of the project.

2.4 Risk Identification

2.4.1 Product Size
Sometimes ,we think that our project could be small but as a result it exceeds more which could also lead to danger risk.

2.4.2. Customer character
First,the customer asks for something and at the end time s/he asks for other things which could lead to the high risk.

2.4.3 Process definition
First developer or senior of the project tells one thing for creating the project like these things should be done or presented for the project then after that s/he tells technical teams to change something which could lead to risk.

2.4.4 Technology to the limit
Technology,which we are using for the ongoing project is upto the point or not.

2.4.5 Development Environment
Suppose,we are using android studio and there happens to be some new update then UI gets changed. For
adapting to the change of new UIit will take a lot of time which could lead to risk.

Strategies for risk management
S1 Diversify the products or services.

S2 The team works should be made aware about the technical prospects required for the project.

S3. Keep the clear transparent records of the division of the workload of the project to each technical expert.

S4 .Use an appropriate noise reduction algorithm prior to segmentation processing.

S5 we should follow up the things as per the demands of the customers.

S6. Whatever the technical tools we are using for the ongoing project,we should focus on the proper use of that one in the project.Proper algorithm should be followed

Chapter 3: System Analysis and Design

3.1.Block diagram project

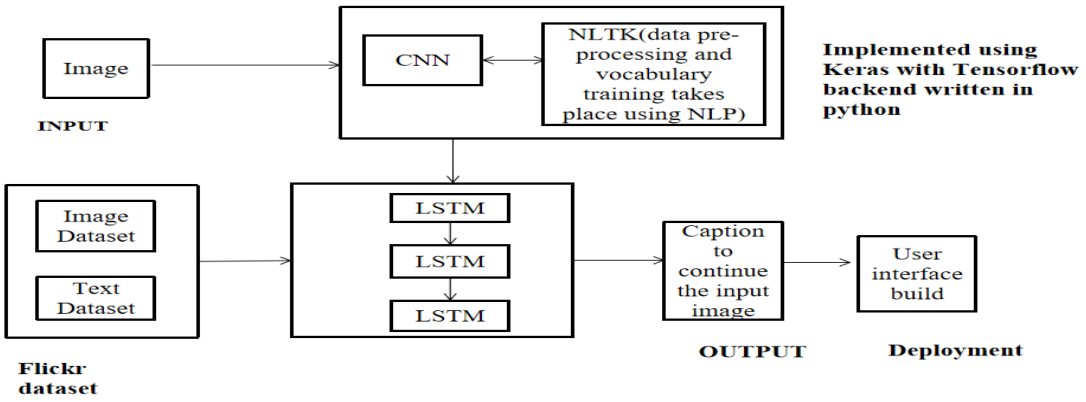


Fig 6. Block diagram of our model

3.2 Flowcharts/DFDs/ERDs

3.2.1 Flowcharts:

Flowchart is the pictorial representation of the working process of the model.It consists of the control flow and decision rules .Here,it has no loops.According to this flowchart,first of all the user uploads the image.Then ,with the help of transfer learning,CNN detects the objects.If the object found, it detects objects from the images,flickr 8k dataset gets trained and captions are generated based on training otherwise user has to upload another image for the further process.

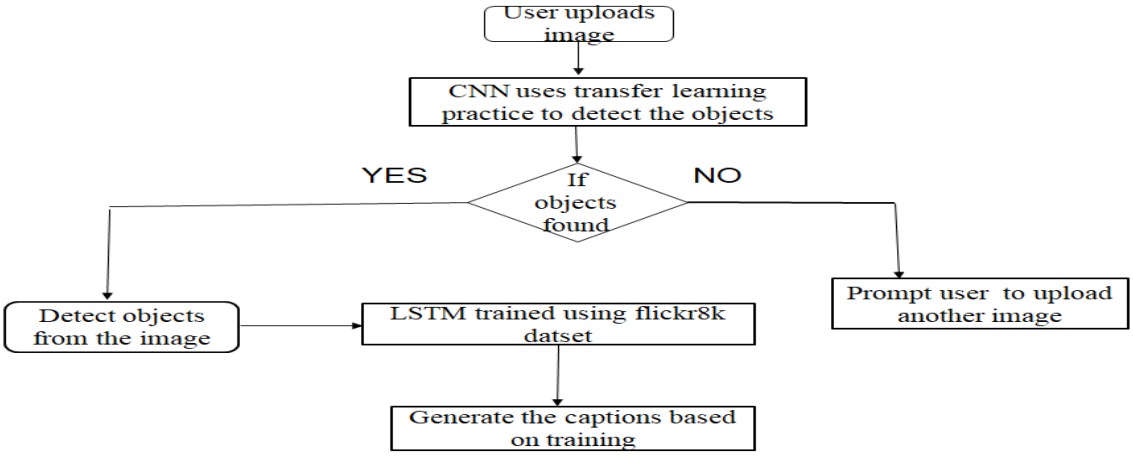


Fig 7. Flowchart Diagram

3.2.2. Use Case Diagram

Use the case diagram model in the performance of treatment system characters and application cases.Initially, the user upload the images and read the captions as the output.Then,convolutional neural network(CNN) detects the object and recognize the objects like what are they,their size ,colors of the objects etc.After recognizing the objects presents in the input images ,CNN provides the name of the objects in a proper way.Now,turn goes to the long short term memory,LSTM takes the object's name as the inputs that further goes for the training process.It trained itself with flickr 8k dataset consisting of both the training and testing images.After getting trained on flickr 8k dataset, relevant captions are generated with accuracy.

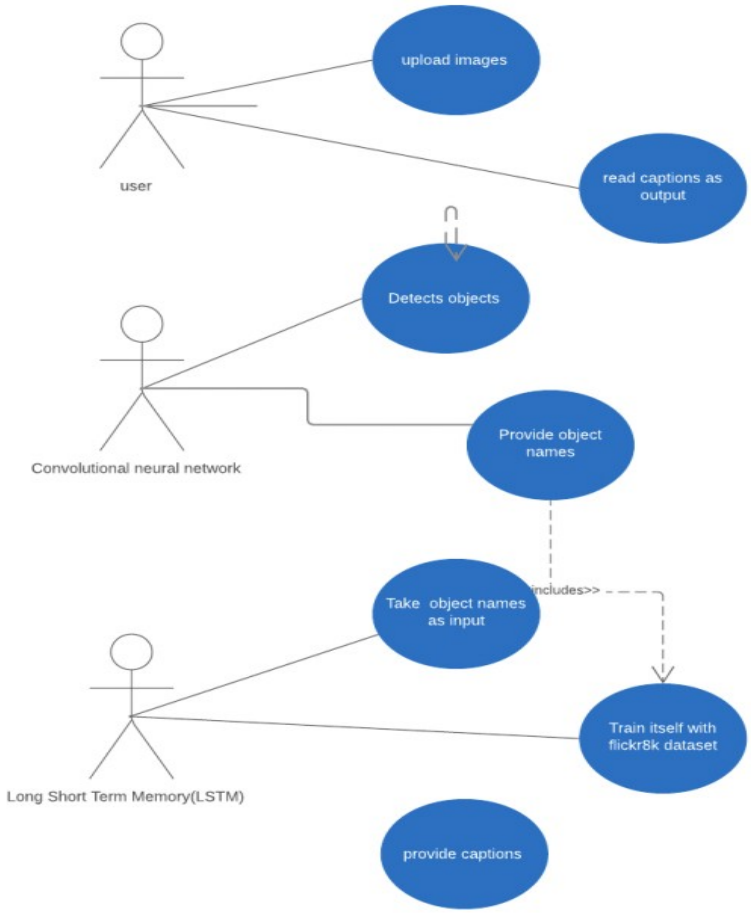


Fig 8. Use Case Diagram(UCD)

3.2.3 Class Drawing:

Classroom paintings are the backbone of every approach focused on something and UML. They describe the structural structure of the system. Categories represent partners certification of businesses with similar characteristics. Organizations represent relationships between categories.

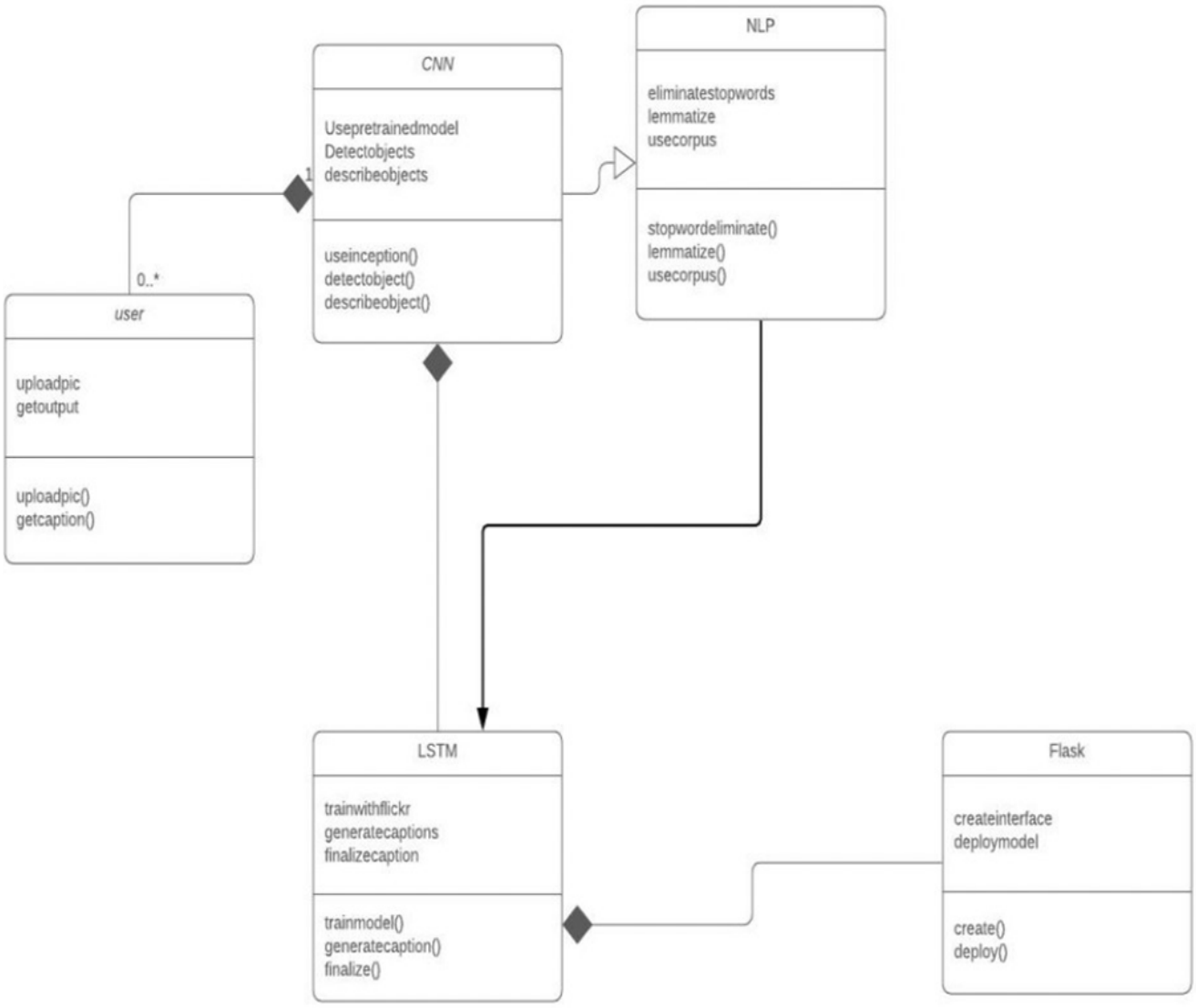


Fig 9. Class of Working process

3.2.4. Component drawing:

The feature diagram describes the arrangement of objects in a large system. The feature can be a system building block. picture a parallel program with tabs**.**

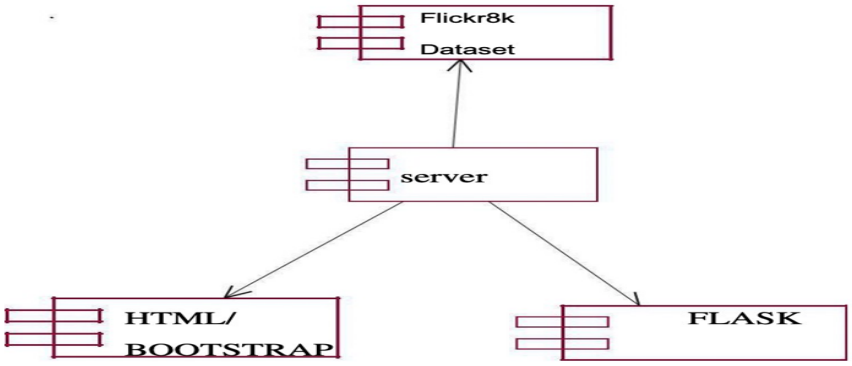


Fig 10. Building Block

3.2.5.Deployment Drawing:

Distribution diagrams show visual resources in an advanced system and nodes, components, and links. A node can be a visual aid to make parts of the code.

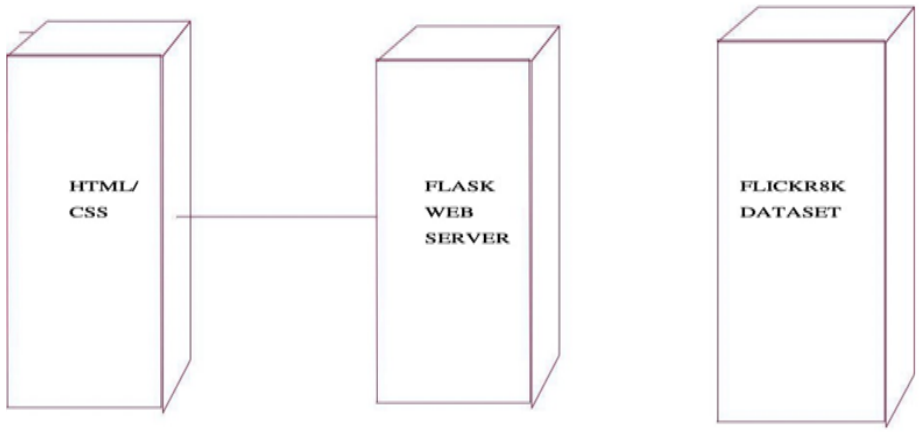


Fig 11. Deployment Drawing

3.2.6.Sequence diagram:

A sequence diagram shows the interaction of an object arranged in chronological order. Displaying items and classes involved in this scenario and the sequence of messages have changed between what is needed to make the task work.

This diagram consists of user,CNN,LSTM.User uploads the images.Under CNN,transfer learning is used for the detection o0f the objects.It creates the names of the objects.These names are passed to the LSTM.This message gets trained with flickr 8k dataset.Then,captions are generated which provides captions through flask interface to the user.This is how sequential diagram works.

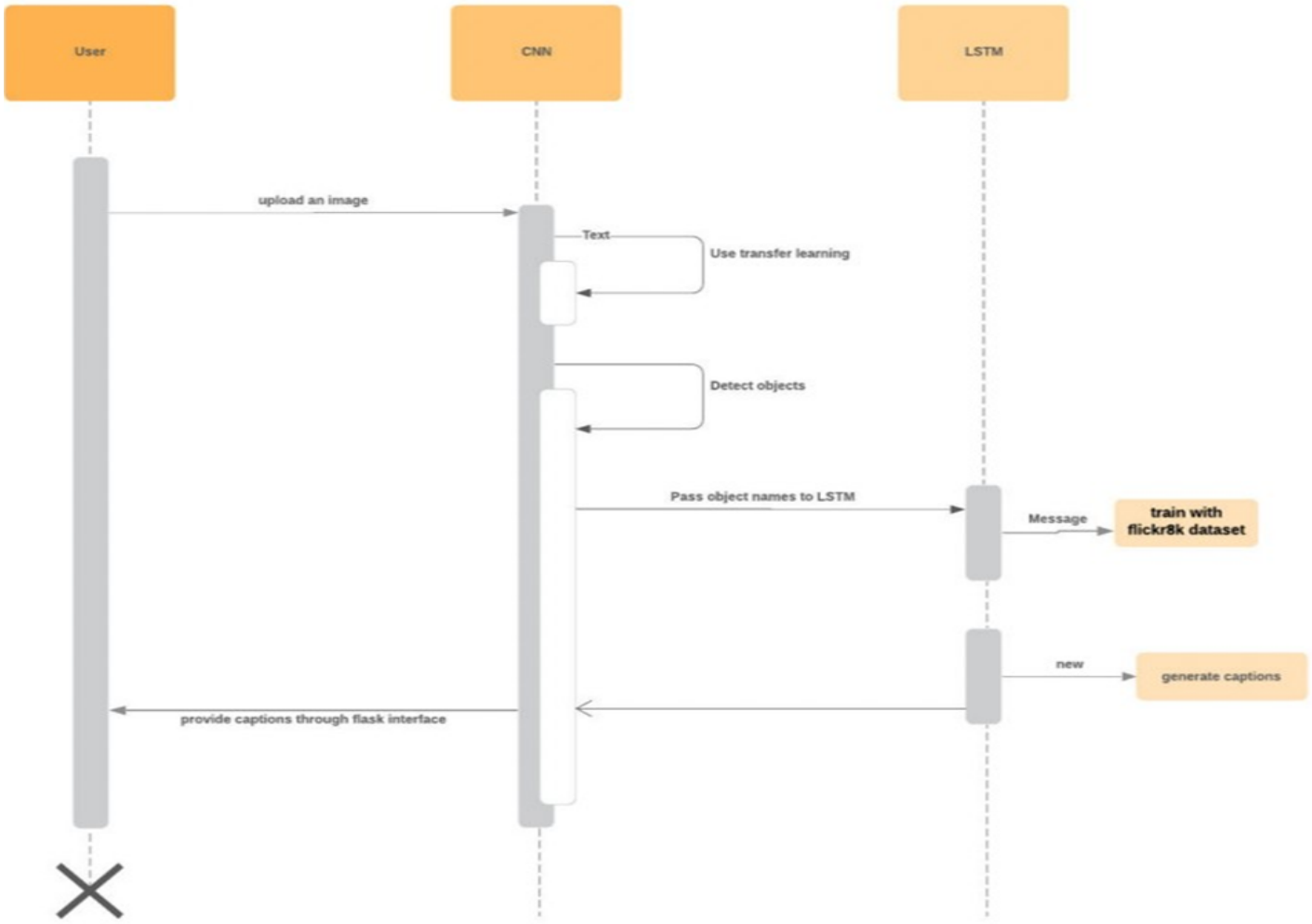


Fig 12.Sequence diagram

3.2.7.Activity diagram:

The task diagram shows the dynamic nature of the system by modeling the flow of management from work to work. The function represents the AN function to open a particular category within a system that leads to amendments within the area of the system. In general, work drawings are usually a continuation of a model or business processes and internal functioning. As a result of AN drawing work may be special A chart diagram of a specific situation, using many continuous modeling principles**.**First of all,images are uploaded,After getting uploaded,image formats are verified for the further process.If the provided image format is valid,it detects the object from the images otherwise it prompts the user to re uploading the images.If,objects are found in the images,names of those images are described otherwise prompts the user for re-uploading the images.After describing the object name,these names are passed to the LSTM layers then LSTM layers firstly get trained on the flickr 8k dataset.After this,it generates the captions.Finally, it shows the best captions.

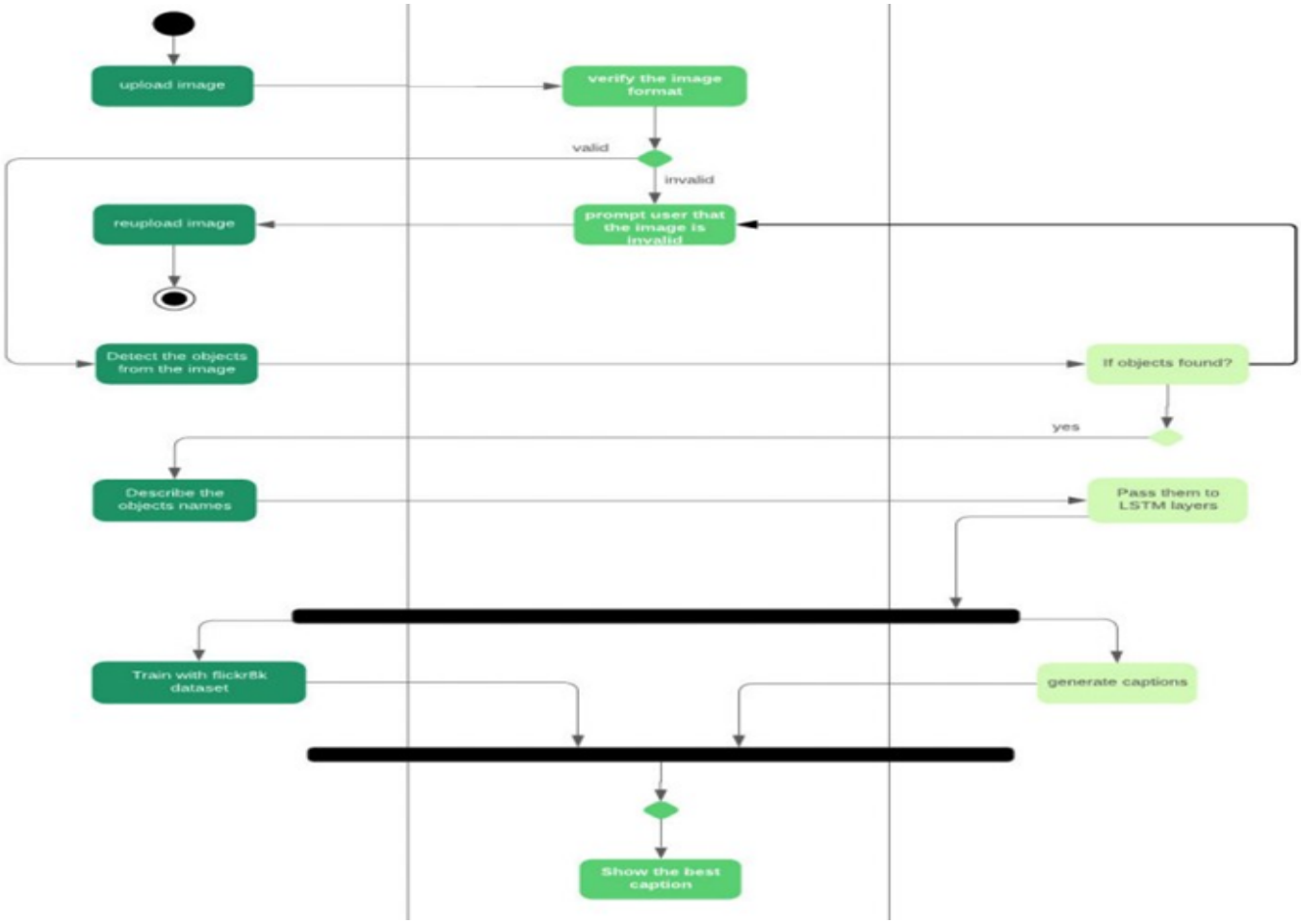


Fig 13.Activity diagram

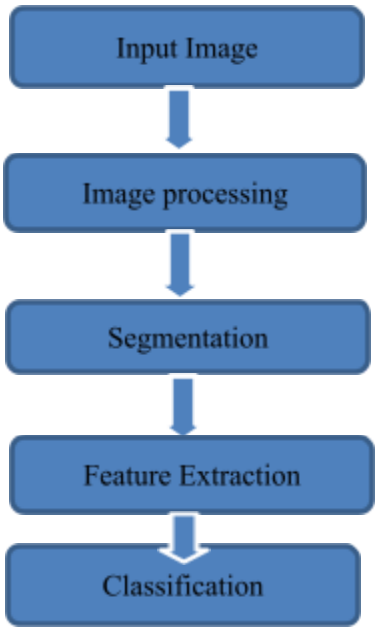
3.3.Design and Implementation

3.3.1 Overview

The main motive behind this project is to create a reliable text in the form of the particular caption of the provided image. CNN-LSTM model is used here for the generation of the caption. Flickr-8k datasets are used here for training the model and the indicator BLEU is used here for the evaluations. This is applicable for the visually impaired people.

3.3.2 Input Image

In this project, we are using the images of flicker-8k datasets as the inputs. Basically, the user inputs the images and these images get detected by the process of the model for the further selection, extraction of the feature found in the object present in the input images. After getting extracted, the input images are used for training the model or we can say that these datasets train the model and also test the model.



3.3.3 Image Preprocessing

Image processing is all about enhancing and extracting information from the input images.The input images are first read by the model used for the project.CNN then resize the picture after reading them.It removes noise from them.Examples of image pre-processing are noise reduction as well as brightness and contrast enhancement.The CNN decodes the content of the picture to grids of pixels.

3.3.4 Image Segmentation

In this process, segmentation sub-divides an image into its constituent regions or objects.Segmentation is based on grayscale.

Segmentation is based on texture also.So,texture is also another characteristics which can distinguish between objects into the images.We have another type of segmentation also which is based on motion.Suppose we have an image of running car.In that image,in what motion ,the car is going on,it represents the motion of the picture and spilt it in words.Now,we have another segmentation which is based on depth.Suppose,we put original image ,after having the segmentation of that image,we obtain a range image then after having a certain algorithm of the image segmentation,we find segmented image.Segmentation should stop when the focus areas of an application have been isolated. For example: the automated inspection of the electronics system. so,here interest lies in analyzing images of the product.

3.3.5 Feature Extraction

The feature is extracted from the image using CNN models. We employ the Visual Geometry Group (VGG 16), which is a pre-trained model on the imageNet datasets, as well as the Vgg16 (16 layers) object recognition and classification algorithms, which are capable of categorizing 1,000 photos into 1,000 different classes with 92.7 percent accuracy. Vgg16 is a common picture classification technique that is simple to employ with transfer learning.

The Vgg-16 model is built on the convolution layer of filter 3*3 with a stride 1 (s=1, move one pixel) and always employed the same padding, as well as the Max pooling layer of filter 2*2 of stride 2.
The convolution and max pooling layers are still active throughout the design. Finally, using softmax as an output, it has two fully connected layers.

Module 1 (Dataloader.py):-

The following improvements were made:-

- Filenames and data routes
- Flickr-8kdataset
- loads data
- create a descriptions
- Unique images id's along with descriptions

Module 2 (Model.py):-

The following improvements were made:-

- Pre-trained InceptionV3, VGG16 on the ImageNet datasets.
- Configure CNN, LSTM
- Configure optimizer to train NN
- Make CNN-LSTM layers and then return the results.
- Encoding image vector
- Decodes the encoded data(information) one word at a time to generate the output,to train the model

Module 3 (CleanDescription.py):-

- Load the captions file then open and read the file
- To generate the descriptions from the caption file.
- Clean the description(removing punctuation,convert all word to lowercase(),remove word that contain the number)
- save the new descriptions(clean descriptions)

Module 4 (Main.py):-

The following improvements were made:-

- Train the model
- Validate the model
- Recognize text in a picture using the file path as a guide.
- Main function

Module 5(Upload.py):-

- To run on the web, we used the Flask framework(python)

i)Flickr-8k Image dataset:

The Flickr 8k dataset comprises 8000 images,each paired with 5 captions for each and every image .Out of 8k images,6000 images are used for training the model,1000 datasets are used for validation and 1000 datasets are used for testing the model. There are many other large dataset like flickr-30k dataset ,MSCOCO datasets etc but we used only flickr-8k dataset because it is realistic and small dataset which helps to train the model easily as it takes less time as comparison to other datasets.If we talk about MSCOCO dataset,it's fact that we can build a good model using this dataset but the problem is it takes a lot of time ,almost one week to train the model.

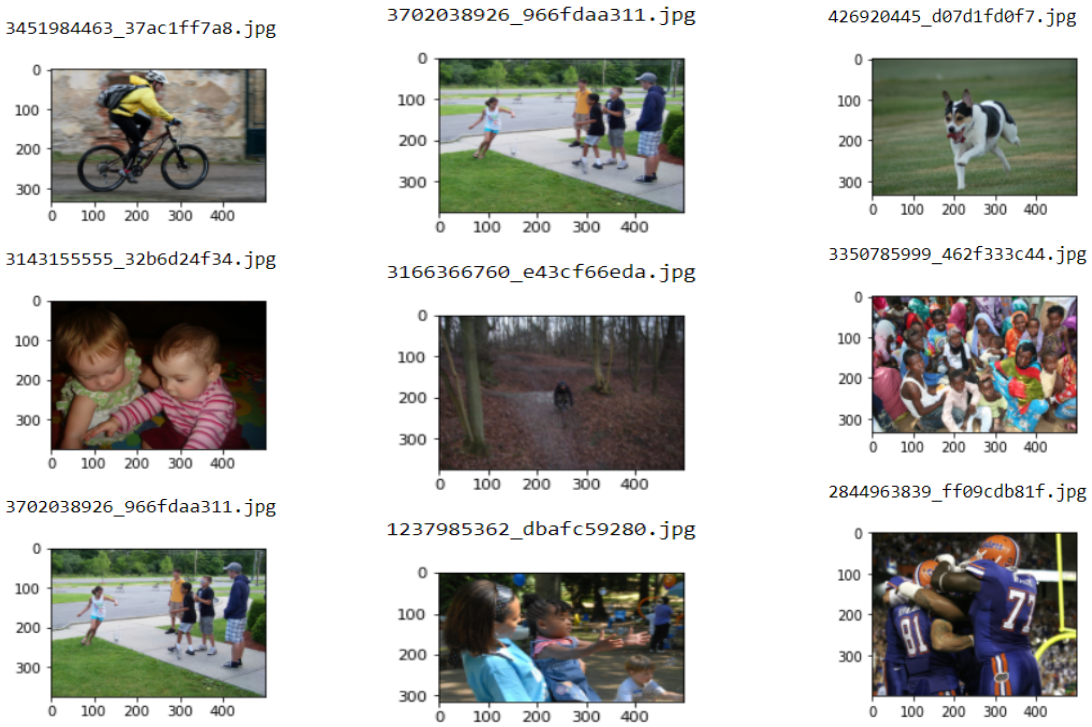


Fig14 : Plot of Images

ii)Flickr-8k captions:

The Flickr-8k captions dataset(flickr-8k.caption.txt) contains 40,000 text descriptions, image-id along with the five captions.

	image_id	caption
0	1000268201_693b08cb0e	[child in a pink dress is climbing up a set of...
1	1001773457_577c3a7d70	[black dog and a spotted dog are fighting, bla...
2	1002674143_1b742ab4b8	[little girl covered in paint sits in front of...
3	1003163366_44323f5815	[man lays on a bench while his dog sits by him...
4	1007129816_e794419615	[man in an orange hat starring at something .,...
5	1007320043_627395c3d8	[child playing on a rope net ., little girl cl...
6	1009434119_febe49276a	[black and white dog is running in a grassy ga...
7	1012212859_01547e3f17	[dog shakes its head near the shore , a red ba...
8	1015118661_980735411b	[boy smiles in front of a stony wall in a city...
9	1015584366_dfcec3c85a	[black dog leaps over a log ., grey dog is lea...
10	101654506_8eb26cfb60	[brown and white dog is running through the sn...
11	101669240_b2d3e7f17b	[man in a hat is displaying pictures next to a...
12	1016887272_03199f49c4	[collage of one person climbing a cliff ., gro...
13	1019077836_6fc9b15408	[brown dog chases the water from a sprinkler o...
14	1019604187_d087bf9a5f	[dog prepares to catch a thrown object in a fi...

Fig 15 : Image Captions

Read the Caption File

i) Read the file

ii) split the strings of text into a list using split() method.

iii) Five different captions for a particular image ids.


```
Image-id:2498897831_0bbb5d5b51.jpg

Out[7]: ['children are pulling faces on a purple bench .',
         'little girls are sitting on a purple bench and making funny faces .',
         'little girls make silly faces on a wooden swing in an over saturated photo .',
         'young girls are sitting on a bench swing in a park .',
         'young girls pose together outside .']
```

Implementation:

1.Import the requirement libraries
2.Loading data
3.Generate and perform data cleaning
i)Load descriptions for each image.
ii)Clean the description by removing punctuations, converting all words to lowercase and removing words that contain numbers.
iii)create a vocabulary from all the descriptions
iv) save the descriptions.
v) loading the train data
4. Data Preprocessing -images and captions
5.Extract the feature from all images using VGG16
(Removing a last layer (output layer of 1000 classes))
6.Training the model
7.Tokenizer the vocabulary
8.Create data generator
9.Model Architecture(cnn-lstm model)
10.Model Evaluation- Belu(Bilingual Evaluation Understudy Score)
11.Testing the model

1) Import all the necessary libraries

Firstly, we have installed keras and tensorflow using an anaconda navigator.
Tensorflow is a machine learning library. keras is a deep learning library for theano and tensorflow. We have imported some libraries from keras and tensorflow such as tensorflow.keras.applications.vgg16,tensorflow.keras.preprocessing.image,tensorflow.keras.preprocessing.sequence,tensorflow.keras.layers.Input,Dense,Dropout,Embedding,LSTM .
we also used the following modules:

a) keras VGG16 modules:

i) VGG16( ) model
ii)Remove the last layer from the vgg16 model

i) VGG 16 model

```
VGG16_model=tf.keras.applications.VGG16(
                include_top=True,weights="imagenet",input_shape=None,classes=1000, classifier_activation='softmax' )
VGG16_model.summary()
```

include_top: wheather or not to consist of the fully connected layer on the top of the network layer.
weights**:** one of None, imagenet (pretrain on imagenet)
input_shape**:** if include_top is false then input shape must be (3,224,224).

classifier_activation: if classifier_activation is None to return the logits of the top layer. In another case,when loading pretrained weights,classifier_activation can most effectively be "none" or "soft-max".

ii)Remove the last layer from the Vgg 16 model:

To remove the last layer and fine-tune model.layer.pop( ) reserving the already trained weight. Using model.summary(),it is possible to verify that the last layer of the model has been replaced.

```
#removing the last layer (output layer of 1000 classes)from the VGG-16
# as we require only the features not the classification
VGG16_model.layers.pop()
new_VGG_model=Model(VGG16_model.input,VGG16_model.layers[-2].output)
new_VGG_model.summary()
```

b) tensorflow.keras.preprocessing.image module:

i)load_img(image_path,grayscale=False,color_mode=rgb,target_size=(224,224)):load the image into python image library format.

- image_path: img-path to image file.
- grayscale: if color_mode=grayscale is depreciated.
- color_mode: one of color_mode=grayscale,color_mode=rgb, color_mode=rgba
- target_size:Default to original size (target_size=None) or target_size=(img_height,img_width)

```
def preprocess_image(input_image)
    image_path=("C:\Flickr8k\Flickr8k_dataset\LittleGirl.jpg")
    image=tf.keras.preprocessing.image.load_img(image_path,grayscale=None,color_mode='rgb',target_size=(224,224))
    input_img= tf.keras.preprocessing.image.img_to_array(image)
    input_img=np.expand_dims(input_img,axis=0)
    input_img= tf.keras.applications.vgg16.preprocess_input(input_img)
    return input_img
```

c)tensorflow.keras.preprocessing.sequence.pad_sequences(...):

```
tf.keras.preprocessing.sequence.pad_sequences(sequences,
                        maxlen=None,dtype='int32',padding='pre',truncating='pre',value=0.0)
```

- sequences: Each sequence is a list of integers.(data type=32)
- maxlen: The maximum length of any sequences.if it is not provided, sequences can be padded to length of single sequences.
- padding: By default the padding=pre (string) otherwise padding=post: padding either before or after each and every sequence.
- truncating: By default the truncating= pre (pre or post)-:delete the values of sequences higher than maxlen ,At the beginning of sequence or at the end.
- value:By default to 0, float or string ,padding value.

2.Load data:

- Flickr-8k Images- contains 8092 images,each paired with 5 captions for each and every image
- Flickr -8k Caption-(caption.txt) 40,000 text descriptions, image-id along with the five captions.

3.Generate and Perform clean descriptions:

Each image contains five descriptions (captions). The important function here is Clean description, which takes all of the descriptions and cleans them up:

- Removing punctuations
- Removing Words that contain number
- Converting all description in lowercase
- Removing special tokens(such as '%', '$', '#', etc.)

We tokenized our data and employed a fixed vocabulary size of 8,680 words.

The removal of noise is one of the most crucial steps of NLP since it allows the machine to detect patterns in the text more quickly. Special characters like hashtags, punctuation, and numerals will be used to create noise. If any of these elements are present in the text, computers will have difficulty understanding and interpreting them. As a result, we must remove them in order to achieve better results. The NLTK library can also be used to remove stop words, as well as to conduct stemming and lemmatization.

4.Count the repeated words:

The frequency of a word is the number of times it appears in a description file.

The size of vocabulary is 3,57,158. The top ten most often used words are as follows:

```
[('a', 38280), ('.', 34212), ('in', 18961), ('the', 15293), ('on', 10729), ('is', 9345), ('and', 8850), ('dog', 7970), ('with',
7760), ('man', 6808), ('of', 6712), ('white', 3873), ('black', 3694), ('are', 3503), ('boy', 3433), ('woman', 3217), (',', 321
0), ('girl', 3209), ('to', 3173), ('wearing', 3061), ('at', 2904), ('water', 2774), ('red', 2660), ('brown', 2470), ('young', 2
426), ('people', 2420), ('."', 2365), ('his', 2357), ('blue', 2257), ('dogs', 2077), ('running', 2072), ('through', 2031), ('pl
aying', 2008), ('while', 1957), ('an', 1899), ('down', 1823), ('shirt', 1803), ('standing', 1786), ('ball', 1778), ('grass', 16
21), ('little', 1610), ('snow', 1481), ('child', 1472), ('jumping', 1469), ('over', 1414), ('person', 1404), ('front', 1386),
('sitting', 1368), ('holding', 1324), ('field', 1276)]
```

Because the words that appear infrequently do not convey much information. We're looking at words that have a frequency of greater than ten.

```
Out[19]: Text(0.5, 1.0, 'plot of count vs word')
```

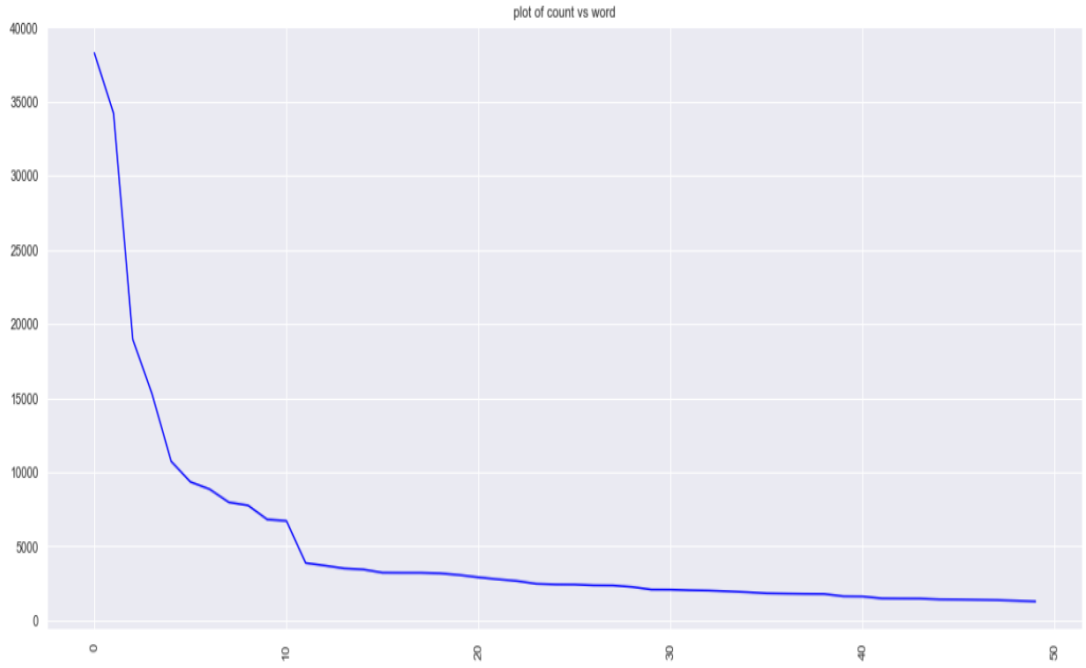


Fig 16. Plot of count vs word

Distribution of word and count:

The top 50 words are distributed as follows:

```
Out[19]: Text(0.5, 1.0, 'Plot of count vs words')
```

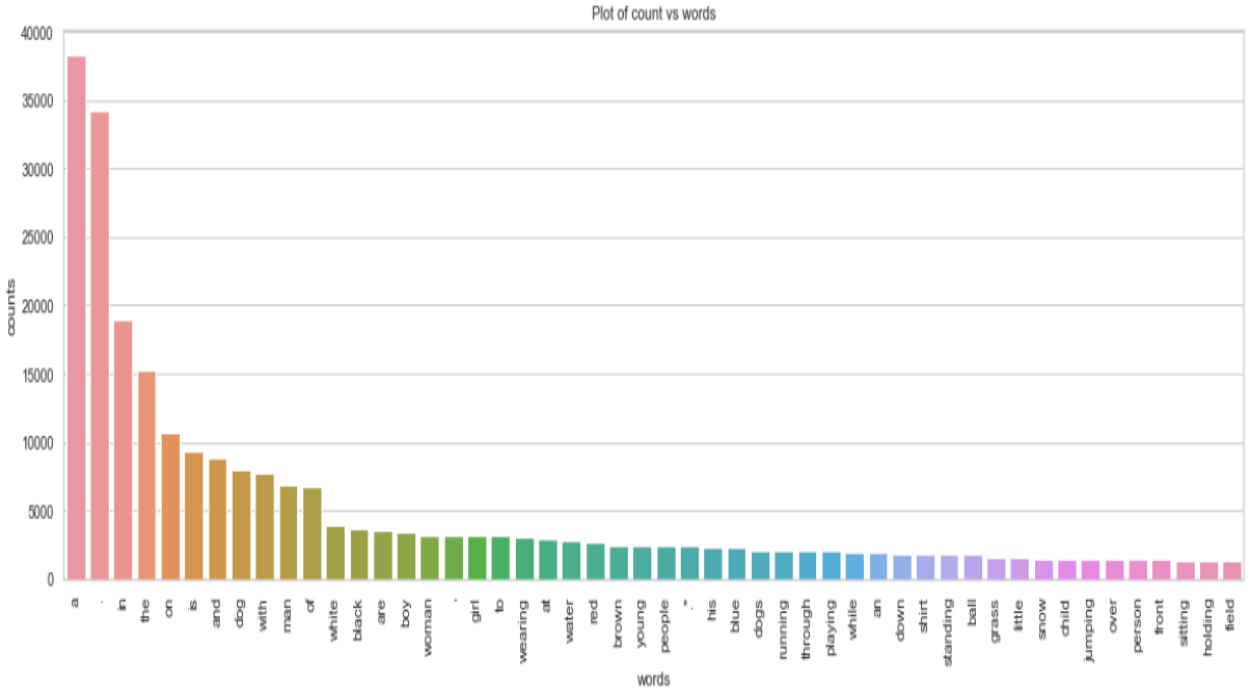


Fig 17. Plot of counts vs words(Descriptions)

5. Data preprocessing for images:

It's obvious that we can't feed an image directly into our model, therefore the first step is to do some preprocessing: Reduce the size of each image to (299 * 299) for the Xception model and (224 * 224) for the VGG16 model. 2- Flatten it, and 3- Image pixel scaling (normalization).

6.Extract the feature from all images using InceptionV3:

Now we offer our model an image as input, but machines, unlike humans, cannot understand images just by looking at them. As a result, we must turn the image into an encoding so that the system can recognize the patterns. we're utilizing transfer learning for this challenge, which means we take a pre-trained model that has already been trained on massive datasets and extract the characteristics from it to use in our work. We are using the InceptionV3 model which was trained on the Imagenet dataset, which had 1000 different classifications to sort through. This model can be easily imported from the Keras. applications module.

To get the (2048,) dimensional feature vector from the InceptionV3 model, we need to remove the last classification layer. Encode the image vector and shape.

```
[11]: 1 encode_img=encode_image(image,new_model)
      2 print(encode_img)
      3 encode_img.shape
```

```
[[0.41659048 0.65269697 0.49971983 ... 0.28184775 0.84851116 0.33954278]]
```

```
t[11]: (1, 2048)
```

7.Create a model:

We'll use the Keras Model from Functional API to define the structure of our model. It consists of three major steps:

i) processing the sequence from the text

ii)Extracting the feature vector from the image

iii) By concatenating the two levels previously, the output is decoded.

```
In [29]: 1 plot_model(model_ic, to_file='model_ic.png', show_shapes=True)
Out[29]:
```

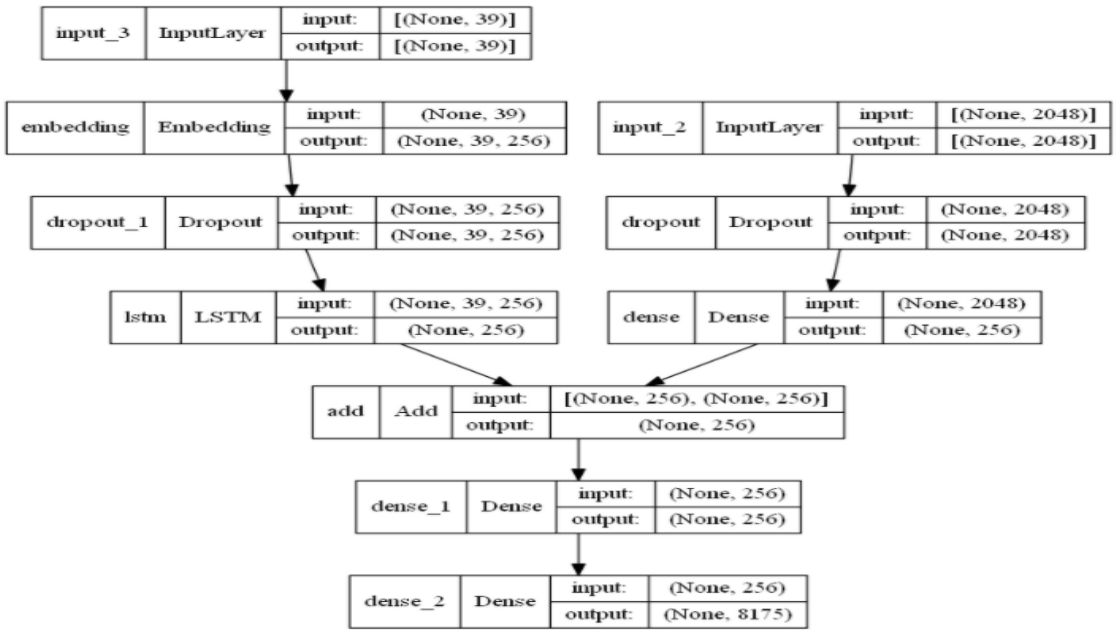


Fig18. Caption Generator model(deep learning model)

8.Glove Vector Word Embeddings:

GloVe (global vectors for word representation) is a term that stands for "global vectors for word representation." It's a Stanford-developed unsupervised learning system for creating word embeddings from a corpus' global word-word co-occurrence matrix.

Furthermore, we have 8000 photos with five captions each.That means we have a total of 30000 samples to train our model with. We can utilize a data generator to feed information to our model in batches rather than all at once.We'll also store the relationships between terms in our vocabulary using an embedding matrix.

```
[58]: 1 for word, i in word_to_index.items():
      2     embedding_vector = embeddings_index.get(word)
      3     if embedding_vector is not None:
      4         # Words not found in the embedding index will be all zeros
      5         embedding_matrix[i] = embedding_vector
      6
      7 print("Embedding_matrix.shape = ",embedding_matrix.shape)
Embedding_matrix.shape =  (8175, 200)
```

9. Divide all captions into train, valid, test:

```
Train-Caption:6000
<BEGIN> Two draft horses pull a cart through the snow .<END>

Valid-Caption:1000
<BEGIN> A woman dressed in a blue jacket and blue jeans rides a brown horse near a frozen lake and snow-covered mountain .<END>

Test-Caption:1000
<BEGIN> A boy in yellow shorts is standing on top of a cliff .<END>
```

Fig 18. Train,Valid, Test Captions

10.Count the number of times each word appears in a train caption:

The size of the training vocabulary is 4253

```
4253
[('a', 6619), ('<BEGIN>', 6001), ('.<END>', 5436), ('A', 4805), ('in', 3058), ('the', 2070), ('and', 1570), ('on', 1545), ('i
s', 1483), ('dog', 1439), ('of', 1226), ('man', 1124), ('with', 967), ('black', 796), ('boy', 726), ('white', 645), ('girl', 62
2), ('brown', 609), ('are', 509), ('to', 453), ('at', 436), (',', 435), ('red', 392), ('his', 375), ('water', 374), ('woman', 3
64), ('wearing', 359), ('while', 347), ('group', 341), ('blue', 334), ('an', 318), ('people', 307), ('shirt', 306), ('running',
298), ('through', 283), ('playing', 282), ('down', 272), ('standing', 269), ('ball', 257), ('jumping', 237), ('Two', 236), ('ch
ild', 235), ('over', 218), ('another', 217), ('two', 213), ('snow', 211), ('grass', 209), ('front', 208), ('green', 196), ('dog
s', 195)]
```

The top 50 words are distributed as follows:

```
Out[52]: Text(0.5, 1.0, 'plot of count vs word')
```

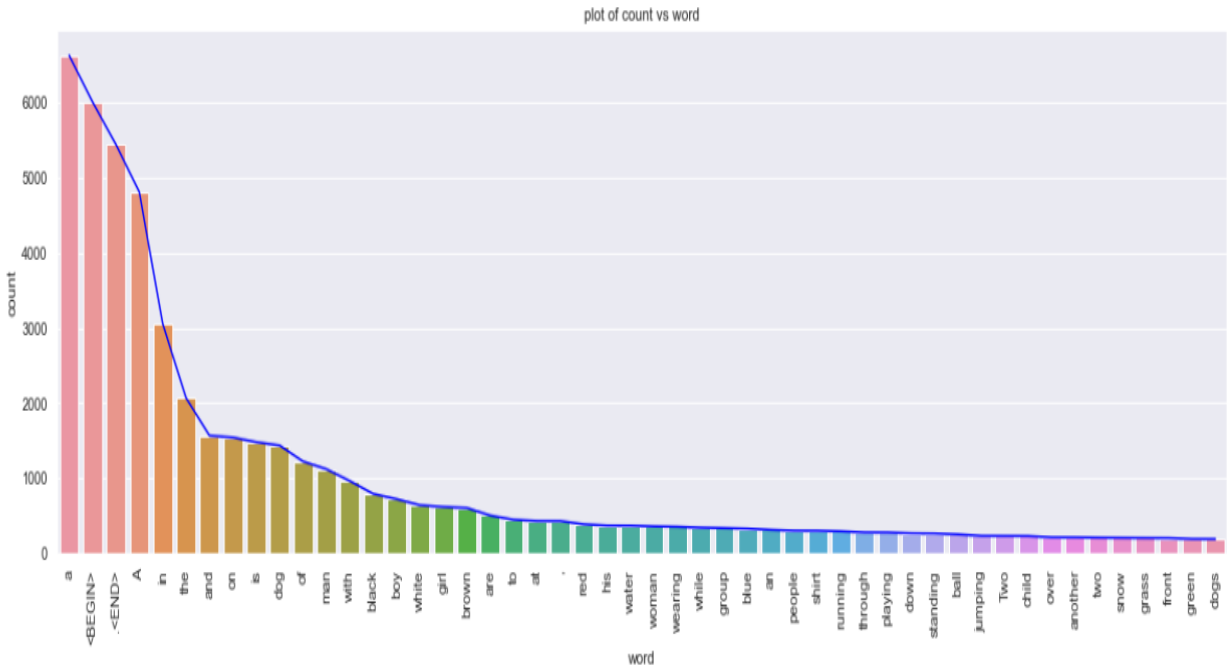


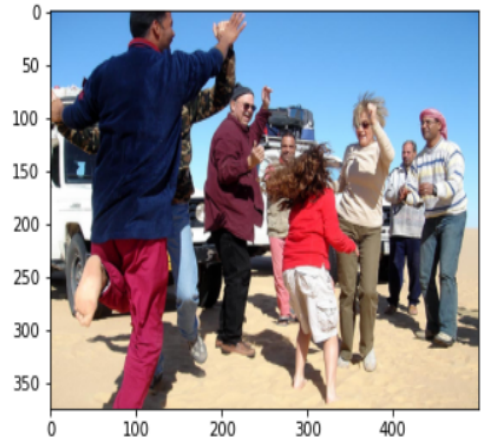
Fig 19. plot of count vs word (Train vocabulary)

11.Generated Captions:

We have successfully created generally correct syntax and human understandable captions summarizing what is happening in the image, as illustrated in Figures 20,21. The majority of the images accurately state what items appear in the scene, count the number of appearances, and provide a logically correct verb to finish the sentence. The colors of the object, as well as their spatial interactions, are effectively represented.

For Example, the left side of the image has generated captions “ girl dances with adults in the sand”. "girl," "adults," and "sand" are the identified objects in the image. So CNN may capture attributes of objects and the right side of the image has generated captions “Black and white dog is jumping in the snow at a park”. “dog”, “snow”, “park” are the identified objects in this image.then, CNN is also able to capture properties of objects such as the dog's color is “black” and “white”. Finally, the RNN and Attention Mechanism will use conjunctions to logically connect these words into a meaningful phrase.

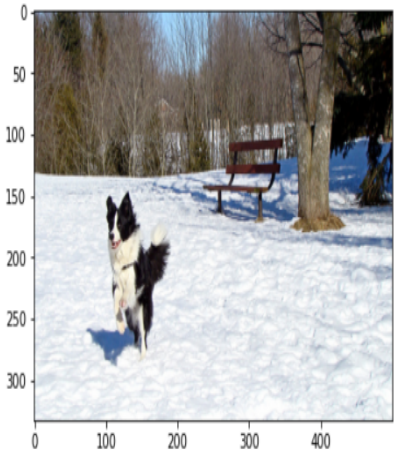
2295216243_0712928988.jpg



['<Begin>girl dances with adults in the sand .<End>']

Fig 20. “ Girl dances with adults in the sand”

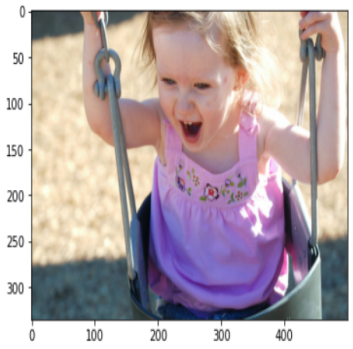
2301525531_edde12d673.jpg



['<Begin>black and white dog is jumping in the snow at a park .<End>']

Fig 21. “Black and white dog is jumping in the snow at a park.”

1682079482_9a72fa57fa.jpg

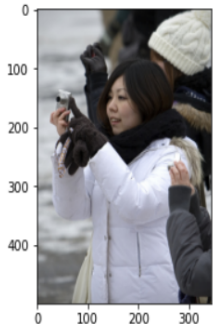


['<Begin>little girl on a kid swing .<End>']

170100272_d820db2199.jpg

['<Begin>child is on a slide .<End>']

2086513494_dbbcb583e7.jpg

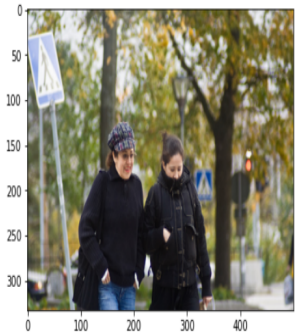


['girl in a white coat takes pictures .']

2105756457_a100d8434e.jpg

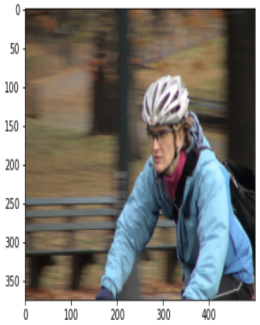
['children standing under an awning .']

2078311270_f01c9eaf4c.jpg



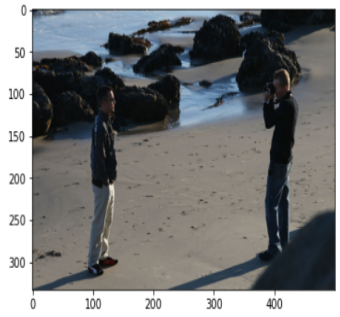
['<Begin>ladies walking on the sidewalk talking to each other .<End>']

2084217208_7bd9bc85e5.jpg



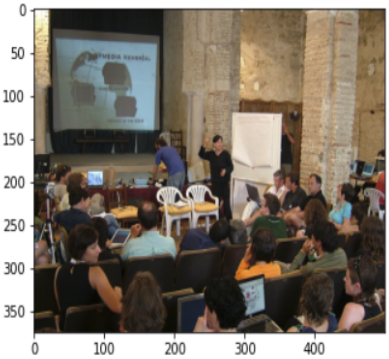
['person in a blue jacket , wearing a bicycle helmet is riding a bike"']

2102360862_264452db8e.jpg



['man on the beach is taking a picture of another man .']

['<Begin>class full of students .<End>']



['<Begin>boy with a toy gun .<End>']

Fig 22: Image and Caption generated

12. Model Evaluation- BLEU:

Bilingual Evaluation Understudy is represented as BLEU.
It's an algorithm for assessing the quality of the machine-translated text. We can utilize BLEU to assess the quality of the caption we've generated.

- BLEU is not limited to a single language.
- Easy to comprehend
- It is simple to calculate.
- It is located between [0,1]. The captain's quality improves as the score rises.

Calculate the Bleu Score:

Predicated Caption- " child is on a slide ".

References Caption- i) "A little girl is sliding down a slide at a park".
ii) "A girl goes down a blue and yellow slide at a park".

To begin, make unigram/bigrams out of the predicted caption and references caption.

$$\text{modified ngram precision} = \frac{\text{max number of times ngram occurs in referance}}{\text{total number of ngrams in hypothesis}}$$

For Example:

Predicated Caption- "A child is on a slide ".

References Caption- i) "A child is sliding down a spiral slide on a playground".
ii) "A girl goes down a blue and yellow slide at a park".

Predicted Caption : (A,child), (child,is),(is,on),(on,a),(a,slide)

References Caption: (A, child),(child,is),(is, sliding),(sliding, down),(down,a),(a,spiral), (spiral,slide),(slide, on), (on, a), (a ,playground).

BLEU(Bigram) = (1/5 + 1/5 + 0/5 + 1/5 + 0/5) = 3/5 = 0.6

12.1 EVALUATION:

Initially, there was a disagreement over which convolutional feature extractor should be used. As a result of their training on the Imagenet dataset, we chose VGG16 and Xception for identical decoding strategies. Instead of classifying an image, we want to create a fixed-length informative vector for each one. Automatic feature engineering is the name given to this method. Each image is then reduced to a 2048-length vector by removing the final softmax layer (bottleneck features). The second problem is to compare a single model to an ensemble. There have been studies showing that ensembles boost performance; however, our results only cover the performance of a single model.Only the BLUE Score's VGG16 features were used in our analysis.

Metric	VGG-16
BELU-1	0.460179
BELU-2	0.320112
BELU-3	0.100245
BELU-4	0.040572

Table 3: BLUE score

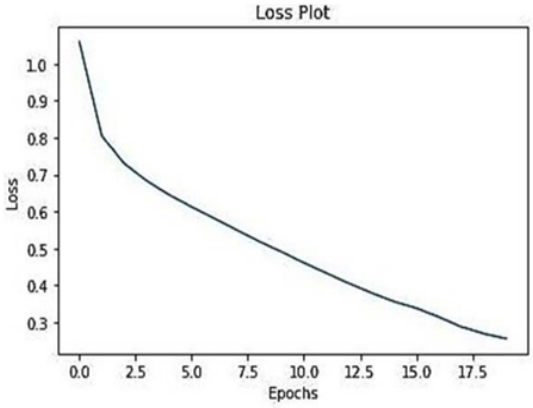
Result of Flick-8k using Neural Network training steps:

In the training phase, 6221 photographs are used.

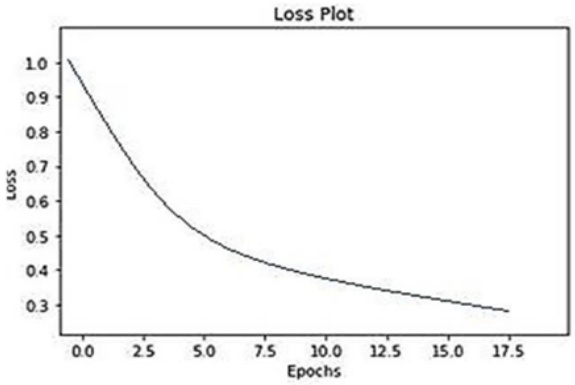
Initially,we defined and fitted our model.Then our model got trained for 20 epochs.At the training,it has been observed that the accuracy of the starting epochs is little bit low.We observed that whether the accuracy is increased or not if our model is trained for 20 epochs .Our model is trained for 20 epochs.Then,we know that the generated captions are related to the provided images after testing or not.After the first epoch,the weights get decent.By feeding the training data to our model again and again,we can improve the weights further or can get the correct captions related to the provided images.The following graph plots are the loss representation without ADAM and with ADAM which is loss vs epochs.

```
Epoch 1/5
6221/6221 [==============================] - 801s 128ms/step - loss: 4.0651
Epoch 2/5
6221/6221 [==============================] - 807s 130ms/step - loss: 2.9229
Epoch 3/5
6221/6221 [==============================] - 807s 130ms/step - loss: 2.6581
Epoch 4/5
6221/6221 [==============================] - 806s 130ms/step - loss: 2.5100
Epoch 5/5
6221/6221 [==============================] - 820s 132ms/step - loss: 2.4098
```

Loss presentation without ADAM



Loss presentation with ADAM



3.4 Testing Process

3.4.1.Software Testing:

3.4.1.1 Introduction:

Software testing is termed as an activity to check whether the actual results is matching with the expected results or not.Software testing is very important nowadays in different sectors as it also saves money for long term.Software development consists of different phases and if bugs are detected in the earliest stages,it costs much less to fix them that's why it is important.

3.4.2 Unit Testing:

It is the phase of the testing process where the individual units/modules of the system are tested.In this testing,the performance is done by the developers before handing the setup to the testing team for the execution of the test cases.It is performed on the individual units of source code assigned areas.We do testing by using the following process:

- After uploading the images,preprocessing is done.
- Segmentation was applied
- We extract features by using feature extraction.
- Take out a relevant captions

3.4.3 Integration Testing:

This is the testing process where individual modules are combined and tested as a group.The aim of this testing is to expose faults in the interaction between the integrated units.Data is grouped into larger aggregates for this testing.It focuses on determining the correctness of the interfaces.There may be chances of getting data lost,global data,subfunction etc.To prevent from these,integration testing is done.Unit tested components are taken one by one and integrated incrementally.In integration testing, each time,a new module is integrated,the subsystem changes,new input/output may occur,new data flow paths and control logic may cause problems.

3.4.4 Validation Testing:

This testing gives focus on the user-visible actions and user recognizable output from the system.This is done when series of tests demonstrates the existence of functionality required by the user in software we are using.After getting this test,performance constraint met,functionality is achieved,documents are correct,correct behavior of the working performance of the system is achieved.When individual components have been exercise, the software is completely associated as a package and interface errors have been corrected.The scenario of validation testing is that ,to what extent,the customer is satisfied with the developed software.

3.4.5 GUI Testing:

Basically,GUI testing is the part of testing which consists of the checking process of the screens with the controls like menus,buttons,menu bar,icons,toolbar,dialog boxes and windows,content spelling etc.

We can check the GUI elements for size,position,length and width of characters or numbers.We can also check whether the provided images have good clarity or not.Here,testers test the application part nad design to see whether it meets the client's requirements or not.Tester is more concerned with look and feel of the application .Tester tests the appearance of the software.We only focus on the interface of the application. We check whether the design layout of the application as per the standards and client's requirements or not.We test only the front end of the application.

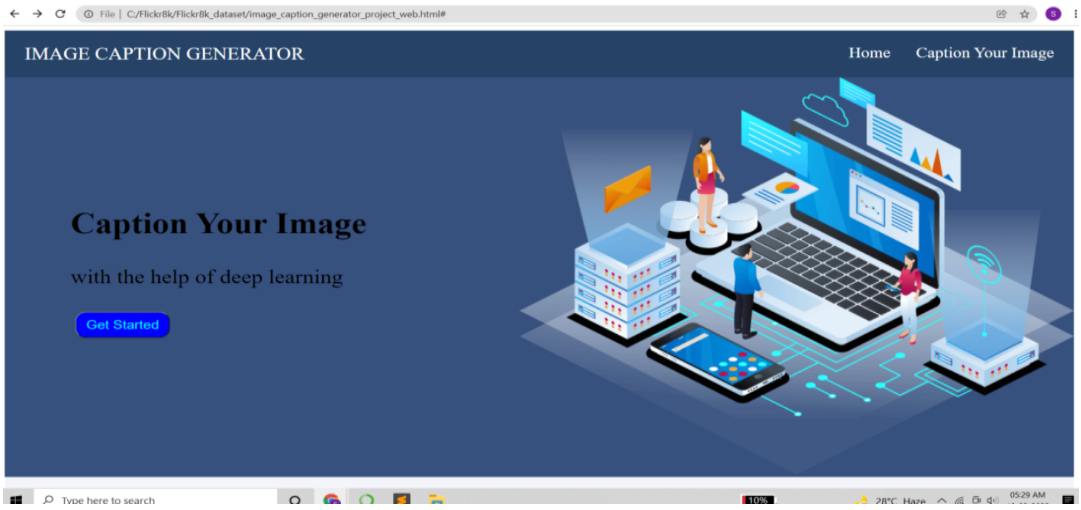
Test Cases:

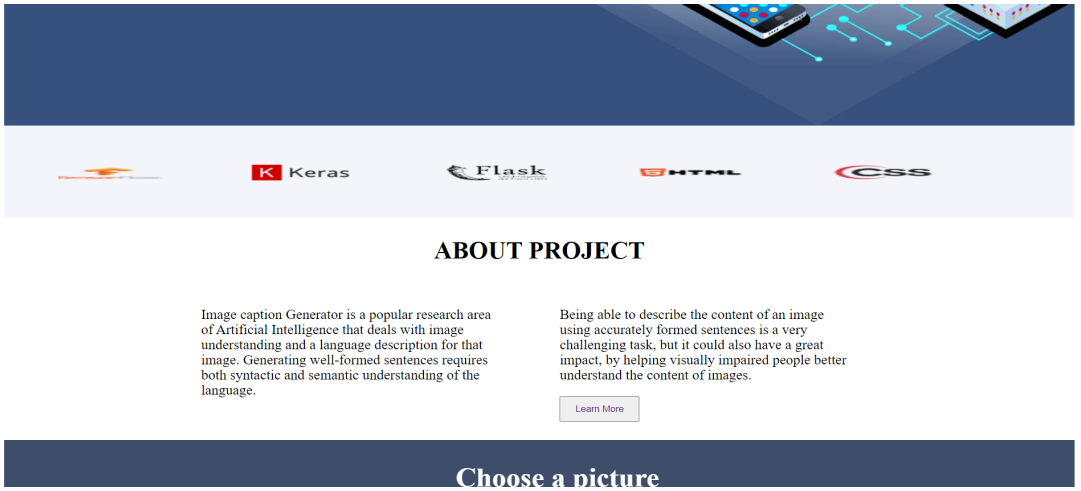
Use Case ID	1
Test Case Name	Check the format of the input images
Test Case Description	For the further process, image format must be in proper way
Steps	i. Open the website ii. Provide image with proper model
Expected Results	Acceptance of uploaded images and displayed

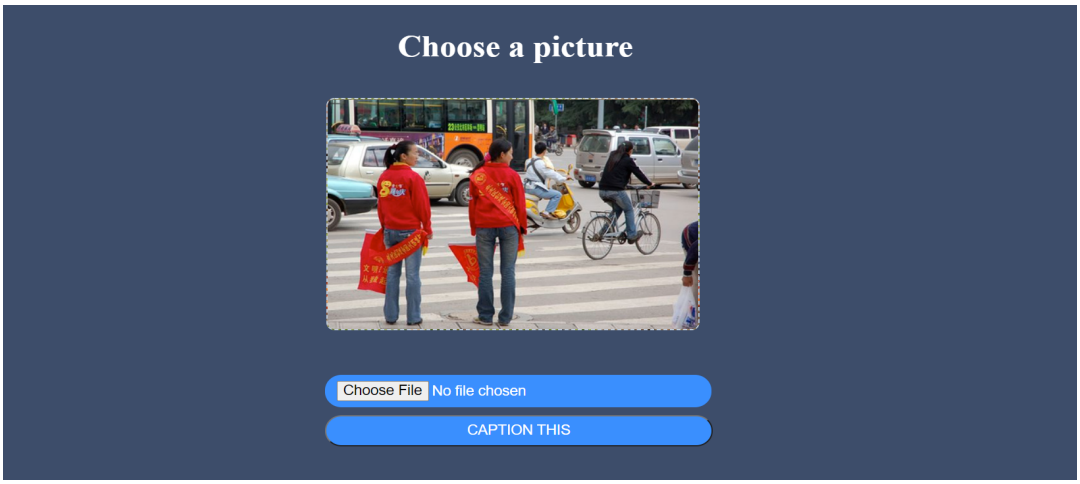
Use Case ID	2
Test Case Name	Training is done
Test Case Description	Commence training images
Steps	Commence training of images which are in .jpg file
Expected Results	Training should be done with accuracy
Actual Results	As expected

Use Case ID	3
Test Case Name	Recognize the objects from the input images
Test Case Description	Object presented in the images should be displayed resulting the correct captions based on them.
Steps	Acceptance of the images after getting scanned.
Expected Results	This application should have the ability to display the recognized objects from the input image.
Actual Results	As expected

Chapter 4: RESULTS / OUTPUTS






Choose a picture
Choose File
No file chosen
CAPTION THIS


Caption Your Image
Girls in red jackets are trying to cross traffic with a lot of cars going by .

Chapter 5: Conclusion

5.1 Conclusion:

We have surveyed CNN-LSTM model based on deep learning techniques in this project.We used a deep learning approach to caption the photos.To create the deep learning architecture,the sequential API of Keras is used with tensorflow as a backend to produce an effective BLEU score of 46.0 percent with Vgg 16 model.We have given the block-diagram of the whole project and some of the model's advantage and limitations are highlighted in the tabular form according to the previous research done in this field.We talked over different datasets and the evaluation metrics along with their positive aspects and limitations also.We have used small datasets containing 8000 images i.e.flickr-8k datasets.This is used for training and testing the model .We have implemented CNN-LSTM model on jupyter notebook using keras and tensorflow.

5.2 Future Improvement

In the upcoming days,the importance of the image caption generator is going to be boundless in many different fields like in industries,in the vast social media.

As there is the vast need of image caption generators nowadays,there must be some improvements in this field for better scope in future.Some of the models used nowadays are not giving accurate results as mostly we are also using small datasets. Instead of using small datasets,we can use MSCOCO(i.e.a large scale object detection,segmentation and captioning datasets)which helps in getting a more accurate model.COCO captions consists of 1.5 million captions describing over 1,30,000 images.We can improve our results with the VGG16 model by making a variety of changes such as making use of the larger datasets to create a effective model.We can modify the model architecture ,for example by adding an attention module.By using the cross validation set ,overfitting of the model can also be learned.We can use beam search instead of greedy search during inference.

As we are not getting good accuracy,a complete research must be done for some methodologies to get better results which can be better in future scope.

We have elucidated about the generated captions for the objects presented in the images.With the ongoing advancement in the field of deep learning,we are not achieving a proper caption with high accuracy .May be,this is because of the thinking capacity of the machine as it can't think as like of human.So in the coming days,we hope to create the captions with best accuracy by increasing the improvement in the hardware requirements to make the effective model.There is also a thought to enlarge the model of image caption generator by doing the conversion of the picture to speech so that it can be very helpful for those having vision problem

Chapter 6: References

1. Karpathy, A., and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306 (2014)
2. Md.zakirhossian,Ferdoussohel,Mohd fairuz shiratuddin,Hamid laga,Mohammed bennamoun-Text to Image Synthesis for Improved Image Captioning (2021)
3. Akash Verma1, Harshit Saxena1, Mugdha Jaiswal1, Dr. Poonam Tanwar2- IEEE-Intelligence Embedded Image Caption Generator using LSTM based RNN Model (2021)
4. O.Vinyals, A. Toshev, S. Bengio, D. Erhan, "Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge", IEEE transactions on Pattern Analysis and Machine Intelligence, 2016.
5. "Anilkumar Holambe, Dr Ravinder C Thool, Dr S.M Jagade - Printed and Handwritten Character & Number Recognition of Devanagari Script using Gradient Features. International Journal of Computer Application Volume 2 No- June 2010."
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
7. K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in Proc. Int. Conf. Learn. Represent. (ICLR), 2015, pp. 1–14.
8. K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.
9. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261–2269.
10. Automatic Image and Video Caption Generation With Deep Learning: A Concise Review and Algorithmic Overlap,2020.
11. S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, ''A guide to convolutional neural networks for computer vision,'' Synth. Lectures Comput. Vis., vol. 8, no. 1, pp. 1–207, Feb. 2018.
12. J. Aneja, A. Deshpande, and S. Alexander, "Convolutional image captioning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 2018.
13. A. Mathews, L. Xie, and X. He, "SemStyle: learning to generate stylised image captions using unaligned text," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 2018.
14. "A gentle Introduction to deep learning Caption Generation Models", by Jason Brownlee, November 22 2017, For deep learning Natural Language Processing.
15. Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015.

16. Liya Ann Sunny , Sara Susan Joseph, Sonu Sara Geogy , K. S. Sreelakshmi , Abin T.Abraham.”Image Caption Generator”.International Journal of Recent Advances in Multidisciplinary Topics Volume 2, Issue 4, April 2021.
17. Eric ke Wang,xun Zhang ,Fan Wang,Tsu-yang Wu ,and Chien-ming Chen -"Multilayer Dense Attention Model for image caption" (2019).
18. Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2018. A Comprehensive Survey of Deep Learning for Image Captioning. ACM Comput. Surv. 0, 0, Article 0 (October 2018), 36 pages. Computing methodologies→Machine learning; Neural networks.
19. S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, ‘‘A guide to convolutional neural networks for computer vision,’’ Synth. Lectures Comput. Vis., vol. 8, no. 1, pp. 1–207, Feb. 2018
20. K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.
21. Md.zakir hossain, Ferdoussohel,Mohd fairuz shiratuddin,Hamid laga,Mohammed bennamoun- “Text to Image Synthesis for Improved Image Captioning” IEEE Access(2021),Digital Object Identifier 10.1109/ACCESS.2021.3075579.
22. Soheyla Amirian et.al.- “Automatic Image and Video Caption Generation With Deep Learning: A Concise Review and Algorithmic Overlap” IEEE Access(December 4, 2020)Digital Object Identifier 10.1109/ACCESS.2020.3042484.
23. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov,R. Zemel, Y. Bengio. “Show, attend and tell: Neural image caption generation with visual attention” In International conference on machine learning, pp. 2048-2057, 2015.
24. M. Tanti, A. Gatt and K.P. Camilleri. "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator" arXiv preprint arXiv:1708.02043, 2017.
25. N. Kumar, D. Vigneswari, A. Mohan, K. Laxman, J. Yuvaraj. “Detection and Recognition of Objects in Image Caption Generator System: A Deep Learning Approach”, In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 107-109, 2019
26. Shobia L,Pradheesa R, Prof.Kala.Image Captioning using deep learning methodsInternational Journal of Research in Social Science and Humanities.Volume 9.Issue 2021
27. JianHui ,Chen,Wen Qiang Dong,Minchen Li.Image Caption generator using deep learning approach,International conference on deep learning,2019.
28. N. Kumar, D. Vigneswari, A. Mohan, K. Laxman, J. Yuvaraj. “Detection and Recognition of Objects in Image Caption Generator System: A Deep Learning Approach”, In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 107-109, 2019.