CAPTIONBOT FOR ASSISTIVE VISION

Project report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

in

Computer Science and Engineering/Information Technology

by

Divyanshu Sinha (171214)

Shivam Sharma (171246)

Under the supervision of

(Dr. Amit Kumar)

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan 173234, Himachal Pradesh

shivam

by Sh Vim

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled "**CaptionBot for Assistive Visision**" in partial fulfilment of the requirements for the award of the degree of the **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of InformationTechnologyWaknaghatisanauthenticrecordofmyownworkcarried outoveraperiodfromJanuary2021toMay2021underthesupervisionof**Dr. Amit Kumar**, Assistant Professor (Senior Grade) in the department of Computer Science &Engineering.

The matter embedded in the report has not been submitted for the award of any other degree or diploma.

Divyanshu [illegible]

(StudentSignature)
DivyanshuSinha(171214)

(StudentSignature)
Shivam Sharma(171246)

This is to certify that the above statement made by the candidates is true in the best of my knowledge.

[illegible]

(Supervisor Signature) Dr.

Amit Kumar

AssistantProfessor

Computer Science & Engineering

I

ACKNOWLEDGEMENT

Besides the hard work of a group, the success of a project also depends highly on the encouragementandguidelinesofmanyothers.Wetakethisopportunitytoexpressmysincereand heartfelt gratitude to the people who have been instrumental in the successful completion of this project.

Our first and foremost acknowledgement goes to our supervisor and mentor, **Dr. Amit Kumar**, withoutwhosehelpthecompletionofthisprojectwouldn’thavebeenpossible.Itisbecauseofhis guidance and efforts that we are able to implement a practical idea based on my field of interest. We would also like to thank my panel **Mr. Prateek Thakral and Mr. Rizwan ur Rehman**, for giving me an opportunity to present my project and for judging my work and providing me feedback which would certainly help me in thefuture.

Last but not the least we would like to acknowledge my institution **Jaypee University of Information Technology** for giving me a platform to give me life and implementation, to the various fields I have studied till date.

Table of Content

Chapter No.	Topics	Page No.
	List of Figures	
	List of Abbreviations	VI
	Abstract	VII
Chapter1	Introduction	1
	1.1 Problemsinexistingsystem	3
	1.2 Objective	4
	1.3 Methodology	5
Chapter2	LiteratureSurvey	8
Chapter3	SystemDevelopment	10
	3.1 SystemArchitecture	10
	3.2 Algarithm	11
	3.3 DatasetUsed	15
	3.4 DataPre-processing	16
	3.5 Extractingthefeaturevector	18
	3.6 TokenizingtheVocabulary	18
	3.7 SentenceGeneration	19
	3.8 ModelBuilding	19
	3.9 Trainingthemodel	20
	3.10 Web Technology	20

Chapter4	PerformanceAnalysis	25
	4.1 Testingthemodel	27
	4.2 EvaluationMetric	31
	4.3 32	
	4.4 AdamOptimizer	32
	4.SQuantitativeAnalysis	32
ChapterS	Conclusion	33
	5.1 Conclusion	35
	5.2 FutureScope	36
References		37

List of Figures

Figure1.1Blindpeoplewillgreatlybenefitfromtoday'stechnology Figure 01

1.2 DFD LevelO 05

Figure1.3DFDLevel1 05

Figure1.4DFDLevel2 05

Figure 1.5 State diagram of model 06

Figure3.1ProposedModelofImageCaptionGenerator 10

Figure3.2SystemArchitectureofImageCaptionGenerator 11

Figure3.3WorkingofCNN 12

Figure 3.4 Working of LSTM 13

Figure 3.5 Image Dataset 15

Figure 3.6 Image Captions Dataset 16

Figure 3.7 Captions grouped in set of five 17

Figure 3.8 Caption dataset after cleaning. 17

Figure3.9Codesnippetforfeatureextraction 18

Figure3.10CodesnippetforCNN-RNNmodel. 19

Figure 3.11 Modeltraining. 19

Figure 3.12 – Figure 3.18 Website Snipet 21-24

Figure 4.1 Final view of our model Figure 25

4.2 Code snippet for testing Figure 4.3 27

Figure 4.9 Result Snipets 27-30

List of Abbreviations

BLEU	Bilingual Evaluation Understudy
CIDER	Consensus-based Image DescriptionEvaluation
CNN	Convolutional Neural Network
DFD	Data Flow Diagram
GLOVE	Global Vectors for WordRepresentation
	a
LSTM	Long Short Term Memory
METEOR	MetricforEvaluationofTranslationwithExplicitOrder
MSCOCO	Common Objects inContext
RNN	Recurrent Neural Network
SEO	Search Engine Optimization
SGD	Stochastic GradientDescent

Abstract

Having the option to naturally portray the substance of a picture utilizing appropri ately shaped English sentences is a difficult assignment, yet it could have incredible effect by helping outwardly disabled individuals better comprehend their environmental factors. Most current cell phones can catch photos, making it feasible for the out wardly hindered to make pictures of their surroundings. These pictures can at that point be utilized to produce subtitles that can be recited for all to hear to the outwardly debilitated, so they can improve feeling of what's going on around them. In this project, we present a profound repetitive design that naturally creates brief clarifications of pictures. Our rnrxlels utilize a convolutional neural network (CNN) to extricate highlights from a picture. These highlights are then taken care of into a vanilla recurrent neural organization (RNN) or a Long Short-Term Memory (LSTM) organization to produce a depiction of the picture in legitimate English. Our inrxlels accomplish practically identical to best-in-class execution, and create exceptionally unmistakable subtitles that can conceivably significantly improve the lives of visually impairedindividuals.

Chapter 1:Introduction

Visual impedance, otherwise called vision weakness or vision misfortune, is a diminished capacity to see to some extent that causes issues not fixable by regular methods, for example, glasses. As per the World Health Organization, 285 million individuals are outwardly weakened around the world, including more than 39 million visually impaired individuals [1]. Living with visual disability can be trying, since some everyday life circumstances are hard to comprehend without great visual sharpness.

Innovation can possibly altogether improve the lives of outwardly disabled individuals (Figure 1.1). Access innovation, for example, screen peruses, screen magnifiers, and refreshable Braille shows empower the incognizant in regards to utilize standard PC applications and cell phones giving them admittance to beforehand distant data. Another such innovation that could improve the lives of the outwardly disabled is picture subtitle age. Most present-day cell phones can catch photos, making it workable for the outwardly impeded to make pictures of their environmental factors. These pictures can be utilized to produce inscriptions that can be recited for all to hear to give outwardly impeded individuals a superior comprehension of their environmental factors. Picture subtitle age can likewise make the web more available to outwardly disabled individuals. The most recent decade has seen the victory of the rich graphical work area, loaded with beautiful symbols, controls, fastens, and pictures. Robotized inscription age of online pictures can make the web an all the more welcoming spot for outwardly impededsurfers.

Figure 1.1 Blind people will greatly benefit from today’s technology

Having the option to consequently portray the substance of a picture utilizing appropriately shaped English sentences is an extremely testing task. This undertaking is fundamentally harder, for instance, than the all-around contemplated picture grouping or article acknowledgment assignments, which have been a principle center in the PC vision network. Undoubtedly, a depiction must catch the articles contained in a picture, yet it additionally should communicate how these items identify with one another just as their properties and the exercises they are engaged with. In addition, the above semantic information must be communicated in a common language like English, which implies that a language model is required notwithstanding visual arrangement.

For this objective of picture subtitling, in light of semantics of pictures should be caught here and communicated in the ideal type of regular dialects. It has an incredible effect in reality, for example by helping outwardly disabled individuals better comprehend the substance of pictures on theweb.

1.1 *Problems intheexisting system*

Most work in visual acknowledgment has initially centered around picture characterization, for example allotting marks relating to a fixed number of classifications to pictures. Incredible advancement in picture characterization has been made in the course of the most recent few years, particularly with the utilization of profound learning strategies [2, 3]. By and by, a class name actually gives restricted data about a picture, and particularly outwardly hindered individuals can profit by more nitty gritty portrayals. Some underlying endeavors at creating more point by point picture portrayals have been made, for example by Farhadi et al. furthermore, Kulkarni et al. [4, 5], however these models are commonly subject to hard- coded sentences and visual ideas. Likewise, the objective of the vast majority of these works is to precisely portray the substance of a picture in a solitary sentence. Be that as it may, this one sentence necessity superfluously restricts the nature of the portrayals produced by the model. A few works, for instance by Li et al., Gould et al., and Fidler et al., zeroed in on acquiring a comprehensive comprehension of scenes and articles portrayed on pictures [6, 7, 8, 9]. In any case, the objective of these works was to effectively allocate marks comparing to a fixed number of classes to the scene sort of a picture, rather than creating more elevated level clarifications of the scenes and items portrayed on apicture.

Producing sentences that portray the substance of pictures has just been investigated. A few works endeavor to tackle this errand by finding the picture in the preparation set that is generally like the test picture and afterward restoring the inscription related with the test picture [4, 10, 11, 12, 13]. Jia et al., Kuznetsova et al., and Li et al. locate numerous comparable pictures, and consolidate their subtitles to produce the subsequent inscription [14, 15, 16]. Kuznetsova et al., and Gupta et al. taken a stab at utilizing a fixed sentence format in mix with object discovery and highlight learning [5]. They attempted to distinguish items and highlights contained in the picture, and dependent on the recognized articles contained in the picture they utilized their sentence layout to make sentences portraying the picture. In any case, this methodology significantly restricts the yield assortment of the model.

As of late there has been a resurgence of premium in picture inscription age, because of the most recent improvements in profound learning [2]. A few profound learning approaches have been created for producing more significant level word portrayals of pictures. Convolutional Neural Networks have been demonstrated to be amazing models for picture arrangement and article identification undertakings. What's more, new models to acquire low-dimensional-vector portrayals of words, for example, word2vec, and GloVe (Global Vectors for Word Representation) and Recurrent Neural Networks can together make models that consolidate picture highlights with language displaying to produce picture portrayals. Karpathy et al. built up a Multimodal Recurrent Neural Network engineering that utilizations gathered arrangements to figure out how to produce novel depictions of picture locales. Likewise, Kiros et al. utilized a log-bilinear model that creates full sentence portrayals for pictures.Bethatasitmay,their modelusesafixed windowsetting.

1.2 *Objective*

Our picture subtitle generator model, we will be blending CNN-RNN structures. Highlight extraction from pictures is finished utilizing CNN. We have utilized the pre-prepared model Exception. The data got from CNN is then utilized by LSTM [17] for creating a depiction of the picture.

Nonetheless, sentences that are produced utilizing these methodologies are normally conventional portrayals of the visual substance and foundation data is overlooked. Such conventional portrayals don't fulfill in new circumstances as they, basically repeat the data present in the pictures and nitty gritty depictions [18] with respect to occasions and elements present in the pictures are not given, which is basic to understanding emanant circumstances.

Consequently, portraying the substance of a picture is an essential issue in man-made brainpower that interfaces PC vision and regular language preparing. Prior techniques initially produce explanations (i.e., things and descriptors) from pictures (Sermanetet al., 2013; Russakovsky et al., 2015), at that point create a sentence from the comments (Gupta and Mannem).

The target of our task is to build up an online interface for clients to get the [19] depiction of the picture and to make an arrangement framework to separate pictures according to their portrayal. It can likewise make the errand of SEO simpler which is confounded as they need to keep up and investigate colossal measures of information.

1.3 *Methodology*

We have made Data flow diagrams (DFD) for better understanding of our model. It provides overview of the System or process being analyzed.

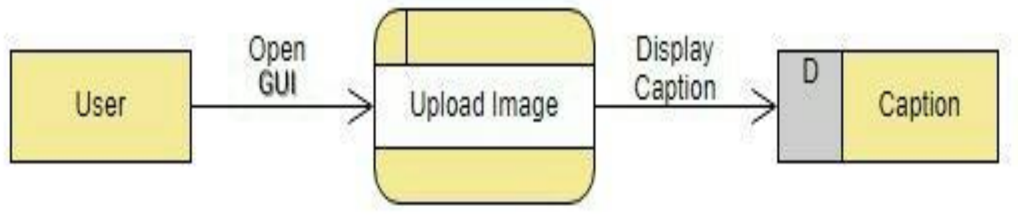


Figure 1.2 DFD Level0

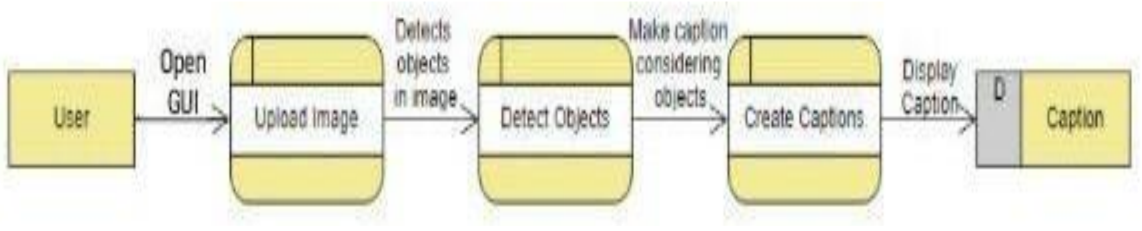


Figure 1.3 DFD Level1

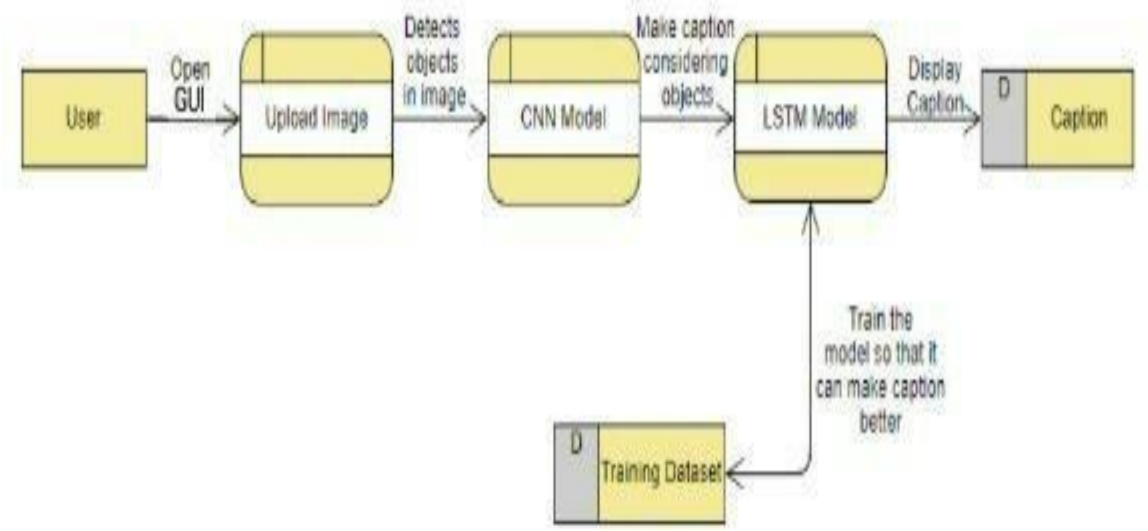


Figure 1.4 DFD Level2

We have also made the state diagram which is shown in Fig 1.5. First client will peruse the site. At that point he will transfer the picture, CNN will recognize the articles present in the picture then LSTM will begin planning subtitles considering the items present in the picture utilizing [20] Training Dataset, which includes Image Data set and Text Data Set, after the preparation a reasonable inscription will be created and shown top the client.

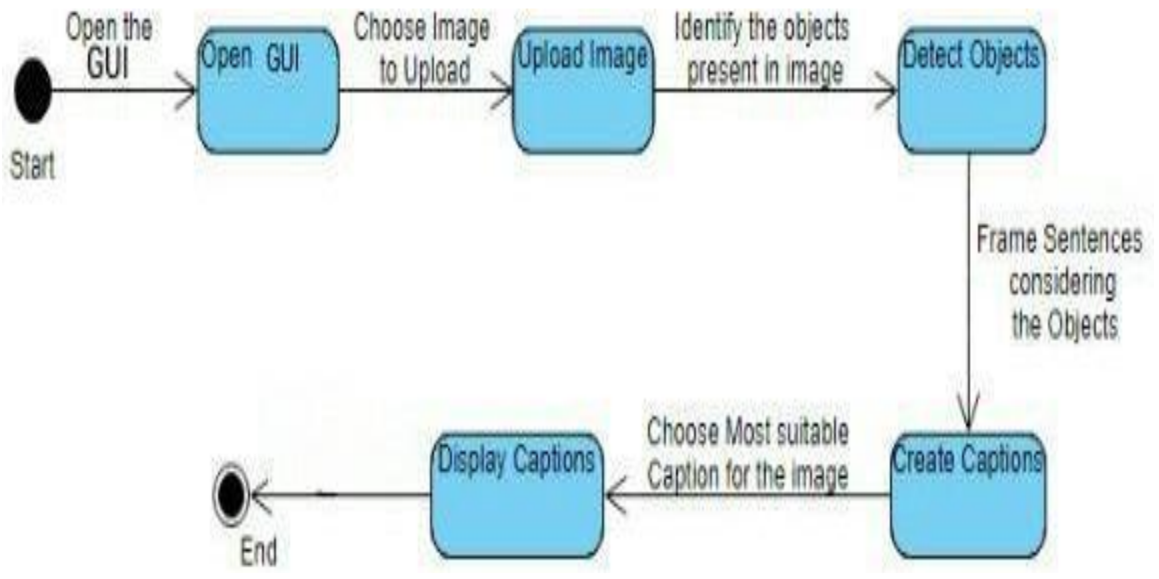

Open the GUI
Start
Open GUI
Choose Image to Upload
Upload Image
Identify the objects present in image
Detect Objects
Frame Sentences considering the Objects
Create Captions
Choose Most suitable Caption for the image
Display Captions
End

Figure 1.5 State diagram of model

The proposed game plan of Image Caption Generator has the capacities to Generate Captions for the Images, given during the Training reason and for the New pictures as well. Our Model acknowledges an Image as Input and by inspecting the image it distinguishes objects present in an image and make an engraving which depicts the image okay for any machine to appreciate what an image is endeavoring to state.

In this venture, we propose a neural and probabilistic system to produce portrayals from pictures. Late advances in factual machine interpretation have shown that, given an amazing succession model, it is feasible to accomplish best in class results by straightforwardly boosting the likelihood of the right interpretation given an information sentence in an "start to finish" design – both for preparing and induction. These models utilize a repetitive neural organization which encodes the variable length contribution to a fixed dimensional vector, and utilizations this portrayal to "interpret" it to the ideal yield sentence. In this manner, it is normal to utilize a similar methodology where, given a picture (rather than an info sentence in the source language), one applies a similar guideline of "making an interpretation of" it into its depiction.

Thus, we propose to directly maximize the probability of the correct description given the image by using the following formulation:

$$\theta^\star = \arg\max_\theta \sum_{(I,S)} \log p(S|I;\theta)$$

where θ are the parameters of our model, I is an image, and S its correct transcription. Since S represents any sentence, its length is unbounded. Thus, it is common to apply the chain rule to model the joint probability over S_0;…….; S_N, where N is the length of this particular example as

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1})$$

where we dropped the dependency on for convenience. At training time, (S; I) is a training example pair, and we optimize the sum of the log probabilities as described in (2) over the whole training set using stochastic gradient descent (further training details are given in Section 4). It is natural to model p(S_{tj}I; S_0; : : : ; S_t1) with a Recurrent Neural Network (RNN), where the variable number of words we condition upon up to t 1 is expressed by a
fixed length hidden state or memory ht. This memory is updated after seeing a new input xt by using a non-linear function f.

To make the above RNN more concrete two crucial design choices are to be made: what is the exact form of f and how are the images and words fed as inputs xt. For f we use a Long-Short Term Memory (LSTM) net, which has shown state-of-the art performance on sequence tasks such as translation. This model is outlined in the next section. For the representation of images, we use a Convolutional Neural Network (CNN). They have been widely used andstudied for image tasks, and are currently state-of-the art for object recognition and detection. Our particular choice of CNN uses a novel approach to batch normalization and yields the current best performance on the ILSVRC 2014
classification competition [12]. Furthermore, they have been shown to generalize to other tasks such as scene classification by means of transfer learning [4]. The words are represented with an embedding model.

Chapter 2: Literature Survey

In Literature audit, different references of the current tasks are contemplated which are like this current undertaking.

1. [22] In this paper one of the most mainstream profound neural organizations is the Convolutional Neural Network (CNN) is clarified. There are different layers in CNN, for example, convolutional layer, and non-linearity layer, and pooling layer and completely associated layer too. The CNN has an astounding exhibition in AI issues and one of the most widely recognizedcalculations.
2. [23] In this paper Sepp Hochreiter clarify about the profound neural organization calculation long short group Memory (LSTM). LSTM is neighborhood in both spaces just as expected; the computational intricacy is per season of step and furthermore the weight design portrayal. In contrast with other calculation LSTM prompts a lot more effective runs, and learn a lot quicker. It's even comprehended unpredictable, fake long delay undertakings that have never been fathomed by past intermittent organization.
3. [24] The crucial issue in man-made reasoning that associates PC vision and Natural language preparing is consequently depicting the substance of a picture. Inthis paper, A.L efficiently dissect a profound neural organizations-based picture inscription age technique. Here a picture is given as the info, and the strategy as yield as sentence in English depicting the substance of the picture. They break down three parts of the technique: convolutional neural network (CNN), Recurrent neural network (RNN) and sentence age. This model dissect picture and produce more trivial and applicable words forpictures.
4. [25] Current image captioning approaches generate descriptions which lack specific information, such as named entities that are involved in the images. Here Di Lu, Spencer Whitehead had proposed a very new task which generates descriptive image captions, given images as input. A simple solution to this problem that we are proposing is that we will train a CNN-LSTM model so that it can generate a caption based on theimage.
5. [26] Automatically describing the content of an image using properly arranged English sentences is a tough challenging task, but it could be something very necessaryforhelping visuallyimpairedpeople.Modernsmartphonesareabletotake

the photographs, which can help in taking surrounding images for visually impaired peoples. Here images as input can generate captions that can be loud enough so that visually impaired can hear, so that they can get a better sense of things present in there surrounding. Here Christoper Elamri uses a CNN model to extract features of an image. These features are then fed into a RNN or a LSTM model to generate a description of the image in grammatically correct English sentences describing the surroundings.

6. Being able to automatically describe the content of an image using properly formed English sentences is a very challenging task, but it could have great impact, for instance by helping visually impaired people better understand the content of images on the web. This task is significantly harder, for example, than the well-studied image classification or object recognition tasks, which have been a main focus in the computer vision community [27]. Indeed, a description must capture not only the objects contained in an image, but it also must express how these objects relate to each other as well as their attributes and the activities they are involved in. Moreover, the above semantic knowledge has to be expressed in a natural language like English, which means that a language model is needed in addition to visual understanding. Most previous attempts have proposed to stitch together tasks [28]. Hence, it is natural to use a CNN as an image "encoder", by first pre-training it for an image classification task and using the last hidden layer as an input to the RNN decoder that generates sentences (see Fig. 1). We call this model the Neural Image Caption, or NIC. Our contributions are as follows. First, we present an end-to-end system for the problem. It is a neural net which is fully trainable using stochastic gradient descent. Second, our model combines state-of-art sub-networks for vision and language models. These can be pre-trained on larger corpora and thus can take advantage of additional data. Finally, it yields significantly better performance compared to state-of-the-art approaches.

Chapter 3: System Development

3.1 *System Architecture*

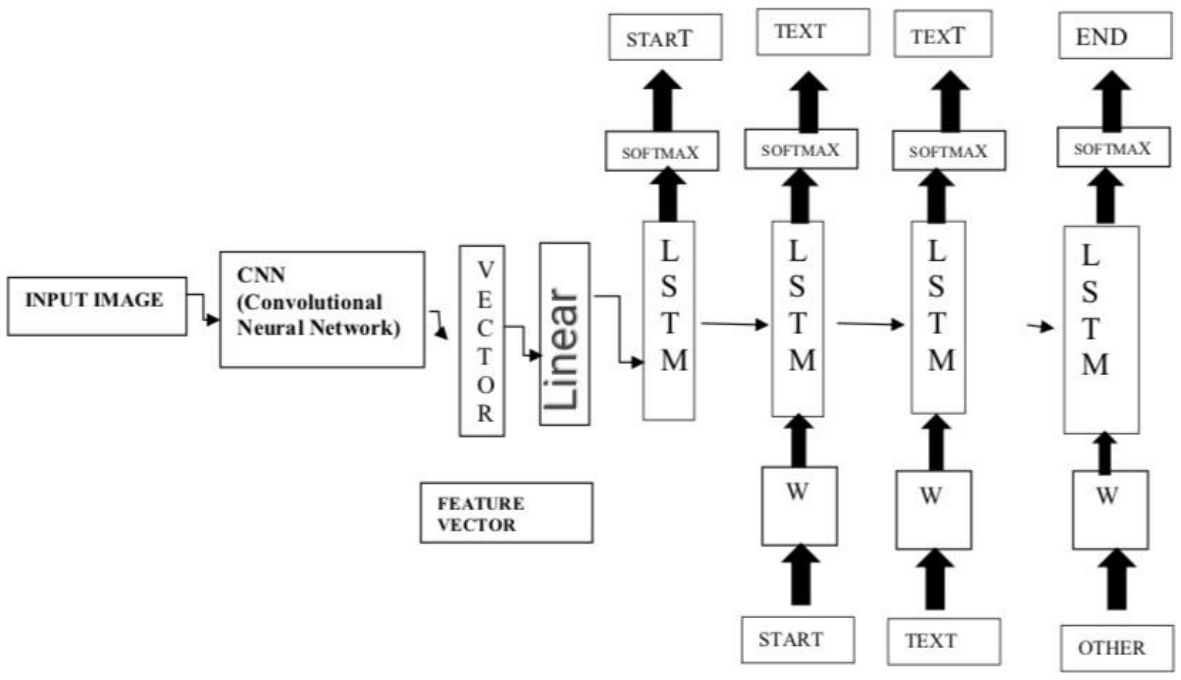


Figure 3.1 Proposed Model of Image Caption Generator

The proposed model of Image Caption Generator is as shown in the above figure 3.1. Here in this model, input picture is given and then A convolutional neural network is utilized to make a thick component vector as appeared in figure. This thick vector, likewise called an implanting, this vector can be utilized as contribution to different calculations, and it creates
[11] appropriate inscription for given picture as yield.

For a picture subtitle generator, this installing turns into a portrayal of the picture and utilized as the underlying condition of the LSTM for creating meaningfull inscriptions, for the picture.

This is our proposed system architecture will look like:

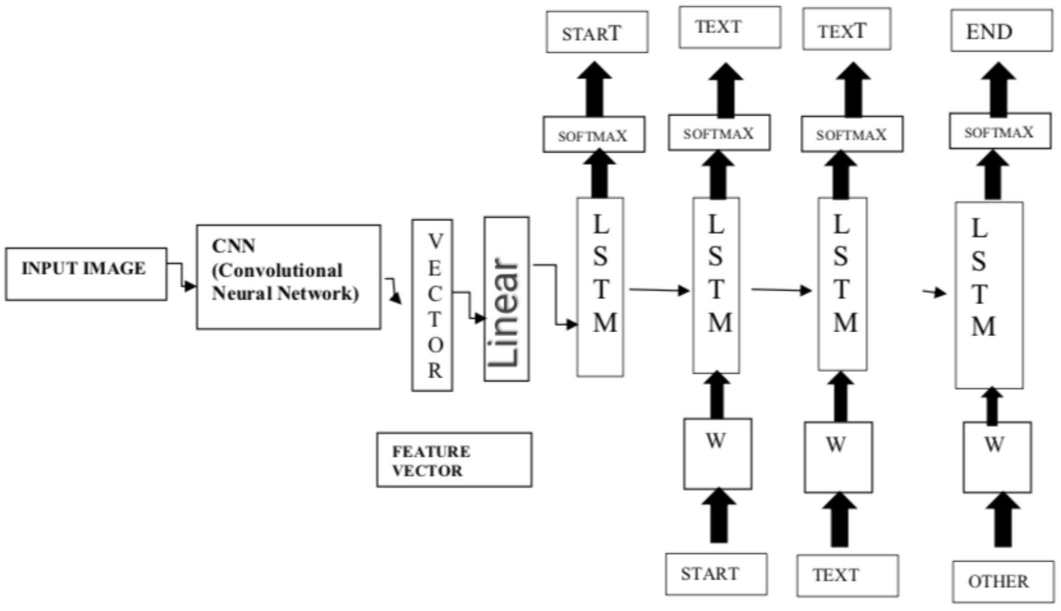


Figure 3.2 System Architecture of Image Caption Generator

System Requirements:

- OS: Windows 7 and above, Recommended: Windows 10. CPU:
- Intel processor with 64-bit support
- Storage: Data is stored on google drive.

For Execution and writing code Google colab is used.

3.2 *Algorithms*

We actualized a deep recurrent architecture that naturally creates short portrayals of pictures. We use Xception model, which was pretrained on ImageNet, to acquire pictures features. We at that point feed these features into either a vanilla RNN or a LSTM network to create a depiction of the picture in legitimateEnglish.

3.2.1 Convolutional NeuralNetwork

Convolutional Neural Network are specific profound neural network which can deal with the information that has input shape like a 2D framework. Pictures are effortlessly spoken to as a 2D lattice and CNN is valuable in working with pictures. It is basically used for image classifications and identifying if it is a bird, a plane or Superman, etc.

CNN scans images from left to right and top to bottom to extract important features from the image and combines all the feature for image classification. It automatically takes care of all the images that has been rotated, maximize, minimize, blurred, translated and done with any changes/ modification.

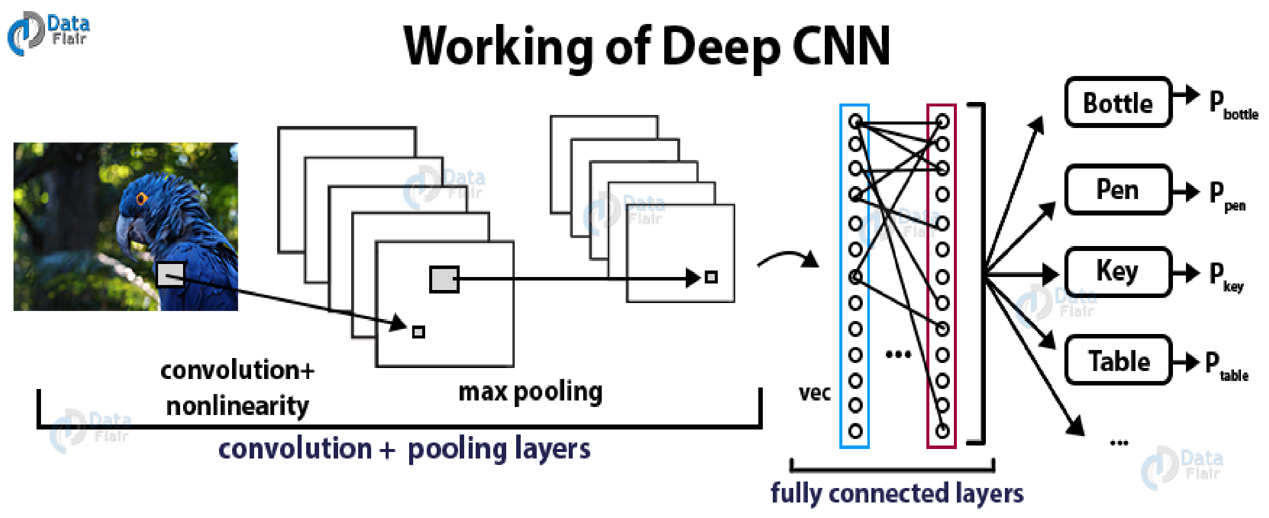


Figure 3.3 Working of CNN

3.2.2 Long Short Term Memory

LSTM represents Long Short Term Memory, they are a kind of RNN (Recurrent neural network) which is appropriate for grouping forecast issues. In view of the past content, we can anticipate what the following word will be. It has substantiated itself powerful from the conventional RNN by defeating the impediments of RNN which had momentary memory. LSTM can do significant data all through the handling of information sources and with a fail to remember door, it disposes of non-applicable data.

Long Short Term Memory organizations – normally called "LSTMs" – are an exceptional sort of RNN, equipped for learning long haul conditions. They were presented by Hochreiter and Schmidhuber (1997), and were refined and advocated by numerous individuals in after work.1 They function admirably on an enormous assortment of issues, and are currently broadly utilized.

LSTMs are expressly intended to keep away from the drawn out reliance issue. Recalling data for significant stretches of time is essentially their default conduct, not something they battle to learn!

All intermittent neural organizations have the type of a chain of rehashing modules of neural organization. In standard RNNs, this rehashing module will have an extremely basic design, for example, a solitary tanh layer.

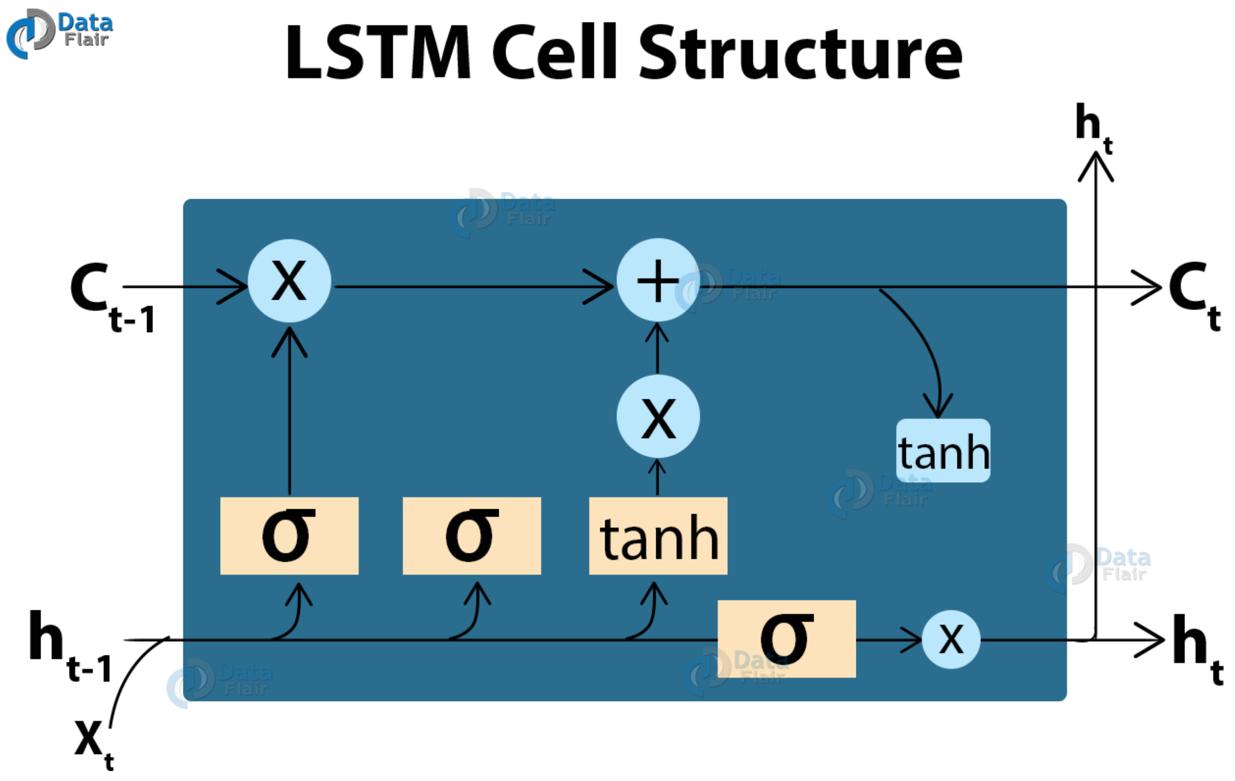
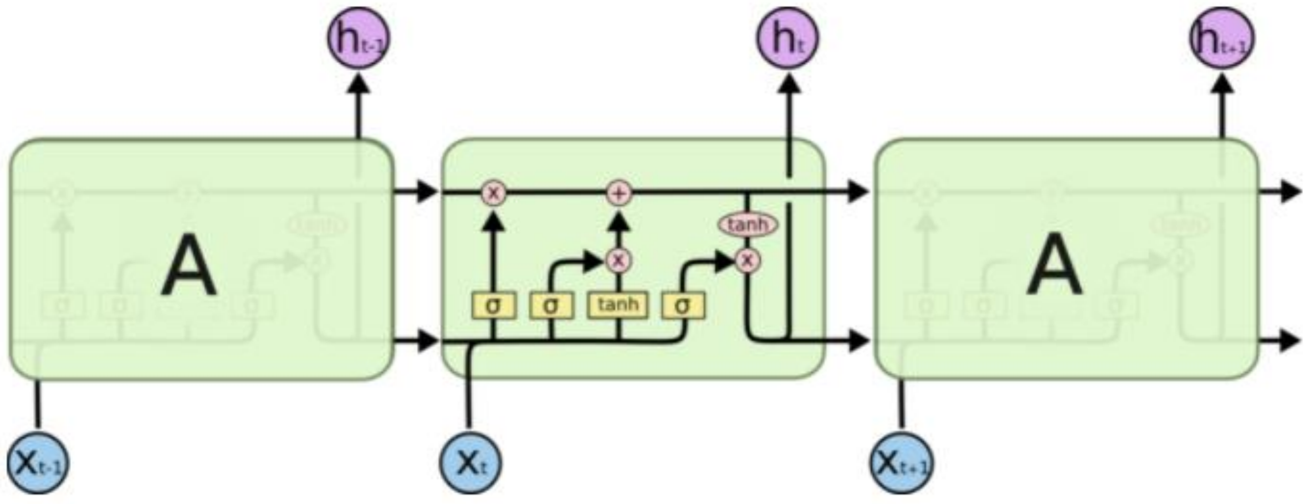


Figure 3.4 Working of LSTM



The repeating module in an LSTM contains four interacting layers.

In the above graph, each line conveys a whole vector, from the yield of one hub to the contributions of others. The pink circles address pointwise activities, similar to vector expansion, while the yellow boxes are learned neural organization layers. Lines blending indicate connection, while a line forking mean its substance being replicated and the duplicates going to various areas.

The way to LSTMs is the cell express, the level line going through the highest point of the graph. The cell state is similar to a transport line. It runs straight down the whole chain, with just some minor direct collaborations. It's exceptionally simple for data to simply stream along it unaltered.

The LSTM can eliminate or add data to the phone state, painstakingly controlled by structures called doors. Entryways are an approach to alternatively let data through. They are made out of a

sigmoid neural net layer and a pointwise duplication activity.
The sigmoid layer yields numbers somewhere in the range of nothing and one, depicting the amount of every part ought to be let through. A worth of zero signifies "let nothing through," while a worth of one signifies "let everything through!". A LSTM has three of these entryways, to secure and control the cell state.

3.2.3 GTTS(Google Text-to-Speech)

The Text-to-Speech API enables developers to generate human-like speech. The API converts text into audio formats such as WAV, MP3, or Ogg Opus. It also supports Speech Synthesis Markup Language (SSML) inputs to specify pauses, numbers, date and time formatting, and other pronunciation instructions.

Google CloudText-to-Speech empowers designers to integrate common sounding discourse with 30 voices, accessible in numerous dialects and variations. It applies DeepMind's momentous exploration in WaveNet and Google's incredible neural organizations to convey high loyalty sound.

Google TTS simply checks the content information and matches it with its data set and essentially plays a sound yield. In the event that something isn't found in the data set it attempts to talk, for example at the point when some Indian attempts to talk familiar French (simply a model). Actually speaking, If you have seen a word reference, every single word is appointed an elocution. Likewise TTS checks its information base for the articulation and arranges the discourse yield. Furthermore, again assuming no such word exists in is word reference, it will articulate straight that is, Raam will be articulated as in the event that you would "Ram" (Hindi).

3.3 *DatasetUsed*

For evaluation we use a number of datasets which consist of images and sentences in English describing these images.

For the image caption generator, we have used the Flickr_8K dataset. There are also other [6] big datasets like Flickr_30K and MSCOCO dataset but it can take weeks for systems having only CPU support just to train the network, so we used a small Flickr8k dataset. Using a huge dataset helps in developing a better model.

- **Flicker8k_Dataset –** Dataset folder which contains 8091 images.

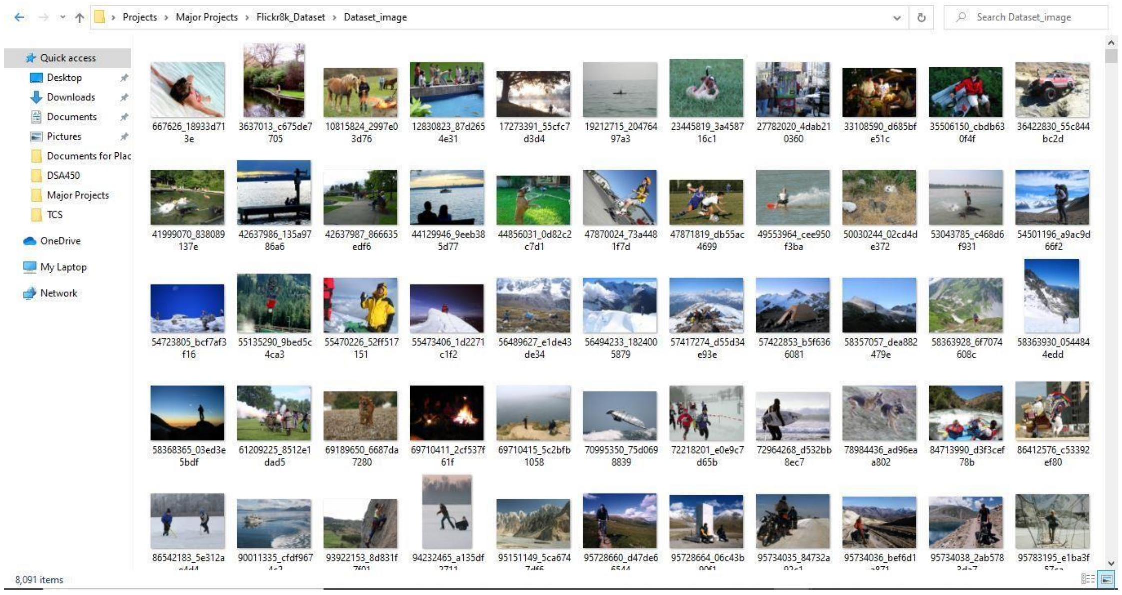
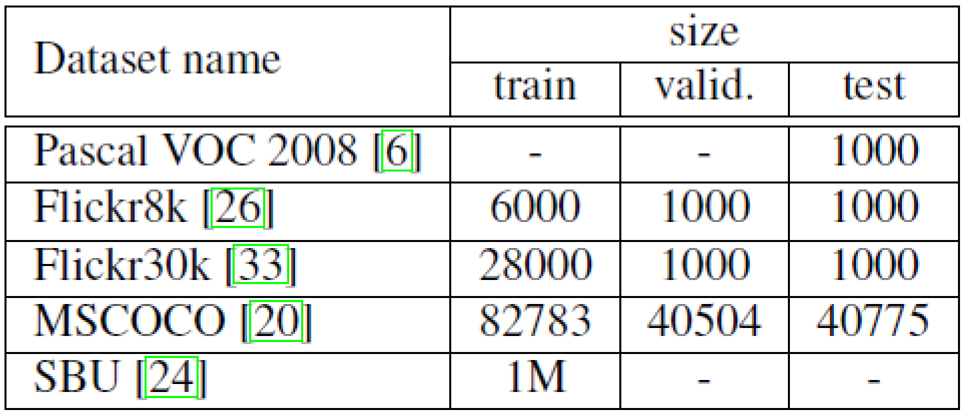


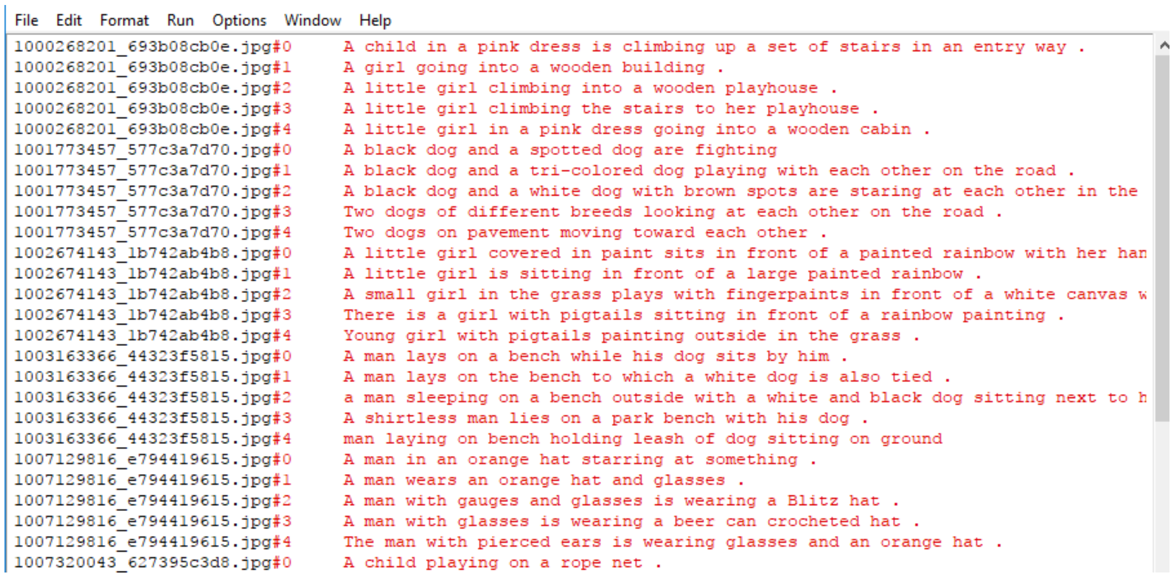
Figure 3.5 Image Dataset



Dataset name	size		
	train	valid.	test
Pascal VOC 2008 [6]	-	-	1000
Flickr8k [26]	6000	1000	1000
Flickr30k [33]	28000	1000	1000
MSCOCO [20]	82783	40504	40775
SBU [24]	1M	-	-

Figure Data Set Available

- **Flickr_8k_text** – Dataset folder which contains text files and captions ofimages



```
1000268201_693b08cb0e.jpg#0	A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg#1	A girl going into a wooden building .
1000268201_693b08cb0e.jpg#2	A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg#3	A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg#4	A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg#0	A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg#1	A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg#2	A black dog and a white dog with brown spots are staring at each other in the
1001773457_577c3a7d70.jpg#3	Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg#4	Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg#0	A little girl covered in paint sits in front of a painted rainbow with her han
1002674143_1b742ab4b8.jpg#1	A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg#2	A small girl in the grass plays with fingerpaints in front of a white canvas w
1002674143_1b742ab4b8.jpg#3	There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg#4	Young girl with pigtails painting outside in the grass .
1003163366_44323f5815.jpg#0	A man lays on a bench while his dog sits by him .
1003163366_44323f5815.jpg#1	A man lays on the bench to which a white dog is also tied .
1003163366_44323f5815.jpg#2	a man sleeping on a bench outside with a white and black dog sitting next to h
1003163366_44323f5815.jpg#3	A shirtless man lies on a park bench with his dog .
1003163366_44323f5815.jpg#4	man laying on bench holding leash of dog sitting on ground
1007129816_e794419615.jpg#0	A man in an orange hat starring at something .
1007129816_e794419615.jpg#1	A man wears an orange hat and glasses .
1007129816_e794419615.jpg#2	A man with gauges and glasses is wearing a Blitz hat .
1007129816_e794419615.jpg#3	A man with glasses is wearing a beer can crocheted hat .
1007129816_e794419615.jpg#4	The man with pierced ears is wearing glasses and an orange hat .
1007320043_627395c3d8.jpg#0	A child playing on a rope net .
```

Figure 3.6 Image Captions Dataset

3.4 *DataPre-processing*

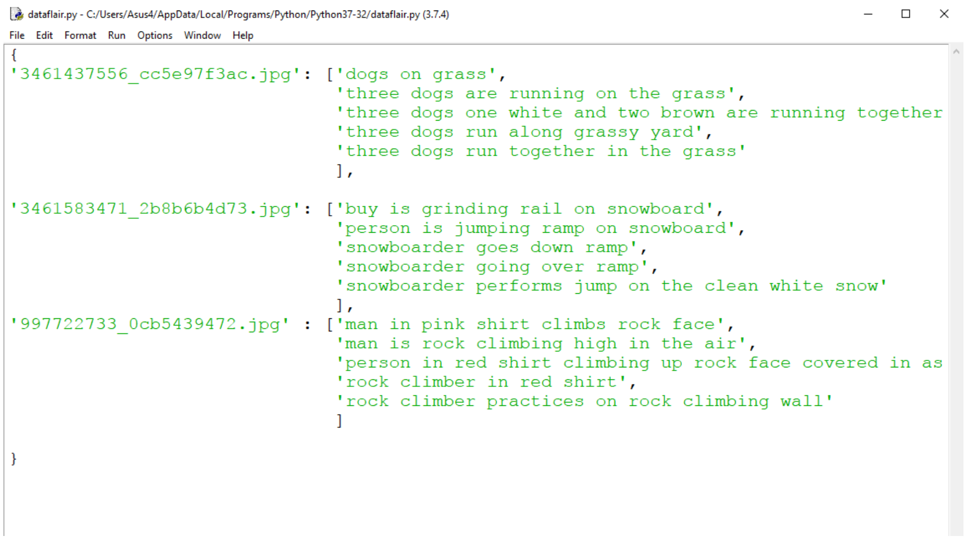
After collecting the dataset, new step is to clean the data. Data pre-processing is very important step in model making. Clean and pre-processed data helps a lot while building of model and giving expected output as desired.

In data pre-processing part, I have cleaned the caption dataset file by performing various operations which is explained below.

So, for data clean I have defined five functions which is described as follows:

- **load_doc(filename)** – It is used for loading the caption dataset file and reading the contents inside the file into astring.

- **all_img_captions(filename)** – This function will make a depictions word reference that guides pictures with a rundown of 5 subtitles. The portrayals word reference will look something likethis:


dataflair.py - C:/Users/Asus4/AppData/Local/Programs/Python/Python37-32/dataflair.py (3.7.4)
File Edit Format Run Options Window Help

```
{
'3461437556_cc5e97f3ac.jpg': ['dogs on grass',
                              'three dogs are running on the grass',
                              'three dogs one white and two brown are running together
                              'three dogs run along grassy yard',
                              'three dogs run together in the grass'
                              ],

'3461583471_2b8b6b4d73.jpg': ['buy is grinding rail on snowboard',
                              'person is jumping ramp on snowboard',
                              'snowboarder goes down ramp',
                              'snowboarder going over ramp',
                              'snowboarder performs jump on the clean white snow'
                              ],
'997722733_0cb5439472.jpg'  : ['man in pink shirt climbs rock face',
                              'man is rock climbing high in the air',
                              'person in red shirt climbing up rock face covered in as
                              'rock climber in red shirt',
                              'rock climber practices on rock climbing wall'
                              ]

}
```

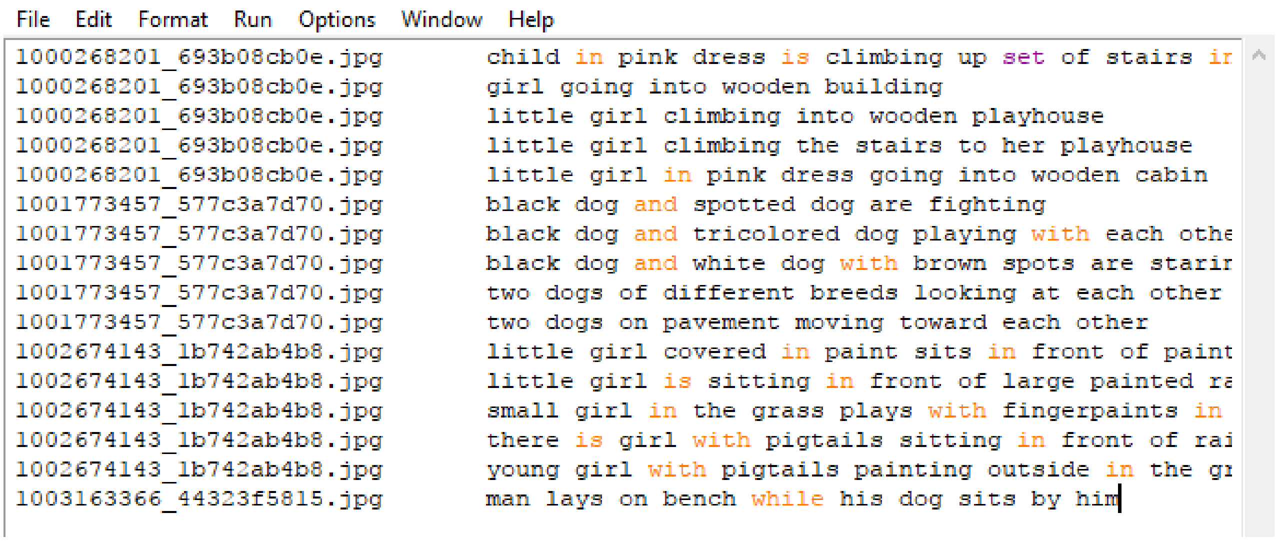
Figure 3.7 Captions grouped in set of five.

- **cleaning_text(descriptions)** – This function takes all depictions and performs information cleaning. This is a significant advance when we work with literary information, as per our objective, we choose what kind of cleaning we need to perform on the content. For our situation, we will eliminate accentuations, changing all content over to lowercase and eliminating words that contain numbers.

  In this way, an inscription like "A man riding on a three-wheeled wheelchair" will be changed into "man riding on three wheeled wheelchair**".**

- **text_vocabulary(descriptions)** – This is a basic function that will isolate all the one of a kind words and make the vocabulary from all the depictions.

- **save_descriptions(descriptions, filename)** – This function will make a rundown of the multitude of depictions that have been preprocessed and store them into a record. We will make a descriptions.txt record to store all the inscriptions. It will look something likethis:



```
1000268201_693b08cb0e.jpg      child in pink dress is climbing up set of stairs ir
1000268201_693b08cb0e.jpg      girl going into wooden building
1000268201_693b08cb0e.jpg      little girl climbing into wooden playhouse
1000268201_693b08cb0e.jpg      little girl climbing the stairs to her playhouse
1000268201_693b08cb0e.jpg      little girl in pink dress going into wooden cabin
1001773457_577c3a7d70.jpg      black dog and spotted dog are fighting
1001773457_577c3a7d70.jpg      black dog and tricolored dog playing with each othe
1001773457_577c3a7d70.jpg      black dog and white dog with brown spots are starir
1001773457_577c3a7d70.jpg      two dogs of different breeds looking at each other
1001773457_577c3a7d70.jpg      two dogs on pavement moving toward each other
1002674143_1b742ab4b8.jpg      little girl covered in paint sits in front of paint
1002674143_1b742ab4b8.jpg      little girl is sitting in front of large painted ra
1002674143_1b742ab4b8.jpg      small girl in the grass plays with fingerpaints in
1002674143_1b742ab4b8.jpg      there is girl with pigtails sitting in front of rai
1002674143_1b742ab4b8.jpg      young girl with pigtails painting outside in the gr
1003163366_44323f5815.jpg      man lays on bench while his dog sits by him
```

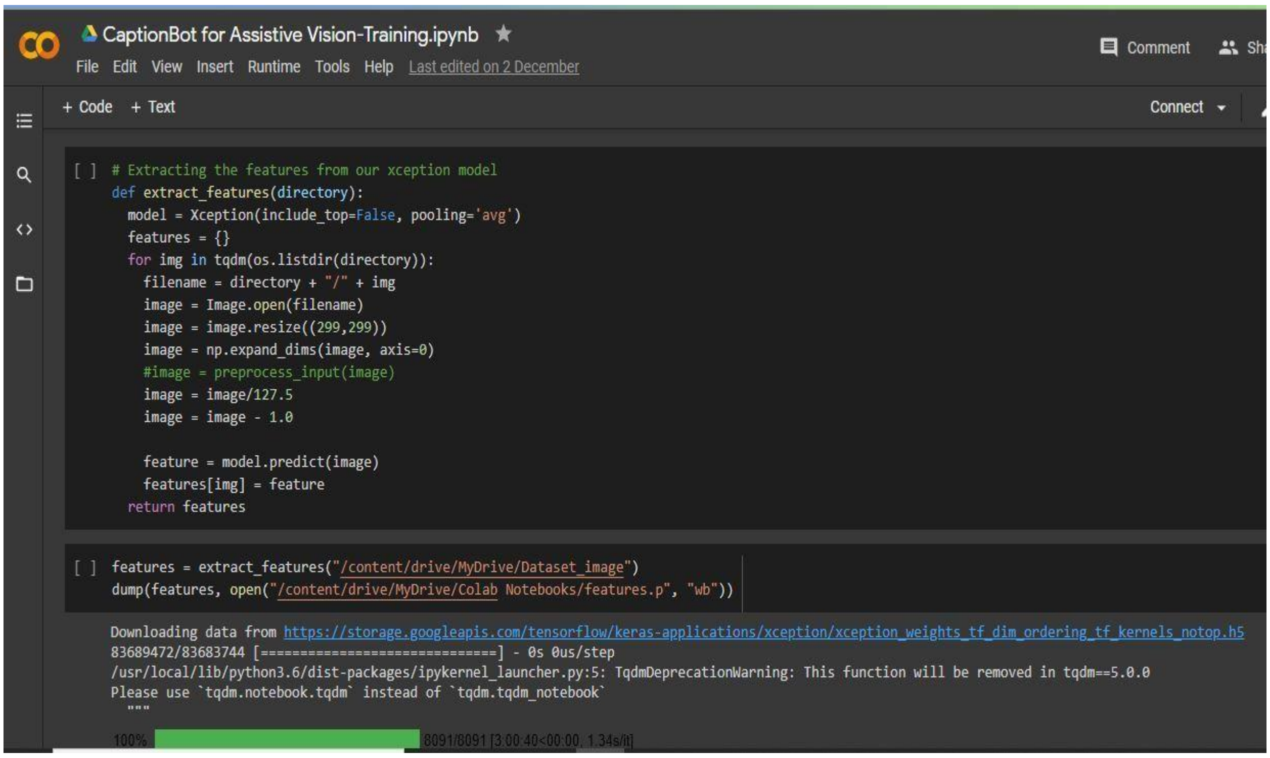
Figure 3.8 Caption dataset after cleaning.

3.5 *Extracting the feature vector*

This procedure is likewise called transfer learning, we don't need to do everything all alone, we utilize the pre-prepared model that have been now prepared on enormous datasets and concentrate the highlights from these models and use them for our errands. We are utilizing the Xception model which has been prepared on imagenet dataset that had 1000 distinct classes to group. We can straightforwardly import this model from the keras. applications . Ensure you are associated with the web as the loads get naturally downloaded. Since the Xception model was initially worked for imagenet, we will do little changes for incorporating with our model. One thing to see is that the Xception model takes 299*299*3 picture size as info. We will eliminate the last order layer and get the 2048 element vector.

model = Xception(include_top=False, pooling='avg')

The function extract_features() will extricate highlights for all pictures and we will plan picture names with their particular element exhibit. At that point we will dump the highlights word reference into a "features.p" pickle document.


CaptionBot for Assistive Vision-Training.ipynb
File Edit View Insert Runtime Tools Help Last edited on 2 December
Comment
+ Code + Text
Connect

```
[ ] # Extracting the features from our xception model
    def extract_features(directory):
      model = Xception(include_top=False, pooling='avg')
      features = {}
      for img in tqdm(os.listdir(directory)):
        filename = directory + "/" + img
        image = Image.open(filename)
        image = image.resize((299,299))
        image = np.expand_dims(image, axis=0)
        #image = preprocess_input(image)
        image = image/127.5
        image = image - 1.0

        feature = model.predict(image)
        features[img] = feature
      return features
```

```
[ ] features = extract_features("/content/drive/MyDrive/Dataset_image")
    dump(features, open("/content/drive/MyDrive/Colab Notebooks/features.p", "wb"))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [==============================] - 0s 0us/step
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  """
100% 8091/8091 [3:00:40<00:00, 1.34s/it]
```

Figure 3.9 Code snippet for feature extraction.

3.6 *Tokenizing the Vocabulary*

PCs don't comprehend English words, for PCs, we should speak to them with numbers. In this way, we will plan each expression of the vocabulary with a one of a kind file esteem. Keras library furnishes us with the tokenizer function that we will use to make tokens from our jargon and spare them to a "tokenizer.p" pickle document.

3.7 *SentenceGeneration*

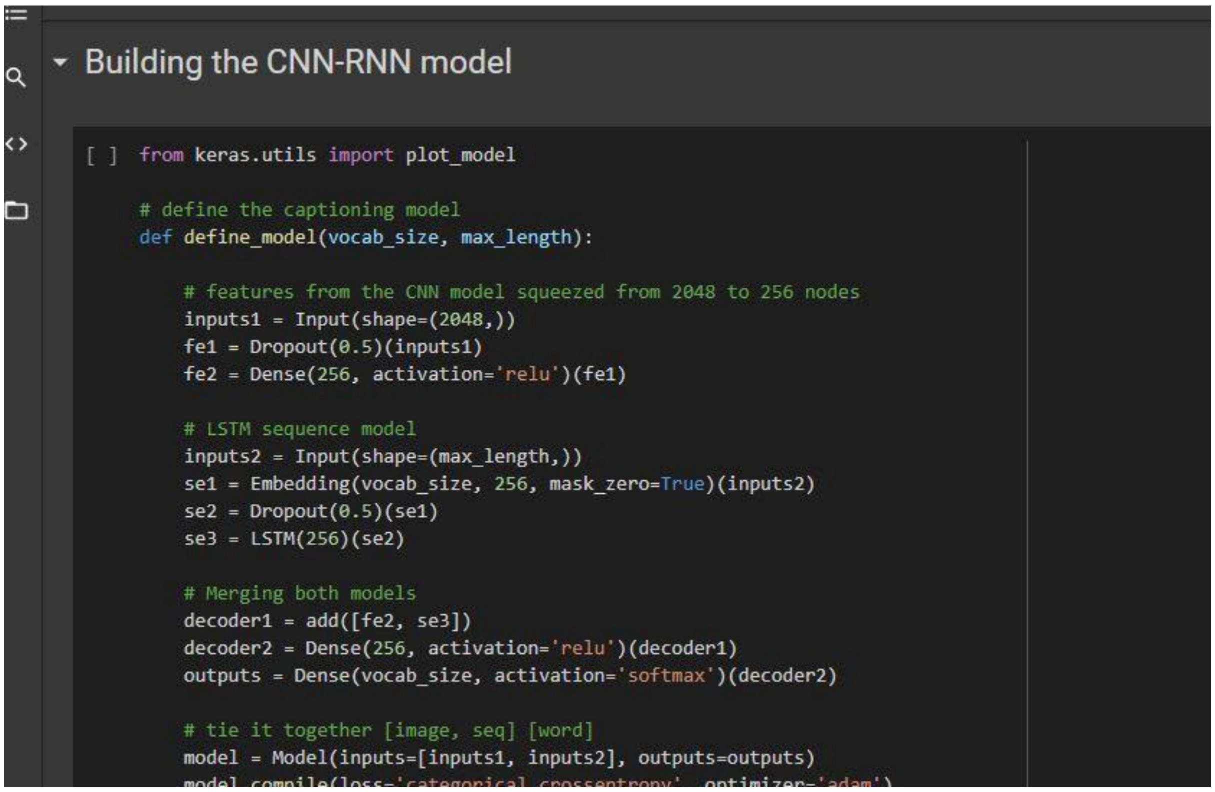
The yield of LSTM is the likelihood of each word in the jargon. Bar search is utilized to create sentences. Bar search is a heuristic pursuit calculation that investigates a diagram by extending the most encouraging hub in a restricted set. Notwithstanding pillar search, we additionally utilize k best pursuit to produce sentences.

It is fundamentally the same as the time coordinated Viterbi search. The strategy iteratively chooses the k best sentences from all the applicant sentences up to time t, and keeps just the subsequent best k of them.

3.8 *Model Building*

To characterize the structure of the model, we will utilize the Keras Model from Functional API. It will comprise of three significant parts:

- **Feature Extractor** – The element removed from the picture has a size of 2048, with a thick layer, we will decrease the measurements to 256 hubs.
- **Sequence Processor** – An implanting layer will deal with the printed input, trailed by the LSTMlayer.
- **Decoder** – By blending the yield from the over two layers, we will measure by the thick layer to make the last expectation. The last layer will contain the quantity of hubs equivalent to our jargonsize.



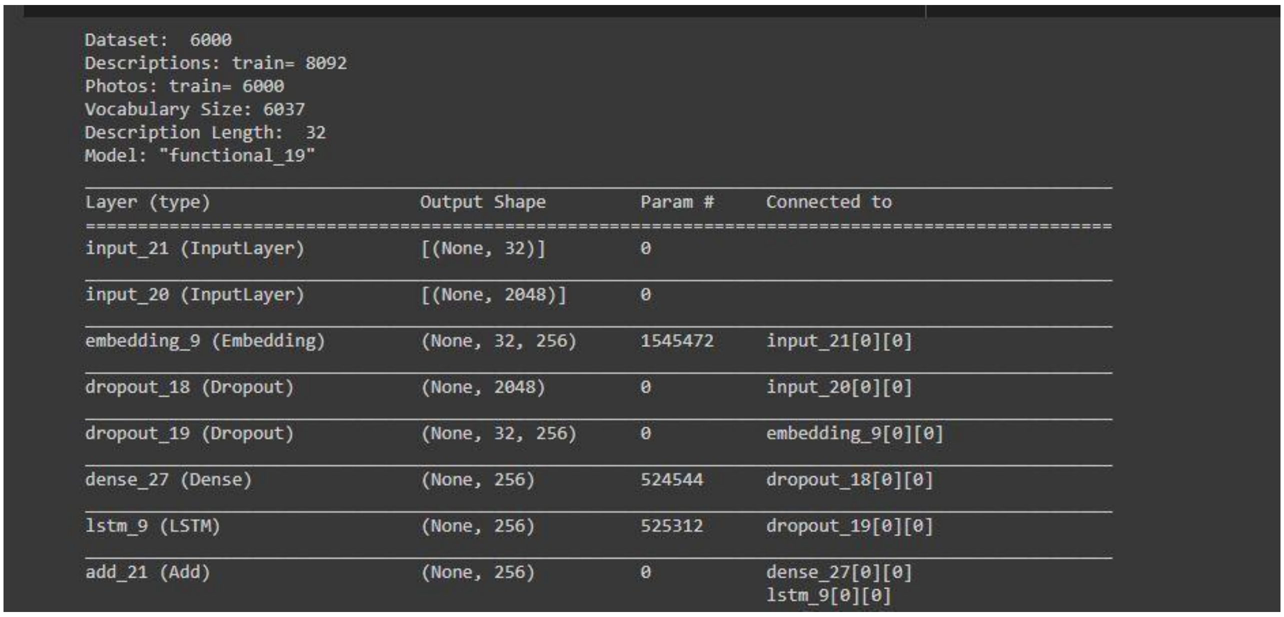
Building the CNN-RNN model

```
[ ] from keras.utils import plot_model

    # define the captioning model
    def define_model(vocab_size, max_length):

        # features from the CNN model squeezed from 2048 to 256 nodes
        inputs1 = Input(shape=(2048,))
        fe1 = Dropout(0.5)(inputs1)
        fe2 = Dense(256, activation='relu')(fe1)

        # LSTM sequence model
        inputs2 = Input(shape=(max_length,))
        se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
        se2 = Dropout(0.5)(se1)
        se3 = LSTM(256)(se2)

        # Merging both models
        decoder1 = add([fe2, se3])
        decoder2 = Dense(256, activation='relu')(decoder1)
        outputs = Dense(vocab_size, activation='softmax')(decoder2)

        # tie it together [image, seq] [word]
        model = Model(inputs=[inputs1, inputs2], outputs=outputs)
        model.compile(loss='categorical_crossentropy', optimizer='adam')
```

Figure 3.10 Code snippet for CNN-RNN model.

3.9 *Training themodel*

To train the model, we will utilize the 6000 preparing pictures by producing the information and yield successions in groups and fitting them to the model utilizing model.fit_generator() strategy. We additionally spare the model to our models envelope. This will take some time contingent upon your framework ability.



```
Dataset:  6000
Descriptions: train= 8092
Photos: train= 6000
Vocabulary Size: 6037
Description Length:  32
Model: "functional_19"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_21 (InputLayer)           [(None, 32)]         0
__________________________________________________________________________________________________
input_20 (InputLayer)           [(None, 2048)]       0
__________________________________________________________________________________________________
embedding_9 (Embedding)         (None, 32, 256)      1545472     input_21[0][0]
__________________________________________________________________________________________________
dropout_18 (Dropout)            (None, 2048)         0           input_20[0][0]
__________________________________________________________________________________________________
dropout_19 (Dropout)            (None, 32, 256)      0           embedding_9[0][0]
__________________________________________________________________________________________________
dense_27 (Dense)                (None, 256)          524544      dropout_18[0][0]
__________________________________________________________________________________________________
lstm_9 (LSTM)                   (None, 256)          525312      dropout_19[0][0]
__________________________________________________________________________________________________
add_21 (Add)                    (None, 256)          0           dense_27[0][0]
                                                                 lstm_9[0][0]
```

Figure 3.11 Model training.

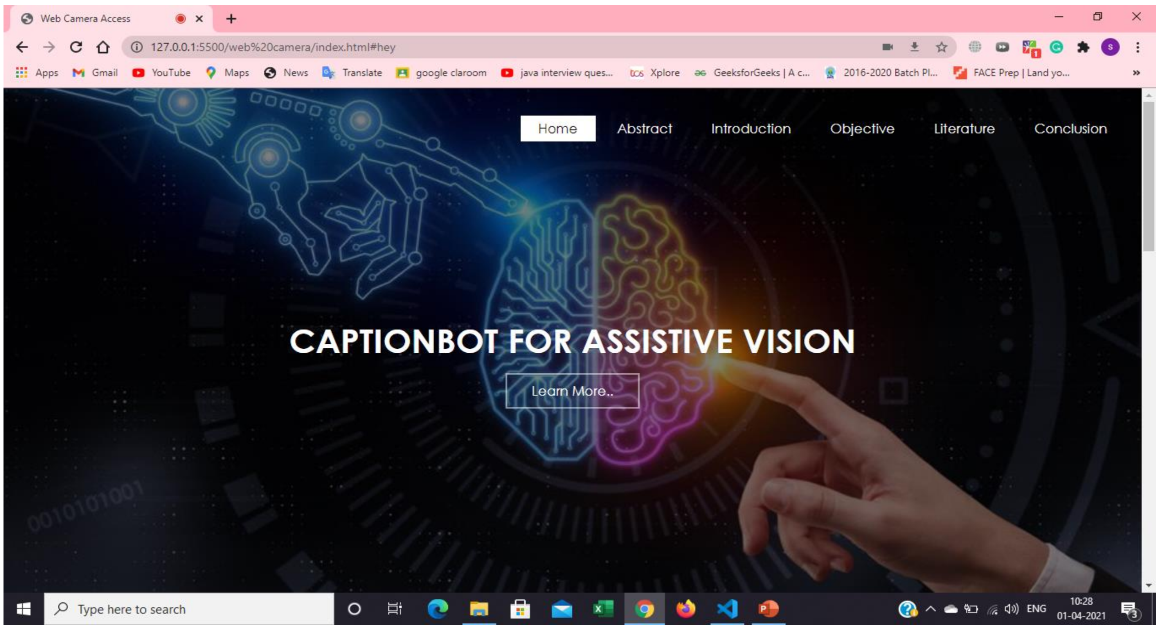
3.10 *Web Technology*

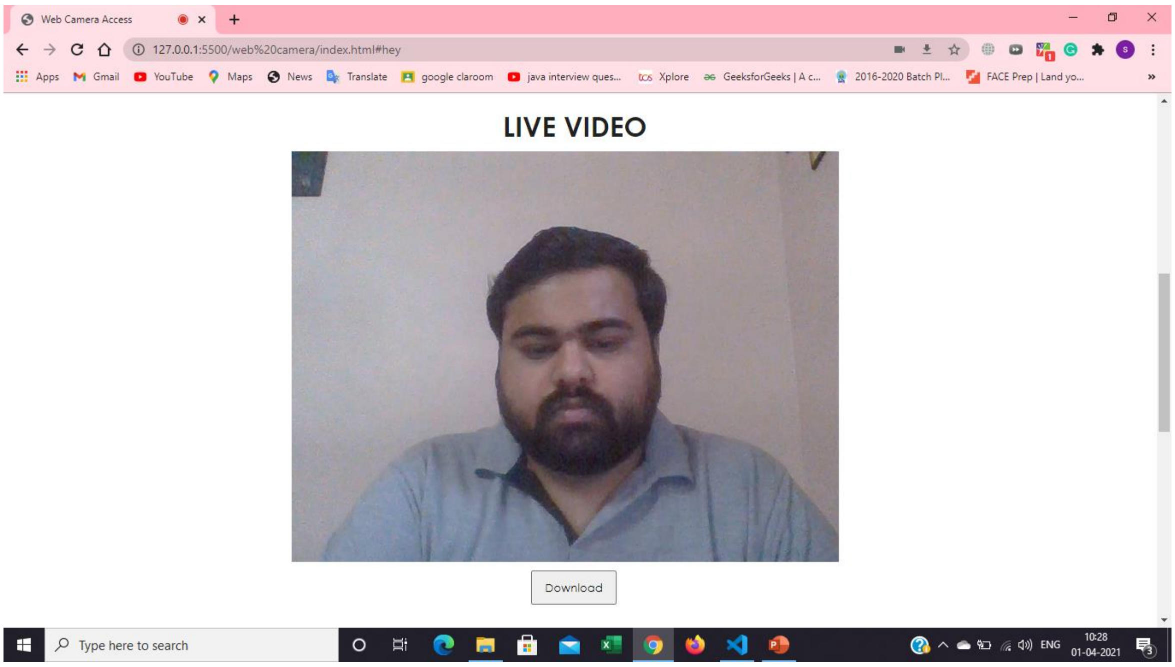
- The **Hypertext Markup Language**, or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.HTML can implant programs written in a prearranging language like JavaScript, which influences the conduct and substance of site pages. Incorporation of CSS characterizes the look and format of substance. The World Wide Web Consortium (W3C), previous maintainer of the HTML and current maintainer of the CSS norms, has empowered the utilization of CSS over express presentational HTMLsince 1997.
- **Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- CSS is intended to empower the division of show and substance, including design, shadings,

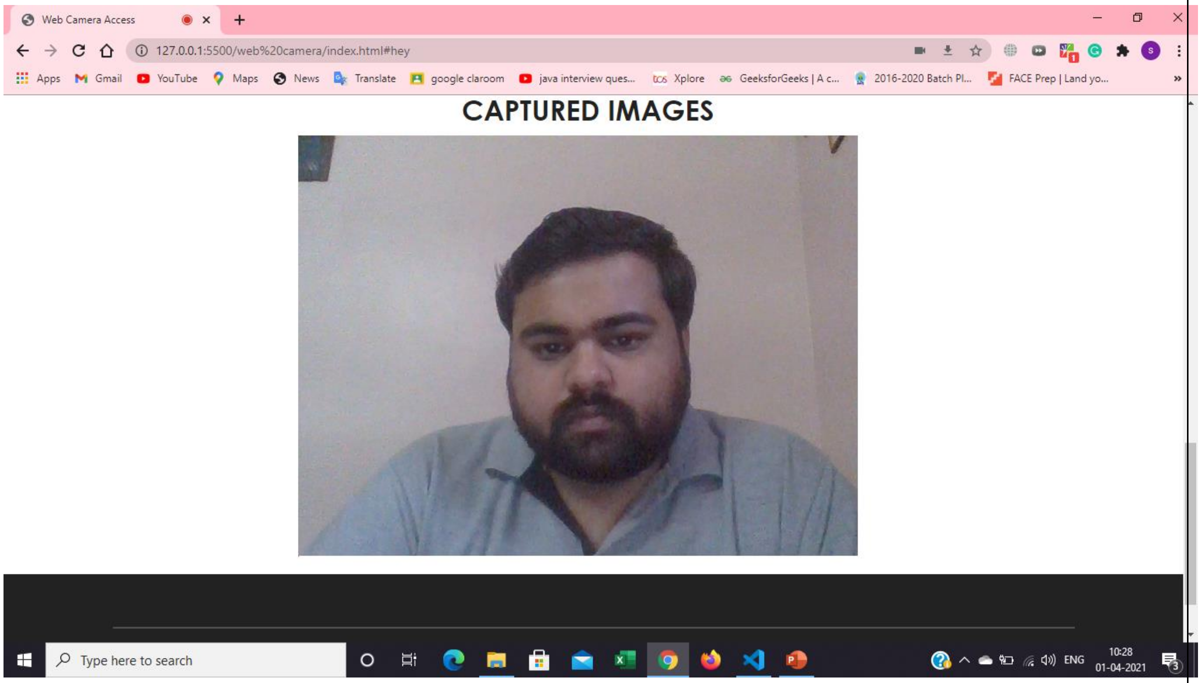
and textual styles. This detachment can improve content openness, give greater adaptability and control in the determination of show attributes, empower numerous site pages to share organizing by indicating the significant CSS in a different .css document which diminishes intricacy and reiteration in the underlying substance just as empowering the .css record to be stored to improve the page load speed between the pages that share the record and its arranging.

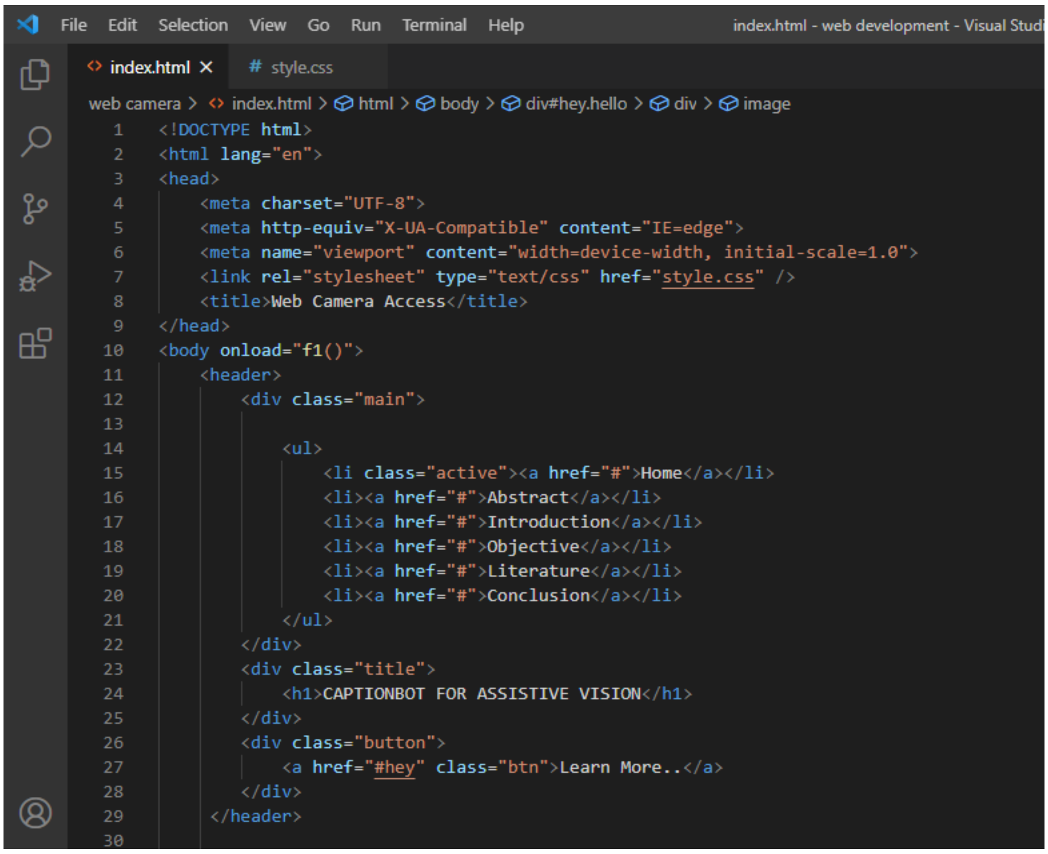
- **JavaScript (JS)** is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

- **Flask** is a lightweight WSGI (Web Server Gateway Interface) web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

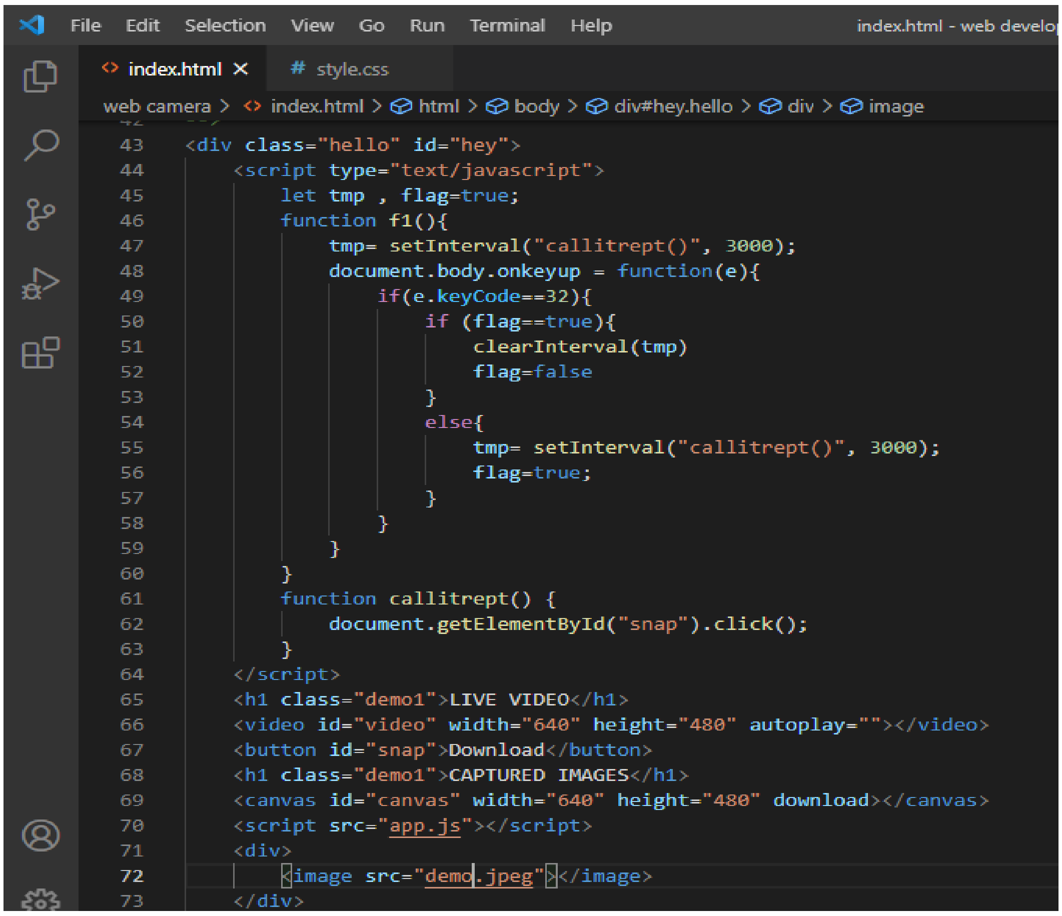



Web Camera Access
127.0.0.1:5500/web%20camera/index.html#hey
LIVE VIDEO
Download
Type here to search

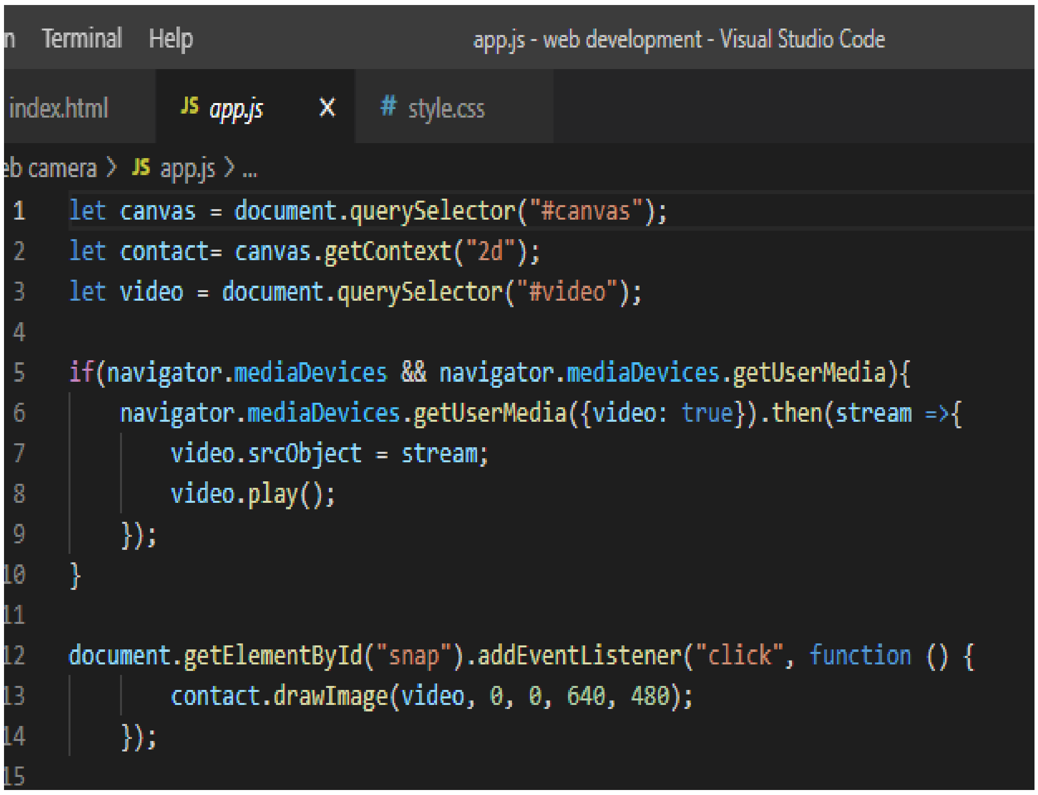

Web Camera Access
127.0.0.1:5500/web%20camera/index.html#hey
CAPTURED IMAGES
Type here to search


File Edit Selection View Go Run Terminal Help
index.html - web development - Visual Studi
index.html
style.css
web camera > index.html > html > body > div#hey.hello > div > image

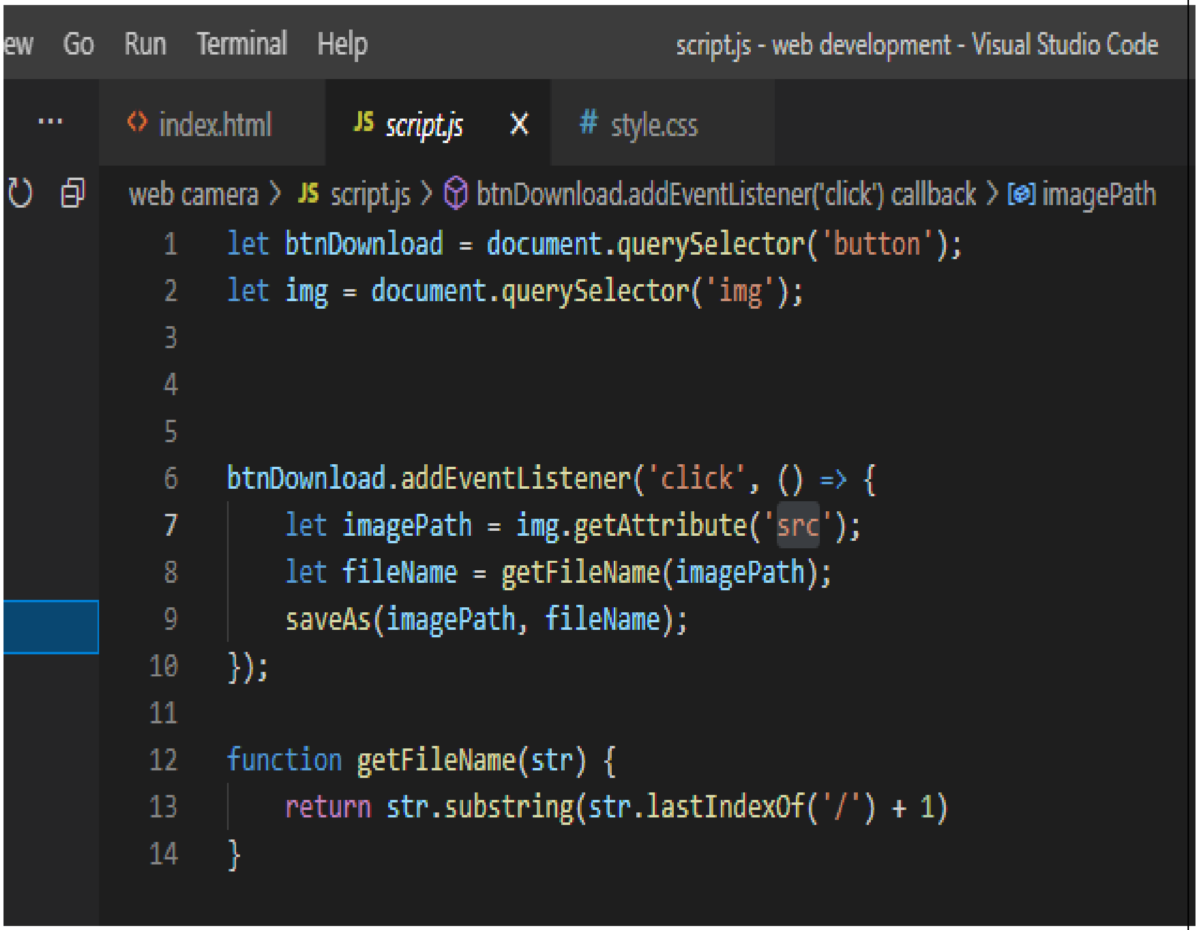
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css" />
    <title>Web Camera Access</title>
</head>
<body onload="f1()">
    <header>
        <div class="main">

            <ul>
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#">Abstract</a></li>
                <li><a href="#">Introduction</a></li>
                <li><a href="#">Objective</a></li>
                <li><a href="#">Literature</a></li>
                <li><a href="#">Conclusion</a></li>
            </ul>
        </div>
        <div class="title">
            <h1>CAPTIONBOT FOR ASSISTIVE VISION</h1>
        </div>
        <div class="button">
            <a href="#hey" class="btn">Learn More..</a>
        </div>
     </header>
```


File Edit Selection View Go Run Terminal Help
index.html - web develop
index.html
style.css
web camera > index.html > html > body > div#hey.hello > div > image

```html
<div class="hello" id="hey">
    <script type="text/javascript">
        let tmp , flag=true;
        function f1(){
            tmp= setInterval("callitrept()", 3000);
            document.body.onkeyup = function(e){
                if(e.keyCode==32){
                    if (flag==true){
                        clearInterval(tmp)
                        flag=false
                    }
                    else{
                        tmp= setInterval("callitrept()", 3000);
                        flag=true;
                    }
                }
            }
        }
        function callitrept() {
            document.getElementById("snap").click();
        }
    </script>
    <h1 class="demo1">LIVE VIDEO</h1>
    <video id="video" width="640" height="480" autoplay=""></video>
    <button id="snap">Download</button>
    <h1 class="demo1">CAPTURED IMAGES</h1>
    <canvas id="canvas" width="640" height="480" download></canvas>
    <script src="app.js"></script>
    <div>
        <image src="demo.jpeg"></image>
    </div>
```



```
let canvas = document.querySelector("#canvas");
let contact= canvas.getContext("2d");
let video = document.querySelector("#video");

if(navigator.mediaDevices && navigator.mediaDevices.getUserMedia){
    navigator.mediaDevices.getUserMedia({video: true}).then(stream =>{
        video.srcObject = stream;
        video.play();
    });
}

document.getElementById("snap").addEventListener("click", function () {
        contact.drawImage(video, 0, 0, 640, 480);
    });

```


ew Go Run Terminal Help
script.js - web development - Visual Studio Code
...
index.html
script.js
style.css
web camera > script.js > btnDownload.addEventListener('click') callback > imagePath

```
let btnDownload = document.querySelector('button');
let img = document.querySelector('img');



btnDownload.addEventListener('click', () => {
    let imagePath = img.getAttribute('src');
    let fileName = getFileName(imagePath);
    saveAs(imagePath, fileName);
});

function getFileName(str) {
    return str.substring(str.lastIndexOf('/') + 1)
}
```

Chapter4:Performance Analysis

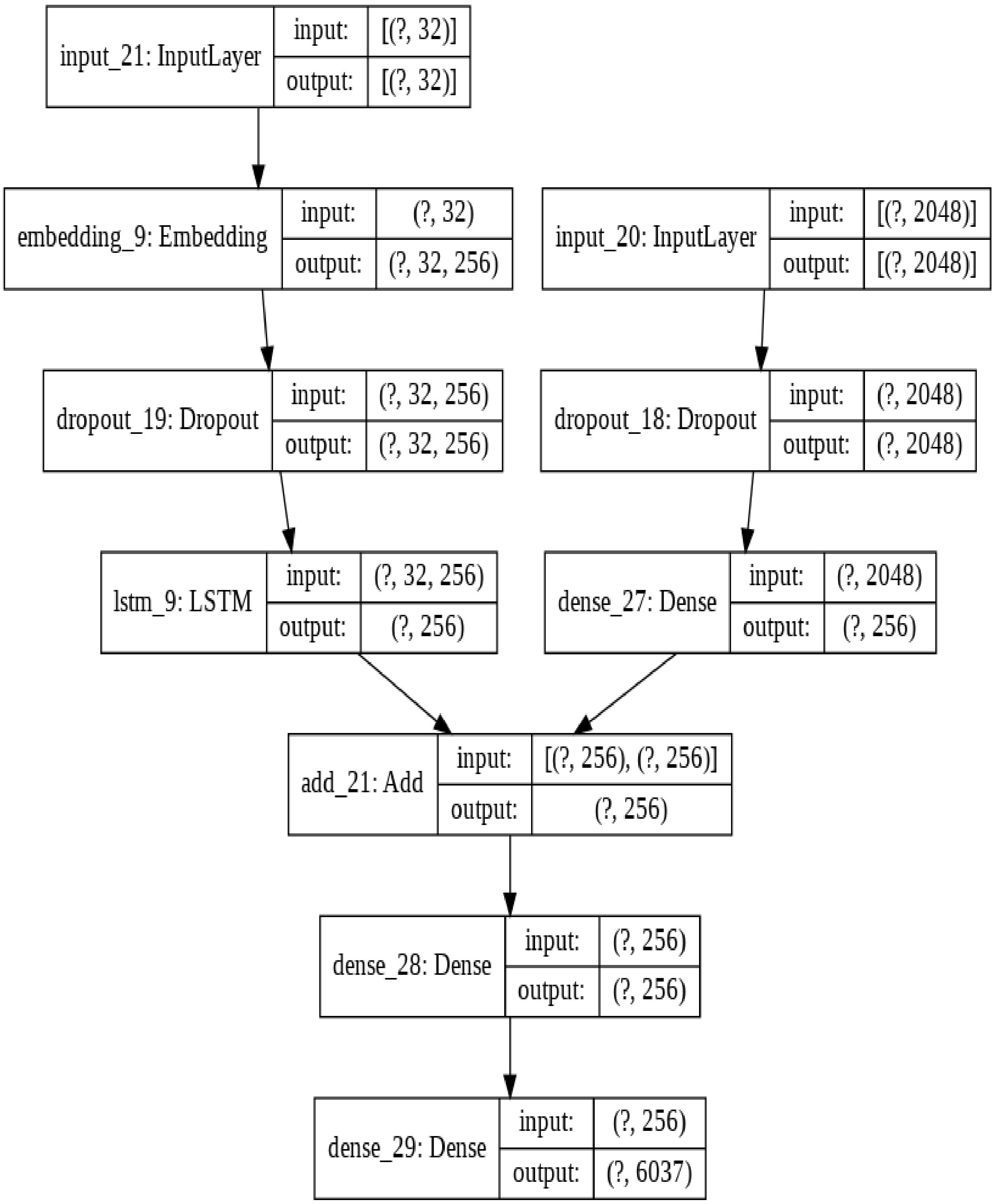


Figure 4.1 Final view of our model

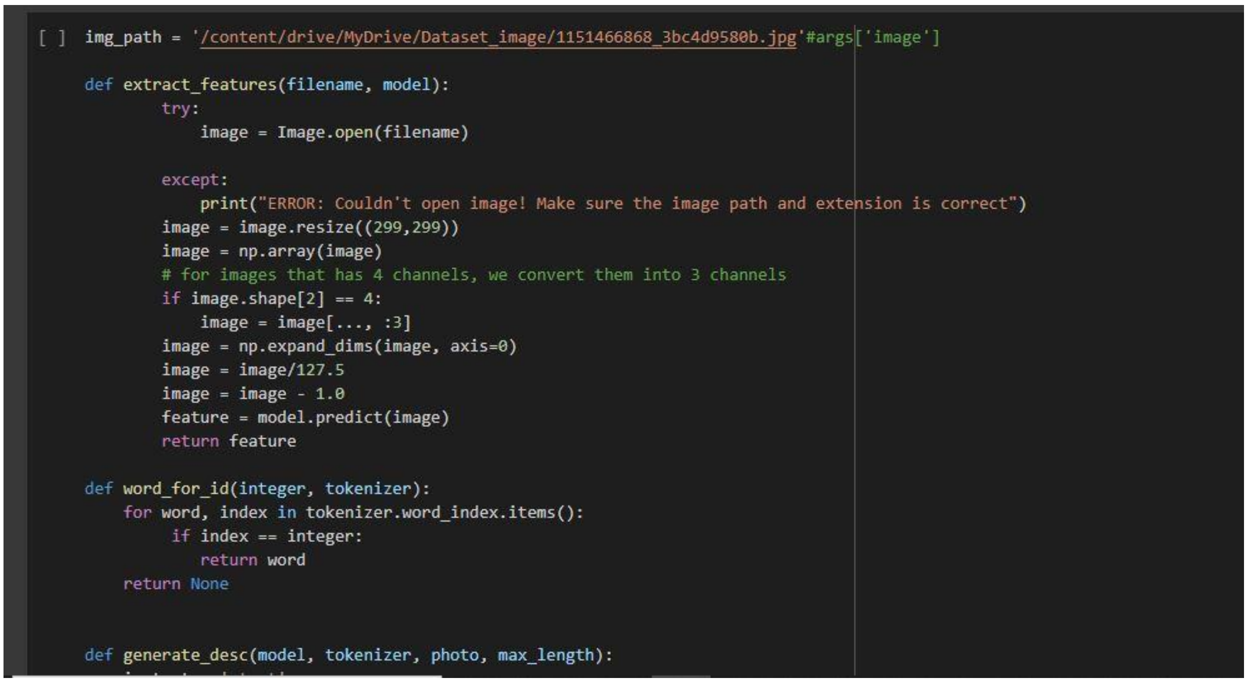
Since our model is information driven and prepared start to finish, and given the wealth of datasets, we needed to respond to questions, for example, "how dataset size influences speculation", "what sorts of move realizing it is ready to accomplish", and "how it would manage pitifully marked models". Thus, we performed probes five distinctive datasets, clarified in Section 4.2, which empowered us to comprehend our model inside and out.

A significant number of the difficulties that we confronted when preparing our models had to do with overfitting. In reality, simply managed approaches require a lot of information, yet the datasets that are of great have under 100000 pictures. The undertaking of appointing a depiction is stringently harder than object order and information driven methodologies have as of late become prevailing on account of datasets as extensive as ImageNet (with multiple times more information than the datasets we portrayed in this venture, except for SBU). Subsequently, we accept that, even with the outcomes we acquired which are very acceptable, the upside of our technique versus most current human-designed methodologies will just expansion in the following not many years as preparing set sizes will develop. In any case, we investigated a few strategies to manage overfitting. The most clear approach to not overfit is to introduce the loads of the CNN part of our framework to a pretrained model (e.g., on ImageNet). We did this in every one of the tests (like [8]), and it helped a considerable amount as far as speculation. Another arrangement of loads that could be reasonably introduced are We, the word embeddings. We took a stab at introducing them from an enormous news corpus [22], however no critical increases were noticed, and we chose to simply leave them uninitialized for effortlessness. Finally, we did some model level overfitting-staying away from procedures. We attempted dropout [34] and ensembling models, just as investigating the size (i.e., limit) of the model by compromising number of covered up units versus profundity. Dropout and ensembling gave a couple of BLEU focuses improvement, and that is the thing that we report all through the paper.

We prepared all arrangements of loads utilizing stochastic angle drop with fixed learning rate and no energy. All loads were haphazardly introduced with the exception of the CNN loads, which we left unaltered on the grounds that transforming them had an adverse consequence. We utilized 512 measurements for the embeddings what's more, the size of the LSTM memory.

4.1 *Testing themodel*

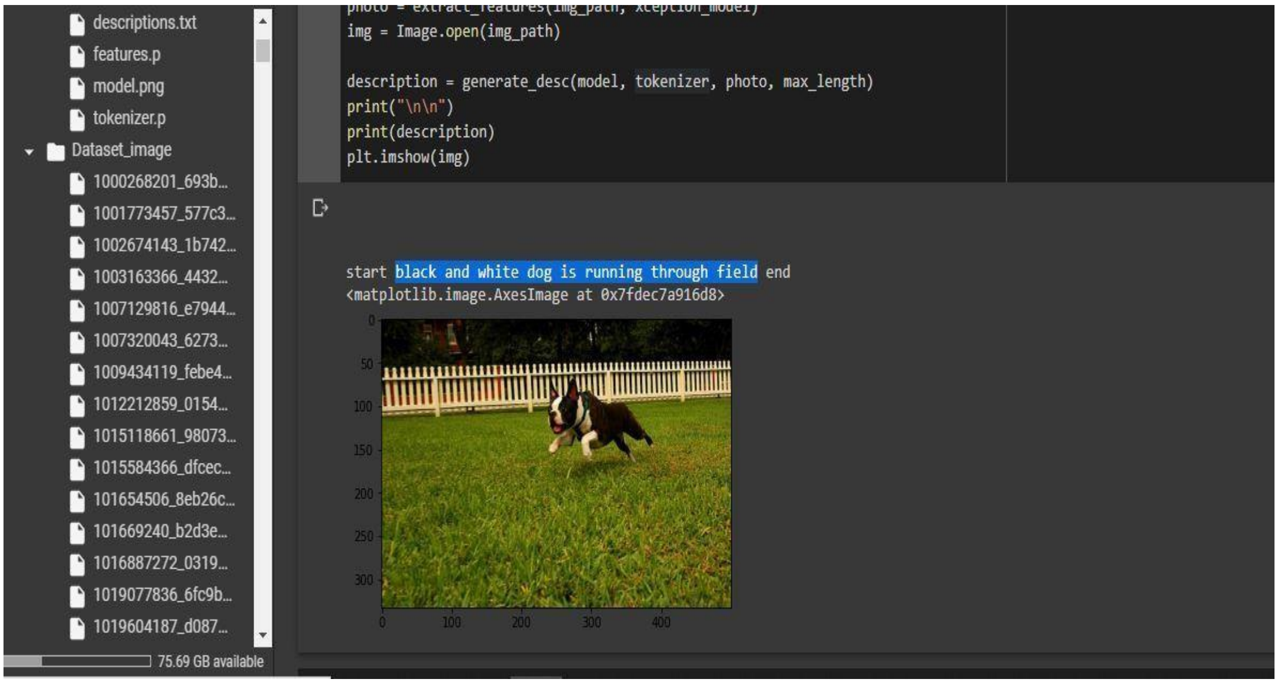
The model has been prepared, presently, we will make a different record testing_caption_generator.py which will stack the model and create expectations. The forecasts contain the maximum length of record esteems so we will utilize the equivalent tokenizer.p pickle document to get the words from their list esteems.



```
[ ] img_path = '/content/drive/MyDrive/Dataset_image/1151466868_3bc4d9580b.jpg'#args['image']

    def extract_features(filename, model):
            try:
                image = Image.open(filename)

            except:
                print("ERROR: Couldn't open image! Make sure the image path and extension is correct")
            image = image.resize((299,299))
            image = np.array(image)
            # for images that has 4 channels, we convert them into 3 channels
            if image.shape[2] == 4:
                image = image[..., :3]
            image = np.expand_dims(image, axis=0)
            image = image/127.5
            image = image - 1.0
            feature = model.predict(image)
            return feature

    def word_for_id(integer, tokenizer):
        for word, index in tokenizer.word_index.items():
             if index == integer:
                return word
        return None


    def generate_desc(model, tokenizer, photo, max_length):
```

Figure 4.2 Code snippet for testing

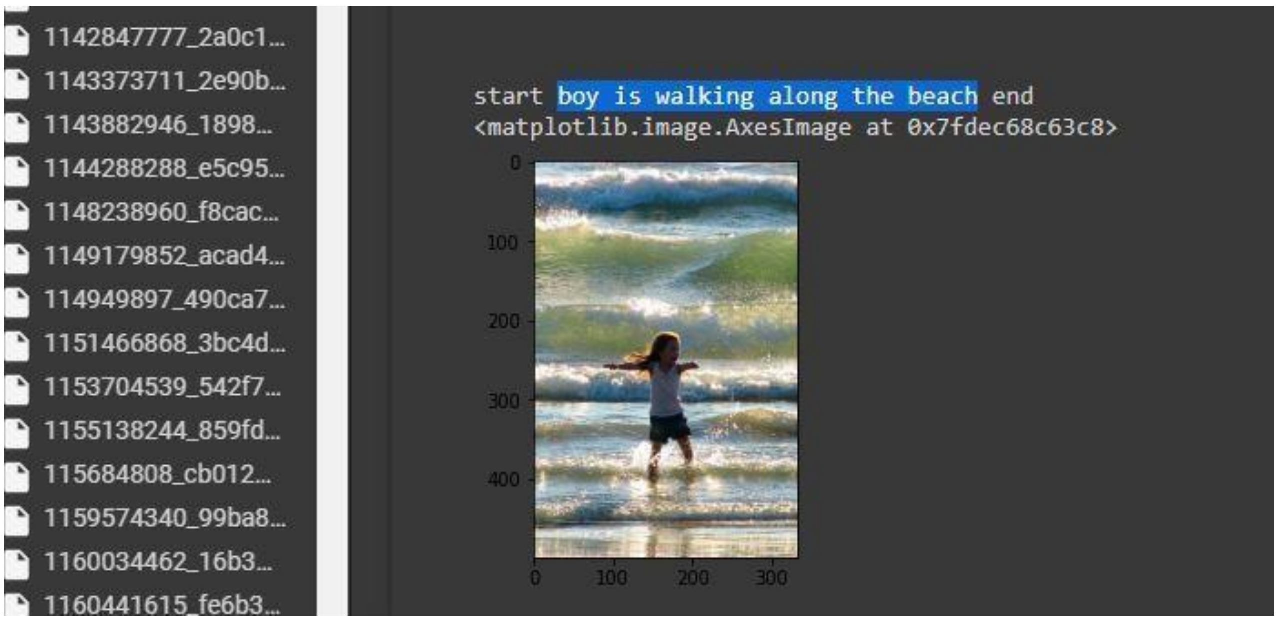
Example Outputs:



```
photo = extract_features(img_path, xception_model)
img = Image.open(img_path)

description = generate_desc(model, tokenizer, photo, max_length)
print("\n\n")
print(description)
plt.imshow(img)
```

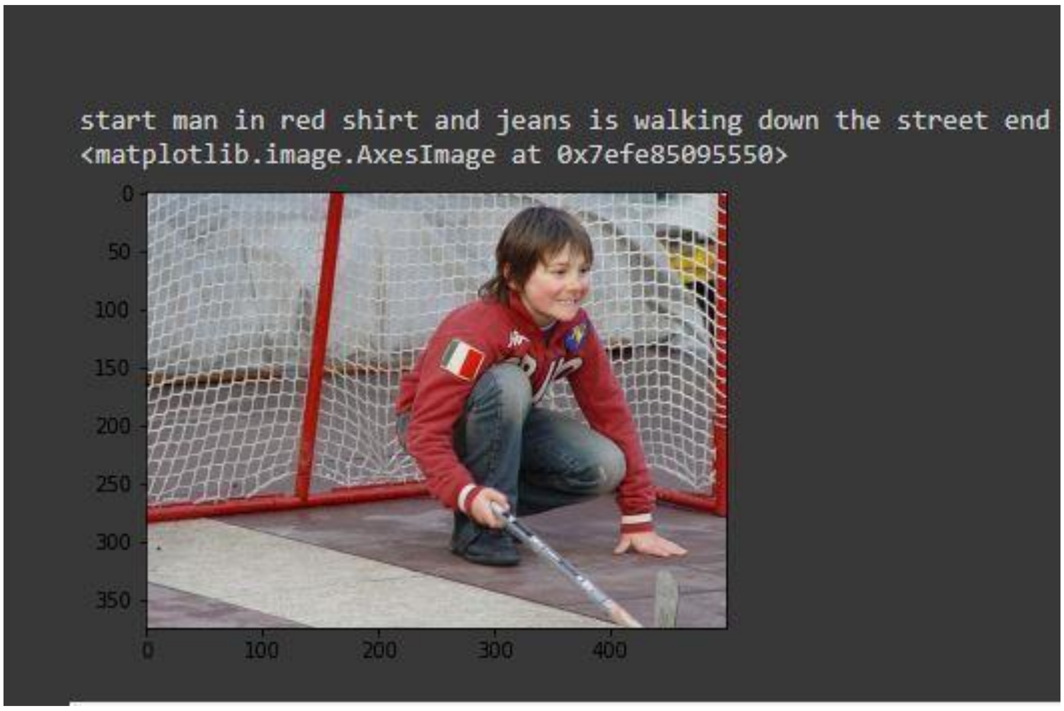
```
start black and white dog is running through field end
<matplotlib.image.AxesImage at 0x7fdec7a916d8>
```

Figure 4.3 Result-1



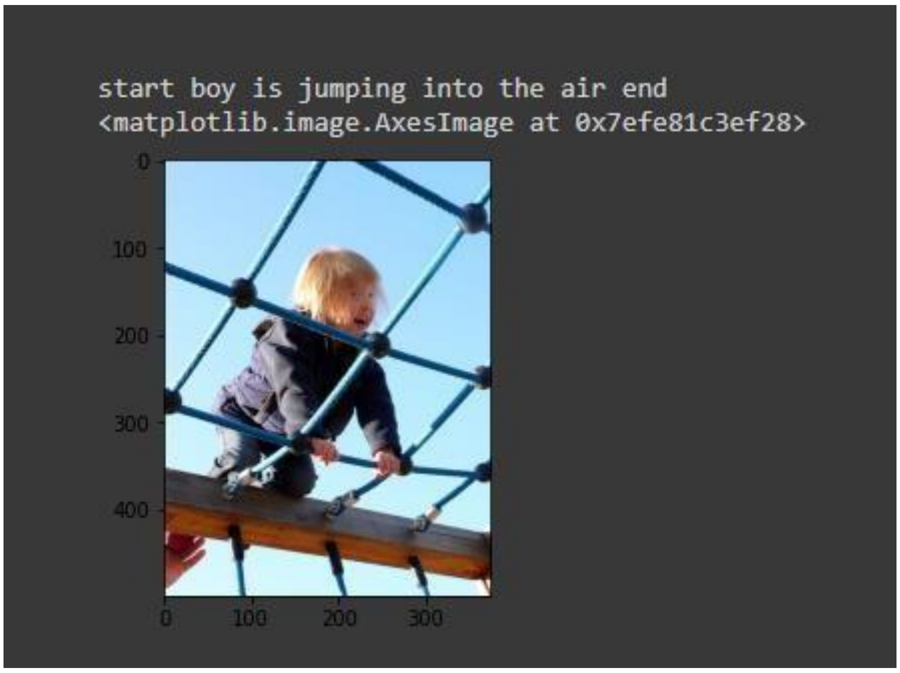
```
start boy is walking along the beach end
<matplotlib.image.AxesImage at 0x7fdec68c63c8>
```

Figure 4.4 Result-2



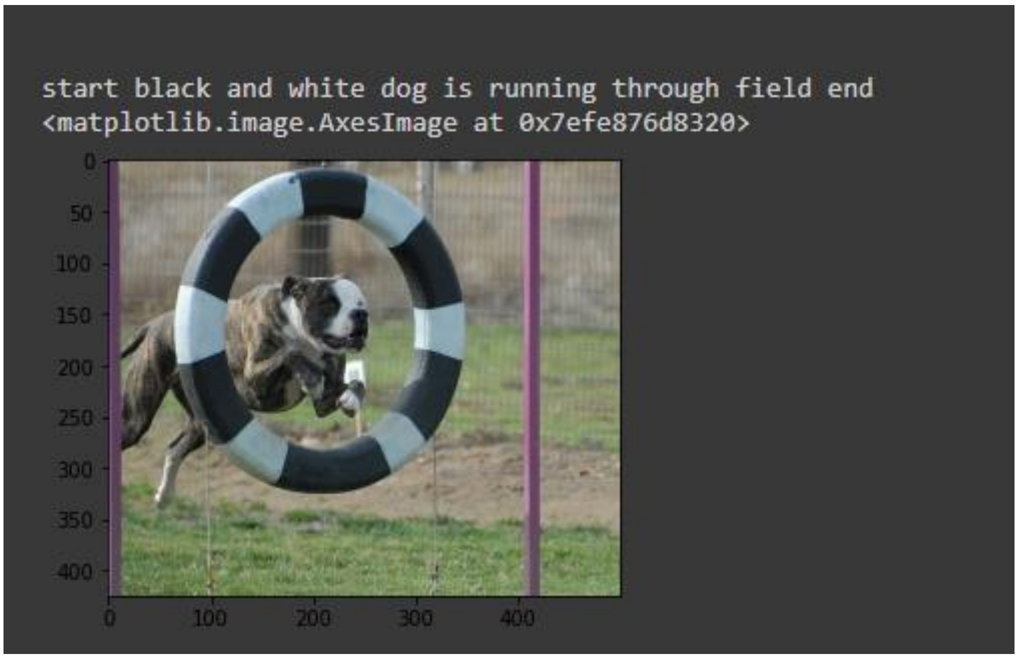
```
start man in red shirt and jeans is walking down the street end
<matplotlib.image.AxesImage at 0x7efe85095550>
```

Figure 4.5 Result-3



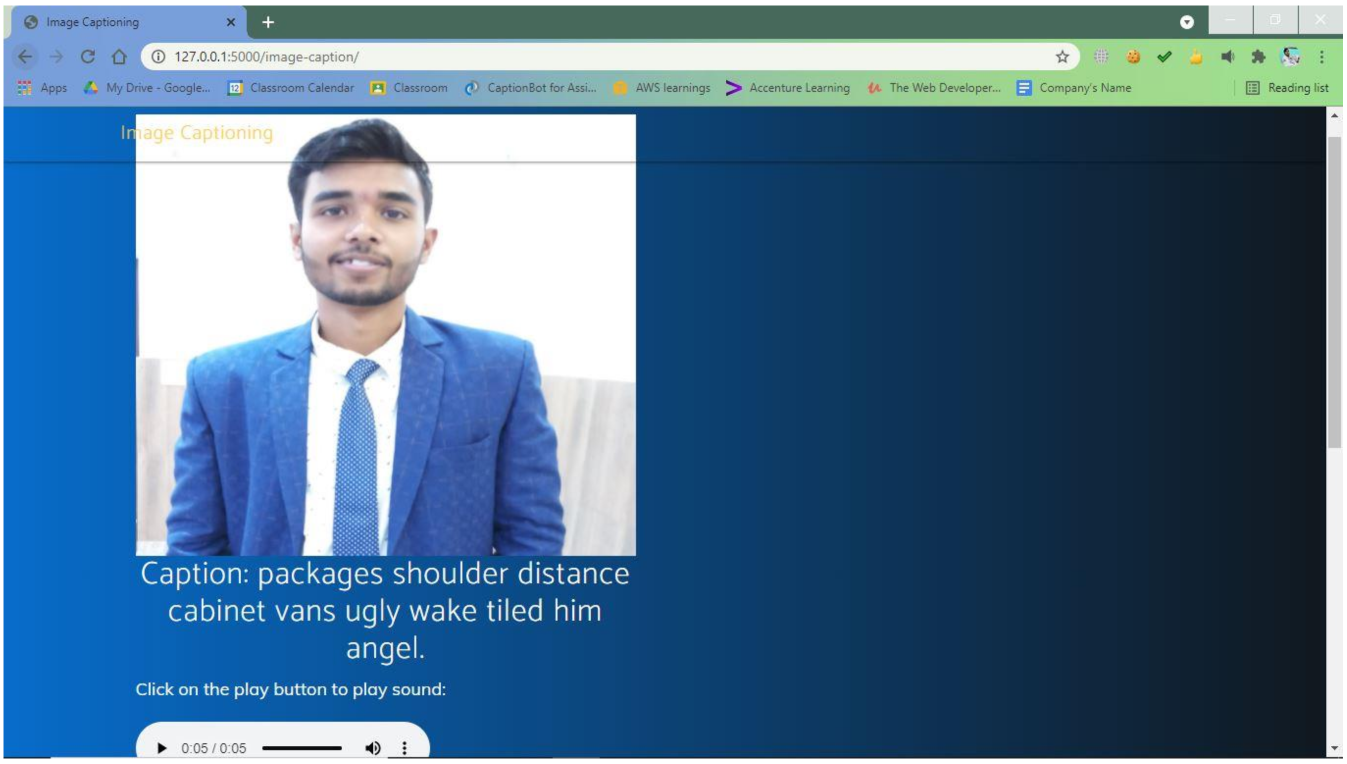
```
start boy is jumping into the air end
<matplotlib.image.AxesImage at 0x7efe81c3ef28>
```

Figure 4.6 Result-4



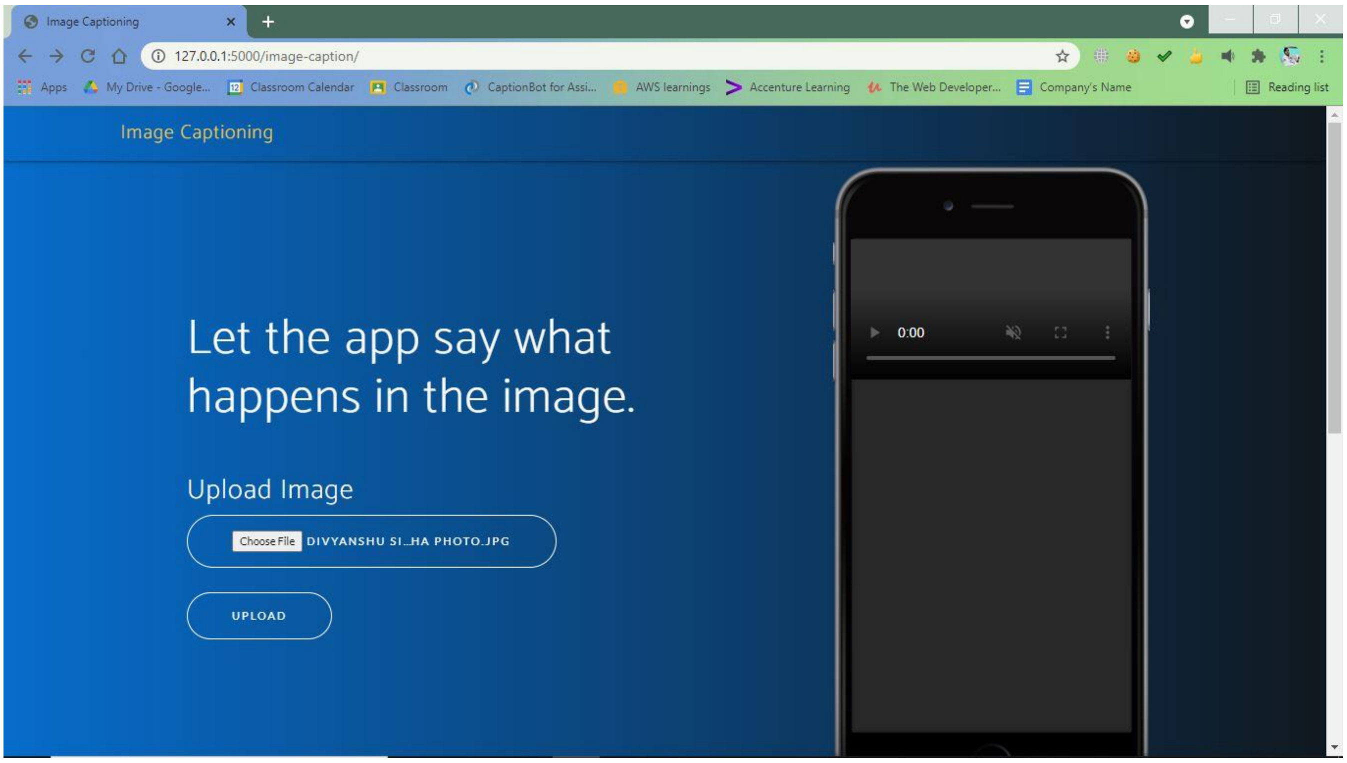
```
start black and white dog is running through field end
<matplotlib.image.AxesImage at 0x7efe876d8320>
```

Figure 4.7 Result-5



Caption: packages shoulder distance cabinet vans ugly wake tiled him angel.

Click on the play button to play sound:



Let the app say what happens in the image.

Upload Image

4.2 *Evaluation Metric*

For each picture we expect an inscription that gives a right however concise clarification in substantial English of the pictures. The closer the created inscription is to the subtitles composed by laborers on Amazon mechanical Turk thebetter.

The viability of our model is tried on 40,000 pictures contained in the Microsoft COCO dataset. We assess the produced inscriptions utilizing the accompanying measurements: BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit Ordering), and CIDEr (Consensus-based Image Description Evaluation). Every strategy assesses an applicant sentence by estimating how well it coordinates a bunch of five reference sentences composed by people. The BLEU score is processed by checking the quantity of matches between the n-grams of the up-and-comer inscription and the n-grams of the reference subtitle. METEOR was intended to fix a portion of the issues found in the more mainstream BLEU metric, and furthermore produce great relationship with human judgment at the sentence or fragment level. METEOR varies from the BLEU metric in that BLEU looks for connection at the corpus level. The CIDEr metric was explicitly created for assessing picture subtitles. It is a proportion of agreement dependent on how frequently n-grams in applicant inscriptions are available in references subtitles. It gauges the agreement in picture inscriptions by playing out a Term Frequency Inverse Document Frequency (TF-IDF) weighting for every n-gram, in light of the fact that incessant n-grams in references are less enlightening. For each of the three measurements (for example BLEU, METEOR, and CIDEr) thehigher thescore,thebetter theapplicantinscription.

4.3 *SoftmaxLoss*

At each time-step, we create a score for each word in the vocabulary. We at that point utilize the ground truth words in mix with the softmax function to register the misfortunes and gradients. We entirety the misfortunes after some time and normal them over the minibatch. Since we work over minibatches and in light of the fact that diverse produced sentences may have various lengths, we attach NULL tokens to the furthest limit of each subtitle so they all have similar lengths. Likewise, our misfortune function acknowledges a veil exhibit that illuminates it on which components regarding the scores includes towards the misfortune to forestall the NULL tokens to tally towards the misfortune or gradient.

4.4 *AdamOptimizer*

We utilize Stochastic Gradient Descent (SGD) with smaller than normal clusters of 25 picture sentence sets and energy of 0.95. We cross-approve the learning rate and the weight rot. We accomplished our best outcomes utilizing Adam, which is a technique for proficient stochastic streamlining that just requires first-request gradients and processes individual versatile taking in rates for various boundaries from evaluations of first and second snapshots of the gradients. Adam's principle preferences are that the extents of boundary refreshes are invariant to rescaling of the gradients, its progression size is roughly limited by the progression size hyperparameter, and it naturally plays out a type of step-size toughening

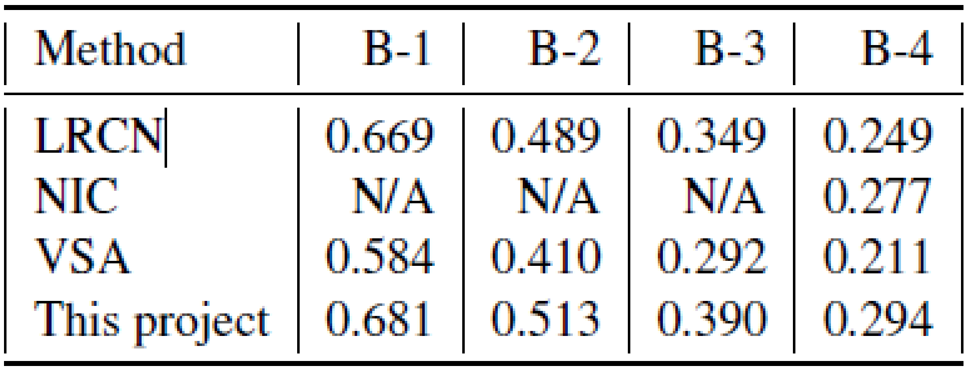
4.5 *QuantitativeAnalysis*

Assessment measurements We use BLEU (Papineni etal., 2002) to gauge the likeness of the subtitles produced by our strategy and individuals. BLEU is a mainstream machine interpretation metric that investigates the co-events of n-grams between the applicant and reference sentences. The unigram scores (B-1) represent the ampleness of the interpretation, while longer n-gram scores (B2, B-3, B-4) represent the familiarity. Distinctive CNNs looks at the presentation of three CNN models (the RNN part use LSTM). The VGG16 accomplishes the best exhibition (BLEU 4) and GoogLeNet has the least score. It is out of our desire from the start since GoogLeNet accomplishes the best presentation in the ImageNet groupingtask.

We examined this wonder with our colleague's understudies. One of them brought up that regardless of its somewhat more vulnerable characterization execution, the VGG16 highlights outflank those of GoogLeNet in various exchange learning assignments (Karpathy, 2015). A drawback of the VGG16 is that it is more costly to assess and it utilizes significantly more memory (11.6 Gb) and boundaries (138 million). It requires some investment to prepare VGG16 and GoogleNet than AlexNet (around 8 hours versus 4 hours).

Distinctive sentence age strategies additionally dissect the effect of bar size in the Beam Search for various CNN structures. All in all, bigger bar size accomplishes higher BLEU score. This marvel is substantially clearer in the VGG16 than other two CNNs.

At the point when the pillar size is 1, AlexNet beats VGG16. At the point when the shaft size is 10, the VGG16 beats AlexNet. The most plausible explanation is that AlexNet is acceptable at recognizing a solitary or few articles in a picture while VGG16 is acceptable at distinguishing various items in a similar picture. At the point when the shaft size increases, the VGG16 based technique can produce more precise sentences.



Method	B-1	B-2	B-3	B-4
LRCN	0.669	0.489	0.349	0.249
NIC	N/A	N/A	N/A	0.277
VSA	0.584	0.410	0.292	0.211
This project	0.681	0.513	0.390	0.294

Assessment of picture subtitle of various strategies. LRCN is tried on the approval set (5,000 pictures). NIC is tried on the approval set (4,000 pictures). VSA is tried on the test set (40,775 pictures). This venture is tried on the approval set (1,000 pictures for B-1, B-2, B-3, and 100 pictures for B-4).

We further contrast the content contingent strategies and best in class consideration-based techniques. For 1-gram word-contingent technique, the consideration on the picture highlight direction is simply dictated by the recently produced word. Clearly, this outcomes in semantic data misfortune. This is steady with our outcome that 1-gram word-restrictive strategy gets somewhat lower scores in a large portion of the measurements than e2e-gLSTM. However, its exhibitions are as yet comparable to or better than stateof-the-craftsmanship consideration-based strategies, for example, Hard-Attention and ATT-FCN. We at that point redesign the word- contingent model to the sentence-restrictive one which prompts around 1% improvement in Bleu and METEOR contrasted with 1-gram word-contingent strategy, and it beats e2e-gLSTM in a large portion of the measurements aside from CIDEr.

With respect to consideration, our model is simply ready to perceive the main piece of the pictures. That is, the considerations at each progression are the equivalent. There are 2 significant reasons. Initially, since the highlights are contribution at the initial step of LSTM, the general data of the picture has been feed into the decoder, which is sufficient to produce a fair sentence, and in this manner the accompanying sources of info can be coarser. This is actually the inspiration of our different models. They are potential to work better given more finetune. Also, the responsive field of commencement 5 is very huge (139 x 139). So to zero in on the fundamental piece of picture is sufficient to get a decentsentence.

To address this issue, we can utilize lower level highlights from CNN with more expressive fatt, i.e., to develop the fatt CNN and grow the quantity of concealed units in each layer. We utilized BLEU score as quantitative measurement for our outcomes. We can see our model can't accomplish the cutting edge. To improve results, we ought to broaden our model, train and tune it further.

Chapter 5: Conclusion

5.1 Conclusion

We have presented a deep learning model that automatically generates image captions with the goal of helping visually impaired people better understand their environments. Our described model is based on a CNN that encodes an image into a compact representation, followed by a RNN that generates corresponding sentences based on the learned image features. We showed that this model achieves comparable to state-of-the-art performance, and that the generated captions are highly descriptive of the objects and scenes depicted on the images. Because of the high quality of the generated image descriptions, visually impaired people can greatly benefit and get a better sense of their surroundings using text-to-speech technology.

In any case, there are a few basic issues in our technique. The first is overfitting. As the Google NIC paper called attention to, we can't get to enough preparing tests, in any event, for the moderately immense dataset MSCOCO. One potential arrangement is joining feebly commented on pictures with current dataset. The subsequent issue is time multifaceted nature. We calibrate the VGG net, which requires additional preparation time. Despite the fact that we use move learning strategies to make the delay shrivel, endeavors should in any case be placed in building quick learningframeworks.

Working in a team, we could discuss and refine a lot of initial ideas. We could also anticipate problems that could become critical of the cases we were working alone.

5.2 Futurescope

Future work can include this text-to-speech technology, so that the generated descriptions are automatically read out loud to visually impaired people. In addition, future work could focus on translating videos directly to sentences instead of generating captions of images. Static images can only provide blind people with information about one specific instant of time, while video caption generation could potentially provide blind people with continuous real time information. LSTMs could be used in combination with CNNs to translate videos to English descriptions.

We might want to investigate techniques to produce various sentences with various substance.Onepotentialrouteistojoinintriguinglocalelocationandpictureinscribing.

In future, this prototype can be deploy in the hardware and automate all the process which is done manually currently by us. Certain optimization should also be made in order to improve the performance of the model.