

**A PROJECT REPORT**

On

**Image caption generator**

*Submitted In Partial Fulfillment of the Requirement For The Award of*

**POSTGRADUATE DIPLOMA IN BIG DATA ANALYTICS**

Under the Guidance of

Nimesh Dagur

(Project Guide)



Submitted By:

**Mohd Jibran (210920525028)**

**Mohd Asif (210920525027)**

CERTIFICATE

**This is to certify that Image caption generator which is submitted by Mohd Jibran and Mohd Asif in partial fulfillment of the requirement for the award of PG Diploma in Post Graduate Diploma in Big Data Analytics (PG-DBDA), CDAC Noida is a record of the candidates own work carried out by them. The matter embodied in this report is original and has not been submitted for the award of any other degree.**

Name: Mohd Jibran

PRN No (210920525028)

Name: Mohd Asif

PRN No (210920525027)

ACKNOWLEDGEMENT

**We would like to express my special thanks of gratitude to Project Guide (Nimesh Dagur) who gave us the golden opportunity to do this wonderful project on the topic (Image caption generator), which also helped us in doing a lot of Research and we came to know about so many new things We are really thankful to them. Secondly we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.**

## ABSTRACT

**In Artificial Intelligence (AI), the contents of an image are generated automatically which involves computer vision The neural model which is regenerative, is created. It depends on computer vision and machine translation. This model is used to generate natural sentences which eventually describes the image. This model consists of Convolutional Neural Network(CNN) as well as LSTM The CNN is used for feature extraction from image and RNN is used for sentence generation. The model is trained in such a way that if input image is given to model it generates captions which nearly describes the image. The accuracy of model and smoothness or command of language model learns from image descriptions is tested on different datasets. These experiments show that model is frequently giving accurate descriptions for an input image.**

## INDEX

**1. Introduction to Problem Statement**

**2. Approach to the problem Statement**

**3. UNDERSTANDING THE DATASET**

**4. CNN-LSTM ARCHITECTUR**

**5. IMAGE CAPTIONING**

**6. PREPROCESSING THE IMAGE**

**7. CREATING VOCABULARY FOR THE IMAGE**

**8. CNN-LSTM MODEL**

**9. TESTING THE MODEL**

**10. CONCLUSION**

**11. REFERENCES**

Introduction

Image caption Generator is a popular research area of Artificial Intelligence that deals with image understanding and a language description for that image. Generating well-formed sentences requires both syntactic and semantic understanding of the language. Being able to describe the content of an image using accurately formed sentences is a very challenging task, but it could also have a great impact, by helping visually impaired people better understand the content of images.

This task is significantly harder in comparison to the image classification or object recognition tasks that have been well researched.

The biggest challenge is most definitely being able to create a description that must capture not only the objects contained in an image, but also express how these objects relate to each other.

Consider the following Image from the Flickr8k dataset: -



What do you see in the above image?  
  
You can easily say ‘A black dog and a brown dog in the snow’ or ‘The small dogs play in the snow’ or ‘Two Pomeranian dogs playing in the snow’. It seems easy for us as humans to look at an image like that and describe it appropriately.

Let’s see how we can create an Image Caption generator from scratch that is able to form meaningful descriptions for the above image and many more!

**Prerequisites before you get started: -**

* Python programming
* Convolutional Neural Networks (CNN)
* LSTM
* Keras

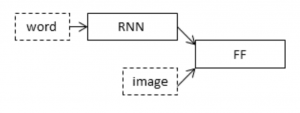
Approach to the problem statement

We will tackle this problem using an Encoder-Decoder model. Here our encoder model will combine both the encoded form of the image and the encoded form of the text caption and feed to the decoder.

Our model will treat CNN as the ‘image model’ and the LSTM as the ‘language model’ to encode the text sequences of varying length. The vectors resulting from both the encodings are then merged and processed by a Dense layer to make a final prediction.

We will create a merge architecture in order to keep the image out of the LSTM and thus be able to train the part of the neural network that handles images and the part that handles language separately, using images and sentences from separate training sets.

In our merge model, a different representation of the image can be combined with the final LSTM state before each prediction.



The above diagram is a visual representation of our approach.

## Understanding the dataset

A number of datasets are used for training, testing, and evaluation of the image captioning methods. The datasets differ in various perspectives such as the number of images, the number of captions per image, format of the captions, and image size. Three datasets: Flickr8k, Flickr30k, and MS COCO Dataset are popularly used.

In the Flickr8k dataset, each image is associated with five different captions that describe the entities and events depicted in the image that were collected. By associating each image with multiple, independently produced sentences, the dataset captures some of the linguistic variety that can be used to describe the same image.

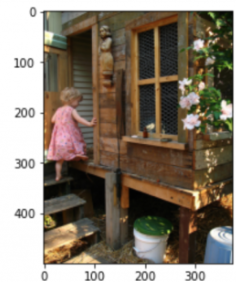
We use Flickr8k dataset is a good starting dataset as it is small in size and can be trained easily on low-end laptops/desktops using a CPU.

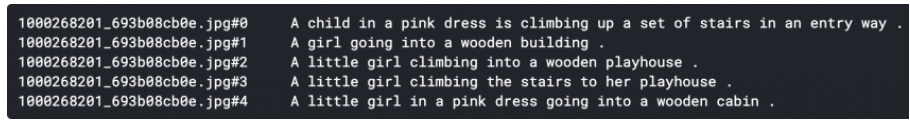
Our dataset structure is as follows:-

* Flick8k/
  + Flick8k\_Dataset/ :- contains the 8000 images

The image dataset is divided into 6000 images for training, 1000 images for validation and 1000 images for testing.

* + Flick8k\_Text/
    - Flickr8k.token.txt:- contains the image id along with the 5 captions
    - Flickr8k.trainImages.txt:- contains the training image id’s
    - Flickr8k.testImages.txt:- contains the test image id’s

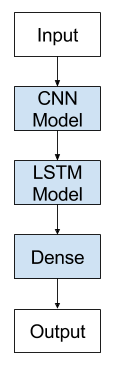
 Image and its **CAPTION**



CNN-LSTM ARCHITECTURE:

The[CNN-LSTM](https://machinelearningmastery.com/cnn-long-short-term-memory-networks/) architecture involves using CNN layers for feature extraction on input data combined with LSTMs to support sequence prediction. This model is specifically designed for sequence prediction problems with spatial inputs, like images or videos. They are widely used in Activity Recognition, Image Description, Video Description and many more.

The general architecture of the CNN-LSTM Model is as follows:



CNN-LSTMs are generally used when their inputs have spatial structure, such as the 2D structure or pixels in an image or the 1D structure of words in a sentence, paragraph, or document and also have a temporal structure in their input such as the order of images in a video or words in text, or require the generation of output with temporal structure such as words in a textual description.

## IMAGE CAPTIONING:

The goal of image captioning is to convert a given input image into a natural language description.

In this blog we will be using the concept of CNN and LSTM and build a model of Image Caption Generator which involves the concept of computer vision and Natural Language Process to recognize the context of images and describe them in natural language like English.

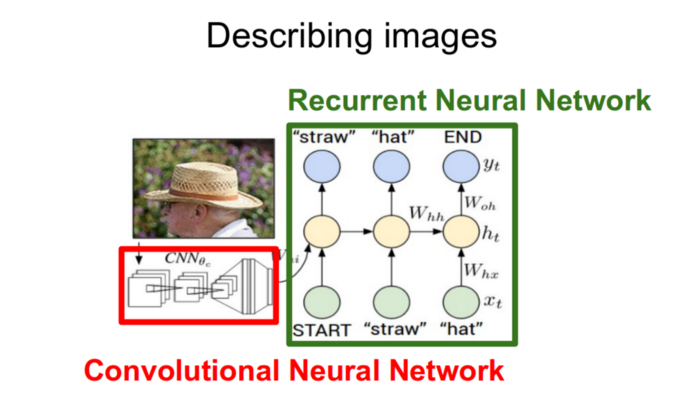
The task of image captioning can be divided into two modules logically –

1. **Image based model —** Extracts the features of our image.
2. **Language based model —** which translates the features and objects extracted by our image based model to a natural sentence.

For our image based model– we use CNN, and for language based model — we use LSTM. The following image summarizes the approach of Image Captioning Generator.

For our image based Model -Usually rely on a Convolutional Neural Network model.

For language based models — rely on LSTM. The image below summarizes the approach.



A pre-trained CNN extracts the features from our input image. The feature vector is linearly transformed to have the same dimension as the input dimension of LSTM network. This network is trained as a language model on our feature vector.

For training our LSTM model, we predefine our label and target text. For example, if the caption is “An old man is wearing a hat.”, our label and target would be as follows –

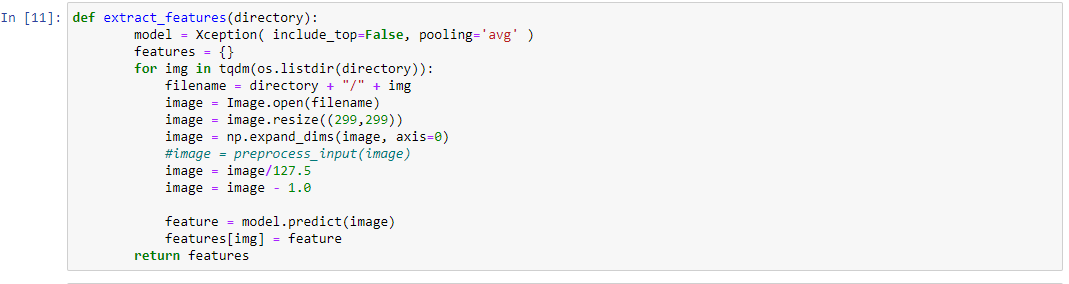
Label — [<start> ,An, old, man, is, wearing, a , hat . ]

Target — [ An old man is wearing a hat .,<End> ]

This is done so that our model understands the start and end of our labelled sequence.

## PREPROCESSING THE IMAGE:

For feature extraction, the image features are in 299\*299 size. The features of the image are extracted just before the last layer of classification as this is the model used to predict a classification for a photo. We are not interested in classifying images; hence we excluded the last layer.

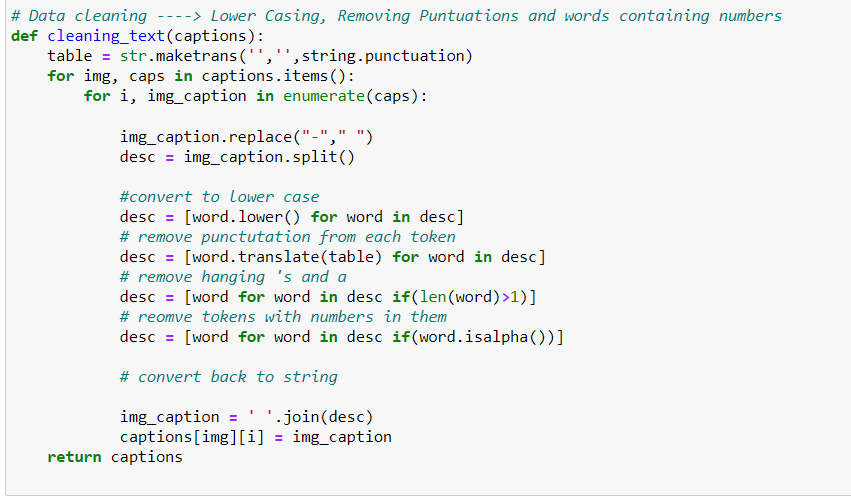


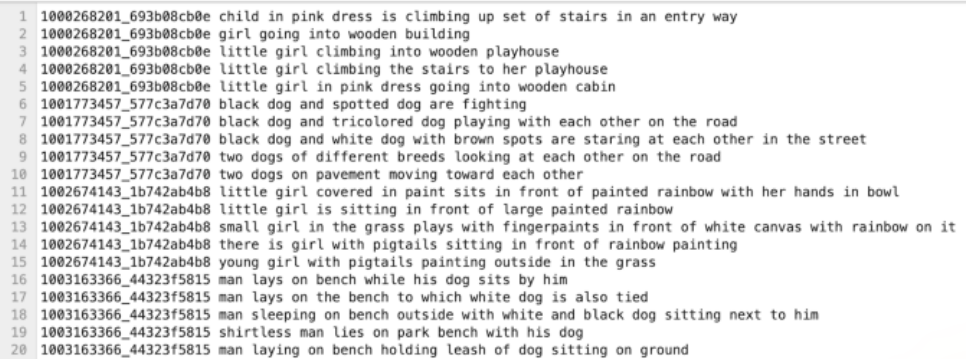
## CREATING VOCABULARY FOR THE IMAGE:

We cannot straight away take the raw text and fit it in a Machine Learning or Deep Learning model. We need to first clean the text, by splitting it into words and handle punctuation and case sensitivity issues. As computers do not understand English words, we have to represent them with numbers and map each word of the vocabulary with a unique index value, and we need to encode each word into a fixed sized vector and represent each word as a number. Only then the text can be readable by the machine and can generate the captions for the image.

We are going to clean the text in following order to achieve the size of vocabulary:

* Convert to lower case
* remove punctutation from each token
* remove hanging 's and a
* remove tokens with numbers in them



A snippet of the output file — description.txt file should look like this: 

## CNN-LSTM MODEL:

For image captioning, we are creating an LSTM based model that is used to predict the sequences of words, called the caption

To train the model, we will be using the 6000 training images by generating the input and output sequences in batches from the above data generation module and fitting them to the model. We are training the model with 30 epochs (hear you can increase no of epochs for better accuracy).

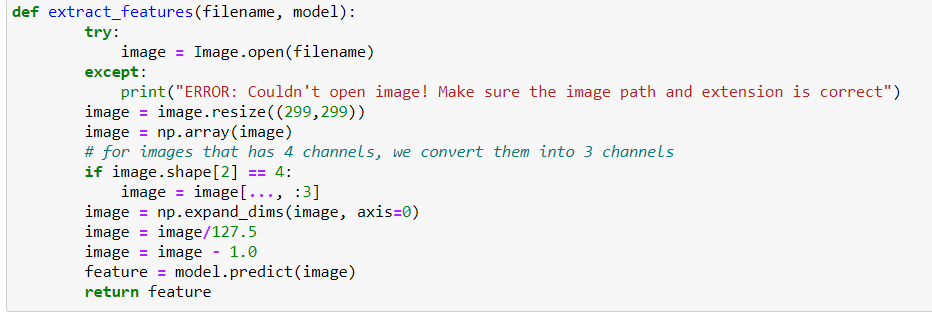


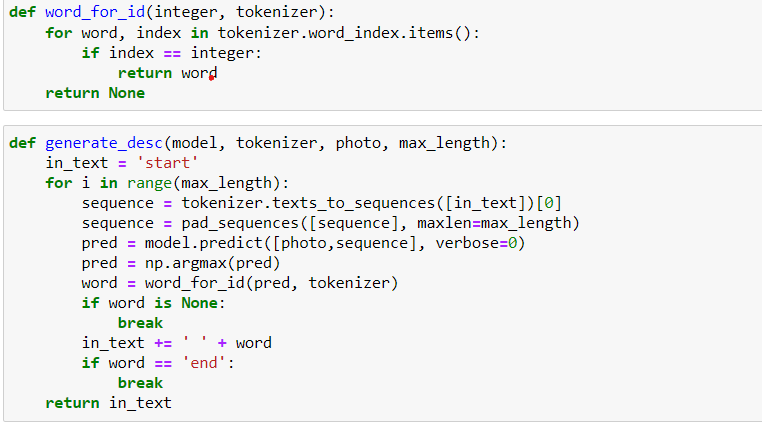
Train model

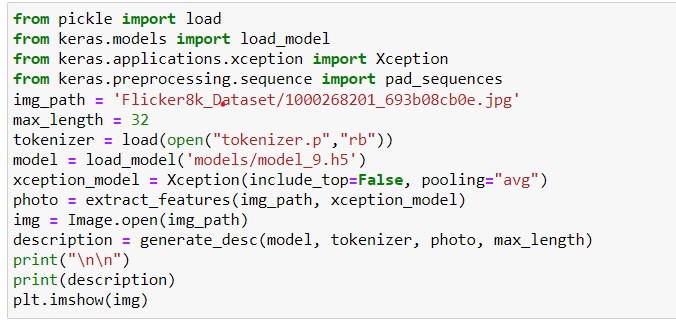


## TESTING THE MODEL

Now that the model has been trained, we can now test the model against random images. The predictions contain the max length of index values so we will use the same tokenizer that we define before to get the words from their index values.







## CONCLUSION:

We have implemented a CNN-LSTM model for building an Image Caption Generator. A CNN-LSTM architecture has wide-ranging applications which include use cases in Computer Vision and Natural Language Processing domains.

### **REFERENCES:**

<https://medium.com/swlh/automatic-image-captioning-using-deep-learning-5e899c127387>

https://www.analyticsvidhya.com/blog/2020/11/create-your-own-image-caption-generator-using-keras/