IMAGE CAPTIONING USING CNN-LSTM DEEP NEURAL NETWORKS

A MINI PROJECT REPORT

Submitted by

JINCY JOY (LTKM18MCA045)

SHIFIN T K (TKM18MCA031)

VIVEK VISWAM (TKM18MCA036)

to

The APJ Abdul Kalam Technological University

In partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

JANUARY 2O21

DECLARATION

We undersigned hereby declare that the project report IMAGE CAPTIONING USING CNN-LSTM DEEP NEURAL NETWORKS, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Prof. Vaheetha Salam. This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place JINCY JOY

Date SHIFIN.T.K

VIVEK VISWAM

Thangal Kunju Musaliar College of Engineering
Department of Computer Applications

C E R T I F I C A T E

This is to certify that, the report entitled "**IMAGE CAPTIONING USING CNN-LSTM DEEP NEURAL NETWORKS"** is a bonafide record of the work submitted by **JINCY JOY (LTKM18MCA045), SHIFIN.T.K (TKM18MCA031), VIVEK VISWAM (TKM18MCA036)**, to the **APJ Abdul Kalam Technological University** in partial fulfillment of the requirements for the award of **Master of Computer Applications** Degree under our guidance and supervision.This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor

Mini Project Coordinator

ACKNOWLEDGEMENT

First and foremost we thank GOD almighty and our parents for the success of this project.We owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of our project.

We would like to thank our coordinator and project guide **Prof.Vaheetha Salam**, Department of Computer Applications, who motivated us throughout the project.

We profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout our course of study.

We owe our thanks to our friends and all others who have directly or indirectly helped us in the successful completion of this project.

JINCY JOY
SHIFIN.T.K
VIVEK VISWAM

Abstract

Captioning images automatically is one of the heart of the human visual system. There are various advantages if there is an application which automatically caption the scenes surrounded by them and revert back the caption as a plain message. In this project, we present a model based on CNN-LSTM neural networks which automatically detects the objects in the images and generates descriptions for the images. It uses various pre-trained models to perform the task of detecting objects and uses CNN and LSTM to generate the captions. It uses Transfer Learning based pre-trained models for the task of object Detection. This model can perform two operations. The first one is to detect objects in the image using Convolutional Neural Networks and the other is to caption the images using RNN based LSTM(Long Short Term Memory). Caption generation is one of the interesting and focussed areas of Artificial Intelligence which has many challenges to pass on. Caption generation involves various complex scenarios starting from picking the dataset, training the model, validating the model, creating pre-trained models to test the images ,detecting the images and finally generating the captions. 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.

Contents

List of Figures

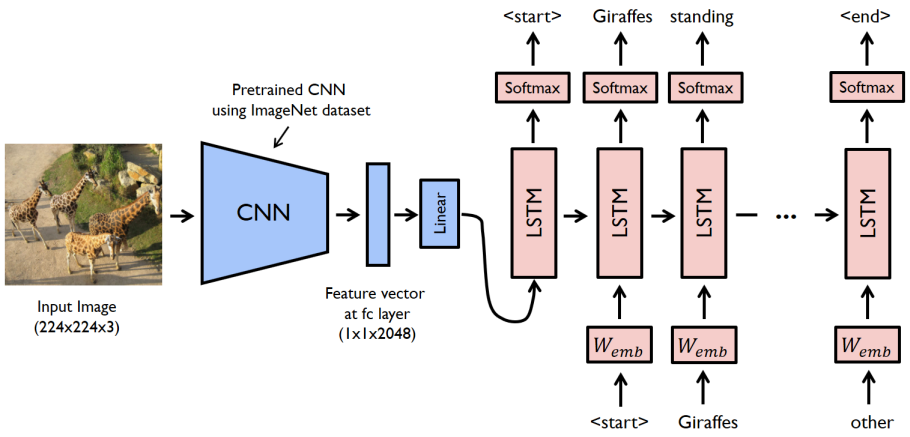
List of Tables

Chapter 1

Introduction

Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe them in a natural language like English. It has emerged as a challenging and important research area following advances in statistical language modelling and image recognition. The generation of captions from images has various practical benefits, ranging from aiding the visually impaired, to enabling the automatic and cost-saving labelling of the millions of images uploaded to the Internet every day. The field also brings together state-of-the-art models in Natural Language Processing and Computer Vision, two of the major fields in Artificial Intelligence. There are two main approaches to Image Captioning: bottom-up and top-down. Bottom-up approaches, generate items observed in an image, and then attempt to combine the items identified into a caption. Top-down approaches, attempt to generate a semantic representation of an image that is then decoded into a caption using various architectures, such as recurrent neural networks. The latter approach follows in the footsteps of recent advances in statistical machine translation, and the state-of-the-art models mostly adopt the top-down approach.

We use a deep convolutional neural network to generate a vectorized representation of an image that we then feed into a Long-Short-Term Memory (LSTM) network, which then generates captions. To make our image caption generator model, we will be merging CNN and LSTM architectures. It is also called a CNN-RNN model. CNN is used for extracting features from the image. We will use the pre-trained model Xception. LSTM will use the information from CNN to help generate a description of the image. For the image caption generator, we will be using the Flickr8K dataset.



1.1 Existing System

In order to generate captions, In the last 5 years, a large number of articles have been published on image captioning with deep machine learning being popularly used. Deep learning algorithms can handle complexities and challenges of image captioning quite well. So far, only three survey papers have been published on this research topic. Although the papers have presented a good literature survey of image captioning, they could only cover a few papers on deep learning because the bulk of them was published after the survey papers. These survey papers mainly discussed template based, retrieval based, and a very few deep learning-based image caption generating models. However, a large number of works have been done on deep learning-based image captioning. Moreover, the availability of large and new datasets has made the learning-based image captioning an interesting research area. To provide an abridged version of the literature, we present a survey mainly focusing on the deep learning-based papers on image captioning.

1.2 Proposed System

Our model uses two different neural networks to generate the captions. The first neural network is Convolutional Neural Network(CNN), which is used to train the images as well as to detect the objects in the image with the help of various pre-trained models like VGG, Inception or YOLO. The second neural network used is Recurrent Neural Network(RNN) based Long Short Term Memory(LSTM), which

is used to generate captions from the generated object keywords. There is lot of data involved to train and validate the model, generalized machine learning algorithms will not work. Deep Learning has been evolved from the recent times to solve the data constraints on Machine Learning algorithms. GPU based computing is required to perform the Deep Learning tasks more effectively.

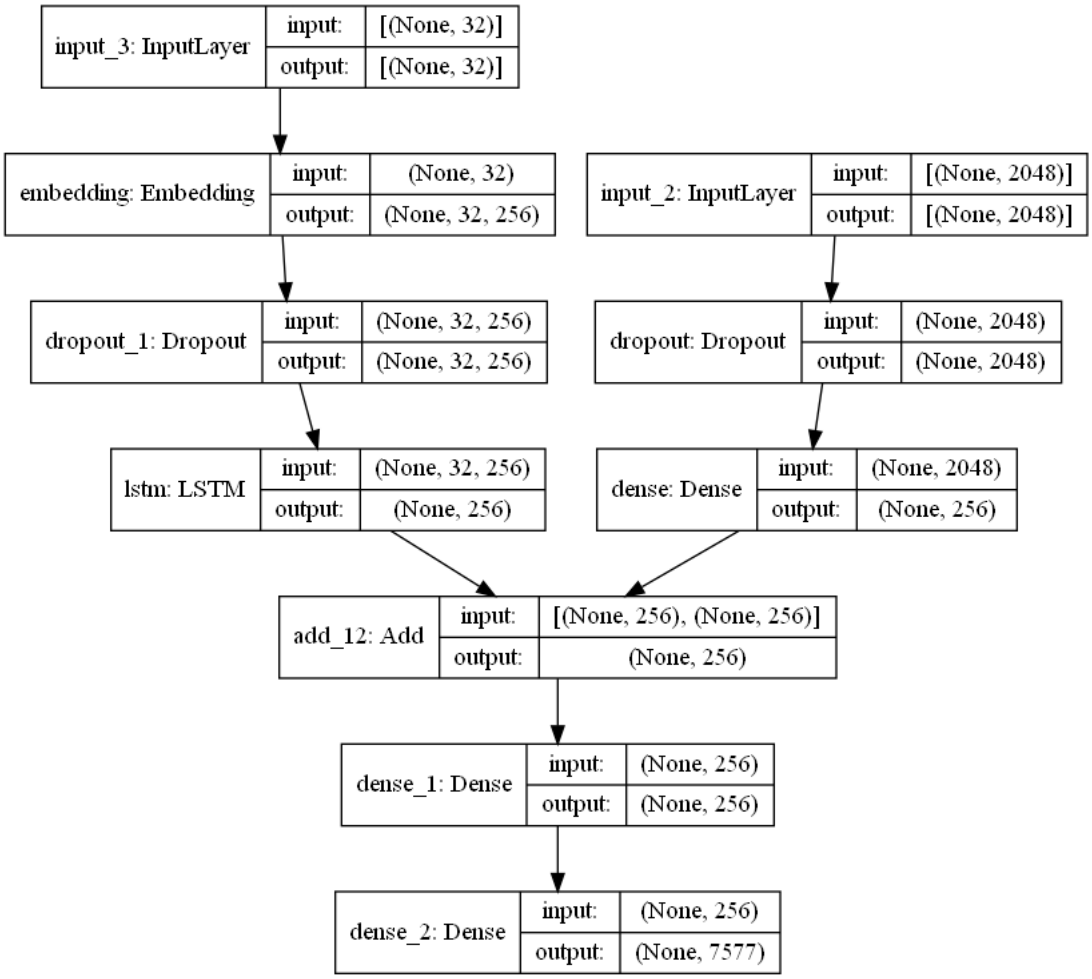


Figure 1.1: Architecture of proposed system

1.2.1 Objective

The objective of our project is to learn the concepts of a CNN and LSTM model and build a working model of Image caption generator by implementing CNN with LSTM.

In this Python project, we will be implementing the caption generator using CNN (Convolutional Neural Networks) and LSTM (Long short term memory). The image features will be extracted from Xception which is a CNN model trained on the imagenet dataset and then we feed the features into the LSTM model which will be responsible for generating the image captions.

Chapter 2

Literature Survey

Literature review is the comprehensive study and interpretation of literature that relates to a particular topic. When one uses literature review research questions are identified, then one seek to answer this research questions by searching for and analyzing relevant literature. Some importance of literature reviews is that new insights can be developed by the re-analyzing the results of the study. A literature review is both a summary and explanation of the complete and current state of knowledge on a topic as found in academic books and journal articles. There are two kinds of literature reviews you might write at university: one that students are asked to write as a stand-alone assignment in a course, and the other that is written as part of an introduction to, or preparation for, a longer work, usually a thesis or research report. The focus and perspective of your review and the kind of hypothesis or thesis argument you make will be determined by what kind of review you are writing. One way to understand the differences between these two types is to read published literature reviews or the first chapters of theses and dissertations in your own subject area. Analyses the structure of their arguments and note the way they address the issues.

2.1 Purpose of the Literature Review

1. It gives readers easy access to research on a particular topic by selecting high quality articles or studies that are relevant, meaningful, important and valid and summarizing them into one complete report.
2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarize, evaluate, and compare original research in that specific area.

3. It ensures that researchers do not duplicate work that has already been done.
4. It can provide clues as to where future research is heading or recommend areas on which to focus.
5. It highlights the key findings.
6. It identifies inconsistencies, gaps and contradictions in the literature.
7. It provides a constructive analysis of the methodologies and approaches of other researchers.

2.2 Literature Review of Image Captioning

Here, we take some of the papers related to Image Captioning using various methods,

In[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould.2016. Spice: Semantic propositional image caption evaluation. In EuropeanConference on Computer Vision. Springer, 382–398.In this paper, they introduce novel semantic evaluation metric that measures how effectively image captions recover objects, attributes and the relations between them. Their experiments demonstrate that, on natural image captioning datasets, SPICE captures human judgment over model-generated captions better than existing n-gram metrics such as Bleu, METEOR, ROUGE-L and CIDEr. Nevertheless, we are aware that significant challenges still remain in semantic parsing, and hope that the development of more powerful parsers will underpin further improvements to the metric.

In [3] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of the IEEE Conference on Com-puter Vision and Pattern Recognition. 5561–5570. In this paper, they reviewed deep learning-based image captioning methods and given a taxonomy of image captioning techniques, shown generic block diagram of the major groups and highlighted their properties.Dicussed different evaluation metrics and datasets withvtheir strengths and weaknesses. A brief summary of experimental results is also given.

In[5] Shuang Bai and Shan An. 2018. A Survey on Automatic Image CaptionGeneration. Neurocomputing. In this paper, they present a survey on image captioning. Based on the technique adopted in each method and classify image captioning approaches into different categories. Representative methods in each category are summarized, and strengths and limitations of each type of work are talked about and discuss early image captioning work which are mainly retrieval based and template based. Then, their main attention is focused on neural network based methods, which give state of the art results. Because different frameworks are used in neural network based methods, they further divided them into subcategories and discussed each subcategory, respectively. After that, state of the art methods are compared on benchmark datasets.

In[6] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator."Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conferenceon. IEEE, 2015. In this paper, they present a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation and that can be used to generate natural sentences describing an image. The model is trained to maximize the likelihood of the target description sentence given the training image. Experiments on several datasets show the accuracy of the model and the fluency of the language it learns solely from image descriptions. Their model is often quite accurate, which verify both qualitatively and quantitatively.

In[9],Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural net-works (m-rnn)." arXiv preprint arXiv:1412.6632 (2014). In this paper, they present a multimodal Recurrent Neural Network (m-RNN) model for generating novel image captions. It directly models the probability distribution of generating a word given previous words and an image. Image captions are generated by sampling from this distribution. The model consists of two sub-networks: a deep recurrent neural network for sentences and a deep convolutional network for images. These two sub-networks interact with each other in a multimodal layer to form the whole m-RNN model. The effectiveness of their model is validated on four benchmark datasets (IAPR TC-12, Flickr 8K, Flickr 30K and MS COCO). Our model outperforms the state-of-the-art methods. In addition, the m-RNN model can be applied to retrieval tasks for retrieving images or sentences, and achieves significant performance improvement over the state-of-the-art methods which directly

optimize the ranking objective function for retrieval.

In[13],Donahue, Jeffrey, et al. "Long-term recurrent convolutional networks forvisual recogni-tion and description." Proceedings of the IEEE conferenceon computer vision and pattern recognition. 2015. They presented LRCN, a class of models that is both spatially and temporally deep, and flexible enough to be applied to a variety of vision tasks involving sequential inputs and outputs. Their results consistently demonstrate that by learning sequential dynamics with a deep sequence model. The methods which take a fixed visual representation of the input and only learn the dynamics of the output sequence.

In[15], Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. 2015. Language mod-els for image captioning: The quirks and what works. arXiv preprintarXiv:1505.01809 (2015). In this paper, they argued that models should be capable of compositional generalization, i.e. the ability to produce captions that include combinations of unseen concepts.

In[24], Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. 2015.Guiding the long-short term memory model for image caption generation.In Proceedings of the IEEE International Conference on Computer Vision.2407–2415. In this paper, they Generating image descriptions in different languages is essential to satisfy users worldwide. However, it is prohibitively expensive to collect large-scale paired image-caption dataset for every target language which is critical for training descent image captioning models. Previous works tackle the unpaired cross-lingual image captioning problem through a pivot language, which is with the help of paired image-caption data in the pivot language and pivot-to-target machine translation models.

In[25],Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg,and Yejin Choi. 2012. Collective generation of natural image descriptions.In Proceedings of the 50th Annual Meeting of the Association for Compu-tational Linguistics: Long Papers-Volume 1. Association for ComputationalLinguistics, 359–368. In this paper they describe the collective generation of natural image descriptions. Their results consistently demonstrate that by learning sequential dynamics with

a deep sequence model. The methods which take a fixed visual representation of the input and only learn the dynamics of the output sequence.

Chapter 3

Methodology

Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe themin a natural language like English. It has emerged as a challenging and important research area following advances in statistical language modelling and imagerecognition. The generation of captions from images has various practical benefits, ranging from aiding the visually impaired, to enabling the automatic andcost-saving labelling of the millions of images uploaded to the Internet every day.The field also brings together state-of-the-art models in Natural Language Pro-cessing and Computer Vision, two of the major fields in Artificial Intelligence.There are two main approaches to Image Captioning: bottom-up and top-down.Bottom-up approaches, generate items observed in an image, and then attemptto combine the items identified into a caption. Top-down approaches, attempt togenerate a semantic representation of an image that is then decoded into a captionusing various architectures, such as recurrent neural networks. The latter approachfollows in the footsteps of recent advances in statistical machine translation, andthe state-of-the-art models mostly adopt the top-down approach.

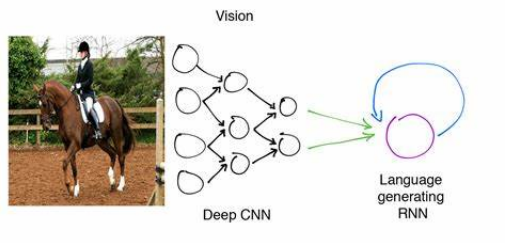


Figure 3.1: How it works

3.1 Software Requirement and Specification

The software used for the project:

- Python
- Anaconda

1.PYTHON

Python is an object-oriented programming language created by Guido Rossum in 1989. it's ideally designed for fast prototyping of complicated applications. it has interfaces to several OS system calls and libraries and is protractile to C or C++. several massive corporations use the Python programming language embody NASA, Google, YouTube, BitTorrent, etc. Python programming is widely utilized in AI, natural language Generation, Neural Networks and other advanced fields of computer science. Python is programming language open supply, high-level artificial language developed by Guido van Rossum within the late Eighties and presently administered by Python Software Foundation. It came from the ABC language that he helped produce early on in his career. Python is a powerful language that you can use develop games, write GUIs, and develop web applications. it's a high-level language. Reading and writing codes in Python is far like reading and writing regular English statements. As a result, they're not written in the machine-readable language, Python programs got to be processed before machines can run them. Python is an understood language. This implies that each time a program is run, its interpreter runs through the code and interprets it into machine-readable byte code. Python is an object-oriented language control users to manage and management data structures or objects to make and run programs. Everything in Python is, in fact, top-notch. All objects, data types, functions, methods, and classes take an equal position in Python. Programming languages are created to satisfy the requirements of programmers and users for an efficient tool to develop applications that impact lives, lifestyles, economy, and society. they assist build lives better by increasing productivity, enhancing communication, and rising potency. Languages die and become obsolete once they fail to live up to expectations and are replaced and superseded by languages that are more powerful. Python programming language artificial language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and

individual users. it's a living, thriving, and extremely helpful language that's extremely recommended as a primary programming language for those that want to dive into and experience programming.

2.ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS. Anaconda distribution comes with more than 1,500 packages as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator], as a graphical alternative to the command line interface (CLI).Big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow.

In some cases, the package may appear to work but produce different results in detail.In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, warning if this cannot be done.Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or your own private repository or mirror, using the conda install command. Anaconda Inc compiles and builds all the packages in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environ-

ment using pip, and conda will keep track of what it has installed itself and what pip has installed.Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud,PyPI or other repositories.The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

3.2 Architecture of CNN

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. CNNs are used for efficient image processing, artificial intelligence(AI) that use deep learning to perform both generative and descriptive tasks, often using machine vision that includes image and video recognition, along with recommender systems and natural language processing (NLP).A neural network is a system consist hardware and/or software patterned after the operation of neurons in the human brain. In image processing traditional neural networks are not ideal and must be fed images in reduced-resolution pieces. CNN are arranged with their “neurons” more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals.A CNN uses a system much like a multilayer perceptron that has been designed for reduced processing requirements .The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing.

- Convolutional Layer:This layer extracts features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as a filter or kernel and image matrix .

- Rectified Linear Unit (ReLU):After convolution operation, the output is subject to an activation function to allow non-linearity. The usual activation function for convnet is the Relu. The negative valued pixels will be replaced by zero.
- Softmax Regression:The Softmax regression is a form of logistic regression that normalizes an input value into a vector of values that follows a probability distribution whose total sums up to 1. The output values are between the range [0,1] which is nice because we are able to avoid binary classification and accommodate as many classes or dimensions in our neural network model. This is why softmax is sometimes referred to as a multinomial logistic regression [4]. The function is usually used to compute losses that can be expected when training a data set. Known use-cases of softmax regression are in discriminative model called Cross-Entropy.
- Dropout layer:The basic idea of the dropout layer is that the input elements with a certain probability are deactivated or dropped out such that the individual neurons can learn the features that are less dependent on their surroundings. This process takes place only during the training phase.
- All the neurons of this layer are connected to all the neurons in the previous layer, thereby combining all the features learned by the previous layer to facilitate classification. This layer produces an N-dimensional vector at t

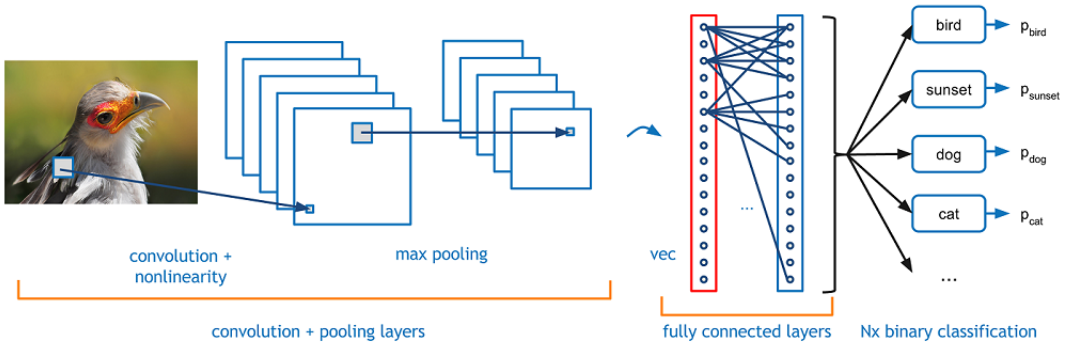


Figure 3.2: Architecture of cnn

3.3 LSTM Architecture

LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous test, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information.

The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another in a way to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got the only single neural net layer of tanh, LSTMs comprises of three logistic sigmoid gates and one tanh layer. Gates have been introduced in order to limit the information that is passed through the cell. They determine which part of the information will be needed by the next cell and which part is to be discarded. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'.

3.4 Overview

The project aim is to is to learn the concepts of a CNN and LSTM model and build a working model of Image caption generator by implementing CNN with LSTM. In this Python project, we will be implementing the caption generator using CNN (Convolutional Neural Networks) and LSTM (Long short term memory). The image features will be extracted from Xception which is a CNN model trained on the imagenet dataset and then we feed the features into the LSTM model which will be responsible for generating the image captions. For the image caption generator, we will be using the Flickr8K dataset. The Flickr8ktext folder contains file Flickr8k.token which is the main file of our dataset that contains image name and their respective captions separated by newline.

The process of sentence generation in neural network is taken from principle of encoder decoder in modelling of network and machine translation. Convolutional Neural networks are specialized deep neural networks which can process the data

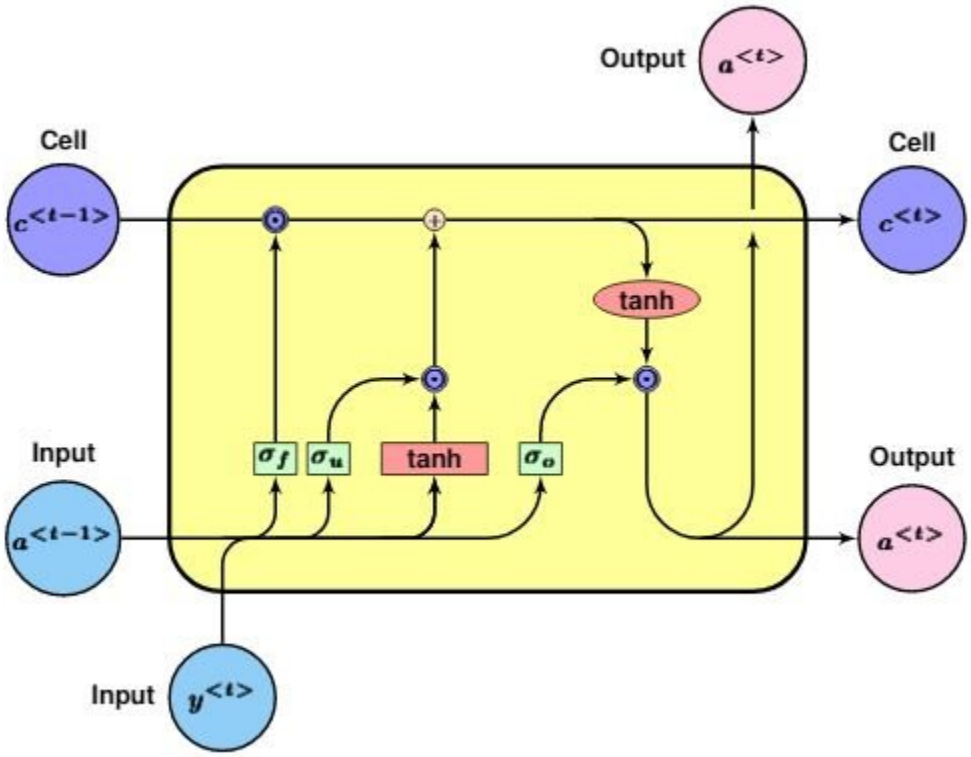


Figure 3.3: Cell structure of LSTM

that has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images. CNN is basically used for image classifications and identifying if an image is a bird, a plane or Superman, etc. It scans images from left to right and top to bottom to pull out important features from the image and combines the feature to classify images. It can handle the images that have translated, rotated, scaled and changes in perspective.

The first step is to import all necessary packages, then pefrom data cleaning operation. For that first we need to load the file Flickr8k.token in our Flickr8ktext folder. The format of our file is image and caption separated by a new line. Each image has 5 captions and we can see that (0 to 4) number is assigned for each caption.For data cleaning we need to define five functions:

1.loaddoc(filename) – For loading the document file and reading the contents inside the file into a string.

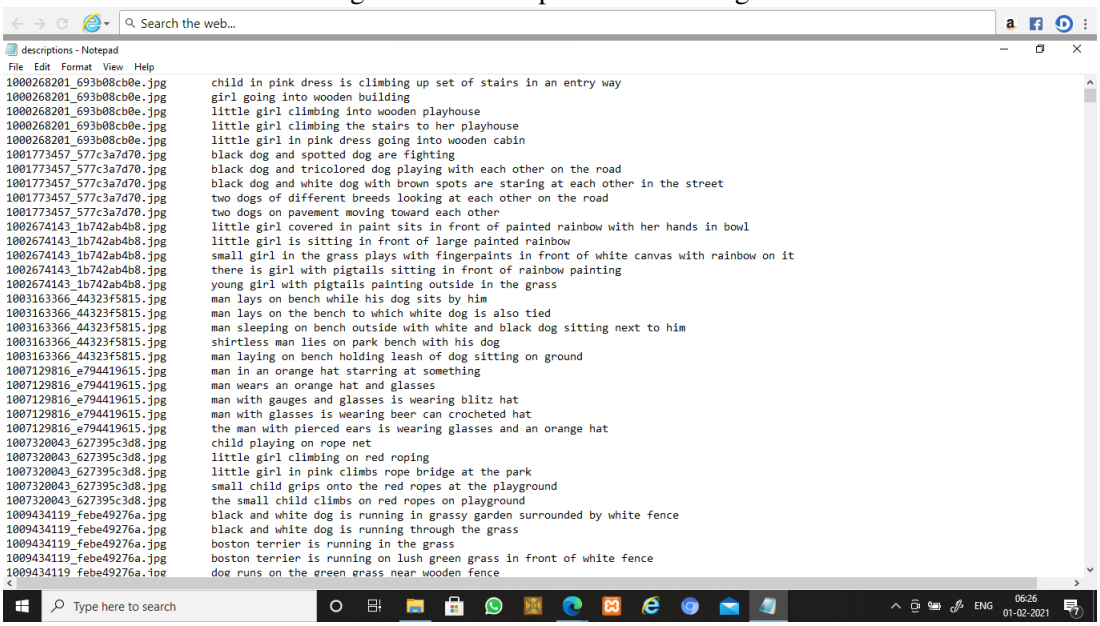
2.allimgcaptions(filename) – This function will create a descriptions dictionary that maps images with a list of 5 captions.

3.cleaningtext(descriptions) – This function takes all descriptions and performs data cleaning. This is an important step when we work with textual data, according to our goal, we decide what type of cleaning we want to perform on the text. In our case, we will be removing punctuations, converting all text to lowercase and removing words that contain numbers. So, a caption like “A man riding on a three-wheeled wheelchair” will be transformed into “man riding on three wheeled wheelchair”.

4.textvocabulary(descriptions) – This is a simple function that will separate all the unique words and create the vocabulary from all the descriptions.

5.savedescriptions(descriptions, filename) – This function will create a list of all the descriptions that have been preprocessed and store them into a file. We will create a descriptions.txt file to store all the captions. It will look something like this:

Figure 3.4: Descriptions of the images



Next is to extract all features from the images. This technique is also called transfer learning, we don't have to do everything on our own, we use the pre-trained model that have been already trained on large datasets and extract the features from these models and use them for our tasks. We are using the Xception model which has been trained on imagenet dataset that had 1000 different classes to classify. We can directly import this model from the keras.applications. Make sure you are connected to the internet as the weights get automatically downloaded. Since the Xception model was originally built for imagenet, we will do little changes for integrating with our model. One thing to notice is that the Xception model takes 299*299*3 image size as input. We will remove the last classification layer and get the 2048 feature vector.

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

The function extractfeatures() will extract features for all images and we will map image names with their respective feature array. Then we will dump the features dictionary into a "features.p" pickle file.

Next is load the image dataset for training the model. In our Flickr8ktest folder, we have Flickr8k.trainImages.txt file that contains a list of 6000 image names that we will use for training. For loading the training dataset, we need more functions:

1. loadphotos(filename) – This will load the text file in a string and will return the list of image names.

2. loadcleandescriptions(filename, photos) – This function will create a dictionary that contains captions for each photo from the list of photos. We also append the ¡start¿ and ¡end¿ identifier for each caption. We need this so that our LSTM model can identify the starting and ending of the caption.

3. loadfeatures(photos) – This function will give us the dictionary for image names and their feature vector which we have previously extracted from the Xception model.

Then we need to tokenize the vocabulary, because computers don't understand English words, for computers, we will have to represent them with numbers. So,

we will map each word of the vocabulary with a unique index value. Keras library provides us with the tokenizer function that we will use to create tokens from our vocabulary and save them to a "tokenizer.p" pickle file. After the vocabulary tokenizing we need to create data generator in this section we define or declare how exactly the output looks like. Next step is to train the model, To train the model, we will be using the 6000 training images by generating the input and output sequences in batches and fitting them to the model using model.fitgenerator() method. We also save the model to our models folder. This will take some time depending on your system capability. The final step is to testing the trained model, after the model has been trained, now, we will make a separate file testingcaptiongenerator.py which will load the model and generate predictions. The predictions contain the max length of index values so we will use the same tokenizer.p pickle file to get the words from their index values.

Chapter 4

Result And Discussion

Testing is the major quality measures employed during the software development. After the coding phase, computer programs available are executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

- Testing is a process of executing a program with the intention of finding an error.
- A good test case is one that has a highest probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

Our objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort. Testing demonstrate that software functions appear to be working according to specification, that performance requirements appears to have been met. Data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But there is one thing that testing cannot do: Testing cannot show the absence of defects it can only show that software defects as present.

4.1 Testing Methods

There are different types of testing methods available.

- **Unit Testing**

  In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as "module" testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In these testing steps each module is found to work satisfactory as regarding to the expected output from the module. There are some validation30 checks for verifying the data input given by the user. It is very easy to find errors and debug the system. In this project, after coding each module have been individually tested to determine whether they are coded correctly so that they satisfy the requirements in the specifications and execute effectively as individual units was tested and run individually.

- **Integration Testing**

  Data can be lost across an interface. One module can have an adverse effect on the other sub functions when combined may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performance. According to this project, using the integrated test plan prepared in the design phase of the system developed as a guide, the integration test was carried out. All the errors found in the system were corrected for the next testing steps.

- **Dry run (testing)**

  A dry run (or a practice run) is a testing process where the effects of a possible failure are intentionally mitigated. Dry run testing is a static test and should be performed by the developer31 to mitigate the effects of a failure of the product - meaning before the end user gets the product and discovers it doesn't do what it says it will. In dry run testing, no hardware is used, but it is assumed that the programmer who is testing the code is aware of what each line of code is supposed to do and gives him or her the opportunity to make corrections to the code before it becomes an issue for the actual software. Basically, a dry run test consists of programmers manually reading their code line by line to find errors and fix them.

4.2 Test Plan

A test plan is a systematic approach to test a system. The plan typically contains a detailed understanding of what the eventual workflow will be. Normally testing of any large system will be in two parts.

- The functional verification and validation against the requirement specification
- Performance evaluation against the indicated requirements

Testing activity is involved right from the beginning of the project. At the very first stage of testing, the goals and objectives are set. This simplifies the limits or boarders of testing process.

Before testing, the tester should plan what kind of data he is giving for test. Give data inputs as functional, boundary, stress, performance, usability values etc.

Characteristics of a Good Test:

- Tests are likely to catch bugs
- No redundancy
- Not too simple or too complex

Test Cases

A specific set of steps and data along with expected results of a particular test objective. A test case should only test one limited subset of a feature or functionality. Test case documents for each functionality/testing area will be written, reviewed and maintained separately in excel sheets. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check our error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if error test cases are not yet written, it is OK for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any, is expected to trigger errors.

Implementation

Implementation is the process of having the system personnel check out and put new equipment to use, train the users to use the new system and construct any file that are needed to see it. The final and impartment phases in the system life cycle are the implementation of the new system. System implementation refers to the steps necessary to install a new system to put into operation. The implementation has different meaning, ranging from the conversion of a basic application to complete replacement of computer system. Implementation includes all these activities that take place to convert from old system to new one. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The methods of implementation and time scale adopted are found out initially. The system is tested properly and at the same time the users are trained in the new procedure. Proper implementation is essential to provide a reliable system to meet organizational requirements. Successful implementations may not guarantee improvement in the organization involves the following things:

- Careful planning
- Investigation of the system and constraint
- Design the methods to achieve the change over
- Train the staff in the changed phase
- Evaluation of change over method Implementation methods

There are several methods for handling the implementation and consequent conversation from the old to new automated system. The most secure for this conversation is to run the old and new system in parallel. This method offers high security but the cost for maintaining the two systems in parallel is very high. Another method is direct cut over the existing system to automated system. The chance may take place within a week or within a day.

Implementation Phase It includes a description of all activities that most occur to implement the new system and put into operation. It consists of the following steps:

- List all files required for the implementation

- Identify all data required to build new files during the implementation
- List all new document and procedure that go to the new system

4.3 Dataset

Dataset: For the image caption generator, we will be using the Flickr8K dataset. It contains Flicker8kDataset – Dataset folder which contains 8091 images and Flickr8ktext – Dataset folder which contains text files and captions of images. The Flickr 8K dataset contains 8,092 images annotated with five human-generated reference captions each. The images were manually selected to focus mainly on people and animals performing actions. The dataset also contains graded human quality scores for 5,822 captions, with scores ranging from 1 ('the selected caption is unrelated to the image') to 4 ('the selected caption describes the image without any errors'). Each caption was scored by three expert human evaluators sourced from a pool of native speakers. All evaluated captions were sourced from the dataset, but association to images was performed using an image retrieval system. In our evaluation we exclude 158 correct image-caption pairs where the candidate caption appears in the reference set. This reduces all correlation scores but does not disproportionately impact any metric.

Figure 4.1: Flickr8k image Dataset

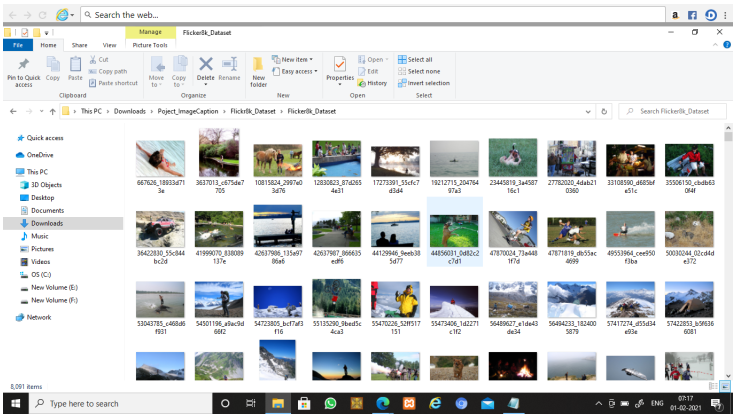


Figure 4.2: Flickr8k text Dataset

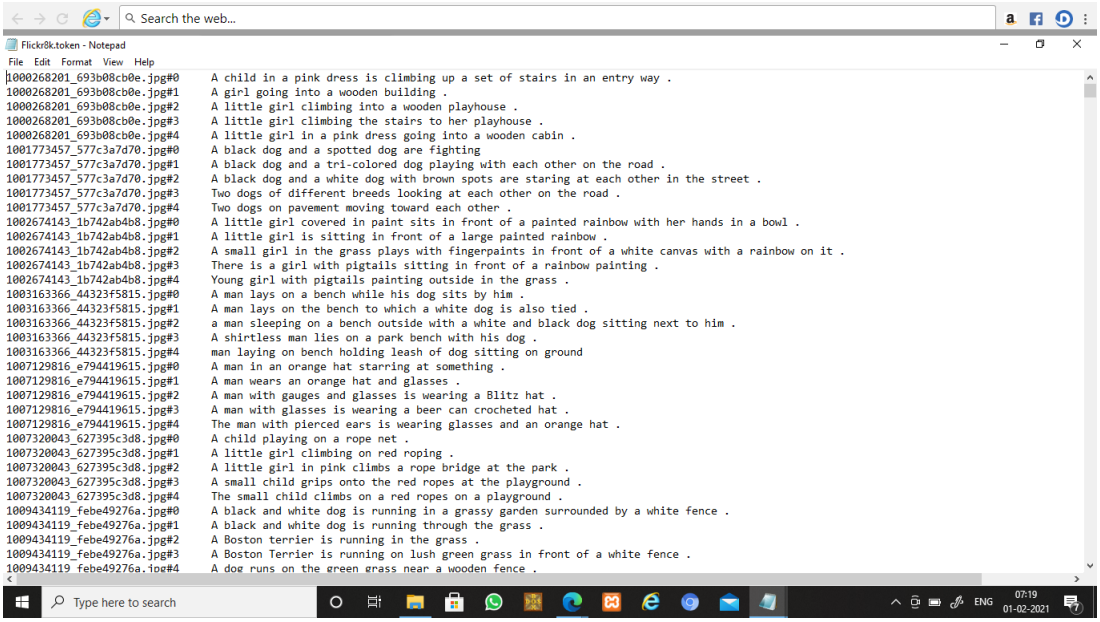


Figure 4.3: Output screen 1

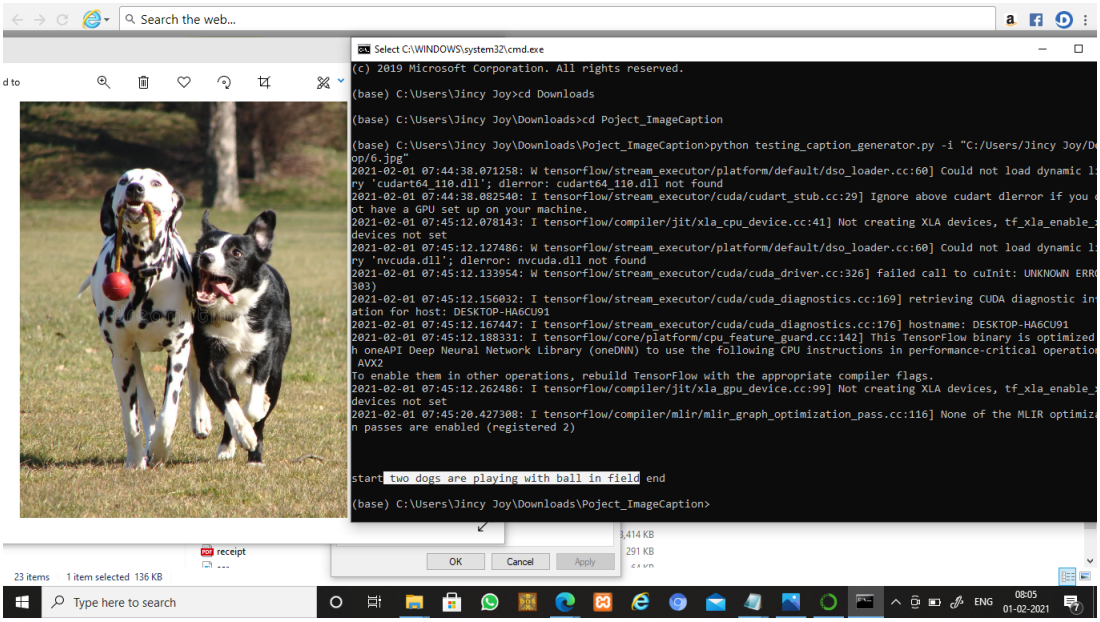
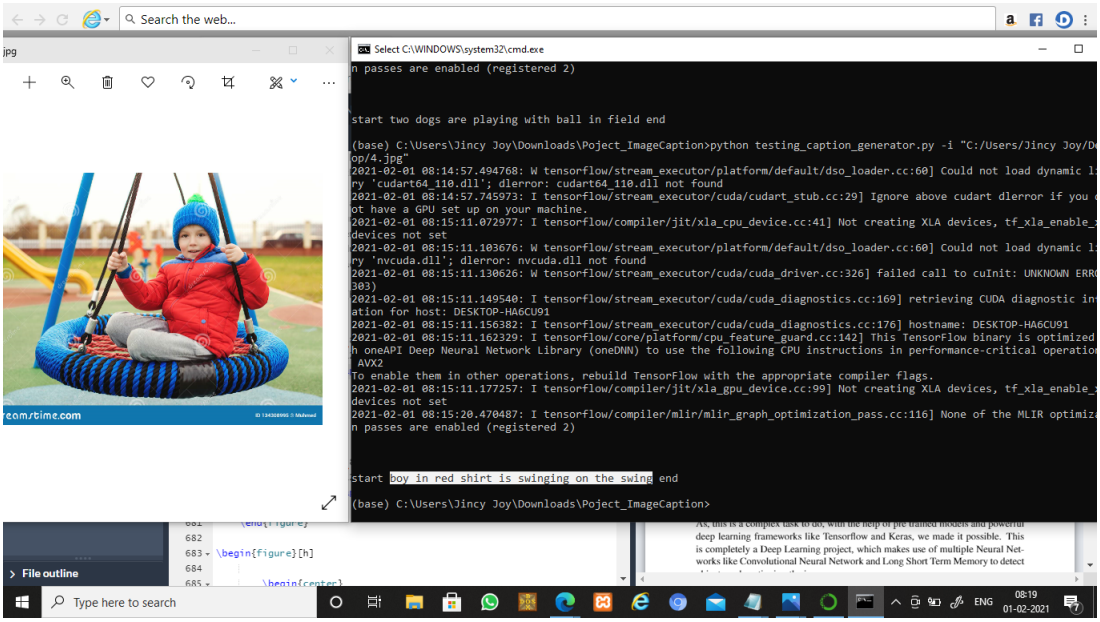


Figure 4.4: Output screen 2



Chapter 5

Conclusion

Image captioning has many advantages in almost every complex area of Artificial Intelligence. The main use case of our model is to help visually impaired to understand the environment and made them easy to act according to the environment. As, this is a complex task to do, with the help of pre trained models and powerful deep learning frameworks like Tensorflow and Keras, we made it possible. This is completely a Deep Learning project, which makes use of multiple Neural Networks like Convolutional Neural Network and Long Short Term Memory to detect objects and captioning the images.

5.1 Advantages

There are various advantages of Image captioning in multiple disciplines.

- It can be used for visually impaired people to understand the environment.
- It can be used in areas where text is more used and it can be used to infer text from images.
- It can be used by social networks to describe the image being uploaded by the user.
- It can be used in various NLP applications, where insights and summary is needed from the images

5.2 Future Enhancement

In future We extend our work in the next higher level by enhancing our model to generate captions even for the live video frame. Our present model generates captions only for the image, which itself a complex task and captioning live video frames is much complex to create. This is completely GPU based and captioning live video frames cannot be possible with the general GPUs. Video captioning is a popular research area in which it is going to change the lifestyle of the people with the use cases being widely usable in almost every domain. It automates the major tasks like video surveillance and other security tasks.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In European Conference on Computer Vision. Springer, 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 (2017).

[3] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5561–5570

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR).

[5] Shuang Bai and Shan An. 2018. A Survey on Automatic Image Caption Generation. Neurocomputing.

[6] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015.

[7] Yao, Benjamin Z., et al. "I2t: Image parsing to text description." Proceedings of the IEEE 98.8 (2010): 1485-1508.

[8] Jia, Xu, et al. "Guiding long-short term memory for image caption generation." arXiv pre-print arXiv:1509.04942 (2015).

[9] Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." arXiv preprint arXiv:1412.6632 (2014).

[10] Karpathy, Andrej, and Li Fei-Fei. ”Deep visual-semantic alignments for generating image descriptions.” Proceedings of the IEEE conference on computer vision and pattern recog-nition. 2015.

[11] Xu, Kelvin, et al. ”Show, attend and tell: Neural image caption generation with visual at-tention.” International Conference on Machine Learning. 2015.

[12] El Housseini, Ali, Abdelmalek Toumi, and Ali Khenchaf. ”Deep Learning for target recognition from SAR images.” Detection Systems Architectures and Technologies (DAT), Seminar on. IEEE, 2017.

[13] Donahue, Jeffrey, et al. ”Long-term recurrent convolutional networks for visual recogni-tion and description.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[14] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. 2017. Towards Diverse and Natural Image Descriptions via a Conditional GAN. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2989–2998.

[15] Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. 2015. Language models for image captioning: The quirks and what works. arXiv preprint arXiv:1505.01809 (2015).

[16] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2625–2634.

[17] Desmond Elliott and Frank Keller. 2013. Image description using visual dependency representations. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 1292–1302.

[18] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr DollÃar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, and John C Platt. 2015. From captions to visual concepts and back. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1473–1482

[19] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3137–3146.

[20] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 1141–1150.

[21] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. 2017. An empirical study of language cnn for image captioning. In Proceedings of the International Conference on Computer Vision (ICCV). 1231–1240.

[22] Yahong Han and Guang Li. 2015. Describing images with hierarchical concepts and object class localization. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. ACM, 251–258.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (CVPR). 5967–5976.

[24] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. 2015. Guiding the long-short term memory model for image caption generation. In Proceedings of the IEEE International Conference on Computer Vision. 2407–2415.

[25] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. 2012. Collective generation of natural image descriptions. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 359–368.