School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled Image Caption Generator using CNN and LSTM is prepared and submitted by Swetha B & Shreya Alajangi (Reg. No. 20MIA1101 & 20MIA1172) to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of Master of Technology in Software Engineering (5 year Integrated Programme) and as part of SWE1017 – Natural Language Processing Project is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission.

Guide/Supervisor	HoD
Name: Krithiga R	Name: Dr. Sivabalakrishnan M
Date:	Date:

(Seal of SCOPE)

Abstract

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

KEYWORDS: generate captions, deep learning techniques, concepts of image captioning

INTRODUCTION:

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

Image captioning has various applications in various fields such as biomedicine, commerce, web searching and military etc. Social media like Instagram , Facebook etc can generate captions automatically from images.

MOTIVATION

Generating captions for images is a vital task relevant to the area of both Computer Vision and Natural Language Processing. Mimicking the human ability of providing descriptions for imagesby a machine is itself a remarkable step along the line of Artificial Intelligence. The main challenge of this task is to capture how objects relate to each other in the image and to express them in a natural language (like English).Traditionally, computer systems have been using pre- defined templates for generating text descriptions for images. However, 1 this approach does not provide sufficient variety required for generating lexically rich text descriptions. This shortcoming has been

suppressed with the increased efficiency of neural networks. Many state of art models use neural networks for generating captions by taking image as input and predicting next lexical unit in the output sentence.

LITERATURE REVIEW

Image captioning has recently gathered a lot of attention specifically in the natural language domain. There is a pressing need for context based natural language description of images, however, this may seem a bit farfetched but recent developments in fields like neural networks, computer vision and natural language processing has paved a way for accurately describing images i.e. representing their visually grounded meaning. We are leveraging state-of-the-art techniques like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and appropriate datasets of images and their human perceived description to achieve the same. We demonstrate that our alignment model produces results in retrieval experiments on datasets such as Flicker.

IMAGE CAPTIONING METHODS

There are various Image Captioning Techniques some are rarely used in present but it is necessary to take a overview of those technologies before proceeding ahead. The main categoriesof existing image captioning methods and they include template-based image captioning, retrieval-based image captioning, and novel caption generation.Novel caption generation-based image caption methods mostly use visual space and deep machine learning based techniques. Captions can also be generated from multimodal space. Deep learning-based image captioning methods can also be categorized on learning techniques: Supervised learning, Reinforcement learning, and Unsupervised learning. We group the reinforcement learning and unsupervised learning into Other Deep Learning. Usually captions are generated for a whole scene in the image. However, captions can also be generated for different regions of an image (Dense captioning). Image captioning methods can use either simple Encoder-Decoder architecture or

Compositional architecture. There are methods that use attention mechanism, semantic concept, and different styles in image descriptions. Some methods can also generate description for unseen objects. We group them into one category as “Others". Most of the image captioning methods use LSTM as language model. However, there are a number of methods that use other language models such as CNN and RNN. Therefore, we include a language model-based category as “LSTM vs. Others".

PROPOSED WORK

The main aim of this project is to get a little bit of knowledge of deep learning techniques. Weuse two techniques mainly CNN and LSTM for image classification. So, to make our image caption generator model, we will be merging these architectures. It is alsocalled a CNN-RNN model.

- CNN is used for extracting features from the image. We will use the pre-trained modelXception.
- LSTM will use the information from CNN to help generate a description of the image.

CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. Convolutional Neural networks are specialized deep neural networks which can process the datathat has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images. 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 been translated, rotated, scaled and changes in perspective.

LONG SHORT TERM MEMORY

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 text, 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. LSTMs are designed to overcome the vanishing gradient problem and allow them to retain information for longer periods compared to traditional RNNs. LSTMs can maintain a constant error, which allows them to continue learning over numerous time-steps and backpropagate through time and layers.

SYSTEM DESIGN

This project requires a dataset which have both images and their caption. The dataset should be able to train the image captioning model.

FLICKR8K DATASET

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

- Diverse category of training images can make the image captioning model to work for multiple categories of images and hence can make the model more robust.

IMAGE DATA PREPARATION

The image should be converted to suitable features so that they can be trained

into a deeplearning model. Feature extraction is a mandatory step to train any image in deep learning model. The features are extracted using Convolutional Neural Network (CNN) with Visual Geometry Group (VGG-16) model. This model also won ImageNet Large Scale Visual Recognition Challenge in 2015 to classify the images into one among the 1000 classes given in the challenge. Hence, this model is ideal to use for this project as image captioning requires identification of images. In VGG-16, there are 16 weight layers in the network and the deeper number of layers help in better feature extraction from images. The VGG-16 network uses 3*3 convolutional layers making its architecture simple and uses max pooling layer in between to reduce volume size of the image. The last layer of the image which predicts the classification is removed and the internal representation of image just before classification is returned as feature. The dimension ofthe input image should be 224*224 and this model extracts features of the image and returns a 1- dimensional 4096 element vector

CAPTION DATA PREPARATION

Flickr8k dataset contains multiple descriptions described for a single image. In the data preparation phase, each image id is taken as key and its corresponding captions are stored as values in a dictionary.

DATA CLEANING

In order to make the text dataset work in machine learning or deep learning models, raw text should be converted to a usable format. The following text cleaning steps are done before using it for the project: • Removal of punctuations. • Removal of numbers. • Removal of single length words. • Conversion of uppercase to lowercase characters. Stop words are not removed from the text data as it will hinder the generation of a grammatically complete caption which is needed forthis project. Table 1 shows samples of captions after data cleaning.

In [22]:

```
import string
text_original = "I ate 8 burgers and 4 pizzas. it's 9:44 am. can you play chess with me"

print("Origial sentense: ",text_original)
print("\nRemoving Punctuations...")

# creating a function that removes punctuation in the sentences
def remove_punctuation(text_original):
    text_without_punct = text_original.translate(str.maketrans('','',string.punctuation))
    return text_without_punct

text_without_punct = remove_punctuation(text_original)
print(text_without_punct)

print("\nRemoving a single character...")

# creating a function that removes single character
def removing_single_char(text):
    text_len_greater_than_one = ""
    for word in text.split():
        if len(word) > 1:
            text_len_greater_than_one += " " + word
    return text_len_greater_than_one

text_len_greater_than_one = removing_single_char(text_without_punct)
print(text_len_greater_than_one)
```

```
# creating a function that removes numerical values
def remove_numeric(text, printTF=False):
    text_without_num = ""
    for word in text.split():
        isalpha = word.isalpha()
        if printTF:
            print("     {:10} : {:}".format(word, isalpha))
        if isalpha:
            text_without_num += " "+ word
    return text_without_num

text_without_num = remove_numeric(text_len_greater_than_one, printTF=True)
print(text_without_num)
```

```
Origial sentense:  I ate 8 burgers and 4 pizzas. it's 9:44 am. can you play chess with me

Removing Punctuations...
I ate 8 burgers and 4 pizzas its 944 am can you play chess with me

Removing a single character...
 ate burgers and pizzas its 944 am can you play chess with me

Removing numeric values...
     ate        : True
     burgers    : True
     and        : True
     pizzas     : True
     its        : True
     944        : False
     am         : True
```

PROJECT FILE STRUCTURE

Downloaded from dataset:

- **Flicker8k_Dataset –** Dataset folder which contains 8091 images.
- **Flickr_8k_text –** Dataset folder which contains text files and captions

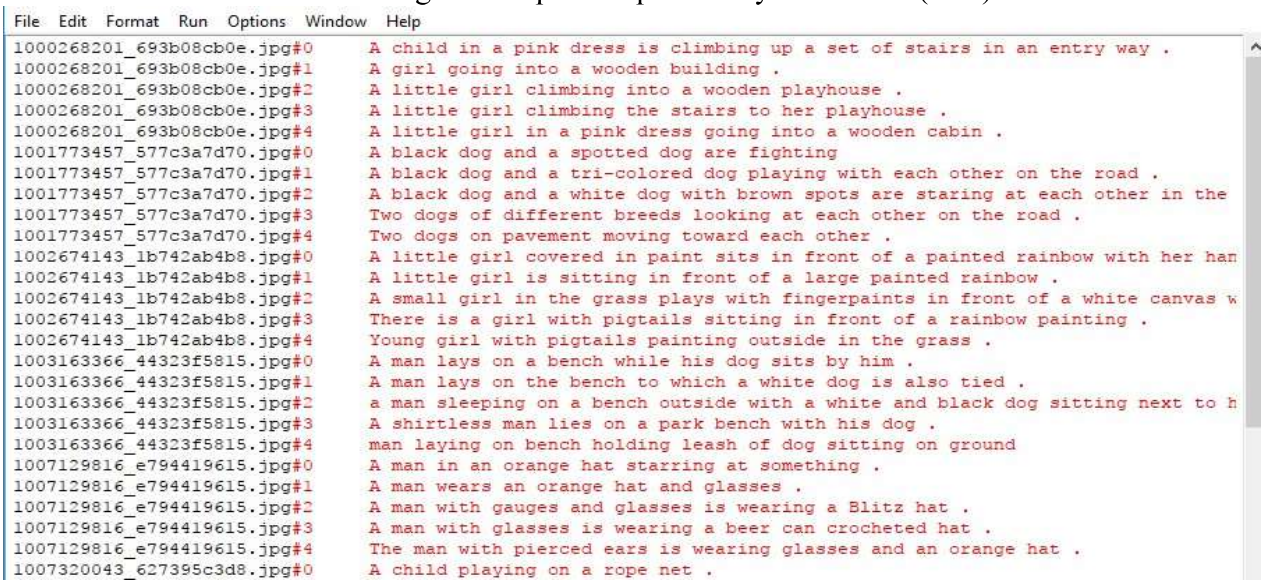
of images.The below files will be created by us while making the project.

- **Models –** It will contain our trained models.
- **Descriptions.txt –** This text file contains all image names and their captions afterpreprocessing.
- **Features.p –** Pickle object that contains an image and their feature vector extracted from theXception pre-trained CNN model.
- **Tokenizer.p –** Contains tokens mapped with an index value.
- **Model.png –** Visual representation of dimensions of our project.
- **Testing_caption_generator.py –** Python file for generating a caption of any image.
- **Training_caption_generator.ipynb –** Jupyter notebook in which we train and build ourimage caption generator.

GETTING AND PERFORMING DATA CLEANING

The main text file which contains all image captions is **Flickr8k.token** in our **Flickr_8k_text** folder.

The format of our file is image and caption separated by a new line ("\n").

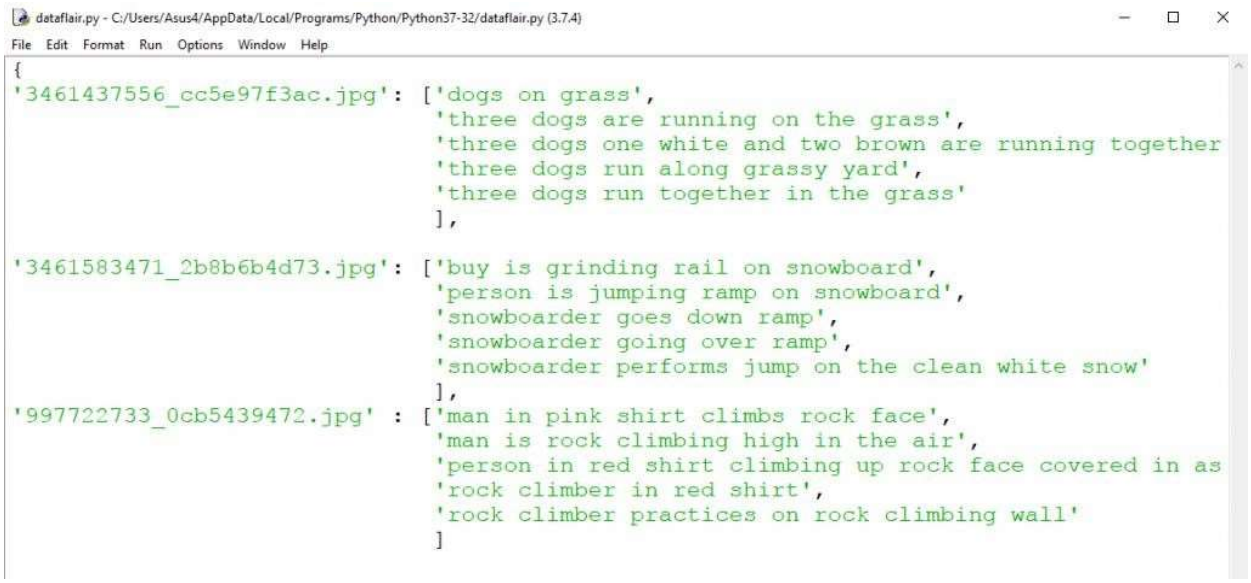


Each image has 5 captions and we can see that #(0 to 5)number is assigned for each caption.We will define 5 functions:

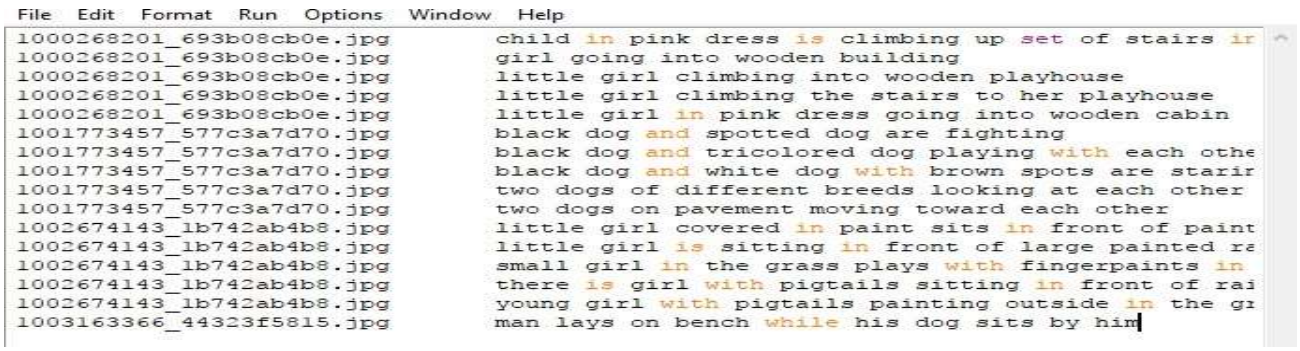
- **load_doc(filename) –** For loading the document file and reading the contents

inside the file into a string.

- **all_img_captions(filename) –** This function will create a **descriptions** dictionary that maps images with a list of 5 captions. The descriptions dictionary will look something like the Figure.



- **cleaning_text(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”
- **text_vocabulary(descriptions) –** This is a simple function that will separate all the uniquewords and create the vocabulary from all the descriptions.
- **save_descriptions(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:



EXTRACTING THE FEATURE VECTOR FROM ALL 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 featuresfrom 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(include_top=False, pooling=‟avg‟)

The function **extract_features()** will extract features for all images and we will map image names with their respective feature array.

TOKENIZING THE VOCABULARY

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 andsave them to a **“tokenizer.p”** pickle file.

Our vocabulary contains 7577 words. We calculate the maximum length of the descriptions. This is important for deciding the model structure parameters.

Max_length of description is 32.

Training 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.fit_generator() method. We also save the model to our models folder.

```
plt.show()
```

man on a bicycle riding on only one wheel .

asian man in orange hat is popping a wheelie on his bike .

a man on a bicycle is on only the back wheel .

a man is doing a wheelie on a mountain bike .

a man does a wheelie on his bicycle on the sidewalk .

five people are sitting together in the snow .

five children getting ready to sled .

a group of people sit in the snow overlooking a mountain scene .

a group of people sit atop a snowy mountain .

a group is sitting around a snowy crevasse .

a water bird standing at the ocean 's edge .

a tall bird is standing on the sand beside the ocean .

a large bird stands in the water on the beach .

a grey bird stands majestically on a beach while waves roll in .

woman writing on a pad in room with gold , decorated walls .

the walls are covered in gold and patterns .

a woman standing near a decorated wall writes .

a woman behind a scrolled wall is writing

a person stands near golden walls .

a rock climber practices on a rock climbing wall .

Testing the model

The model has been trained, now, we will make a separate file testing_caption_generator.py which will load the model and generate predictions. Thepredictions 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.

startseq black and white dog is running in the gr

startseq girl in blue shirt is standing on the wate

startseq black dog is running through the water e

startseq girl in blue shirt is standing on the beach

CONCLUSION

In this report, we have reviewed deep learning-based image captioning methods. We have given taxonomy of image captioning techniques. Although deep learning-based image captioning methods have achieved a remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for some time.

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