Synopsis Report

on

AI Image Captioning Bot

Submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

Session 2022-23 in

CSE-Data Science

By:

Harshit Kumar Rai

Aditya Agrawal

Abhishek Pandey

Under the guidance of:

Mr. Prabhat Singh
Assistant Professor

DEPARTMENT OF CSE-DS

ABES ENGINEERING COLLEGE, GHAZIABAD

AFFILIATED TO

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW

(Formerly UPTU)

Student's Declaration

I / we hereby declare that the work being presented in this report entitled **"AI Image Caption Bot** is an authentic record of my/ our own work carried out under the supervision of **Mr. Prabhat Singh, Assistant Professor, CSE-DS.** The matter embodied in this report has not been submitted by us for the award of any other degree.

Date:

Signature of student	**Signature of student**	**Signature of student**
(Name: Harshit Kumar Rai)	**(Name: Aditya Agrawal)**	**(Name: Abhishek Pandey)**
(Roll No.2000321540028)	**(Roll No.2000321540004)**	**(Roll No.2000321540003)**

Department: CSE – Data Science

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Signature of HOD	**Signature of Supervisor**
……	**Mr. Prabhat Singh**
CSE-DS	**Assistant Professor**
Date:	**CSE-DS**

Acknowledgement

We would like to convey our sincere thanks to **Mr. Prabhat Singh** for giving the motivation, knowledge and support throughout the course of the project. The continuous support helps in a successful completion of project. The knowledge provided is very useful for us.

We also like to give a special thanks to the department of CSE – Data Science for giving us the continuous support and opportunities for fulfilling our project.

We would also like to extend our sincere obligation to **Mr. Prabhat Singh, Head of Department, CSE-DS** for providing this opportunity to us.

Signature of student	**Signature of student**	**Signature of student**
(Name: Harshit Kumar Rai)	**(Name: Aditya Agrawal)**	**(Name: Abhishek Pandey)**
(Roll No.2000321540028)	**(Roll No.2000321540004)**	**(Roll No.2000321540003)**

Table of Contents

ABSTRACT

We are extremely intrigued by how machines can consequently portray the substance of pictures utilizing human language. To acquire a more profound understanding of this PC vision subject, we chose to execute present status of-the-craftsmanship picture subtitle generator Show, join in and tell: Brain picture subtitle generator with visual consideration Our brain network based picture subtitle generator is carried out in Python controlled by TensorFlow AI library.

In this undertaking, we use CNN and LSTM to produce the subtitle of the picture. As the profound learning strategies are developing, enormous datasets and PC power are useful to construct models that can produce inscriptions for a picture. This is the very thing we will carry out in this Python based project where we will utilize profound learning strategies like CNN and RNN. Picture subtitle generator is a cycle which includes regular language handling and PC vision ideas to perceive the setting of a picture and present it in English. In this overview paper, we cautiously follow a portion of the center ideas of picture subtitling and its generally expected approaches. We talk about Keras library, NumPy and Jupiter journals for the creation of this venture. We likewise examine about Flickr dataset and CNN utilized for picture characterization.

Chapter 1

Introduction

Consistently, we experience countless pictures from different sources like the net, news stories, record graphs and promotions. These sources contain pictures that watchers would need to decipher themselves. Most pictures don't have a depiction, however the human can to a great extent figure out them without their definite subtitles. Nonetheless, machines need to decipher some assortment of picture inscriptions in the event that people need programmed picture subtitles from it. Picture inscribing is vital in light of multiple factors. Subtitles for each picture on the web can bring about quicker and spellbindingly precise pictures searches and ordering.

Consequently, producing subtitles to a picture shows the comprehension of the picture by PCs, which is a major errand of knowledge. For a subtitle model it not just has to find which articles are contained in the picture and furthermore should have the option to communicating their connections in a characteristic language like English. As of late work likewise accomplish the presence of consideration, which can store and report the data and connection between a few most notable highlights and groups in the picture. In Xu's work, it depict ways to deal with subtitle age that endeavor to consolidate a type of consideration with two variations: a "hard" consideration component and a "delicate" consideration system.

The objective of this undertaking is to create fitting subtitles for a given picture. The subtitles will be produced to catch the context-oriented data on the pictures. Current strategies utilize convolutional brain organizations (CNNs) and intermittent brain organizations (RNNs) or their variations to produce proper inscriptions. These organizations give an encoder-decoder technique to do this errand, where CNNs encode the picture into highlight vectors and RNNs are utilized as decoders to create language portrayals.

Picture subtitling finds different applications in different fields, for example, trade, biomedicine, web looking and military and so on. Virtual entertainment like Instagram, Facebook and so forth can create inscriptions consequently from pictures

Chapter 2

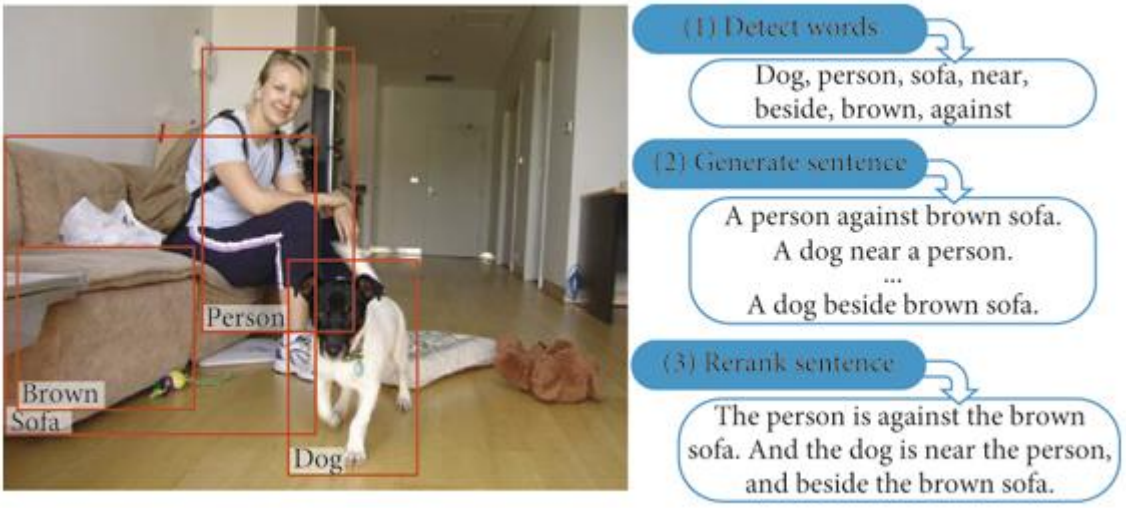
Related Work

The related work associated with our project is given below:

1.1. Existing Approaches

Handcraft Elements with Measurable Language Model

This strategy is a Midge framework in light of most extreme probability assessment, which straightforwardly gains the visual locator and language model from the picture portrayal dataset, as displayed in Figure 1.first break down the picture, distinguish the item, and afterward create an inscription. Words are identified by applying a convolutional brain organization (CNN) to the picture region and coordinating the data with MIL. The design of the sentence is then prepared straightforwardly from the inscription to limit the priori suspicions about the sentence structure. At long last, it transforms a picture inscription age issue into an enhancement issue and looks for the most probable sentence.



Deep Learning Features with Neural Network

The recurrent neural network (RNN) has attracted a lot of attention in the field of deep learning. It was originally widely used in the field of natural language processing and achieved good results in language modelling. In the field of speech, RNN converts text and speech to each other, machine translation, question and answer session, and so on.

Of course, they are also used as powerful language models at the level of characters and words. Currently, word-level models seem to be better than character-level models, but this is certainly temporary. RNN is also rapidly gaining popularity in computer vision. For example, frame-level video classification, sequence modelling, and recent visual question-answer tasks.

The image description generation method based on the encoder-decoder model is proposed with the rise and widespread application of the recurrent neural network. In the model, the encoder is a convolutional neural network, and the features of the last fully connected layer or convolutional layer are extracted as features of the image. The decoder is a recurrent neural network, which is mainly used for image description generation. Because RNN training is difficult, and there is a general problem of gradient descent, although it can be slightly compensated by regularization, RNN still has a fatal flaw that it can only remember the contents of the previous limited time unit, and LSTM is a special RNN architecture that can solve problems such as gradient disappearance, and it has long-term memory. In recent years, the LSTM network has performed well in dealing with video-related context. Similar with video context, the LSTM model structure in Figure 3 is generally used in the text context decoding stage.

1.2. Comparative Analysis of Existing Works

A comparative analysis of various methods is done. The comparisons of the results obtained are made with the help of two novel approaches. The quality of captions are evaluated by the standard evaluation metric BLEU and METEOR respectively. The BLEU and METEOR scores of various approaches are mentioned below:

Models	MS-COCO				
	B1	B2	B3	B4	Meteor
Image cap with semantic attention	0.709	0.537	0.402	0.304	0.243
Hard Attention	0.718	0.504	0.357	0.250	0.230
Adaptive attention via visual sentinel	0.742	0.580	0.439	0.332	0.266
SCST:Att2in	-	-	-	0.313	0.26
SCST:att2all	-	-	-	0.30	0.259
ResNet	0.745	-	-	0.334	0.261
Bottom up and top down attention	**0.772**	-	-	**0.362**	**0.27**

Table 1.Comparision of different Image Captioning Methods.

Chapter 3

Project Objective

The objective of picture subtitling is to consequently create portrayals for a given picture, i.e., to catch the connection between the items present in the picture, produce regular language articulations and judge the nature of the produced portrayals.

The essential objective of this venture is to find out about profound learning strategies. For picture inscribing, we fundamentally utilize two methods: CNN and LSTM. Therefore, we'll consolidate these designs to make our picture inscription generator model. It's otherwise called the CNN-RNN model.

The CNN calculation is utilized to extricate highlights from a picture. We'll use the VGG16 model, which has proactively been prepared. The contribution from CNN will be utilized by LSTM to assist with delivering a portrayal of the picture.

Chapter 4

Proposed Methodology

This project needs a dataset that includes both photographs and captions. The image captioning model should be able to be trained using the dataset.

Flickr 8K Dataset

The Flickr 8k dataset is a publicly available image-to-sentence description benchmark. There are 8091 photos in this collection, each with five captions. These photos were taken from a variety of groups on the Flickr website. Each caption gives a detailed description of the objects and events seen in the photograph. The collection represents a wide range of events and settings and excludes photographs of well-known persons and locations, making it more generic. The dataset has the following characteristics that make it ideal for this project are:

- When many captions are mapped to a single image, the model becomes more general and avoids overfitting.

- Using a variety of training images allows the image captioning model to cope with a variety of image types, making the model more robust.

Why Flickr 8K dataset?

1. It's tiny in size. As a result, the model may be quickly trained on low-end laptops .
2. The data has been appropriately labeled. There are five captions for each image.
3. The dataset can be downloaded for free.

Image Data Preperation

The image should be transformed into appropriate features that can be used to train a deep learning model. In order to train any image in a deep learning model, feature extraction is required. I have divided the dataset into three parts namely train image, validation image and test image containing 4855, 1618 and 1618 images respectively.

Inscription Information Planning

For each photo in the Flickr 8k assortment, there are five subtitles. For each photo, I just utilized one subtitle. Each picture id is utilized as a key in the information planning stage, and the subtitles are saved as values in a word reference.

Data Cleaning

Crude text should be changed to a useable configuration before the text dataset can be utilized in AI or profound learning models. Prior to involving the text for the undertaking, it should be cleaned as follows:

• Eliminate all accentuation marks.

• Dispose of the numbers.

• Single-length words evacuation.

• Lowercase to capitalized character transformation. Stop words are not disposed of from the text information since doing so would make it more challenging to produce a linguistically right inscription, which is expected for this task. After information purging,

Chapter 5

Design and Implementation

The steps involved in implementation are:

1. Get Dataset
2. Load Data
3. Prepare Photo data
4. Prepare Text data
5. Encode Text data
6. Generate output text dataset
7. Define model
8. Fit model
9. Evaluate model
10. Generate caption

1. Get Dataset

I decided to use the Flickr 8k dataset. It has 8091 images and 5 captions for each image. Each image has 5 captions because there are different ways to caption an image. I downloaded the data set from Flickr website.

2. Load Data

I will copy the path of the caption data file and image file separately and will access the data one by one from disk.

3. Prepare Photo Data

I use VGG16 pretrained CNN models to extract features from images. We remove the last layers of the model because I am not interested in classifying images. I am interested in the representation Of the images. I saved the feature of all images in a file named <feature.pkl= VGG16 models require images of a concrete size, 224 pixels respectively. So, images had to be resized, then converted to array and reshaped. The extracted features are vectors of size 4096.

4. Prepare Text Data Firstly

I load all the descriptions of the images. I create a dictionary that maps image names to descriptions. To prepare the text data, I needed to clean the description of the images. For this, we converted all the words to lowercase, I removed all the punctuation, words one character long and words with numbers. Next, I create a vocabulary(dictionary) with the unique words of the descriptions. The size of the created vocabulary is 4373.

5. Encode Text Data

Here I use a tokenizer to create a map from words of the vocabulary to integers. I also calculate the size of the vocabulary and the maximum length of descriptions which is 29 to use later.

6.Generate Output Text Data

First, let's look at our model's input and output. We must provide input and output to the model for training to turn this task into a supervised learning activity. Our model will be trained on 4855 photos, each of which will include a 4096-length feature vector and a caption encoded as numbers. For training and validation purposes, we will generate an output dataset from the image caption dataset for a specific input text sequence.

7. Define Model

I've previously extracted the features from all of the images; now I shall use the Keras Model to determine the structure of the final model. It will be divided into two parts:

- **Sequence Processor** – The textual input will be handled by an embedding layer, which will be followed by the LSTM layer.
- **Decoder** - To Decoder - To make the final prediction, we will combine the output from the above two layers and process it via the dense layer. The number of nodes in the final layer will be equal to the size of our vocabulary.

8. Fit Model

LSTM+VGG16 In this part, we fit the model to the data we had. We monitor training and validation loss.We save the best model with lowest validation loss in order to use it later,because fitting is taking a very long time. The model receives image features, a sequence of words and the output word.we defined the

maximum no of epochs to 30. Finally, we visualize the training and validation loss to know better how our model is learning.

9. Evaluate Model

Now we need to evaluate our model so I choose to calculate the mean BLEU-scores on the test data set. This requires comparing the original caption with the predicted one. The BLEU score ranges from 0 to 1. ● 0 indicate no similarity between two sentences , ● 1 indicates two sentences are exactly similar. The mean of BLEU score on test data we got from our model is 0.39.

10. Generate Caption

We generate captions for some images of the test set to view the performance of our model with real examples. They have some mistakes but it capture many aspects of the images correctly.

CODE SNIPPETS

```
import os
import pickle
import numpy as np
from tqdm.notebook import tqdm

from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Model
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add
```

Load Caption data

In [15]:
```
with open(os.path.join(BASE_DIR, 'captions.txt'), 'r') as f:
    next(f)
    captions_doc = f.read()
```

In [16]:
```
# create mapping of image to captions
mapping = {}
# process lines
for line in tqdm(captions_doc.split('\n')):
    # split the line by comma(,)
    tokens = line.split(',')
    if len(line) < 2:
        continue
    image_id, caption = tokens[0], tokens[1:]
    # remove extension from image ID
    image_id = image_id.split('.')[0]
    # convert caption list to string
    caption = " ".join(caption)
    # create list if needed
    if image_id not in mapping:
        mapping[image_id] = []
    # store the caption
    mapping[image_id].append(caption)
```

Preprocessing

In [22]:
```
def clean(mapping):
    for key, captions in mapping.items():
        for i in range(len(captions)):
            # take one caption at a time
            caption = captions[i]
            # preprocessing steps
            # convert to lowercase
            caption = caption.lower()
            # delete digits, special chars, etc.,
            caption = caption.replace('[^A-Za-z]', '')
            # delete additional spaces
            caption = caption.replace('\s+', ' ')
            # add start and end tags to the caption
            caption = 'startseq ' + " ".join([word for word in caption.split() if len(word)>1]) + ' endseq'
            captions[i] = caption
```

In [38]:
```
# train the model
epochs = 30
batch_size = 32
steps = len(train) // batch_size

for i in range(epochs):
    # create data generator
    generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch_size)
    # fit for one epoch
    model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)
```

Visualize the Results

In [63]:
```
from PIL import Image
import matplotlib.pyplot as plt
def generate_caption(image_name):
    # load the image
    # image_name = "1001773457_577c3a7d70.jpg"
    image_id = image_name.split('.')[0]
    img_path = os.path.join(BASE_DIR, "Images", image_name)
    image = Image.open(img_path)
    captions = mapping[image_id]
    print('---------------------Actual---------------------')
    for caption in captions:
        print(caption)
    # predict the caption
    y_pred = predict_caption(model, features[image_id], tokenizer, max_length)
    print('--------------------Predicted--------------------')
    print(y_pred)
    plt.imshow(image)
```

In [64]:
```
generate_caption("1003163366_44323f5815.jpg")
```

```
---------------------Actual---------------------
startseq man lays on bench while his dog sits by him endseq
startseq man lays on the bench to which white dog is also tied endseq
startseq man sleeping on bench outside with white and black dog sitting next to him endseq
startseq shirtless man lies on park bench with his dog endseq
startseq man laying on bench holding leash of dog sitting on ground endseq
--------------------Predicted--------------------
startseq man in plaid shirt lays on the bench endseq
```

Chapter 6

Results and Discussion

Output of the project will look like as follows, where we will provide image's path and image's caption will be predicted: -

two children playing soccer on field

two dogs are walking through the woods

crowd of people are waiting in line in front of crowd of people

man in red shirt and black shorts is standing on beach

two girls in orange shirts are standing

man in blue shirt and blue shirt is hitting tennis ball

Chapter 7

Conclusion and Future Scope

CONCLUSION

We looked at deep learning-based picture captioning algorithms in this report. While deep learning-based picture captioning systems have made significant progress in recent years, a robust image captioning approach capable of producing high-quality captions for practically all photos has yet to be developed.

Automatic image captioning will remain a hot research topic for a while, thanks to the emergence of novel deep learning network architectures. We used the Flickr 8k dataset, which contains over 8000 photos as well as the captions for each image. Despite the fact that deep learning-based picture captioning systems have made significant progress in recent years, a reliable image captioning method capable of producing high-quality captions for practically all photos has yet to be developed.

Automatic image captioning will remain a hot research topic for quite some time, thanks to the emergence of innovative deep learning network architectures. Because the number of users on social media is growing by the day, and the majority of them will share images, the scope of image captioning will be enormous in the future. As a result, this project will be of greater assistance to them.

FUTURE SCOPE

Image caption, which automatically generates English descriptions based on the information identified in a picture, is an important aspect of scene interpretation, which combines data from computer vision with language processing.

We can improve our results by making a variety of changes, such as:

1. Making use of a larger dataset.
2. Extensive hyper parameter tuning (learning rate, batch size, variety of layers, variety of units, dropout rate, batch standardization etc.).
3. To avoid overfitting, use the cross validation set. Image retrieval research that uses the context of the photos, such as image captioning, will help solve this challenge in the future. This project could be improved in the future by training it with new image captioning datasets to improve the identification of classes with lesser precision. This methodology can be used in conjunction with other image retrieval methods such as histograms, shapes, and so on, to see whether the image retrieval results improve.

References

1. 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. arXivpreprint arXiv:1707.07998 (2017)
2. https://www.youtube.com/watch?v=7nnSjZBJVDs&list=PLQflnv_s49v_i1OVqE0DENBk-QJt9THjE
3. https://www.youtube.com/watch?v=rfAvjCf1ZI&list=PLyqSpQzTE6M_PI-——— rIz4O1jEgff hJU9GgG
4. https://d2l.ai/d2l-en.pdf
5. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
6. CS771 Project Image Captioning by Ankit Gupta, Kartik Hira, Bajaj Dilip.
7. https://researchrepository.murdoch.edu.au/id/eprint/60782/1/Hossain2020.pdf