IMAGE CAPTION GENERATOR

Mainly, we need to use the Supervised Machine Learning because our project requires a model to learn a mapping between an image and its corresponding captions, and this can be easily learnt by the labeled dataset of image and captions.

Our model is an deep learning model.

Python Libraries used in the Project –

1. **NumPy-**

   Used to perform **mathematical and logical operations** on Arrays.
   Contains Multi-dimensional Arrays and Matrices.

2. **os-** for handling the files.

3. **Pickle-** Storing some NumPy Features such as, If we are **extracting image features**, then we have to **store it** somewhere, for this we use this library.

4. **Tqdm – from tqdm.notebook** for giving us UI to see how many data is processed till now. This gives overall estimation of a process.

5. **Tensorflow –** For building and training deep learning model.

6. **Keras –** Provides interface for building and training deep learning models.

    VGG16 – for **extracting features** from image data.

    Preprocess_input – preprocessing the image data for the VGG.

    Load_img, img_to_array – requires for preprocessing of image.

    Tokenizer- used to **convert a sequence of text into a sequence of integers**

    Pad_sequences- It will **even out** the whole representation of the text of features.

    Model – This class is used for **defining, training, and evaluating deep learning models.**

    To_categorical, plot_model – gives clear representation of out whole model in the form of image.

7. **Pillow -** To load the image files.
8. **Pydot –** For plotting the Model
9. **NLTK -** Needed for BLEU_score

REQUIREMENT –

1. Download Kaggle dataset-

https://www.kaggle.com/datasets/adityajn105/flickr8k?resource=download

Put this DATASET, on this Repository Main-Folder by creating

" Dataset/ " folder.IT IS MUST TO DOWNLOAD, to run model.

2. pip install jupyterlab
3. upgrade pip -> python.exe -m pip install --upgrade pip
4. pip install tensorflow
5. pip install tqdm
6. pip install NLTK - needed for BLEU_score
7. pip install pillow - to load the image
8. pip install pydot – for plotting the Model
9. To start jupyter – type -> jupyter-lab
10. Create new folder as 'Image Caption Generator' in jupyter
11. Create New folder as 'Dataset' and paste the downloaded dataset files i.e., Images Folder and captions.txt file.
12. Then, open an Image_Caption_Generator.ipynb and run line by line code.
13. For plotting model –
    a. In windows, we need to download the graphviz software from https://graphviz.gitlab.io/download/
    b. Then, copy bin folder path like, C:\Program Files\Graphviz\bin
    c. ADD this path to the Environmental variables in user section.

Steps need to follow while making model –

1. First, we need the dataset for predictions, we use flickr8k dataset which contains 8k images with their different captions.
   https://www.kaggle.com/datasets/adityajn105/flickr8k?resource=download

2. First, we need to **load the VGG16** model using **VGG16()**.

   model = VGG16()

3. Then, **restructure** the model by **removing predictions Layer** using **model.layers[-2]** in the code. If we type layers[-1], then it takes the predictions layer.

model=Model(inputs=model.inputs, outputs=model.layers[-2].output)

4. Then, extract the features of each image from the dataset by **loading the image from the Images Directory path**.

   i. **Load the Image** using load_img –

   Image = load_img(img_path, target_size=(224,224))

   ii. Convert image **pixels to NumPy Array**-

   Image = img_to_array(image)

iii. **Reshape the data** for model to extract features –

Image = image.reshape(1, image.shape[0], image.shape[1], image.shape[2])

This is the **RGB** image, we are having **3 - dimensions and 1 for single sample**

iv. **Preprocess Image for VGG-**
Image = preprocess_input(image)

v. **Extract features from image-**
feature = model.predict(image, verbose=0)

vi. **Getting Image ID, simply ID – name of image previous of .jpg -**
Image_id =img_name.split('.')[0]

vii. Store that extracted feature of image in the dictionary where all features are stored using image_id-

features[image_id] = feature

viii. If we **don't want to re-extract the features** or **don't want to lose the features**, because we have near about 8000 images as a dataset. So, re-extracting the features waste our time, we can use **pickle to store features in a pickle file-**

pickle.dump(features, open(os.path.join('./','features.pkl'),'wb'))

'./' – is a working directory, where pkl file stores.

And, then if we want to use these features somewhere else, then we can **load the features** where needed using that pickle file **features.pkl**-

with open(os.path.join('./','features.pkl'),'rb') as f:

features = pickle.load(f) – f is for the one by one feature

5. It's time **to load the captions data** from base directory when it is present, **'Dataset'** is the folder in current directory where all our inputs i.e., all images and captions are stored.

```
with open(os.path.join('.//Dataset',captions.txt), 'r') as f:
    next(f)
    captions_doc = f.read()
```

6. Now, we need to create the mapping of image to captions-

   i. First, we **split the caption_doc** using the **'\n'**, because we have different captions at next line, so that's why we split captions using '\n'.

```
captions_doc.split('\n')
```

   ii. Then, **split the single line into the token** from caption_doc using **comma(,),** that gives us an image_name and its caption.

```
tokens = line.split(',')
```

   iii. Separating the **token** as **image_name and caption**.

```
image_id, caption = token[0], token[1:]
```

   iv. Then, **removing extension** from image_name by spliting it using **'.'** and storing as an image_id

```
image_id = image_id.split('.')[0]
```

   v. **Storing the multiple captions** in mapping for single image using **append()** -

mapping[image_id].append(caption)

7. PREPROCESSING OF TEXT DATA IN CAPTION-

We simply convert all text into the **lowercase letters**, then we **remove the special characters and digits** from the caption and also **removing the additional spaces** with **ignoring the letter** in caption which has **length = 1**.

And, further we add the **startseq** and **endseq** to understand machine that, there is end of the caption.

8. Creating Instance of **Tokenizer** & using this, it converts **all caption's word into the sequences of numbers which are the unique tokens for each word**.

Tokenizer **simplifies the task of converting text data into numerical data** which can be fed into a neural network, this leads to improvement of **model accuracy and performance**.

9. Then, we need to **split the data into train & test** in the ratio of 90-10 for retrieving data very effectively,

we split it into different parts so that normal

system can also run efficiently.

10. Its time to **CREATE THE MODEL** with two layers- **Image feature layer, sequence feature layer.** We

can see the model in the png form by plotting it using **plot_model().**

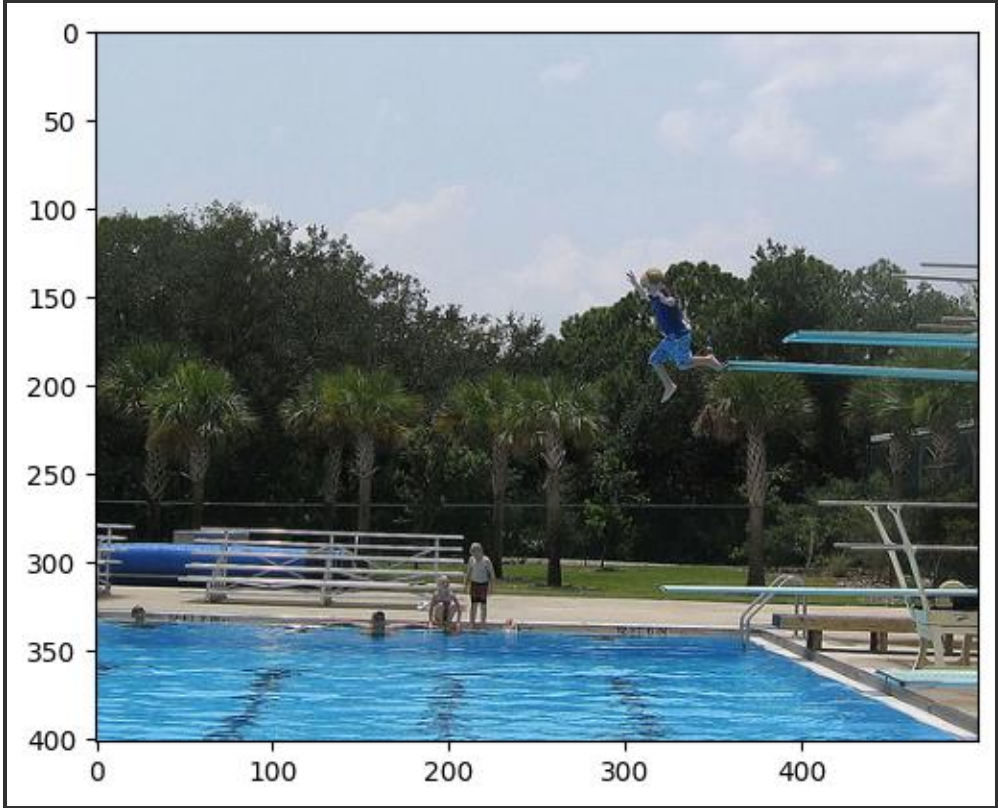
11. Then, its time to **TRAIN THE MODEL** by fetching the images and captions in batch to avoid the session crash, in my case, it will train the model for the 15-epochs, If you want more accuracy, then increase the number of epochs.

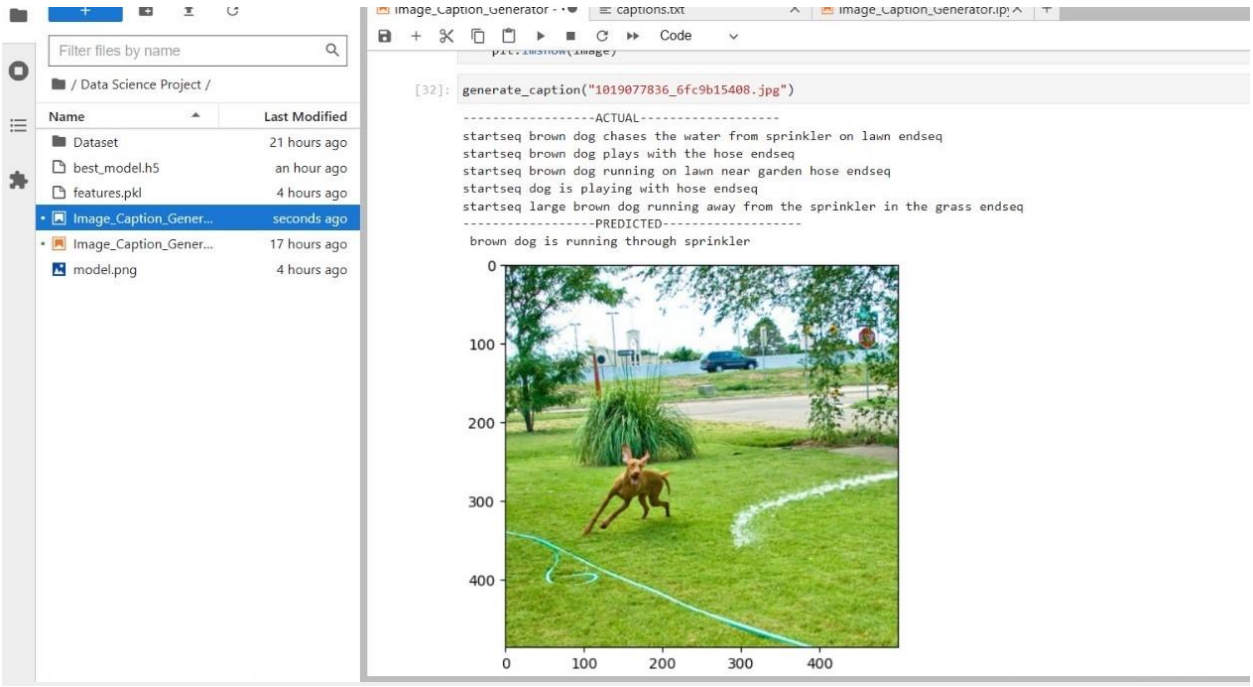
12. Lastly, we are **validating our model** using the **test data** means, predicting the captions for the test data and checking for correctness and we will able to see the score of accuracy in **bleu_score which gives value in between 0-1.**

13. ITS TIME TO **VISUALIZE OUR PREDICTED CAPTIONS FOR THE SPECIFIC IMAGE.** It will generate the caption by validating the model and above all operations.
14. **FINALLY, WE SEE THE NEWLY PREDICTED CAPTION with startseq & endseq word,** which was used while we are training the model and generating the caption.

TESTED INPUTS ARE BELOW:

```
------------------ACTUAL-------------------
startseq boy descends off the end of high diving board endseq
startseq child jumps off high diving board into the pool endseq
startseq kid jumps off the diving board and into the swimming pool below endseq
startseq little kid is jumping off high dive at the pool endseq
startseq the boy is jumping off high diving board into the pool endseq
------------------PREDICTED-------------------
startseq boy jumps off diving board into pool endseq
```




Filter files by name
/ Data Science Project /
Name
Last Modified
Dataset
21 hours ago
best_model.h5
an hour ago
features.pkl
4 hours ago
Image_Caption_Gener...
seconds ago
Image_Caption_Gener...
17 hours ago
model.png
4 hours ago
captions.txt
Code
[32]:
generate_caption("1019077836_6fc9b15408.jpg")
----------ACTUAL----------
startseq brown dog chases the water from sprinkler on lawn endseq
startseq brown dog plays with the hose endseq
startseq brown dog running on lawn near garden hose endseq
startseq dog is playing with hose endseq
startseq large brown dog running away from the sprinkler in the grass endseq
----------PREDICTED----------
brown dog is running through sprinkler
0
100
200
300
400
0
100
200
300
400