IMAGE CAPTION GENERATOR USING NEURAL NETWORKS

OVERVIEW:

THE GOAL OF THIS PROJECT IS TO DEVELOP A DEEP LEARNING MODEL THAT CAN GENERATE CAPTIONS FOR IMAGES. THE MODEL TAKES AN IMAGE AS INPUT AND GENERATES A NATURAL LANGUAGE CAPTION THAT DESCRIBES THE CONTENT OF THE IMAGE. THIS TASK IS ALSO KNOWN AS IMAGE CAPTIONING AND REQUIRES THE USE OF BOTH COMPUTER VISION AND NATURAL LANGUAGE PROCESSING TECHNIQUES.

The model is trained on a large dataset of images and their corresponding captions. It learns to associate visual features in the images with words and phrases in the captions, and it uses this knowledge to generate new captions for novel images that it has not seen before.

The project has several applications, including assisting the visually impaired to understand images and helping with content-based image retrieval. Additionally, image captioning can be useful in fields such as journalism, social media, and e-commerce, where image descriptions are necessary for users to understand and engage with visual content.

JOURNEY:

The project goes under several phases of Data Analysis or steps as mentioned below for cleaning, transforming, processing, visualizing, and modelling data to get results:

1. Data Collection

2. Data Exploration

3. Data Pre-processing

4. Feature Extraction

5. Training Model

6. Testing Model and Evaluation

1. Data Collection:

The data collection process for this project involved downloading the Microsoft Common Objects in Context (COCO) dataset, which contains over 330,000 images with more than 2.5 million captions. Specifically, the "train2014" subset of the dataset was used, which contains approximately 83,000 images.

To download the dataset, the annotations file was first obtained by using the URL "http://images.cocodataset.org/annotations/annotations_trainval2014.zip".
The annotations were then extracted and loaded into a Pandas data frame containing the image IDs and corresponding captions.

Next, the top 10,000 images with the highest number of captions were selected and downloaded from the "train2014" subset of the COCO dataset. This was done to reduce the size of the dataset and make it easier to work with.

Overall, the data collection process for this project involved obtaining a large and diverse dataset of images and captions, which can be used to train and evaluate a deep learning model for image captioning.

2. Data Exploration:

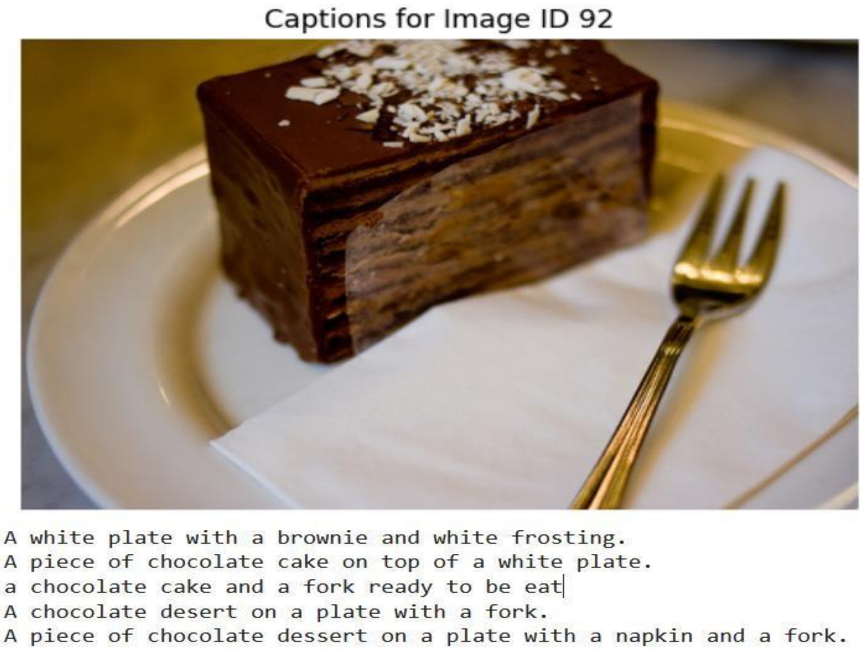
Data Exploration in this project involves two main categories.

Visualizing Images:

Visualizing a sample of images from the dataset provides an initial glimpse into the nature of the data. This step helps in understanding the image content, quality, and diversity.

Exploring Captions:

Analyzing the captions provides insights into the language used, caption lengths, vocabulary size, and distribution of captions across images. Descriptive statistics, such as the average caption length, can be calculated to understand the typical length of captions in the dataset.


Captions for Image ID 92

```
A white plate with a brownie and white frosting.
A piece of chocolate cake on top of a white plate.
a chocolate cake and a fork ready to be eat
A chocolate desert on a plate with a fork.
A piece of chocolate dessert on a plate with a napkin and a fork.
```

3. Data Pre-processing:

Data Pre-processing in this project involves two main categories.

- *Pre-processing the text data:* The captions were pre-processed to remove any unnecessary characters, convert all characters to lowercase, and tokenize the captions.
- *Tokenizing the captions:* The captions were tokenized using the Tokenizer class from the Keras library. This involved creating a vocabulary of all the unique words in the captions and assigning a unique integer to each word. The tokenized captions were then padded to ensure that they all had the same length.
- *Creating the training dataset:* The pre-processed image features and tokenized captions were combined to create the training dataset. The model was trained on this dataset to learn to generate captions for images.

4. Feature Extraction:

In this project, the VGG16 model is used to extract features from the images. VGG16 is a pre-trained convolutional neural network that is widely used for image classification tasks. The model has 13 convolutional layers and 3 fully connected layers.

To extract features, first, each image is resized to (224, 224) pixels, which is the required input size of the VGG16 model. Then, the pre-trained VGG16 model is used to predict the output for each image. Specifically, the output of the second last layer (the layer before the classification layer) is used as the image feature. This layer outputs a vector of size 4,096, which represents the features of the input image.

The extracted features are then saved in a dictionary where the key is the image filename (without the file extension), and the value is the extracted feature vector. This dictionary is used later to train the caption generator model.

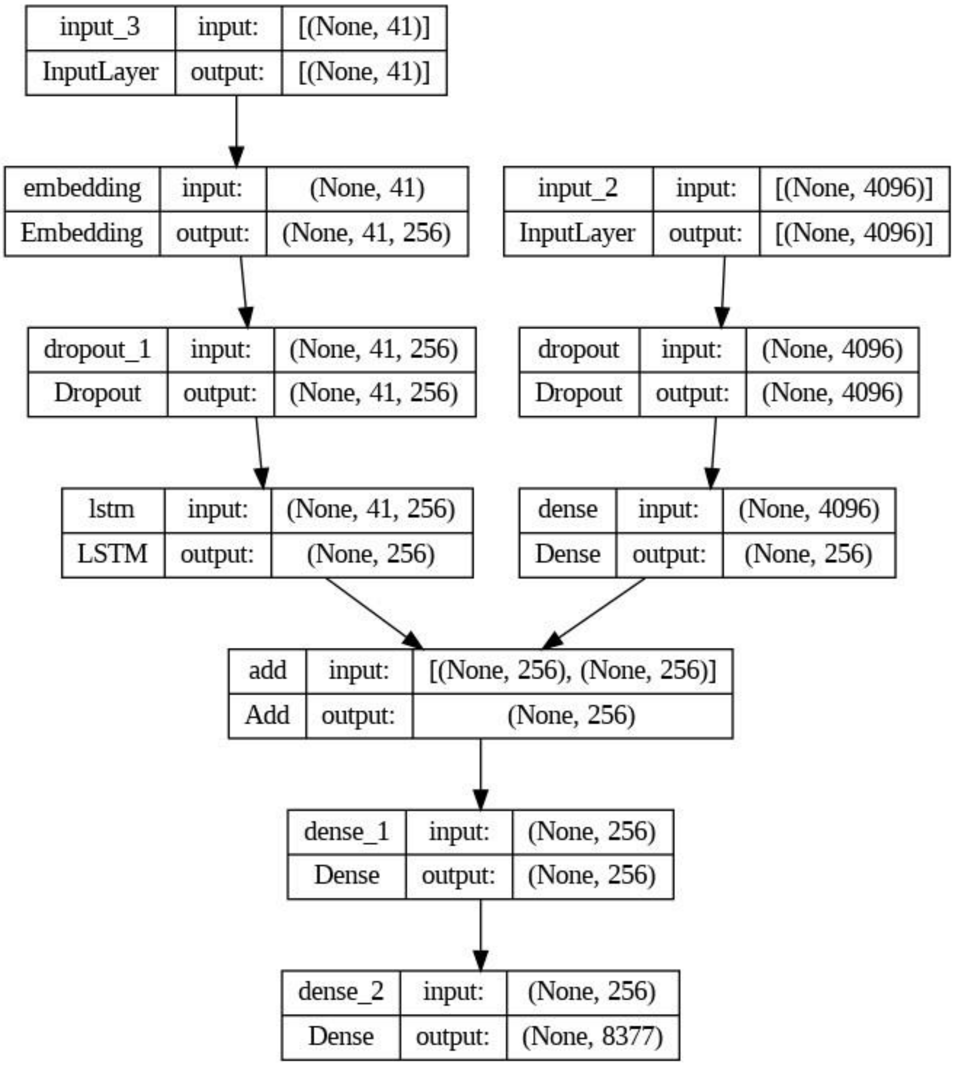
5. Training Model:

The purpose of this project is to generate captions for images using a deep learning model. After the data collection and pre-processing steps, a model is trained on the pre-processed data to learn the relationship between image features and their corresponding captions.

The training process involves the following steps:

1. *Model Architecture:* The model architecture used in this project is a combination of a convolutional neural network (CNN) and a recurrent neural network (RNN). CNN is responsible for extracting image features, while the RNN generates captions based on the extracted features.
2. *Data Preparation:* The pre-processed data, which includes the image features and corresponding captions, is split into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate the model's performance.
3. *Model Training:* The model is trained using the training set. During training, the model takes an input pair consisting of an image feature vector and a partial caption sequence. It predicts the next word in the caption sequence based on the input pair. The model's parameters are adjusted using an optimization algorithm (e.g., Adam optimizer) to minimize the loss between the predicted and actual captions.

4. *Batch Training:* To avoid memory constraints and improve training efficiency, the training data is divided into batches. Each batch contains a subset of the training data, and the model is trained on each batch iteratively. This approach allows the model to process and update its parameters in smaller increments, leading to more stable training.
5. *Training Parameters:* The model training process involves setting various parameters, such as the number of epochs, batch size, and learning rate. The number of epochs determines the number of times the entire training dataset is processed by the model. The batch size determines the number of samples processed in each training iteration. The learning rate controls the step size in the optimization algorithm and affects how quickly the model learns.


input_3 | InputLayer | input: [(None, 41)] | output: [(None, 41)]
embedding | Embedding | input: (None, 41) | output: (None, 41, 256)
input_2 | InputLayer | input: [(None, 4096)] | output: [(None, 4096)]
dropout_1 | Dropout | input: (None, 41, 256) | output: (None, 41, 256)
dropout | Dropout | input: (None, 4096) | output: (None, 4096)
lstm | LSTM | input: (None, 41, 256) | output: (None, 256)
dense | Dense | input: (None, 4096) | output: (None, 256)
add | Add | input: [(None, 256), (None, 256)] | output: (None, 256)
dense_1 | Dense | input: (None, 256) | output: (None, 256)
dense_2 | Dense | input: (None, 256) | output: (None, 8377)

6. Testing Model and Evaluation:

To test the trained model, we use a separate test set of image-caption pairs that were not used during training. For each image in the test set, we generate a predicted caption using the trained model and compare it to the actual captions using the BLEU score.

The BLEU score is a common evaluation metric for image captioning that measures how well the predicted caption matches the actual captions. It ranges from 0 to 1, with a higher score indicating a better match.

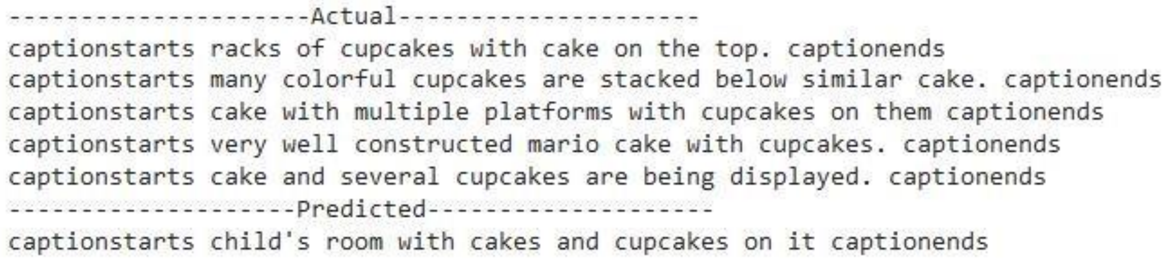
In our implementation, we use the NLTK library to calculate the BLEU score. We iterate over each image in the test set, generate a predicted caption, and compare it to the actual captions using the corpus_bleu function. We calculate two different scores: BLEU-1 and BLEU-2. BLEU-1 only considers unigrams (individual words), while BLEU-2 considers bigrams (pairs of adjacent words) as well.

The results of the evaluation are printed out in the console. We can see the BLEU-1 and BLEU-2 scores for the entire test set. These scores give us an indication of how well the model is able to generate captions that match the actual captions.

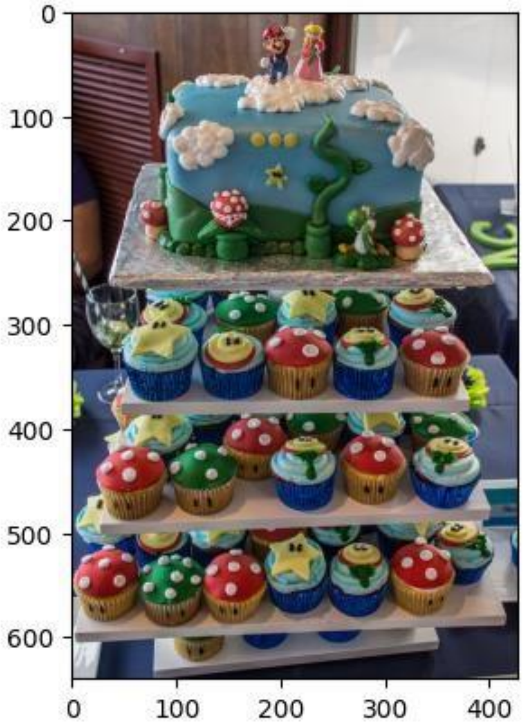
In addition to the BLEU score, we also perform a visual inspection of the predicted captions. We randomly select a few images from the test set and display the actual and predicted captions along with the corresponding image using the generate_caption function. This allows us to get a qualitative sense of how well the model is able to generate captions that make sense and accurately describe the image.

Overall, the evaluation report provides a comprehensive view of how well the trained model performs on the test set. The BLEU scores give us a quantitative measure of the model's performance, while the visual inspection allows us to get a qualitative sense of how well the model is able to generate captions. Based on these metrics, we can determine whether the model needs further refinement or if it is ready for deployment.

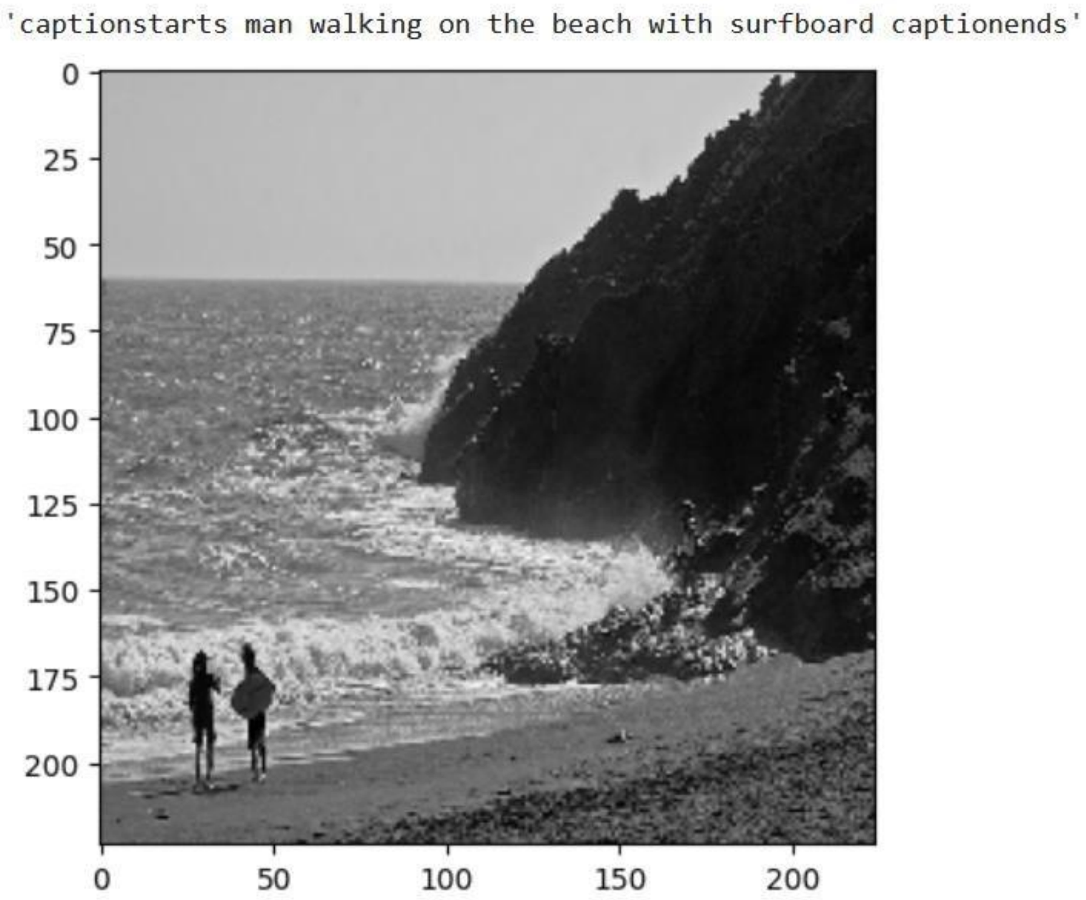
Below is the caption generated for an image from the dataset along with their actual captions.



```
---------------------Actual--------------------
captionstarts racks of cupcakes with cake on the top. captionends
captionstarts many colorful cupcakes are stacked below similar cake. captionends
captionstarts cake with multiple platforms with cupcakes on them captionends
captionstarts very well constructed mario cake with cupcakes. captionends
captionstarts cake and several cupcakes are being displayed. captionends
--------------------Predicted-------------------
captionstarts child's room with cakes and cupcakes on it captionends
```



Now below is the predicted caption of a new image.



```
'captionstarts man walking on the beach with surfboard captionends'
```

Future Scope:

The image captioning model has a wide range of potential future applications. Here are a few areas where this model can be extended or applied:

- *Multilingual image captioning*: The current model is trained only on English captions. However, with the rise of multilingual content on the internet, there is a need for models that can generate captions in different languages. The model can be trained on a multilingual dataset to generate captions in multiple languages.
- *Video captioning:* Video captioning is the process of generating captions for videos. This model can be extended to generate captions for videos as well. The model can be trained on a dataset of videos and their corresponding captions to generate accurate captions for videos.
- *Contextual image captioning:* The current model generates captions solely based on the content of the image. However, the meaning of the caption can be enhanced by incorporating contextual information such as time, location, and user preferences. For instance, the model can generate captions that are relevant to a specific time of day or a particular location.
- *Image captioning for the visually impaired:* Image captioning can be a valuable tool for the visually impaired to understand the content of images. The model can be trained on a dataset of images and their corresponding captions, and the generated captions can be read out loud using text-to-speech technology.
- *Image search engine:* The model can be used as a search engine for images based on captions. Users can enter a caption, and the model can retrieve images that are relevant to the caption. This can be a useful tool for finding specific images based on their content.