Image Caption Generator using VGG-16 and LSTM

- By Chaturya Katragadda, Mahima Chowdary Maddineni

Introduction

The objective of the project was to train convolutional neural networks with several hundreds of hyperparameters, then apply it on a massive dataset of images and combine the results of this image classifier in order to generate a caption for the image that had been classified.

The basic working of the project is that the features are extracted from the images using pre-trained VGG16 model and then fed to the LSTM model along with the captions to train. The trained model is then capable of generating captions for any images that are fed to it.

Generating a textual description of an image is a popular subject in the field of artificial intelligence research known as "image caption generator." Producing sentences with proper syntax and meaning calls for fluency in both areas of language study. Capturing the essence of an image in a few well-crafted sentences is a formidable challenge, but the results could be life-changing for the visually impaired community.

Captioning images is a challenging problem that has received less attention in the computer vision community than picture categorization. A good image description will include details about the connections between the various elements in the picture. All of the foregoing semantic knowledge must be conveyed in a natural language such as English, which necessitates a language model in addition to a visual understanding of the image.

For example if we take an image

If we were asked to describe it we would do it in this way a dog is trying to run in the snow.

So, how are we doing this, how are we thinking about the descriptions. We are looking at the image but also at the same time we are looking for meaning ful relations between the objects in the image and then projecting then as a sentence. I first part is done by VGG-16 CNN model and the second is handled by LSTM.

Data set Overview

In this dataset, the images have been labeled manually with captions. The collection includes pictures with captions written in English.

https://www.kaggle.com/datasets/adityajn105/flickr8k

The dataset consists of a folder of images and a textual explanation. There are 8000 photographs in the image directory, and for each one, the description file includes 5 captions. Six thousand of the 8,000 photos are utilized for training, one thousand for development, and one thousand for testing. Here are some randomly selected photos from the collection, together with their English reference captions.Typically, a set of captions will be 12 words long. Input photos can have a resolution anywhere from 256x500 all the way up to 500x500.

If you have a powerful system with more than 16 GB RAM and a graphic card with more than 4 GB of memory, you can try to take **FLICKR 30K** which has around 30,000 images with captions.

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 white crane stands tall as it looks out upon the ocean .
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 .

Methodology

Cleaning the text data

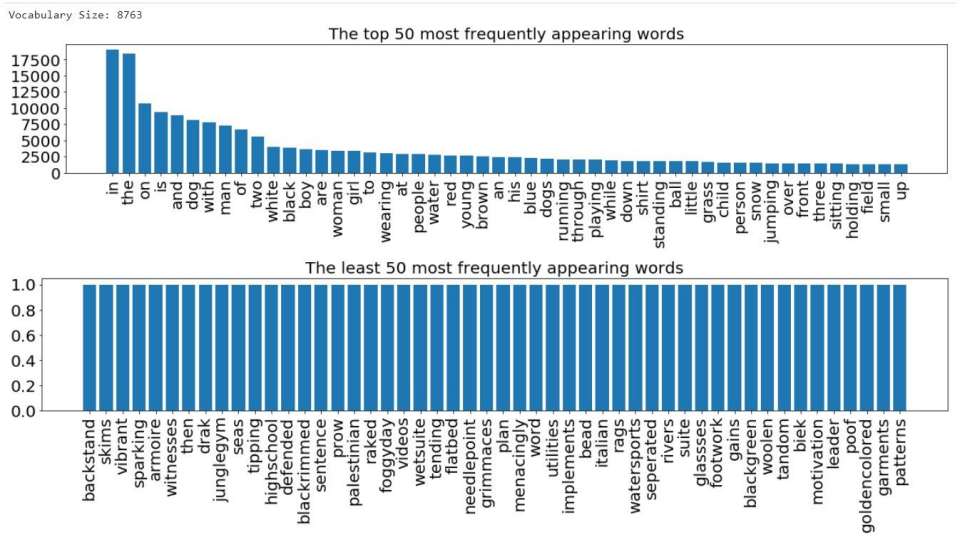
The caption dataset contains punctuations, singular words and numerical values that need to be cleaned before it is fed to the model because uncleaned dataset will not create good captions for the images

We initialized a function for removing punctuation so that the captions will be accurate while predicting

Then we Removed single character words as well as numeric values that are present in captions.

```
I ate 1000 apples and a banana. I have python v2.7. It's 2:30 pm. Could you buy me iphone7?

Remove punctuations..
I ate 1000 apples and a banana I have python v27 Its 230 pm Could you buy me iphone7

Remove a single character word..
 ate 1000 apples and banana have python v27 Its 230 pm Could you buy me iphone7

Remove words with numeric values..
    ate        : True
    1000       : False
    apples     : True
    and        : True
    banana     : True
    have       : True
    python     : True
    v27        : False
    Its        : True
    230        : False
    pm         : True
    Could      : True
    you        : True
    buy        : True
    me         : True
    iphone7    : False
 ate apples and banana have python Its pm Could you buy me
```

There are the top 50 Keywords before and after cleaning



We will be considering and training the model using clean data captions.

Then we added startseq at the brginning of the sentence and endseq at the end of the caption. The first word will indicate the beginning of the caption, and the last word will indicate the end of the caption.

Feature Extraction

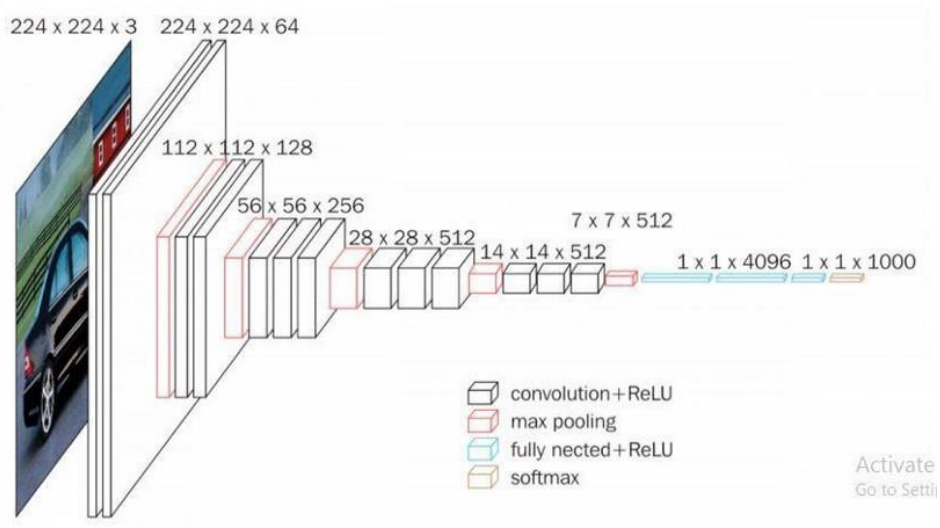
In this step, features are gathered from every image in the dataset. Images of size 224 x 224 are processed with the VGG-16 model, which returns 4096 features.

Our model does not understand strings, so we must transform them into vectors for it to work with. To separate the tokens from the descriptions, we first constructed a tokenizer object with the tokenizer() class and then used the fit_on_texts method.

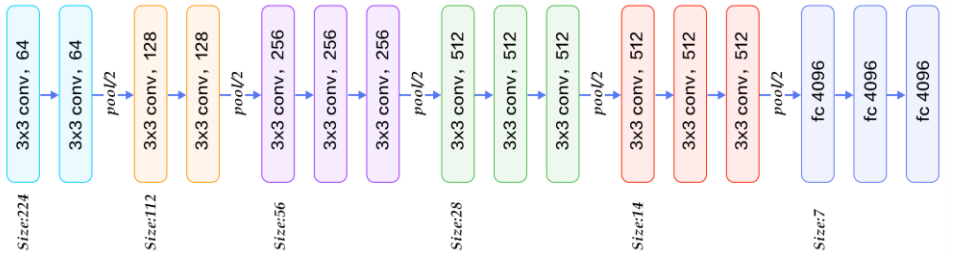
We downloaded a pretrained VGG-16 model, imported it and then loaded it into our notebook. Then we removed thew last layer

VGG-16 Model

VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.



VGG-16 is a convolutional neural network that is 16 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

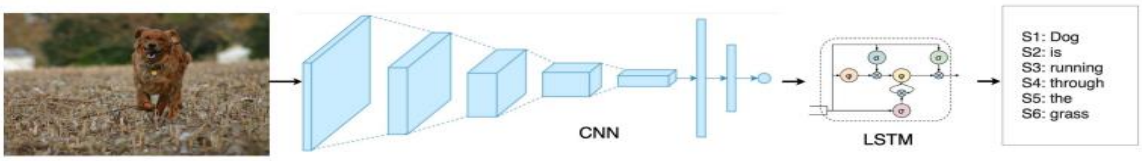


LSTM

It is used for time-series data processing, prediction, and classification. LSTM has feedback connections, unlike conventional feed-forward neural networks. It can handle not only single data points (like photos) but also complete data streams (such as speech or video).

LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images.

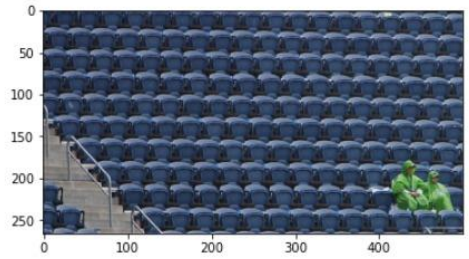
Whole processes that is being done in the project



Initially we tried without cleaning the data, we could not get proper sentences for the images. The sentences were incorrectly predicted.

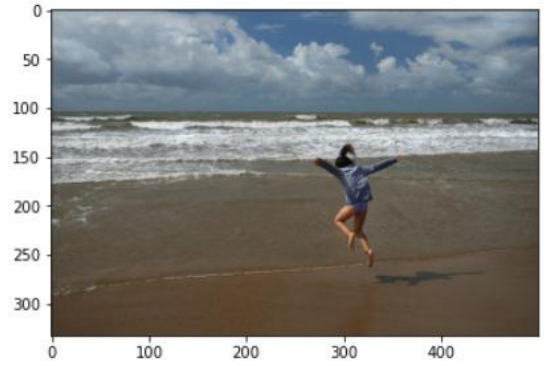
For example:

```
Caption generated by LSTM Network: start man is standing with blue helmet wait for the street end
Caption generated by RNN: start man and woman pose for the street end
BLEU for LSTM caption=1.2145546596302331e-231
BLEU for RNN caption=0

<matplotlib.image.AxesImage at 0x1ada043e1f0>
```



The predicted caption is totally wrong as there is no street there and it can be a stadium

```
Caption generated by LSTM Network: start man holds surfboard with the water end
Caption generated by RNN: start two brown dogs runs through the water end
BLEU for LSTM caption=1.3531653690559654e-231
BLEU for RNN caption=0

<matplotlib.image.AxesImage at 0x1ada24b9a30>
```



For this image it is predicting partially correct that there is water but there is no surfboard

Therefore, we thought of cleaning the data and removing all the unnecessary items in the captions

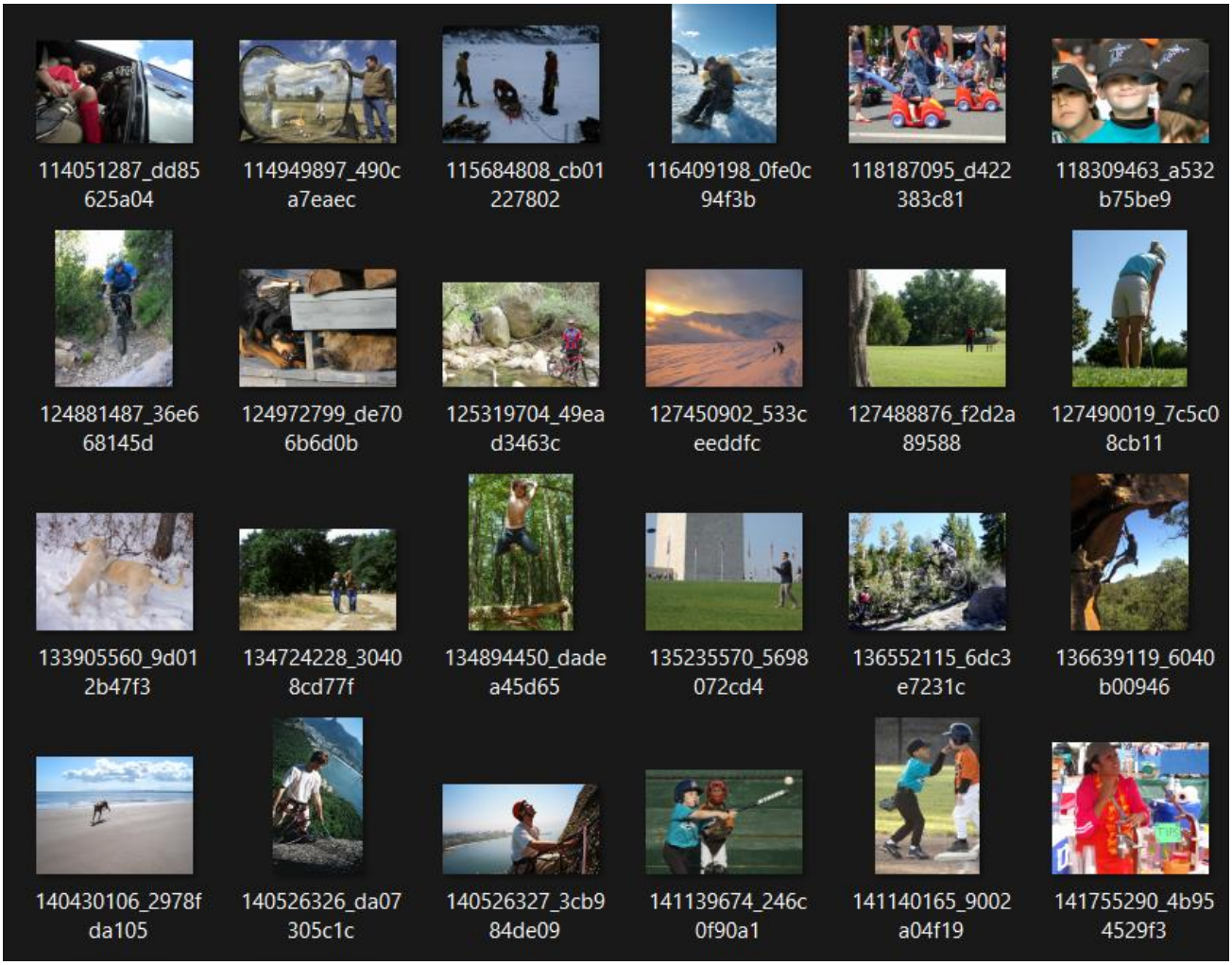
MODELLING

The decoder model will be fed the output of a combined feature extractor and sequence encoder. A 4096-element vector is the standard input to the feature extractor model. To accomplish this, we are utilizing the Input class found in keras.layers. Accordingly, the input feature vector size for this model is 4096 elements. Finally, we employ regularization via a dropout layer, which is typically employed to lessen the training dataset's overfitting. Then we use a dense layer to transform the 4096 input layer elements into a 256-element image representation.

The embedding layer of a sequence model is given sentences or descriptions as input. With the parameter mask true set to true, the padded values in the input sequences will be disregarded. We then employ a dropout layer to mitigate the effects of overfitting. Next, we'll feed these phrase descriptions into an LSTM layer with 256 memory units.

After the encoder model has been completed, the decoder model combines the vectors from the two input models by adding them. Moving on, we're use a 256-unit completely connected layer.

Similar photos from the clusters are displayed together once the VGG-16 model has finished extracting features from all the images in the dataset, allowing us to check whether or not the model has correctly extracted the features.



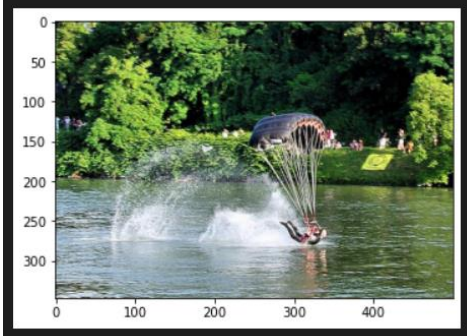
Merging the caption with the respective images

- The next stage is to combine the captions with the corresponding photos for training purposes. Due to the difficulty of training with all 5 captions, we are just using the first caption for each image in this dataset.
- Next, we must tokenize all of the captions before feeding them into the model.

Also, we tried implementing with the help of an exception model to predict the captions, here are the results

```
original caption:  ['dog standing in the shallow part of the ocean while the waves splash
around', 'dog standing in the water', 'wrinkled dog wading in shallow water', 'the large brown
dog is walking through water in the ocean', 'the large brown dog stands in the choppy water']

LSTM generated caption:  start brown dog running through the sand end

RNN generated caption:  start brown dog and brown dog is running through the water end

BLEU score for LSTM predicted Caption:  0.20425834259995942
BLEU score for RNN predicted Caption:  0

<matplotlib.image.AxesImage at 0x2873fe7a9d0>
```


```
original caption:  ['man crashes into the water with his parachute', 'parasailer splashes
along the surface of lake', 'person hanging from parachute makes splash as their body hits the
water', 'person in parachute sliding across the water', 'the parasailer is skipping across the
water']

LSTM generated caption:  start little girl is standing through the water end

RNN generated caption:  start young boy runs through the streets end

BLEU score for LSTM predicted Caption:  0.20412414523193154
BLEU score for RNN predicted Caption:  0

<matplotlib.image.AxesImage at 0x287bfba39a0>
```



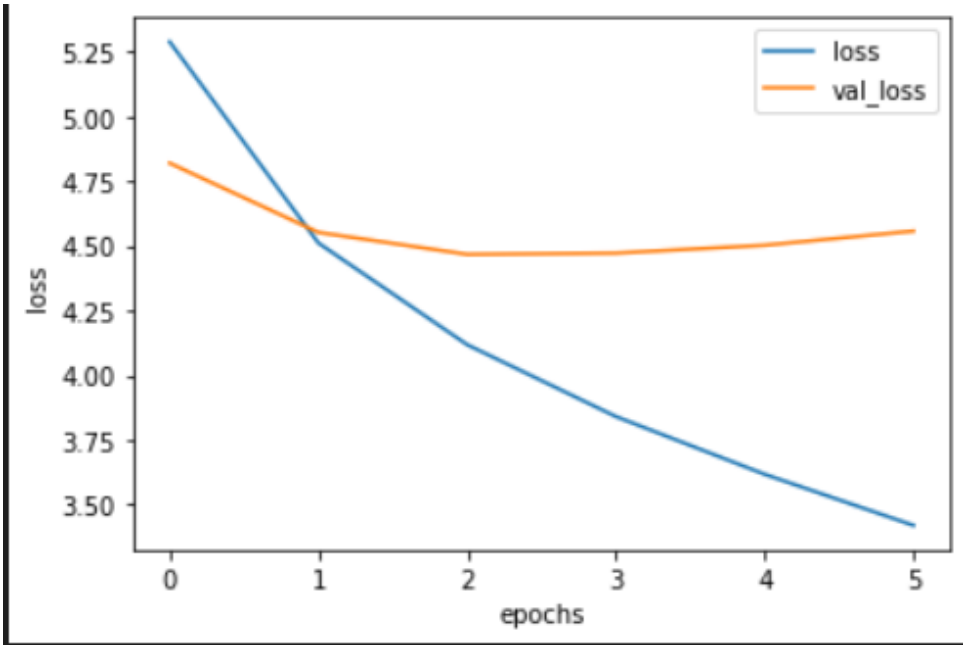
Building and training LSTM model

The LSTM model has been implemented since its output depends on both the state of the preceding cell's output and the state of the current cell's input. When creating image captions, this is helpful.

This phase entails constructing the LSTM model with two or three input layers and one output layer to generate the captions. Different configurations of the model's nodes and layers are available for training. We'll begin with 256 and then try 512 and 1024. Acceptable captions are generated by tuning the model with a variety of hyperparameters.

```
Epoch 1/6
1551/1551 - 3143s - loss: 5.2890 - val_loss: 4.8201 - 3143s/epoch - 2s/step
Epoch 2/6
1551/1551 - 2837s - loss: 4.5110 - val_loss: 4.5520 - 2837s/epoch - 2s/step
Epoch 3/6
1551/1551 - 2306s - loss: 4.1178 - val_loss: 4.4671 - 2306s/epoch - 1s/step
Epoch 4/6
1551/1551 - 2334s - loss: 3.8400 - val_loss: 4.4718 - 2334s/epoch - 2s/step
Epoch 5/6
1551/1551 - 2341s - loss: 3.6167 - val_loss: 4.5024 - 2341s/epoch - 2s/step
Epoch 6/6
1551/1551 - 2377s - loss: 3.4192 - val_loss: 4.5573 - 2377s/epoch - 2s/step
```

While training LSTM method with 6 epochs, the loss for each epoch has decreased



After the model finishes training, we can test out its performance on the some of the test images to figure out if the generated captions are good enough. If the generated captions are good enough we can generate the captions for the whole test dataset.

startseq man in white shirt is standing on the street endseq

startseq black and white dog is running on the grass endseq

startseq dog is running through the snow endseq

Results

BLEU Score

To evaluate our model, we will use BLEU score (Bilingual Evaluation Understudy score). Bleu score helps to measure how good a particular caption that is generated by the trained model is correct when compared to the original caption. The value is closer to 1 if the predicted is close to 1 and 0 if predicted to be wrong.

We initialize two lists, one for the actual captions and the other for the predicted captions

true: child in pink dress is climbing up set of stairs in an entry way

pred: boy in blue shirt is standing on the street

BLEU: 7.176794039009363e-232

true: black dog and spotted dog are fighting

pred: black and white dog is running through the water

BLEU: 1.384292958842266e-231

true: little girl covered in paint sits in front of painted rainbow with her hands in bowl

pred: man in red shirt is jumping off the water

BLEU: 4.832402486973385e-232

true: man lays on bench while his dog sits by him

pred: black and white dog is running in the water

BLEU: 9.412234823955334e-232

true: man in an orange hat starring at something

pred: man and woman are standing on the camera

BLEU: 1.0832677820940877e-231

By the BLEU score that we got for these images we can say that these are bad images

true: black dog is running in the water

pred: black dog is running in the water

BLEU: 1.0

These are the good captions predicted for the image and the BLEU score is maximum.

References:

https://assets.researchsquare.com/files/rs-1282936/v1_covered.pdf?c=1655838167

https://www.researchgate.net/publication/357955678_IMAGE_CAPTION_GENERATOR_USING_DEEP_LEARNING-Convolutional_Neural_Network_Recurrent_Neural_Network_Bilingual_Evaluation_UnderstudyBLEU_scoreLong_Short_Time_Memory

https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b