Vivek Mankar
mankarvivek172000@gmail.com
Pune India

Automatic Image Captioning

26th October 2020

DEFINITION

OVERVIEW

Artificial Intelligence shows us the tremendous possibility to change our way of working in different fields. Take the "Snapshot Serengeti"[1] project, Where more than 200 automatic cameras across the Savannah have amassed millions of photos of animals. But wildlife Biologists have had trouble figuring out how to actually deal with all that information. Means those were just photos and they needed data (example, "Two lions are eating a buffalo" rather than an Image) which is easier to store and manipulate. Turning all those photos into data has taken more than **30,000 hours**. The scientists build a computer program using AI(which usages Image classification and image captioning) that completed the task in just **one day.**

In this project, we will develop and deploy a Deep Learning model that can automatically generate captions for the given image[2]. We will use the **Flickr8K** dataset[3] to train our model and streamlit package to develop a simple UI so that users can play with it.

PROBLEM STATEMENT

For a scientific project generating captions for images can become a tedious task and take years to complete. One of the examples is Project Snapshot Serengeti. To tackle such problems, to give professionals (from non-technical backgrounds ex. Biologists, Archaeologists, etc.) an IDEA of how artificial intelligence can improve their quality of work. Develop a simple web app where users can upload images to get automatically generated captions. Basically, the web app will use a deep learning model (trained on the **Flickr8K** dataset) that can generate captions for the given image.

METRICS

There are various ways to measure the performance of an image captioning model like BLEU, ROUGE, CIDEr, METEOR, SPICE, etc.[4] but out of these BLEU(Bilingual Evaluation Understudy) is most common and widely used in the evaluation of image annotation results, which is based on the n-gram precision.[5] For this project, we will be using the BLUE score to check and compare the performance of our model. The principle of the BLEU measure is to calculate the distance between the evaluated and the reference sentences. BLEU method tends to give a higher score when the caption is closest to the length of the reference statement.

Scores are calculated for individual translated segments—generally sentences—by comparing them with a set of good quality reference translations. Those scores are then averaged over the whole corpus to reach an estimate of the translation's overall quality. Intelligibility or grammatical correctness are not taken into account. BLEU's output is always a number between 0 and 1. This value indicates how similar the candidate text is to the reference texts, with values closer to 1 representing more similar texts. Few human translations will attain a score of 1, since this would indicate that the candidate is identical to one of the reference translations. For this reason, it is not necessary to attain a score of 1. Because there are more opportunities to match, adding additional reference translations will increase the BLEU score.[8]

ANALYSIS

DATA EXPLORATION

The dataset used for this project is the “**Flickr8K**” dataset[3].

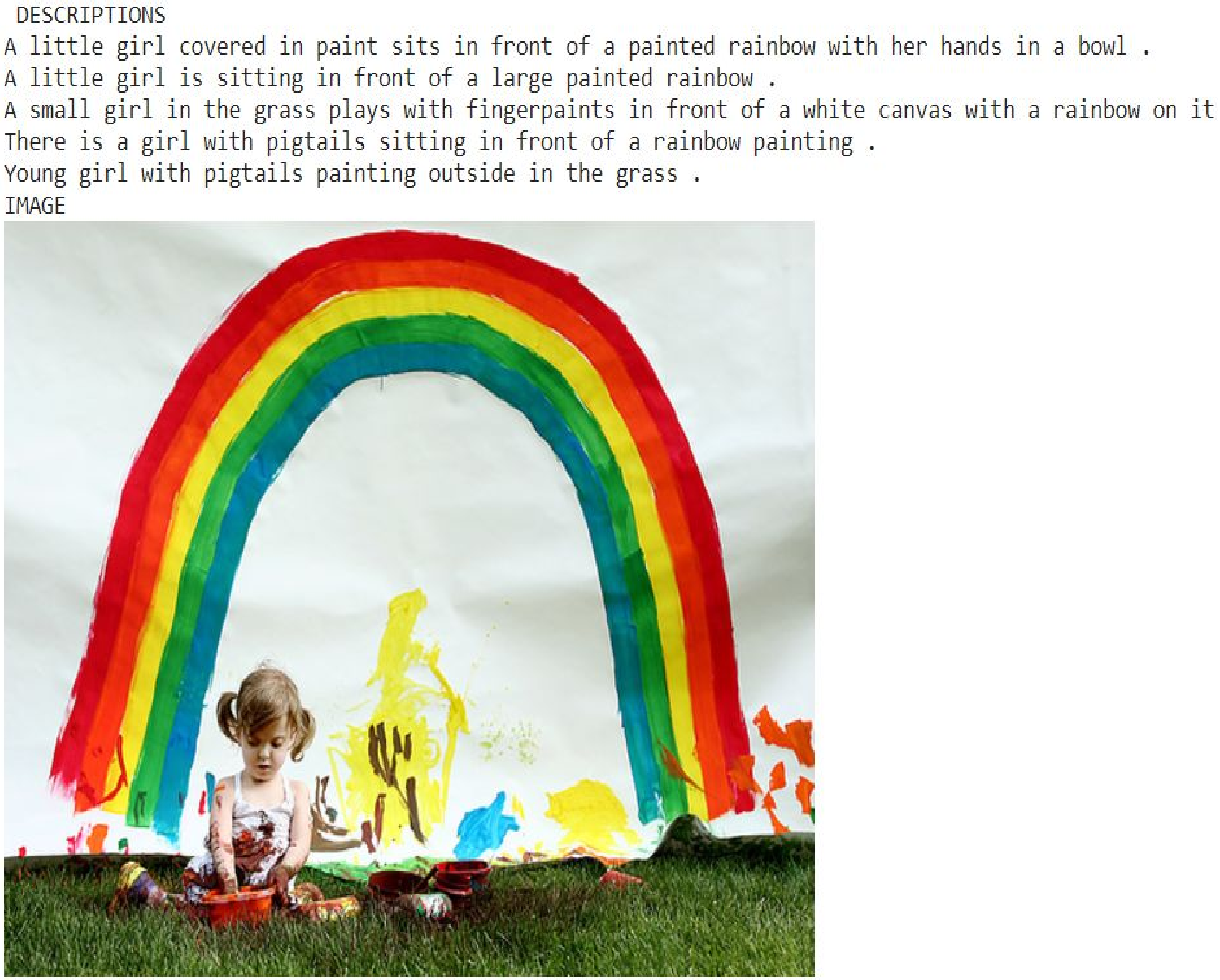
Flickr8K dataset includes images obtained from the Flickr web-site

- It is a labeled dataset.
- The dataset consists of **8000 photos**.
- All images are in JPEG format with different **shapes** and **sizes**. (Since we are using the VGG16 model to extract features from images, we will reshape all the images to the size 224*224 (check ref[11]))
- There are **5 captions** for each photo.
- The dataset can be found at Kaggle[3]
- The dataset is small and the size is 1.14 GB.

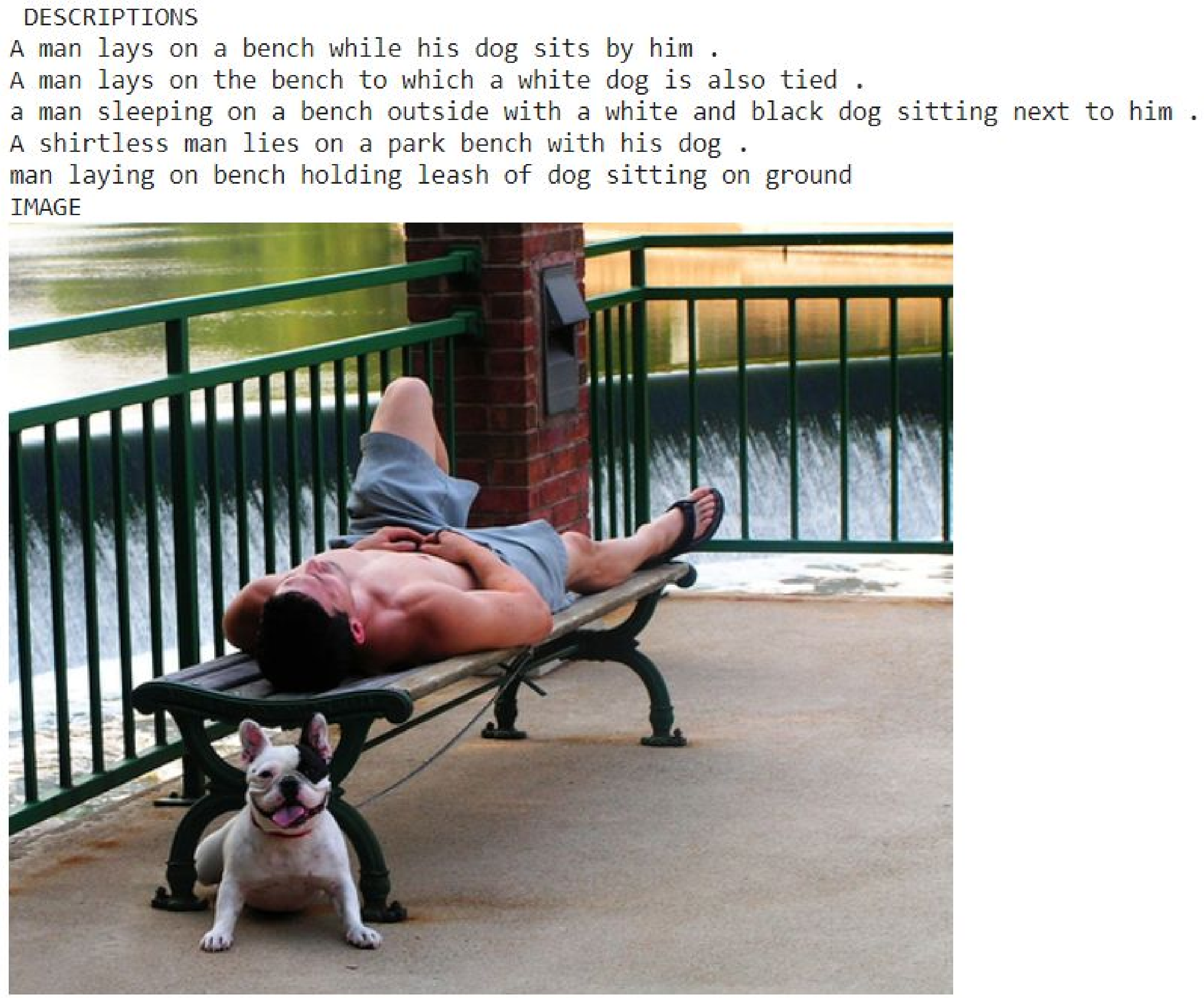
Thus this dataset is best for this project.

EXPLORATORY VISUALIZATION

There are 5 Captions for each image. Some of the examples are :



```
 DESCRIPTIONS
A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
A little girl is sitting in front of a large painted rainbow .
A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it
There is a girl with pigtails sitting in front of a rainbow painting .
Young girl with pigtails painting outside in the grass .
IMAGE
```



```
 DESCRIPTIONS
A man lays on a bench while his dog sits by him .
A man lays on the bench to which a white dog is also tied .
a man sleeping on a bench outside with a white and black dog sitting next to him .
A shirtless man lies on a park bench with his dog .
man laying on bench holding leash of dog sitting on ground
IMAGE
```

ALGORITHMS AND TECHNIQUES

We will use a combination of CNN(Convolutional Neural Network) and RNN(Recurrent Neural Network) to develop this system.

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNN's are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNN takes a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extremity.[9]

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models, and other sequence learning methods in numerous applications. The advantage of an LSTM cell compared to a common recurrent unit is its cell memory unit. The cell vector has the ability to encapsulate the notion of forgetting part of its previously-stored memory, as well as to add part of the new information. To illustrate this, one has to inspect the equations of the cell and the way it processes sequences under the hood.[10]

First, we will extract features of the image by using CNN, and then we will feed this feature vector to an LSTM language model that will generate captions. Thus we will have a CNN Encoder and an LSTM Decoder. (LSTM is a special kind of RNN, capable of learning long-term dependencies). We will also use pre-trained models on a standard Imagenet dataset(provided in Keras) to develop the CNN encoder. We will use pre-trained GLOVE 200d embeddings for words to improve the performance of our language model.

Thus this algorithm is a good example where we are using concepts like **Image Processing, Natural Language Processing,** and **Transfer learning.**

BENCHMARK

In the past few years, the problem of generating descriptive sentences automatically for images has garnered a rising interest in natural language processing and computer vision research. All the papers are focused on increasing the performance of the image captioning model. There are various ways to measure the performance of an image captioning model like BLEU, ROUGE, CIDEr, METEOR, SPICE, etc.[4] but out of these BLEU(Bilingual Evaluation Understudy) is most common and widely used in the evaluation of image annotation results, which is based on the n-gram precision. Details are described in the metrics section above.

Benchmark: Research paper published in Yahoo Research achieved an effective BLEU score of 0.683 for their model.[5]

METHODOLOGY

DATA PREPROCESSING

Since the Flickr8K dataset contains 8000 different photos with 5 descriptions for each photo. The preprocessing will involve both processing of images as well as texts.

We will first develop a feature extractor using the transfer Learning(VGG16 model) to extract features from the images and save them in a .pkl file.

Then we will move towards text processing, we will start with loading and cleaning the descriptions (tokenization, converting to lowercase, removing punctuation, etc) and then save the cleaned descriptions in a .txt file.

We will also create a vocabulary consisting of all the words in the descriptions.

IMPLEMENTATION

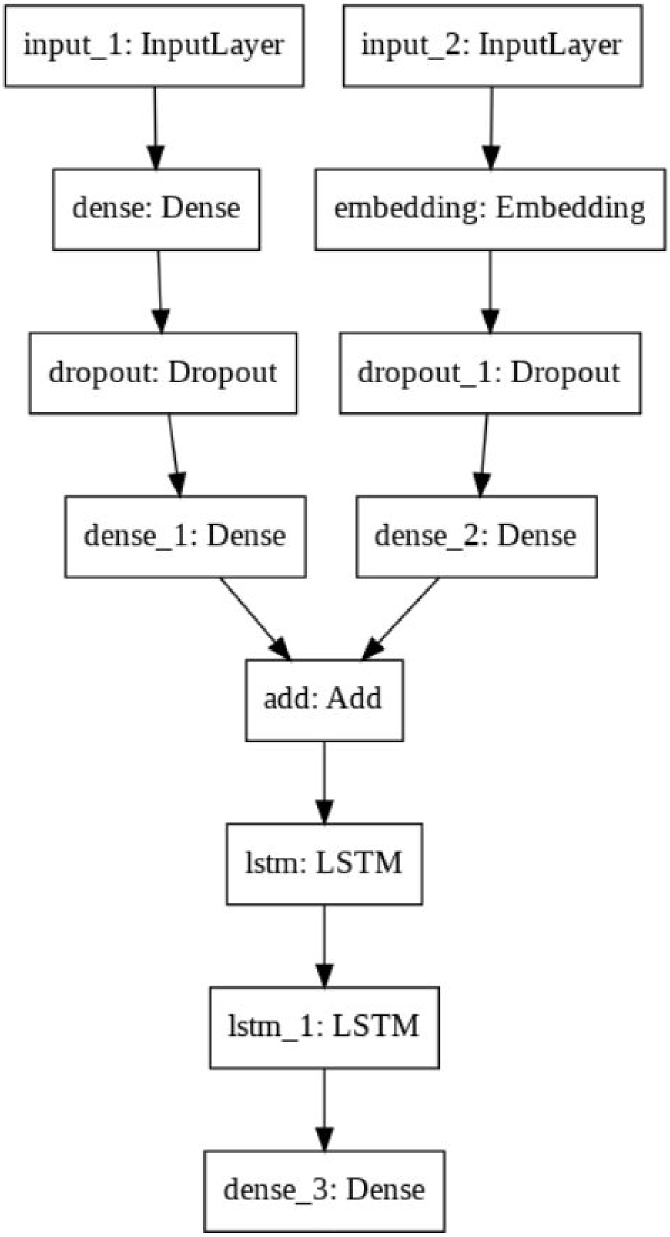
We will first split data into training and testing sets.

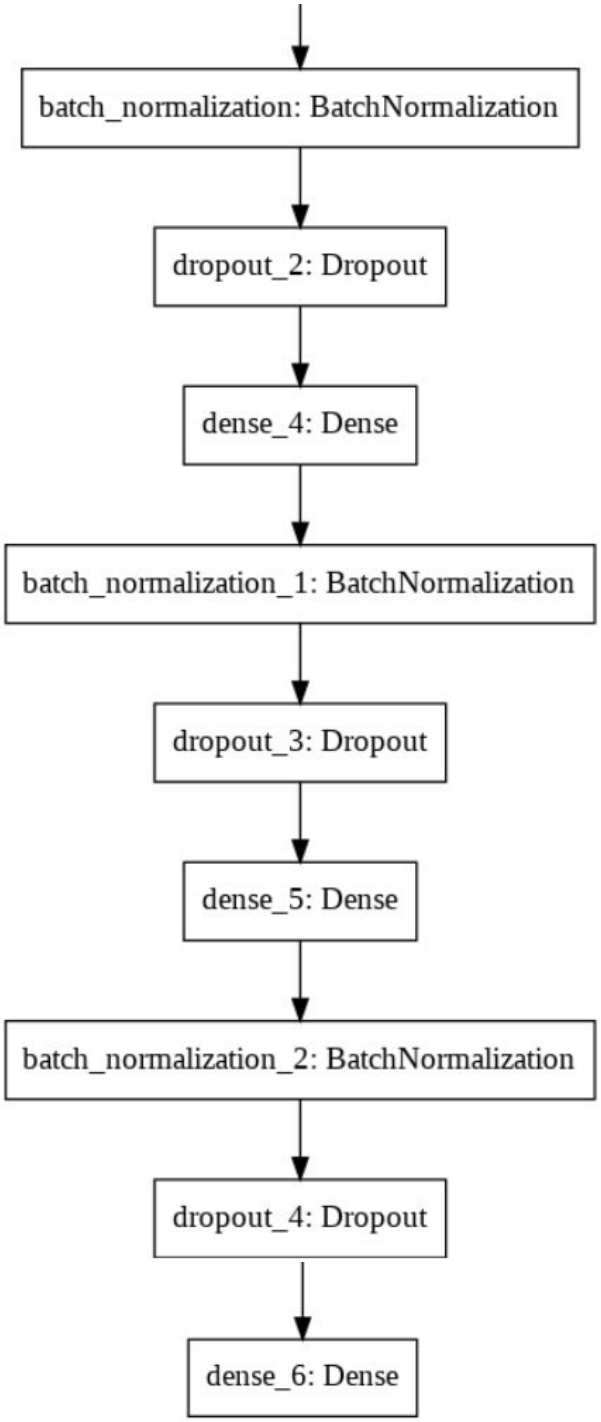
Then we will develop a deep learning model that will take an image and an input_sequence(since we are training a language model) as inputs and will output a sequence of words(a caption for the given image).

We will use Keras Functional API to develop the model. As described in the Algorithms and techniques section, first, we will extract features of the image by using CNN, and then we will feed this feature vector to an LSTM language model that will generate captions. Thus we will have a CNN Encoder and an LSTM Decoder.

Doing this directly i.e. extracting the features and training the Language Model requires a lot of Computational Power so **the trick** is extracting the features from all the images prior to the training and saving it on some location.

The architecture of the final Deep Learning model:




batch_normalization: BatchNormalization
dropout_2: Dropout
dense_4: Dense
batch_normalization_1: BatchNormalization
dropout_3: Dropout
dense_5: Dense
batch_normalization_2: BatchNormalization
dropout_4: Dropout
dense_6: Dense

The training will be done on google Colab or AWS Sagemaker to achieve faster results.

Then we will save the model weights to use them in our final **web application**.

REFINEMENT

Since it's a deep learning model there is always room for improvement considering the model architecture, size of the dataset, etc.

The VGG16 model is used to extract features of the images. One Can try Improving the feature extractor model ex. Using VGG19 rather than VGG16 or something else.

With the improvement in the data preprocessing techniques specially text-preprocessing we may achieve better results.

Since the data set is small in size and most of the deep learning model requires a large dataset. With an increase in the dataset size, we will see improvements in the performance.

RESULTS

MODEL EVALUATION AND VALIDATION

After training and hyperparameter turning (such as changing the no of hidden layers, increasing the no of training epochs, etc.) We have achieved an **effective BLEU score of 0.55** [BLEU's output is always a number between 0 and 1. This value indicates how similar the predicted caption is to the actual caption, with values closer to 1 representing more similar texts.]

To calculate the BLEU score we have used *nltk.translate.bleu_score.corpus_bleu()* method, where we pass the actual and the predicted captions as arguments.

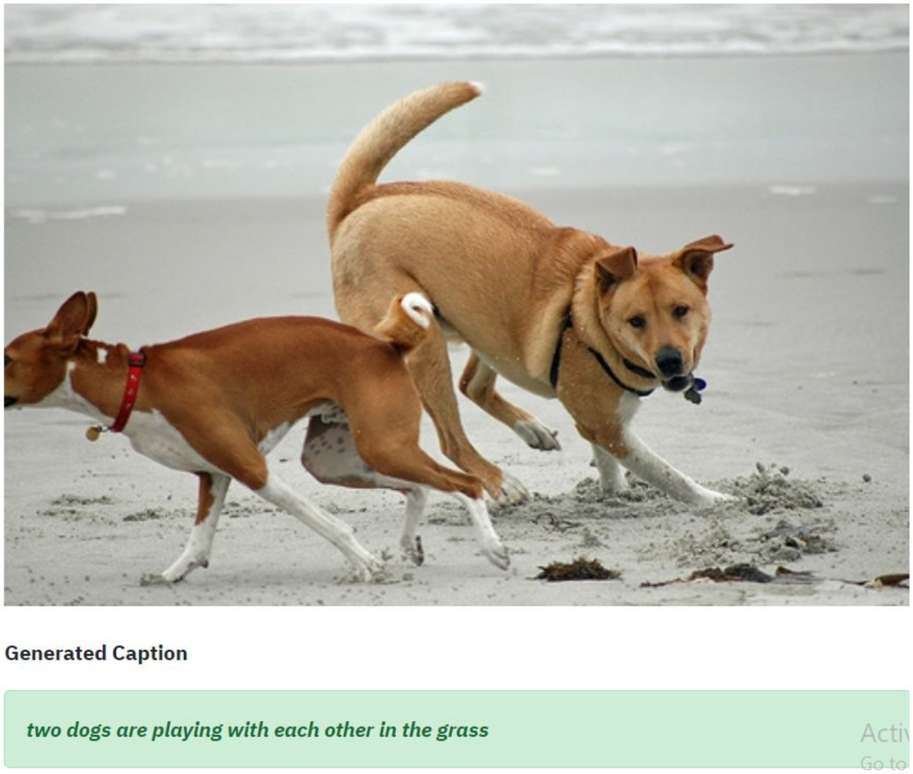
Since the model is trained on a small number of images(8000) and a small vocabulary(8763 words), the robustness of the model can be improved by increasing the dataset size(ex. Adding some noise to the images) and changing the model parameters accordingly.

JUSTIFICATION

We have achieved the BLEU score of 0.55 which is a little less than our benchmark model 0.68. BLEU method tends to give a higher score when the caption is closest to the length of the reference statement. One of the reasons could be the size of the dataset because of the computational limitations from our end. To increase the robustness and accuracy of our model the MS-COCO dataset can be used rather than the flickr8K dataset. Because the

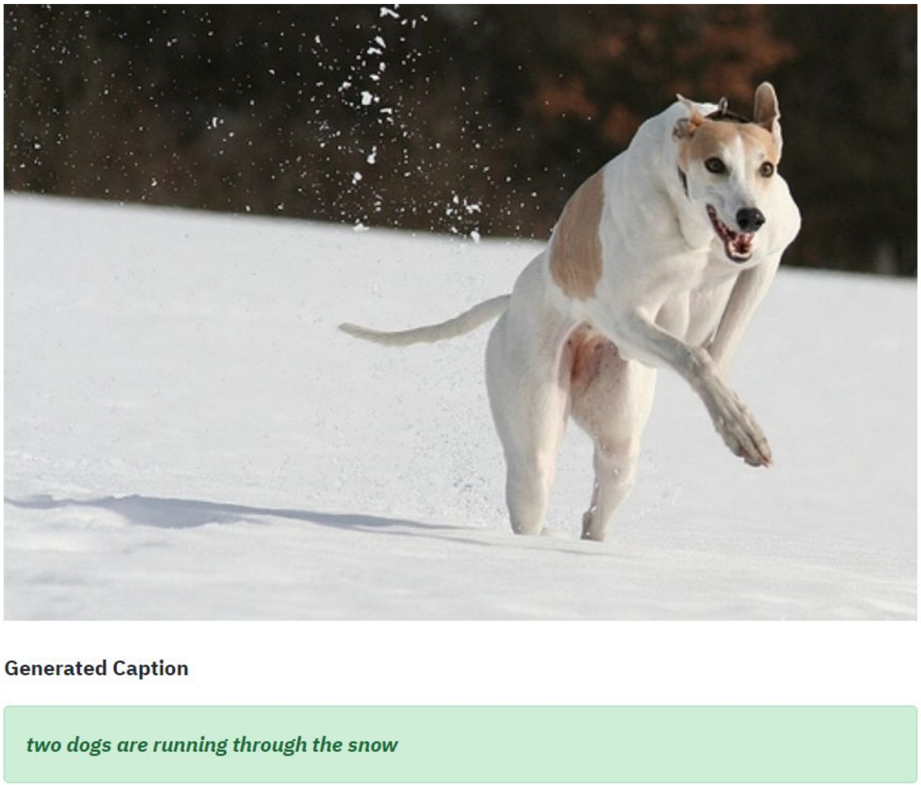
MS COCO dataset contains non-iconic images, unlike other datasets and the dataset has 2,500,000 labeled instances in 328,000 images while the flickr8K dataset contains only 8000 labeled images.Thus the model is generating good captions for the given images but the performance of the model can surely be increased if we have large computational power by increasing the dataset, adding noisy images to increase robustness, etc.

The performance of the model is :



Generated Caption

two dogs are playing with each other in the grass



Generated Caption

two dogs are running through the snow

REFERENCES:

[1]

https://www.zooniverse.org/projects/zooniverse/snapshot-serengeti

[2]

https://www.youtube.com/watch?v=tSoqJpisKIg

[3]

https://www.kaggle.com/shadabhussain/flickr8k

[4]

Shuang Liu, Liang Bai,a, Yanli Hu and Haoran Wang College of Systems Engineering, National University of Defense Technology,410073 Changsha, China

[5]

Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares :Image Captioning: Transforming Objects into Words,Yahoo Research San Francisco, CA, 94103

[6]

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, Serge Belongie,Department of Computer Science, Cornell University
https://vision.cornell.edu/se3/wp-content/uploads/2018/03/1501.pdf

[7]

Lakshminarasimhan Srinivasan, Dinesh Sreekanthan,Amutha A.L : Image Captioning - A Deep Learning Approach, https://www.ripublication.com/ijaer18/ijaerv13n9_102.pdf

[8]
https://en.wikipedia.org/wiki/BLEU#:~:text=BLEU%20(bilingual%20evaluation%20understudy)%20is,one%20natural%20language%20to%20another.&text=Scores%20are%20calculated%20for%20individual,of%20good%20quality%20reference%20translations.

[9]

https://en.wikipedia.org/wiki/Convolutional_neural_network

[10]

https://en.wikipedia.org/wiki/Long_short-term_memory

[11]

Karen Simonyan & Andrew Zisserman. Visual Geometry Group, Department of Engineering Science, University of Oxford: VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION