Visual Image Caption Generator Using Deep Learning

Grishma Sharma
Asst. Professor of Department Of Computer Engineering
K.J Somaiya College Of Engineering, Mumbai
neelammotwani@somaiya.edu

Priyanka Kalena
Department Of Computer Engineering
K.J Somaiya College Of Engineering, Mumbai
kalenapriyanka@gmail.com

Nishi Malde
Department Of Computer Engineering
K.J Somaiya College Of Engineering, Mumbai
nshmalde97@gmail.com

Aromal Nair
Department Of Computer Engineering
K.J Somaiya College Of Engineering, Mumbai
aromaln31197@gmail.com

Saurabh Parkar
Department Of Computer Engineering
K.J Somaiya College Of Engineering, Mumbai
saurabh.parkar@somaiya.edu

***Abstract* - Image Caption Generation has always been a study of great interest to the researchers in the Artificial Intelligence department. Being able to program a machine to accurately describe an image or an environment like an average human has major applications in the field of robotic vision, business and many more. This has been a challenging task in the field of artificial intelligence throughout the years. In this paper, we present different image caption generating models based on deep neural networks, focusing on the various RNN techniques and analyzing their influence on the sentence generation. We have also generated captions for sample images and compared the different feature extraction and encoder models to analyse which model gives better accuracy and generates the desired results.**

Keywords - CNN, RNN, LSTM , VGG, GRU, Encoder - Decoder.

I. INTRODUCTION

Generating accurate captions for an image has remained as one of the major challenges in Artificial Intelligence with plenty of applications ranging from robotic vision to helping the visually impaired. Long term applications also involve providing accurate captions for videos in scenarios such as security system. “Image caption generator”: the name itself suggests that we aim to build an optimal system which can generate semantically and grammatically accurate captions for an image. Researchers have been involved in finding an efficient way to make better predictions, therefore we have discussed a few methods to achieve good results. We have used the deep neural networks and machine learning techniques to build a good model. We have used Flickr 8k dataset which contains around 8000 sample images with their five captions for each image. There are two phases : feature extraction from the image using Convolutional Neural Networks (CNN) and generating sentences in natural language based on the image using Recurrent Neural Networks (RNN). For the first phase, rather than just detecting the objects present in the image, we have used a different approach of extracting features of an image which will give us details of even the slightest difference between two similar images. We have used VGG-16 (Visual Geometry Group) , which is a 16 convolutional layers model used for object recognition. For the second phase, we need to train our features with captions provided in the dataset. We are using two architectures LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) for framing our sentences from the input images given. To get an estimation of which architecture is better we have used the BLEU (Bilingual Evaluation

Understudy) score to compare the performances between LSTM and GRU.

II. LITERATURE SURVEY

There have been several attempts at providing a solution to this problem including template based solutions which used image classification i.e. assigning labels to objects from a fixed set of classes and inserting them into a sample template sentence. But more recent work have focused on Recurrent Neural Networks [2,5]. RNNs are already quite popular with several Natural Language Processing tasks such as machine translation where a sequence of words is generated. Image caption generator extends the same application by generating a description for an image word by word.

The computer vision reads an image considering it as a two dimensional array. Therefore, Venugopalan (et al)[9] has described image captioning as a language translation problem. Previously language translation was complicated and included several different tasks but the recent work[10] has shown that the task can be achieved in a much efficient way using Recurrent Neural Networks. But, regular RNNs suffer from the vanishing gradient problem which was vital in case of our application. The solution for the problem is to use LSTMs and GRUs which contain internal mechanisms and logic gates that retain information for a longer time and pass only useful information.

One of the major challenges we faced was choosing the right model for the caption generation network. In their research paper, Tanti (et al)[8] has classified the generative models into two kinds – inject and merge architectures. In the former, we input both, the tokenized captions and image vectors to an RNN block whereas in the latter, we input only the captions to the RNN block and merge the output with the image. Although the experiments show that there is not much difference in the accuracy of the two models, we decided to go with the merge architecture for the simplicity of its design, leading to reduction in the hidden states and faster training. Also, since the images are not passed iteratively through the RNN network, it makes better use of RNN memory.

III. METHODOLOGY

The complete system is a combination of three models which optimizes the whole procedure of caption description from an image. The models are (a) Feature Extraction Model (b) Encoder Model (c)Decoder model.

A. Feature Extraction Model

This model is primarily responsible for acquiring features from an image for training. When the training begins the features of the images are input to this model.

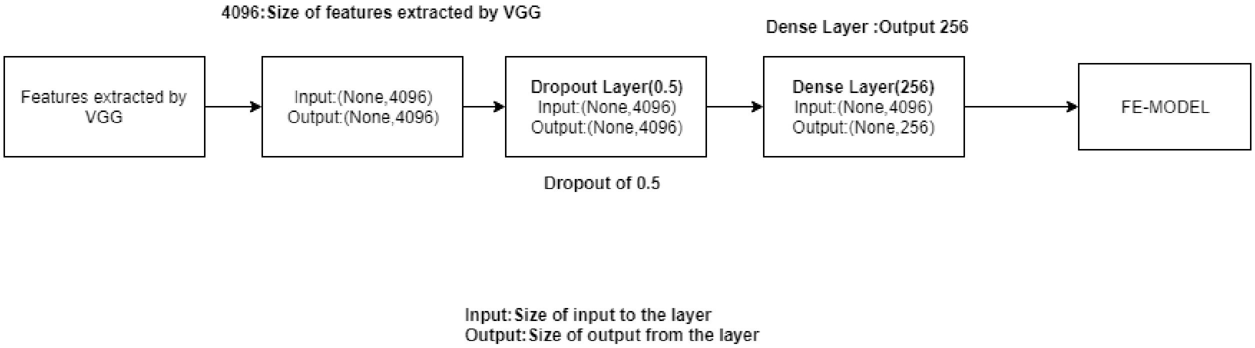


Fig 1. Feature Extraction Model

The model uses a VGG16 architecture as seen in Fig.1, to efficiently extract the features from the images using a combination of multiple 3*3 convolution layers and max pooling layers. The output of a VGG16 network would be vectors of size 1*4096, which are used to represent the features of the images.

A dropout layer is added to the model with a value of 0.5 to reduce overfitting. An optimal value is between 0.5 to 0.8 which indicates the probability at which the outputs of the layer are dropped out.

A dense layer is added after the dropout layer which basically applies the activation function on the input, kernel with a bias. The activation function used is ‘ReLU’(Rectified Linear Units) and the size of output space is specified as 256. These vectors of size 256 are the output of the feature extraction model which will then be used in the decoder model.

B. Encoder Model

The encoder model, as seen in Fig. 2, is primarily responsible for processing the captions of each image fed while training. The output of the encoder model is again vectors of size 1*256 which would again be an input to the decoder sequences.

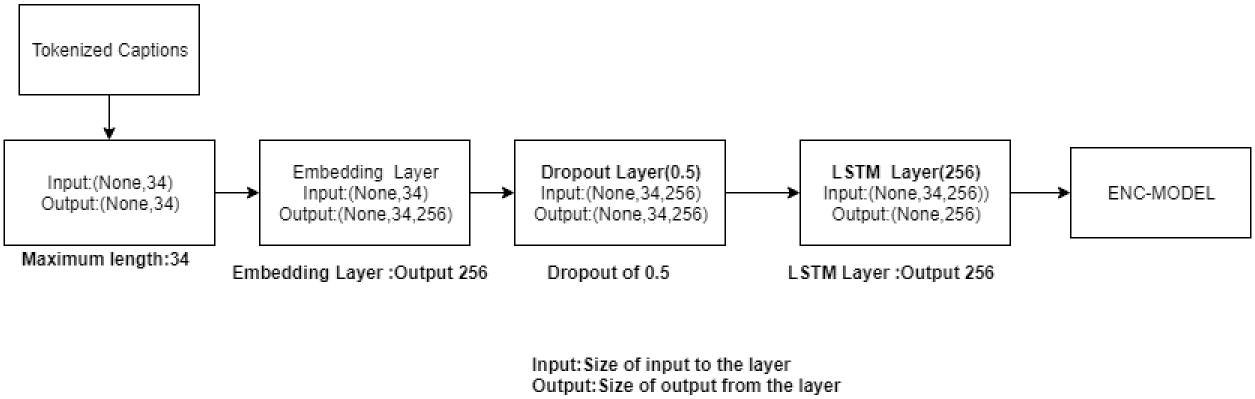


Fig 2. Encoder Model

Initially the captions present with each images are tokenized ie the words in the sentences are converted to integers so that the neural network can process them efficiently. The tokenized captions are padded so that the length is equal to the size of the longest sentence and all the sentences can be processed at an equal length.

Then an Embedding layer is attached to embed the tokenized captions into fixed dense vectors with an output space of 256 by 34. 34 is chosen as the maximum number of words in all the captions of Flickr8k dataset is 34. These vectors would further ease out the processing by providing a convenient way of representing the words in the vector space. A dropout layer is attached with again a probability of 0.5 to reduce the overfitting in the model.

The most important part of the Encoder model is the LSTM layer or Long Short Term Memory Layer. This layer helps the model in learning how to generate valid sentences or generating the word with highest probability of occurrence after a specific word is encountered. The activation function used is ReLU, a linear activation function and the output space defined is 256.

For comparison between the complete models VGG+GRU and VGG+LSTM this particular layer will be replaced by a GRU or Gated Recurrent Units layer, and results will be analyzed for the same .The output space for the GRU layer is same i.e 256 . Thus the only major difference between the two models will be in the encoder part. The output of the LSTM layer is our output for the encoder layer.

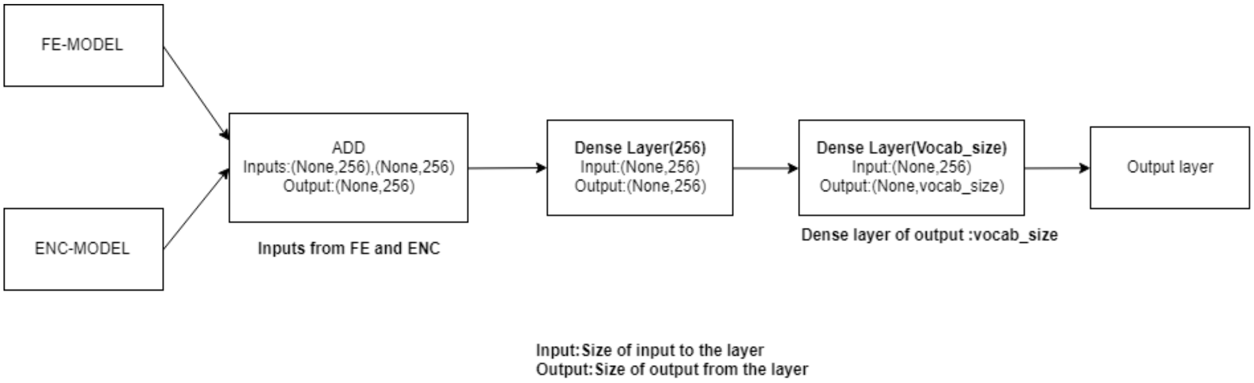


Fig 3. Final Decoder Model

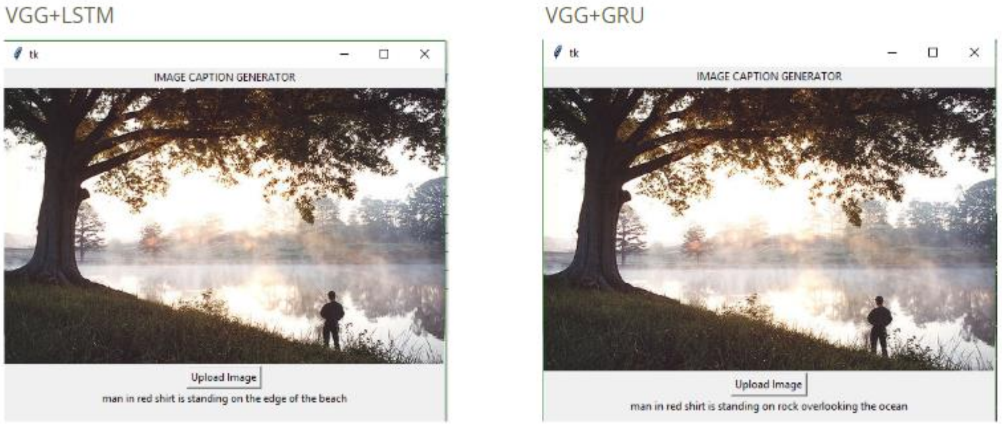
C. *Decoder Model*

The decoder model, as shown in Fig. 3, is basically the model which concatenates both the feature extraction model and encoder model and produces the required output which is the predicted word given an image and the sentence generated till that point of time.

As shown in the above diagram the Decoder model takes in the input from the Feature extraction model and the encoder model both of which outputs vectors of dimension 256. The output from the concatenated models are passed through a dense layer which uses the 'ReLU' activation function. Another Dense Layer is added to the decoder model with the vocabulary size as the output space. The vocabulary size in Flickr 8k was found to be 7579 and the activation function used was softmax activation which basically outputs a word for the integer predicted. The predicted word is the output of the decoder layer.The model is trained by the following input output parameters:
<ip>=<image,<in-seq>
<op>=<word>
Where input parameters are the image and the input sequence and the output of the model is the word predicted provided the model has the image and the caption generated till that point of time.

When the caption is generated we calculate the bleu score for each architecture. Four types of bleu scores were found out : BLEU -1 (1.0, 0, 0, 0), BLEU -2 (0.5, 0.5, 0, 0), BLEU -3 (0.33, 0.33, 0.33, 0) and BLEU -4 (0.25, 0.25, 0.25, 0.25). We have used the cumulative weights since they give better output.

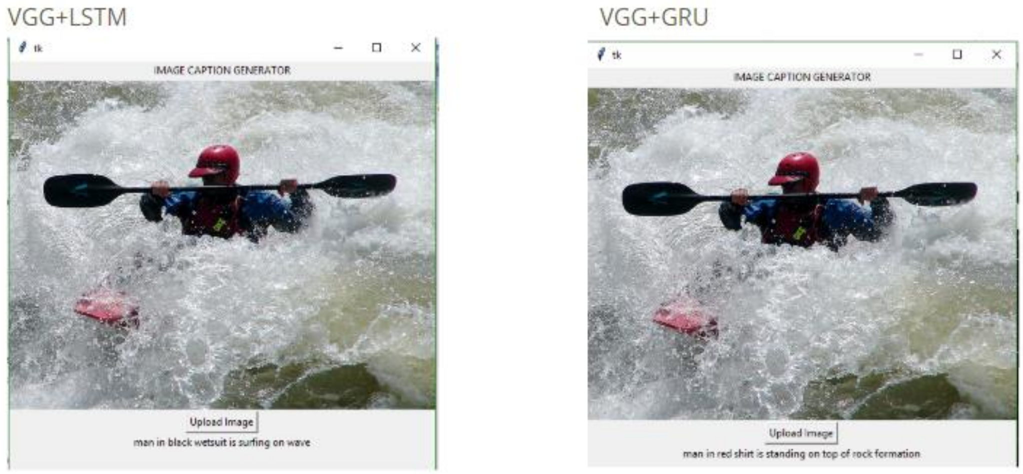
IV. ANALYSIS



man in red shirt is standing on the edge of the beach

man in red shirt is standing on rock overlooking the ocean

Fig 4. Comparison of caption for Sample Image 1.


VGG+LSTM

man in black wetsuit is surfing on wave

VGG+GRU

man in red shirt is standing on top of rock formation

Fig 5. Comparison of caption for Sample Image 2.

Fig 4 and Fig 5 are some example images on which the testing was done. We have tested various images with both the methods ie VGG+LSTM and VGG+GRU. The training of the models was done on Google Colab which provides the 1xTesla K80 GPU with 12GB GDDR5 VRAM and took approximately 13 minutes per epoch for LSTM and 10 minutes per epoch for GRU. This happens due to the lesser amount of operations occurring in GRU than LSTM. While the loss calculated for LSTM was less than GRU, the user can prefer any model according to his need, either with maximum accuracy or one which takes lesser time to process. GRUs generally train faster on less training data than LSTMs and are simpler and easy to modify .

V. CONCLUSION

We have presented a deep learning model that tends to automatically generate image captions with the goal of not only describing the surrounding environment but also helping visually impaired people better understand their environments. Our described model is based upon a CNN feature extraction model that encodes an image into a vector representation, followed by a RNN decoder model that generates corresponding sentences based on the image features learned . We have compared various encoder decoder models to see how each component influences the caption generation and have also demonstrated various use cases on our system. The results show that LSTM model generally works slightly better than GRU although taking more time for training and sentence generation due to its complexity. The performance is also expected to increase on using a bigger dataset by training on more number of images. Because of the considerable accuracy of the generated image captions, visually impaired people can greatly benefit and get a better sense of their surroundings using the text-to-speech technology that we have incorporated as well.

VI. FUTURE WORK

Our model is not perfect and may generate incorrect captions sometimes. In the next phase, we will be developing models which will use Inceptionv3 instead of VGG as the feature extractor. Then we will be comparing the 4 models thus obtained i.e. VGG+GRU, VGG+LSTM, Inceptionv3+GRU, and Inceptionv3+LSTM . This will further help us analyze the influence of the CNN component over the entire network.

Currently, we are using a greedy approach for generating the next word in the sequence by selecting one with the maximum probability. Beam search instead selects a group of words with the maximum likelihood and parallel searches through all the sequences. This approach might help us increase the accuracy of our predictions.

Our model is trained on the Flickr 8K dataset which is relatively small with less variety of images. We will be training our model on the Flickr30K and MSCOCO datasets which will help us to make better predictions. Other optimizations include tweaking the hyperparameters like batch size, number of epochs, learning rate etc and understanding the effect of each one of them on our model.

REFERENCES

[1] CS771 Project Image Captioning by Ankit Gupta , Kartik Hira, Bajaj Dilip.

[2] ”Every Picture Tells a Story: Generating Sentences from Images.” Computer Vision ECCV (2016) by Farhadi, Ali, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hocken-maier, and David Forsyth

[3] Automatic Caption Generation for News Images by Yansong Feng, and Mirella Lapata, IEEE (2013).

[4] Image Caption Generator Based on Deep Neural Networks by Jianhui Chen, Wenqiang Dong and Minchen Li, ACM (2014).

[5] Show and Tell: A Neural Image Caption Generator by Oriol Vinyal, Alexander Toshev, Samy Bengio, Dumitru Erhan, IEEE (2015).

[6] Image2Text: A Multimodal Caption Generator by Chang Liu, Changhu Wang, Fuchun Sun, Yong Rui, ACM (2016).

[7] The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions by Sepp Hochreiter.

[8] Where to put the Image in an Image Caption Generator by Marc Tanti, Albert Gatt, Kenneth P. Camilleri.

[9] Sequence to sequence -video to text by Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko.

[10] Learning phrase representations using RNN encoder-decoder for statistical machine translation by K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares,H. Schwenk, and Y. Bengi.

[11] TVPRNN for image caption generation .Liang Yang and Haifeng Hu.

[12] Image Captioning in the Wild: How People Caption Images on Flickr Philipp Blandfort, Tushar Karayil, Damian Borth, Andreas Dengel,German Institute for Articial Intelligence, Kaiserslautern, Germany.

[13] Image Caption Generator Based On Deep Neural Networks Jianhui Chen ,Wenqiang Dong, Minchen Li ,CS Department. ACM 2014.

[14] BLEU: A method for automatic evaluation of machine translation. InACL, 2002 by K. Papineni, S. Roukos, T. Ward, and W. J. Zhu.