Image Content Descriptor

Aarchi Agrawal*
aarchia@iitgn.ac.in
20250017

AthulKumar R*
athulkr@iitgn.ac.in
20250018

K S Ashin Shanly*
ksashin@iitgn.ac.in
20250019

Sudip Das*
sudipd@iitgn.ac.in
20210005

ABSTRACT

Image captions are those crisp descriptions that you see under images. Image captions generally provide the viewer a brief idea about the image. A tool used for generating captions for images based on their contents is called an "Image Content Descriptor". In recent years with the development of deep learning, the image captioning problem has risen in popularity. This problem requires a semantic understanding of images by the model, and the model generates an accurate description of the input image. Here we use a Resnet 101 encoder and an LSTM based decoder along with Attention and Beam Search. An adaptive learning rate with a shrinkage factor of 0.8 is employed. These techniques are added cumulatively on a baseline model, and the improvements brought about by each of them are recorded. The model is trained and tested on the Flicker8K dataset. We use the BLEU-4 score as the evaluation metric. The results are compared with the state-of-the-art models.

KEYWORDS

Attention, Beam Search, Long Short-Term Memory(LSTM), Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN)

1 INTRODUCTION

The process of generating descriptions for an image based on its contents is image captioning. Captioning images has uses in many fields, including self-driving cars, aiding visually impaired people, usage in virtual assistants, tagging in social media, and many more. Though they seem to be very simple, image captions are tricky, grabs much Attention compared to regular content. An average human being can effortlessly point out the features and describe them by looking at the picture. This relatively simple task for human beings is not that trivial for a visual recognition model.

In this work, we intend to create a visual recognition model that accepts an image and returns a caption generated based on the features in the image. Our model incorporates Resnet 101 encoder and LSTM based Decoder along with Attention, Beam Search, and adaptive learning rate with a shrinkage factor of 0.8. The model learns where to look with the help of the Attention mechanism. Beam search further optimizes the sequence of words in the caption. It provides us the best possible sequence by selecting the most appropriate word having the maximum score after each decoding step.

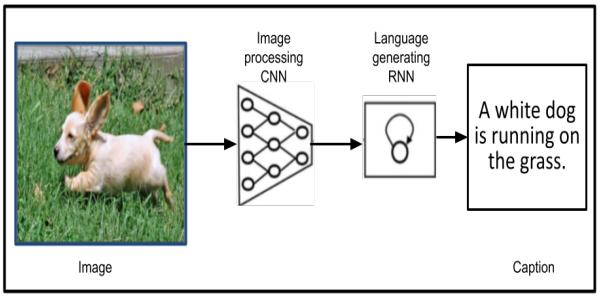


Figure 1: An image captioning system

2 RELATED WORKS

With the advancement in technology, a lot of work is being done in the field of image captioning. Before the rise of deep learning, image captioning was considered a hard problem. But now, many computer scientists are working towards the development of a variety of models for this particular task.

Most of the previous works use a combination of LSTM, CNN, and RNN. LSTM is used for dealing with caption generation. CNN is used as an encoder and for handling the image feature descriptor part. RNN acts as a decoder. Few works also use CNN as a decoder. In the paper by Shuang Liu et al.[3] they have done the task of image captioning by three approaches: CNN-RNN based approach, CNN-CNN based and, reinforcement-based framework. The work done by [4] considers the spatial relationship between the objects present in the image. For this, they have used "Object Relation Transformer."

The paper[5] has put forward a neural and probabilistic model for image captioning. They call their model Neural Image Caption or NIC. NIC is based on a convolution neural network that helps in encoding an image into a compact representation, followed by a recurrent neural network that generates a corresponding sentence. Their model is trained in such a way so that they can maximize the likelihood of the sentence given the image. In Andrej Karpathy work [1], they study the inter-model correspondences between language and visual data. Their model is a combination of CNN over image regions and a bidirectional RNN over sentences. This paper [2], is towards the efforts of combining the top-down and bottom-up attention mechanism. They have quantitatively shown that this approach is useful for calculating Attention at different parts of objects and identifying other essential features of the image.

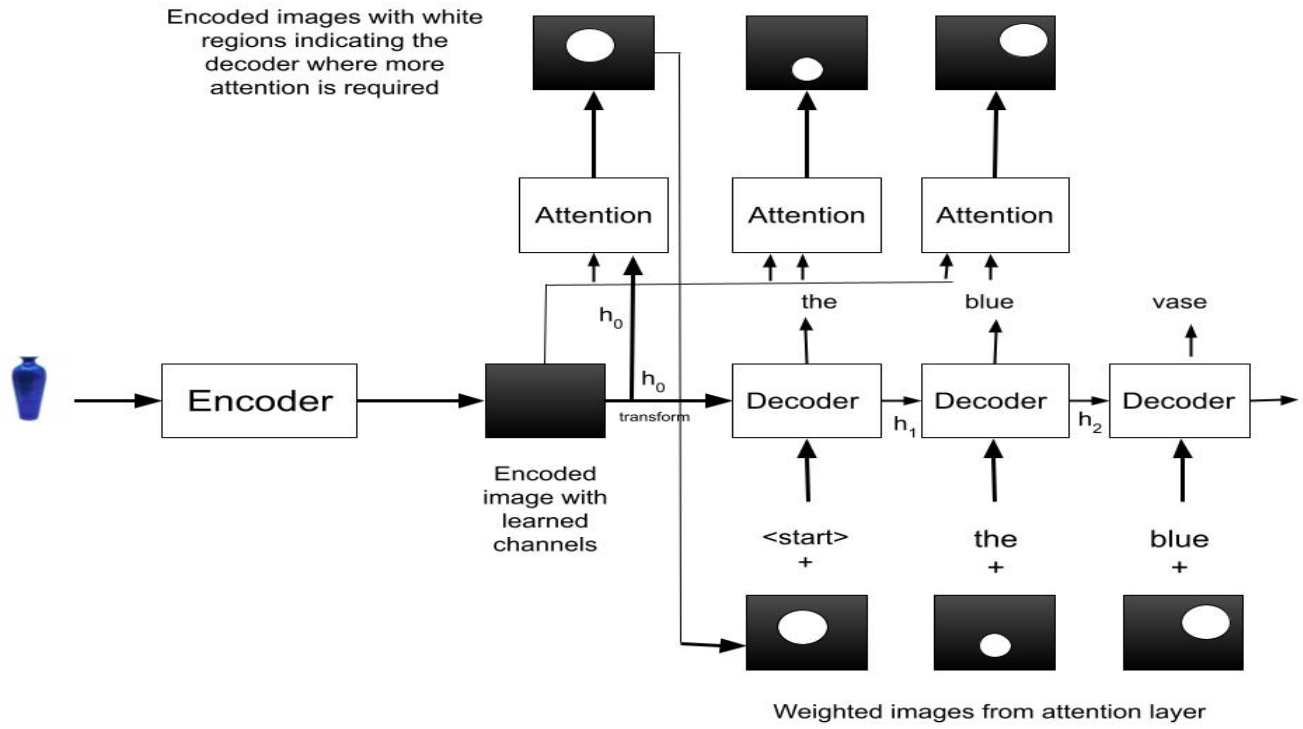


Figure 2: Attention mechanism

3 APPROACH

The model has an encoder-decoder architecture. The encoder initially encodes the input to a stable form which is then given as input to the decoder. The decoder then processes the input taking one word at a time and decodes it to generate a meaningful sequence of words. This sequence in our model is the required caption for the image provided.

We have used Resnet 101 as our encoder. It is used for encoding the input image into small-sized images with learned channels. Encoded small images tell us about the prominent features of the actual image in a concise way. Generally, for encoding images, Convolutional Neural Networks (CNNs) are used, so instead of training an encoder from scratch, we have used Resnet 101 as the encoder for our model. It is a 101 layered Residual network that has been trained for the classification task on the ImageNet data set.

Once the encoded image is obtained, the decoder comes into action. The decoder takes the encoded image as input, and then word by word generates the caption for the image. The model uses a Long term short memory (LSTM) for handling the caption generation task. Further, an attention mechanism is implemented on top of this, which helps the model choose the parts of the image which are of higher importance. In the task of image captioning, some pixels hold more importance compared to others. In the absence of an attention mechanism, the model simply takes the average of all the pixels for the encoded image. But with Attention, the Decoder learns where to look, and more Attention is given to relevant pixels. It is wise to use weighted average across pixels rather than simple averaging by giving higher weights to pixels of higher importance. The model then adds the weighted representation to the word generated before to generate the words ahead.

Furthermore, beam search is incorporated into the model. This is useful as it helps the Decoder select the optimum word sequence in the caption by choosing the word with the optimum score at each decoding step. By using a linear layer, the output of the Decoder is converted to a word score. The greedy approach selects the word having a maximum score to predict the next word. However, this is not the best approach as the whole sequence ahead depends on choosing the leading word. If the best choice is not made at this stage, then the rest of the options will also be sub-optimal. This is true for all the words that act as a predecessor for other words yet to come.

For performing the task of optimum selection, assuming things before the completion of the decoding phase is not recommended. Once the decoding is done, ideally, the sequence with the greatest overall score from a list of options of potential sequences would be selected. Beam Search helps in performing this specific task. Based on the beam width, beam search examine various best options with the help of conditional probability, and this gives better results than the sub-optimal Greedy search. In this way, the decoder functions and chooses the most optimal sequence.

4 EXPERIMENTAL SETUP

We will discuss the dataset we used and the evaluation metric used to compare our model's performance with the state-of-the-art models in this section.

4.1 Dataset

There are many open-source datasets available for this problem, like *the Pascal VOC dataset, Flickr8K, Flickr30K* and, *MSCOCO Dataset.* For this problem, we will be using Flickr8K dataset. As the name suggests, this dataset contains 8092 images from Flickr. In the Flickr8K dataset, each image has five captions. Hence 40460 manually annotated captions. We divided the dataset into 6000, 1000, 1000 images for training, validation, and testing, respectively. Fig 3 shows a graph indicating the length of the captions in the dataset.

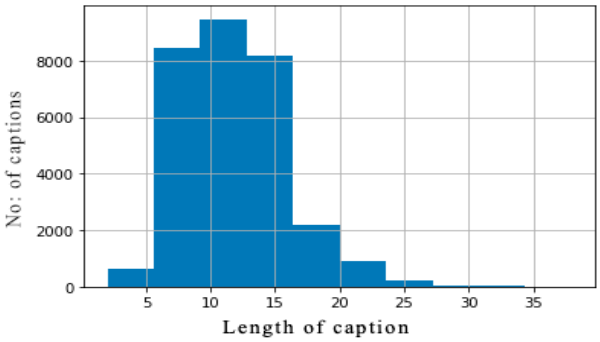


Figure 3: Length of the captions in Flickr8k dataset.

4.2 Evaluation Metrics

For evaluating the generated image and caption pairs, it is necessary to evaluate the model's capability to associate previously unseen images and descriptions with each other. We are using the Bilingual Evaluation Understudy (BLEU) score to evaluate the caption generated. BLEU is an algorithm that evaluates the quality of text translated by a machine. It describes how program generated natural sentence is compared to a human-generated sentence. The n-grams of the generated captions are compared with the n-grams of the actual reference captions. Then a score is calculated which is in the range of 0 to 1. Good scores are higher and close to 1. We are using BLEU scores for 4-grams (also termed as BLEU-4) as a metric since it makes a stricter evaluation compared to 1-grams, 2-grams, and 3-grams and is also the most expressive.

5 RESULTS

We present the results that we obtained for our model in this section.

5.1 Training Details

Our model is trained on 6000 images. As the number of images was low, we faced the problem of overfitting on the training set. To counter this, we set a dropout of 0.15. The embedding space for the Resnet101 encoder for our model is 2048 and for the LSTM decoder it is 512 dimensions. We used a batch size of 32. We also used an adaptive learning rate. The initial learning rate is $3 * 10^{-5}$. If we see no improvement in training accuracy in the next epoch, we shrink the learning rate by a factor of 0.8. If the training accuracy stops improving even after decaying the learning rate for four consecutive epochs, we stop the model training process. With this method, the model stopped training after 16 epochs.

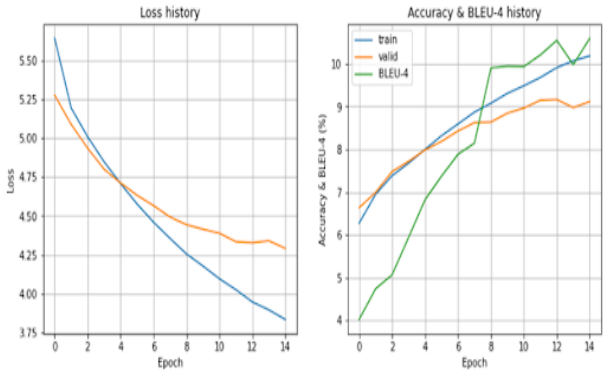


Figure 4: a. Training and Validation loss v/s number of epochs b.Improvement of train, test and validation accuracy v/s number of epochs

5.2 Generated Results

- Our model predicted meaningful captions more often than not. However, the model got confused due to the lack of tokens in the dictionary in some cases. It often gave unknown tokens as output.
- Our model achieved a BLEU-4 score of 0.15790 after training for 15 epochs. The training and validation losses and the BLEU-4 scores achieved over each epoch were recorded (Fig 4).
- We recorded the BLEU-4 scores after using different beam widths (Fig 5). Maximum BLEU-4 score was obtained with beam width = 14. Fig 6 shows some of the cases where our model predicted meaningful captions. An image where the model got confused and predicted some absurd captions is shown in Fig 7.

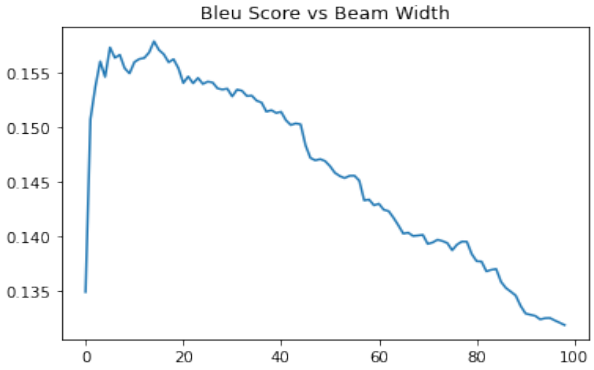


Figure 5: Beam width v/s BLEU-4 Score

Captions in Dataset

A playful dog is running. through the grass.

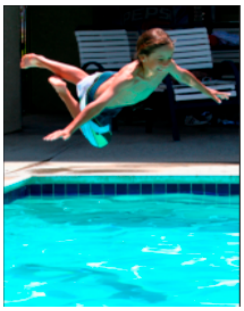
A small dog ears stick up as it runs in the grass.

The small dog is running across the lawn.

This is a small beige dog running through a grassy field

Caption Generated by the Model

A white dog is running on a grass.



Captions in Dataset

A boy is diving through the air into a swimming pool

A boy jumps into a pool on a bright sunny day

A boy jumps into the blue pool water

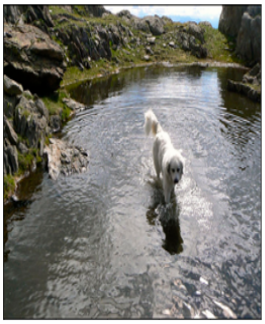
A smiling child dives into a pool

THe young boy is jumping into the pool on his belly

Caption Generated by the Model

A little boy is jumping in a pool

Figure 6: Two sample images with the actual captions in the dataset and the captions generated by the model.



Captions in Dataset

A dog in some water enclosed by a rocky ledge

A white dog is standing in a pond

A white dog walks through shallow water in the wildernes

A white dog is wading in the pond atop the mountain

A white dog is walking through a river that runs through a rocky landscape

Caption Generated by the Model

A white dog the the the a in

Figure 7: Sample image with the actual captions in the dataset and the captions generated by the model.

6 DISCUSSION

Our model attained a BLEU-4 score of 0.15784 after training for 15 epochs. Though this is far less compared to the state-of-the-art models by Ting Yao et al. [6] (BLUE-4 = 0.559) and Quanzeng et al. [7] (BLUE-4 = 0.412), it will increase if it is trained for more number of epochs. These models were trained on MSCOCO and Flickr30K datasets, respectively. We expect our model to give a higher BLEU score when we train it on larger datasets like MSCOCO or Flickr30K. In the future we intend to use non-parametric Kernel Activation Function (KAF) in Graph Convolutional Networks (GCN) along with LSTM for generating captions for images. We also wish to extend our work to generating some 'hashtags', which surely is an inevitable trend in today's social media.

7 CONCLUSION

Our Image Content Descriptor using Resnet-101 encoder and an LSTM based encoder with Attention and Beam search generated a meaningful caption most of the time given an input image. We evaluated the accuracy of the model using the BLEU-4 score. Although the state-of-the-art results could not be achieved, experiments using Flickr8K dataset gave reasonable results. The model can give better results if trained on a more extensive dataset like Flickr30k or the MSCOCO. We observed that the vocabulary of the captions is different for different datasets. However, it is same for the Flickr8K and Flickr30k as the images was annotated by the same group of annotators. Also with the help of transfer learning, we can load the weights on Flickr8K network. Then this network would certainly perform better, as the model will be trained on a bigger dataset.

REFERENCES

[1] Andrej Karpathy and Li Fei-Fei. 2015. Deep Visual-Semantic Alignments for Generating Image Descriptions. arXiv:1412.2306 [cs.CV]
[2] Chris Buehler Damien Teney Mark Johnson Stephen Gould Lei Zhang Peter Anderson, Xiaodong He. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering.
[3] Yanli Hu Shuang Liu, Liang Bai and Haoran Wang. 2018. Image Captioning Based on Deep Neural Networks.
[4] Kofi Boakye Joao Soares Simao Herdade, Armin Kappeler. 2019. Image Captioning: Transforming Objects into Words.
[5] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. arXiv:1411.4555v2 [cs.CV]
[6] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2016. Boosting Image Captioning with Attributes. arXiv:1611.01646 [cs.CV]
[7] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image Captioning with Semantic Attention. arXiv:1603.03925 [cs.CV]

APPENDIX A

We have created a simple web application to showcase our model as well. A sample view of the web app is given in Fig 8.

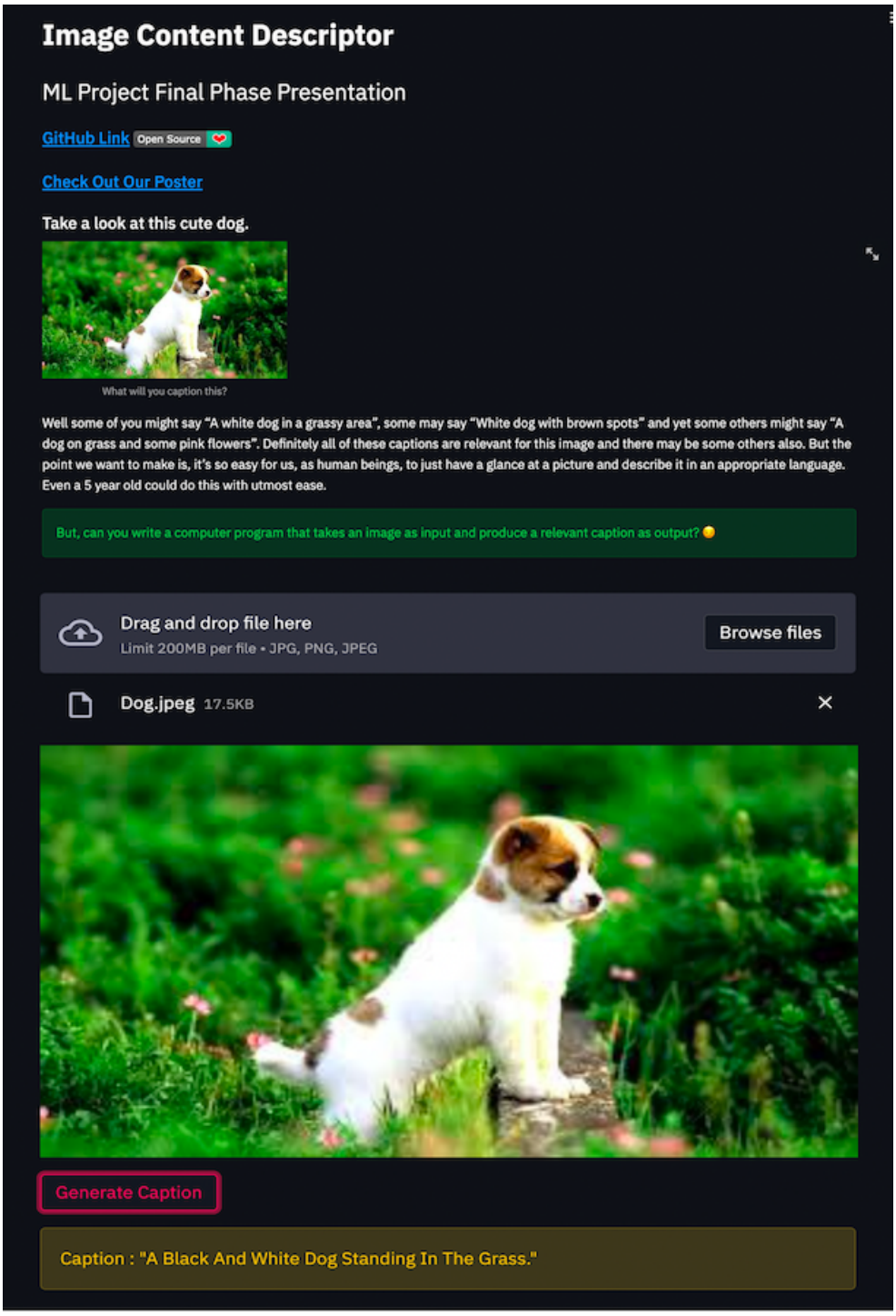


Figure 8: A screenshot of the web app that we have created