Image Caption Generator

B Deepika[1], S. Pushpanjali Reddy [2*], S. Gouthami Satya [3], K. Rushil Kumar[4]

[1,2,3,4]VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad.

pushpaseelamreddy@gmail.com

Abstract. Image captioning, also defined as describing the image, has consistently sparked the curiosity of expert system researchers and accurate description of an image has been a significant task. Image caption generator involves describing the characteristics, attributes of the image. It has a plenty of applications in the field of Robotic vision, story-telling from album uploads, business and many more. For instance, it can be used in Image segmentation as used by Google Photos and its application can also be extended to video frames. It has grown to become one of the most prevalent tools in the contemporary period. This paper aims in employing computer vision and machine translation for captioning the image. It involves recognizing the objects, actions, attributes in an image and identify the relation between the objects and the generated descriptions. Most of them use encoder-decoder framework, where the image, which is given as input, is encoded to an intermediary representation of the image's information and then decoded into a series of descriptions and descriptive text. The dataset employed for the same is Flickr8k dataset and the programming language is python. The project involves developing an app that takes an input image, extract features, and generate accurate descriptions, using Flutter. It has an immense potential in helping the visually impaired. It helps in automating the job of radiologists.

Keywords: LSTM, RNN, CNN, Deep Learning. Natural Language Processing

1 Introduction

We, as humans, are able to identify and describe colossal elements of an image with just a quick glance at the scene. But programming the machine to do the same is challenging as it requires identifying not only the object but other elements such as actions and attributes as well as creating fluent sentences summarizing the image. The well-studied visual perception tasks or image classification, which are of primary focus within the computer vision are considerably less challenging compared to this task[1,2]. This problem could be defined as that of machine translation with pixels of image as source language and English being the target language.

The objective is to provide an efficient system that can provide syntactically and semantically accurate text descriptions for an image. The study employs use machine learning techniques and deep neural networks to build a model. This paper utilized the

C. Kiran Mai et al. (eds.), *Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACECS 2023)*, Atlantis Highlights in Computer Sciences 18,
https://doi.org/10.2991/978-94-6463-314-6_35

Flickr8k dataset. Although new datasets can encourage innovation, benchmark datasets too, require fast and competitive evaluation metrics for quick progress.

Caption Generation is conducted in two phases: The first being feature extraction using Convolutional Neural Networks (CNN) and the second being sentence generation, achieved through RNN (Recurrent Neural Networks). For the initial phase, different approach of feature extraction, which provides us with information of the slightest variation between the two images, have been employed. The study employed a model having 16 convolutional layers, VGG16 (Visual Geometry Group), used in object recognition. The second phase involves training the features with the captions provided in the dataset. LSTM (Long Short-Term Memory) is employed for framing images from the given input image. Bilingual Evaluation Understudy (BLEU) score was employed for evaluating the performance of LSTM.

The following is how the paper is organized: Section 2 summarizes previous work on image captioning. Section 3 depicts the working method of detecting features and producing descriptions. Section 4 demonstrates the experimental evaluation. Section 5 concludes the work.

2 Literature Survey

The idea of using deep neural networks for caption generation was proposed in the research paper by Vinal (et al) [3]. The architecture proposed was a fusion of Convolutional Neural Network (CNN) and a long short-term memory (LSTM), the former for image feature extraction and the latter for sentence generation. The paper by Xu, Kelvin (et al) [4] expanded on previous research and introduced the concept of attention mechanism, enabling the model to focus on distinct regions of the image while captioning.

One of the widely used benchmarks for image captioning, COCO dataset was presented by Hodosh (et al) [5]. It also introduced the evaluation metrics such as BLEU, CIDEr, METEOR for assessing the quality of captions generated. A novel attention mechanism combining bottom-up and top-down attention mechanisms was introduced by Anderson (et al) [6], the former focused on salient objects present in the image, the latter employed linguistic cues to guide the generation procedure[9,10].

Meshed-memory transformer, a modified version of transformer architecture for image captioning was proposed by Cornia (et al) [7]. Zhou (et al) [8] introduced a unified pretraining approach that tackled image captioning and Visual Question Answering (VQA) simultaneously, showcasing better performance on both tasks.

The images are considered as two-dimensional arrays, while being read by the computer vision. Image captioning's analogy with language translation was explored by Venugopalan (et al) [12]. While traditional translation was complex and included many tasks, the contemporary research [13] underscores RNN's efficient handling of such tasks. Conventional RNNs faced challenges with vanishing gradients, necessitating the

integration of Long Short-Term Memory (LSTM) networks, which incorporate internal mechanisms and logic gates to effectively retain and transfer pertinent information.

One of the most significant challenges encountered was during the selection of the ideal model for the caption generation. Tanti (et al) [11], in the paper, has identified generative models as either “inject” or “merge” architectures, the former entails the RNN receiving inputs of both tokenized captions and image vectors, whereas the latter involves feeding only captions into the RNN block and then merging the output with the image.

Although there was comparable accuracy in the experiments, the project preferred the merge architecture for its streamlined design, which resulted in efficient utilization of RNN memory and expedited training. It is simple in design, having fewer hidden states. RNN RAM is better used as the images are not passed iteratively.

3 Proposed Methodology

3.1 System overview

Fig 1 describes the structure of the system. The Image Captioner system is designed to automatically generate textual descriptions of the image. To bridge the gap between visual content and natural language, it employs deep learning techniques, particularly Recurrent Neural Networks (RNNs). It follows a sequential procedure:

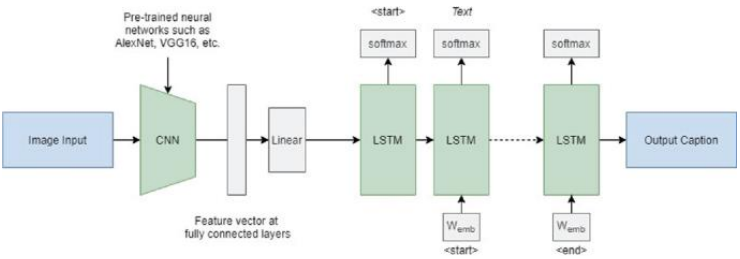

Fig.1. System Overview

1.Image Feature Extraction: The processing of the input image and extracting relevant visual features is done by employing a CNN (Convolutional Neural Network) These features capture the image’s salient elements and the context.

2.Text Generation: The features extracted form image are fed into an RNN, such as a LSTM or GRU. The RNN generates words progressively, resulting in a cohesive caption. At each phase, RNN considers both previously generated words and the features of image.

3.Attention Mechanism: It may be employed to improve the alignment between image regions and generated words. This lets the system focus on different sections of image while processing corresponding words.

4.Language Modelling: The language model of RNN is trained on dataset of images and their respective captions. Based on the attributes and features of image and the preceding words, the model learns to predict the most probable next word.

5.Evaluation and Refinement: To assess the quality of the captions generated, evaluation metrics like BLEU, METOR, and CIDEr are used. Based on the results, refinements might be made, incrementally enhancing the system's performance.

3.2 Modules

3.2.1 Task

The objective is to devise a system that takes the image input as a dimensional array, break it down for extracting different objects, attributes and generate the accurate description as an output.

In machine translation, the objective is to enhance the likelihood of p(T|S) while converting a source sentence S from the source language into its target counterpart T. Machine translation has traditionally entailed discrete tasks such as each word translation, alignment, and reordering. Recent improvements, however, show that employing Recurrent Neural Networks streamlines this process while retaining excellent performance levels. The source sentence is read and translated into a fixed-length vector representation, which serves as the decoder's initial hidden state when creating the target phrase.

For each caption, two additional symbols are added, to denote the start and end of sequence. When a stop word is encountered, the sentence generator pauses and end of string is marked.

The loss function, Cross-Entropy, is used to evaluate dissimilarity between the predicted probability distribution of words in a generated caption and actual ground-truth distribution of words in reference caption. It is calculated as:

$$Cross - Entropy\ Loss\ =\ -\sum yi\ log(pi)\ where\ 1 \leq i \leq N \tag{1}$$

3.2.2 Corpus

The Flickr8k Dataset was employed as our corpus, comprising approximately 8000 images. Each image is associated with five captions, which collectively assist in understanding a range of possible contexts.

3.2.3 Feature Extraction

The main role of this model is to extract image features essential for training, which are input to the model.

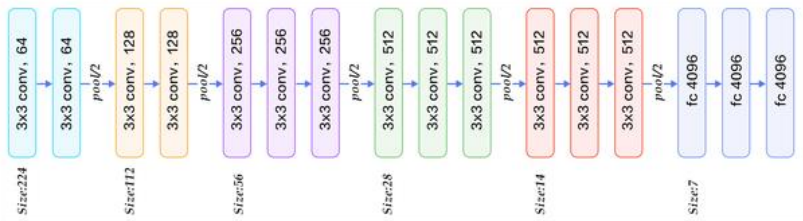

Fig. 2. VGG16 architecture

The model employs VGG16 architecture Fig 2 as shown in Fig 3 to extract or obtain the image features efficiently, using max pooling layers and 3*3 convolutional layers. The output would be vectors having size 1*4096, representing the features of image. A dropout layer, Values falling within the recommended range of 0.5 to 0.8 are generally deemed optimal, signifying the probability at which the layer's outputs are omitted, with a value of 0.5 is introduced into the model, to counteract overfitting.

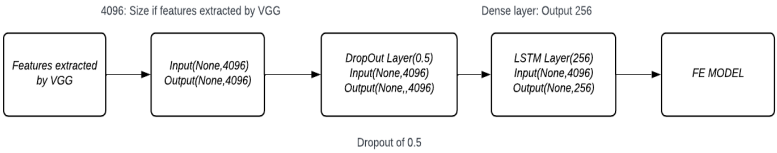

Fig. 3. Feature Extractor

A dense layer, which includes an activation function for the input as well as a kernel with bias, is added following the dropout layer. Rectified Linear Units (ReLU) are used as the activation function, and the output space dimension is set at 256. The 256-bit vectors generated by the feature extraction model are used by the decoder model.

3.2.4 Encoder

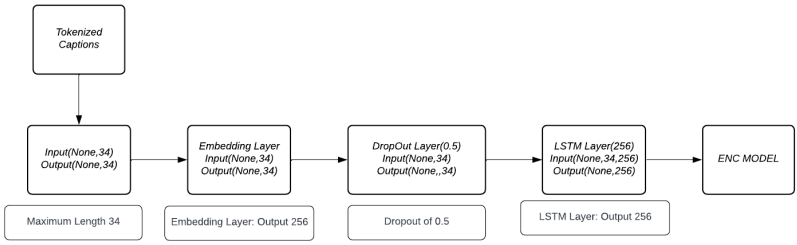

Fig. 4. Encoder Model

As The captions for each image, input while training, are mostly processed by the encoder, Fig 4. The decoder would again receive the input of vectors of size 1*256, generated by the encoder model.

At the outset, captions for each image are subjected to tokenization, wherein words within the sentences are converted into integers. This conversion is intended to enhance neural network processing. Following this, the tokenized captions, referred to as tokens, are padded to align with the length of the longest caption. This guarantees consistent processing of all captions at a uniform length.

Following tokenization, the processed captions are embedded into unchanging dense vectors, which possess a specified output space dimension of 256 by 34, 34 being the maximum possible count of words in the captions specified in Flickr8k, accomplished through the utilization of an Embedding layer. These vectors streamline processing by providing a straightforward approach to representing words within the vector space. A dropout layer with a dropout rate of 0.5 is introduced to counteract overfitting.

At the core of the encoder lies the LSTM layer, instrumental in guiding the model to grasp the intricacies of constructing accurate and meaningful phrases. This includes generating words with the highest probability following the encounter of specific words. The selected activation function is Rectified Linear Units (ReLU), featuring an output space dimension of 256. This layer can be replaced by GRU for comparison purposes.

3.2.5 Decoder

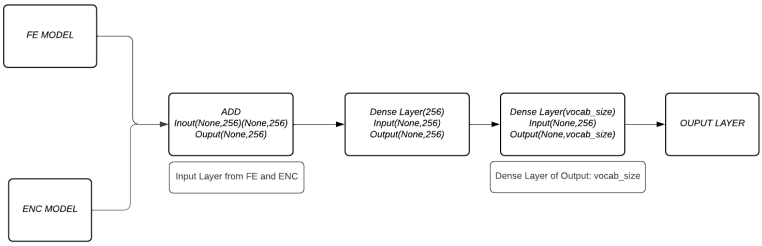

Fig. 5. Decoder model

As depicted in Figure 5, the decoder receives input from both the encoder model and the feature extraction model, both of which generate output vectors with dimensions of 256. The outputs from these models are then fed through a dense layer utilizing the ReLU activation function. A second dense layer is added, with the dimension of output space chosen to match the vocabulary size.

The activation function employed in this context is softmax, which generates a word or phrase corresponding to the predicted integer. The decoder layer's output constitutes the predicted word. The model is trained using the input-output parameters specified.

<input> = <image>, <in-seq>

<output> = <word>

The input format is as follows: <input> corresponds to an image and <in-seq> represents a sequence. The output format is: <word> represents a word.

4 Evaluation

4.1 Experiment Setup

The Flickr8k dataset comprises over 8000 images, with each image accompanied by five captions. Due to its relatively small size, it can be trained efficiently. The dataset is appropriately labeled and accessible on platforms like Kaggle. The fundamental stage in building the model involves data pre-processing and cleaning, a crucial step that contributes to the accurate creation of models through a comprehensive understanding of the data. Upon extracting the zip files, two folders are identified:

Flickr8k_Dataset: It comprises of 8092 images in JPEG format with different actions and attributes, out of which 6000 images are employed for training, 1000 for testing and 1000 for development.

Flickr8k_text: It contains text files describing the training and testing sets, namely train_set, test_set. A total of 40460 captions i.e five captions for each image is present in Flickr8k.token.txt.

4.2 Evaluation Results

Fig. 6. caption generated

Fig 6 consists of the captions generated for an image input. The vocabulary size was found to be 7371 and the ten most frequently occurring words are found to be.

```
('a', 46784),('in', 14094),('the', 13509),('on', 8007),('is', 7196),
('and', 6678),('dog', 6160),('with', 5763),('man', 5383),
('of', 4974)
```

Fig. 7. Frequently occurring words

Fig 7 gives the words having frequency more than 10 were considered, as the words occurring seldom do not deliver much of the information. Fig 8 gives average caption length and Fig 9 gives BELU score with graph.

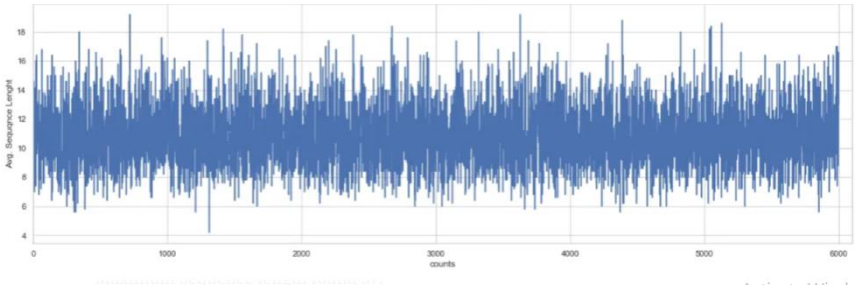

Fig. 8. Average Caption Length

EPOCHS	DROPOUT	BLEU SCORE
10	0.5	0.614561
20	0.5	0.628900
30	0.5	0.634662
40	0.5	0.649467
50	0.5	0.632956

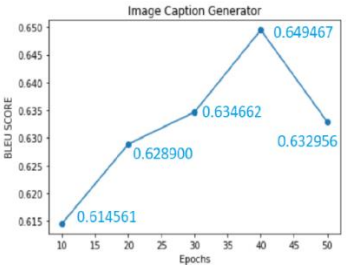

Fig. 9. BLEU Score and Graph

5 Conclusion

This deep learning model tends to generate the descriptions automatically with objective of not only defining the surroundings but also to assist the visually impaired gain knowledge of the surroundings better. The suggested model employs a CNN feature extraction component to encode the image into a vector representation, followed by an RNN decoder that generates sentences based on the recognized image properties. This approach of processing the image and generating captions uses computer vision and natural language processing. The entire model has created a distinction among the various image recognition models.

The BLEU score obtained for this model is 0.64, which is a quite satisfactory score. We might foresee improved models with enhanced algorithms that will revolutionize image processing in the future, as technology is advancing rapidly. In recent years, there were significant advancements in neural networks and computer vision. We might expect models with improved performance in the future as development of next-level LSTM is going on.

The quality of captions is influenced by the model as well as the data preprocessing and using an appropriate dataset. The of epochs the model is trained on determines the accuracy of the model.

On using a larger dataset and hyperparameter tuning, the performance is expected to be enhanced. It is of great benefit to the visually impaired due to its increased accuracy of the generated captions for the image and the text to speech technology incorporated in the application as well.

References

1. ” 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.
2. Image Caption Generator Based on Deep Neural Networks by Jianhui Chen, Wenqiang Dong and Minchen Li, ACM (2014).
3. Show and Tell: A Neural Image Caption Generator by Oriol Vinyal, Alexander Toshev, Samy Bengio, Dumitru Erhan, IEEE (2015).
4. Show, attend and tell: Neural image caption generation with visual attention by Xu, Kelvin.
5. Framing image description as a ranking task: Data, models and evaluation metrics by Hodosh, Micah, Peter Young, and Julia Hockenmaier (2013)
6. Bottom-up and top-down attention for image captioning and visual question answering by Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. (2018)
7. Meshed-memory transformer for image captioning by Cornia, Marcella, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara (2020)
8. Unified vision-language pre-training for image captioning and vqa by Zhou, Luowei, et al (2020)
9. Image2Text: A Multimodal Caption Generator by Chang Liu, Changhu Wang, Fuchun Sun, Yong Rui, ACM (2016).
10. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions by Sepp Hochreiter.
11. Where to put the Image in an Image Caption Generator by Marc Tanti, Albert Gatt, Kenneth P. Camilleri.
12. Sequence to sequence -video to text by Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko.
13. 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.