Proceedings of the 5th International Conference on Smart Systems and Inventive Technology (ICSSIT 2023)
IEEE Xplore Part Number: CFP23P17-ART; ISBN: 978-1-6654-7467-2

2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT) | 978-1-6654-7467-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSSIT55814.2023.10061040

Attention based Image Caption Generation (ABICG) using Encoder-Decoder Architecture

Uday Kulkarni
Asst. Professor, Dept. of CSE
KLE Technological University
Hubballi, India
uday_kulkarni@kletech.ac.in

Kushagra Tomar
Department of CSE
KLE Technological University
Hubballi, India
kushagratomar2016@gmail.com

Mayuri Kalmat
Department of CSE
KLE Technological University
Hubballi, India
mayurikalmat1@gmail.com

Rakshita Bandi
Department of CSE
KLE Technological University
Hubballi, India
rakshitabandi0@gmail.com

Pranav Jadhav
Department of CSE
KLE Technological University
Hubballi, India
jadhavpranav250@gmail.com

Dr. Meena S M
Professor and Head, Dept. of CSE
KLE Technological University
Hubballi, India
msm@kletech.ac.in

***Abstract*—The image captioning is utilized to develop the explanations of the sentences describing the series of scenes captured in the image or picture forms. The practice of using image captioning is vast although it is a tedious task for the machine to learn what a human is capable of. The model must be built in a way such that when it reads the scene, it recognizes and reproduce to-the-point captions or descriptions. The generated descriptions must be semantically and syntactically accurate. Hence, availability of Artificial Intelligence (AI) and Machine Learning algorithms viz. Natural Language Processing (NLP), Deep Learning (DL) etc. makes the task easier. Although majority of the existing machine-generated captions are valid, they do not focus on the crucial parts of the images, which results in lesser clarity of the captions. In the proposed paper, anew introduction to attention mechanism called Bahdanau's along with Encoder-Decoder architecture is being used so as to reflect the image captions that are more accurate and detailed. It uses a pre-trained Convolutional Neural Network (CNN) called InceptionV3 architecture to gather the features of images and then a Recurrent Neural Network (RNN) called Gated Recurrent Unit (GRU) architecture in order to develop captions. This model is trained on Flickr8k dataset and the captions generated are 10% more accurate than the present state of art.**

Keywords— Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Encoder, Decoder, Attention mechanism, Image captioning.

I. INTRODUCTION

Language is the medium through which the society constantly interacts, be it written or spoken. It typically describes the perceptible world around us. The photos and symbols are otherwise to speak and perceive by the physically disabled individuals. Automatic description generation from an image in proper sentences is a tedious task, nevertheless it can have an ample impression on visually challenged individuals for better understanding of the description of pictures on the web. Long back, "Image or Picture Captioning" [2] [3] [4] has always been a rigid mission and the generated captions for the given image were not so pertinent.

In conjunction with the progress of Deep Learning [21], CNN and techniques for processing text like NLP [22], a number of previously challenging pieces of work became straightforward using Machine Learning. These are profitable in recognition, classification and captioning of images and several additional AI [19] applications. This "Image Captioning" [2] is pragmatic in various applications viz. The concept of self-driving cars, surveillance which are at present the matters of the moment. In comparison with classification and object recognition, the task of automatically generating captions and describing images is significantly more complicated. A description of an image must include more than just the objects in it, also how the objects are related to their attributes and activities just like shown in the Fig. 1

a. white bird flaps its wings above the water
b. hockey player in helmet
c. man holds baby on the table at restaurant
d. three dogs playing in the green water

Fig. 1. Examples of image captioning

However, it is best to express semantic knowledge in natural language such as English. A single model is to be

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

designed which intakes an image and trains it to give out a string of words in which each word is affiliated to the glossary that narrates the picture accordingly. In this paper, an ABICG model is proposed which is worthy of describing images in a latest and novel way. For this job, the dataset composed of 8000 images and five descriptions per image is taken from the Flickr8k dataset [20]. As new applications are being developed every day, Deep Learning has emerged to be a prepotent world today. Test and trial is the mere way to explore Deep Learning to a greater extent. In this way, you gain a better understanding of the topic and become a more profound professional. Real-life applications of this technique are numerous. The use of Deep Learning for image description has been proposed in many different models, including detection of objects, captioning on the basis of visual attention and image captioning using Deep Learning. Different Deep Learning models exist as well, such as the InceptionV3 model [9], the Visual Geometry Group 16 (VGG–16) model [14], the Residual Networks (ResNet) [18] – Long Short-Term Memory (LSTM) model [15] and the traditional CNN – RNN model [23]. CNN as well as RNN are used here. For image encoding i.e. as an encoder, in order to classify images, a pre-trained CNN is used. Input from the last hidden layer is used to train the RNN. This network is a decoder to generate captions. In LSTM networks, memory cells are endowed with a limited number of phases that have been determined by long-term dilution of existing memory information. A total of 16 layers supported by VGG–16 is an ingenious model for object recognition. For the following stage, the extracted features are trained with wording specified in the dataset. Two architectures LSTM and GRU [16] are used for framing sentences from the given input images given.

In ABICG, the Flickr8k dataset [20] is used and it is subjected to an elaborated preprocessing steps to optimize the input. The preprocessed data is fed to the model which uses Inception-V3 as Encoder and GRU as Decoder. Bahdanau's Attention model is applied to this encoder-decoder model to fetch more focused captions. Meaningful captions are generated by the model. Performance of this model is evaluated by the BLEU scores [31].

The paper is organized as, the Section 'Related Work' discusses the previous works and methodologies related to this domain which includes the scope of improvements in the methods that already exist and the uniqueness of the model presented in this paper. In Section 'Proposed Work', it comprises the proposed work where the ABICG architecture has been discussed along with explanation to each component of the architecture. Section 'Experimental Results' further elaborates on the system specifications, dataset used, data pre-processing, results, comparison of training loss and evaluation of model using BLEU scores [32]. Finally, the conclusion and the future scope have been discussed below as can be seen in the Section 'Conclusion'.

II. RELATED WORK

In [1], the authors have made use of CNN as an encoder to extract the characteristics or attributes from the images. CNN is a pre-trained InceptionV3. Owing to the InceptionV3's fact that it is a deep network for object detection, it demands to be altered slightly to assist in encoding. A feature vector is obtained from this deep network by removing the terminating layer. The feature vector obtained is of the size (8x8x2048). The feature vector is the input to RNN. The RNN employed for decoding is GRU [16]. To generate more focused captions, Bahdanau model is used.

The writers of [2] have proposed a model where the input sent is Flickr8k dataset, and the output obtained is passed to the latest layer which is completely connected and is introduced at the termination of the InceptionV3 model. The task of this layer is to transform the model's output into a vector which embed words. It serves as an input to an LSTM cell order by implanting a vector. The LSTM unit attaches the series of information and collects it progressively hence enabling the establishment of meaningful captions. The start-V3 component of this model is trained to recognize complete possible objects in a picture. Each word in the picture is predicted using the previous words in the phrase. The main intention of training is to reduce the failure function. They have used the Flickr8k dataset which has nearly 8000 images and each image is tagged with five unique captions or descriptions which offer compact reports of the noteworthy features.

In [3], the authors have put forth a model that allows neural networks to view an image automatically and yield meaningful captions similar to natural English sentences. It is a well-trained model to perform the above-mentioned tasks. Here, the pre-trained CNN is utilized to classify images. This network handles the task of encoding images. The input to the RNN (decoder here) is the hindmost hidden layer of the encoder. The decoder generates sentences. The dataset being used here is Flickr8k [20] consists of about 8000 images and five descriptions tagged to every image. They used VGG [14] for large-scale image recognition. They conclude that using a bulky dataset boosts the performance of the model. In addition to reducing losses, it also improves accuracy.

To achieve better results, the authors of [4] worked on a model that combines CNN architecture and LSTM for image captioning. The proposed model uses three CNN architectures: ResNet-50 [25], Xception [24], and Inception-V3 [9]. The aptest combination of CNN and LSTM is chosen based on the model's accuracy. Training is performed on the Flickr8k data. Combining Xception with LSTM has the highest accuracy of 75% across epochs among the three CNN models.

The authors of [5] proposed a model where CNN features

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

are extracted from an image and encoded into vector representations using 16 convolutional layers of the VGG-16. Next, a RNN decoder model is used to develop corresponding sentences based on the learned image features i.e., training the features with captions or descriptions provided in the dataset. The input images are processed using two architectures viz, GRU and LSTM. Through the results, it is evident that the LSTM model achieves better results than the GRU [16] model. Although, it takes a longer time to train and generate captions due to the model's complexity.

In [6], the authors have developed a model where pre-processed images are fed to the Inception V3 model, and the features are extracted. Later, a D-dimensional representation of each and every part of the image is produced by the extractor such as L vectors. With the spatial features of a CNN convolution layer, the decoder calculates the context vector according to the specific regions of the input image. For the decoder's job, GRU is utilized which has a simpler structure than LSTM [15] . The vanishing gradient problem does not affect GRU, unlike RNN. Thus, it proves that usage of GRU gives better results than LSTM.

The authors display a method to overcome the vanishing gradient problem which hinders the existing CNN-RNN models in [7] . They have proposed ResNet-LSTM as an encoder-decoder technique for image captioning. The ResNet (encoder) extracts the features and the LSTM (decoder) generates the caption from the extracted features. For this, the images are resized to (224x224x3) and subjected to several pre-processing steps. They have used the Flickr8k dataset for training the model. After a minimum of 20 epochs, meaningful captions begin to generate. It is better than VGG and CNN-RNN models.

The authors of [8] explain a multi-feature fusion model to generate image captions. Models that currently exist focus on the global characteristics of an image, but with the comprehensive features, this model also considers the localized features of images. A global feature extraction of global features is performed using the VGG16 network and Faster-CNN is used to excerpt the local characteristics. The local and global features are mixed and fed as input, through an attention layer to the Bi-LSTM [28]. The caption obtained is corrected if any error occurs. ImageNet dataset with image size (224x224x3) is used to train VGG16 [14], Pascal VOC dataset is used to train Faster-RCNN [29] (1:1 positive and negative sample ratio maintained). Bi-LSTM is trained with the MSCOCO dataset [27]. The fused features have turned out to be superior to global or local features alone. The test accuracies in the training set and verification set are 78.20% and 66.50% respectively.

Majority of the existing models are hindered by the vanishing gradient [12] problem. Usage of the CNN is prevalent in the existing models. Due to the vanishing gradients, as the depth of hidden layers increase, the learnability of models fall to zero. In most of the works, LSTMs are used which are slower, and computationally less efficient as compared to GRU. To solve this issue of vanishing gradients, the GRU is utilized in the current work.

III. Proposed Work

To accomplish the task of image captioning, ABICG comprises bipartite architecture, viz encoder and decoder. Images are fed to the encoder where the image information is transformed into features vectors. The output of the encoder is passed to the decoder, which translates the features into English sentences. This method is termed as *"Classic Image Captioning"*. The problem with this method is that, it is not possible to take into account the spatial features of an image with this classic captioning method. As a result, a caption is generated taking into consideration the full image as a scene and not considering the sensitive or important features of the image. To enable the model to focus on important features of the image, Bahdanau's attention has been used jointly with the encoder-decoder architecture.
The captions generated by the proposed model are semantically and grammatically correct. The generated captions are close to the human generated or with human centric meaning and they not only describe the scene in the image but also the intricate details and relationship of the object with the background.

A. Model Overview

In ABICG, the CNN used is InceptionV3 [9] pretrained on ImageNet weights, serves as an encoder. This extracts the features from the receptive fields of the images and forwards it to the decoder. Here, the RNN used is GRU, which is used as a decoder. The use of the decoder is to decode the sentence from the encoding. The Bahdanau attention model [11] is used to enhance the capability of the decoder by allowing it to focus on the important aspects of the images while producing the captions. Thus, taking care that the sensitive parts of the image are not left out in the generated caption.

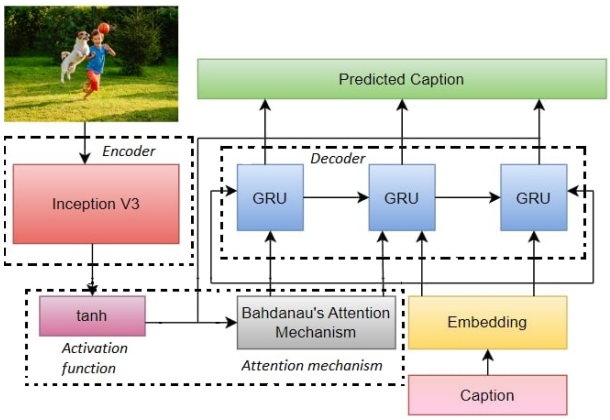


Fig. 2. Overall Architecture

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

The above figure shows the overall architecture of the model which includes the CNN encoder (InceptionV3), RNN decoder (GRU) and the attention model. Image is fed into the encoder and then the tanh activation function is applied to introduce non-linearity. The output is then fed to the decoder and for each timestamp of the decoder, the attention model enables the decoder to focus on specific parts of the images.

B. Convolutional Neural Network (CNN)

InceptionV3 [9] is often used for image recognition and has been very popular in field of image processing because of its up to the mark accuracy on different datasets. It encompasses the building blocks which are of types asymmetric and symmetric, along with convolutions, max pooling, average pooling, concatenations, dropouts and various fully connected layers as shown in the Fig 3.

It was built for the purpose of object detection on receiving a (299x299x3) image. Since InceptionV3 [9] is mostly used for object detection, the refinement of it to some extent is required so as to make it an encoder for extracting the image feactures. The last layer is eliminated which is used for classifying the images into the labels since classification of images is not required. Thus, a feature vector is obtained which is of the size (8x8x2048). The resulting feature vector is static and does not alter at each timestamp. Therefore, this vector is passed to the attention model along with the hidden state of the decoder to create the context vector.

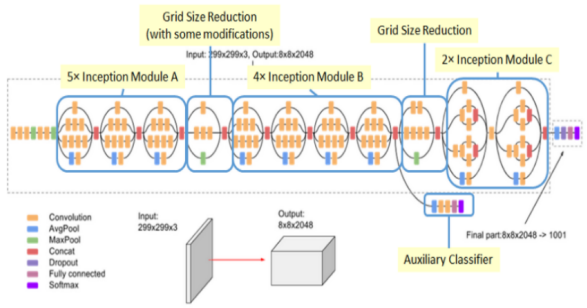


Fig. 3. Convolutional Neural Network (Inception V3) [9]

The benefits of using InceptionV3 CNN for the encoder part is that it generates fewer parameters for computation which makes it computationally less expensive in comparison to the other models and is memory efficient. There is no comparison between InceptionV3 and the other models when it comes to depth and accuracy.

C. Attention Mechanism

Attention model is a deep learning technique that makes use of attention mechanism which provides attention or additional focus on specific components. The *Bahdanau's Attention Model* [11] is used in ABICG. It selectively highlights the relevant features of the input data. It is also referred to as an interface that connects the encoder and decoder. It instructs the decoder with the relevant details from each and every encoder hidden state. Decoding begins with the context vector generated by the attention model to predict the word at that particular timestamp. The context vector changes with each timestamp since it is adaptive in nature.

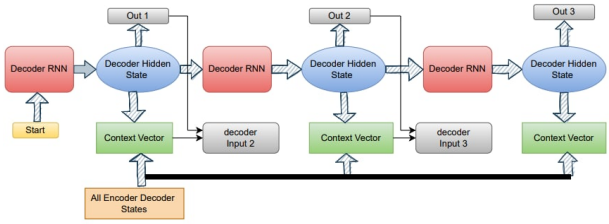


Fig. 4. The Bahdanau's Attention Model [11]

Attention model Fig 4 [11] does a linear transformation of the input by applying $tanh$ (1) to it so as to introduce non-linearities henceforth achieving a smoother distribution. Then, the attention score a_s is computed. The output is required in the range (0,1). The $softmax$ function is applied to the attention score and the final attention weights are obtained. This model intends to overcome the curb of the orthodox CNN–RNN models. This model facilitates passing various parts of the image instead of the whole. This also makes it swift and hikes its accuracy.

$$a_s = \tanh(W_1 h_{d1} + W_2 h_{d2}) \tag{1}$$

With this score, the weights of the attention are calculated using (2)

$$\alpha = softmax(a_s) \tag{2}$$

Then, by using the attention weights (α) from (2) and features (h_{d2}) which were obtained from an encoder, the context vector c_{vec} is obtained, with (3).

$$c_{vec} = \alpha h_2 \tag{3}$$

Ultimately, the fixed length vector "c_{vec}" is unified with the decoder's output from the predecessor timestamp h_t and then fed into the RNN cell in order to obtain the decoder's output for the current timestamp.

In the above image, there is a white color bird sitting on the sign board. The image is fed into the encoder which extracts the image features and gives it as an input to the decoder which transforms the image feature vector into concise caption. So here, the caption generated would be "a white bird perched on top of a red stop sign" all in lower case.

The project aims at mimicking the human brain because of its abilities to generate a caption for every scene it senses. Therefore, it becomes crucial to add an attention mechanism using which the CNN-RNN model focuses on the more

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

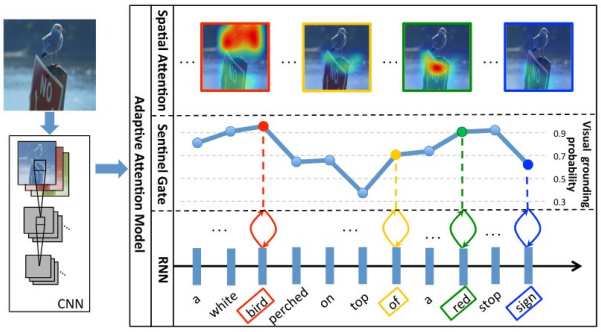


Fig. 5. Complete architecture of the proposed CNN-RNN-Attention model [17]

important parts of the image. There is no static vector encoding of the whole image in the attention mechanism. Instead, it adds the spatial information corresponding to the image to the extraction of image features. As a result, statements are described in a more detailed manner as shown in the above Fig. 5 [17].

By such means, while generating sentences, simulation of the human vision using the attention mechanism can be encouraged with the generation process of word sequence. This is to ensure that the generated sentence will reflect the expression habits of the people.

D. Recurrent Neural Network (RNN)

GRU is used as a decoder. It works on the mechanism of RNN which anticipates expressions in the natural language. The other probably used RNNs are LSTM, Vanilla RNN, and the GRU. Vanilla RNN is not preferred due to its *vanishing gradient problem.*

GRU is similar to LSTM. It owns a few key differences from LSTM: GRU has only two gates whereas, LSTM has three gates. It exposes its total memory and also the hidden layers. GRU not only requires fewer parameters for training, but also has way more effective computation. Thus making it computationally efficient.

GRU is composed of two gates, viz., The update gate and the reset gate. Both these gates in GRU together act as a convex combination which gives the verdict of which information or the data of the hidden state is to be updated and which is to be forgotten.

A large number of layers in the network lead to a fall in the derivative product till the partial derivative of the loss function tends to zero, and the partial derivative is abolished. This phenomenon is known as the vanishing gradient problem. In simple words, this means that the initially predicted words are wiped out as the new words are predicted therefore giving less weightage to the initial words and vice versa in the output generated. To tackle this problem, the LSTM was inaugurated. There is not much difference between GRU and LSTM. But GRU has a simpler network cell architecture as shown in Fig. 7 [10] as compared to that of LSTM. Hence, the GRU is used in this caption generator model.

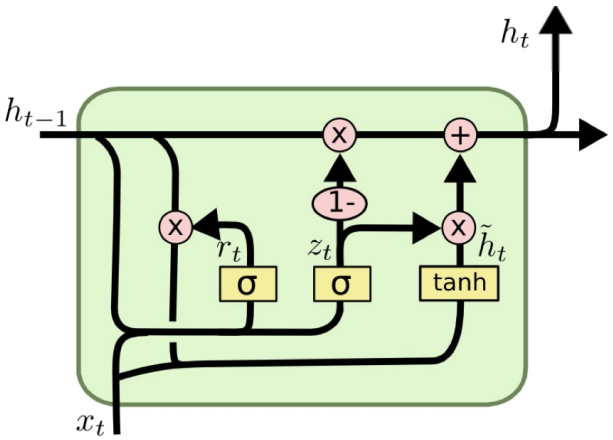


Fig. 6. The Gated Recurrent Unit [16]

Fig. 6 displays the working principle of GRU with a diagram. Here, x_t is an input vector, z_t is an update gate vector, h_{t-1} is a previous output, h_t is the current output, r_t is the reset gate vector and $\tilde{h_1}$ is an activation vector. Sigma (σ) represents the sigmoid activation function and $tanh$ represents the tan hyperbolic operation. Firstly, the update gate vector z_t is calculated for time step t using (4).

$$z_t = \sigma(Weight_{input_{update}} * X_t + Weight_{hidden_{update}} * h_{t-1}) \quad (4)$$

The input vector x_t and previous vector h_{t-1} are multiplied with their respective own weights viz. $Weight_{input_{update}}$ and $Weight_{hidden_{update}}$. The obtained products are added and the sum is squashed between 0 and 1 using the sigmoid activation function. Update gate enables the model to decide how much of the previous content needs to be sent to the future.

Then, the forget gate vector (5) r_t is calculated using the same formula as used in (4). A gate like this allows the model to calculate how much information has to be forgotten from the past.

$$r_t = \sigma(Weight_{input_{reset}} * X_t + Weight_{hidden_{reset}} * h_{t-1}) \quad (5)$$

For the current memory content, the input x_t is multiplied with a weight W_{h_1} and h_{t-1} is multiplied with a weight W_{X1}. Then, the Hadamard (element-wise) product is calculated is between the reset gate r_t and $W_{h_1} h_{t-1}$. All these are added and a nonlinear activation function $tanh$ is applied as shown in (6).

$$\tilde{h_t} = \tanh(r_t \odot W_{h_1} * h_{t-1} + W_{X1} * X_t) \quad (6)$$

For the final memory content at current time step, element-wise multiplication is applied to the update gate z_t and h_{t-1}, and $1 - z_t$ and $\tilde{h_t}$. Then, these two products are added, as shown in (7).

$$h_t = (1 - z_t) \odot \tilde{h_t} + z_t \odot h_{t-1} \quad (7)$$

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

IV. Experimental Results

In the proposed ABICG, the neural framework is proposed for generating the descriptions for the given input images.

A. System Specifications

The model was trained on the system having specifications mentioned in Fig. 7 .The hyper parameters decided were epochs of 25, batch size of 64, and learning rate of 0.001. NVIDIA's CUDA was used to achieve parallel processing. The training took approx 1 hour 31 minutes.

System Specifications	
Processor	AMD Ryzen 7 4800H
Core	8 core
Thread	16 threads
RAM	16GB
Frequency	1600Mhz
Graphics	NVIDIA (GeForce GTX 1650) - 4GB GDDR6

Fig. 7. Hardware specifications of the system on which the model was trained.

The Tensor-Flow [26] is an back-to-back open source platform for this domain, Machine Learning. Google is the pioneer of the Tensor-Flow. It has various frameworks of Deep Learning. That being utterly flexible, portable, and reliable, is used in ABICG.

B. Dataset

The dataset used to develop ABICG is Flickr8k [20]. There are approximately 8000 images in the collection (8091 to be precise) and each image has 5 captions. So, there are a total of 40455 captions generated to build the aimed model. Then, Python's TensorFlow library is used to preprocess the images.

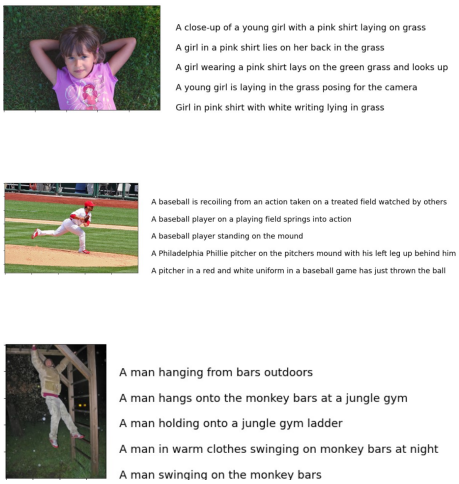


Fig. 8. Sample data from Flickr8k Dataset [20]

C. Data Preprocessing

The considered dataset comprises 5 captions which correspond to each image. Therefore, preprocessing is performed twice, one for the images and another one for the captions or the annotations. Captions preprocessing includes the removal of punctuation and alpha-numeric values from each caption. Also, $< start >$ and $< end >$ tags are introduced at the beginning and ending of each caption respectively. Then, creating the tokenized vectors by tokenizing the captions i.e. splitting them into words using spaces and other filters. This gives a lexicon of all unrepeated words in the data. For memory efficiency, the total vocabulary is restricted to 5000 words. All other words with the unknown token are replaced with $< UNK >$. Then follows the creation of a word-to-index mapping and a index-to-word mapping.
The input to the decoder should be of same size and shape. Therefore, padding is used to bring all captions to a fixed length before proceeding further. In order to ensure that all samples have a standard length, zero padding is applied before or after a sequence. In this model zero padding is done at the end (of the caption sequence). But padding can result in a risk of adding penalty to the model. Masking is applied to rectify the same and this will truncate down all the added penalties back to zero. As for the image preprocessing, the images are reshaped into (299, 299) and normalized within the range of -1 to 1, such that it is in correct format for CNN encoder (InceptionV3).
Afterwards, the captions are mapped with their corresponding image names in the dataset. So when training, the vectors corresponding to caption and image feature are mapped together and trained suitably.

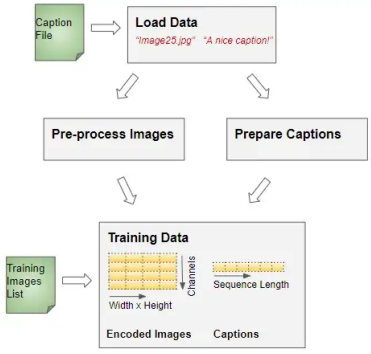


Fig. 9. Process of data preprocessing [13]

D. Results

1) Image 1

The below image is fed to the model and the generated caption is "large bird swooping down towards the ground" in comparison with the human generated annotation – "a white

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

bird swooping down the ground".

Fig. 10. "large bird swooping down towards the ground"

2) Image 2

The below image is fed to the model and the caption generated is "two girls hanging upside down on monkey-bars at a park" in comparison with the human generated annotation – "two girls are hanging upside down".

Fig. 11. "two girls hanging upside down on monkey-bars at a park"

3) Image 3

The below image is fed to the model and the caption generated is "the people are standing in front of the building" in comparison with the human generated annotation – "the people are standing before a building".

Fig. 12. The people are standing in front of building

4) Graph of Loss vs Epoch for ABICG model

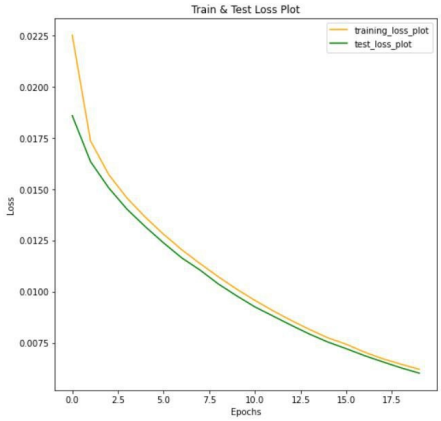


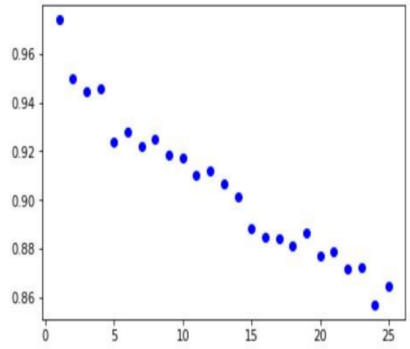
Fig. 13. Train-Test Loss vs Epoch for ABICG model

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

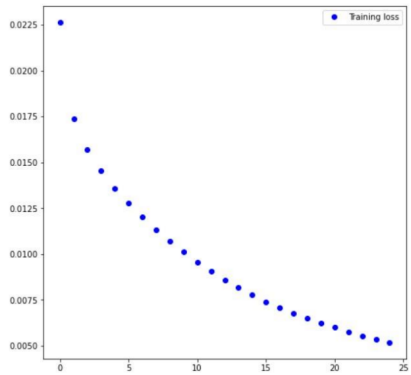
ABICG model was trained for an epoch of 25 with the hyperparameters like learning rate and batch size set to 0.001 and 64 respectively. The resultant Test and Training vs Loss plot obtained is shown in Fig. 13. As expected the training and testing loss decreases as the number of epochs increases.

E. Comparision of Training Losses

'Train Loss VS Epoch' graph was plotted for 25 epochs on both Traditional InceptionV3-GRU model (without attention) and InceptionV3-GRU model with attention(ABICG model). From the below comparision, it is evident that the train loss is higher in the InceptionV3-GRU model without attention [32] (Loss = 0.8647) as compared to the InceptionV3-GRU model with attention (ABICG model) (Loss = 0.0050).



14.1 InceptionV3-GRU (Without Attention): Train Loss= 0.8647



14.2 InceptionV3-GRU ABICG model(With Attention) : Train Loss=0.0050

Fig. 14. Comparison of train losses between InceptionV3-GRU without attention and InceptionV3-GRU with attention (ABICG)

F. Metrics

The volumetric data hinders the way to show the result for each image. Hence, it becomes essential to look for a method to assess the system's average accuracy on the entire dataset. There are multiple ways to evaluate the quality of machine-generated text. For this model, the Bilingual Evaluation Understudy (BLEU) [31] has been chosen as the evaluation metric, owing to its popularity and ease of usage. Before giving introduction to BLEU, knowledge about 'precision', a simpler and more well-known metric is customary. Let machine-generated n-grams and ground truth n-grams be denoted by the vectors x and y respectively. For instance, x could be taken as the words of a caption generated from an image, with x_i representing an individual word, and y could be the words from actual captions describing the same scene. It is expected always to denote the several possible captions of a single idea.

$$p = \frac{1}{N} \sum_{i=1}^{N} 1\{x_i \in y\}$$

The BLEU score and precision are equivalent, except the fact that there can only be a single instance of an n-gram in x for every incidence of an n-gram in y. Say, the statement "is is is is is" would receive an absolute precision if the word 'is' was present in the reference translation, but not compulsorily a perfect BLEU score, as it limits to counting only the number of occurrences of 'is' as it appears in y.

For ABICG model, a BLEU score of 90% was obtained for the weights (0.75, 0.25, 0, 0) and (0.50, 0.25, 0, 0).

Model	Accuracy
InceptionV3-GRU (without Attention)	79%
InceptionV3-GRU with Attention (ABICG model)	90%

Fig. 15. Comparision of accuracies

From the above table, it is evident that the accuracy of the InceptionV3-GRU model with attention (ABICG model) is approximately 10% more than that of traditional InceptionV3-GRU model (without attention).

V. CONCLUSION

In this paper, the caption generator for any given input image is being proposed using the encoder decoder techniques. The novel attention mechanism is the prime focus of the paper. The attention mechanism which is introduced after the InceptionV3 layer of networks here, makes the model focus on the highlighted receptive fields in the image to facilitate the decoder to produce captions solely for those parts. This greatly hikes the performance or the process of spawning the captions as compared to the orthodox encoder-decoder models. Results fetched from the model are budding and generated captions are clear.

Since the model had been exposed to a confined training set and vocabulary, the model may be deficient in connecting the input images to those features or the characteristics which are not present in the vocabulary. So, words like these are replaced with $< UNK >$ tag which means that these are unknown to the model. The model might not do well with such kinds of input images where the $< UNK >$ tag occurs. In such cases, the captions reproduced might be too trivial.

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.

Future scope of this work includes the usage of the transformer based models instead of the existing Encoder-Decoder based models, with the multi-head attention coupled with the positional embedding helps render information regarding how the different words are related in correct order.

REFERENCES

[1] V. Agrawal, S. Dhekane, N. Tuniya and V. Vyas, "Image Caption Generator Using Attention Mechanism," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1-6, doi: 10.1109/ICCCNT51525.2021.9579967.

[2] S. Degadwala, D. Vyas, H. Biswas, U. Chakraborty and S. Saha, "Image Captioning Using Inception V3 Transfer Learning Model," 2021 6th International Conference on Communication and Electronics Systems (ICCES), 2021, pp. 1103-1108, doi: 10.1109/ICCES51350.2021.9489111.

[3] Amritkar, Chetan Jabade, Vaishali. (2018). Image Caption Generation Using Deep Learning Technique. 1-4. 10.1109/IC-CUBEA.2018.8697360.

[4] C. S. Kanimozhiselvi, K. V, K. S. P and K. S, "Image Captioning Using Deep Learning," 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, pp. 1-7, doi: 10.1109/ICCCI54379.2022.9740788.

[5] Sharma, Grishma Kalena, Priyanka Malde, Nishi Nair, Aromal Parkar, Saurabh. (2019). Visual Image Caption Generator Using Deep Learning. SSRN Electronic Journal. 10.2139/ssrn.3368837.

[6] Manish Raypurkar, Abhishek Supe, Pratik Bhumkar, Pravin Borse, Dr. Shabnam Sayyad5(2021). Deep Learning Based Image Caption Generator.

[7] Aishwarya Maroju , Sneha Sri Doma, Lahari Chandarlapati, 2021, Image Caption Generating Deep Learning Model, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) Volume 10, Issue 09 (September 2021).

[8] M. Duan, J. Liu and S. Lv, "Encoder-decoder based multi-feature fusion model for image caption generation," Journal on Big Data, vol. 3, no.2, pp. 77–83, 2021

[9] Szegedy, Christian Vanhoucke, Vincent Ioffe, Sergey Shlens, Jon Wojna, ZB. (2016). Rethinking the Inception Architecture for Computer Vision. 10.1109/CVPR.2016.308.

[10] Cho, Kyunghyun Merrienboer, Bart Gulcehre, Caglar Bougares, Fethi Schwenk, Holger Bengio, Y.. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 10.3115/v1/D14-1179.

[11] Bahdanau, Dzmitry Cho, Kyunghyun Bengio, Y.. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv. 1409.

[12] Hochreiter, Sepp. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 6. 107-116. 10.1142/S0218488598000094.

[13] Doshi, K. (2021, April 30). Image Captions with Attention in Tensorflow, Step-by-step. Medium.com. Retrieved December 28, 2022, from https://link.medium.com/s77SJEyi7vb

[14] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[16] Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[17] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2016.

[18] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[19] Chaitin, G.. (2013). Computing Machinery and Intelligence. Alan Turing: His Work and Impact. 551-621. 10.1016/B978-0-12-386980-7.50023-X.

[20] Hodosh, Micah, Peter Young, and Julia Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics." Journal of Artificial Intelligence Research 47 (2013): 853-899.

[21] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[22] J., Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. Commun. ACM 9, 1 (Jan. 1966), 36–45. https://doi.org/10.1145/365153.365168

[23] Liu, S., Bai, L., Hu, Y., Wang, H. (2018). Image captioning based on deep neural networks. In MATEC Web of Conferences (Vol. 232, p. 01052). EDP Sciences.

[24] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[25] Mo, Nan Yan, Li Zhu, Ruixi Xie, Hong. (2019). Class-Specific Anchor Based and Context-Guided Multi-Class Object Detection in High Resolution Remote Sensing Imagery with a Convolutional Neural Network. Remote Sensing. 11. 272. 10.3390/rs11030272.

[26] Abadi, Martín Barham, Paul Chen, Jianmin Chen, Zhifeng Davis, Andy Dean, Jeffrey Devin, Matthieu Ghemawat, Sanjay Irving, Geoffrey Isard, Michael Kudlur, Manjunath Levenberg, Josh Monga, Rajat Moore, Sherry Murray, Derek Steiner, Benoit Tucker, Paul Vasudevan, Vijay Warden, Pete Zhang, Xiaoqiang. (2016). TensorFlow: A system for large-scale machine learning.

[27] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)

[28] Graves, Alex Fernández, Santiago Schmidhuber, Jürgen. (2005). Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition.. 799-804.

[29] Ren, Shaoqing He, Kaiming Girshick, Ross Sun, Jian. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39. 10.1109/TPAMI.2016.2577031.

[30] Hubel, David Wiesel, Torsten. (2012). David Hubel and Torsten Wiesel. Neuron. 75. 182-4. 10.1016/j.neuron.2012.07.002.

[31] Papineni, Kishore Roukos, Salim Ward, Todd Zhu, Wei Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. 10.3115/1073083.1073135.

[32] Hyunju1 (2018) Image-Captioning [Source code] https://github.com/HyunJu1/Image-Captioning

Authorized licensed use limited to: KLE Technological University. Downloaded on June 20,2023 at 06:12:39 UTC from IEEE Xplore. Restrictions apply.