A Hybridized Deep Learning Method for Bengali Image Captioning

Mayeesha Humaira[1], Shimul Paul[2], Md Abidur Rahman Khan Jim[3], Amit Saha Ami[4], Faisal Muhammad Shah[5]
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

***Abstract*—An omnipresent challenging research topic in computer vision is the generation of captions from an input image. Previously, numerous experiments have been conducted on image captioning in English but the generation of the caption from the image in Bengali is still sparse and in need of more refining. Only a few papers till now have worked on image captioning in Bengali. Hence, we proffer a standard strategy for Bengali image caption generation on two different sizes of the Flickr8k dataset and BanglaLekha dataset which is the only publicly available Bengali dataset for image captioning. Afterward, the Bengali captions of our model were compared with Bengali captions generated by other researchers using different architectures. Additionally, we employed a hybrid approach based on InceptionResnetV2 or Xception as Convolution Neural Network and Bidirectional Long Short-Term Memory or Bidirectional Gated Recurrent Unit on two Bengali datasets. Furthermore, a different combination of word embedding was also adapted. Lastly, the performance was evaluated using Bilingual Evaluation Understudy and proved that the proposed model indeed performed better for the Bengali dataset consisting of 4000 images and the BanglaLekha dataset.**

***Keywords*—Bengali image captioning; hybrid architecture; InceptionResNet; Xception**

I. Introduction

An image is worth a thousand stories. It is effortless for humans to describe these stories but it is troublesome for a machine to portray them. To obtain captions from images it is necessary to combine computer vision and natural language processing. Previously lots of research has been done on image captioning but most of them were done in English. Research done on Image captioning using other languages [13], [15], [16] is still limited. Few works until now have been conducted on image captioning in Bengali [5], [23], [37] so we aim to explore image captioning in the Bengali language further.

About 215 million people worldwide speak in Bengali among those 196 million individuals are natives from India and Bangladesh. Bengali is the 7th most utilized language worldwide[1].As a result, it is momentous to generate image captions in Bengali alongside English. Moreover, most of the natives have no knowledge of English. Additionally, image captioning can be used to aid blind people by converting the text into speech blind people who can understand the image. Also, surveillance footage can be captioned in real-time so that theft, crime or accidents can be detected faster.

The main issue of image captioning in the Bengali language is the availability of a dataset. Most of the datasets available are in English. English datasets can be translated using manual labor or using machine translation. At any rate, manual translations have higher accuracy, they are extremely monotonous and troublesome. Machine translation on the other hand provides a better solution. In our experiment, we used a Machine translator such as Google translator[2] to translate English captions to Bengali and modified those sentences that were syntactically incorrect manually. Furthermore, we also utilized BanglaLekha[3] dataset which is the only publicly available Bengali dataset for image captioning till now. All the captions in this dataset were in Bengali and human annotated. We employed two approaches to captioning images in Bengali. Firstly, a hybrid model was used as demonstrated in Fig. 1 where two embedding layers were concatenated. Among those concatenated embedding one was GloVe [22] which utilize a pre-trained file in Bengali and another was fastText [7] which was trained on the vocabulary available. Secondly, two different models were trained to have a single embedding. One was conducted with only a trainable fastText embedding and the other experimented on GloVe embedding which was pre-trained in Bengali. For all three of the cases, InceptionResnetV2 [28] and Xception [38] was used as a Convolution Neural Network (CNN) to detect objects from images.

In this work, we proposed a hybridized Deep Learning method for Image captioning. This was achieved by concatenating two word embedding. The contribution of this paper is as follows:

- We introduced a hybridized method of image captioning where two word embedding pre-trained GloVe and fastText were concatenated.
- Experiments were carried on both our models using Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU). BiGRU has not been used before for image captioning using different languages other than English.
- Moreover, these two models have been tested on two Flickr8k datasets of varying lengths. One dataset contains 4000 images and the other contains 8000 images. To our best knowledge, no paper used Flickr8k full dataset translated in Bengali for image captioning.
- Additionally, our model was also tested on the BanglaLekha dataset which contains 9154 images.

[1]https://www.vistawide.com/languages/top_30_languages.htm

[2]https://translate.google.com/

[3]https://data.mendeley.com/datasets/rxxch9vw59/2

- Lastly, it was shown that our proposed hybrid model achieved higher BLEU scores for both the Flickr4k-BN dataset and the BanglaLekha dataset.

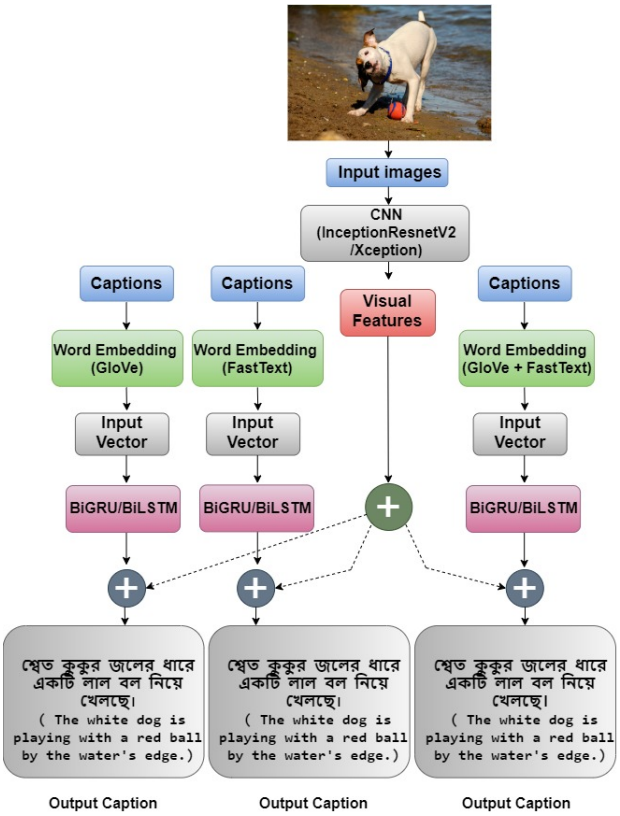

Fig. 1. Illustration of Hybridized (Right) Model and Model with Single Embedding FastText or GloVe (Left).

II. Related Work

This section depicts the progress in image captioning. Hitherto, many kinds of research have been conducted and many models have been developed to get captions that are syntactically corrected. The authors in [2] presented a model that deems the probabilistic distribution of the next word using previous word and image features. On the other hand, H. Dong et al. [6] proposed a new training method Image-Text-Image which amalgamate text-to-image and image-to-text synthesis to revamp the performance of text-to-image synthesis. Furthermore, J. Aneja [21] and S. J. Rennie [25] adapted the attention mechanism to generate caption. For vision part of image captioning Vgg16 were used by most of the papers [2], [11], [24], [25], [27], [30] as CNN but some of them also used YOLO [9], Inception V3 [6], [31], AlexNet [24], [30] ResNet [11], [18], [24] or Unet [4] as CNN for feature extraction. Concurrently, LSTM [6], [9], [11], [17], [31] was used by most of the papers for generating the next word in the sequence. However, some of the researcher also utilized RNN [19] or BiLSTM [4], [30]. Moreover, P. Blandfort et al. [32] systematically characterize diverse image captions that appear "in the wild" in order to understand how people caption images naturally. Alongside English researchers also generated captions in Chinese [15], [16], Japanese [1], Arabic [12], Bahasa Indonesia [13], Hindi [26] German [29] and Bengali [5], [23]. M. Rahman et al. [23] generated image caption in Bengali for the first time followed by T. Deb et al. [5]. Researchers of paper [23] used VGG-16 to extract image features and stacked LSTMs. On the contrary, researchers of paper [5] generated image caption using InceptionResnetV2 or VGG-16 and LSTM. They utilized 4000 images of the Flickr8k dataset to generate captions. We modified the merge model adapted by paper [5] to get much better and fluent captions in Bengali.

Only three works have been done on image captioning in Bengali till now. In [23], author's first paper, was where in image captioning in Bengali followed by [5] and [37]. Rahman et al. [23] have aimed to outline an automatic image captioning system in Bengali called 'Chittron'. Their model was trained to predict Bengali caption from input image one word at a time. The training process was carried out on 15700 images of their own dataset BanglaLekha. In their model Image feature vector and words converted to vectors after passing them through the embedding, the layer was fed to the stacked LSTM layer. One drawback of their work was that they utilized sentence BLEU score instead of Corpus BLEU score. On the other hand, Deb et al. [5] illustrated two models Par-Inject Architecture and Merge Architecture for image captioning in Bengali. In the Par-Inject model image, feature vectors were fed into intermediate LSTM and the output of that LSTM and word vectors were combined and fed to another LSTM to generate caption in Bengali. Whereas, in the Merge model image feature vectors and words vector were combined and passed to an LSTM without the use of an intermediate LSTM. They utilized 4000 images of the Fickr8k dataset and the Bengali caption their models generated were not fluent. Paper [37] used a CNN-RNN based model where VGG-16 was used as CNN and LSTM with 256 channels was used as RNN. They trained their model on the BanglaLekha dataset having 9154 images.

To overcome the above mentioned drawbacks of fluent captions we conducted our experiment using a hybridized approach. Moreover, we used 8000 images of the Flickr8k dataset alongside the Flickr4k dataset. We further validated the performance of our model using the human annotated BanglaLekha dataset.

III. Our Approach

We employed an Encoder-Decoder approach where both InceptionResnetV2 and Xception were used separately in different experimental setups to Encode Images to feature vectors and different word embedding were used to convert vocabulary to word vectors. Image feature vectors and word vectors after passing through a special kind of RNN were merged and passed to a decoder to predict captions word by word this process is illustrated in Fig. 2. We propose a hybrid model that consists of two embedding layers unlike the merge model [5]. We also conducted experiments on the merged model having either pre-trained GloVe [22] or trainable fastText [7] embedding. To be more precise, we trained the merge model using three settings as shown in Fig. 1.

Our proposed hybrid model is shown in Fig. 3. It consists of two part which is encoder and decoder.

- Encoder
  The encoder comprised of two parts one for han-

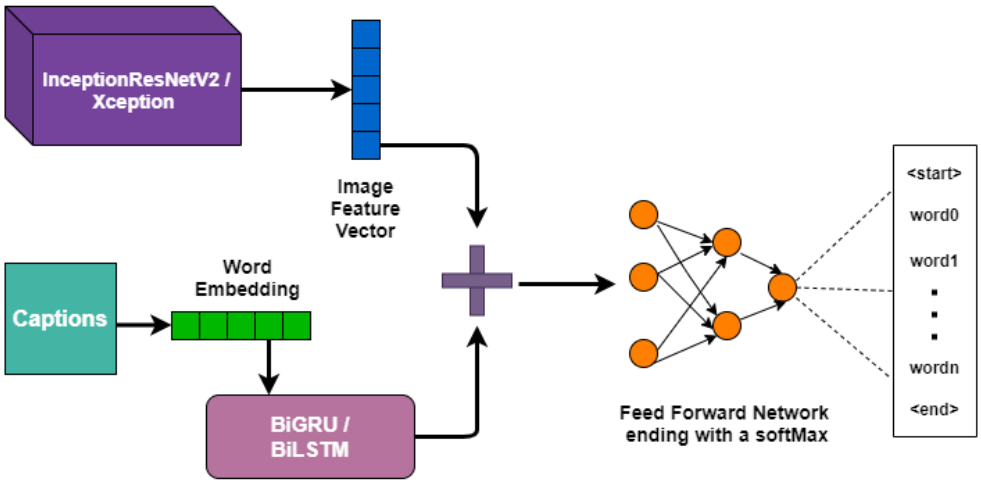

Fig. 2. An overview of how captions are generated word by word using our model.

dling image features and another for handling word sequence pair. Firstly, image features were extracted using InceptionResnetV2 [28] or Xception [38]. These image features were preceded down to a dropout layer followed by a fully connected layer and then another dropout layer. A fully connected layer was used to reduce the dimension of the image feature vector from 1536 or 2048 to 256 to match the dimension of word prediction output. Secondly, Input word sequence pairs are feed to two embedding layers one was pre-trained GloVe embedding and another was fastText which was not pre-trained. Both embeddings were used to convert words to vectors of dimension 100. The vector from the two embeddings was then passed through a separate dropout layer followed by either BiLSTM or BiGRU of dimension 128. To match the dimension of visual feature vector output these vectors were passed through an additional fully connected layer of dimension 256. These two outputs were then concatenated. This concatenated output was then mapped to the visual part of the encoder using another concatenation and then forwarded to the decoder.

- Decoder
  The decoder is a Feed Forward Network which ends with a SoftMax. It takes the concatenated output of the encoder as input. This input was first passed through a fully connected layer of 256 dimensions followed by a dropout layer. Finally, via probabilistic Softmax function outputs the next word in the sequence. The SoftMax greedily selects the word with maximum probability.

IV. Experimental Setup

This section narrates the total strategy adapted to obtain captions from images. Also, different tuning techniques availed are described here.

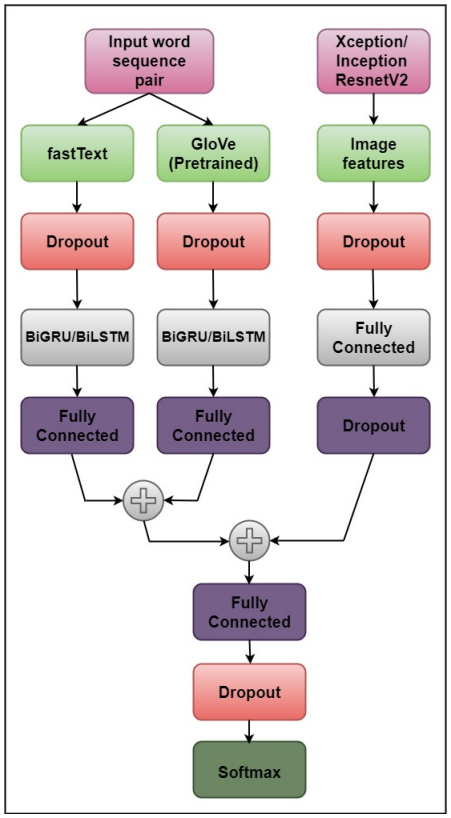

Fig. 3. Proposed Hybrid Model.

A. Dataset Processing

Flickr8k dataset has 8091 images of which 6000 (75%) images are employed for training, 1000 (12.5%) images for validation and 1000 (12.5%) images are used for testing. Moreover, with each image of the Flickr8K dataset five ground truth captions describing the image are designated which adds up to a total of 40455 captions for 8091 images. For image captioning in Bengali, those 40455 captions were converted

to Bengali language using Google Translator. Unfortunately, some of the translated captions were syntactically incorrect. Hence, we manually checked all 40455 translated captions and corrected them. We utilized these 8000 images as well as selected 4000 images as done by Deb et al. [5] in Bengali(Flickr4k-BN and Flickr8k-BN). These 4000 images were selected based on the frequency of words in those 40455 captions. Using POS taggers most frequent nouns Bengali words were identified from ground truth captions. The most frequent words in the Bengali Flickr8k dataset are shown in Fig. 4 for Bengali and English respectively. 4000 images analogous to these words are selected and made two small datasets Flickr4k-BN.

We also utilized the BanglaLekha dataset which consists of 9154 images. It is the only available Bengali dataset till now. All its captions are human annotated. One problem with this dataset is that it has only two captions associated with each image resulting in 18308 captions for those 9154 images. Hence, vocabulary size is lower than Flickr4k-BN and Flickr8k-BN. Flickr8k-BN consists of 12953 unique Bengali words, Flickr4k-BN consist of 6420 unique Bengali words and BanglaLekha consists of 5270 unique Bengali words. It can be seen that the BanglaLekha dataset has a vocabulary size even lower than Flickr4k-BN. Hence, we employed the Flickr8k-BN dataset alongside Flickr4k-BN and BanglaLekha datasets. The split ratio of all three datasets for training, testing and validating are shown in Table I.

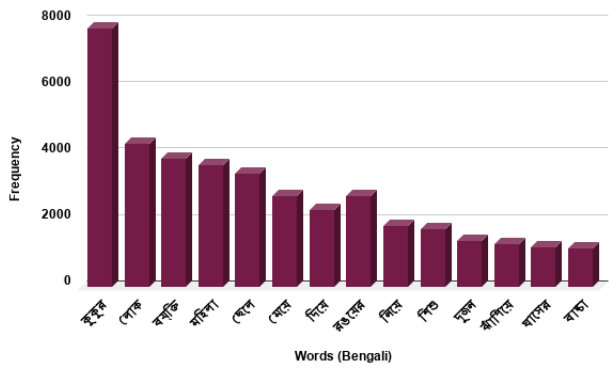

Fig. 4. Illustration of Most Frequent Noun Bengali Words in Flickr8k Bengali Dataset.

B. Image Feature Extraction

One essential part of image captioning is to extract features from given images. This task is achieved using Convolutional Neural Network architectures. These architectures are used to detect objects from images. They can be trained on a large number of images for extracting image features. This training process requires an enormous number of images and time. Due to the shortage of a large number of images, we utilized Convolutional Neural Network architecture which was pre-trained on more than a million images from the ImageNet [33] dataset in our model known as InceptionResnetV2 [28] and Xception [38]. These two pre-trained architectures were used separately for different experimental setups. The reason for using InceptionResnetV2 and Xception is that these models can achieve higher accuracy at lower epochs. The last layer which is used for prediction purposes of this pre-trained of InceptionResnetV2 model is pulled out and the last two layers of the pre-trained Xception model were pulled out. Finally, the average pooling layer was used to extract image features and convert them into a feature vector of 1536 dimensions for InceptionResnetV2 and 2048 dimensions for Xception. All the images are given an input shape of 299x299x3 before entering the InceptionResnetV2 model. Here 3 represents the three-color channels R, G and B.

C. Embeddings

Handling word sequences requires word embedding that can convert words to vectors before passing them to special recurrent neural networks (RNN). In our model GloVe [22] and fastText [7] have been used as an embedding.

- GloVe is a model for distributed word representation. The model employs an unsupervised learning algorithm for acquiring vector representations for words. This is achieved by mapping words into a meaningful space where the distance between words is related to semantic similarity.
- fastText is a library for the learning of word embeddings and text classification created by Facebook's AI Research (FAIR) lab. The model employs unsupervised learning or supervised learning algorithms for obtaining vector representations for words. fastText yields two models for computing word representations namely skipgram and cbow. Skipgram model learns to forecast a target word using the nearby word. conversely, cbow model forecasts the target word according to its context where context depicts a bag of the words contained in a fixed size window around the target word.

Both GloVe and fastText have pre-trained word vectors that are trained over a large vocabulary. These embeddings can also be trained. In the hybrid model shown in Fig. 3, two embeddings have been used GloVe and fastText. There GloVe was pre-trained but fastText has been trained on vocabulary available in the dataset. Trainable fastText instead of pre-trained fastText was used to enrich the vocabulary with words in Flickr8k and BanglaLekha datasets. Also, results of pre-trained fastText have already been demonstrated by Deb et al. [5]. The combination of two embedding leads to redundancy of words but it gives fluent caption in Bengali as the vocabulary size increases. On the other hand, pre-trained files for both GloVe and fastText in the hybrid model will give much greater redundancy and the vocabulary size becomes small as the vocabulary does not contain unique words in the dataset.

Two other models were trained alongside the hybrid model. Unlike the hybrid model, these two models had a single embedding either a trainable fastText embedding or a pre-trained GloVe embedding. GloVe file "bn_glove.39M.100d"[4] pre-trained in Bangali Language was used for Bengali datasets.

D. Word Sequence Generation

Flickr8k dataset has five captions associated with each image and BanglaLekha has two captions associated with each

[4]https://github.com/sagorbrur/GloVe-Bengali

TABLE I. DISTRIBUTION OF DATA FOR THREE BENGALI DATASET USED. SAME DISTRIBUTION WAS USED FOR FLICKR8K ENGLISH AND BENGALI DATASETS.

Dataset	Total Image	Training	Validation	Testing
Flickr4k	4000	2400 (60%)	800 (20%)	800 (20%)
Flickr8k	8000	6000 (75%)	1000 (15%)	1000 (15%)
BanglaLekha	9154	7154 (78%)	1000 (11%)	1000 (11%)

image. One of the difficult tasks of image captioning is to make the model learn how to generate these sentences. Two different types of special Recurrent Neural Network (RNN) were used to train the model to generate the next word in the sequence of a caption. The input and output sizes were fixed to the maximum length of the sentence present in the dataset. In the case of Flickr4k-BN and Flickr8k-BN maximum length was 23. On the other hand, two different maximum lengths of the sequence 40 and 26 were used for the BanglaLekha dataset. Reducing the maximum sequence length significantly increased the evaluation scores. While training if any sentence were generated having a length less than the maximum length zero-padding was applied to make that sentence length equal to the fixed length. Additionally, an extra start token and end token is added to the sequence pair for identification in the training process. During training, image features vector and previous words converted to vector using embedding layer were used to generate the next word in the sequence probabilistic Softmax with the help of different types of RNN. Fig. 5 illustrates the input and output pair.

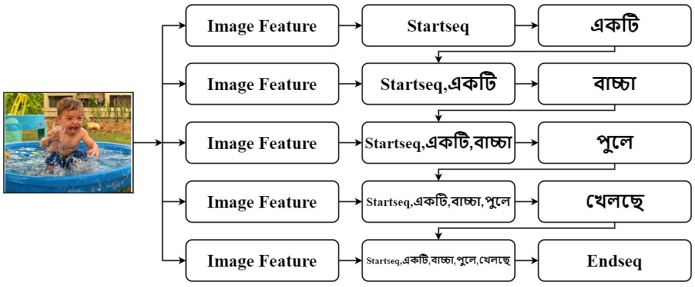

Fig. 5. Demonstrates How Word Sequences are Generated.

Due to the limitation of the basic Recurrent Neural Network (RNN) [34] to retrain long term memory a better approach was taken by Deb et al. [5] which uses Long Short-Term Memory (LSTM). However, LSTM [10] only preserve preceding words but for proper sentence generation succeeding words are also necessary. As a result, our model uses Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU) which are illustrated in Fig. 6. Each box marked as A or A' was either a Long Short-Term Memory (LSTM) or a Gated Recurrent Unit GRU [8] unit. X [0...i] are the input words and Y [0...i] are the output words. Y [0...i] are determined using the Eq. 1.

$$\hat{y}^{<t>} = g(W_y[\overrightarrow{a}^{<t>}\overleftarrow{a}^{<t>}] + b_y) \tag{1}$$

Where $\hat{y}^{<t>}$ is the output at time t when activation function g is applied to recurrent component's weight W_y and bias by with both forward activation $\overrightarrow{a}^{<t>}$ at time t and backward activation $\overleftarrow{a}^{<t>}$ at time t.

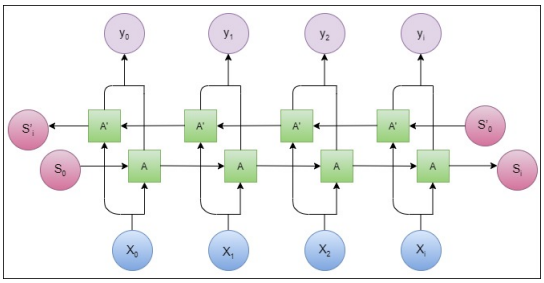

Fig. 6. Illustrates Bidirectional RNN having $X_0 \ldots i$ as Input and $Y_0 \ldots i$ as output. A and A' boxes are both either BiLSTM or BiGRU where A is the Forward Recurrent Component an A' is the Backward Recurrent Component.

- GRU is a special type of RNN. Reset and update the gate of a GRU helps to solve the vanishing gradient problem of RNN. The update gate of GRU seeks how much information from the previous units must be forwarded. The update gate adopted is computed by the following formula:

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{2}$$

where z_t is update gate output at the current timestamp, W_z is weight matrix at update gate, h_{t-1} information from previous units, and x_t is input at the current unit.
The reset gate is used by the model to find how much information from the previous units to forget. The reset gate is computed by the following formula:

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{3}$$

where r_t is reset gate output at current timestamp, W_r is weight matrix at reset gate, h_{t-1} information from previous units, and x_t is input at the current unit.
Current memory content used to store the relevant information from the previous units. It is calculated as follows:

$$\tilde{h}_t = tanh(W.[r_t * h_{t-1}, x_t]) \tag{4}$$

where $\tilde{h}_t$ is current memory content, W is weight at current unit, r_t is reset gate output at current timestamp, h_{t-1} is information from previous units, and x_t is input at the current unit.
Final memory at the current unit is a vector used to store the final information for the current unit and pass it to the next layer. It is calculated using a formula:

$$h_t = (1 - z_t) * h_{t-1} + z_t\tilde{h}_t \tag{5}$$

where h_t is final memory at the current unit, z_t is update gate output at current timestamp, h_{t-1} is information from previous units, and $\tilde{h}_t$ is current memory content.

- LSTM is another Special type of RMNN. Unlike the GRU the LSTM has three gates, namely, the forget gate, update gate and the output gate. The equations for the gates in LSTM are:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

where i_t represents input gate, f_t represents forget gate, o_t represents output gate, σ represents sigmoid function, W_x represents weight of the respective gate(x) neurons, h_{t-1} represents output of previous LSTM block at timestamp t-1, x_t represents input at current timestamp and b_x represents biases for the respective gates(x).
Input gate tells what new information is going to be stored in cell state. Forget gate determine what information to throw away from cell state and Output get is used to provide output at timestamp t. The equations for the cell state, candidate cell state and the final output are:

$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{9}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{10}$$

$$h_t = o_t * tanh(c^t) \tag{11}$$

where c_t represents cell state at timestamp t and $\tilde{c}_t$ represent candidate for cell state at timestamp t. candidate timestamp must be generated to get memory vector for current timestamp c_t. Then the cell state is passed through a activation function to generate h_t. Finally, h_t is passed through a softMax layer to get the output y_t.

E. Hyperparameter Selection

One major problem of machine learning is overfitting. Overfit models have high variance. These models cannot generalize well. As a result, this is a huge problem for image captioning. We observed the performance of our model and noticed that it was suffering from overfitting rather than underfitting. To minimize this overfitting problem some hyperparameter tuning has been adapted in our model. Firstly, different values of dropout [35] have been used for sequence model image features and decoder. Dropouts help prevent overfitting. For feature extractor dropout value of 0.0 was used, a dropout of 0.3 was used for the sequence model and in the case of decoder dropout value of 0.5 was utilized. Secondly, different activation functions were employed for different fully connected layers. For example, regarding the feature extractor model and decoder ELU [3] activation function was availed and for the sequence model, ReLU [36] activation function was employed. Thirdly, we employed external validation to provide an unbiased evaluation and ModelCheckpoint was availed to save models that had minimum validation loss. On the other hand, ReduceLROnPlateau was used for models that had Xception as CNN. Moreover, Adam optimizer [14] was utilized and the models were trained for 50 and 100 epochs having learning rates of 0.0001 and 0.00001. A short summary of the hyperparameters adapted in different models are shown in Table II and the loss plot of BanglaLekha dataset and Flickr8K-BN dataset are ornamented in Fig. 7 and Fig. 8, respectively. From these plots, it can be seen that the model converges towards epoch 100. Another important factor that improved the result was maximum sentence length. In the BnglaLekha only a few sentences had lengths greater than 26. As a result, we took a maximum length of sentences in this dataset to 26. This enhanced the evaluation scores greatly.

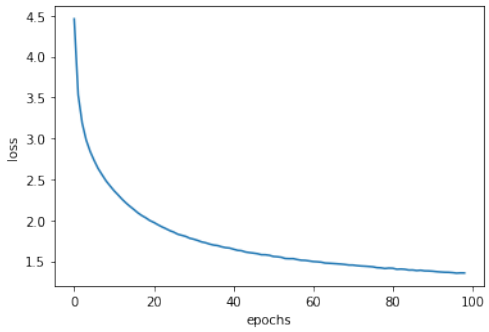

Fig. 7. Loss Plot of BanglaLekha Dataset for 100 epochs.

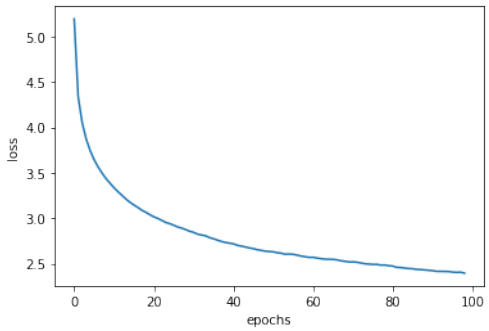

Fig. 8. Loss Plot of Flickr8k-BN Dataset for 100 epochs.

V. Analysis

We implemented the algorithm using Keras 2.3.1 and Python 3.8.1. Additionally, we ran our experiments on GPU RTX 2060. Our code and Bengali Flickr8k dataset is given in GitHub[5]. We translated the Flickr8k dataset to Bengali using Google Translator Like that done by [16]. Bilingual Evaluation Understudy (BLEU) [20] score was used to evaluate the performance of our models as it is the most wielded metric nowadays to evaluate the caliber of text. It depicts how normal sentences are compared with human generated sentences. It is broadly utilized to evaluate the performance of Machine translation. Sentences are compared based on modified n-gram precision for generating BLEU scores. BLEU scores are computed using the following equations:

$$P(i) = \frac{Matched(i)}{H(i)} \tag{12}$$

[5]https://github.com/MayeeshaHumaira/A-Hybridized-Deep-Learning-Method-for-Benga

TABLE II. HYPERPARAMETERS ADAPTED IN DIFFERENT MODELS.

Search Type	Model	Learning Rate	Loss Function	Callback	Epoch
Greedy	Xception +BiLSTM	0.00001	Sparse Categorical Crossentrophy	ReduceLROnPlateau	100
	InceptionResnetV2 +BiLSTM	0.0001	Categorical Crossentrophy	ModelCheckpoint	50
Beam=3	Xception +BiLSTM	0.00001	Sparse Categorical Crossentrophy	ReduceLROnPlateau	100
Beam=5	Xception +BiLSTM	0.00001	Sparse Categorical Crossentrophy	ReduceLROnPlateau	100

where P(i) is the precision that is for each i-gram where i = 1, 2, ...N, the percentage of the i-gram tuples in the hypothesis that also occur in the references is computed. H(i) is the number of i-gram tuples in the hypothesis and Matched(i) is computed using the following formula:

$$Matched(i) = \sum_{t_i} \min\{C_h(t_i), \max_j C_{hj}(t_i)\} \quad (13)$$

where t_i is an i-gram tuple in hypothesis h, $C_h(t_i)$ is the number of times t_i occurs in the hypothesis, $C_{hj}(t_i)$ is the number of times t_i occurs in reference j of this hypothesis.

$$\rho = exp\{\min(0, \frac{n-L}{n})\} \quad (14)$$

where ρ is brevity penalty to penalize short translation, n is the length of the hypothesis and L is the length of the reference. Finally, the BLEU score is computed by:

$$BLEU = \rho\{\prod_{i=1}^{N} P(i)\}^{\frac{1}{N}} \quad (15)$$

Two different search types Greedy and Beam search were used to compute these BLEU scores. In a Greedy search word with maximum probability is chosen as the next word in the sequence. On the other hand, Beam search considers n words to choose from for the next word in the sequence. Where n is the width of the beam. For our experiment, we considered beamwidth of 3 and 5. We computed 1-gram BLEU (BLEU-1), 2-gram BLEU (BLEU-2), 3-gram BLEU (BLEU-3), 4-gram BLEU (BLEU-4) for various architectures. These are illustrated in Table III, Table IV and Table V.

Performance of the proposed Hybrid architecture and single embedding GloVe or fastText on Flickr4k-BN dataset consisting of 4000 data for Bengali are demonstrated in Table III. From Table III it can be stated that the Hybrid model performed better for both BiLSTM and BiGRU on the Bengali dataset than only GloVe and only fastText word embedding. Moreover, we obtained better BLEU scores than paper [5]. The greedy search was employed to compute these BLEU scores.

Consequently, the performance of the single embedding GloVe or fast Text and hybrid architecture on Flickr8k-BN dataset consisting of 8000 data and BanglaLekha dataset are displayed in Table IV and Table V, respectively. There also it can be observed that the proposed Hybrid model performed better for both BiGRU and BiLSTM than the other models. The Highest BLEU score was obtained using BiLSTM on Flickr4k-BN and Flickr8k-BN as a result the captions generated by the Hybrid model for both datasets are illustrated in Fig. 9. Furthermore, our proposed Hybrid model also gave the highest BLEU scores for the BanglaLekha dataset for both BiLSTM and BiGRU as shown in Table V. From there it can be observed that Xception and the learning rate played a vital role in increasing the BLEU scores. These scores were even better than BLEU scores obtained by paper [37]. Table VI illustrates a brief comparison of the BLEU scores obtained by our proposed model and the scores obtained by other papers. From there it can be observed that our proposed Hybrid model indeed gave a better performance. The captions generated by these models for test images of the BanglaLekha dataset are shown in Fig. 10. Flickr8k-BN dataset consisting of 8000 images were not previously used by any other papers for generating captions in Bengali.

Fig. 9. Illustration of Captions Generated by Best Performing Hybrid Architecture using Fickr4k-BN and Flickr8k-BN Datasets.

TABLE III. RESULT OF INCEPTIONRESNETV2 USED BY FLICKR4K-BN

Experimental Model	RNN	Training Accuracy	Validation Accuracy	BLEU			
				1	2	3	4
Proposed	BiLSTM	0.421	0.387	**0.661**	**0.508**	**0.382**	**0.229**
Hybrid architecture	BiGRU	0.432	0.386	0.660	0.503	0.371	0.215
GloVe	BiLSTM	0.432	0.388	0.644	0.491	0.369	0.220
	BiGRU	0.429	0.386	0.651	0.497	0.373	0.223
fastText	BiLSTM	0.414	0.372	0.638	0.490	0.370	0.219
	BiGRU	0.426	0.379	0.653	0.505	0.381	0.226

TABLE IV. BLEU SCORES OBTAINED USING FLICKR8K-BN DATASET

Search Type	Learning Rate	Word Embedding	Experimental Model	BLEU			
				1	2	3	4
Greedy	0.00001	Hybrid	Xception+BiLSTM	0.504	0.326	0.232	0.119
			Xception+BiGRU	0.536	0.352	0.246	0.126
		GloVe	Xception+BiLSTM	0.539	0.356	0.249	0.129
			Xception+BiGRU	0.532	0.352	0.241	0.121
		fastText	Xception+BiLSTM	0.190	0.055	0.000	0.000
			Xception+BiGRU	0.194	0.068	0.012	0.000
Greedy	0.0001	Hybrid	InceptionResnetV2+BiLSTM	**0.540**	**0.370**	**0.268**	**0.145**
			InceptionResnetV2+BiGRU	0.526	0.360	0.261	0.141
		GloVe	InceptionResnetV2+BiLSTM	0.534	0.369	0.265	0.142
			InceptionResnetV2+BiGRU	0.512	0.350	0.255	0.138
		fastText	InceptionResnetV2+BiLSTM	0.528	0.363	0.269	0.140
			InceptionResnetV2+BiGRU	0.530	0.362	0.260	0.140
Beam=3	0.00001	Hybrid	Xception+BiLSTM	0.416	0.246	0.176	0.089
			Xception+BiGRU	0.414	0.247	0.178	0.093
		GloVe	Xception+BiLSTM	0.395	0.239	0.174	0.089
			Xception+BiGRU	0.404	0.245	0.178	0.090
		fastText	Xception+BiLSTM	0.034	0.000	0.000	0.000
			Xception+BiGRU	0.059	0.003	0.001	0.000
Beam=5	0.00001	Hybrid	Xception+BiLSTM	0.409	0.240	0.175	0.090
			Xception+BiGRU	0.403	0.239	0.171	0.089
		GloVe	Xception+BiLSTM	0.377	0.226	0.162	0.079
			Xception+BiGRU	0.393	0.241	0.172	0.085
		fastText	Xception+BiLSTM	0.034	0.000	0.000	0.000
			Xception+BiGRU	0.059	0.003	0.001	0.000

VI. CONCLUSION

In this work, we exhibited a notion for automatically generating caption from an input image in Bengali. Firstly, a detailed description of how the Flickr8k dataset was translated in Bengali and distributed into a dataset of two sizes was presented. Secondly, how image features were extracted and the different combinations of word embedding utilized were also conferred. Moreover, the reasons for using a special kind of word sequence generator was elucidated. Furthermore, different parts of the proposed architecture were ornamented. Finally, using the BLEU score it was authenticated that the proposed architecture performs better for both Flickr4k-Bn and BanglaLekha datasets. This validates the fact that image captioning using the Bengali language can be refined further in the future. We will try to adapt the visual attention and transformer model in the near future for better feature extraction and getting more precise captions. Additionally, we aim to make our own dataset having five captions with each image, unlike the BanglaLekha dataset that has two captions associated with each image to enrich the vocabulary of our dataset.

REFERENCES

[1] Y. Yoshikawa, Y. Shigeto, and A. Takeuchi, "STAIR captions: Constructing a large-scale Japanese image caption dataset," ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 2, pp. 417–421, 2017, doi: 10.18653/v1/P17-2066.

[2] J. Gu, G. Wang, J. Cai, and T. Chen, "An Empirical Study of Language CNN for Image Captioning," Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-October, pp. 1231–1240, 2017, doi: 10.1109/ICCV.2017.138.

[3] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc., pp. 1–14, 2016.

[4] W. Cui et al., "Landslide image captioning method based on semantic gate and bi-temporal LSTM," ISPRS Int. J. Geo-Information, vol. 9, no. 4, 2020, doi: 10.3390/ijgi9040194.

[5] T. Deb et al., "Oboyob: A sequential-semantic Bengali image captioning engine," J. Intell. Fuzzy Syst., vol. 37, no. 6, pp. 7427–7439, 2019, doi: 10.3233/JIFS-179351.

[6] H. Dong, J. Zhang, D. Mcilwraith, and Y. Guo, "I2T2I: LEARNING TEXT TO IMAGE SYNTHESIS WITH TEXTUAL DATA AUGMENTATION."

[7] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval., pp. 3483–3487, 2019.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555.

[9] M. Han, W. Chen, and A. D. Moges, "Fast image captioning using LSTM," Cluster Comput., vol. 22, pp. 6143–6155, May 2019, doi: 10.1007/s10586-018-1885-9.

TABLE V. BLEU SCORES OBTAINED USING BANGLALEKHA DATASET

Search Type	Learning Rate	Word Embedding	Experimental Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4
Greedy	0.00001	Hybrid	Xception+BiLSTM	0.673	0.525	0.454	0.339
			Xception+BiGRU	**0.674**	**0.527**	**0.454**	**0.344**
		GloVe	Xception+BiLSTM	0.612	0.453	0.380	0.265
			Xception+BiGRU	0.610	0.454	0.383	0.272
		fastText	Xception+BiLSTM	0.618	0.463	0.389	0.277
			Xception+BiGRU	0.624	0.473	0.402	0.290
Greedy	0.0001	Hybrid	InceptionResnetV2+BiLSTM	0.568	0.396	0.287	0.160
			InceptionResnetV2+BiGRU	0.571	0.402	0.301	0.173
		GloVe	InceptionResnetV2+BiLSTM	0.568	0.401	0.301	0.174
			InceptionResnetV2+BiGRU	0.570	0.403	0.303	0.176
		fastText	InceptionResnetV2+BiLSTM	0.553	0.390	0.291	0.169
			InceptionResnetV2+BiGRU	0.567	0.398	0.300	0.171
Beam=3	0.00001	Hybrid	Xception+BiLSTM	0.434	0.344	0.303	0.234
			Xception+BiGRU	0.411	0.324	0.286	0.221
		GloVe	Xception+BiLSTM	0.383	0.285	0.245	0.176
			Xception+BiGRU	0.401	0.302	0.263	0.196
		fastText	Xception+BiLSTM	0.419	0.320	0.283	0.214
			Xception+BiGRU	0.434	0.329	0.293	0.221
Beam=5	0.00001	Hybrid	Xception+BiLSTM	0.420	0.335	0.297	0.232
			Xception+BiGRU	0.399	0.316	0.280	0.219
		GloVe	Xception+BiLSTM	0.368	0.273	0.234	0.170
			Xception+BiGRU	0.385	0.292	0.256	0.194
		fastText	Xception+BiLSTM	0.422	0.324	0.288	0.219
			Xception+BiGRU	0.429	0.326	0.291	0.222

TABLE VI. A BRIEF COMPARISON OF BLEU SCORES FOR EXISTING MODELS AND OUR PROPOSED HYBRID MODEL.

Dataset	Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4
BanglaLekha	VGG-16+LSTM [37]	66.7	43.6	31.5	23.8
	Xception+BiGRU (Our Hybrid Model)	**0.674**	**0.527**	**0.454**	**0.344**
Flickr8k(4000 images)	Merge Bengali(Inception+LSTM) [5]	0.62	0.45	0.33	0.22
Flickr4k-BN	Our Hybrid Model (InceptionResnetV2+BiLSTM)	**0.661**	**0.508**	**0.382**	**0.229**

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

[11] K. Xu, H. Wang, and P. Tang, "IMAGE CAPTIONING WITH DEEP LSTM BASED ON SEQUENTIAL RESIDUAL Department of Computer Science and Technology , Tongji University , Shanghai , P . R . China Key Laboratory of Embedded System and Service Computing , Ministry of Education ," no. July, pp. 361–366, 2017.

[12] V. Jindal, "Generating Image Captions in Arabic Using Root-Word Based Recurrent Neural Networks and Deep Neural Networks." Available: www.aaai.org.

[13] A. A. Nugraha, A. Arifianto, and Suyanto, "Generating image description on Indonesian language using convolutional neural network and gated recurrent unit," 2019 7th Int. Conf. Inf. Commun. Technol. ICoICT 2019, pp. 1–6, 2019, doi: 10.1109/ICoICT.2019.8835370.

[14] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

[15] W. Lan, X. Li, and J. Dong, "Fluency-guided cross-lingual image captioning," MM 2017 - Proc. 2017 ACM Multimed. Conf., pp. 1549–1557, 2017, doi: 10.1145/3123266.3123366.

[16] X. Li, W. Lan, J. Dong, and H. Liu, "Adding Chinese captions to images," ICMR 2016 - Proc. 2016 ACM Int. Conf. Multimed. Retr., pp. 271–275, 2016, doi: 10.1145/2911996.2912049.

[17] C. Liu, F. Sun, and C. Wang, "MMT: A multimodal translator for image captioning," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10614 LNCS, p. 784, 2017.

[18] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain Images with Multimodal Recurrent Neural Networks," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1410.1090.

[19] Q. You, H. Jin, Z. Wang, ... C. F.-P. of the I., and undefined 2016, "Image captioning with semantic attention," openaccess.thecvf.com Available: http://openaccess.thecvf.com/.

[20] K. Papineni, S. Roukos, T. Ward, W. Zhu, and Y. Heights, "IBM Research Report Bleu: a Method for Automatic Evaluation of Machine Translation," Science (80-.)., vol. 22176, no. February, pp. 1–10, 2001, doi: 10.3115/1073083.1073135.

[21] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional Image Captioning," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5561–5570, 2018, doi: 10.1109/CVPR.2018.00583.

[22] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf., no. June, pp. 1532–1543, 2014, doi: 10.3115/v1/d14-1162.

[23] M. Rahman, N. Mohammed, N. Mansoor, and S. Momen, "Chittron: An Automatic Bangla Image Captioning System," Procedia Comput. Sci., vol. 154, pp. 636–642, 2018, doi: 10.1016/j.procs.2019.06.100.

[24] S. Liu, L. Bai, Y. Hu, and H. Wang, "Image Captioning Based on Deep Neural Networks," MATEC Web Conf., vol. 232, pp. 1–7, 2018, doi: 10.1051/matecconf/201823201052.

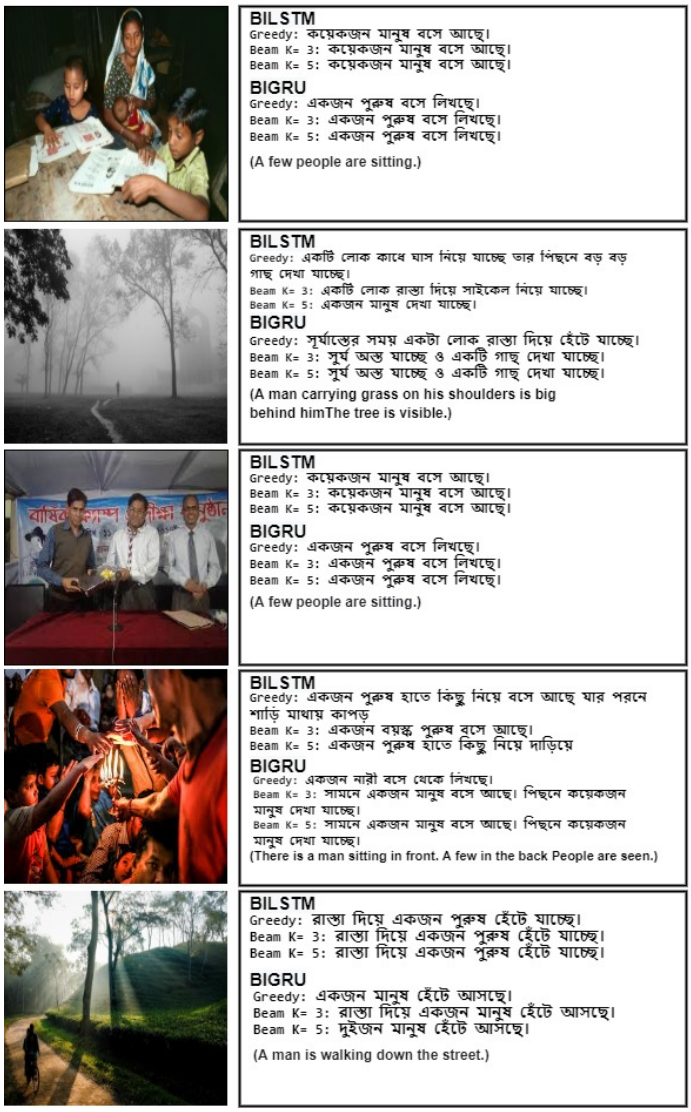

Fig. 10. Illustration of Captions Generated by Best Performing Hybrid Architecture using BanglaLekha Dataset.

[25] [1] K. Xu et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." Available: http://proceedings.mlr.press/v37/xuc15.

[26] S. R. Laskar, R. P. Singh, P. Pakray, and S. Bandyopadhyay, "English to Hindi Multi-modal Neural Machine Translation and Hindi Image Captioning," pp. 62–67, 2019, doi: 10.18653/v1/d19-5205.

[27] R. Subash, R. Jebakumar, Y. Kamdar, and N. Bhatt, "Automatic image captioning using convolution neural networks and LSTM," J. Phys. Conf. Ser., vol. 1362, no. 1, 2019, doi: 10.1088/1742- 6596/1362/1/012096.

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 31st AAAI Conf. Artif. Intell. AAAI 2017, pp. 4278–4284, 2017.

[29] A. Jaffe, "Generating Image Descriptions using Multilingual Data," pp. 458–464, 2018, doi: 10.18653/v1/w17-4750.

[30] C. Wang, H. Yang, and C. Meinel, "Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning," ACM Trans. Multimed. Comput. Commun. Appl., vol. 14, no. 2s, 2018, doi: 10.1145/3115432.

[31] Y. Xian and Y. Tian, "Self-Guiding Multimodal LSTM - When We Do Not Have a Perfect Training Dataset for Image Captioning," IEEE Trans. Image Process., vol. 28, no. 11, pp. 5241–5252, 2019, doi: 10.1109/TIP.2019.2917229.

[32] P. Blandfort, T. Karayil, D. Borth, and A. Dengel, "Image captioning in the wild: How people caption images on flickr," MUSA2 2017 - Proc. Work. Multimodal Underst. Soc. Affect. Subj. Attrib. co-located with MM 2017, pp. 21–29, 2017, doi: 10.1145/3132515.3132522.

[33] L. Fei-Fei, J. Deng, and K. Li, "ImageNet: Constructing a large-scale image database," J. Vis., vol. 9, no. 8, pp. 1037–1037, 2010, doi: 10.1167/9.8.1037.

[34] M. Tanti, A. Gatt, and K. Camilleri, "What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?," pp. 51–60, 2018, doi: 10.18653/v1/w17-3506.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," 2014. doi: 10.5555/2627435.2670313.

[36] [1] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines." Available: https://www.cs.toronto.edu/ hinton/absps/reluICML.pdf.

[37] A. H. Kamal, M. A. Jishan and N. Mansoor, "TextMage: The Automated Bangla Caption Generator Based On Deep Learning," 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 2020, pp. 822-826, doi: 10.1109/DASA51403.2020.9317108.

[38] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.