Automatic Bangla Image Captioning Based on Transformer Model in Deep Learning

Md. Anwar Hossain[1*], Mirza AFM Rashidul Hasan[2], Ebrahim Hossen[3],
Md Asraful[4], Md. Omar Faruk[5], AFM Zainul Abadin[6], Md. Suhag Ali[7]
Dept. of Information and Communication Engineering, Pabna University of Science and Technology, Pabna, Bangladesh [1,3,4,5,6]
Dept. of Information and Communication Engineering, University of Rajshahi, Rajshahi, Bangladesh[2]
Dept. of Software Engineering, Daffodil International University, Dhaka, Bangladesh[7]

Abstract—**Indeed, Image Captioning has become a crucial aspect of contemporary artificial intelligence because it has tackled two crucial parts of the AI field: Computer Vision and Natural Language Processing. Currently, Bangla stands as the seventh most widely spoken language globally. Due to this, image captioning has gained recognition for its significant research accomplishments. Many established datasets are found in English but no standard datasets in Bangla. For our research, we have used the BAN-Cap dataset which contains 8091 images with 40455 sentences. Many effective encoder-decoder and Visual Attention approaches are used for image captioning where CNN is utilized for the encoder and RNN is used for the decoder. However, we suggested a transformer-based image captioning model in this study with different pre-train image feature extraction models like Resnet50, InceptionV3, and VGG16 using the BAN-Cap dataset and find out its effective efficiency and accuracy based on many performances measured methods like BLEU, METEOR, ROUGE, CIDEr and also find out the drawbacks of others model.**

Keywords—Bangla image captioning; image processing; natural language processing; attention mechanism; transformer model

I. Introduction

For image captioning, humans first look at the image and detect the object. Each human brain has a huge local language vocabulary. After detecting the object, it finds out the vocabulary of this respective object and generates a description of this image. This process is easier for humans but it needs some steps for our machine. For machines, it integrates two fundamental components of artificial intelligence, namely Computer Vision (CV) [1] and NLP [2]. When it comes to computer vision, various pre-trained CNN models [3] are employed to extort image features. These appearances are subsequently fed through an RNN [4] to generate captions utilizing the LSTM mechanism [5]. These days, there are several uses for picture captioning, including self-driving cars, social media, security and surveillance, travel and tourism, healthcare, robotics, and many more. Nowadays, the seventh most used language worldwide is Bangla [6]. For the huge number of populations, it is recognizable for significant research.

However, this research can find out the lack of a present days' model where existing model RNN is performed for generating word of sequences and discover the absence of context awareness where the existing model is trained using tokenizer word format without any relative positioning and attention mechanism. In this paper we try to take the following objectives:

- Build up an image captioning sculpt lying on transformers.
- Attention on the context.
- Compare the suggested model performance with other popular images captioning model.

The proposed model's performance undergoes assessment through various Natural Language Processing evaluation methods that rely on both machine-generated captions and reference captions. These methods include BLEU [7], METEOR [8], ROUGE [9], and CIDEr [10].

II. Literature Review

We discuss several approaches and cutting-edge techniques used in Bangla picture captioning in this part. This section is separated into two sections: The visual attention-based method and the CNN-LSTM-based approach. Ultimately, we endeavour to identify the limitations of the prevailing models.

A. CNN-LSTM-based Approach

The initial Bangla image captioning model is constructed based on the CNN-LSTM architecture, as outlined by [11]. These are separated into two parts :1. Image features extraction 2. Language is generated based on the features. The image feature is extracted by the VGG16 model [12]. The model operates with two inputs: the image and the tokens' order(which represent unique words in the dictionary). It employs an embedding layer to derive the respective word embeddings from the tokens. The word embedding layer's output is subsequently compressed to 512 dimensions using a dense layer. This result is reciprocated to the sequence produced by the embedding layer as its output data that is stacked on the LSTM. The stacked LSTM sequence data generate the n-length caption. Here is a blocked diagram Fig. 1.

B. Visual Attention-based Approach

The visual attention-based approach [13] is described in three parts: 1. Image feature extracted by CNN [3] 2. Attention mechanism for getting weighted image features [14]

3. GRU [13] for generating the caption. Here is a block diagram Fig. 2 of this model.

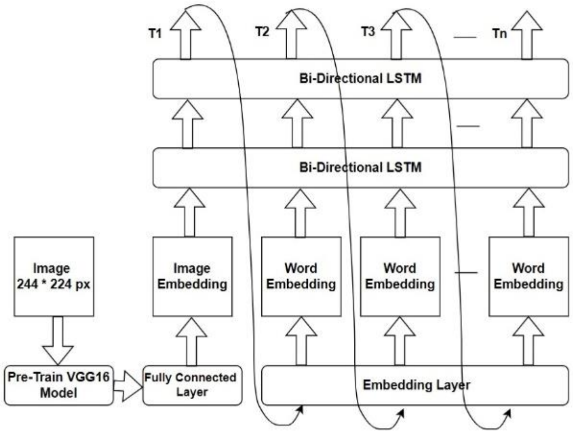


Fig. 1. Image captioning CNN-LSTM model [11].

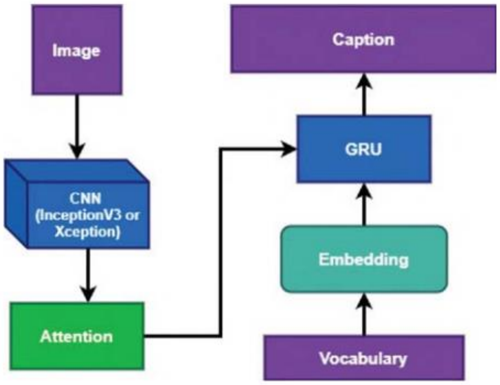


Fig. 2. Visual attention model for image captioning [13].

In this model image features are extracted by the CNN to focus on an important portion image. the image feature vector is directed to the attention mechanism, which generates a context vector specific to this image. In this attention mechanism there are three parts: 1. calculate the alignment score 2. calculate the weight 3. calculate the context vector. For the alignment score, the Eq. (1) is:

$$A_{t,i} = a(s_{t-1}, h_i) \quad (1)$$

Where $A_{t,\ i}$ is the alignment score, s_{t-1} is the previous decoder output, h_i is the encoded hidden state and a() is a function of attention. Calculating the weight is only the output of the softmax function.

$$w_{t,i} = softmax(A_{t,i}) \quad (2)$$

To calculate the context, vector the equation is as follows:

$$\mathrm{C}_t = \sum_i^T w_{t,i}\, \mathrm{h}_i \quad (3)$$

Where C_t is the context vector. In the Embedding section that creates the vector of words. The GRU [15] received inputs of the context vector along with the word vector from the Embedding layer, subsequently producing the caption.

After understanding on popular model, we can specify some major areas in which we can improve our Bangla image captioning model.

TABLE I. DRAWBACKS OF PREVIOUS MODEL

Model Name	Drawbacks
CNN-LSTM Model [11]	1. Limited Parallelization: Sequential execution to generate caption. 2. Context Understanding: It has no attention mechanism to keep the context of the image. 3. Complexity and Training Efficiency: Require more time to execute.
Visual Attention Model [13]	1. Position Tracking: No track on the position in caption words. 2. Sequential Computation: it has no capability of parallel processing. 3. Scalability and Generalization: GRU-based models may encounter difficulties in scaling to larger and more diverse datasets.

Using our proposed model Fig. 4 we solve the drawbacks which are indicated in Table I. In our proposed model where multi-head self-attention layer has multiple numbers of layers that help to parallel execution and the self-attention mechanism helps to focus on the contextual information in the image which eliminates the limitation of parallelization and context Understanding in the CNN-LSTM model approach. Secondly, In our proposed model (see Fig. 4) the positional embedding layer tracks the position of the input vector and the masked self-attention layer filters the key point of the Bangla caption sentences which eliminates the limitation of position tracking and scalability in the Visual attention model approach. In the word embedding layer that tokenizes the Bangla sentence and turns to convert the respective word vector which reduces the complexity of the model. All the layers are fully described in the methodology section.

III. METHODOLOGY

This section provides an impression of the operational construction of our intended model. That's are divided into two parts: Data Collection and Transformer model. In the data collection section, we describe the procedure of data collection and data pre-processing. In proposed transformer model section, we describe the model architecture.

A. Data Collection

We have used two data sets first for the image dataset Flickr8k [16] which contains 8091 images and second for the Bangla caption using the BAN-Cap dataset [17] which contains 40455 captions. Each image has five captions. Here is a Fig. 3 of the most frequent words in this dataset.

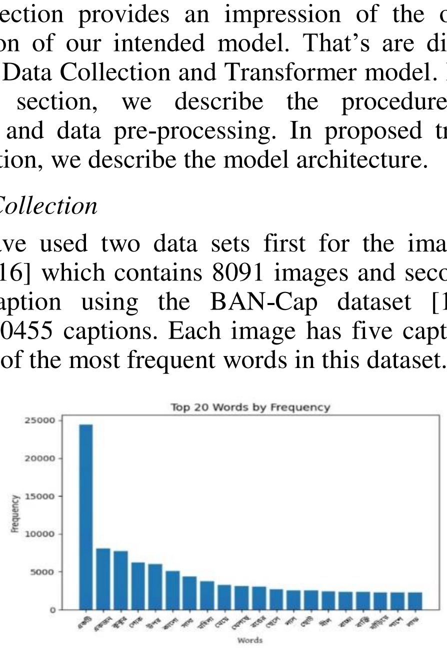


Fig. 3. Top most 20 frequently used word.

B. Transformer Model

The term "Transformers" was first used in the article "Attention is all you need." [18]. It serves as the foundation for some popular versions like GPT-2 [19] and BERT [20]. Language translation models and question-and-answer-based models are two examples of transformer models' many applications. Transformers can be utilized for a variety of application cases because of their versatile architecture**.** For Bangla image captioning, our proposed model, depicted in Fig. 4, is formulated based on the transformer model. We describe our proposed model into two parts: Encoder and Decoder.

C. Encoder

The encoder is tasked with handling the input data. The inputs may be a sequence of words in a natural language sentence, Image data, or any other sequential data. The Encoder is the main work of extracting meaningful representations from the input data that are known as "contextual embeddings". In this section, we describe several parts which are Image Feature Extraction, Positional Embedding, Multi-head self-attention, Add and normalization, and Feed Forward Layer.

1) Image feature extraction: Image feature extraction is a procedure that transforms raw data into a numerical form. Which helps us to identify and capture the relevant patterns of the image. For feature extraction we have used different pre-train models ResNet50[21], Inception V3 [22], and VGG16 models [12]. After the feature extraction, the features are represented with vectors that are shown in Fig. 4.

In our proposed model Fig. 4 we try to solve the drawbacks of the previous model. In this model first, image and captions sequential data are converted into vectors respectively patch embedding and word embedding. To track the position of this sequential data we add a positional encoder that solves the drawbacks of the previous model which are fully described below. After positional encoding the multi-head self-attention it handles every element of the input sequence in parallel, making them more suitable for parallelization during both training and inference. This capacity to handle data in parallel improves training effectiveness and lessens the vanishing gradient issue that helps the limitation of parallelization in the previous model. In the Word embedding layer, we used word tokenizers that easily tokenize the word and convert it into word vectors that store synthetic and semantic information about those words, and all the layers are described briefly in the below section.

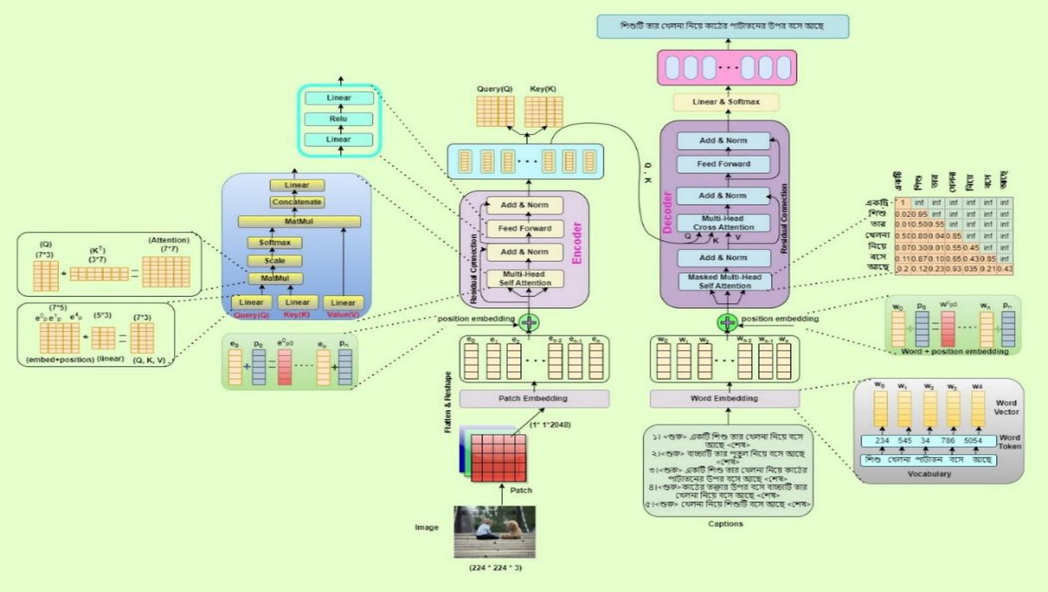

শিশুটি তার খেলনা নিয়ে কাঠের পাটাতনের উপর বসে আছে
Linear & Softmax
Query(Q)
Key(K)
Linear
Relu
Linear
Concatenate
MatMul
Softmax
Scale
MatMul
Linear
Linear
Linear
Query(Q)
Key(K)
Value(V)
(Q)
(7*3)
(K^T)
(3*7)
(Attention)
(7*7)
(7*5)
e⁰p e¹p e⁴p
(5*3)
(7*3)
(embed+position) (linear) (Q, K, V)
e0 p0 e0p0 en pn
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self Attention
Residual Connection
Encoder
position embedding
e0 e1 e2 en-2 en-1 en
Patch Embedding
Flatten & Reshape
(1* 1*2048)
Patch
Image
(224 * 224 * 3)
Q, K
Decoder
Add & Norm
Feed Forward
Add & Norm
Multi-Head Cross Attention
Q K V
Add & Norm
Masked Multi-Head Self Attention
Residual Connection
position embedding
w0 w1 w2 wn-2 wn-1 wn
Word Embedding
১। <শুরু> একটি শিশু তার খেলনা নিয়ে বসে আছে <শেষ>
২।<শুরু> বাচ্চাটি তার পুতুল নিয়ে বসে আছে <শেষ>
৩।<শুরু> একটি শিশু তার খেলনা নিয়ে কাঠের পাটাতনের উপর বসে আছে <শেষ>
৪।<শুরু> কাঠের তক্তার উপর বসে বাচ্চাটি তার খেলনা নিয়ে বসে আছে <শেষ>
৫।<শুরু> খেলনা নিয়ে শিশুটি বসে আছে <শেষ>
Captions
একটি শিশু তার খেলনা নিয়ে বসে আছে
একটি 1 inf inf inf inf inf inf
শিশু 0.02 0.95 inf inf inf inf inf
তার 0.01 0.50 0.55 inf inf inf inf
খেলনা 0.50 0.80 0.04 0.85 inf inf inf
নিয়ে 0.07 0.30 0.01 0.55 0.45 inf inf
বসে 0.11 0.87 0.10 0.65 0.43 0.85 inf
আছে 0.2 0.12 0.23 0.93 035 0.21 0.43
w0 p0 w⁰p0 wn pn
Word + position embedding
w0 w1 w2 w3 w4
Word Vector
234 545 34 786 5054
Word Token
শিশু খেলনা পাটাতন বসে আছে
Vocabulary

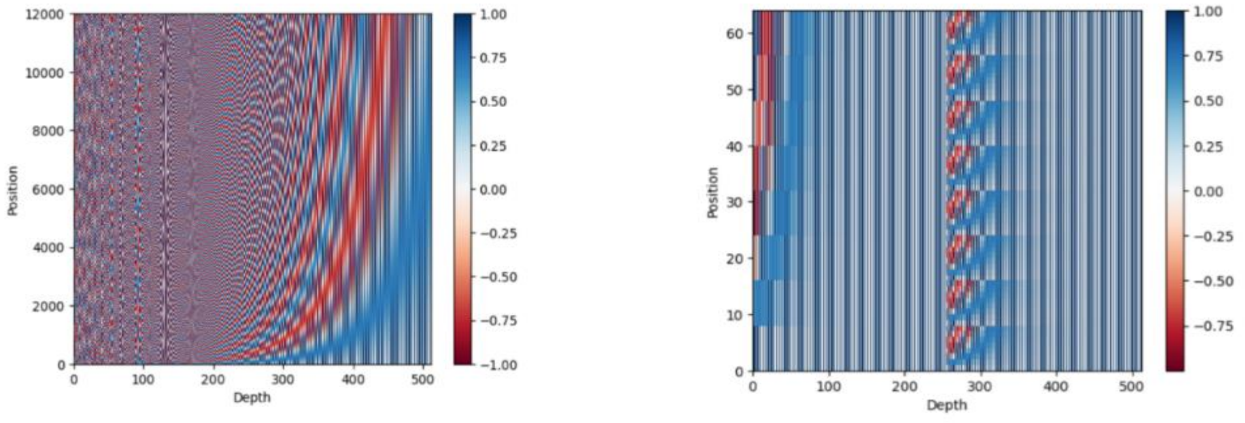
Fig. 4. Diagram of our proposed model.

2) Position embedding: Since ours is devoid of repetition and convolution, to ensure the model effectively utilizes the sequence order, it is vital to furnish precise information about the token positions within the sequence. To achieve this, we incorporate "position encodings" into the input embeddings of both the encoder and decoder layers. For better understanding here an example "রহিমের দুটি ঘর রয়েছে, সে দীর্ঘ দশ বছর ধরে রহিমার সাথে ঘর করছে" In this sentence the first word "ঘর" meaning the number of House but the second word "ঘর" meaning the Family. For this reason, we need to track the word that's called position embedding. In Fig. 4 we see how the position value is added with an image vector in the encoder and a word vector in the decoder to track them. There are a variety of methods of position encoding systems. We use the sine and cosine functions for position encoding. The sine function is used for odd positions that are:

$$P_{(pos,2k)} = sin\left(\frac{pos}{10000^{2k/d}}\right) \tag{4}$$

The cosine function is used for even positions that are a mathematical form:

$$P_{(pos,2k+1)} = cos\left(\frac{pos}{10000^{2k/d}}\right) \tag{5}$$



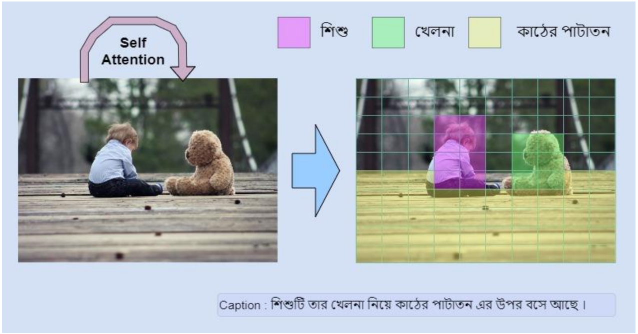
(a)

(b)

Fig. 5. (a) Caption position encoding (b) Image position encoding.

From Fig. 5 we show that the value of each position is represented between 1 to -1. On the graph, Position is shown by the Y axis and the associated position value by the X axis for both image and caption encoding.

3) Multi-head self attention: As the name suggests, Attention means to focus on the input data that are relatively close to the features and therefore establishing a relationship with them. Fig. 6 shows that the main three focusing points of this image are the baby, doll, and wood. There is a caption generated based on those points.



Caption : শিশুটি তার খেলনা নিয়ে কাঠের পাটাতন এর উপর বসে আছে।

Fig. 6. Self-Attention in an image.

From Fig. 4 we show the mechanism of the multi-head attention layer with a better example. Firstly, the Linear section receives the positional embedding's output. The linear layer functions as a fully connected neural network that performs multiplication with positional encoding, thereby generating a matrix. This matrix is duplicated into the Query, Key, and Value matrices.

- Query(Q) - Represents the whose value needs to be determined
- Key(K) - Represents the features
- Value(V) - Represents the actual value of the input

Secondly, the MatMul section creates an attention-value matrix that is shown in Fig. 4. The mathematical representation is:

$$A_{(Q,K)} = Q.k^T \tag{6}$$

Thirdly, to normalize the attention values within the attention matrix, each value is divided by $\sqrt{dk}$ where (dk) is the dimension of Key metrics. The mathematical representation is:

$$s_i = \frac{A_{(Q,k)}}{\sqrt{d_k}} \tag{7}$$

Fourthly, the S_i value passed into the softmax() and the output vector of the activation function is a matrix multiplied by the Value(V) matrix in the MatMul section and creates an attention-weight matrix. Which attention weight value is high which represents the focus point. Here the final mathematical equation is:

$$A_{(Q,K,V)} = softmax\left(\frac{A_{(Q,K)}}{\sqrt{d_k}}\right).V \tag{8}$$

This attention equation is for one head. For the name multi-head, the concatenate layer Fig. 4 is added to those heads. The mathematical equation is:

$$\begin{aligned} &MultiHead_{(Q,K,V)} \\ &= concat(head_1, head_2, head_3 \ldots\ldots head_n) \end{aligned} \tag{9}$$

n denotes the number of heads.

4) Add & Norm: This Layer does two activities, as its name indicates. The 'Add' portion of the process, which controls flow via residual connections, is the initial phase. 'Norm', the next step, accomplishes layer normalization.

5) Feed forward: This Layer incorporates a fully connected point-wise feed-forward network, employing the ReLU activation function to conduct two linear transformations, as illustrated in Fig. 4. This layer determines the weights used during exercise. The mathematical representation is:

$$FF(x) = ReLU(xw_1 + b_1)w_2 + b_2) \tag{10}$$

Weight matrices denote by W_1 and W_2, and bias denoted by b_1 and b_2, where the ReLU [23] function is :

$$ReLU(x) = max(0, x) \tag{11}$$

D. Decoder

For tasks like language translation, text generator, and text summarization, the decoder in a Transformer is in charge of producing an output sequence. It works in conjunction with the encoder, which analyzes the input sequence. In Fig. 4 Decoder is comprised of: 1. word embedding layer 2. Position embedding 3. Masked multi-head self-attention layer 4. Addition & Normalization 5. Multi-head cross attention 6. Feed Forward layer 7. Linear & SoftMax layer. In the previous Encoder section, we have already explained those layers which similar to both the encoder and decoder. So, in this section, we describe the below layer.

1) Word embedding: It is an essential component of the transformer concept. That is responsible for converting input tokens of words and generating a word vector. This vector records the word's semantic and grammatical details thereby assisting this model in acquiring a meaningful representation of the input text. Fig. 4 shows an example of Bangla words converted into word vectors by tokenization which are detonated by $w_{0.......}w_n$.

2) Masked multi-head self-attention: This level is essential in the Transformer model because it prevents the model from attending to future locations and maintains the autoregressive characteristic while enabling it to focus on different input sequence segments. After receiving the previous decoder output stack, the first sublayer adds positional information to it and applies self-attention to it. Decoders are altered to focus exclusively on the words that come before them, whilst the encoder is made to pay attention to every word respective to the input sequence of where it appears in the sequence. Consequently, the forecast for a word at a specific position in the sequence can only rely on the known outputs for the preceding words. This is accomplished in the multi-head attention mechanism. Through the application of a mask to the outcomes of the scaled matrix multiplication, the values that would otherwise correspond to prohibited values are suppressed to achieve this masking. Fig. 4 gives a pattern of a mask filter on the words. In Fig. 4 infinity(inf) means that has no probability with the next words and maximum value means the high probability of the next word. Here is a simple representation of the mask filter.

$$mask(Q, K^T) = mask\left(\begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{k1} & w_{k2} & \dots & w_{kn} \end{bmatrix}\right) =$$

$$\begin{bmatrix} w_{11} & inf & \dots & inf \\ w_{21} & w_{22} & \dots & inf \\ \dots & \dots & \dots & \dots \\ w_{k1} & w_{k2} & \dots & w_{kn} \end{bmatrix} \tag{12}$$

E. Model Parameters

We trained our proposed model with different hyper-parameter value that's are indicated on Table II. Those internal variables are adjusted in our model during the training process to minimize the error between predicted and target captions. Those model parameters perfectly capture the features, relation, and pattern of the data.

TABLE II. EXPERIMENTAL PARAMETERS

Parameters name	Value
Vocabulary size	12000
Batch size	64
Buffer size	1000
Dropout rate	0.001
Number of Layer	8
Dimension of model	512
Number of head	8
Number of Epoch	40
Maximum length of sentence	25

IV. RESULT AND DISCUSSION

This part describes the functionality and visualization of the model we've suggested, and then we compare it to other models. Here in Table III, we show our proposed model's performance with different pre-train CNN models. The performance is measured by BLEU [7], METEOR [8], ROUGE [9], and CIDEr [10]. In Table III, the ResNet50+Proposed model gives the maximum output. Ok, Table IV compares our model to several model methodologies.

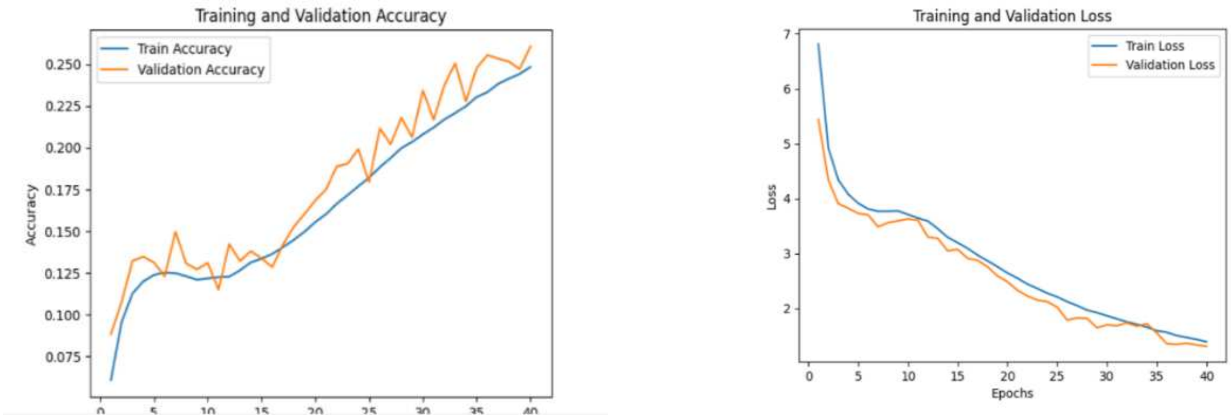
Now, here below in Fig. 7, 8, and 9 show the accuracy and loss curve of our model. The validation curve is represented by a yellow color line and the training curve is represented by a blue color line. The learning curve is the identifier of model overfitting and underfitting evaluation. From Those figures, we see that the accuracy curve of our model is closely together and gradually increasing. Other side, the loss curve is closely together and gradually decreasing. For this, the model is free from overfitting and underfitting issues and called it is called a good fit model. We hope that our model are more balanced fit for large datasets like Flick30k [25].

TABLE III. PERFORMANCE TABLE FOR OUR PROPOSED MODEL

Model	Dataset	BLEU1	BLEU2	BLEU3	BLEU 4	METEOR	ROUGE	CIDEr
Resnet50+ proposed model	Flick8k+ BAN-Cap_captiondata	**64.38**	**58.58**	**43.40**	**24.30**	**0.31**	**0.39**	**0.28**
InceptionV3+ proposed model	Flick8k+ BAN-Cap_captiondata	61.01	56.22	41.03	23.93	0.31	0.41	0.29
VGG16+proposed model	Flick8k+ BAN-Cap_captiondata	60.38	55.44	39.94	21.42	0.29	0.38	0.26

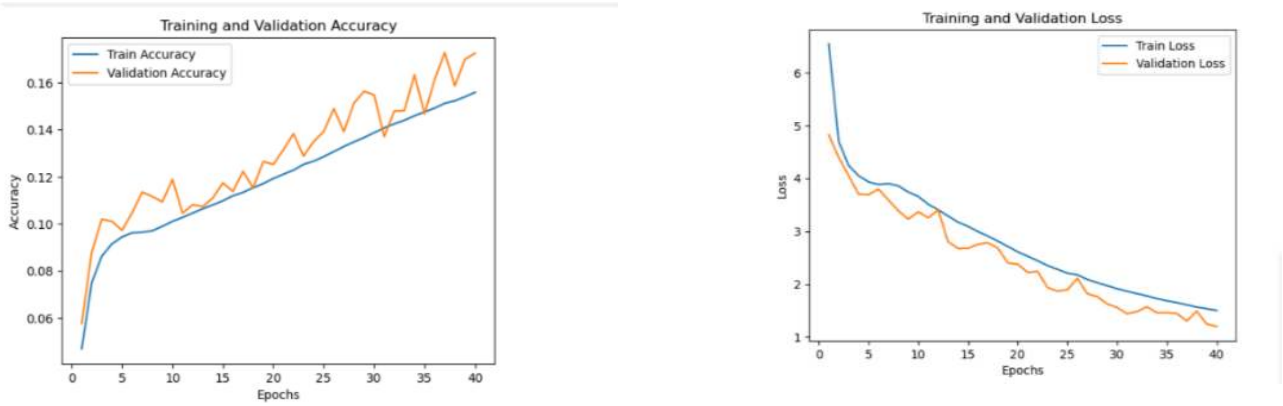
TABLE IV. COMPARE PERFORMANCE WITH OTHER MODEL

Model name	Dataset	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE	CIDEr
CNN-Merge based [24]	Flick8k+ BAN-Cap_captiondata	56.5	35.5	22.1	13.1	0.281	0.290	0.178
Visual-Attention based [13]	Flick8k+ BAN-Cap_captiondata	58.7	36.8	25.4	14.4	0.293	0.288	0.199
Resnet50+ proposed model	Flick8k+ BAN-Cap_captiondata	**64.38**	**58.58**	**43.40**	**24.30**	**0.31**	**0.39**	**0.28**
InceptionV3+ proposed model	Flick8k+ BAN-Cap_captiondata	**61.01**	**56.22**	**41.03**	**23.93**	**0.31**	**0.41**	**0.29**
VGG16+ proposed model	Flick8k+ BAN-Cap_captiondata	**60.38**	**55.44**	**39.94**	**21.42**	**0.29**	**0.38**	**0.26**



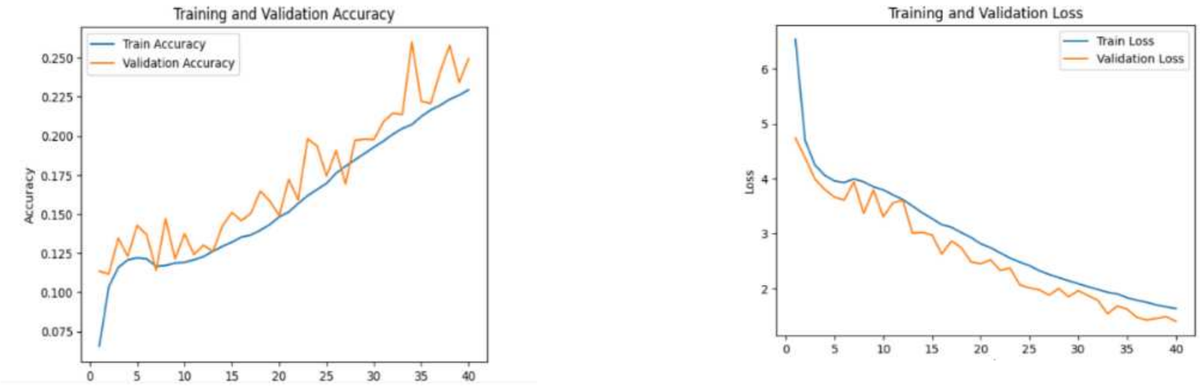
(a) (b)

Fig. 7. (a) Accuracy scheme and (b) Loss scheme for ResNet50+Proposed-model.



(a) (b)

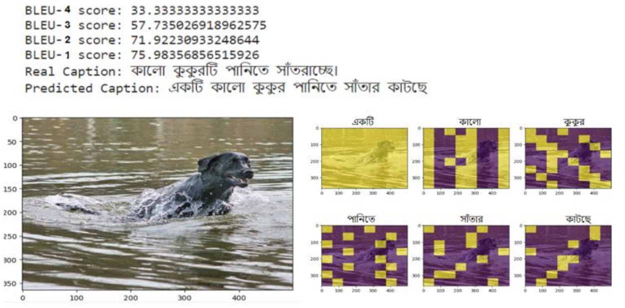
Fig. 8. (a) Accuracy scheme (b) Loss scheme for Inception V3+Proposed-model.



(a) (b)

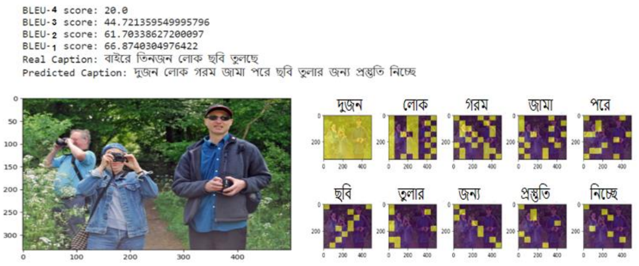
Fig. 9. (a) Accuracy scheme and (b) Loss scheme for VGG16+Proposed-model.

Now, below Fig. 10, Fig. 11, and Fig. 12 are shown some examples of our model prediction with attention mechanism.



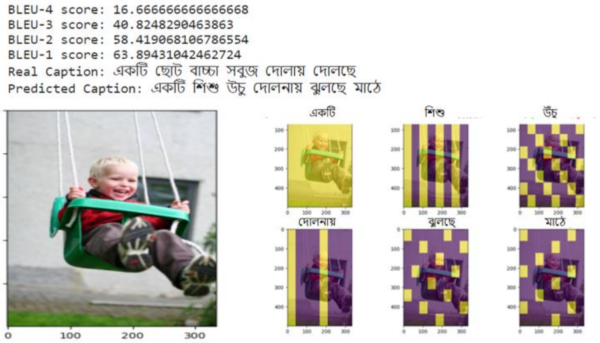
```
BLEU-4 score: 33.33333333333333
BLEU-3 score: 57.735026918962575
BLEU-2 score: 71.92230933248644
BLEU-1 score: 75.98356856515926
Real Caption: কালো কুকুরটি পানিতে সাঁতরাচ্ছে।
Predicted Caption: একটি কালো কুকুর পানিতে সাঁতার কাটছে
```

Fig. 10: Performance ResNet50+ Proposed model with Attention Plot



```
BLEU-4 score: 20.0
BLEU-3 score: 44.721359549995796
BLEU-2 score: 61.70338627200097
BLEU-1 score: 66.8740304976422
Real Caption: বাইরে তিনজন লোক ছবি তুলছে
Predicted Caption: দুজন লোক গরম জামা পরে ছবি তুলার জন্য প্রস্তুতি নিচ্ছে
```

Fig. 11: Performance Inception V3+Proposed model with Attention Plot



```
BLEU-4 score: 16.666666666666668
BLEU-3 score: 40.8248290463863
BLEU-2 score: 58.419068106786554
BLEU-1 score: 63.89431042462724
Real Caption: একটি ছোট বাচ্চা সবুজ দোলায় দোলছে
Predicted Caption: একটি শিশু উঁচু দোলনায় ঝুলছে মাঠে
```

Fig. 12: Performance VGG16+ Proposed model with Attention Plot

V. Conclusions and Future Work

This study presents a transformer-based paradigm for captioning images in Bangla. This proposed model is turned up with better efficiency and performance based on different evaluation methods. It also proves that this model is a good fit for Bangla image captioning on the Flick8k+BAN_Cap dataset. we expect that this paper will help to encourage to others develop more efficient transformer-based models in different NLP and Computer Vision tasks. We also expected that it would be helpful for Image Captioning on higher datasets like Flick30k and others datasets and in the future, we also try to do this.

Acknowledgment

We appreciate the Department of Information and Communication Engineering at Pabna University of Science and Technology for supporting this investigation.

References

[1] Fang, W., Ding, L., Love, P. E., Luo, H., Li, H., Pena-Mora, F., Zhong, B., & Zhou, C. "Computer vision applications in construction safety assurance," Automation in Construction, 110, 103013. 2020, https://doi.org/10.1016/j.autcon.2019.103013.

[2] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I., "Improving language understanding by generative pre-training," 2018,

[3] Asraful, M., Hossain, M. A., & Hossen, E. "Handwritten Bengali Alphabets, Compound Characters and Numerals Recognition Using CNN-based Approach," Annals of Emerging Technologies in Computing (AETiC), 7(3), 60–77. 2023, https://doi.org/10.33166/aetic.2023.03.003.

[4] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. "Learning representations by back-propagating errors," nature, 323(6088), 533-536, 1986, https://doi.org/10.1038/323533a0.

[5] Hochreiter, S., & Schmidhuber, J. "Long short-term memory. Neural computation," 9(8), 1735-1780. 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[6] Szmigiera, M. "Most spoken languages in the world," Statista. Retrieved Oct. 01, 2023 from https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/.

[7] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. "Bleu: a method for automatic evaluation of machine translation," Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

[8] Denkowski, M., & Lavie, A. "Meteor universal: Language specific translation evaluation for any target language," Proceedings of the ninth workshop on statistical machine translation. 2014.

[9] Lin, C.-Y. "Rouge: A package for automatic evaluation of summaries," Text summarization branches out. 2004.

[10] Vedantam, R., Lawrence Zitnick, C., & Parikh, D. "Cider: Consensus-based image description evaluation," Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[11] Rahman, M., Mohammed, N., Mansoor, N., & Momen, S. "Chittron: An automatic bangla image captioning system," Procedia Computer Science, 154, 636-642, 2019. https://doi.org/10.1016/j.procs.2019.06.100.

[12] Sutskever, I., Vinyals, O., & Le, Q. V. "Sequence to sequence learning with neural networks," Advances in neural information processing systems, 27, 2014.

[13] Ami, A. S., Humaira, M., Jim, M. A. R. K., Paul, S., & Shah, F. M. "Bengali image captioning with visual attention," 2020 23rd International Conference on Computer and Information Technology (ICCIT).

[14] Bahdanau, D., Cho, K., & Bengio, Y. "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473. 2014. https://doi.org/10.48550/arxiv.1409.0473.

[15] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555. 2014. https://doi.org/ 10.48550/arXiv.1412.3555.

[16] Hodosh, M., Young, P., & Hockenmaier, J. "Framing image description as a ranking task: Data, models and evaluation metrics," Journal of Artificial Intelligence Research, 47, 853-899. 2013. https://doi.org/10.1613/jair.3994.

[17] Khan, M. F., Shifath, S., & Islam, M. S. "BAN-cap: a multi-purpose English-Bangla image descriptions dataset," arXiv preprint arXiv:2205.14462.2022. https://doi.org/10.48550/arxiv.2205.14462.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. "Attention is all you need," Advances in neural information processing systems, 30. 2017.

[19] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. “Language models are unsupervised multitask learners,” OpenAI blog, 1(8), 9, 2019.

[20] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805. 2018 https://doi.org/10.48550/arXiv.1810.04805.

[21] He, K., Zhang, X., Ren, S., & Sun, J. "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition. 2016

[22] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. "Rethinking the inception architecture for computer vision," Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[23] LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. "Efficient backprop," In Neural networks: Tricks of the trade (pp. 9-50). 2002. Springer. https://doi.org/10.1007/3-540-49430-8_2.

[24] Faiyaz Khan, M., Sadiq-Ur-Rahman, S., & Saiful Islam, M. "Improved bengali image captioning via deep convolutional neural network based encoder-decoder model," Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020.

[25] Young, P., Lai, A., Hodosh, M., & Hockenmaier, J. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," Transactions of the Association for Computational Linguistics, 2, 67-78. 2014 https://doi.org/10.1162/tacl_a_00166.