Comparative Study of Different Image Captioning Models

Sahil Takkar
Department of Computer Science and Engineering
Delhi Technological University
Delhi, India
Email: sahiltakkar4@gmail.com

Anshul Jain
Department of Computer Science and Engineering
Delhi Technological University
Delhi, India
Email: jainanshul3863@gmail.com

Piyush Adlakha
Department of Computer Science and Engineering
Delhi Technological University
Delhi, India
Email: piyushadlakha204@gmail.com

2021 5th International Conference on Computing Methodologies and Communication (ICCMC) | 978-1-6654-0360-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCMC51019.2021.9418451

Abstract—**This paper has compared various deep learning models for generating caption of images gathered from Flickr 8k Dataset. Also, this research work attempts to combine a CNN type encoder for extracting features from images and a Recurrent Neural Network for generating caption for the extracted features. The CNN encoders used are VGG16 and InceptionV3. The extracted features are then passed to a unidirectional or a bidirectional LSTM for generating captions. The proposed model has used beam search as well as greedy algorithms to generate captions from vocabulary. The generated captions are then compared with actual captions with the help of BLEU scores. The Bilingual Evaluation Understudy score (BLEU) is used to compare how close a given sentence is to another sentence. The BLEU score of captions generated using beam search as well as greedy algorithms are analyzed and compared to see which is better.**

Keywords— VGG16, InceptionV3, Bidirectional LSTM, BLEU, Beam Search

I. INTRODUCTION

Nowadays, Artificial Intelligence (AI) is a very important part of the innovation sector and hence the basis of our project is also Machine Learning and AI. In the recent history, the sector of Deep Learning[1] has impressed everybody when compared to already famous currently present Machine Learning(ML) methodologies like Decision Trees, Logistic Regression, SVM, Naive Bayes, K Nearest Neighbors, Random Forest, etc. because of its extraordinary results in terms of accuracy as compared to that of already present traditional Machine learning models like KNN, Logistic Regression etc. It is a difficult task to generate a relevant description for an image but once done it can prove to be a great benefit to society. This can help visually impaired people to have better understanding of their surroundings by generating a suitable caption for an environment. It has many other applications like usage in virtual assistants, recommendations in editing applications, for social media etc.

Generating a caption[2] from an image is a notably harder task as compared to that of classifying an image, which has been the centre of attraction for the computer vision community. A description for an image must take into account the relationship between different objects presents in the image. Along with the visual description of the objects in image, the knowledge mentioned above has to be stated in a natural language understandable by humans. It means that, a language model is required, where it not only understands the image but also expresses it in a natural language. The attempts made in the past have all been to use two different models, one for understanding the image[3] and another for using that understanding to generate a caption and then stitching the two models together.

The proposed method has attempted to combine the two models into one combined model, which consists of a CNN[4] type encoder that aids us in extracting features of image by creating encodings of images. Here, the pre-trained VGG16 and Inception V3 architecture model is used for encoding images. The CNN encoders extract features from the image and store them in the form of numerical encodings which can be easily understood by the machine. These extracted features are then passed to a type of Recurrent Neural Network namely LSTM network. The network architecture of the LSTM network works in almost the same way as that used in natural language machine translators. LSTM is replaced with bidirectional LSTM in order to see which one works better for prediction captions from extracted features.

The proposed project uses the Flickr 8k set of data which consists of eight thousand (8000) images and for each and every image, there are 5 captions respectively. By default, the dataset is splitted into two folders, image folder and text folder. For each image the caption is stored along with the respective ID as there is a distinctive image-id for every image in the set. The images in the dataset are divided into three parts: Training set, development set and Test set. Test and development set consist of 1000 images each whereas the training set consists of 6000 images. The model predicts a caption based on the vocabulary it creates from the tokens of words that it gathers from descriptions of images gathered from the training dataset. The description predicted by our model is then compared with the actual description provided in the dataset via BLEU score.

The upcoming sections of the paper will briefly discuss about the tools, techniques and dataset. Also, this research work attempts to discuss the CNN encoder namely VGG16 and Inception-v3 and the RNN[5] decoders namely LSTM and Bidirectional-LSTM decoder in full length. Also, this research work discusses about algorithm for caption generation, which has used to generate the predicted captions, namely argmax and Beam search. The BLEU score metric is used for comparing the accuracy of the different image captioning models being proposed. BLEU score helps in analyzing the text quality which has been generated by the ML model. BLEU score was among the earliest developed metrics to get such high correlation with actual verdict. The value of BLEU score always lies between 0-1. If BLEU score is zero, it means machine translation is not relevant to actual description at all. On the other hand, a BLEU score of 1 means that the machine translation is equivalent to actual description. BLEU score has also been discussed in detail in coming sections. At the end this paper includes the examples of some images, which have been used to test the proposed model.

II. Related Work

Caption Recommender System is an integral part of understanding the environment, which has various applications (e.g. - subtitle generation, helping visually impaired people to understand their surroundings, storytelling from albums, search using image, etc.). Since many years, many different image caption recommendation approaches have been developed.

There have been a lot of contributions from the architecture created by the winner of the ILSVRC. Along with the VGG the research made in the field of natural language translation have helped us continuously in bettering the performance in text generation.

Researchers at AI Lab used a Convolution Neural Network for each potential object in the image for producing high-level features of the image. Then a Multiple Instance Learning (MIL) [6] was used for figuring out the best area which matches with each word. This method gave a BLEU score of 22.9% on MS-COCO dataset.

The Vinyals came up with a new model called NIC (Neural Image Caption), Show and Tell model [7], which was nothing but an encoder RNN which was given input through a CNN model for computer vision. After this a group of researchers took the NIC model and modified it. They used a technique that makes use of images datasets and their corresponding captions to study the inter-modal correlations between natural language and image data. The model used by those researchers was based on a new combination of CNN around image fields, the LSTM or bidirectional RNN over textual descriptions, and a planned aim of putting the two modals together via bimodal embedding. Flickr 30K, Flickr 8K and MSCOCO were the datasets used by them to achieve these bests in business results. Jonathan further modified their model in 2015 when he suggested an idea of a model related to dense captioning in which the model detects each of the different areas of the image and then suggests a group of captions. Chen Wang also suggested a model which makes use of multiple LSTM networks and a deep CNN in the year 2016.

Over a period of time there has been enhancements not only in the captioning models but also in score metrics used for evaluating the accuracy of the models. This project has used the BLEU score for evaluation. BLEU - being a standard evaluation metric adopted by many of the groups. Now, new state of the arts metrics has come like CIDEr which are replacing older metrics like BLEU score, etc. CIDEr was proposed by Vedantam [8].

III. Approach

Recent developments in the field of technologies related to image captioning has been the main source of motivation for our research work. The model proposed in this paper has an eventual aim as to predict natural language descriptions for various areas of the image.

The research work focuses mainly on obtaining the results for several image captioning models by making use of BLEU score metric and hence comparing the performance of different image captioning models. Various CNN models such as VGG16, Inception-V3 etc are used for encoding the images and extracting features from the images. Further these encoded images are used with two types of decoders, namely unidirectional LSTM and bidirectional LSTM to obtain the results. We have used greedy search and beam search algorithm to generate the caption from encoded features. The generated caption is then compared with the original caption from dataset on the basis of Bilingual Evaluation Understudy score.

A. Convolution Models : Encoders

This section discusses various convolution models used for the research work. There are two encoders namely, VGG16 and Inception-V3. Each convolution model has been described in brief in the following subsections.

VGG16: VGG16[9] consists of a 16-layer network for the completion of the task of encoding the image. Out of 16 layers present in the VGG16 network, 3 are dense layers and rest 16 are convolution layers. The architecture of VGG16 is shown in Fig. 1. For the feature extraction to be done on the image, the dimension of the image has to be a 224*224 image. We have fixed the length of the stride to be 1 for the CNN layer which have filters of size 3*3. The next step is Max pooling, it is executed using a window size of 2*2-pixel with a length of stride taken to be 2.

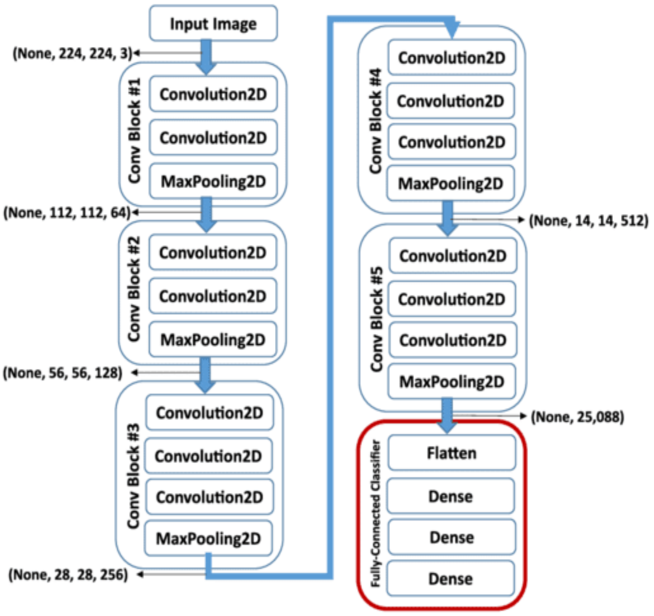

Fig. 1. VGG16 Architecture

Inception-V3: InceptionV3[10] consists of a 48-layer deep convolutional network for performing the task of encoding the image. InceptionV3 stacks together 11 inception modules each of which consists of convolution and max-pooling layers. For the feature extraction to be done on the image, the dimension of the image has to be a 229*229 image. Three fully connected layers of size 512, 1024 and 3 are added to the final concatenation layer. The architecture of Inception-V3 is shown in Fig. 2.

B. Decoders

This section discusses various decoder models used for the generation of captions for images. There are two decoders used in this research work namely, unidirectional

LSTM and Bidirectional LSTM. Each type of LSTM network has been described in brief in the following subsections.

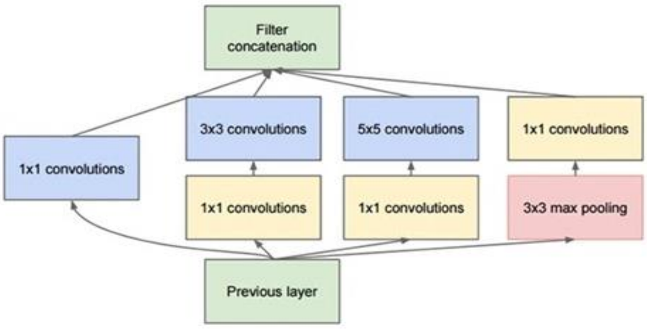

Fig. 2. Inception-V3 Architecture

LSTM (Long Short-Term Memory): LSTM[11] have been widely used by the researchers in the areas of text translation, audio to text conversion etc. As in the traditional RNNs, the straight structures are also present in LSTM, but there is a difference in the building manner of the reiterating modules. The main method by which LSTM preserves the past info is by line running on the top of LSTM network which is called as cell states. All of the modules in the network consist of a cell state. These cell states are fed information with the help of different gates. Fig. 3 shows four contacting layers of our LSTM model.

These gates are composed up of sigmoid function -whose value varies between 0 and 1- so it can be decided how much information is to be passed to the next layer. If the value of the sigmoid function is 1, it means the whole of the information is passed to the next cell else if it is 0 then no information is passed. Hence the cell states help the network to maintain the info in the system.

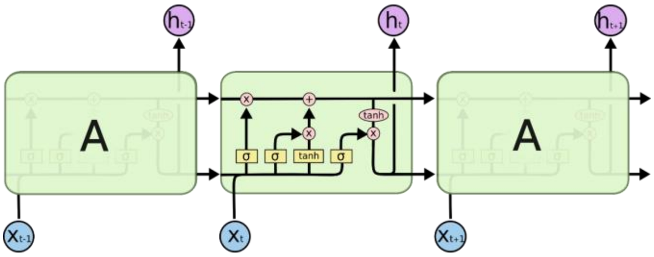

Fig. 3. Four contracting layers of LSTM

Bidirectional LSTM: Bidirectional LSTM[12] are an addendum to the conventional LSTMs and can help in significantly enhancing the performance of the model problems related to sequence classification. A Bidirectional LSTM, or bi-LSTM, is a model for sequence processing that consists of 2 LSTMs: one taking the input in a forward direction, and the other in a backwards direction. Bidirectional LSTMs work upon 2 LSTMs in place of one on the sequence provided as input. Fig. 4 shows the architecture of our bi-LSTM network. The first LSTM trains itself on the input sequence as-is and the second LSTM works upon the reversed copy of the input sequence. By using the bi-LSTMs the amount of information available to the network is increased effectively, which helps in enhancing the context available to the algorithm and thus result in complete and faster learning of the model.

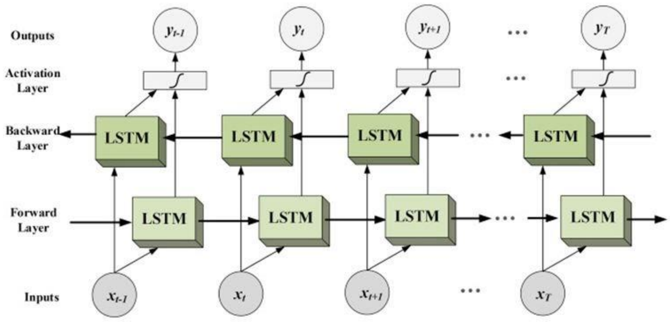

Fig. 4. Bi-Directional LSTM network

C. Dataset Collection

We have used the Flickr 8k[13] dataset for training and validation purposes. This dataset has been provided by University of Illinois at Arbana - Champaign. The dataset contains 8000 images and for each image it has corresponding 5 descriptions. By default, the dataset is split into two folders, image folder and text folder. For each image the caption is stored along with the respective id as we have a distinctive image-id for every image in the set. The images in the dataset are divided into three parts: Training set, validation set and Test set. Test and validation set consist of 1000 images each whereas the training set consists of 6000 images.

Apart from this there are other datasets also available like MS-COCO[14] and Flickr30k[15] for captioning images but both these datasets have at least 30,000 images and training the model on these datasets requires a lot of power and is computationally very expensive.

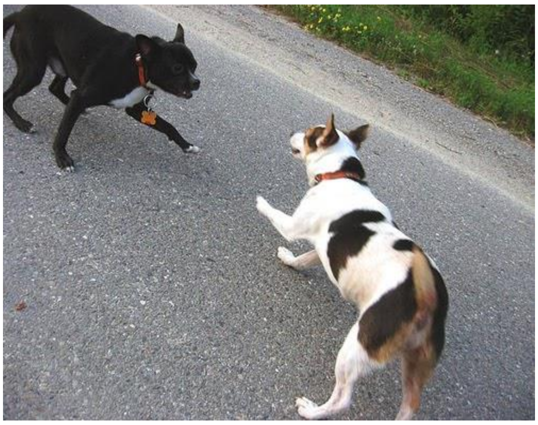

Fig. 5. A random image from dataset along with the following captions.

1. black dog and spotted dog are fighting
2. black dog and tri coloured dog playing on the road
3. two dogs of different breeds looking at each other on the road
4. two dogs on pavement moving toward each other
5. black dog and white dog with brown spots are staring at each other in the street

D. Data Preprocessing

Flickr 8k dataset consists of nearly 6000 train images and for each image we have corresponding 5 descriptions. These text descriptions require some minimal pre-processing before we can use it to train the model.

We first loaded the file containing all the descriptions along with their corresponding image id. We looped through the file and created a dictionary which maps each photo identifier to a list containing textual descriptions for the image. After this we did some cleaning of the textual data in order to reduce vocabulary size. Cleaning of textual descriptions involve: removing punctuations, converting text to lowercase, removing stop words like 'a', 'an' etc. and removing tokens containing digits.

Next step is to create a vocabulary of all the unique words present across all the image descriptions. Finally, for each description which corresponds to an image in training dataset we need to add a '<startseq>' token at the start of each caption and an <endseq> token at the end of each caption. The <startseq> token signifies the start of a sequence while <endseq> token signifies end of a sequence.

E. Feature Extraction

In our research work, image acts as an input to the decoder network. For training the decoder, the image data must be provided in the form of fixed size vectors. Therefore, each image is converted into a fixed size vector which will then be fed as input to RNN.

We use a transfer learning method for extracting features from the images. For this purpose, we used pre-trained models and its weight trained on larger similar data. We computed the image features using these pre-trained models and saved them in a file. Later we loaded these features and fed them into the neural network as the interpretation of the image given in the dataset.

F. Model Training and Evaluation

For training purposes, we used the Google colaboratory notebook. We trained the decoder model on a batch size of 32 and 64 using Adam optimizer and categorical_crossentropy as loss function. We used training and validation loss as the metric to evaluate the model after each epoch. We monitored the validation loss of the model during training. When the validation loss of the model improves at the end of an epoch, we saved the model into a file.

At the end of the training period, we used the model with best skill on the training dataset as our final model. The final code for our research work is available at [16].

G. Performace Measures

We have used two algorithms to generate the captions from the features extracted using CNN encoders.

Greedy Search Algorithm using Argmax function: Greedy Search algorithm chooses one best candidate at each step while generating caption. It selects the word with the highest probability by applying the argmax function to the vocabulary of words and selecting the word with the highest probability to generate captions of image. Choosing one best candidate may be optimal in beginning but for complete sentences, it may not be the best choice.

Beam Search algorithm: Beam search[17], [18] algorithm is a greedy tree search algorithm based on heuristics. The advantage over greedy search algorithm is that it selects multiple alternatives at each step instead of one. It selects the top k words with the highest probability from vocabulary of words, where k = beam width. The procedure for beam search is as followed:

1. Select the first k words with the highest probability from the vocabulary of words by applying SoftMax function.
2. For each word selected in the first step, find the conditional probability of the next word given that the previous pair of words occurred.
3. Repeat the process iteratively until the end of sentence. In simple words, at each step we consider the possibility of a pair of words occurring together instead of just focusing on a single word each time while generating a caption.

The number of alternatives selected at each step can be changed with beam width parameter, k. For example, if k=3, three alternatives are selected at each step of beam search.

BLEU Score:

After generating captions from extracted features, the next step is to compare the accuracy of our generated captions with the actual captions given in the dataset. We have used BLEU[19] score metrics as a parameter to measure the accuracy of our generated captions. BLEU score helps in analyzing the text quality which has been generated by the Machine Learning model. BLEU score was among the earliest developed metrics to get such high correlation with actual verdict.

The value of BLEU score always lies between 0-1. If BLEU score is zero, it means machine translation is not relevant to actual description at all. On the other hand, a BLEU score of 1 means that the machine translation is equivalent to actual description.

For calculating BLEU score we followed the following procedure:

1) Produced captions by taking all images which belong to the test set.
2) After that we used these captions generated by model as our predicted or candidate sentences.
3) Next each of the candidate sentences is associated with 5 of the reference sentences which are given by humans.
4) The BLEU score of candidate sentences related to each of the references is averaged.

IV. RESULT

The BLEU scores of different models are shown in table I and table III respectively. The y-axis shows the models along with their configurations while the x-axis shows the BLEU scores using the greedy algorithm and beam-search algorithm. From table I, we can infer that given a defined set of configurations (batch size = 64 and optimizer = adam and decoder = unidirectional LSTM), Inception V3 (BLEU-I score = 0.605097) performs better than Vgg16 model (BLEU-1 score = 0.578993). From table III, we can infer that given a defined set of configurations, a bidirectional LSTM decoder outperforms a unidirectional LSTM decoder for both Inception V3 and VGG16 encoders. One can also see that beam search algorithm for generating captions is better than greedy algorithm although the time required for beam search is more. The Inception V3 + Unidirectional LSTM model gives a BLEU-1 score of 0.5695 with batch size =32. But on increasing batch size to 64, the BLEU score improved to 0.605097. This took roughly 8GB of ram. We could not increase batch size to 64 with bidirectional LSTM models as that required more than 12GB of ram which is more than what is available with us. Table II shows an example of caption predicted on an image taken randomly from internet.

TABLE I. BLEU SCORES KEEPING BATCH SIZE = 64

Model and Config	Argmax (Greedy)	BEAM Search
InceptionV3 + Unidirectional LSTM **Epochs = 11** **Batch Size = 64** **Optimizer = Adam**	Cross-entropy loss (Lower the better) loss(train_loss): 2.5254 val_loss: 3.1769 BLEU Scores on Validation data (Higher the better) BLEU-1: 0.591272 BLEU-2: 0.340125 BLEU-3: 0.236282 BLEU-4: 0.105637	k = 3 (beam width) BLEU Scores on Validation data (Higher the better) BLEU-1: 0.601173 BLEU-2: 0.349092 BLEU-3: 0.248659 BLEU-4: 0.119507
VGG16 + Unidirectional LSTM **Epochs = 7** **Batch Size = 64** **Optimizer = Adam**	Cross-entropy loss (Lower the better) loss(train_loss): 2.6297 val_loss: 3.3486 BLEU Scores on Validation data (Higher the better) BLEU-1: 0.557626 BLEU-2: 0.317652 BLEU-3: 0.216636 BLEU-4: 0.105288	k = 3 (beam width) BLEU Scores on Validation data (Higher the better) BLEU-1: 0.578993 BLEU-2: 0.326569 BLEU-3: 0.226629 BLEU-4: 0.113102

TABLE II. AN EXAMPLE OF AN IMAGE TAKEN RANDOMLY FROM THE INTERNET.

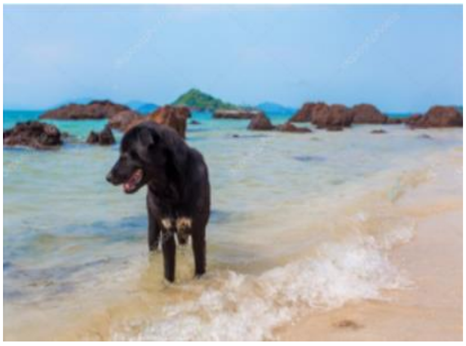	Predicted Caption: dog is running through the water

TABLE III. BLEU SCORES KEEPING BATCH SIZE = 32

Model and Config	Argmax (Greedy)	BEAM Search
InceptionV3 + Unidirectional LSTM **Epochs = 11** **Batch Size = 32** **Optimizer = Adam**	Cross-entropy loss (Lower the better) loss(train_loss): 2.5254 val_loss: 3.1769 BLEU Scores on Validation data (Higher the better) BLEU-1: 0.564183 BLEU-2: 0.314968 BLEU-3: 0.210921 BLEU-4: 0.098583	k = 3 (beam width) BLEU Scores on Validation data (Higher the better) BLEU-1: 0.569564 BLEU-2: 0.315819 BLEU-3: 0.219372 BLEU-4: 0.111061
InceptionV3 + Bidirectional LSTM **Epochs = 20** **Batch Size = 32** **Optimizer = Adam**	Cross Entropy loss (Lower the better) loss(train_loss): 2.4200 val_loss: 3.0724 BLEU Scores on Validation data (Higher the better) BLEU-1: 0.575166 BLEU-2: 0.332099 BLEU-3: 0.228444 BLEU-4: 0.111307	k = 3 (beam width) BLEU Scores on Validation data (Higher the better) BLEU-1: 0.581609 BLEU-2: 0.339489 BLEU-3: 0.240200 BLEU-4: 0.124673
VGG16 + Unidirectional LSTM **Epochs = 7** **Batch Size = 32** **Optimizer = Adam**	Cross-entropy loss (Lower the better) loss(train_loss): 2.6297 val_loss: 3.3486 BLEU Scores on Validation data (Higher the better) BLEU-1: 0.560285 BLEU-2: 0.308491 BLEU-3: 0.210819 BLEU-4: 0.105209	k = 3 (beam width) BLEU Scores on Validation data (Higher the better) BLEU-1: 0.566529 BLEU-2: 0.315291 BLEU-3: 0.212491 BLEU-4: 0.103105
VGG16 + Bidirectional LSTM **Epochs = 18** **Batch Size = 32** **Optimizer = Adam**	Cross Entropy loss (Lower the better) loss(train_loss): 2.2342 val_loss: 3.1726 BLEU Scores on Validation data (Higher the better) BLEU-1: 0.568254 BLEU-2: 0.312748 BLEU-3: 0.218816 BLEU-4: 0.112289	k = 3 (beam width) BLEU Scores on Validation data (Higher the better) BLEU-1: 0.579914 BLEU-2: 0.323926 BLEU-3: 0.227842 BLEU-4: 0.113637

V. CONCLUSION

In this paper, we have used Flickr 8k dataset with various image captioning models to compare the performance of different models. We have used CNN encoders like VGG16, InceptionV3 etc. for converting features into numeric vectors. These features are then passed to unidirectional or a bidirectional LSTM for generating captions. We used the BLEU score metric for comparing the accuracy of different image captioning models. To conclude we can say that for all types of Convolutional networks (encoders) the Bidirectional LSTM gave better results than the unidirectional LSTM. Also, for same type of decoder i.e. LSTM or BiLSTM, inceptionV3 encoder model performed better than the VGG16 model. Each type of method has its own merits and limitations like we have seen a BiLSTM performs better than a unidirectional LSTM as a unidirectional LSTM runs an input in only one direction so it preserves context only from the past whereas a BiLSTM runs input in both directions, once in a forward direction and once in a backward direction such that it preserves information from both past and future which helps it in understanding the context better. At the same time, hidden layers in BiLSTM are more complex as compared to LSTM and require huge computational power. Also, BiLSTM cannot be used for purposes like speech translation where you can't wait for whole input before beginning the inference.

VI. LIMITATIONS AND FUTURE WORK

Although experimentation with given models, datasets and hyperparameters show pretty good results but there are certain limitations to the proposed work like we did not have machines with higher processing power. Higher computational powers would have enabled us to further fine-tune the hyperparameters like batch size and learning rates which we believe would have resulted in better performance. Also, the dataset we have used contains only 8000 images. Using larger datasets like Flickr30k, MS-COCO etc. would mean we have more images to train the model on and it will ultimately lead to better accuracy. Also, larger datasets would mean we have larger vocabulary of words to train the model which would lead to better and more grammatically correct captions. But for working with these larger datasets, we would require machines with high computational powers otherwise it will take a lot of time to train the model on these datasets. The work we have done in this paper is just a small part of a large research area, there is lot of research which can be done in this field. For future prospects we suggest following improvements:

1. Using larger datasets: We can make use of larger datasets like MS-COCO, Flickr30k or Stock 3M datasets which will increase the vocabulary size thus enhancing the model accuracy significantly. It will help to generate better and diverse captions for an image.
2. Hyperparameter Tuning: The hyperparameters related to the model can be further fine-tuned to improve the accuracy score of the model.
3. Implementing Attention based Model: Nowadays attention mechanism is becoming quite popular. In future prospects, we can make use of attention-based mechanism which can easily focus on different parts of the image while output sequence is being produced.
4. Apart from this, newer models like inception-v4[20] or inception-resnet[20] can be used to improve the BLEU score. Also, we can make use of other RNNs like Gated Recurrent Unit[21] to have more detailed comparison of different models.

ACKNOWLESDGMENT

We would like to express our gratitude towards our research guide Dr. Akshi Kumar (Assistant professor, DTU) for her constant guidance, supervision and constructive criticism during the successful completion of the research work.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[2] X. Chen and C. Zitnick, "Learning a Recurrent Visual Representation for Image Caption Generation," 2014.

[3] D. V. T. and V. R., "RETRIEVAL OF COMPLEX IMAGES USING VISUAL SALIENCY GUIDED COGNITIVE CLASSIFICATION," J. Innov. Image Process., vol. 2, no. 2, pp. 102–109, Jun. 2020, doi: 10.36548/jiip.2020.2.005.

[4] Y. Bengio and Y. Lecun, "Convolutional Networks for Images, Speech, and Time-Series," 1997.

[5] D. E. Rumelhart and J. L. McClelland, "Learning Internal Representations by Error Propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, 1987, pp. 318–362.

[6] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7298968.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7298935.

[8] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7299087.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-December, doi: 10.1109/CVPR.2016.308.

[11] S. Hochreiter and J. J. Urgen Schmidhuber, "Long short term memory. Neural computation," Mem. Neural Comput., vol. 9, no. 8, 1997.

[12] M. Basaldella, E. Antolli, G. Serra, and C. Tasso, "Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction," 2018, pp. 180–187.

[13] "Flickr 8k Data | Illinois." https://forms.illinois.edu/sec/1713398 (accessed Mar. 05, 2021).

[14] "COCO - Common Objects in Context." https://cocodataset.org/#download (accessed Mar. 05, 2021).

[15] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models," in 2015 IEEE International Conference on Computer Vision (ICCV), Dec. 2015, pp. 2641–2649, doi: 10.1109/ICCV.2015.303.

[16] "Image_Caption - Google Drive." https://drive.google.com/drive/u/2/folders/181xqs33zg5-PIv_VReG8VE13Fi6rbPIi (accessed Mar. 16, 2021).

[17] "9.8. Beam Search — Dive into Deep Learning 0.16.1 documentation." https://d2l.ai/chapter_recurrent-modern/beam-search.html (accessed Mar. 13, 2021).

[18] C. Meister, T. Vieira, and R. Cotterell, "Best-First Beam Search," arXiv. 2020, doi: 10.1162/tacl_a_00346.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," ACL, pp. 311–318, 2001, doi: 10.3115/1073083.1073135.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2017.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014.