Image Captioning using Deep Learning Models

Sougandhika Ayathamaraju
SIS ID: 015383771
San Jose State University
San Jose, US

Neelima Jagtap
SIS ID: 015295956
San Jose State University
San Jose, US

Rishikumar Ravichandran
SIS ID: 015280707
San Jose State University
San Jose, US

Greeshma Venkatesh
SIS ID: 015277041
San Jose State University
San Jose, US

***Abstract*—The term image captioning refers to describing the image based on the objects present in the image. It is a joint method of computer vision and Natural Language Processing. The focus of this project is to create and compare the different neural network methods for image captioning and create a web application that helps the old aged and visually challenged people to learn about the neighboring objects and their surroundings. In this project, four different deep learning models are created using Bi-LSTM, Local attention, global attention, and transformers. All the models are trained, tested, and validated using the same Flickr 8K dataset, and the prime evaluation metric used is the BLEU score. Upon testing the transformer model with the Inception V3 provided the highest score on which a web application is built.**

***Index Terms*—Captions, Bi-LSTM, Attention, Transformers, BLEU score.**

I. Background & Motivation

The establishment in data science made the data scientists to break down the complex problems and provide a solution that is helpful for mankind. Additionally, the focus on these enhancements paves a way to come up with new innovations which makes a safer environment for human beings. Image captioning is one such innovation that focuses on providing a caption to the image in a proper grammatical sentence based on the objects present in the image. This method is achieved in the world of computer vision by training the system with neural networks. The goal of this project is to provide captions for images using different deep learning models and analyzing the best approach. Based on the best approach obtained from the four models with the help of Flask API an image captioning bot is built.

In this project there are four deep learning models created with the help of an encoder and decoder mechanism to provide a sequence to sequence captions [1]. For object detection on the images, the Convolutional Neural Networks (CNN) are mostly being used as it acts as the encoder for extracting the feature vectors from the images [2]. The pre-trained CNN models used in this project are VGG16 and Inception V3 [3]. In coming to the decoder part, four different architectures are used that are Bi-LSTM in RNN [4], local and global attention [5], and also transformers [6] are used to provide the captions to the images. This project also paves the way to analyze the images on social media and perform the visual sentimental analysis.

II. Literature Review

The necessity of image captioning grabbed the attention of data scientists and technology companies resulting in investing on building different models and approaches for captioning the images. In [7], created a model in order to caption the images using the k-nearest neighbors algorithm. This model was implemented on the coco dataset and the method was initiated by converting the images into a size of 32x32. The pre-trained model used for the feature extraction is VGG16 and then the captions are generated using k-nearest neighbors. The model was initially able to produce captions with the Bleu score of 26. But, later on, in the long run, the model wasn't capable of producing a good result. It happened as the KNN algorithm was unable to calculate the distance among the points.

According to [8], suggests a new method for captioning the image using a Generative Adversarial Network (GAN). The authors claim that most of the existing image captioning model provides the caption or describes the images accurately but the words generated don't have any emotions attached to the image. To address this problem, it is suggested that creating the model with GAN can provide captions to the images with the emotion attached to it. However, the research in this paper is still ongoing and there is no architecture or implementation performed in order to show the strong evidence.

In [9] a new model is proposed to complete the incomplete sentences by using the bi-LSTM model. The main focus of the project is to provide suggestions to the users about the next word in the sentence based on the previous words typed in the sentences. The CNN architecture for feature extraction is not required as there is no image is used. The project resulted in two LSTM being able to predict the next words with the Bleu score of 24.4 on the validation. Though this is not an image captioning project, this proves that by using the bi-LSTM model the prediction of words can be generated on an above-average metric score.

In [10], the researchers proposed a model to find the request sentences by using the LSTM and Attention mechanism. Their method starts with splitting the sentences into words and capturing the word embedding vectors. This word embedding

vector acts as an input for the LSTM model and the LSTM generates the sentence vector. These vectors are then given to the self-attention model to get the weight of the vector. Then, this calculated weight is sent to the multi-layer perceptron in order to find whether the sentence is a request sentence or not. Though it is not an image captioning project this research helps in proving the insight that, the attention mechanism is faster in word or sequence to sequence processing. On studying all the above researches and other open sources researches, it is clear that Bi-LSTM and attention provide a greater result with image captioning when using it with the CNN. In this project mainly four models with different combinations of encoders and decoders are used to look for the model that provides a greater Bleu score for the image captions.

III. Methodology

In this project data collection is the first step here we choose the Flickr 8K dataset.After data collection we save this data on drive so every team member can access it. The next step is data preprocessing. Here we did image preprocessing and data preprocessing part.Data collection and preprocessing part is common for all above Which is mentioned in the data preprocessing part in detail. After that pretraining model is done. For feature extraction on the transformer we used inception v3, and for other models we used VGG 16.

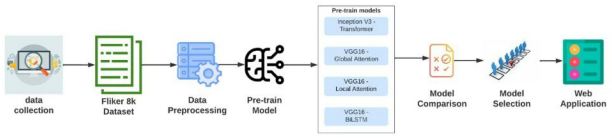


Fig. 1. Methodology

We used four different models: Transformer, Global attention, Local attention, and Bi-LSTM. After comparing the model based on the BELU score here in this comparison point we used the same dataset on the above four models. Based on the score for the web application part we choose the Transformer model.The majority of similar answers is taken in count for the images caption generation. This project is deployed on a webserver and published in a website using Flask framework Upon obtaining the desired result of the proposed model, for future considerations, the project will be integrated and published in mobile applications. Also, this project opens the way for visual sentimental analysis based on the images posted on social media for future considerations.The web application is created which will take the image path as an input to process the given image and display the predicted caption of the image.

A. Data Collection

For the purpose of this project we used the popular Flickr 8K dataset that has set a benchmark for sentenced-based image captioning in recent times [11]. This data set contains about 8,000 images, each of these images have about five unique sentences as captions. These sentences are framed by AMT (Amazon Mechanical Turk) workers by providing detailed descriptions of the images considering all the important entities involved in the image. The dataset consists of two files: the dataset file that contains all the 8K images which are of size 1GB and the text file that contains text captions for the images in the dataset file which is of size 2.2MB. In the total of 8K images we use 1000 images for validation, 1000 images for testing and the remaining 6000 images for training.

B. Data Preprocessing

We perform preprocessing of the data as it helps in eliminating huge inconsistencies and uncertainty. It makes sure that the data is clean and is free from all types of human errors, thus resulting in the accurately predicting model [12]. The data initially contained html tags such as $< doc >$,$< html >$ etc., for cleaning the data all these html tags were first removed. or data preprocessing we initially clean the data. Next we looked for the punctuations that were present in the dataset and removed. Punctuation is used to determine if the text is a sentence or a paragraph and having punctuations affects the text processing approach. This also affects in determining the frequencies of the words as these punctuations occur many number of times in the text.

Further the stop words such as ‘a’, ‘the’, ‘is’ etc., are removed to eliminate the most commonly used words that do not contain any useful information and meaning. The single characters such as comma, brackets, colons etc., and non-alphanumerics were removed from the text. The data was looked for incorrect capitalization and the text with capitalization at incorrect places were removed. Then the tokens are mapped to its numerical indices which is known as vocabulary . The number of times the token that is present in the corpus is calculated and a numerical index is assigned to each of these tokens depending on the number of times it is repeated. For the deep learning model to understand the beginning and ending of each sentence special tokens are added at the beginning of the sentence and the ending of the sentence. The special tokens that are used : ¡start¿ and ¡end¿ that are used to identify the start and end of a sentence respectively. The initial count of vocabulary before data preprocessing was 8871 and the count of the vocabulary after performing the data preprocessing techniques were 8114. The sample of the texts after preprocessing and adding the special tokes is shown in the image below.

```
['<start>  child in pink dress is climbing up set of stairs in an entry way <end>',
 '<start>  girl going into wooden building <end>',
 '<start>  little girl climbing into wooden playhouse <end>',
 '<start>  little girl climbing the stairs to her playhouse <end>',
 '<start>  little girl in pink dress going into wooden cabin <end>',
 '<start>  black dog and spotted dog are <end>',
 '<start>  black dog and dog playing with each other on the road <end>',
 '<start>  black dog and white dog with brown spots are staring at each other in the street
<end>',
 '<start>  two dogs of different breeds looking at each other on the road <end>',
 '<start>  two dogs on pavement moving toward each other <end>']
```

Fig. 2. Sample Dataset after Pre-Processing

C. Data Augmentation

In order to increase the diversity of the images and to train the model by making it understand and predict the same images given in different forms correctly a method known as data augmentation was performed. It is a technique of image processing that is used to artificially expand the image dataset size. We used two main types of data augmentation: Position Augmentation and Color Augmentation . The position augmentation is performed on the images to generate images with various imaging positions where pixel positions of the same image are changed to generate multiple types. The Color augmentation is performed on the images to generate images with various imaging colors where pixel values of the same image are changed to generate multiple types [13].

In Position Augmentation we performed methods such as scaling, cropping, flipping and padding.

- Scaling: In this the original image size is altered by increasing or decreasing the the length and width of an image.
- Cropping: In this the original image is cropped at different center points and returns the cropped portion of the images.
- Flipping: In this the original image is flipped on its x-axis or y-axis i.e horizontally or vertically respectively.
- Padding: In this the original image is padded with the given set of values on each of the sides of the actual image.

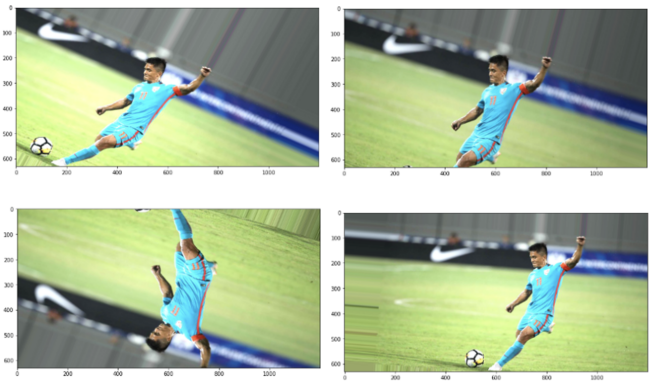


Fig. 3. Sample Data after Position Augmentation

In Color Augmentation we performed methods such as brightness, contrast, saturation and hue.

- Brightness: In this the brightness of the original image is altered such that the original color of the image will become darker or lighter.
- Contrast: In this the contrast of the original image is altered such that the color varies and is adjusted to be between the darkest and the lightest areas of an image.
- Saturation: In this the saturation of the original image is altered such that the color varies and is adjusted to be between actual colors of an image.
- Hue: In this the hue of the original image is altered such that the color shade of the original image is changed.

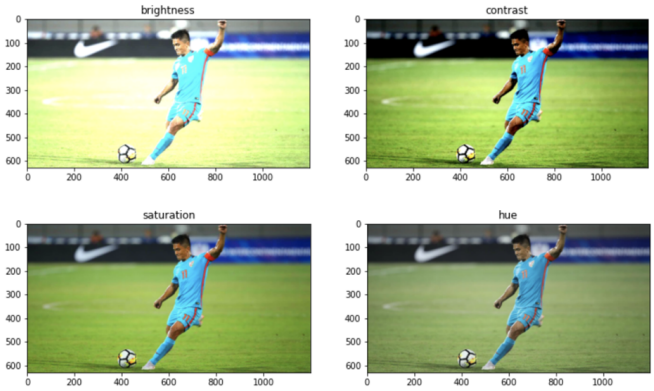
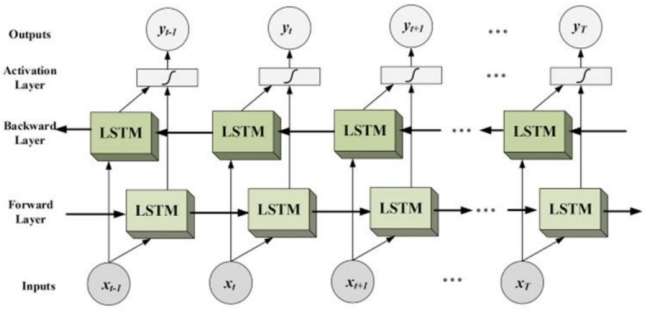


Fig. 4. Sample Data after Color Augmentation

IV. Modeling & Implementation

A. Bi-LSTM Model

Bi-Directional Long-Short term memory (Bi-LSTM) model is nothing but an extension of the traditional Long-Short term memory (LSTM). This model helps in eliminating the disadvantages of the LSTM in cases of long text prediction and the time consumption. We directly chose Bi-LSTM over LSTM because the output of any statement is dependent on both the left and right side of that word which LSTM cannot support that effectively and its complexity increases during the long texts and takes more time to process [14].



Source-(Dhruv, et.al.2022)

Fig. 5. Bi-LSTM Architecture

Bi-LSTMs model uses two LSTMs: forward pass which considers the previous layers and a backward pass which considers the next or future layers thus it keeps both information about the sequence at every time step. Hence the usage of Bi-LSTM provided significant improvements in a network as it understands the context better way. Then the

output from both the layers is passed through an activation function, we use the softmax layer so that the decision can be made based on the probability.

Model Implementation Details:
In this project Bi-LSTM model is implemented for image captioning. The model architecture is leveraged and used with VGG16 as encoder.

1) VGG-16 is used for encoder and Bi-LSTM is used as decoder.
2) Number of Epochs ran in the model - 30
3) Average processing time per epoch - 60 seconds
4) Dropout rate - 0.5
5) Batch size – 128
6) Buffer Size - 1000
7) Embedding Dimension - 256, 512

B. Global Attention Model

Global attention model, often called Luong attention, takes input from all the hidden states of the encoder. When speaking in context to sentence generation for image captioning all the source words are studied in the input before generating the alignment scores. The working architecture of global attention is very similar to the conventional attentional model [15].

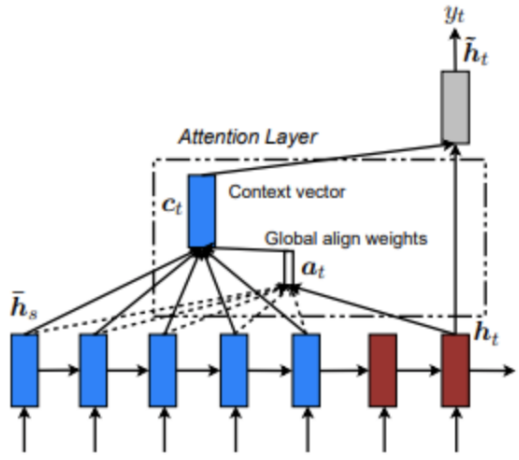


Fig. 6. Global Attention Architecture

The context vector is obtained by considering all hidden states and as for each time step the alignment weight vector length varies based on the existing source words. Suppose if the third time step is running then previous time steps hidden states are considered. The alignment scores are calculated using different approaches which are dot, general and concat.

Model Implementation Details:
In this project Global attention model is implemented for image captioning. The model architecture is leveraged and used with VGG16 as encoder. The model used General score for alignment score.

1) VGG-16 is used for encoder and Luong global attention as decoder.
2) Number of Epochs ran in the model - 15
3) Average processing time per epoch - 80 seconds
4) Alignment score used - 'general'
5) Dropout rate - 0.5
6) Batch size – 64
7) Embedding Dimension - 256, 512

C. Local Attention Model

The local attention model that is also called as Bahdanau attention algorithm gives attention only to selected state in the input [16]. In this project the local attention is used as it takes concatenation of both forward and backward source hidden state. The advantage of the local attention is that the overall cost for attention computation is reduced.

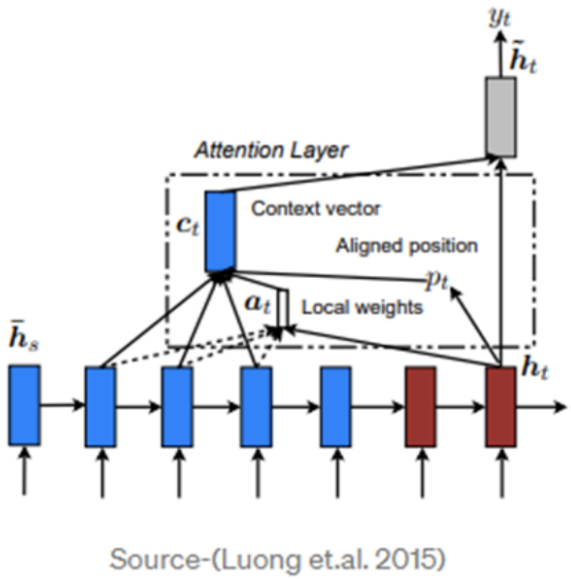


Fig. 7. Local Attention Architecture

In this attention, during the time period 't', the 't-1' hidden state is also taken in count and followed by obtaining the context vector and alignment score. The concatenation is performed with the 't-1', so instead of having different alignments this model only takes the concatenation score for the alignment.

Model Implementation Details:
In this project local attention model is implemented for getting the captions to the images. The model architecture is leveraged and enhanced with the following details:

1) VGG-16 is used for encoder and Bahdanau attention as decoder.
2) Number of Epochs ran in the model - 15
3) Average processing time per epoch - 54 seconds
4) Dropout rate - 0.5
5) Batch size – 64
6) Embedding Dimension - 256, 512

D. Transformer Model

Transformer-Xl, Entangled Transformer, Meshed memory types of transformer present for the deep learning applications. The main advantage of the transformer is it will avoid the recursion problem which will cause more memory utilization and increase the time complexity of the application. The architecture is similar to the RNN but the transformer gets the input sequence in parallel form.

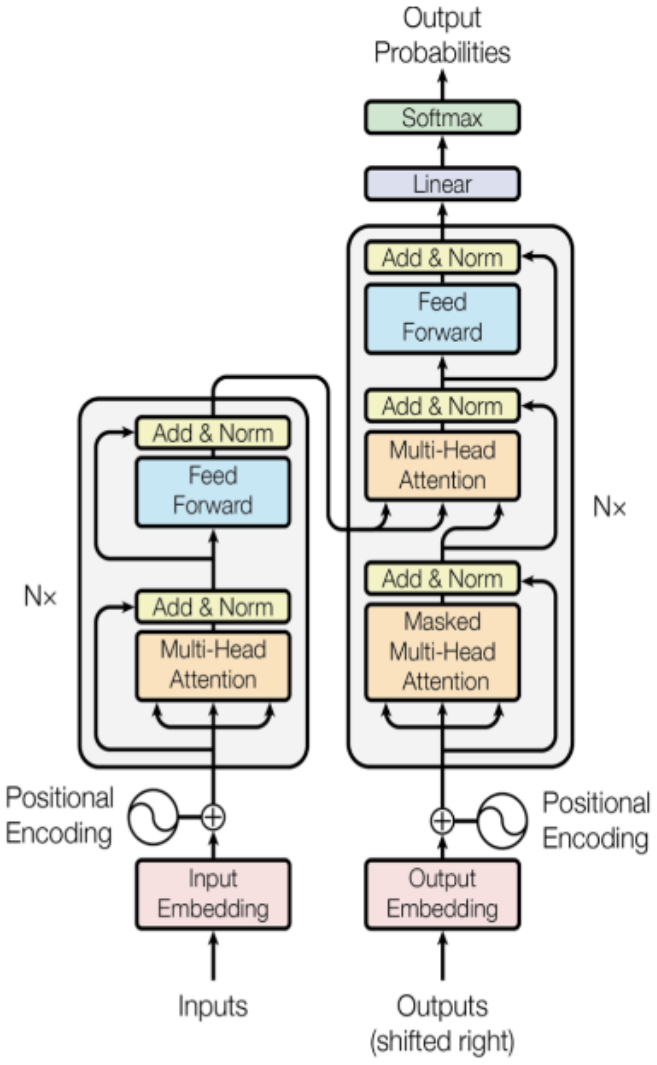


Fig. 8. Transformer Architecture

For Image captioning application if similar words in a sentence are present then according to the word position it gives a different meaning for the particular sentence. It is nothing but position encoding. This problem is solved in the transformer as it considers the absolute position of words in sequence. Implementation of the idea is done by fixed or learning weights which will encode the information in the form of tokens.

The encoder consists of two sub-layers: the first layer is a Multihead attention mechanism and the second layer is FC feed-forward network. Every word can generate the attention vector and link relationships between words present in the sentence. In the decoder, it will add the next third layer for the output of the encoder stack [17]. The attention block will capture the relationship between vectors with respect to the other vectors. The decoder is nothing but a classifier. Here the softmax function is used for word probabilities in the sentences.

For the feature extraction in this project, we used the Inception V3. Firstly we implemented the feature extraction by using VGG 16 but it took a long time to train the model and the disk and bandwidth weight themselves were pretty large [18]. So to avoid the above problem we used the Inception V3. It extracts the image vector for the image dataset. The shape of output layer is 8*8*2048. Map each image caption name to the particular function to load the images. We divided the dataset into training, testing, and validation set. The next step is to create an odd index for the input vector by using the cos function. In a similar way, even an index is created by the sin function. After that, add those vectors to their appropriate input embeddings, which offers the network information on each vector's position. The next step is to calculate attention weight for the query(q), key(k) and ,value(v). In such a way that seq_len_k = seq_len_v it should match. Define hyperparameters for training and here define 30 epochs. Last, is the step to calculate the BELU score.

V. Results & Model Comparison

To understand how a mode is predicting varies completely depending on the output of the model, if the model is classifying the images, the model's prediction is evaluated differently, and the predicted class is verified with the actual class of the image. The accuracy should be enough to understand the prediction but when the output is image captioning which is our project, understanding our prediction is complex because the caption generation needs to appropriately explain the image and there is no one exact class or solution to verify with the label. In order to understand the prediction of the model a good metric is necessary to assess the range of the possibilities in the captions and aptly represent them when compared to the actual caption.

BLUE score is such a metric that is widely used for image captioning to understand how good the model is predicting. Bilingual Evaluation Understudy Score is often called the BLEU score. It is widely adopted due to its various

advantages. It is very economical to implement, and it is very rapid. The interpretability is high, and the reason is further explained in the following report. It is independent with language as it ignores language nuances. When considering image captioning, each image has more than one caption especially in our dataset then BLEU score is the best metrics to be considered as it correlates with human evaluation.

BLEU score is evaluated using precision n-gram and brevity penalty. N-gram represents the set of sequential words in a sentence. The precision n-gram is calculated by considering words which are represented by n, if it is 1 gram one word at a time is taken to compare the predicted sentence. So precision n-gram calculated the correct predicted n-grams to total predicted n-grams [19].

Precision n-gram = Number of correct predicted / Number of total predicted

Geometric precision n-gram is evaluated using different weight values and this is used to calculate BLEU score Here brevity penalty is the key factor that makes blue score reliable. Suppose a sentence is really short then it will have a high blue score when calculated precision so to balance the parameter is multiplied to precision called brevity that penalizes the short sentences. If geometric precision1-gram is used, then it is called BLUE-1 score and similarly [20].

In our project four different models are implemented and each model is tested for the same image to compare and understand the prediction of the model. We have used an image of a group of people sitting on the wall. The real caption used is "A group of teens sit on a wall by a beach". This real caption is human generated, and this is used to include and understand the real captions generated by humans. As indicated previously, BLUE score is used here to understand the model prediction.

From our results, it can be seen that the local attention model performs better than Bi-LSTM model.

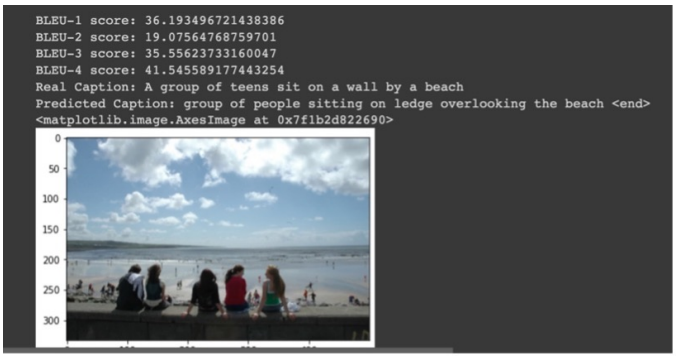


Fig. 9. Bi-LSTM Model Prediction

Fig. 10. Local Attention Model Prediction

The BLEU-2 score is lowest for Bi-LSTM which is 19 and the same for attention but higher than others with 21.3. When compared for BLEU-4 which is the highest score metrics for both models still local attention is good with around 44 when compared to 41.5 of Bi-LSTM. The reason for this is because of the model architecture's superiority, the attention model is more rapid compared to the Bi-LSTM model.

Fig. 11. Global Attention Model Prediction

Fig. 12. Transformer Model Prediction

The Transformer model when compared with all other models is superior in terms of its prediction and performance. The global attention model performance is poor comparatively and the generated captions are that accurate. When compared for BLEU-4 which is the highest score metrics for both models' transformer model is good with 49.4 when compared 36.1 of global attention model being the lowest. It is clear that transformer caption generation is more accurate and identifies all the objects in the image and superior to other models implemented for this dataset.

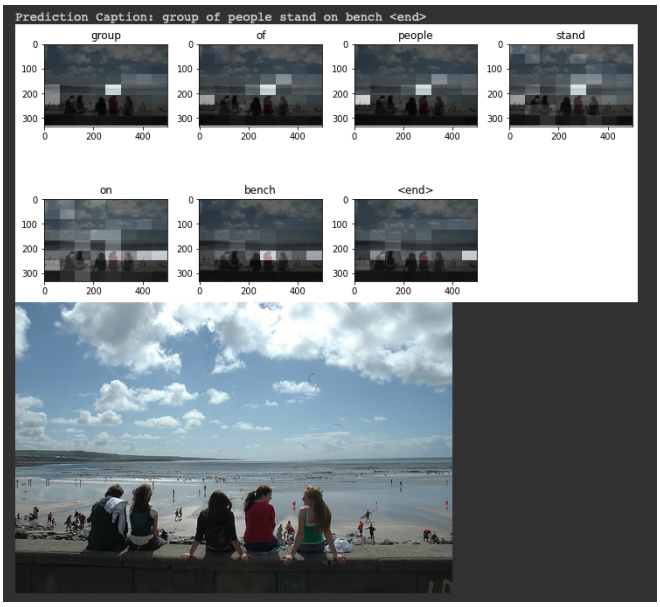


Fig. 13. Attention on different parts of image

Model Overfitting is an important concern to be taken care of, usually understanding validation and training loss helps to observe if the model is overfitting or not but when it is about image captioning the prediction phase is different from training unlike other deep learning models. In order to assess the model overfitting problem, we have implemented checkpoints in the code and observed the training loss at different epochs. We carefully implemented dropouts and batch normalization to avoid any overfitting issues and verified the model.

Web Portal:

The web application has been built using the Python Flask framework. Behind the scenes, from the python flask framework, the image will get converted into a dataset and a pre-trained transformers model will run on the image. To load the pre-trained model's checkpoint, the BertTokenizer python library has been used along with other python libraries. Which will return the predicated caption of the image. And the same predicated caption will get returned in a python flask API as a response. And using the text area, output will get displayed on UI. Following are the screenshots from the web application which will showcase the predicated caption for the image where our team is seating around a table with laptops.

Fig. 14. Input Image

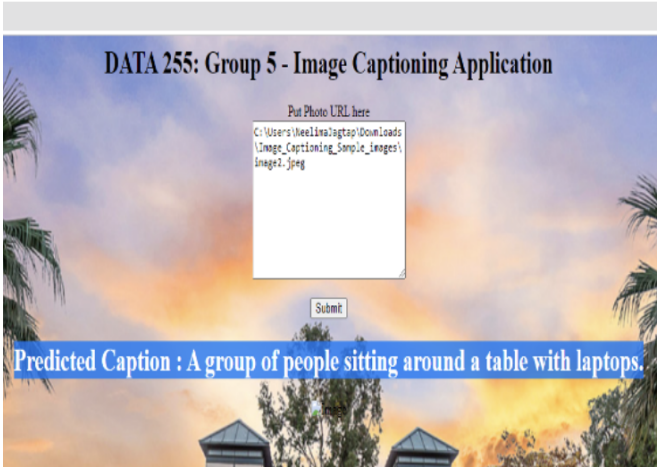


Fig. 15. Predicted Caption from Our Application

VI. Discussion

There can be many factors that influence a model by implementing four different models. We wanted to compare the performance in terms of accurate prediction. We observed that transformers have higher performance. This is because it avoids recursion problems and processes data in a parallel way. Transformers using TensorFlow can capture only the dependencies within a fixed input size which is used to train them. Initially, we have implemented VGG16 for feature extraction, but it took a long processing time to train the model and the disk and bandwidth weight themselves were pretty

large. So, to avoid the above problem we used the Inception V3 to extract the image vector from the image dataset. We also observed that global attention model performance was not very good compared to the local attention model for our dataset.

VII. Conclusion & Future Improvements

This project aims to address the models that are best suited for image captioning. The scope of image captioning is tremendous in upcoming computer vision technologies. In this project, we have selected four different models that we understood from our literature survey can be best suited for image captioning. The four models were Bi-LSTM, Local Attention, Global Attention and Transformer models. We implemented and compared these models to understand the best fit model for image captioning and observed that the transformer model performs superior to other models when predicting accurate captions for an image.

A web portal is developed using flask API to test any image with back-end code connected to our best model. This integration has helped to aptly test the image with ease and to represent our work in the form as a demo. Even though the model performance was good it can be further improved with using a larger dataset. The hypertuning of the model architecture can be furthermore experimented. Due to constraints with computation, epochs are restricted and as part of future work this can be examined. Furthermore, including more advanced visualizations to understand the process would be helpful for interpretability of the model. More advanced hybrid models can be implemented for caption generation.

References

[1] Q. Yang, Z. He, F. Ge, and Y. Zhang, "Sequence-to-sequence prediction of personal computer software by recurrent neural network," *2017 International Joint Conference on Neural Networks (IJCNN),* May 2017, doi: 10.1109/ijcnn.2017.7965952.

[2] H. Wang, Y. Zhang, and X. Yu, "An Overview of Image Caption Generation Methods," *Computational Intelligence and Neuroscience, vol. 2020, pp. 1–13,* Jan. 2020, doi: 10.1155/2020/3062706.

[3] Y. Wu, X. Qin, Y. Pan, and C. Yuan, "Convolution Neural Network based Transfer Learning for Classification of Flowers," *IEEE Xplore,* Jul. 01, 2018. https://ieeexplore.ieee.org/abstract/document/8600536

[4] A. Aziz Sharfuddin, Md. Nafis Tihami, and Md. Saiful Islam, "A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification," *2018 International Conference on Bangla Speech and Language Processing (ICBSLP),* Sep. 2018, doi: 10.1109/icbslp.2018.8554396.

[5] S. He, P. E. Grant, and Y. Ou, "Global-Local Transformer for Brain Age Estimation," *IEEE Transactions on Medical Imaging,* vol. 41, no. 1, pp. 213–224, Jan. 2022, doi: 10.1109/tmi.2021.3108910.

[6] L. Liu and P. Li, "Transformer with Local-feature Extractor for Relation Extraction," *2021 International Joint Conference on Neural Networks (IJCNN),* Jul. 2021, doi: 10.1109/ijcnn52387.2021.9534183.

[7] Devlin, J., Gupta, S., Girshick, R., Mitchell, M. and Zitnick, C., 2022. "Exploring Nearest Neighbor Approaches for Image Captioning," May. 2015, doi: 10.48550/arXiv.1505.04467

[8] S. Amirian, K. Rasheed, T. R. Taha, and H. R. Arabnia, "Image Captioning with Generative Adversarial Network," *2019 International Conference on Computational Science and Computational Intelligence (CSCI),* Dec. 2019, doi: 10.1109/csci49370.2019.00055.

[9] Z. Jia and X. Li, "iCap: Interactive Image Captioning with Predictive Text," *Proceedings of the 2020 International Conference on Multimedia Retrieval,* Jun. 2020, doi: 10.1145/3372278.3390697.

[10] T. Ueda, M. Okada, N. Mori, and K. Hashimoto, "A Method to Estimate Request Sentences using LSTM with Self-Attention Mechanism," *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI),* Jul. 2019, doi: 10.1109/iiai-aai.2019.00013.

[11] H. Tekman, Mehmet lauda. "Image Captioning - FLICKR8K." Kaggle, Kaggle, 28 Feb. 2021, https://www.kaggle.com/code/mehmetlaudatekman/image-captioning-flickr8k/notebook.

[12] Samsani , Surekha. "An RST Based Efficient Preprocessing Technique for Handling Inconsistent Data" *IEEE Xplore,* 8 May 2017, https://ieeexplore.ieee.org/document/7919591.

[13] Xu, Hao, et al. "An Enhanced Framework of Generative Adversarial Networks (EF-Gans) for Environmental Microorganism Image Augmentation with Limited Rotation-Invariant Training Data." *IEEE Xplore,* 14 Oct. 2020, https://ieeexplore.ieee.org/document/9223631.

[14] Sharma, Dhruv, et al. "Automated Image Caption Generation Framework Using Adaptive Attention and Bi-LSTM." *2022 IEEE Delhi Section Conference (DELCON),* 11 May 2022, https://doi.org/10.1109/delcon54057.2022.9752859.

[15] Luong, M. T., Pham, H., & Manning, C. D. (2015, September 20). "Effective approaches to attention-based neural machine translation," arXiv.org. Retrieved May 19, 2022, from https://arxiv.org/abs/1508.04025

[16] P. M. Hanunggul and S. Suyanto, "The Impact of Local Attention in LSTM for Abstractive Text Summarization," *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI),* Dec. 2019, doi: 10.1109/isriti48646.2019.9034616.

[17] Park, C.; Kim, B.; Kim, G. "Attend to you: Personalized image captioning with context sequence memory networks," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA,* 21–26 July 2017; pp. 895–903

[18] Feng, Y.; Ma, L.; Liu, W.; Luo, J. "Unsupervised image captioning," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA,* 16–20 June 2019; pp. 4125–4134.

[19] Papineni, K., & Ward, T. (2002, July). "Bleu: A method for automatic evaluation of Machine Translation," retrieved May 20, 2022, from https://aclanthology.org/P02-1040.pdf

[20] Callison-Burch, C., Osborne, M., & Koehn, P. (n.d.). "Re-evaluating the role of Bleu in Machine Translation Research," ACL Anthology, retrieved May 19, 2022, from https://aclanthology.org/E06-1032/