Comparative Study for Neural Image Caption Generation Using Different Transfer Learning Along with Diverse Beam Search & Bi-Directional RNN

Yash Indulkar[1] and Abhijit Patil[2]

[1]Information Technology, Thakur College of Science & Commerce

[2]Information Technology, Thakur College of Science & Commerce

Email: [1]yashindulkar31@gmail.com and [2]abhijitpatil976@gmail.com

***Abstract*—** Captions are important to understand the meaning of an image or to represent an image in a better way possible. Image Captioning needs a very precise and apt point of view of an image based on features present in that. Deep learning has advanced to a better level where we can use the power of the computer to tag an image with captions intended for it. Computer Vision is an ideal approach to use feature extraction for understanding the features present in an image for further captioning. This research is based on different transfer learning models used for feature extraction from an image along with caption generation. The transfer learning models used in this research are Xception, InceptionV3, VGG16 for feature extraction from an image. Along with the use of features, the caption needs to be generated for which this paper proposed an alternate RNN model for better caption generation, this model uses a bi-directional layer which is compared with the standard RNN model to select the best model along with the best transfer learning for neural image caption generation. For creating an apt caption with the help of feature extraction and the RNN model, the diverse beam search algorithm is used for getting the k-top best alternative values with the highest probability, which will produce a better caption as compared to argmax. The evaluation for each model with a combination of Xception-RNN, InceptionV3-RNN, InceptionV3-ARNN, VGG16-RNN, and finally VGG16-ARNN was done by using BLEU (Bilingual Evaluation Understudy) along with the training and validation loss.

***Keywords*—** Beam Search, Bi-directional RNN, Caption Generation, Natural Language Processing, Transfer Learning.

I. INTRODUCTION

A neural network is a set of neurons that are organized in layers. A neural network is a set of algorithms operating tasks on a set of data in a way by mimicking the way the human brain functions [1]. A neuron is the fundamental unit of a brain so just how the brain tries to read someone's facial expression or recognizes whether the person standing in front is a male or a female. Likewise, a neural network understands a pattern to be followed to perform a particular task on a data. It understands the pattern, makes decisions the way a human mind would make & gives outputs. Neural networks thus function in a way that gives a machine the fundamental capacity to work how a human brain works with connected neurons thereby increasing the depth of understanding real-time problems by using a technique of deep learning. Deep learning is an approach wherein algorithms are inspired by the structure of the brain. In this paper, image caption generation is done using different transfer learning [3]. Transfer learning is nothing but the transfer of learning or knowledge from an already performed task in a new task. It's a machine learning technique where a model trained on one task is reused on another task. In transfer learning, the base network is first trained on a base dataset. Thus, the neural networks used in this paper are diverse Beam search algorithms and Bi-directional RNN. A beam search algorithm is an optimized version of the Best First Search (BFS) algorithm [7]. It is also a Heuristic search algorithm wherein we find certain heuristic values. It explores a graph by expanding the most promising node in a limited set and most importantly reduces memory requirement. RNN is a class of neural networks created in the 1980s.

They are used to model sequence data. RNN has an internal memory that helps it to remember important inputs from the data received [2]. This thereby helps in increasing the prediction and the best-suited output. This is one of the main reasons why we see RNN's are used majorly for sequential data like time series, speech recognition, images, etc [6]. It consists of different LSTM model, the one LSTM model is taking the input in a forward direction which updates the weights in front line, and the other in a backward direction which updates the weights and bias in back line. We then extract the feature vector from all the images and then train the model by loading the dataset. Furthermore, we define the models used and start training the dataset.

II. LITERATURE REVIEW

TABLE I. Literature Review Table for Cited Research Papers

Sr No	Paper Year	Author & Title	Review
01	2010	Ahmet Aker and Robert Gaizauskas Generating image descriptions using dependency relational patterns.	This paper dated 2010 used dependency relational patterns to generate image descriptions.
02	2013	Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. Devise A deep visual-semantic embedding model	This paper dated in 2013 came up with a new deep visual-semantic embedding model trained to identify visual objects.
03	2014	Chen, Xinlei, and C. Lawrence Zitnick Learning a recurrent visual representation for image caption generation	This paper dated 2014 used bi-directional mapping between images and their sentence-based descriptions.
04	2015	Chen, Xinlei, and C. Lawrence Zitnick Mind's eye A recurrent visual representation for image caption generation	The paper dated 2015 gave us an idea of a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation which can be used to caption natural sentences describing an image.
05	2016	Wang, Minsi, et al. A parallel-fusion RNN-LSTM architecture for image caption generation	This paper dated in 2016 throws light upon a novel parallel-fusion RNN-LSTM architecture, which obtains better results than a dominated one and improves the efficiency as well.
06	2018	Wang, Cheng, Haojin Yang, and Christoph Meinel Image captioning with deep bidirectional LSTMs and multi-task learning	This paper presents an end-to-end trainable deep bidirectional LSTM (Long-Short Term Memory) model for image captioning. The model builds on a deep convolutional neural network (CNN) and two separate LSTM networks.
07	2019	Cao, Pengfei Image captioning with bidirectional semantic attention-based guiding of long short-term memory	This paper dated in 2019 focused on proposes a bidirectional semantic attention-based guiding of long short-term memory (Bag-LSTM) model for image captioning.
08	2020	Kumar, Akshi, and Shikhar Verma. CapGen A Neural Image Caption Generator with Speech Synthesis	This paper dated in 2020 focusses on the hybrid framework is proposed utilizing the multilayer convolutional neural network model to produce humanly apprehensible.

III. METHODOLOGY

This methodology of the paper deals with different concepts used to obtain the unknown outputs which are explained in the experimental results below. The methodology is divided into 4 sub-sections that are:

Transfer Learning

Transfer Learning is a method to use the pre-knowledge obtained from the trained model, to extract the features from a given image. Using the pre-trained models, the computational power is reduced as well as the time needed to train the custom model for feature extraction is minimized. Different transfer learning models are available from which this paper focuses on the Xception model, InceptionV3 model, and VGG16 model, which are pre-trained on a huge number of classes.

Diverse Beam Search

Diverse Beam Search is a searching algorithm that is used for getting the K-top best alternative values with the highest probability. This searching algorithm

acquires various alternatives concerning an input sequence that is based on conditional probability. This searching of various alternatives based on input sequence is done with the help of K Beam, which selects the K values with the highest probability of occurrences.

Recurrent Neural Network

Recurrent Neural Network differs from the normal neural networks, the main difference between the RNN is that it takes the input for a sequence with no determined limit set to it, this helps to capture the relationship between the inputs meaningfully. Another major difference between a normal NN and RNN is that it remembers the past and its decision is based on what the model has learned in past.

Bi-directional Recurrent Neural Network

Bi-directional RNN is a new approach for solving problems while captioning generation which is proposed in this paper. The Alternate RNN (Bi-directional) uses the final layers differently as compared to the standard RNN, the bi-directional layers connect the hidden layers of opposite directions to the same output layer, which helps to get information from the past as well as future states as the same time, which in turn generated captions more precisely as compared to standard RNN.

IV. EXPERIMENTAL RESULTS

The experimental results section of this particular research paper consists of various models and outputs obtained by performing different epochs. This paper consists of captions generated with the help of deep learning. Datasets used for Image Caption Generation are Flickr 8k image datasets with 8000 images and pre-labeled captions to them. Fig 1 shows the sample images from the dataset used for training purposes.

Fig 1. Flickr 8k Image Dataset used in this research

The various dataset was also available for caption generation such as Flickr30k as well as COCO datasets. Along with the images, the dataset also consists of tokens consisting of captions related to every image present. In total every image has 5 captions pre-set which made up the sum of 40k captions available for this research. Fig 2 shows the token file from the dataset with captions attach to it.

Fig 2. Flickr8k token file containing 40k captions

For caption generation, it is necessary to map the token file with the image present in the training folder. Once the mapping is done it becomes easier for pre-processing the data based on various parameters. The pre-processing of data is carried in 6 different steps which are shown in Fig 3 below.

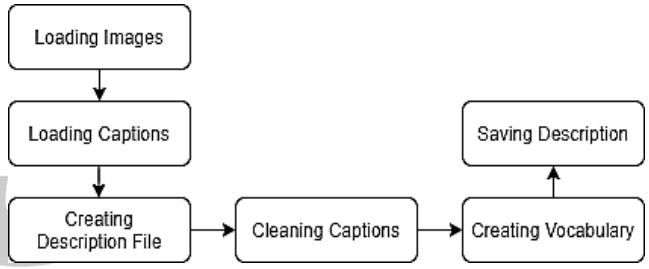

Fig 3. Pre-Processing steps involved for caption generation

These steps can be further explained as

- **Loading Images:** This step involves the initial loading of images from a dataset with bifurcation for training and testing purposes.
- **Loading Captions:** This particular step is carried after loading the images, every image contains 5 different captions associated with it.
- **Creating Description File:** New file (txt) is created for storing the pre-processed captions related to every image.
- **Cleaning Captions:** This step is carried out for cleaning the captions by removing stop words, filters, hyperlinks, etc.
- **Creating Vocabulary:** The vocabulary is necessary for generating better captions, which is done in this step.
- **Saving Description:** The final step involved pre-processing the saving the data to the newly created description file.

Once the data is pre-processed the final description file contains the cleaned caption associated with each image which can be feed to the neural network for training

purposes. Fig 4, shows the cleaned captions for every image saved in the description file.

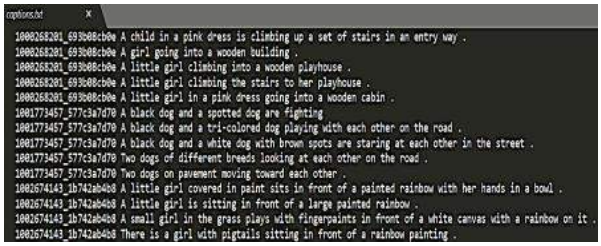

Fig 4. Cleaned captions after pre-processing steps for training purposes

These captions are further used for training purposes with the help of a neural network. Generating captions based on features extracted from images is the primary goal or objective to be performed, which requires a lot of data based on features available from the image. Flickr 8k is the smallest dataset available to train with as compared to other datasets which contain a huge number of images as well as captions associated with every image. Apart from training the captions for generating better captions, transfer learning plays an important role in feature extraction. Transfer Learning is the process of using pre-trained models that are already available for extracting the feature present in the image. This helps to reduce the computational time for training the model to extract features. The architecture for transfer learning and models presents to do the transfer learning is shown in Fig 5 below.

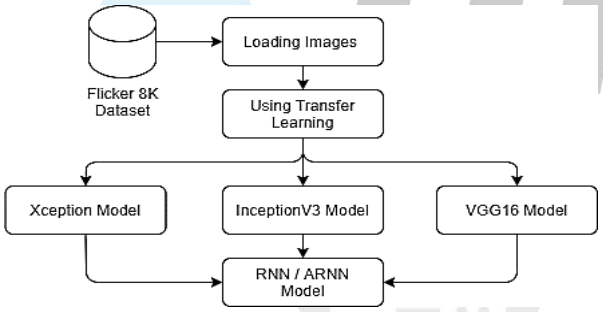

Fig 5. Transfer Learning with a different model available

The above figure shows three different models available for using transfer learning, which are the Xception model, InceptionV3 model & finally the VGG16 model. Using these models, the features can be extracted which will help the RNN algorithm to generate captions based on the extracted features. The initial step involved in using transfer learning is the load the images from which the features are going to be extracted. Every model has a different number of layers present which vary from model to model. The objective of using different models is to understand which model gives better results for extracting the features. Finally, once the features are extracted it is passed to the RNN or ARNN model for generating captions.

Diverse Beam Search is a searching algorithm that is used in this research for accurately generating better captions with the help of a searching algorithm. A different algorithm is available such as Greedy Search Algorithm, which searches the data greedily and selects the data based on a greedy search. The diverse beam search uses the k-top best alternative values with the highest probability for selecting the word in sequence. Fig 6 shows the flow for Diverse Beam Search Decoding Algorithm concerning the Image Caption Generation.

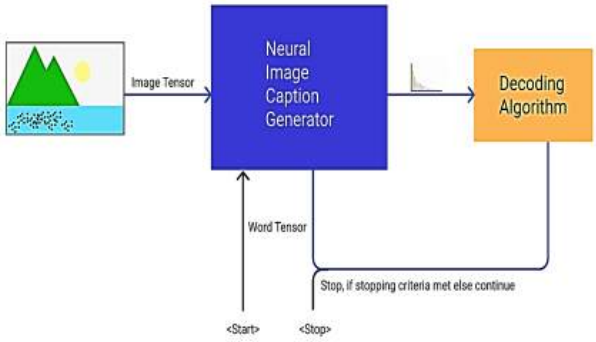

Fig 6. Diverse Beam Search Algorithm with Caption Generation

The caption generation has two tags always linked to it, the start tags and the stop tags which tells the algorithm, when to start the caption and when to end it, if it is not stated then the algorithm will keep on generating the caption without stop limit which will make no sense at all. The CNN-RNN model is created for extracting features and generating captions based on features which are shown in Fig 7.

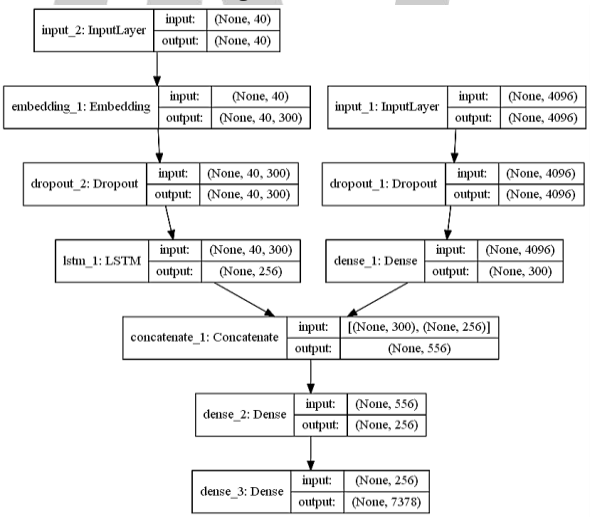

Fig 7. CNN-Recurrent Neural Network Model for caption generation

From the above figure, we can observe that the model consists of two different models clubbed together. The input layer to the CNN depends on the transfer learning model selected (for InceptionV3 the input to the model is 2048-dimension vector as well as for the VGG16 the input is 4096-dimension vector). Both the model is concatenated to create a final model with ReLU Activation Function and final layer with the Softmax Activation Function to generate the captions. Further

compilation of model is done with categorical cross-entropy loss function for minimizing the loss carried out during training phase along with Adam optimizer for solving gradient descent problem.

This research also proposes an alternate RNN model for creating captions with the final layer modified. The Alternate Recurrent Neural Network also called as Bi-directional Recurrent Neural Network is used which follows the same pattern as the normal RNN model shown in Fig 7 with final layers modified for precise caption generation. The bi-directional RNN model architecture can be seen in Fig 8. It can be observed that the initial layers where CNN is used for feature extraction are the same as compared to the standard RNN used with the 2048 & 4096 feature vector supplied from the transfer learning model. The changes done majorly are at the final layers where the bi-directional RNN model uses Time Distributed layer with dense connection to other layers for keeping the one-to-one connection based on time with input and output. The time Distribution layer takes the input from the Long Short-Term Memory layer which consists of captions provided to the model for keeping time logs which will be further supplied to the bi-directional layer after concatenating. The Bi-directional layer connects the input from concatenating layer to itself in both the forward and backward direction for remembering the past and future data at the same time, which will help the model to generate the captions in a much better way possible as compared to the standard RNN.

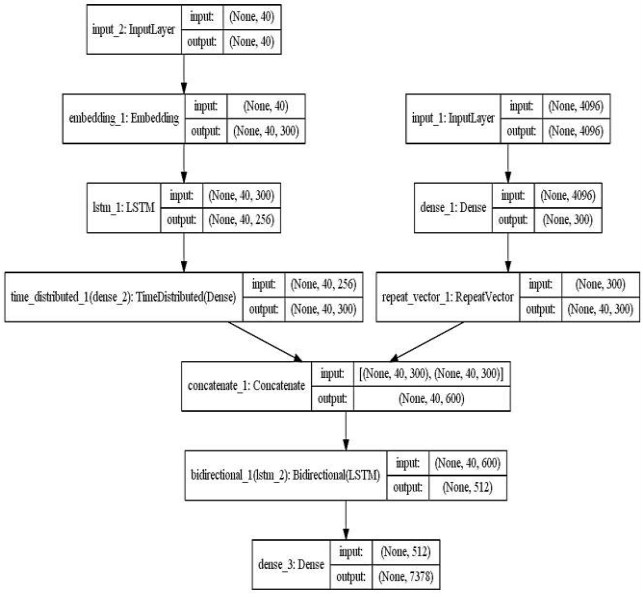

Fig 8. CNN-Bi-directional Recurrent Neural Network Model for caption generation.

The model compiles for above Fig 8 is the same as that of Fig 7, which consists of cross-entropy loss function for handling the loss while training and Adam optimizer for GD problems. Once the model is trained for a batch size of 32 and epochs 10 for every model the training and validation loss is calculated to understand how well the model is trained with given parameters. Training and Validation Loss is calculated to every model along with the Early Validation score to keep a track of good validation score change.

Below Table II shows the Training and Validation Loss for InceptionV3 Alternate Recurrent Neural Network model (Bi-directional).

TABLE II. InceptionV3-ARNN Training & Validation Loss

InceptionV3 Bi-directional RNN			
Epoch	**Early Validation**	**Training Loss**	**Validation Loss**
1	8.5794	5.5367	5.1731
2	5.1730	5.1007	4.8211
3	4.8210	4.7501	4.5381
4	4.5381	4.3339	3.9533
5	3.9838	3.7042	3.5219
6	3.5219	3.3300	3.3313
7	3.3313	3.0922	3.2241
8	3.2242	2.9167	3.1620
9	3.1620	2.7776	3.1262
10	3.1262	2.5501	3.0904

It can be observed that after the 10th epoch the training loss was 2.5501 with the validation loss of 3.0904, comparing with the initial epoch the difference was around 50 % for training loss and around 40 % for the validation loss. Similarly, the same process was carried out for the Standard RNN model with the InceptionV3 feature extraction model. Table III below shows the training and validation loss for the InceptionV3 Recurrent Neural Network model.

TABLE III. InceptionV3 RNN Training & Validation Loss

InceptionV3 RNN			
Epoch	**Early Validation**	**Training Loss**	**Validation Loss**
1	8.9512	4.8803	3.9362
2	3.9361	3.6725	3.4985
3	3.4985	3.2745	3.3172
4	3.3173	3.0412	3.2436
5	3.2436	2.8787	3.2056
6	3.2056	2.7474	3.1761
7	3.1760	2.6433	3.1571
8	3.1571	2.5516	3.1463
9	3.1463	2.4700	-
10	3.1463	2.3964	-

From the above Table III, it can be observed that epoch 9th and 10th had no validation loss, this was because the validation loss couldn't improve more (reduced) after the 8th epoch for the standard RNN model, also the training loss for the 10th epoch was 2.3964 along with the final validation loss of 3.1463 which remained unchanged. Similarly, it was carried on VGG16 along with a Bi-directional RNN model which can be seen in Table IV below.

TABLE IV. VGG16 Bi-directional RNN Training & Validation Loss

VGG16 Bi-directional RNN			
Epoch	**Early Validation**	**Training Loss**	**Validation Loss**
1	8.9341	5.6018	5.3250
2	5.3249	5.3326	5.1775
3	5.1775	5.0962	4.7301
4	4.7300	4.3496	3.8939
5	3.8938	3.6538	3.5097
6	3.5091	3.2976	3.3469
7	3.3468	3.0704	3.2626
8	3.2626	2.8922	3.2174
9	3.2174	2.7442	3.2025
10	3.2024	2.6233	3.1823

The final loss calculated at the 10th epoch for the VGG16 model is 2.6233 with the validation loss of 3.1823. The initial loss started as 5.6018 along with 5.3250 for 1st epoch which nearly made the difference of 48 % for training loss and around 39% for the validation loss respectively.

The same VGG16 model was used with standard RNN for training purposes, which in turn gave the training and validation loss for each epoch. Below Table V shows the training and validation loss for VGG16 with RNN.

TABLE V. VGG16 RNN Training & Validation Loss

VGG16 RNN			
Epoch	**Early Validation**	**Training Loss**	**Validation Loss**
1	8.9452	4.7601	3.8811
2	3.8810	3.5884	3.4989
3	3.4988	3.2011	3.3783
4	3.3782	2.9453	3.3295
5	3.3294	2.7509	3.2147
6	3.2147	2.6020	-
7	3.2145	2.4841	-
8	3.2147	2.3855	-
9	3.2147	2.3065	-
10	3.2147	2.2345	-

From the above Table, it can be observed that the validation loss was constant after the 5th epoch, which is similar to the observation from Table IV, the validation loss for the above table was 3.2147 which remain constant till 10th epoch, the training loss achieved for 10th epoch was 2.2345.

TABLE VI. Xception RNN Training & Validation Loss

Xception RNN			
Epoch	**Early Validation**	**Training Loss**	**Validation Loss**
1	8.9452	6.1254	5.2231
2	3.8810	5.2145	5.1425
3	3.4988	5.1024	5.1121
4	3.3782	4.0235	4.8214
5	4.8214	3.8575	-
6	4.8214	3.2145	-
7	4.8214	3.0124	-
8	4.8214	2.9995	-
9	4.8214	2.5412	-
10	4.8214	2.4451	-

It can be observed from the above Table VI, that the final training loss for the 10th epoch was around 2.4451 and that of validation loss was 4.8214 which remained constant after the 4th epoch till the end. The same pattern was found with other transfer learning model which used the standard RNN model for training purpose. BLEU (Bilingual Evaluation Understudy) was also calculated for all the above models to understand the quality of text generated by the models from one natural language to another. Below Table VII shows the BLEU score for all the models concerning the Beam Search algorithm for K=3.

TABLE VII. BLEU Score for Beam Search (K=3)

Model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
I-ARNN	0.609702	0.353392	0.249101	0.129054
I-RNN	0.605907	0.351905	0.243541	0.126761

V-ARNN	0.590513	0.349078	0.247277	0.129828
V-RNN	0.571849	0.330977	0.228270	0.112746
X-ARNN	0.415789	0.310544	0.215478	0.104531
X-RNN	0.405574	0.301458	0.214569	0.104451

The BLEU score is evaluated for the range of 0 to 1, where 1 been the highest perfect match score, followed by 0 being the lowest with a perfect mismatch score. The BLEU is divided into Individual N-Gram and Cumulative N-Gram score, this research is based on Cumulative N-Gram with weights starting from 1 to 0.25 range for BLEU-1, BLEU-2, BLEU-3, BLEU-4 respectively.

- **BLEU 1** – 1 Gram (Single Word)
- **BLEU 2** – 2 Gram (Word Pair)
- **BLEU 3** – 3 Gram (Words with 3 Pair)
- **BLEU 4** – 4 Gram (Word with 4 Pair)

From the above table, it can be observed that the highest score was achieved by the I-ARNN model which was 0.609702 for BLEU-1, lowest was achieved by the X-RNN model which was 0.104451 for BLEU-4. Similarly, the BLEU score was calculated for the Argmax function which gives the maximum value from the specified target variable, which in this research are captions generated from the model. Table VIII shows the BLEU score for the Argmax function for all the models.

TABLE VIII. BLEU Score for Argmax Function

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
I-ARNN	0.601124	0.355142	0.252146	0.129421
I-RNN	0.592911	0.353905	0.249141	0.126761
V-ARNN	0.599704	0.350491	0.242220	0.121935
V-RNN	0.571373	0.321116	0.214736	0.100530
X-ARNN	0.401254	0.312465	0.205614	0.100412
X-RNN	0.400214	0.305541	0.201127	0.100021

From the above Table, it can be observed that the highest BLEU score was achieved by the I-ARNN model which was 0.601124 and the lowest BLEU score was achieved by the X-RNN model which was 0.100021, which states that perfect match of words for BLEU-1 was carried out by I-ARNN model and perfect mismatch was carried out by X-RNN model. The results generated by the model for testing data based on beam search with k parameter as 3 is shown in below Fig for each algorithm used with different transfer learning models. Fig 9 shows the results of the I-ARNN model with beam search K=3.

Fig 9. InceptionV3 Alternate Recurrent Neural Network Output

Fig 10. InceptionV3 Standard Recurrent Neural Network Output

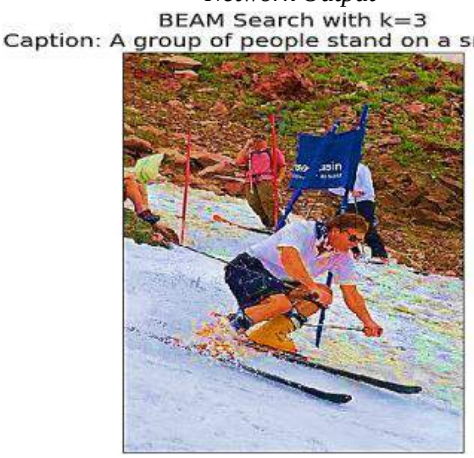

Fig 11. VGG16 Alternate Recurrent Neural Network Output

Fig 12. VGG16 Standard Recurrent Neural Network Output

V. CONCLUSION

The study aims to understand the various transfer learning model used in this research along with diverse beam search algorithms & bi-directional RNN. It can be observed by the above research is that three different transfer learning models were used with a combination of algorithms, the three models are Inception V3, VGG16 & Xception. The evaluation of these models was done by comparing validation loss & BLEU score for each. This showed that the highest BLEU score was achieved by the I-ARNN model followed by the V-ARNN model and the last was the X-RNN model. Similarly, the validation loss was also compared which showed that the minimum loss was occupied by the I-ARNN model which was 3.0904 for the 10^{th} epoch followed by the I-RNN model which was 3.1463. Along with minimum loss, the RNN model validated the data for certain epochs after that it remained unchanged, which was not the case with the ARNN model because of the bi-directional layer and time-distribution layer. The proposal for an alternate model helped caption generation by giving minimum loss as compared to other models as well as maximum BLEU score as compared to other models. Finally, the captions were generated for testing images with beam search parameter k=3, which showed that the apt caption was generated by the I-ARNN model as compared to other models with InceptionV3 transfer learning.

VI. FUTURE WORK

This research for the best transfer learning model along with RNN and ARNN algorithm showed various flaws that can be the training of epoch, customization with various optimizers, or even change in different layers for better results. The optimizer used for ARNN as well as the RNN model was Adam for solving GD problems, in the future it can be modified at least in the case of RNN for better results related to validation training. Also, the number of epochs was very less due to multiple model training, which can be increased too, the standard epoch took into consideration for training in this research was 10. Similarly, the Alternate RNN model was proposed using a single Time Distribution layer and Bi-directional layer, which can be also increased or modified concerning caption generation. The beam search algorithm used K=3 as a default parameter which can be further increased for finding more apt captions. Finally, the dataset can be increased for more training features and better image captions as more the training data, the better the algorithm will perform. Different transfer learning approaches can also be incorporated for image caption generation, which is not specified in this research paper such as VGG19.

ACKNOWLEDGMENT

This research paper was used to generate captions based on different images obtained from open-source sites such as Kaggle, Open Source Data, dew images were obtained by primary sources. The authors would like to thanks all the contributors for providing images & captions so this research could be successful.

REFERENCES

[1] Ahmet Aker and Robert Gaizauskas. 2010. Generating image descriptions using dependency relational patterns. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 1250--1258.

[2] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. 2013. Devise: A deep visual-semantic embedding model. In NIPS. 2121—2129.

[3] Chen, Xinlei, and C. Lawrence Zitnick. "Learning a recurrent visual representation for image caption generation." arXiv preprint arXiv:1411.5654 (2014).

[4] Chen, Xinlei, and C. Lawrence Zitnick. "Mind's eye: A recurrent visual representation for image caption generation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[5] Wang, Minsi, et al. "A parallel-fusion RNN-LSTM architecture for image caption generation." 2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016.

[6] Wang, Cheng, Haojin Yang, and Christoph Meinel. "Image captioning with deep bidirectional LSTMs and multi-task learning." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 14.2s (2018): 1-20

[7] Cao, Pengfei, et al. "Image captioning with bidirectional semantic attention-based guiding of

long short-term memory." Neural Processing Letters 50.1 (2019): 103-119.

[8] Kumar, Akshi, and Shikhar Verma. "CapGen: A Neural Image Caption Generator with Speech Synthesis." Data Analytics and Management. Springer, Singapore, 2021. 605-616.

[9] Yang, Zhilin, et al. "Review networks for caption generation." *arXiv preprint arXiv:1605.07912* (2016).

[10] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International conference on machine learning*. PMLR, 2015.

[11] Amirian, Soheyla, et al. "Automatic Image and Video Caption Generation With Deep Learning: A Concise Review and Algorithmic Overlap." *IEEE Access* 8 (2020): 218386-218400.

[12] Vijayakumar, Ashwin, et al. "Diverse beam search for improved description of complex scenes." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

[13] Tanti, Marc, Albert Gatt, and Kenneth P. Camilleri. "What is the role of recurrent neural networks (rnns) in an image caption generator?." *arXiv preprint arXiv:1708.02043* (2017).

[14] Kesavan, Varsha, Vaidehi Muley, and Megha Kolhekar. "Deep Learning based Automatic Image Caption Generation." *2019 Global Conference for Advancement in Technology (GCAT)*. IEEE, 2019.

[15] Katpally, Harshitha, and Ajay Bansal. "Ensemble Learning on Deep Neural Networks for Image Caption Generation." *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*. IEEE, 2020.