Image Caption Generator with an Attention Model

Aksheshkumar Shah(aashah16)
Ravendra Raghavendra(rraghav7),
Varij Patel(vpatel25)

Abstract:

This project has developed a framework using the RESNET50 V2/EfficientNetB4 and GRU with an attention model to generate a caption for images autonomously. Used the RESNET/EfficientNetB4 to extract features from the photos. Later the extracted features were fed to RNN using the GRU structure to develop the Seq2Seq model. This model alone cannot produce the best results, so we decided to use the attention model. The attention model used for this purpose was a soft attention model by Bahdanau based upon his thesis paper [2]. *The advantage of the soft attention model over the hard attention model is that it is smooth and is differentiable in dimension. The developed model was further trained and tested on state-of-the-art datasets, Flikr8K and, Flikr30K, to evaluate its performance using Adam optimizer and calculating loss using Sparse Categorical Cross entropy.*

1. Introduction

Computer Vision's primary goal is to understand the meaning of images and videos, a similar approach to how human vision and mind works. Image caption generation is one such task where visual interpretation is being translated to words; it looks simply to describe the image by using the human brain and language that one uses but making a machine learn on interpreting is much harder than perceived. Thus, while making a model which depicts the human should be capable enough to solve both interpreting's what the image shows as well as how to translate into the language which one understands so in conclusion, a reworking of human brains using 1's and 0's would not be an easy task in the field of Computer Vision and Deep Learning because compressing all the images metadata and describing them in a way one can interpret is something that requires a tremendous amount of processing.

Due to the growing field of computer vision in recent years, there have been numerous research interests in image captioning. It is observed that due to large data sets and current works in the field, the quality of image caption generation using an amalgamation of Convolutional Neural Networks which reorients images in from of numbers and vectors and Recurrent Neural Network for decoding these vector representations into Natural Language sentences has improved significantly [3]. One of the most critical aspects of the human neuron system is how data is interpreted using attention to details, and those features are also used in conjunction with the RNN.

Attention allows the significant and noticeable features to dynamically move to the top of the data description by compressing an entire image into a vector representation; this is especially important when there is noise in the image. Traditional information extraction is still apparent and purified this some essential features are lost, which can be reinformed and gained using the attention layer. In a general sense, there are two types of attention layer,

1. Soft attention:

The weights are learned and placed softly over the patches of images where the data can be extracted [2]. The advantage of this model is that it is smooth and differentiable, but it gets expensive as image size increases.

2. Hard attention:

Weights are calculated at a time for every single patch from the image [4]. The advantage of this model is less calculation for a single patch as it can be worked parallel but, the model is not smooth and differentiable.

In this project, we describe the approach to caption generation to incorporate a form of attention using soft attention by Bahdanau [2]. We have also tried to study how the advantage of including attention can visualize and predict the model. The soft generator framework is trained by using the standard backpropagation methods. We have also tried to ascertain how we will gain insight and interpret the results of this framework by visualizing where and what attention is concentrated on within the image. The performance of the model is evaluated on state-of-the-art datasets such as, Flikr8K and, Flikr30K. For the CNN part, we have used state of the art Model such as ResNet50 V2, and for the NLP part, we have enforced RNN using the attention model to extract essential features from the image, which are checked against one another to get the best results.

2. Related work and Motivation

In recent years there have been several methods that have been proposed for image captioning and most of them have taken help of Recurrent Neural Network in conjunction with Seq2Seq model [2] [5]. The major reason why image caption is well synonyms to encoder-decoder architecture is because it resembles translation model while translating between two different languages, only difference being the language is image rather than being word.

The first approach to use neural network for caption generation consisted of a multimodal log-bilinear model; this model was influenced from the feature set gained from image. This work was later followed by a method that was designed to explicitly allow a natural way of doing both ranking and generation. In the later work the feed-forward neural network was replaced by a recurrent neural network. Researchers also tried using LSTM RNNs for their models where the image is shown to RNN only at the beginning, [6]unlike the previous approaches where the models see the image at each time step of the output word sequence. Along with the image's LSTM was also applied to videos to allow the model to generate video descriptions.

In all of these works the image was represented as a single feature vector from the top layer of a pre-trained convolutional network. Further, advancements were made where joint embeddings were proposed, and models were given scores based upon the similarity of image as a function of R-CNN object detection with outputs of a bidirectional RNN. A 3step pipeline for generation by incorporating object detections. In this pipeline the model first learns detectors for several visual concepts based on a multi-instance learning framework a language model was trained on captions was then applied to the detector outputs, followed by rescoring from a joint image-text embedding space.

Unlike these models our attention model uses CNN to encode the images and then the RNN is used with Bahdanau attention to enhance the salient features in an image which is later

combined with NLP to give us the captions pertaining to the image. Prior to the use of Neural Network for generating captions, two approaches were used such as

1. Attribute discovery and object detection method were used to generate templates for image captioning.
2. First retrieving similar captioned images from a large database then modifying these retrieved captions to fit the query.

In the above-mentioned approaches, have been known to remove specifics of the caption when put all to gather. But it is observed that both mentioned methods have fallen out of favour to the now dominant neural network methods.

3. Mechanism of Architecture

3.1. Model Details

Our analysis focusses mainly on the Soft deterministic approach by Bahdanau [2] attention layer. Let us first look at the basic framework of image caption. Here the Model takes single image and generates a caption $\boldsymbol{w}$, as a sequence of 1 to C length where the C is the length of the caption,

$$w = \{w_1, w_2, \dots, w_C\}, w_C \in \mathbb{R}^D ,$$
where D is the length of dictionary

3.2. Encoder: Convolution Neural Network

Suppose in the vector space the output of the word is in matrix/vector from instead of the words then the output of the image would look like,

$$y = \{y_1, y_2, \dots, y_C\}, y_C \in \mathbb{R}^K ,$$
where K is the length of dictionary

Here, y is a vector instead of the word w. These output vectors must be extracted from the set of input image vectors more specifically combination of vectors by using the Convolution Neural Network. Thus, exaction will produce a matrix with individual column vector each having the length of the dimension of the image,

$$a = \{a_1, a_2, \dots, a_i\}, a_i \in \mathbb{R}^D ,$$
where D is dimention of the image

Now, to obtain the caption convolution must extract the features form the 2-D image and these are being extracted from the lower convolution layers and not by using the fully connected layer. It is common knowledge that to increase the accuracy of the CNN model the more the number of hidden layers the better the output will be, but it is sometimes counterproductive. Thus, a novel approach of Deep Residual learning Framework is being implemented.

3.2.1. ResNet50V2

ResNet is debatably one of the most interesting discovery in field of Deep Neural Network and Computer Vision; One may think that by simply increasing the layers in CNN there is a way

of increasing the accuracy of the model, but the drawback is that in Deep networks the training parameters are hard to detach due to backpropagation as gradient descent method implementation and thus we have a case if vanishing/exploding gradient [7].

The core idea of ResNet is called skipping method or identity shortcut method where information is skipped between different layers [6]. Here a mapping function is created such that the staking of the layers is non-linear,

$$\mathcal{F}(x) = H(x) - x$$

The original mapping is $H(x) = \mathcal{F}(x) + x$

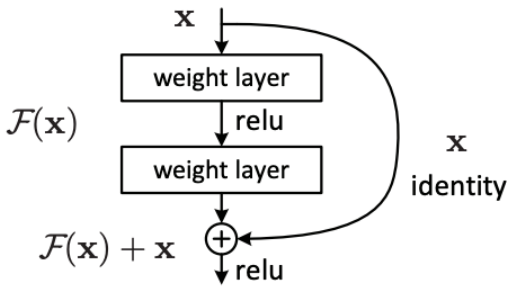


Figure 1: ResNet Basic Block

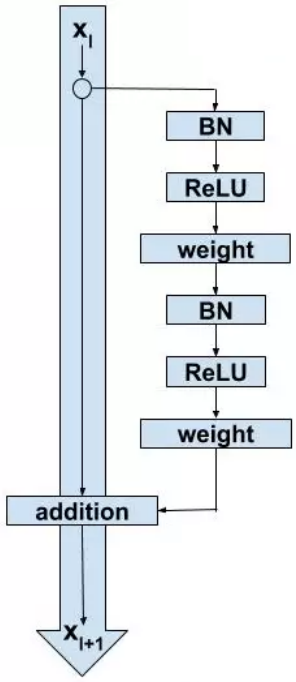


Figure 2: ResNet V2 [8]

Thus, Kaiming He *el At.* [6] that by letting the mapped layer fit is better than individual stack layers being fit directly as seen in figure 1. An advance version of ResNet50 V2 is used as seen from figure 2 where batch normalization and activation are performed twice before adding it in the layer.

3.2.2. EfficientNetB4

EfficientNet works on 3 rudimentary principals. Firstly, it divides convolution into pointwise and Depth wise convolution which in turn decreases the cost of calculation at the same time minimizes the loss of accuracy [9]. Furthermore, it adds on to ResNet in the way such that it first extends the layers and then compress it to get information from image by using MBconv. Lastly, it uses the ReLU function to prevent the loss of the information [9].

EfficientNet also has a family of networks ranging from B0 to B7 each with increasing set of total trainable parameters with increasing the validation accuracy but also consuming more time than others in the family. EfficientNet also has good recommendation for ImageNet models as of the ability to reach convergence is faster than other neural networks with fewer iterations (epochs).

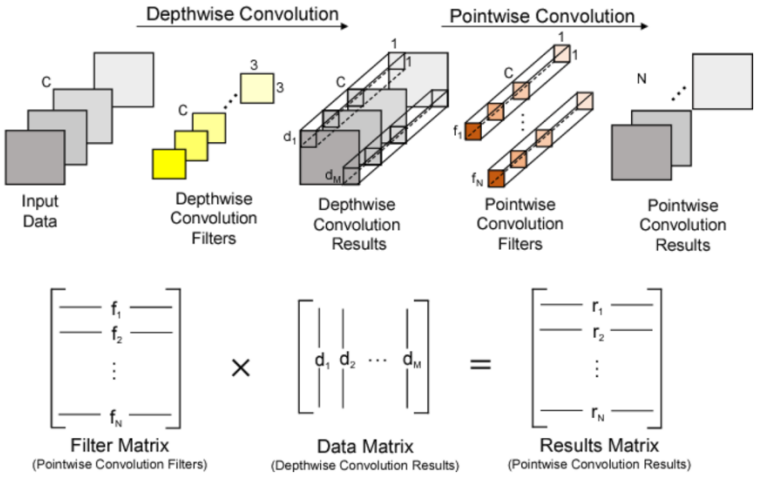


Figure 3: Point-wise and Depth-wise representation

3.3. Decoder: GRU memory network

Once the feature has been extracted form CNN network, they must be processed by Seq2Seq model to predict the captioning of the image and this process is called decoding since we will extract out all the features form the CNN into a new sequence vector which than can be compared to the dictionary such that they can be converted into human language.

Thus, to calculate the matrix output for decoding the caption GRU was used over LSTM because the number of parameters to be trained was far less than the later mentioned architecture. The reason behind using GRU is due to the newly added architecture of reset and update gate. Here, the input vectors are being translated between 0 and 1 in such a way that their combination will provide a viable score output for a better prediction. The functioning of the two gates can be described as mentioned below:

1. Reset Gate:
   It allows us control of the information between the previous hidden state and the current hidden state. The architecture of the gate can be seen in the below given figure 3 defined here as r_t,

$$r_t = \sigma_g(\boldsymbol{W_r} x_t + \boldsymbol{U_r} h_{t-1})$$

2. Update Gate:
   It allows us to control how much information from the current hidden state can enter the next hidden state. Given here by,

$$z_t = \sigma_g(\boldsymbol{W_z} x_t + \boldsymbol{U_z} h_{t-1})$$

3. Hidden State:
   They can be further separated into candidate hidden state and hidden state. The purpose of candidate hidden state is to incorporate the element wise multiplication of previous hidden state and reset gate output. Depending on how close the output of candidate hidden state is to 1 the more likely it is to recover an RNN. It can be given by,

$$\widetilde{\boldsymbol{H}}_{\boldsymbol{t}} = \tanh\left(\boldsymbol{W_h} x_t + (\boldsymbol{R_t} \odot \boldsymbol{H_{t-1}}) \boldsymbol{W_{hh}}\right)$$

Hidden state uses the update gate information to determine how much information is supposed to be carried out to the next gate which can be mathematically represented by,

$$\boldsymbol{H_t} = \boldsymbol{H_{t-1}} \odot z_t + (1 - z_t) \odot \boldsymbol{\tilde{H}_t}$$

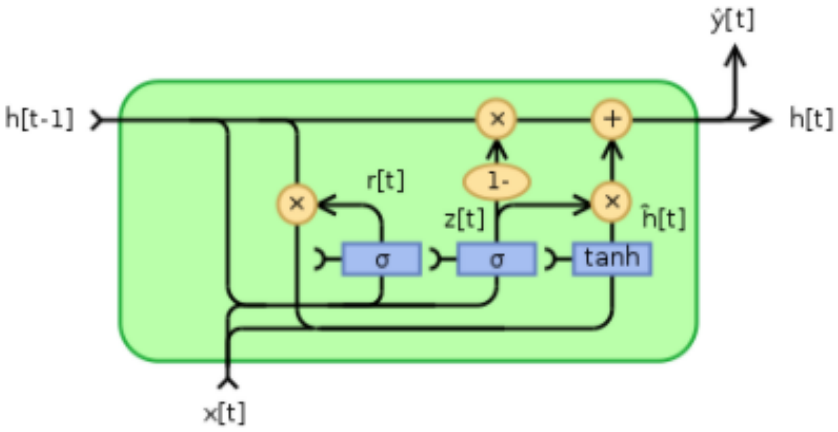


Figure 4: GRU Architecture [10]

Thus, the predicted output from the RNN architecture is being represented in the mathematical form by,

$$y_t = \boldsymbol{B H_t}$$

4. Attention Module

Attention in English is the focus on a particular key area. To mimic this kind of human behaviour in Machine Learning is quite a challenging task. It has in recent years somehow become possible due to the development of certain techniques. One such area where this attention module can be bought to application is identification of key features from an image using pixel density information. This kind of similar approach has also been implemented for Natural Language Processing where the proceeding word can be predicted from the context, but the weightage being that the context has been closely weighted such that the output is heavily dependent on what the preceding word's input features. Thus, we can use a combination of both an encoder and decoder model and are trying to include an attention module in it to improve the overall objection detection cycle and hence, enhancing the caption output. One such ways to implement attention module is deterministic soft attention model by Bahdanau [2].

4.1. Deterministic Soft attention

Once the hidden states are generated form the Encoder-Decoder model, alignment vectors are being calculated between the pervious encoder and decoder hidden state and these scores are being combined are represented into a single vector. This vector puts weight on the encoder, and it helps on what to put in turn for the decoder mechanism.

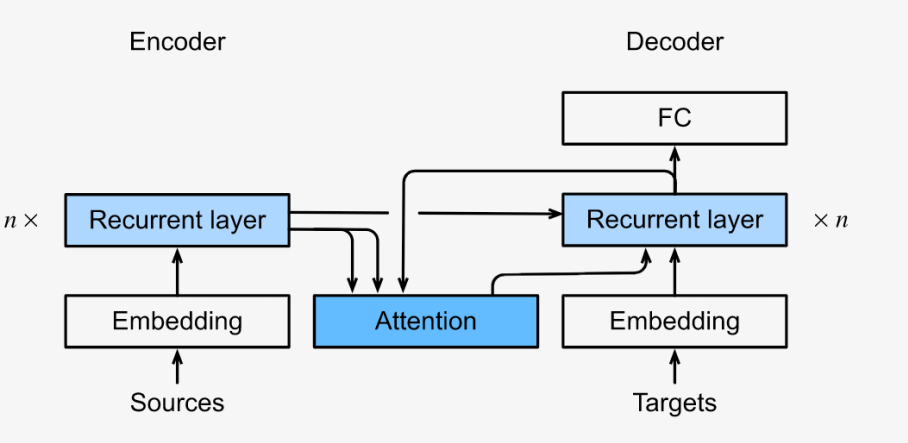


Figure 5: Bahdanau Attention Model (Encoder-Decoder)

$$Score\left(H_t,\ \widetilde{H}_q\right) = v_a^T\ tanh(W_1 H_t + W_2 \widetilde{H}_q)$$

The context vector is formed by multiplying the hidden state and the alignment score. Then passed to decoder where context vector computes the output,

$$\alpha_{tq} = \frac{\exp\left(score(H_t, \widetilde{H}_q\ )\right)}{\exp\left(score(H_t, \widetilde{H'}_q)\right)}$$

$$C_t = \sum_q \alpha_{tq} \widetilde{H}_q$$

$$a_i = f(C_t, H_t) = \tanh\left(W_c[C_t; H_t]\right)$$

α represents the weights of input vector a_i this whole model is smooth and differentiable. Since, it is differential back propagation can be applied to get a better gradient decent. The hidden activation of GRU is linear projection of a non-linear stochastic function activated by tanh(). Likelihood method can be used to calculate the normalized weighted geometric mean for the word prediction,

$$Normalized\ weighted\ geometirc\ mean[P(y_t = k\ |a)]$$
$$= softmax\left(\frac{\exp\left(\mathbb{E}_{p(s_t|a)}\ [\hat{z}_{t,k}]\right)}{\Sigma_i \exp\left(\mathbb{E}_{p(s_t|a)}\ [\hat{z}_{t,j}]\right)}\right)$$

It shows that the normalized weights can be approximated by context vector [4]. Thus, deterministic method is another form of approximation of likelihood of the context target vector over the attention features of the image vector.

5. Training Procedure

For training we had used Flickr8k dataset which consists of 8091 images with 5 annotations for each, and trained RNN model with Bahdanau Attention using Stochastic Gradient Descent using different learning rates. For the dataset used we found that Adam optimizer algorithm gave optimal results.

To create annotation to be used by decoder we made a dictionary using collections.defaultdict() to overcome any *keyerror* raised with normal dictionary container and used “ResNet50V2”/ “EfficientNetB4” with image-net weights for encoding the feature map. For our implementation time was proportional to sentence length as longer the caption group is more computational time it is going to take and dividing whole dataset as 5 to 10 batches was computationally wasteful as it took long to reach convergence and caused GPU memory exception many times. To mitigate this issue, we randomly used different sample sizes and found mini-batch sizes of 100-200 yielded better results and reached noticeable convergence faster. The training for around 20 epochs took about 1-2 hours on the GPU setup used on cluster.

We also used checkpoint for storing the weights for encoder and decoder with condition for best validation losses.

6. Implementation

6.1. Data Loader

The Flikr8K dataset used consist of 8,000 images while Flikr30K has 30,000 images of which 95% were used in training and 5% in validation, the 8K and 30K dataset comes with 5 annotation sentences per image, to tokenize the data we used tokenizer present in keras.preprocessing, and a fixed vocabulary size of 10,000 so in every run we make the reference points could be consistent. These captions were first assigned to a dictionary with the image path as keys and then appended in a way to separate them but keep their respective key and value which were then passes to sci-kit learns train_test_split which then distributed the data with .95 parameter for training set and validation set.

6.2. Training

Before we started the training which was done for the decoder or the RNN model we used a feature extraction model was used to get image feature tensors using pre-trained ResNet50V2/ EfficientNetB4 with image-net weights and then passed to the CNN encoder which consisted of fully connected layer to soft-max and normalize the feature set.

The RNN model consisted of Attention layer which in this case of Bahdanau Attention which works in way that it uses t-1 hidden state of the decoder and using that calculate alignment, context vectors which are then concatenate this context with hidden state of the decoder at t-1 and apply SoftMax on this concatenated vector which goes inside a GRU layer and the recurrent model proceeds as it is thereafter. This helps get the context for the next word in caption/annotation and increases the chances of getting sensible results rather than bad predictions.

The loss function that we used was Sparse Categorical Cross Entropy with reductions set to none and from_logits set to True, and for optimizer Adam algorithm was used which gave better results than RMSprop and SGD which were also tested during initial evaluation

The images were also resized to 224x224 to increase speed of computation and keeping all images at same resolution which was near the golden size of 256x256, but we chose the earlier one as computation time was less.

The training was done for 20 Epochs for both Flikr8k and Flikr30k dataset with same parameters to check the model and the dataset.

6.3. Evaluation and Results

After the training was completed, the results with Flikr8k and Flikr30k dataset were analysed and the Flikr30k gave more promising results thanks to its large dataset when compared to Flikr8k, although captions generated by both varied with the size of vocabulary but than it was set to 10,000 words for consistent results.

For feature extraction two CNN models were compared and used namely ResNet50V2 and EfficientNetB4. After testing both networks the results showed that EfficientNetB4 was faster than residual net with getting better validation losses at fewer epochs. The batch size was the second challenge faced (first being the selection of feature model), as if the size were too large GPU would reach out of memory and if too small computational time would drastically increase. The batch size of 100-200 images (& captions) yielded good results with multi-GPU system with less computational time.

The results of flikr8k and flikr30k with both image models are shown below along with the image captioning result with both dataset (only with optimal algorithm). The below shown images also provide visual attention with predicted and actual caption.

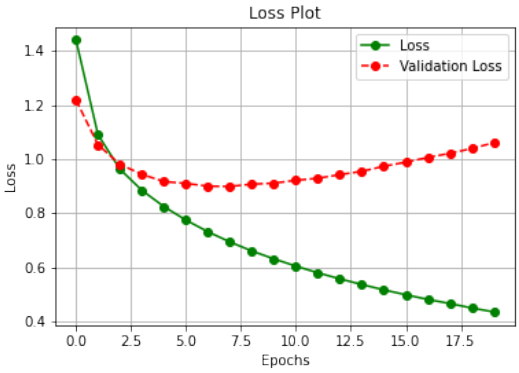


Figure 6: Loss plot for ResNet50_V2 Flikr8K

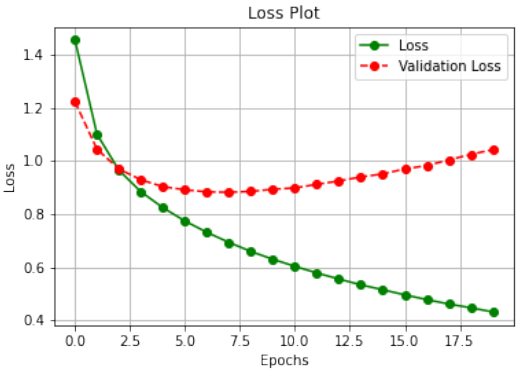


Figure 7: Loss Plot for EffNetB4 Flikr8K

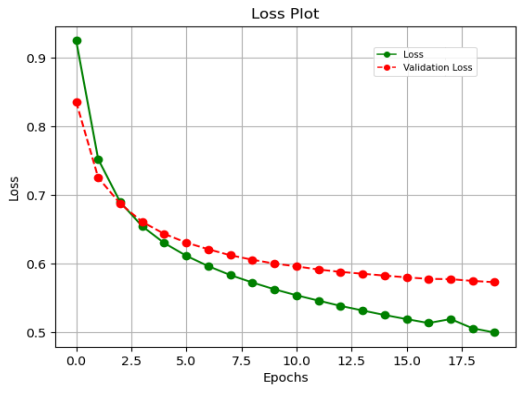


Figure 8: Loss plot for ResNet50_v2 Flikr30K

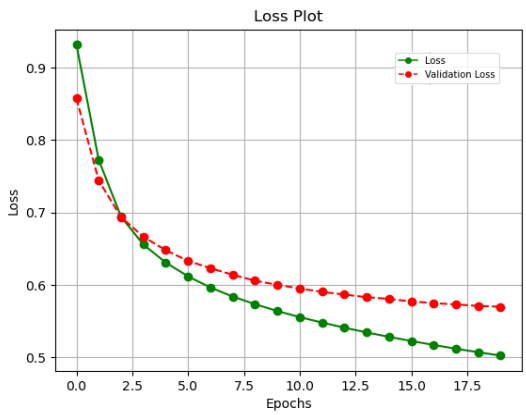
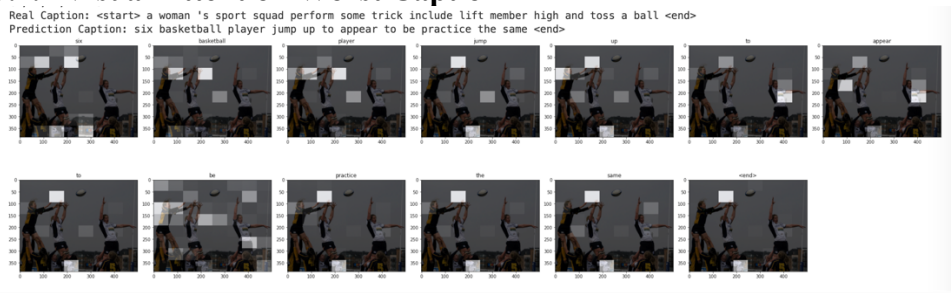
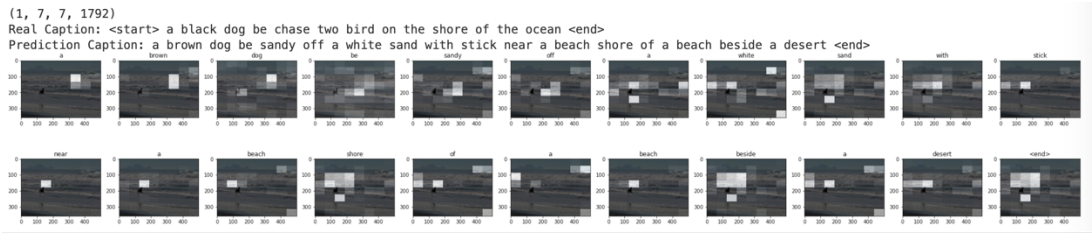


Figure 9: Loss plot for EffNetB4 Flikr30K

6.3.1. Visual Attention Best Caption

6.3.2. Visual Attention Worst Caption





7. Conclusion

We propose a neural network using visual attention-based weights for generating image captions for the well-known Flikr8k and Flikr30k dataset with minimum overfitting. We also showed how attention-based plots are generated and demonstrated the performance difference between ResNet50(V2) and EfficientNetB4 with keeping batch and vocabulary at fixed values.

As there are new research going on in fields of all Convolutional Image-Nets, Attention based neural network and Natural Language Processing, we hope in future implementation of this architecture there might be results showing better visual attention and higher validation results for even small datasets.

8. Acknowledgment

We would like to thank Dr Sebastien Motsch for giving us guidance on the workings and basic architecture of CNN, RNN, NLP and attention neural network, and we also would like to appreciate Arizona State University for the access to Agave Cluster for computation support.

9. References

[1] M. a. Q. L. Tan, " "Efficientnet: Rethinking model scaling for convolutional neural networks." International Conference on Machine Learning.," *PMLR, ,* 2019..

[2] D. K. C. a. Y. B. Bahdanau, "Neural machine translation by jointly learning to align and translate.", arXiv:1409.0473 , 2014..

[3] M.-T. H. P. a. C. D. M. Luong, "Effective approaches to attention-based neural machine translation.", arXiv:1508.04025, 2015..

[4] Q. e. a. You, "Image captioning with semantic attention.", In Proceedings of the IEEE conference on computer vision and pattern recognition,, 2016.

[5] K. v. M. B. G. C. B. F. S. H. a. B. Y. Cho, Learning phrase representations using RNN encoder-decoder for statistical machine translation., In EMNLP, , October 2014.

[6] K. e. a. He, "Deep residual learning for image recognition.", Proceedings of the IEEE conference on computer vision and pattern recognition. , 2016..

[7] K. S. a. A. Zisserman., Very deep convolutional networks for large-scale image recognition., arXiv preprint arXiv:1409.1556, 2014..

[8] cv-tricks.com. [Online]. Available: https://cv-tricks.com/keras/understand-implement-resnets/.

[9] M. a. Q. L. Tan, " "Efficientnet: Rethinking model scaling for convolutional neural networks.","* *International Conference on Machine Learning. PMLR, ,* 2019. .

[10] www.wikiwand.com. [Online]. Available: https://www.wikiwand.com/en/Gated_recurrent_unit.