10701 (Introduction to Machine Learning) Project Final Report: Caption Generation

Aboli Marathe
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
abolim@cs.cmu.edu

Malavika Varma
Department of Physics
Carnegie Mellon University
Pittsburgh, PA 15213
malavikv@andrew.cmu.edu

Shamika Dhuri
Department of Statistics and Data Science
Department of Biology
Carnegie Mellon University
Pittsburgh, PA 15213
sdhuri@andrew.cmu.edu

1 Introduction

With the rapid growth of artificial intelligence in recent years, image caption has steadily drawn the attention of many researchers in the field of artificial intelligence and has become a fascinating and difficult task. Image captioning, which automatically generates natural language descriptions based on the elements observed in an image, is a key aspect of scene understanding, which integrates computer vision with natural language processing knowledge. In image captioning, feature generation models can be divided into two main categories: a method based on a statistical probability language model [1], [2] and a neural network model based on an encoder-decoder language model [3], [4], rooted in natural language processing ideas. We employ the latter in this project. Another important aspect of caption generation we plan to take into account is attention. People have the ability to willfully disregard some of the primary information while simultaneously ignoring other secondary information. Attention is the term for this capacity for self-selection. This technique was first presented for use in the field of visual image classification employing the attention mechanism on the RNN model. There are mechanisms proposed by various researchers such as soft attention [5], hard attention, multi-head attention [6] etc. Global attention and local attention models have also been proposed [7] recently. Although image captioning has various applications and the variety of image caption systems are available today, there are many avenues for improvement. Some of the main challenges are: how to generate complete natural language sentences like a human being, how to make the generated sentence grammatically correct and how to make the caption semantics as clear as possible and consistent with the given image content. While overcoming any of these challenges completely is beyond the scope of our project, we have attempted to better understand the sources of these issues and approach the project with awareness regarding the relevance of these challenges.

2 Data

We used two main datasets for the training of our model: the Flickr8k dataset, and the Flickr30k dataset. The Flickr8k dataset [9] is a curated dataset for sentence-based image descriptions applicable for captioning. The images were manually selected, containing diverse scenes from 6 Flickr groups. The special feature is that 5 captions have been provided for each image, which provides a more robust representation for each scene for the model to train on [9]. Some problems that it could have

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

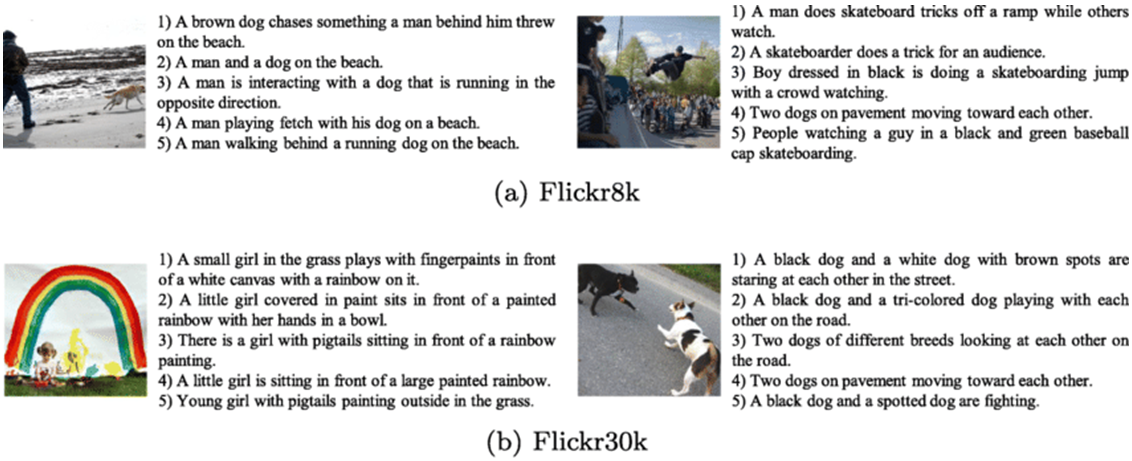

Figure 1: Examples of training images from our datasets. Image courtesy: [8].

could be the underrepresentation of other Flickr groups, absence of the many possible real-world scenarios, over-simplified captions, lack of diversity in lighting or seasonal conditions and so on. We adapted our code to a larger dataset, Flickr30k, to overcome a few of these shortcomings. The Flickr30k dataset [10] contains over 31,000 images collected from Flickr, together with 5 reference sentences provided by human annotators. This was obtained from an online source set up by the corpus creators[11]. While the organization of the images and captions is similar to that of Flickr8k, the almost 4-fold increase in the number of images makes it a more diverse dataset in many aspects. Still, there are many adversarial factors that could throw the feature detector off the right predictions when trained on this dataset. In the scope of our project, the dataset models basic scenarios for which we aim to generate captions, like a man climbing a wall. This dataset is suitable for our task, and would also be suitable for other tasks like text-to-image generation or style transfers.

3 Background

Figure 2: The images used to test the model.

For Fig. 2A:
Training epoch = 1 ; Caption = man in red shirt is climbing rock
Training epoch = 3 ; Caption = man in red shirt is climbing large rock face
Training epoch = 5 ; Caption = man climbing up cliff
Training epoch = 7 ; Caption = man in red shirt climbing rock
Training epoch = 9 ; Caption = man climbing up cliff
Training epoch = 10 ; Caption = man grasps large rock

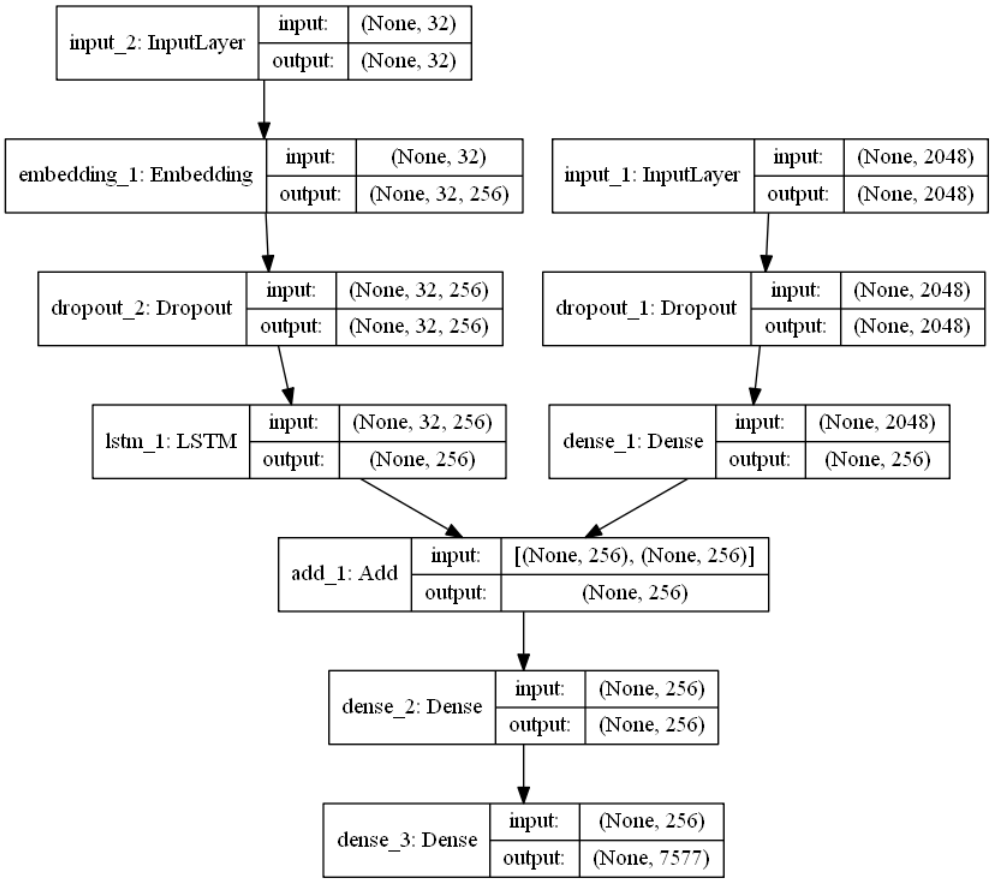

Figure 3: Original Model Architecture

Midway through the project, we implemented a baseline model 3 taken from an article [12] with a similar goal to ours, generating captions from images. The model we implemented used a pretrained convolutional neural network from Keras, Xception, to classify images based on what the image portrays [13] as can be seen in Figure 2. Then, an LSTM was used to generate a caption for the image using sequence prediction to create the caption word by word. From this model, we could see that the model was good at captioning images in the training dataset after 10 training iterations, but the accuracy of the model when used on images that are not related training dataset is very poor. This was shown using captions generated as training progressed.

In addition, the progression of loss as training went on can be seen in Fig. 5.

4 Related Work

Our project, though based originally on the paper about caption generation, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" [14], a majority of our starting code was based on the project "Learn to Build Image Caption Generator with CNN LSTM" [12]. The code that we started from used a pretrained model, Xception [13], to extract a set of feature vectors from each raw input image in the training set. Each feature vector corresponds to a specific portion of the

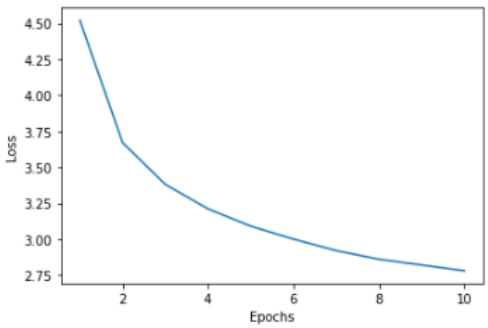

Figure 4: Training Loss Curve

2D input image. Then a Recurrent Neural Network (RNN), specifically a Long Short-Term Memory network (LSTM), was used to generate captions corresponding to the feature vectors, as a sequence of words [12]. The model as a whole was trained on the Flickr8k dataset. In our project, we made significant extensions to this model in order to improve performance.

5 Methods

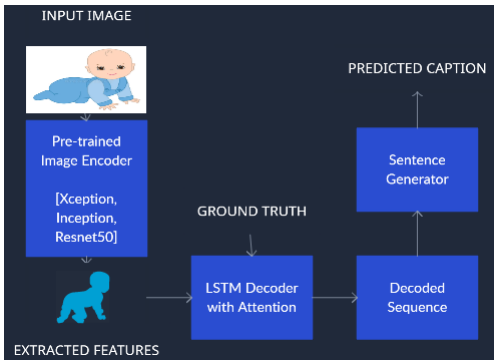

Figure 5: Caption Generation Pipeline

Though our model was based on code that we found with a similar goal [12], there were significant extensions that we implemented to improve the performance of our model. The first of this was implementing new preprocessing so that we could use a different dataset for training, Flickr30k.

5.1 Preprocessing

Preprocessing of the dataset involved 3 major steps. First, all captions were "cleaned" by converting all letters to lowercase, removing punctuation and removing words that contained numbers. Contextually irrelevant words that do not contribute much to the meaning of the captions, like "a" and " 's " were also removed. After this, a vocabulary was built, consisting of all unique words found in the descriptions of the training images. For the Flickr8k dataset, the length of vocabulary was 8763 words (for 8092 images) and for Flickr30k, the length of vocabulary was 19735 words (for 31783 images). Due to the size of the Flickr30k vocabulary, we expect a model trained on this dataset to be capable of generating more unique and detailed captions than simplistic ones.

5.2 Feature Extraction

After prepossessing of the data was done, it was run through a Convolutional Neural Network (CNN) to extract features from the images, which could in turn be used to generate captions from them. After our midway checkpoint, we implemented a new pre-trained model for feature extraction. This

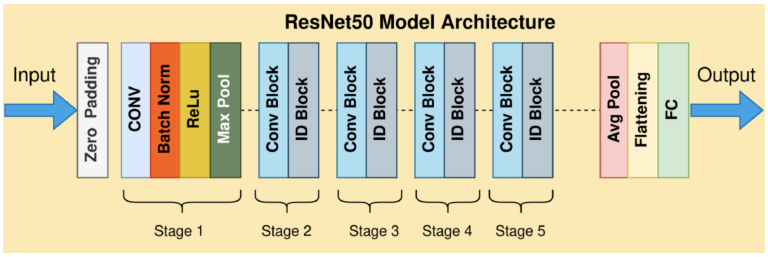

Figure 6: ResNet50 layers. Image credits:[15]

model, ResNet-50, is a 50 layer CNN made to extract a set of feature vectors from each raw input image in the training set. The CNN has 50 convolutional layers and uses identity mapping between layers to ensure that the network's accuracy does not degrade due to the vanishing gradient problem. Each feature vector will correspond to a specific portion of the 2D input image. We then use a Recurrent Neural Network (RNN), specifically a Long Short-Term Memory network (LSTM), to generate captions corresponding to the feature vectors, as a sequence of words [12]. An LSTM was used specifically to avoid the vanishing gradient problem as it has the required memory capabilities.

5.3 Attention

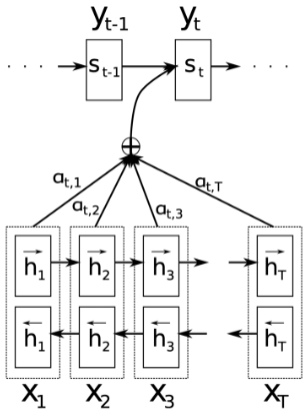

Figure 7: Bahnadau Attention Mechanism [16]

As part of our LSTM, we implemented [14], [17], [18] an attention encoder that we created to help the model focus more on relevant parts of the image during the captioning task, like people or objects. This attention module, Bahnadau Attention[19], replaces the classic fixed-length vector with a variable-length one to improve the translation performance of the basic encoder-decoder model. The decoder decides parts of the source sentence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector [16]. After important parts of the image are identified, sequence prediction is used to create captions for images word by word.

5.4 Loss

After creating the model, we trained it and calculated the loss using multi-class cross entropy loss given by the following expression:

$$\mathcal{L} = -\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right)$$

As can be seen from the batch wise loss curves in Figure 9, there was a significant decrease in the loss as our model was being trained with both of the Flickr datasets. Due to long runtime, we were

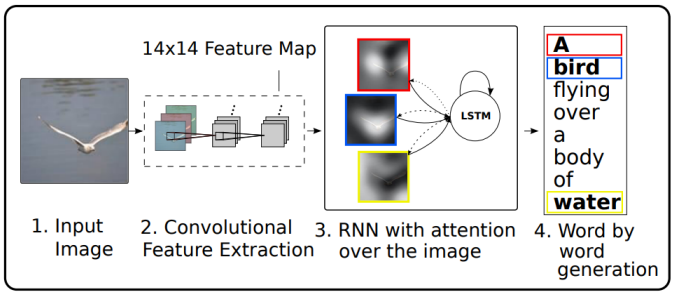

Figure 8: Attention-based Caption Generation Model Architecture. Figure taken from [14].

unable to train the Flickr 30k dataset for more than one epoch (5 batches), but the Flickr 8k dataset was trained on 10 epochs similar to the training done on the original model from our midway report. The model training loss decreased at each training epoch, similar to that of the original model.

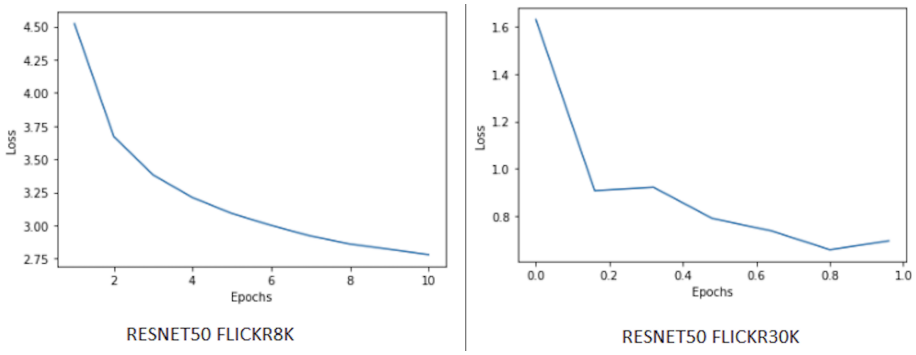

Figure 9: Final Loss Curves

6 Results

6.1 Quantitative Results

Dataset	Feature Extractor	Algorithm	Corpus BLEU BLEU 1	BLEU 2	BLEU 3	BLEU 4
Flickr8k	xCeption	Vanilla LSTM + CNN	0.01	0.15	0.05	0.01
Flickr8k	Resnet50	LSTM + CNN + Bahnadau Attention	0.01	0.15	0.04	0.01
Flickr30k	Resnet50	LSTM + CNN + Bahnadau Attention	2.09E-155	0.06	1.92E-104	2.09E-155

Table 1: Corpus BLEU Scores

The main quantitative metric we assessed our model on was the BLEU Score metric. This metric measures how well text matches with a perfect match having a score of 1.0 and a perfectly opposite section of text will give a score of 0.0 [20]. As can be seen in Table 1, BLEU 1- BLEU 4 vary in the n-grams they consider for calculation of score (1 indicating a 1-gram metric and 4 being a 4-gram metric). As the model trains, we observe gradual increase in the BLEU score which indicates the model is slowly improving as the captions get closer to the text. Corpus BLEU (1) and sentence BLEU (2) are two levels on which the BLEU is computed, but on this smaller experimentation setup they show similar scores. The BLEU Score is very low at this point, meaning that the captions are not very matched to the training data. This could be caused by the low number of training epochs run due to the limitations on computing power we faced. The second was using the BLEU, Bilingual

Dataset	Feature Extractor	Algorithm	Sentence BLEU			
			BLEU 1	BLEU 2	BLEU 3	BLEU 4
Flickr8k	xCeption	Vanilla LSTM + CNN	0.01	0.15	0.05	0.01
Flickr8k	Resnet50	LSTM + CNN + Bahnadau Attention	0.01	0.15	0.04	0.01
Flickr30k	Resnet50	LSTM + CNN + Bahnadau Attention	4.19E-186	2.09E-155	0.06	3.07E-104

Table 2: Sentence BLEU Scores

Evaluation Understudy [20], [21], score to check how well generated captions match the given captions when training. The BLEU scores can be seen in Tables 1 and 2.

The model with attention mechanism shows comparable BLEU Scores to the original model without attention, with minor differences due the fact that attention, because of its more complex nature, takes longer to train. We also note that a change in feature extractor would account for some differences in the performance. The models trained on Flickr30k were only trained for a single epoch due to the massive training time requirements and thus we note significantly poor performance. The attention mechanism in general requires longer training time which is why we could only train for shorter intervals.

6.2 Qualitative Results

Model Without Attention Mechanism:
A = man grasps large rock
B = dog is running through the water
C = man in red shirt is walking down dirt road

Model With Attention Mechanism:
A = man in red pants is rock climbing stone
B = german ash colored dog is bearing its neck of water
C = person in blue car whose front of traffic

Above, we have listed examples of captions that were generated for the images in Figure 2 by our final model with attention, trained on Flickr8k dataset. We see that with attention, our model is improved and able to identify relevant parts of the image more accurately. For example, Figure 2C was captioned "man in red shirt is walking down dirt road" by our model without attention after 10 training epoch. Upon implementation of attention, the new caption generated under similar training conditions describes the image much better, including words such as "car" and "traffic".

7 Discussion and Analysis

We faced difficulty in training our final version of the model with all extensions on the large Flickr30k dataset, due to limitations in computing resources available to us. However, we believe that our methods for improvement on the original model were successful and appropriate, resulting in improved performance of our model. We note that attention mechanisms are widely used in current encoder/decoder frameworks of image captioning, where a weighted average on encoded vectors is generated at each time step to guide the caption decoding process. But, the decoder has little idea of whether or how well the attended vector and the given attention query are related, which could make the decoder give misled results, as we observe in some of our test cases. This is an avenue for improvement in future.

Due to the significant increase in computation time needed to train the model, both due to the added attention and the larger dataset, we had to limit the training of our final model. As is evident from our results, especially if the assumption that both training and testing error would decrease after a second training epoch holds true, using a different pretrained CNN for feature extraction, using attention to direct captions to important parts of images, and using a broader dataset with more

types of training images all contributed to making our current model able to caption a wider variety of images with more accurate and detailed captions. Our final model was an improvement on the initial model we started with, but significant improvements to performance could still be made. For further improvements, it may be good to optimize runtime by looking into other, faster models of attention, a more efficient LSTM algorithm, or a different broad but less populated dataset to train on. Improvements on runtime will allow for more training to be done in a reasonable amount of time, giving better results and allowing us to see the true capabilities of our model.

8 Teammates and work division

We decided to assign one extension per person which they successfully accomplished, and then integrated to create the final project.

Malavika was tasked with implementing a different dataset to train on. This was an important extension towards robustness because using a different larger dataset introduced diversity into the learning phase. A different dataset (Flickr30k) also necessitated different pre-processing and integration into the current implementation.

Shamika was tasked with changing the feature extraction method. She introduced the Resnet-50 feature extraction module to the caption generation model. This extension was explored as it could lead to the incorporation of informative features which could improve the model's accuracy and efficiency.

Aboli was tasked with implementing the attention mechanism into the model with the goal to improve the model performance, especially for images where multiple captions could technically be correct despite there being a clear idea of what the caption is supposed to be. Upon successful implementation, this mechanism was responsible for the model to be able to identify relevant parts of the image more accurately like car and traffic.

Aboli also handled quantitative evaluation of the model using evaluation metrics like BLEU scoring and Malavika has done qualitative analysis of the model performance and a large part of the core architecture design.

9 Supplementary Materials

Google Drive Link for Video:
https://drive.google.com/file/d/14wECkBJBHCLycl7kUHapR0ptE7F0w201/view?usp=sharing

Google Drive Folder for Code:
https://drive.google.com/drive/folders/1IedddkN0Bp1Uv6KfDZcQMy8LGHZ6P9N$?usp = sharing$

References

[1] H. Fang, S. Gupta, F. Iandola, *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.

[2] C. Zhang, J. Platt, and P. Viola, "Multiple instance boosting for object detection," *Advances in neural information processing systems*, vol. 18, 2005.

[3] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[4] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, Makuhari, vol. 2, 2010, pp. 1045–1048.

[5] B. Dzmitry and B. Yoshua, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2015.

[6] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] L. Minh-Thang, P. Hieu, and D. Christopher, "Effective approaches to attention-based neural machine translation (2015)," *arXiv preprint arXiv:1508.04025*,

[8] M. Cornia, L. Baraldi, H. Rezazadegan Tavakoli, and R. Cucchiara, "A unified cycle-consistent neural model for text and image retrieval," *Multimedia Tools and Applications*, vol. 79, Sep. 2020. DOI: 10.1007/s11042-020-09251-4.

[9] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk," in *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk*, 2010, pp. 139–147.

[10] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *TACL*, vol. 2, pp. 67–78, 2014.

[11] P. Young, J. Hockenmaier, A. Lai, and M. Hodosh. [Online]. Available: http://hockenmaier.cs.illinois.edu/DenotationGraph/.

[12] D. Team, *Python based project - learn to build image caption generator with cnn and lstm*, Aug. 2020. [Online]. Available: https://data-flair.training/blogs/python-based-project-image-caption-generator-cnn/.

[13] F. Chollet *et al.* "Keras." (2015), [Online]. Available: https://github.com/fchollet/keras.

[14] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.

[15] Gorlapraveen123, *Resnet50*, File: ResNet50.png, 2021. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/9/98/ResNet50.png.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[17] K. Doshi, *Image captions with attention in tensorflow, step-by-step*, Apr. 2021. [Online]. Available: https://ketanhdoshi.github.io/Image-Caption-Attn/.

[18] *Image captioning with visual attention nbsp;: nbsp; tensorflow core*. [Online]. Available: https://www.tensorflow.org/tutorials/text/image_captioning.

[19] S. Cristina, *The bahdanau attention mechanism*, Nov. 2022. [Online]. Available: https://machinelearningmastery.com/the-bahdanau-attention-mechanism/.

[20] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[21] J. Brownlee, *A gentle introduction to calculating the bleu score for text in python*, Dec. 2019. [Online]. Available: https://machinelearningmastery.com/calculate-bleu-score-for-text-python/.