A COMPARISON BETWEEN VGG16 AND XCEPTION MODELS USED AS ENCODERS FOR IMAGE CAPTIONING

Asrar Almogbil, Amjad Alghamdi, Arwa Alsahli, Jawaher Alotaibi, Razan Alajlan and Fadiah Alghamdi

Department of Computer Science, college of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

ABSTRACT

Image captioning is an intriguing topic in Natural Language Processing (NLP) and Computer Vision (CV). The present state of image captioning models allows it to be utilized for valuable tasks, but it demands a lot of computational power and storage memory space. Despite this problem's importance, only a few studies have looked into models' comparison in order to prepare them for use on mobile devices. Furthermore, most of these studies focus on the decoder part in an encoder-decoder architecture, usually the encoder takes up the majority of the space. This study provides a brief overview of image captioning advancements over the last five years and illustrate the prevalent techniques in image captioning and summarize the results. This research study also discussed the commonly used models, the VGG16 and Xception, while using the Long short-term memory (LSTM) for the text generation. Further, the study was conducted on the Flickr8k dataset.

KEYWORDS

Image Captioning, Encoder-Decoder Framework, VGG16, Xception, LSTM.

1. INTRODUCTION

One of the most challenging and important topics in computer vision and natural language processing is image captioning [1], [2]. Image captioning aims to generate a natural language description based on the association between the objects in the given image. Image captioning can be helpful in different applications such as human-computer interaction and providing help for visually impaired persons [3]. Therefore, several studies have developed an image captioning model [4,5]. Initially, the studies related to image captioning were focused mainly on generating natural language descriptions for video [6], following the studies describing neural caption generation architectures [7, 8], such as the encoder-decoder architectures proposed in [9]. Recently, the encoder-decode architecture has shown much improved outcomes in efficiently generating natural language descriptions of an image [10]. At first, the CNN layers are used to extract the features of the image. Then the collected features are used by the Recurrent neural network (RNN) model to attain the information from the image [11].

This study reviews the current advancement of image captioning models and summarizes the underlying framework. Although much attention has been paid to the decoder, there has not been enough focus on the encoder. To fill this gap, this study will compare the performance of two

David C. Wyld et al. (Eds): SIPP, NLPCL, BIGML, SOEN, AISC, NCWMC, CCSIT - 2022
pp. 185-195, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121316

different encoder models, namely: VGG16 and Xception. Moreover, a comprising that focus mainly on the performance of two widely used encoder - VGG16 and Xception is poorly investigated, which will help further researchers to decide on the encoder model.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 discusses the materials and methods used in this work. Experiments done are described in Section 4. The result obtained is illustrated in section 5. Conclusions and future work in Section 6.

2. RELATED WORK

In this section, we will summarize multiple related studies from different sources. The studies will be organized in chronological order ascendingly. The purpose of the related work is to gain an understanding of the published studies relevant to the image captioning field.

In [12], they used the MSCOCO dataset and LSTM to encode the text and used CNN as an image encoder to extract features, and they obtained the best result compared with their benchmark. Another study [13] used VGG16 as an encoder, which aids in creating image encodings. Then, the encoded images are fed into an LSTM. The proposed model was enhanced with hyper-modifying parameters. As a result, the model's accuracy increased to attain state-of-the-art results. In [14], different models of image captioning were used. A merge architecture was applied in this study. CNN-5, vgg16, and vgg19 are the different CNN that are used along with the LSTM. The experiment is done on Flickr8K dataset. A Bilingual Evaluation Understudy (BLEU) evaluation metric is used to evaluate the models. The result showed that VGG16 is perform better than other models. The authors in [15] compared different models of image captioning. All models were conducted on the Flickr8K dataset. The architecture used in this study is encoder-decoder architecture. For the encoder, two different CNN models are used, which are VGG16 and InceptionV3. For the decoder part, two types of LSTM were used. The first type is a unidirectional LSTM that works in one direction. The second type is bidirectional LSTM which works in two directions. The proposed models used greedy search and beam search algorithms to generate the captions. The results show that the InceptionV3 with bidirectional LSTM with beam search gave the best result. The evaluation metric used is BLEU. In [16], the study proposed an image caption generator in the Bengali language using a merged dataset of two languages by combining flickr8k, BanglaLkey, and Bornon datasets. The transform-based and visual attention approaches were used to implement the proposed model. The Transform-based approach used an inceptionV3 encoder and fed to a dense layer that contains an activation function. The visual attention approach implements an Encoder-decoder framework as well. In the encoder part, the InceptionV3 and Xception models were used. For the evaluation of the proposed model, the BLEU and Metor were used.

In [17], the study proposed an image captioning model to use the model on any website to generate the description of the inputted image. The proposed model followed the CNN-LSTM concepts and was conducted on the flicker8k dataset. In [18], the study used CNN and RNN models, and the Xception was trained using the flickr8k dataset. Another study used the xception model coupled with LSTM in [19] to discover the object found in the image, detect the relationship among the objects, and generate the proper captions. This study was trained using the fliker8k dataset. The criteria to evaluate the model was the loss value. In [20], the authors compared the most popular CNN architecture: Xception, Resnet50, InseptionV3, Vgg16, and Densent201. Along with the LSTM decoder. The comparison was done to see the effect of the performance by implementing different encoder models. The study used flicker8K dataset. The evaluation of the comparison was the loss value and the accuracy to compare the model's performance. The study [21] proposed different CNN models VGG16, Xception, and inception coupled with bi-directional layer RNN models for an enhanced image captioning model. The

models were trained using flicker30K and coco datasets. The BLUE score and training and loss are used to evaluate each model.

3. MATERIALS AND METHODS

This section includes the description of the dataset used in the study and the different encoders: VGG16 and Xception. Finally, the decoder model.

3.1. Dataset pre-processing

The dataset used in this work is Flicker8k, and it is available on GitHub [22]. Flicker8k consists of two folders, the first folder contains only images, and the second folder contains a text file with the image descriptions. For the data pre-processing phase, we start working on the text file and organize it by mapping the image ID to a list of five corresponding descriptions. After that, we worked on data cleaning by making all letters in lower case, removing all the punctuations, and removing words with one character (e.g. 'A'). Lastly, we saved all changes made in a new text file.

3.2. The Encoder models

3.2.1. VGG16 model

VGG16 is one of the most preferred CNN models as it has a very uniform architecture. Simonyan and Zisserman developed this model in 2014 [23]. It contains 16 convolutional layers. By having this amount of layers, the complexity would increase compared to the initial versions of the CNN architecture. In the below Figures, the size is proportionally getting reduced. The two layers are convolutional, and the output of these two layers is 224x224, followed by the max-pooling layer, and the final output after the max-pooling layer of size 2x2 and stride of 2 will be reduced to 112x112. Finally, we have three fully connected layers called dense. Figure 1 shows the architecture of the VGG16 model.

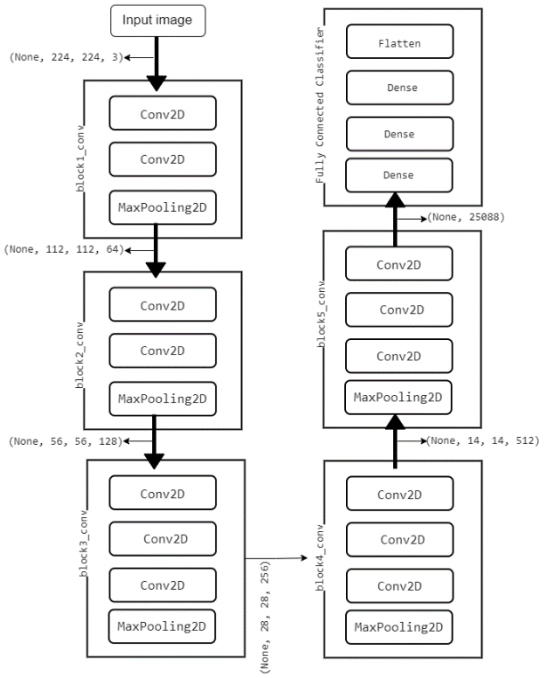


Figure 1. VGG16 Architecture

3.2.2. Xception model

The Xception model, also called “Extreme Inception” was proposed by Francois Chollet. It is a kind of CNN model used to extract the features from the image. Also, it is an extension of the inception model that is also considered a type of CNN model [24], but a better and enhanced version by reversing some steps to be more efficient and easier to modify [25]. The Xception model contains 37 layers [20]. The model uses the depthwise separable convolutional layers approach, which divides the image into K input channel with depth equal to 1, then applies the filter into each part with depth equal to 1, after that compressed all input channels space then applying 1*1 convolutional. The accuracy of the Xception model considers the highest among the CNN model in agreement with the LR in [15]. Therefore, it gives the best result compared to the other CNN models. Figure 2 illustrate the layers of the Xception model.

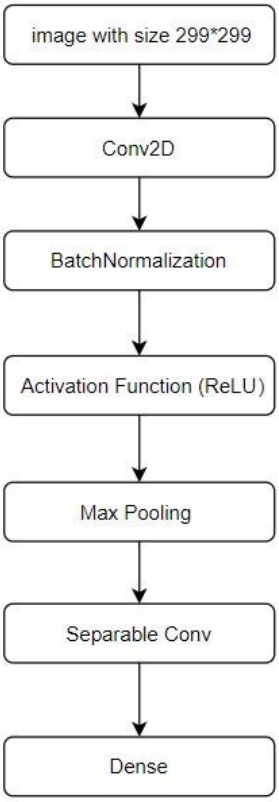


Figure 2. Layers of Xception Model

3.3. The Decoder model

For the decoder model, LSTM based model was used, which takes input from the feature extraction model to predict a sequence of words, called the caption.

Because LSTM overcomes the RNN's constraints, LSTM is more effective and superior to the regular RNN. With a forget gate, LSTM can keep relevant information throughout the processing of inputs while discarding non-relevant information. It can process not only single data points but also complete data sequences [26].

4. EXPERIMENTAL

For the experiments, our model follows the encoder-decoder framework. Therefore, we tested and evaluated two different encoder models. Furthermore, we illustrated the conducted processes for developing the models for each model and how we trained the models. Whereas the decoder remains fixed during the experiment, as mentioned before, in order to focus on comparing the performance of the encoder model.

4.1. The encoder

In the feature extraction step, the size of the image features is 224x224. Extracting the features of the image is done before the last layer. The goal of the last layer is to predict the classification of an image. For this reason, the last layer is dropped. The models were trained on Flickr8k dataset as was described in Section 3.

4.1.1. VGG16

• Before optimization

When we started the model's training, we split the dataset into two parts. The first part is for training, and the second part is for testing. Flicker8k dataset contains a file named "Flickr_8k.trainImages.txt" that includes 6000 image ID; this file is used for the training part. The training phase will be done in three steps. The first step, load the features extracted from the VGG16 model. In the second step, we will initiate a dictionary that contains descriptions for each image. The third step, create tokenizing vocabulary by using Keras, which provides the tokenizer class, and it can do the mapping from the loaded description data. In this step, we need to fit the tokenizer given the loaded photo description text. The create_tokenizer() function is responsible for fitting the created tokenizer given the loaded photo description text. In addition, it's for mapping each word of vocabulary with a unique index value.

• After optimization

To optimize the result and reduce the loss obtained, we implement Adam algorithm, which is an optimizer that increase efficiency of neural network weights.

4.1.2. Xception

• Before optimization

Our CNN-RNN model consists of three main parts: feature extraction (encoder), sequence processor, and decoder. In the experiment, we used images with a size equal to 299x299. In the features extraction step, which is done before the last layer of the model, we got an 8091 feature vector. In training, feature extraction is loaded to the model, and the dataset is divided into two parts: training with 7091 images and testing with 1000 images. Then, we tokenized the vocabulary by mapping each word with a unique index value, and each image will have a maximum length of sentence equal to 31. After that, we created a data generator to train the model to yield the image in batches.

• After optimization

The Adam algorithm was implemented to optimize the model to improve its performance.

5. RESULT AND DISCUSSION

In this study, a total of four models were tested and evaluated —VGG16, VGG16 with optimization, Xception, and Xception with optimization. The criteria for the comparison are taken to be the loss instead of the accuracy value, and the standard metric for comparison used here is the BLEU score.

Table 1. Evaluation Table

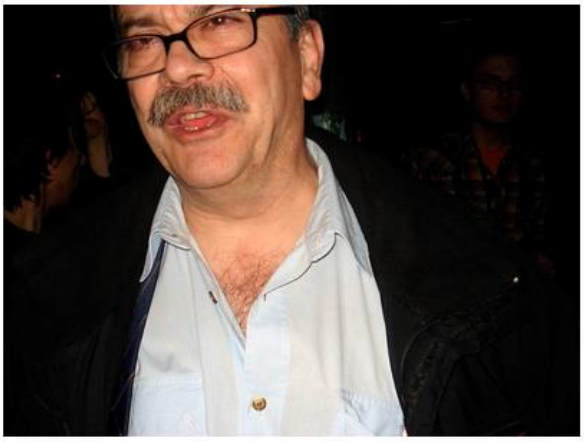
Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
VGG16 Epoch= 100 Loss=3.0345	0.522997	0.279958	0.186401	0.079141
VGG16 with optimization Epoch = 100 Loss= 3.3746 Optimizer= Adam	0.498937	0.251331	0.168155	0.068864
Xception Epoch= 50 Loss= 4.3955	0.096406	0.031889	0.020180	0.004638
Xception with optimization Epoch= 50 Loss=3.3618 Optimizer= Adam	0.550791	0.309441	0.216791	0.105341

The above table shows each model's performance in terms of the BLEU score, testing loss of the implemented models, and the number of epoch with the optimizer if used.

Our results demonstrate that Xception with optimization BLEU scores outperformed the other three models. The highest BLEU score achieved in the study was 0.550791. Both Xception with optimization and VGG16 before optimization have similar scores. However, the loss of VGG16 was less than Xception with optimization. The main motivation for using the adam algorithm was to show a significant improvement in the runtime and memory consumption and increase the efficiency of neural network weights, as mentioned in the previous section. The caption generated from the Xception with optimization model gives the best probability and more accurate captions (see Figure 6). In contrast, the captions generated by the other three models (Figure 3-5) were long sentences compared to Xception with optimization. We can infer from the experiment that when the sentences are long, the more probable to make mistakes. In most situations, we found that the short sentences are sufficient to explain an image, whereas lengthier sentences frequently contain duplicate information and grammatical errors. The main challenge was to reduce the loss in Xception models, and after using the optimizer, the loss decreased. Yet, it remained higher than the loss obtained in VGG16 before optimization (see figure 7). Hence, we observed that when the number of the Epoch is increased, the number of loss models will increase in the Xception models due to the small size of the dataset.

startseq two men are playing soccer on field endseq

Figure 3. VGG16 Before Optimization

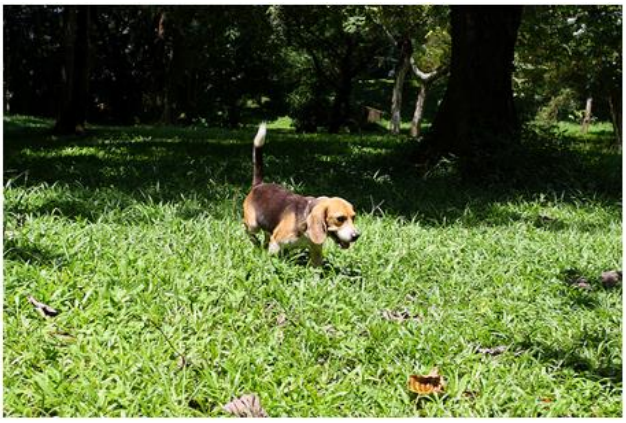


startseq man in black shirt and black pants is sitting on the street endseq

Figure 4. VGG16 After Optimization

the man is sitting on the top of the rock

Figure 5. Xception Before Optimization



startseq dog is running through the grass endseq

Figure 6. Xception After Optimization

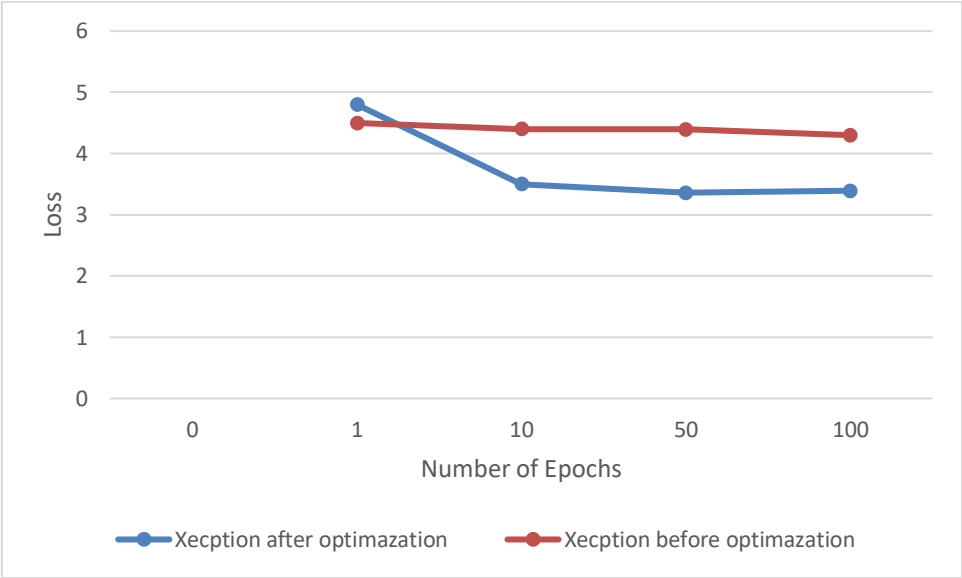


Figure 7. Testing Loss Curve for Xception Before and After Optimization.

6. CONCLUSION

In this study, we used an encoder-decoder framework that been used in the previous studies. We evaluated two different encoder models for the purpose of comparing the VGG16 and Xception encoder models. So far, no study has been published comparing these two models which will help researchers figure out which model is outperforming the other. The outcome of the comparison shows that the Xception model, when implemented adam algorithm, will generate the most accurate caption compared to the other three models. Moreover, the study attempted to use Flickr8k open-source datasets. Despite the precise caption achieved, there is still a need for a larger dataset. A large dataset will enhance the model's performance.

ACKNOWLEDGEMENTS

We would like to thank Ms. Asrar Almogbil for her cooperation on providing the instructions. We also extend our appreciation to Dr. Nida Aslam and Ms. Abrar Alotaibi for their continuous efforts in helping and answering our questions during the experiment.

REFERENCES

[1] Raimonda Staniut¯ e and Dmitrij ˙ Se˘ sok. A systematic literature review ˘ on image captioning. Applied Sciences, 9(10):2024, 2019.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In Proceedings of the IEEE international conference on computer vision, pages 2425–2433, 2015.

[3] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In ´ Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 326–335, 2017.

[4] A. Ramisa, F. Yan, F. Moreno-Noguer, and K. Mikolajczyk, "Breakingnews: Article annotation by image and text processing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 5, pp. 1072–1085, 2018.

[5] H. Ben, Y. Pan, Y. Li et al., “Unpaired image captioning with semantic-constrained self-learning,” IEEE Transactions on Multimedia, vol. 24, pp. 904–916, 2021.
[6] Vinyals, Oriol, et al. ”Show and tell: A neural image caption generator.” Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015.
[7] M. Tanti, A. Gatt, and K. P. Camilleri, “What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?,” Aug. 2017.
[8] Sulabh Katiyar and Samir Kumar Borgohain. Comparative evaluation of cnn architectures for image caption generation. arXiv preprint arXiv:2102.11506, 2021.
[9] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. ”Sequence to sequence learning with neural networks.” In Advances in neural information processing systems, pp. 3104-3112. 2014.
[10] F. Huang, X. Zhang, Z. Zhao, and Z. Li, “Bi-directional spatial-semantic attention networks for image-text matching,” IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 2008–2020, 2019.
[11] S. Li, Z. Tao, K. Li, and Y. Fu, “Visual to text: Survey of image and video captioning,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, no. 4, pp. 297–312, 2019.
[12] A. T. S. B. a. D. E. Oriol Vinyals, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLEGENCE, vol. 39, p. 12, 2017
[13] V. V. P. M. M. Ashish Pateria, "Enhanced Image Capturing using CNN," International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 4, p. 6, 2019.
[14] A. a. D. S. a. Y. M. V. Jmail, "IMAGE CAPTIONING: TRANSFORMING SIGHT INTO SCENE," International Research Journal of Modernization in Engineering Technology and Science, vol. 02, no. 06, pp. 54-66, 2020.
[15] S. Takkar, A. Jain, and P. Adlakha, “Comparative Study of Different Image Captioning Models.” Fifth International Conference on Computing Methodologies and Communication, India, 2021.
[16] F. M. Shah, M. Humaira, M. A. R. K. Jim, A. S. Ami and S. Paul, "BORNON: BENGALI IMAGE CAPTIONING WITH TRANSFORMER-BASED DEEP LEARNING APPROACH," arXiv, p. 20, 2021.
[17] M. M. Patil, "Experiment based on Deep Learning: Image," INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS - IJCRT , vol. 9, no. 12, p. 6, 2021.
[18] N. L. C. K. Satyabrat Mandal, "Automatic Image Caption Generation System," International Journal of Innovative Science and Research Technology, vol. 6, no. 6, p. 4, 2021.
[19] V. U. G. S. V. M. Megha J Panicker, "Image Caption Generator," 2021.
[20] S. R. Sahrial Alam, "Comparison of Different CNN Model used as Encoders for Image Captioning," 2021.
[21] A. P. Yash Indulkar, "Comparative Study for Neural Image Caption Generation Using Different Transfer Learning Along with Diverse Beam Search & Bi-Directional RNN," 2021.
[22] The dataset available in: https://github.com/goodwillyoga/Flickr8k_dataset
[23] A. D. Hussam, "Compressed residual-VGG16 CNN model for big data places image recognition," in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, 2018.
[24] M. j. Panicke, V. Upadhayay, G. Sethi and . V. Mathur, "Image Caption Generator," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 10, no. 3, p. 6, 2021.
[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
[26] P. G. Shambharkar, P. Kumari, P. Yadav, and R. Kumar, “Generating Caption for Image using Beam Search and Analyzation with Unsupervised Image Captioning Algorithm.” Fifth International Conference on Intelligent Computing and Control Systems, India, 2021.