Proceedings of the 8th North American International Conference on Industrial Engineering and Operations Management, Houston, Texas, USA, June 13-16, 2023

Convolutional Neural Network Architectures Analysis for Image Captioning

Jun Seung Woo, Shin Dong Ho,
Student and Professor, MY PAUL SCHOOL
12-11, Dowontongmi-gil, Cheongcheon-myeon, Goesan-gun,
Chungcheongbuk-do, Republic of Korea
eavatar@hanmail.net

Abstract

The Image Captioning models with the Attention method have developed significantly compared to previous models, but it is still unsatisfactory in recognizing images. The early Image Captioning models were built by combining CNN as an encoder and RNN as a decoder, making them susceptible to the influence of each CNN and RNN model. In particular, the CNN network has shown performance differences over time, which affects the RNN model used as a decoder. In this paper, we experiment with various CNN architectures to improve the performance of image captioning based on the CNN architecture as a reference. We analyze the performance of Image Captioning based on the performance of various CNN architecture models. We compared seven different CNN Architectures, according to Batch size, using public benchmarks: MS-COCO datasets. All CNN architectures used in this study are pre-trained networks on the ImageNet dataset. In our experimental results, DenseNet (Huang et al. 2017) and InceptionV3 (Szegedy et al. 2016) got the most satisfactory result among the seven CNN architectures after training 50 epochs on GPU.

Keywords
Deep Learning, Computer Vision, Image Captioning, CNN and DenseNet

1. Introduction
Image Captioning task, which automatically generates natural language descriptions of images, is among the most useful technologies, such as providing more detailed explanations to the visually impaired and explaining various situations that the driver may not fully perceive while driving. However, Image Captioning should be able to subjectively express the relationship between objects that appear in an image. This also requires a computer vision system that can accurately recognize various images and a high level of natural language processing capability to express the descriptions in the natural language we use.

Image Captioning is one of the most important topics in artificial intelligence fields because it connects two major topics: Convolutional Neural Networks(CNN) for computer vision and Recurrent Neural Networks (RNN) for natural language processing. One intuitive way to improve Image Captioning performance is 1) Use high-performing CNN and RNN networks. 2) Use a good combination of CNN and RNN networks. In this paper, we applied various advanced CNN architectures that have been developed consistently over time to Image Captioning experiments as the 1) and analyzed each of their performance.

In this paper, we experiment with seven interesting pre-trained CNN architectures on the ImageNet dataset, including DenseNet (Huang et al. 2017) and InceptionV3 (Szegedy et al. 2016) to improve Image Captioning performance. Our experimental results DenseNet and InceptionV3 got the lowest Loss among the seven CNN architectures.

2. Related Work
In recent Image Captioning studies such as (Vinyals et al. 2015)(Herdade et al. 2019), a specific model structure was designed for Image Captioning. The structure of Image Captioning can be divided into two categories: CNN+RNN and CNN+Transformer. In this paper, we utilized the CNN+RNN structure used by (Vinyals et al. 2015). In this paper, the author was inspired by the existing translation model in the aforementioned article and devised a similar model structure. While the existing translation model receives text as input and translates it into another text, the model

structure proposed in this paper takes an image as input, extracts its features using pre-trained Convolutional Neural Networks(CNN), compresses them into a fixed vector, and generates a textual description using Recurrent Neural Networks(RNN). Specifically, Long Short-Term Memory(LSTM) (Sak et al. 2014), which improves the issue of RNN, is employed for sentence generation. RNN Networks suffer from a long-term dependence problem, where previous information tends to disappear as the model becomes deeper due to the lack of long-term memories. However, LSTM leverages both short-term and long-term memories to update and store critical information that should not be forgotten, leading to better long-term dependence and more accurate sentence generation. In other words, 1) The given image is fed into the pre-trained CNN Encoder input, 2) The feature-map is generated, and 3) The LSTM Decoder input receives the textual description, as illustrated in Figure 1.

Recent studies on Image Captioning, such as (Xu et al. 2015), have utilized the Attention method to address issues with fixed-size vector output from pre-trained CNN Encoder. The use of fixed-size vectors causes performance degradation and bottleneck problems, as it fails to account for the size and characteristics of the image. To overcome these issues, (Xu et al. 2015) applied the Attention method to focus on the relevant characteristics of the input image and generate accurate sentences. The Attention method evaluates which part of the image should be used to generate a particular word in the sentence and assigns a weight accordingly. This weight is then taken into consideration when generating the word, allowing the model to produce a sentence that reflects the unique features of the image. For example, when generating the word "apple", the part of the image that corresponds to an apple would be given a high weight, and this weight would be incorporated by the LSTM Decoder to increase the probability of generating the word "apple". Thus, the Attention method utilizes these weights to generate more precise descriptions of images by refining the output from the LSTM Decoder.

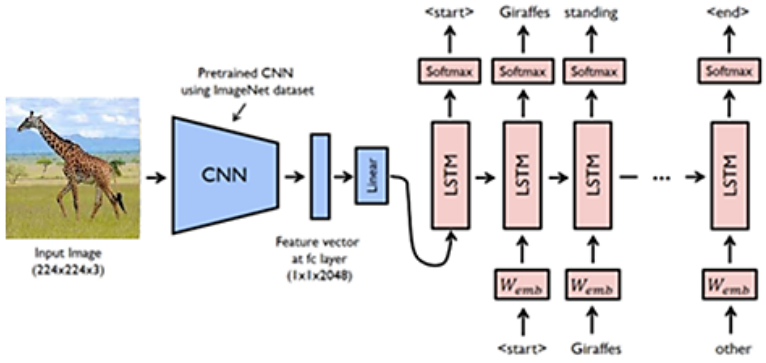

Figure 1. Model structure of (Vinyals et al. 2015)

3. Method

In this paper, we propose an approach to improve Image Captioning through the use of advanced CNN architectures, building upon the work presented by (Xu et al. 2015). Specifically, we explore seven pre-trained CNN architectures trained on the ImageNet dataset, with a focus on identifying the architectures that yield the best performance for Image Captioning. Our experimental results demonstrate that DenseNet (Huang et al. 2017) and InceptionV3(Szegedy et al. 2016) outperform the other CNN architectures. Additionally, all seven CNN architectures have less computation and higher accuracy than the VGGnet (Simonyan et al. 2014) used in (Xu et al. 2015). All experiments were conducted using the TensorFlow implementation.

3.1 DenseNet

In recent CNN architectures, the vanishing-gradient problem has become a major issue, in which the gradient is not updated during the backpropagation process and the gradient is gradually reduced as the depth of the network increases. To overcome this challenge, many CNN architectures proposed solutions to build deeper layers while mitigating this vanishing-gradient problem. ResNet (He et al. 2016) and Highway Networks (Srivastava et al. 2015) resolved this issue by training through identity connections from one layer to the next. Similarly, DenseNet also addressed the vanishing/exploding gradient problem by using identity connections.

As illustrated in Figure 2, DenseNet connects the feature-map output of each layer to all subsequent layers. The authors of this paper identified a common feature of this approach, which creates a short path from the input to each layer. Building on this concept, they designed a model in which each layer connects to all other layers. This process helps alleviate the vanishing-gradient problem by preserving the feature-map of each layer without mixing information from previous and current layers. Additionally, DenseNet requires fewer parameters and computations compared to previous CNN architectures, as it avoids the need to retrain duplicate feature-maps. This advantage enables DenseNet to achieve good results in computationally demanding tasks such as Image Captioning.

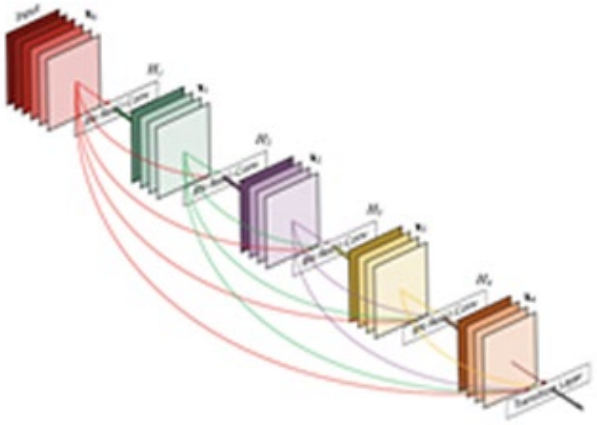

Figure 2. Identity connections of DenseNet (Huang et al. 2017)

3.2 InceptionV3

InceptionV3 (Szegedy et al. 2016) is a model that simplifies the structure of the Inception (Szegedy et al. 2015) mod el using Convolution Factorization. The author divided the 5 × 5 convolution into two 3 × 3 convolutions, where the operation amount of the 5 × 5 convolution is 25, while the 3 × 3 convolution has an operation amount of 9. Even if t wo 3 × 3 convolutions are used, the operation amount is 18, which is better than using 5 × 5 while maintaining simil ar accuracy. Additionally, general convolution calculated in the form of N × N is divided into 1 × N and N × 1 to red uce computation. InceptionV3 also employs Batch Normalization (Ioffe et al. 2015), Label Smoothing(Müller et al. 2019), and other techniques for improved performance. Due to its ability to effectively reduce computation, Inceptio nV3 has achieved satisfactory results in this Image Captioning study that requires a lot of computation.

3.3 Model Architecture

Our paper adopts the framework proposed by (Xu et al. 2015) that incorporates the Attention method in Image Captioning. Specifically, we employ seven pre-trained CNN architectures, including DenseNet (Huang et al. 2017) and InceptionV3 (Szegedy et al. 2016), as the CNN Encoder. Our Image Captioning model extracts image features through the CNN Encoder and uses an LSTM Decoder to generate a sentence. Notably, the CNN Encoder utilizes seven CNN architectures, including DenseNet and InceptionV3, that are less computationally demanding than VGGnet. The output of the CNN Encoder is then processed by the LSTM Decoder, which applies the Attention method to determine which part of the image is more relevant and which word should be generated based on the Attention scores, as illustrated in Figure 3.

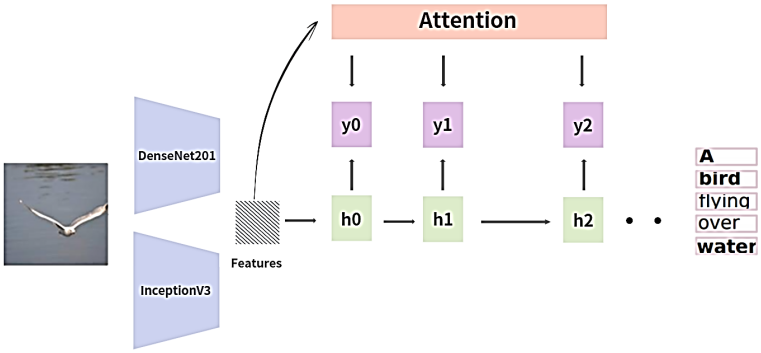

Figure 3. The structure of our Image Captioning model. In the CNN Encoder, seven CNN architectures, including DenseNet and InceptionV3, were utilized. The symbol "h" represents hidden layers, and "y" represents attention scores.

4. Experiments

In this paper, we compared seven pre-trained CNN architectures on the ImageNet, namely EfficientNetB7 (Tan et al. 2019), mobileNetV3Large (Howard et al. 2019), ResNet152(He et al. 2016), mobileNetV3Small (Howard et al. 2019), InceptionResNetV2 (Szegedy et al. 2017), DenseNet201(Huang et al. 2017), and InceptionV3 (Szegedy et al. 2015). We implemented these architectures using Keras Applications. The training dataset consisted of 82,000 images from the MS-COCO dataset, each containing at least 5 captions. We used Sparse Category Cross-entropy as the loss function and Adam as the optimizer for model training. The models were trained and evaluated 50 times using the total number of training epochs. We designed experiments based on batch size and compared the results with Table 1, which showed that batch size had a significant impact on loss and training time. Based on our findings, DenseNet and InceptionV3 showed a loss of about 15% and a training time of about 160 seconds per epoch. Therefore, among the seven CNN architectures experimented with in this paper, DenseNet and InceptionV3 showed relatively satisfactory results. Interestingly, the other five CNN architectures showed unstable training losses and irregular fluctuations depending on the batch size. In particular, CNN architectures like EfficientNetB7, which have recently shown high performance, may have performed poorly in the Image Captioning experiment due to the potential mismatch between the CNN encoder and RNN decoder mentioned in the introduction.

Finally, we applied our Image Captioning model using InceptionV3 (Szegedy et al. 2015) to specific images, as shown in Figure 4. We confirmed that the generated sentences were of good quality, and by outputting each word using the Attention method, it was possible to determine in detail which part of the image was used as a reference.

Table 1. Compared to the other five CNN architectures, InceptionV3 (Szegedy et al. 2015) and DenseNet201 (Huang et al. 2017) showed better Loss and Training epoch(sec). As a result of experimentation with Batch sizes, we also showed that the 128 batch size generally has better accuracy and learning time than the other various batch size.

Networks	Batch_Size=64		Batch_Size=128	
	Loss Top-1	Time(sec) Training	Loss Top-1	Time(sec) Training
EfficientNetB7	45.54%	176.19	125.66%	143.03
mobileNetV3Large	41.06%	112.16	53.00%	58.52
ResNet152	19.41%	206.24	17.27%	164.44
mobileNetV3Small	52.51%	62.66	58.74%	152.76
InceptionResNetV2	14.30%	178.17	16.59%	163.68
InceptionV3	**14.45%**	**165.59**	**13.98%**	**124.95**
DenseNet201	**14.28%**	**172.50**	**12.97%**	**142.85**

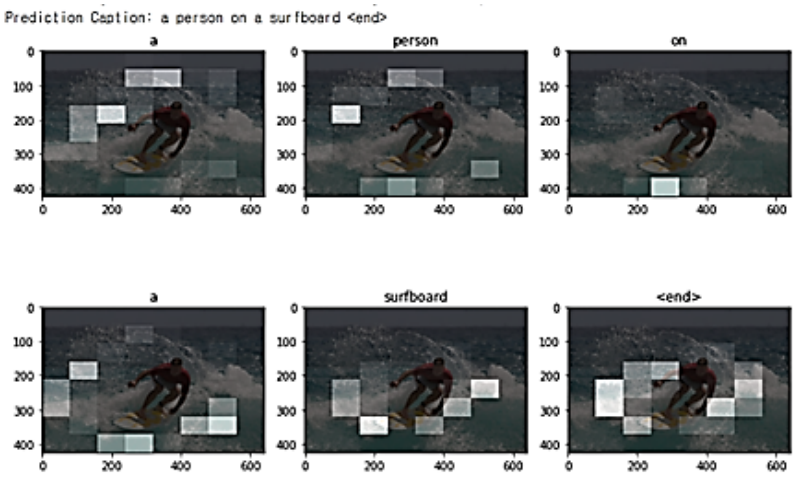

Figure 4. Image Captioning result using InceptionV3 (Szegedy et al. 2015)

5. Conclusion

In this paper, we experimented with seven pre-trained CNN architectures on the ImageNet, to improve the CNN encoder for Image Captioning. Among them, DenseNet, and InceptionV3 showed the best results with a loss of about 15% and a training time of around 160 seconds per epoch in the MS-COCO dataset. Interestingly, the other five CNN architectures, which performed similarly well to DenseNet and InceptionV3, exhibited unstable training losses and irregular fluctuations depending on the batch size. We argue that this may be due to poor compatibility between certain CNN encoders and the LSTM decoder used in our Image Captioning model.

References

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

Herdade, S., Kappeler, A., Boakye, K. and Soares, J, Image captioning: Transforming objects into words. *Advances in neural information processing systems*, 32, 2019.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., and Bengio, Y, Show, attend and tell: Neural image caption generation with visual attention. *In International conference on machine learning*, pp. 2048-2057, 2015.

Sak, H., Senior, A. W., and Beaufays, F, *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*, 2014.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q, Densely connected convolutional networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708, 2017.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z, Rethinking the inception architecture for computer vision. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826, 2016.

Simonyan, K., and Zisserman, A, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:*1409.1556, 2014.

He, K., Zhang, X., Ren, S. and Sun, J, Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 770-778, 2016.

Srivastava, R. K., Greff, K., and Schmidhuber, J, *Highway networks*, arXiv preprint arXiv:1505.00387, 2015.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., and Rabinovich, A, Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015.

Ioffe, S., and Szegedy, C, Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In International conference on machine learning*, pp. 448-456, Pmlr, 2015.

Müller, R., Kornblith, S., and Hinton, G. E, When does label smoothing help?. *Advances in neural information processing systems,* 32, 2019.

Tan, M., and Le, Q, Efficientnet: Rethinking model scaling for convolutional neural networks. *In International conference on machine learning*, pp. 6105-6114, PMLR, 2019.

Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., and Adam, H, Searching for mobilenetv3. *In Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314-1324, 2019.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A, Inception-v4, inception-resnet and the impact of residual

connections on learning. *In Proceedings of the AAAI conference on artificial intelligence*, Vol. 31, No. 1, 2017.

Biographies

Jun Seung Woo is student in MY PAUL SCHOOL. He is interested in artificial intelligence, deep learning, cryptography, block chains, autonomous vehicles, etc., and is conducting related research.

Shin Dong Ho is Professor and Teacher in MY PAUL SCHOOL. He obtained his Ph.D. in semiconductor physics in 2000. He is interested in artificial intelligence, deep learning, cryptography, robots, block chains, drones, autonomous vehicles, mechanical engineering, the Internet of Things, metaverse, virtual reality, and space science, and is conducting related research.