Comparative Analysis on Image Caption Generator using Deep Learning techniques

Aditya Biswakarma
IIT2020033, B.Tech IT
Indian Institute of Information Technology
Prayagraj, India
iit2020033@iiita.ac.in

Harshit Kushwah
IIT2020039, B.Tech IT
Indian Institute of Information Technology
Prayagraj, India
iit2020043@iiita.ac.in

Sanjeet
IIT202052, B.Tech IT
Indian Institute of Information Technology
Prayagraj, India
iit2020104@iiita.ac.in

Rohan Tirkey
IIT2020088, B.Tech IT
Indian Institute of Information Technology
Prayagraj, India
iit2020088@iiita.ac.in

***Abstract*—Image captioning is the process of writing narratives to go along with an image's occurrences. Virtual assistants, editing software, picture indexing, and assistance for those with impairments can all benefit from image captioning, which is an essential task. It connects computer vision with natural language processing, the two main branches of artificial intelligence. Encoder-decoder frameworks are frequently the foundation of more recent methods for picture captioning. Encoders frequently employ large convolutional neural networks that have already been trained. However, different authors' image captioning models employ various encoder designs. Because of this, it is more challenging to ascertain how the encoder affects the performance of the model as a whole. Because of this, figuring out how the encoder affects the performance of the model as a whole is more challenging. In this paper, we have done a comparative study between two popular convolution networks architectures – VGG and ResNet –as encoders for the identical image captioning model to determine which approach is the most effective at representing images used to generate captions. Based on this data we can determine how big the encoder plays and how significantly it can improve the model without changing a decoder architecture.**

***Index Terms*—image captioning, encoder-decoder framework, convolutional neural networks, VGG, ResNet,BLeU**

I. Introduction

To create human-like descriptions of images, the work of image captioning combines computer vision and natural language processing. It has several useful uses, including helping the blind, strengthening search engines, and making digital media more accessible.

Encoding and decoding are the two phases that make up the image's captioning process. From the input image, the encoder extracts significant features, and the decoder uses these elements to create a sentence that explains the image. The capacity of convolutional neural networks (CNNs) to extract high-level information from images makes them a popular choice for encoders in image captioning systems.

The CNN architectures that can be utilised for image captioning are numerous. VGG and ResNet are two popular architectural types. Due to the network's extensive usage of tiny 3x3 convolutional filters, VGG is distinguished by its simplicity. The vanishing gradient problem, which happens when gradients in deep neural networks get too small to be effective for weight updates, is a challenge that ResNet is well known for being able to solve. Skip connections, which enable gradients to flow straight from earlier layers to later layers and allow for the successful training of much deeper networks, are a feature of the ResNet architecture.

Despite VGG and ResNet's and other computer vision algorithms' successes, it is still necessary to investigate how well they perform when used for image captioning. Other CNN architectures, including Inception, have also been suggested and have demonstrated promising performance in image classification applications. Therefore, it's crucial to evaluate how well various CNN architectures perform when captioning images in order to assess their relative merits and shortcomings. This knowledge will advance the creation of successful image captioning systems and aid in the selection of a suitable model for particular use cases.

II. Models

A. VGG19:

VGG19 is a convolutional neural network that has 19 layers, including 16 convolutional layers and three fully connected layers. One of the main features of the VGG19 architecture is the use of small convolutional filters. The convolutional layers in VGG19 use filters of size 3x3, which are stacked on top of each other. This allows the network to learn more complex features in the input image, as the filters can be combined to create more sophisticated representations of the data. The use of small filters also reduces the number of parameters in the network, which helps to reduce overfitting.

Another feature of the VGG19 architecture is the use of max pooling. Max pooling is a down-sampling operation that reduces the spatial dimension of the feature maps. Max pooling is applied after some of the convolutional layers in VGG19, with a 2x2 pixel window and a stride of 2. This operation helps to reduce the computational cost of the network and prevent overfitting.

The fully connected layers in VGG19 have 4096 channels each, except for the final layer, which has 1000 channels corresponding to the number of output classes in the ImageNet dataset. The output of the final layer is passed through a softmax function to produce the probability distribution over the output classes.

B. VGG16:

VGG-16 is a convolutional neural network that is 16 layers deep.VGG-19 consists of 19 layers, including convolutional layers, max-pooling layers, and fully connected layers. It follows a sequential structure with convolutional layers stacked one after another.VGG-19 takes RGB images of size 224x224 pixels as inputs.The network's filters are 3x3 convolutional filters with a 1 pixel stride and 1 pixel padding. The rectified linear unit (ReLU) activation function, which introduces non-linearity, comes after each of the 16 convolutional layers in the VGG-19. After every 2 convolution layer a max-pooling layer with a 2x2 window size and a stride of 2 pixels is applied.

C. ResNet-50:

ResNet-50 is a convolutional neural network that is 50 layers deep. ResNet-50 starts with a 7x7 convolutional layer with a stride of 2, followed by a max-pooling layer with a 3x3 window and a stride of 2. It includes a total of 16 convolutional layers, divided into different blocks.

ResNet-50 is made up of several residual blocks that are responsible for the task of leaning residual mappings. Multiple convolutional layers, batch normalisation, and ReLU activations make up each residual block.

D. LSTM:

LSTM, or Long Short-Term Memory, is a type of recurrent neural network (RNN) architecture that is capable of handling sequence data such as speech, text, and time series data. It was designed to overcome the vanishing gradient problem that occurs in traditional RNNs, which can make it difficult for the model to learn long-term dependencies in the data.

The architecture of an LSTM network consists of repeating blocks of memory cells and gates that control the flow of information into and out of the cells. The memory cells are responsible for maintaining a "memory" of the past inputs and outputs, while the gates regulate how much information is stored in the cells and how much is discarded.

Each block contains three gates: the input gate, the forget gate, and the output gate. The input gate controls how much new information is added to the memory cell, while the forget gate decides what information should be discarded from the cell. The output gate controls how much of the memory cell is used to generate the output at each time step.

The input and forget gates are controlled by sigmoid activation functions, which produce values between 0 and 1. A value of 0 indicates that the gate is closed and no information is passed through, while a value of 1 indicates that the gate is open and all information is passed through. The output gate is controlled by a hyperbolic tangent (tanh) activation function, which produces values between -1 and 1 and determines the strength of the output signal.

III. METHODOLOGY

A. Dataset

We will evaluate our method on the Flickr8k dataset,which is widely used as a benchmark dataset for image captioning tasks. It includes approximately 8,000 photographs pulled from the Flickr website along with several captions that were written by humans. These photos include people, animals, objects, landscapes, and a variety of other subjects and settings. The pictures range in complexity, size, and topic.

Five distinct captions are linked to each image in the collection, offering a variety of descriptions for the same picture. Different human annotators contributed the captions, therefore there are differences in style, language, and concentration. The many captions for each picture are meant to show various viewpoints and perceptions.

The dataset contains supplementary annotation files in addition to the picture files and caption texts. These files include metadata like caption IDs, picture IDs, and file names for the accompanying images. During the training, assessment, and testing phases, they serve as a reference for tying photos to their corresponding descriptions. Out of this 1600 samples were chosen which were trained for 100 epochs with batch size 64 and the split of the dataset is as follows: 85% for training and 15% testing.For model selection and hyperparameter tweaking, a smaller piece of the dataset known as the validation set is employed. The chosen model is ultimately evaluated using the testing set. It acts as a standard against which to compare the generalization and effectiveness of the trained model on untrained images. The human-produced reference captions are compared to the captions generated by the model for the test images, and evaluation metrics such the BLEU score are computed to assess the captions' quality and accuracy.

B. Evaluation

In tasks involving natural language processing, the metric known as BLEU (Bilingual Evaluation Understudy) is frequently used to assess the fineness of automatically generated translations or captions. It measures the degree of similarity

between a machine-generated candidate translation and one or more human-provided reference translations.

Given a single hypothesis sentence and multiple reference sentences, it returns a value between 0 and 1. The metric close to 1 means that the two are very similar. The primary programming task for a BLEU implementer is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position-independent. The more matches, the better the candidate translation is.

N-gram precision: BLEU assesses how accurately the candidate translation matches the reference translations in terms of n-grams. It records how many n-grams from the candidate translation can be found in any of the reference translations. Modified precision: A modified form of precision is employed to prevent favoring extremely brief translations. It counts the most instances of each n-gram in each reference translation while excluding instances when the n-gram appears in the candidate translation.

N-gram scores cumulatively: BLEU computes the modified precision for several values of n (usually 1 to 4) and then adds them together using a geometric mean. This promotes the automatic translation to match longer phrases as well as individual words. . Brevity penalty:Translations that are much shorter than the reference translations are subject to a brevity penalty under BLEU. In order to prevent favoring extremely brief translations, this penalty is used.

C. Implementation Details

Preprocessing The most common words are articles such as ”a”, or ”the”, or punctuations.These words do not have much information about the data. In order to clean the caption, we have created three functions that perform the following task: remove punctuation remove single character remove numeric characters Performing these three tasks significantly reduces the vocabulary size. After this we add start and end sequence tokens to the texts in caption. The final preprocessing function includes pad sequences function from Keras which is used to ensure that the input sequences have a fixed length. After processing all the caption-image pairs, the lists are converted into numpy arrays

Feature extraction We have used different models such as VGG16,VGG19 and Resnet architecture for comparative study. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. As we are using these models just for extracting features not for classification we will remove the last layer from the network.

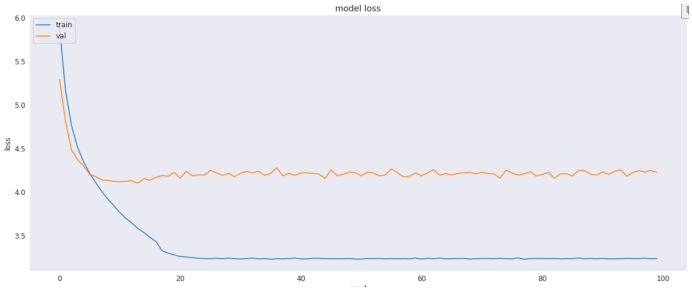
Tokenizer is used to change character vectors to integer vectors.

Prediction The model can predict the t+1st word in the caption given the caption prediction up to the t th word. The input caption may then be supplemented with the predicted word to contain the caption up to the t+1st word.

To predict the t+2 nd word in the caption, one can utilize the augmented caption up to the t+1 st word as input. Up till the ”end seq” is anticipated, the procedure is repeated.

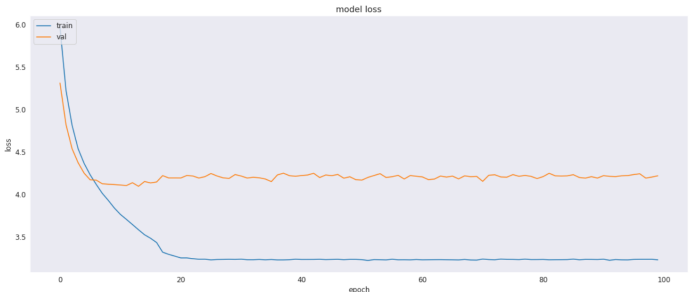
IV. **RESULT**

VGG19



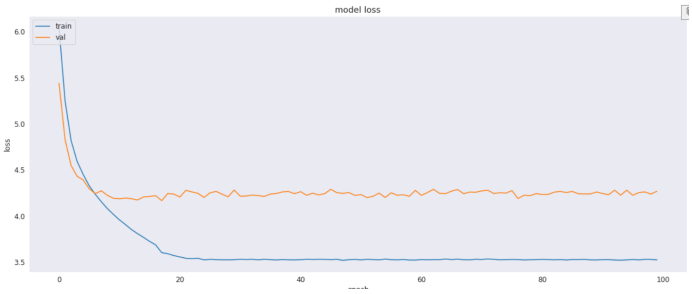
Mean BLEU value: 0.61
Min BLEU value: 0.1887535121456652
Max BLEU value: 0.8931539818068694

VGG16



Mean BLEU value: 0.57
Min BLEU value: 0.2202568481652174
Max BLEU value: 0.8931539818068694

ResNet50



Mean BLEU value: 0.61
Min BLEU value: 0.19694347419590436
Max BLEU value: 0.8931539818068694

V. Outputs

A subset of test data was to predict the captions. The example outputs are:

VI. Conclusion

In conclusion, our comparison of picture caption generators trained on ResNet50, VGG16, and VGG19 has shed light on the advantages and disadvantages of each architecture for this task. It's important to keep in mind that the models were only trained on a limited dataset of 1600 samples from the Flickr8k dataset, restricting the generalizability of our findings, even if the results showed that ResNet50 had the highest BLEU score of 0.61.

It's also important to note that the models were trained using the Adam optimizer and categorical cross-entropy loss function across 100 epochs with a 64 batch size. We admit that more hyperparameter optimisation could have produced even better results, despite the fact that these hyperparameters have a considerable impact on the model's performance.

Overall, our results show the possibility of utilising several CNN architectures for image captioning tasks, and we think that our comparative analysis might serve as a jumping-off point for more research in this area. Future research could examine how well these architectures perform on bigger datasets with other hyperparameters and loss functions. Image captioning technology may be further enhanced through research and development, which will help to produce more inclusive and accessible digital media.

VII. References

- Viktar Atliha, Dmitrij Se˘ sok. "Comparison of VGG and ResNet used as Encoders for Image Captioning" Proceedings of 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences.
- Aneja, Jyoti, Aditya Deshpande, and Alexander G. Schwing. "Convolutional image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- Cornia, Marcella, et al. "Meshed-memory transformer for image captioning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
- R. Staniut¯ e and D. ˙ Se˘ sok, "A systematic literature review on image ˘ captioning," Applied Sciences, vol. 9, no. 10, p. 2024, 2019