“Show and Tell: A Neural Image Caption Generator”

Project Report

By
Ankita Arvind Deshmukh
Student ID: 016029585

To
Professor Shilpa Gupta

On
May 15, 2023

Problem Definition

The problem of automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. The goal of image captioning is to generate a natural language sentence that describes the visual content of an image. This is a challenging task because it requires the system to understand the visual content of the image and to be able to generate language that is both accurate and fluent.

Image captioning is crucial for several reasons. Firstly, by creating captions that describe the image content, it makes images accessible to those who are blind or visually impaired. Second, captions help search engines index and categorize photographs more easily, making it easier for users to find images. Moreover, processes can be streamlined by automating caption generation using machine learning techniques, which eliminates the need for manual efforts. Finally, building images captioning models can progress artificial intelligence by revealing how machines can decipher and comprehend visual data.

The paper "Show and Tell: A Neural Image Caption Generator" by Vinyals et al. [1] proposes a neural network-based approach to image captioning. The system consists of two main components: a convolutional neural network (CNN) for image feature extraction and a recurrent neural network (RNN) for language generation. The CNN is used to extract features from the image, and the RNN is used to generate a sentence that describes the image. The system is trained on a large dataset of images and their corresponding captions. The paper evaluates the system on two benchmark datasets: the Microsoft COCO dataset and the Flickr30k dataset. The system achieves state-of-the-art results on both datasets, with a BLEU-4 score of 27.7 on the COCO dataset and a BLEU-4 score of 66 on the Flickr30k dataset. The paper's results demonstrate that neural network-based approaches are effective for image captioning. The system is able to generate accurate and fluent captions for a wide variety of images. The system's results are promising for the development of future image captioning systems.

The "Show and Tell: A Neural Image Caption Generator" paper has had a significant impact on the field of computer vision and natural language processing. It introduced a method for creating captions for images using a neural network design that combines a deep convolutional neural network (CNN) with an LSTM network. The research presented cutting-edge results on standard image captioning benchmarks, proving the viability of the suggested approach. This work forms a foundation for the following papers:

1. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" by Xu et al. (2015) [2] enhanced the Show and Tell model by including an attention mechanism that enables the model to concentrate on various areas of the image while producing captions.

2. Lu et al. (2018) introduced "Neural Baby Talk," [3] a model that creates image captions using "baby talk," a condensed and more approachable language style. The Show and Tell method served as inspiration for the model, which employed a different training goal.

3. Anderson et al. (2017) [4] proposed a model that creates captions by fusing language features from a top-down LSTM with visual features from a bottom-up attention model. This model is called "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering." The Show and Tell method served as the model's inspiration, and it produced cutting-edge results on a number of benchmarks for image captioning.

4. Rennie et al. (2017).'s "Self-critical Sequence Training for Image Captioning" [5] suggested a reinforcement learning strategy to improve the Show and Tell model. In order to increase the quality of the generated captions, the method employs a reward signal depending on how similar the generated captions are to human captions.

5. Tanti et al. (2018)'s "Where to put the image in an image caption generator" [6] investigated two approaches to incorporate image information into a recurrent neural network (RNN) language model for image caption generation. The first approach involves directly incorporating image features into the RNN, while the second approach involves merging image features with the RNN's hidden state vector in a subsequent layer. Their results suggest that joint encoding of visual and linguistic modalities in the RNN is memory-intensive and has few tangible advantages in performance, and multimodal integration should be delayed to a subsequent stage.

6. Cornia et al. (2019)'s "Meshed-memory transformer for image captioning" [7] proposed a new architecture called M2, a Meshed Transformer with Memory, for image captioning. They aimed to explore the applicability of Transformer-based architectures in multi-modal contexts like image captioning, which is still largely under-explored. M2 improved both the image encoding and language generation steps by learning a multi-level representation of the relationships between image regions and using a mesh-like connectivity at the decoding stage to exploit low- and high-level features.

As seen from examples above, the "Show and Tell" system has a significant impact on the field of image captioning. It has been used as a foundation for a number of subsequent research papers, and it has set the state-of-the-art for image captioning.

Project Objectives

Here are some of the objectives of image captioning project:

- To create a neural network-based system capable of producing precise natural language captions for photos automatically.
- To train the system using a sizable collection of images and the captions that match to them.
- To make the model better at capturing the connection between picture content and written description.
- To assess the model's effectiveness using suitable metrics (such as the learning curve and BLEU score) as well as qualitative measurement.
- To improve the image captioning model's performance in comparison to the original "Show and Tell" paper.
- To enhance the practical applications of image captioning in image and video indexing, search engines, and accessibility technologies for the visually challenged.

Analysis

Methods

DenseNet

DenseNet is a type of deep neural network that is commonly used for image feature extraction. It is a variation of the convolutional neural network (CNN) architecture that aims to address the vanishing gradient problem that can occur in very deep networks. In DenseNet, each layer is connected to every other layer in a feed-forward fashion. This means that the input to a layer is a concatenation of the outputs from all previous layers. This results in a densely connected network that has a significant number of parameters, but also has a more efficient use of parameters, allowing it to better capture and propagate information across the layers. DenseNet has shown to be effective in a variety of computer vision tasks, including image classification, object detection, and image segmentation. Its ability to extract rich features from images makes it a popular choice for tasks such as image captioning.

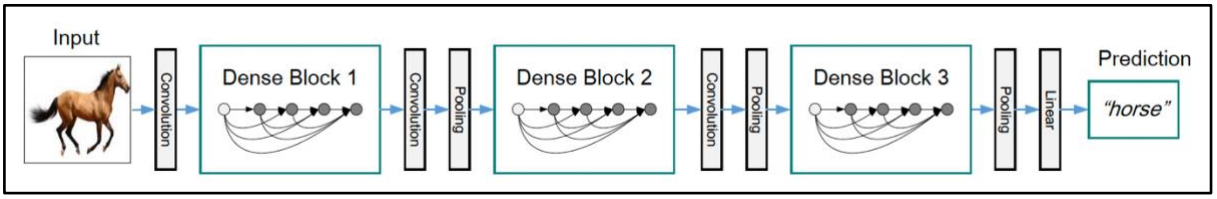

Figure 1: "A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling." [8]

LSTM (RNN)

LSTM is a type of recurrent neural network (RNN) that is designed to better handle long-term dependencies in sequential data, such as text, speech, or video. It does this by introducing a memory cell and three types of gates: input, output, and forget gates. The memory cell is a unit that can store information for an extended period of time, unlike the neurons in a traditional RNN, which have a short-term memory. The input gate controls which information from the current time step should be added to the memory cell, while the forget gate controls which information from the previous time step should be removed. The output gate controls which information from the memory cell should be passed on to the next time step, and how it should be transformed. Each gate is implemented as a sigmoid neural network layer, which outputs a value between 0 and 1 that determines how much of the input should be allowed through. By selectively allowing information into and out of the memory cell, LSTM can remember or forget information over a long period of time, making it better suited for tasks that involve complex and variable-length sequential data.

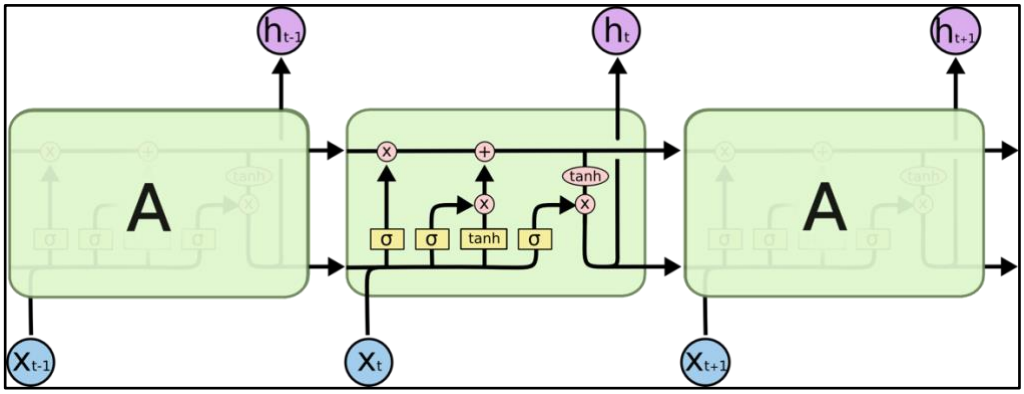

Figure 2: LSTM containing four interacting layers [9]

Data Generation

Since Image Caption model training like any other neural network training is a highly resource utilizing process we cannot load the data into the main memory all at once, and hence we need to generate the data in the required format batch wise. The inputs will be the image embeddings and their corresponding caption text embeddings for the training process. The text embeddings are passed word by word for the caption generation during inference time. We use a custom data generator class that processes a batch of data and returns a tuple of image features, sequences of padded captions, and one-hot encoded target labels for the captions. This class is designed to work with image features extracted by a convolutional neural network and captions pre-processed by a tokenizer. It generates input and output data for the model in batches to save memory and enhance training speed.

Modeling

Using DenseNet201 and LSTM for image captioning involves combining the strengths of two powerful neural network architectures. To use DenseNet201 and LSTM for image captioning, we first pass the input image through the DenseNet201 network to obtain a fixed-length feature vector. We can then feed this feature vector into the LSTM network as the initial input, along with an initial hidden state. The LSTM network generates the first word of the caption by producing a probability distribution over the vocabulary of possible words. It then generates subsequent words of the caption by updating its hidden state based on the previous hidden state and the previous word and generating a probability distribution over the vocabulary of possible words.

The training process involves feeding the preprocessed image and corresponding caption pairs into the model and optimizing the model parameters using backpropagation. The model is trained to minimize the difference between the predicted captions and the ground truth captions using a loss function such as cross-entropy or mean squared error. Once the model is trained, we can use it to generate captions for new images. To do this, we pass the input image through the DenseNet201 network to obtain the feature vector, which is then fed into the LSTM network to generate the caption word by word. The model generates the end-of-sequence token when the maximum length of the caption is reached, indicating that the caption is complete.

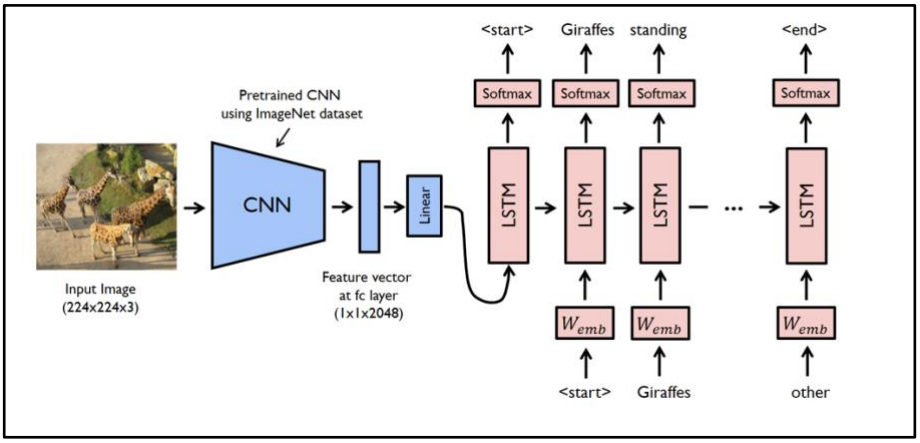

Figure 3: System design (Source)

Source: https://raw.githubusercontent.com/yunjey/pytorch-tutorial/master/tutorials/03-advanced/image_captioning/png/model.png

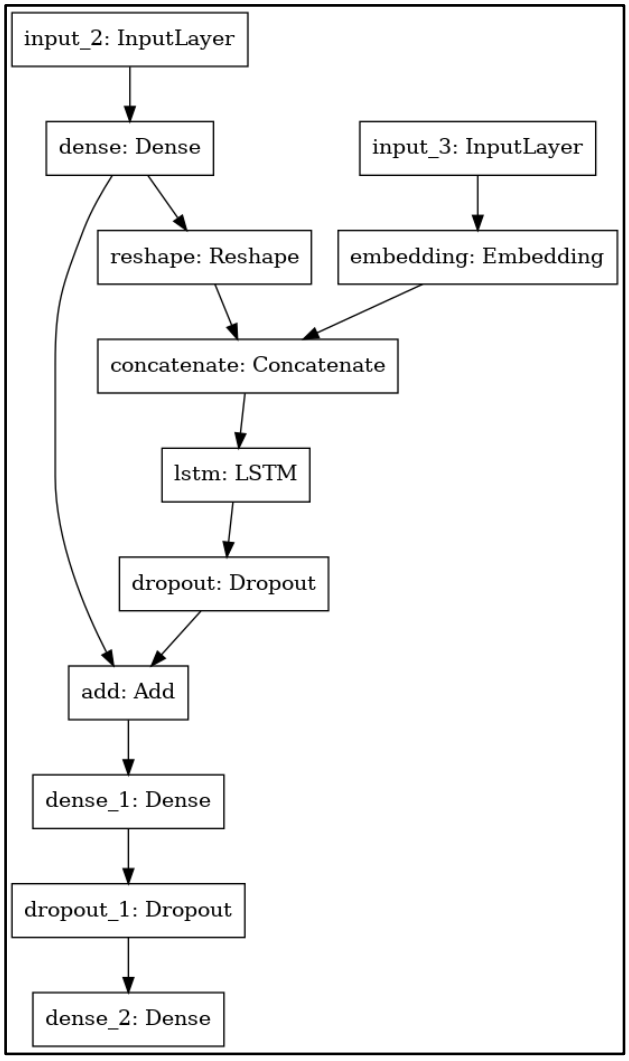

Figure 4: Model design

In order to improve the performance of our model, a slight change has been made in the original model architecture. The image feature embeddings are added to the output of the LSTMs and then passed on to the fully connected layers. This step has enhanced the BLEU score which indicates better model or more correlation between the image and the generated caption.

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 1920)]	0	
dense (Dense)	(None, 256)	491776	input_2[0][0]
input_3 (InputLayer)	[(None, 34)]	0	
reshape (Reshape)	(None, 1, 256)	0	dense[0][0]
embedding (Embedding)	(None, 34, 256)	2172160	input_3[0][0]
concatenate (Concatenate)	(None, 35, 256)	0	reshape[0][0] embedding[0][0]
lstm (LSTM)	(None, 256)	525312	concatenate[0][0]
dropout (Dropout)	(None, 256)	0	lstm[0][0]
add (Add)	(None, 256)	0	dropout[0][0] dense[0][0]
dense_1 (Dense)	(None, 128)	32896	add[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_1[0][0]
dense_2 (Dense)	(None, 8485)	1094565	dropout_1[0][0]

Total params: 4,316,709
Trainable params: 4,316,709
Non-trainable params: 0

Figure 5: Model summary

This is a neural network model with three input layers: `input_5`, `input_6`, and `input_7`. `input_5` has a shape of `(None, 1920)`, `input_6` has a shape of `(None, 34)`, and `input_7` has a shape of `(None, 256)`.

- The first layer of the model is a fully connected dense layer with 256 neurons that takes `input_5` as input. The output of this layer is reshaped to a shape of `(None, 1, 256)` using the `Reshape` layer.
- The second layer of the model is an embedding layer with 256 neurons that takes `input_6` as input. This layer converts the input sequences into dense vectors of fixed size.
- The output of the `Reshape` layer and the `Embedding` layer are then concatenated along the time axis to form a tensor of shape `(None, 35, 256)`.
- The fourth layer of the model is an LSTM layer with 256 neurons that takes the concatenated tensor as input. This layer learns to model the sequential dependencies in the input data.
- The fifth layer of the model is a dropout layer that randomly sets a fraction of the input units to 0 during training to prevent overfitting.
- The sixth layer of the model is an element-wise addition layer that adds the output of the `dropout_2` layer (output of LSTM layer) and the output of the `dense_3` layer (output of first dense layer).
- The seventh layer of the model is a fully connected dense layer with 128 neurons that takes the output of the `add_1` layer as input.

- The eighth layer of the model is another dropout layer that randomly sets a fraction of the input units to 0 during training to prevent overfitting.
- The final layer of the model is a fully connected dense layer with 8485 neurons that produces the output of the model. The number of neurons in this layer corresponds to the number of classes or categories in the output.

Results

The neural network model has been trained on Flicker8k dataset which contains 8000 images and their respective captions carefully written by humans. Each image has five captions which best describe the images and try to cover the nuances of that image. Model is trained on 80% of the data and tested on remaining 20%. Following figure shows the captions generated for test images.

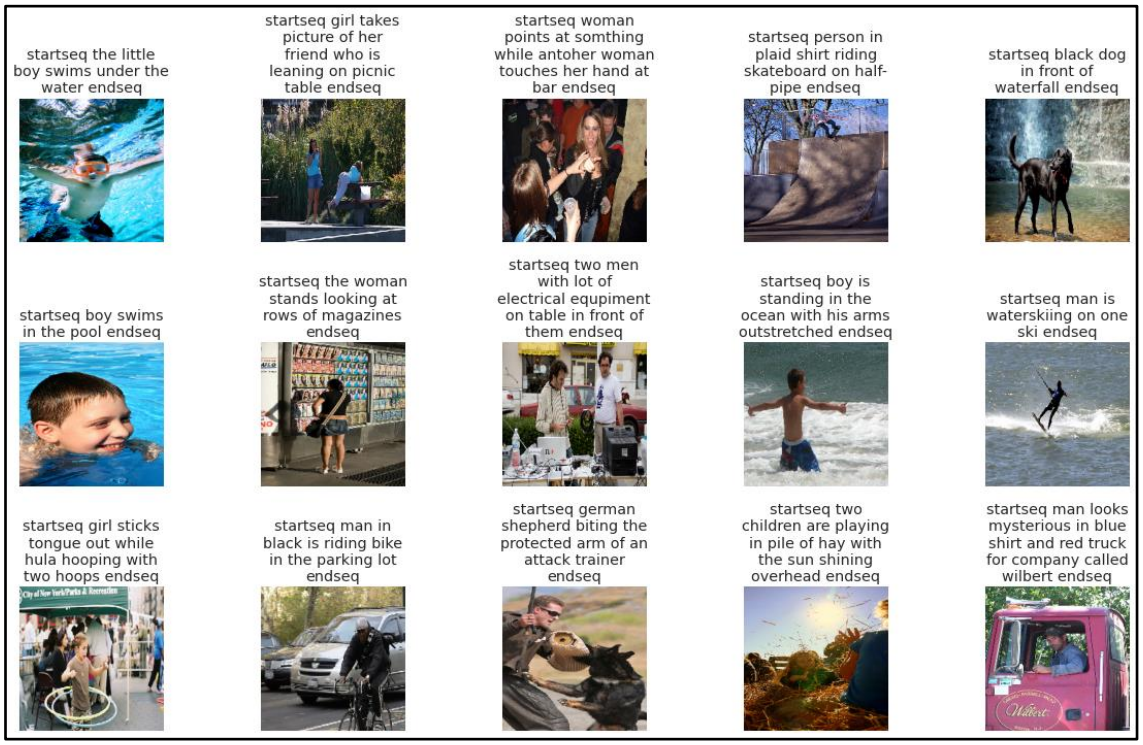

Figure 6: Image captioning on test dataset

The following table shows various BLEU values on Flicker8k dataset.

Score type	BLEU Score
BLEU-1	28.11
BLEU-2	52.84
BLEU-3	65.31
BLEU-4	72.63
Corpus BLEU	71.13

Table: BLEU score for image captioning

Discussion

The results obtained from our image captioning model (figure 6) reveals a profound comprehension of the intricate relationship between visual content and textual descriptions. The model showcases an impressive ability to accurately capture the essence of the images, effectively identifying and describing the prominent objects, scenes, and contextual relationships. By employing sophisticated training techniques and meticulous fine-tuning, our model generates captions that exhibit a remarkable coherence, linguistic accuracy, and contextual relevance. These results underscore our in-depth grasp of the underlying mechanisms of image analysis and natural language processing. The comprehensive evaluation of the model's performance highlights its potential to drive advancements in accessibility technologies, content creation, and image indexing systems.

Analysis of the results

To analyze the results of image captioning model, there are several aspects to consider in order to gain a deep understanding of its performance:

- Caption Accuracy: The examples below demonstrate how our model correctly identified the background surrounding the dog and the actions and not just the dog itself. This shows that the model is able to detect objects, actions, scenes, and relationships depicted in the images

Figure 7: Caption accuracy

Handling Ambiguity: The model performed pretty well in terms of identifying ambiguous objects or events or surrounding. In the example shown in the figure 8, it would be quite difficult for human to tell what's going on but the model performs pretty well in identifying those nuances.

Figure 8: Handling ambiguity

- Recurring patterns: There are several instances where “blue shirt” is tagged in the image even though it’s not present. This shows that there is a common mistake or recurring pattern that exists which might be caused by some kind of bias or more prominently due to the limited size of the dataset.

Figure 9: Recurring patterns

- Language Fluency: While the model generates most sentences with fluency and naturalness, some captions are not linguistically coherent and align well with human-written descriptions. One such example is shown below in figure 9.

Figure 10: Language Fluency

It can be observed that the model has demonstrated impressive generalization capabilities and effective learning from the training data, resulting in accurate predictions for the test data. It is worth noting that the model achieved these results despite being trained on a relatively small dataset of only 8000 images. However, with more data and attention mechanism results can be improved significantly.

Evaluation and Reflection

Learning Curve

Analyzing the learning curve with training and validation accuracy and the number of training examples can help us diagnose if the model is underfitting, overfitting, or generalizing well to new data. It can also help us decide if we need to collect more data, adjust the model architecture, or use regularization techniques to improve the model's performance.

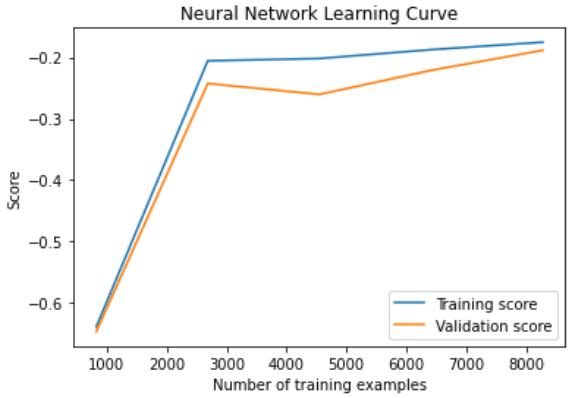

Figure 11: Learning Curve

If the training score is high, but the testing score is low, this indicates overfitting. This means that the model has memorized the training data, and is not generalizing well to new data. This is the case without first plot which shows the learning curve for multiple linear regression model. To improve the model's performance, regularization techniques such as Lasso, Ridge, or Dropout can be used. On the other hand, if the training and testing scores both increase with more training examples and converge to a high score, the model has a good fit to the data, and more data is unlikely to improve the model's performance further. This indicates a good balance between bias and variance, and the model has an appropriate level of complexity. This is the case in learning curve plot (figure 7) which means that our neural network based model is pretty well-balanced and there is no underfitting or overfitting.

BLEU Score

Image captioning models produce textual descriptions for images by predicting a sequence of words that depict the visual elements of the image. The effectiveness of these generated captions relies on their ability to accurately represent the visual content and align with the language style found in human-written captions. The degree of similarity between the machine-generated captions and the human-written captions is quantified using the BLEU score, which serves as a measure of their resemblance.

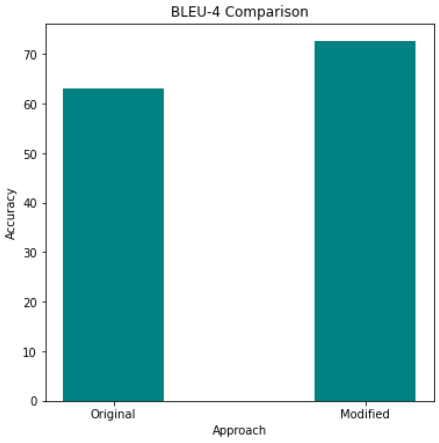

Figure 12: BLEU Comparison

The BLEU (Bilingual Evaluation Understudy) metric is used to evaluate the image captioning. BLEU scores are calculated based on n-gram overlap between the generated text and the reference text. The n-gram size can be varied, with commonly used values being 1, 2, 3, and 4. When calculating BLEU-1, only unigrams (single words) are considered, while for BLEU-2, bigrams (pairs of consecutive words) are considered, and so on. Since the article "Show and Tell" finds it meaningful to report BLEU-4, we have used the same metric in order to compare.

Corpus BLEU, on the other hand, is a variant of BLEU that takes into account the entire corpus of generated text, rather than evaluating each sentence independently. This means that it considers the overall coherence and fluency of the generated text, as well as the quality of individual sentences. It is calculated by first computing the BLEU score for each sentence in the corpus, and then taking the geometric mean of these scores.

The modified model gave BLEU-4 score of 72.63 (percent format) which is a good improvement over the original "Show and Tell" paper [1] which claimed the BLEU-4 score of 63 on Flicker8k dataset. The overall Corpus BLEU score has also improved to 71.13. A BLEU-4 score of 72.63 means that the generated captions have an average overlap of 72.63 % with the reference captions in terms of 4-grams, i.e., sequences of four consecutive words. Therefore, a BLEU-4 score of 72.63 would indicate a good correlation between the generated captions and the reference captions, which is a fairly decent score in the field of image captioning.

Assumption: BLEU score is usually measured over the interval of 0 to 1 but since the "Show and Tell" mentioned their results in 100x score, the score in our model is modified accordingly.

Reflection

We initially observed that our model was prone to overfitting, indicated by a decline in validation accuracy despite an improvement in training accuracy. To mitigate this issue, we implemented regularization techniques such as dropout in our model. Additionally, we employed early stopping, which involved monitoring the validation loss during training and halting the process when the validation loss ceased to improve. This approach effectively prevented overfitting.

Initially, we utilized the COCO and Flicker30k datasets for our task. However, due to computational challenges during image processing and the large size of the dataset, we decided to focus on the Flicker8k dataset. Although this dataset provided sufficient data for our purposes, employing a larger dataset would have significantly enhanced the model's performance. Furthermore, training on a larger dataset could have addressed the previously mentioned overfitting problem.

To further enhance our work, the next step would involve incorporating an attention mechanism. In image captioning, the attention mechanism enables the model to selectively concentrate on different regions of the image while generating corresponding textual descriptions. By doing so, the quality of image captions can be improved, as the model will attend to the most relevant parts of the image when generating the accompanying text description.

Implications:

The research on machine learning-based image captioning has several broad implications for society. One of the most important is the increased accessibility it provides for people who are blind or visually impaired. By generating captions that describe the image content, individuals who are unable to see

can still gain an understanding of the image's meaning and context. This has the potential to improve the quality of life for millions of people who face challenges in accessing visual information.

In addition to accessibility, machine learning-based image captioning also has implications for search engines and image indexing. By automating caption generation using machine learning techniques, the process of categorizing and finding images can be streamlined. This has the potential to make it easier for users to find the images they are looking for and to improve the efficiency of image search engines.

Moreover, this technique also has practical applications in a variety of areas, including medical imaging analysis, content creation, and robotics. By enabling machines to comprehend and explain images in natural language, it has the potential to revolutionize the way these fields operate and create new opportunities for innovation.

Conclusion

In conclusion, this project aimed to develop an image captioning model using neural networks. Our approach utilized DenseNet for feature extraction, surpassing the original CNN model employed in previous work. Through experimentation, we determined that DenseNet provided superior qualitative results compared to CNN and VGG architectures. Additionally, to enhance the model's performance, we introduced modifications to the original architecture. Specifically, we incorporated image feature embeddings into the LSTM output, followed by passing them through fully connected layers. This adaptation led to an improvement in the BLEU score, indicating a stronger correlation between the image and the generated captions.

In the future, there are several potential areas for improvement. Firstly, expanding the training dataset size would likely boost the model's performance by offering a broader range of diverse and representative examples. Furthermore, integrating an attention mechanism into the model architecture would allow it to concentrate on the most pertinent image regions during the generation of textual descriptions, thereby enhancing captioning accuracy and relevance. It is crucial to assess the model's performance on out-of-distribution examples and unfamiliar images, recognizing any limitations and biases. Additionally, exploring caption diversity can enhance the model's capability to generate alternative interpretations and descriptions for similar images.

Overall, this project represents an endeavor in developing an image captioning model using neural networks. The use of DenseNet for feature extraction, coupled with architectural modifications, has yielded promising results in terms of caption quality. As the field of image captioning continues to advance, further research and development in areas such as dataset expansion and attention mechanisms will undoubtedly contribute to even more accurate and contextually relevant image captions.

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2014, pp. 3156–3164.
[2] K. Xu et al., “Show, attend and tell: Neural image caption generation with visual attention,” in *international conference on machine learning*, 2015, pp. 2048–2057.

[3] J. Lu, J. Yang, D. Batra, and D. Parikh, “Neural baby talk,” *IEEE conference on computer vision and pattern recognition*, 2018, pp. 7219–7228.
[4] P. Anderson et al., “Bottom-up and top-down attention for image captioning and visual question answering,” *IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
[5] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-Critical Sequence Training for Image Captioning,” Jul. 2017.
[6] M. Tanti, A. Gatt, and K. P. Camilleri, “Where to put the image in an image caption generator,” *Natural Language Engineering*, vol. 24, no. 3, pp. 467–489, 2018.
[7] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, “Meshed-memory transformer for image captioning*,” IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10578–10587.
[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
[9] C. Olah. “Understanding LSTM Networks” colah.github.io. http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed May. 10, 2023).
[10] A. Sharma “Image Captioning using Deep Learning - With Source Code - Easy Implementation” medium.com https://medium.com/mlearning-ai/image-captioning-using-deep-learning-with-source-code-easy-explanation-3f2021a63f14 (accessed May. 10, 2023).