Image Captioning with InceptionV3

Amin Rabinia, Tom Gause

1. Introduction

Image captioning is a useful subtask of computer vision which gained significant attention in the past few years. Image captioning, compared to other machine learning tasks, is a fairly complex task. The reason for this complexity relates to the dual nature of image captioning models, where two tasks of objection detection and text generation are combined for creating captions for the input images. Image captioning models usually entail a feature extraction model and a RNN model for sequence generation. An integration of these two models can create a sophisticated deep neural network that can be trained on image captioned datasets.

In this project, we chose a baseline model [13] for image captioning and improved this model with a better feature extraction model (InceptionV3) and some RNN modifications (with GloVe embedding). In the following sections, we describe the details of our models, report the results of evaluations and discuss the challenges of our project.

2. Background

Object recognition and text generation have been commercial capstones of machine learning. As big tech companies pour billions of dollars into data science and breakthroughs in algorithms lead to more effective and efficient models, both fields are now producing even more accurate results. These improvements have granted the emergence of a new field of vision-and-language tasks such as text-to-image retrieval, image-to-text retrieval, visual question answering, and the subject of our project, image captioning. Image captioning, as a subtask of computer vision, exploits an image processing module accompanied by a text generation algorithm, to create a short descriptive caption for any given image.

VGG-16 [9] and InceptionV3 [6] are among the most important object recognition models, trained on ImageNet. ImageNet [7] is a large visual database with more than 14 million hand-annotated images indicating objects pictured, including bounding boxes in at least 1 million of the images. This dataset includes over 20,000 categories with several hundred images each. The baseline model of our project uses VGG-16, while our improved model uses InceptionV3. More details about these two models are provided in section 4. In addition to InceptionV3, our final model, also entails GloVe 300d word embeddings [5]; a semantic language representation model pre-trained on the English Gigaword Fifth Edition, made up of approximately 26 gigabytes of newswire text data.

3. Datasets

To train our networks, we applied two popular benchmark datasets, Flickr8k [11] and Flickr30k [12]. These datasets are made up of respectively eight and thirty thousand images. In our training and evaluation, we split the 8k set into 6k train and 2k test and the 30k set into 26k train and 4k test. Each image has annotations containing at least five captions and a series of object fields. These publicly available datasets target scene understanding, capturing complex images of daily scenes. Example images with accompanying captions can be seen in Figure 1.

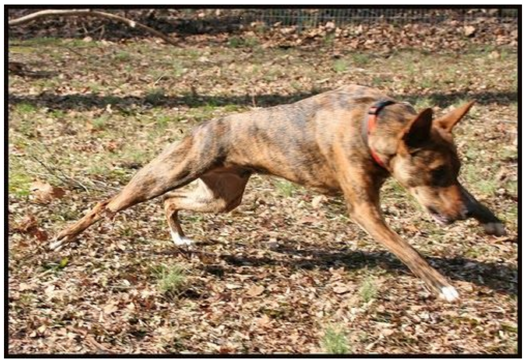
Six adults sitting at picnic table with woman in yellow sweater talks to them. Seven people are sitting at picnic table with evergreen trees in the background. Group of people are sitting together at picnic table. Young men and women sit at picnic table in park. Family sitting on picnic bench.	 **Dog with brindlecolored coat is running across the yard. Brown dog with red collar jumping across the leafy lawn. Brown and black dog runs through the leaves. The brown dog is wearing red collar. Brown dog is running.**

Figure 1. Examples from the Dataset

4. Methods (Models)

In our project we had three stages of developments: 1) implementing a baseline model with VGG-16, 2) improving the baseline with InceptionV3, and 3) improving the RNN model with GloVe and GRU. The models are developed using Keras and Tensorflow libraries. Throughout these three stages we ran our models either on Google Colab, a platform with free GPU access, or the Jupyter Notebook, on a local machine.

For our baseline, we implemented the model proposed in [13] created for educational means. This model includes the VGG-16 network with pre-trained ImageNet weights for object detection. The VGG-16 is a convolutional neural network proposed in 2014 by K. Simonyan and A. Zisserman [9], achieving a 92.7% top-5 test accuracy in ImageNet. This model takes a 224x224 RGB image as input to the first convolutional layer, where it is passed through a stack of convolutional layers using filters with a 3x3 receptive field. Figure 2 shows the architecture of the VGG-16 model.

We improved the baseline model by implementing InceptionV3. Proposed in 2015 by Szegedy et. al [16], the InceptionV3 model improves upon Google's original InceptionV1 (GoogleNet) model. This model introduces the idea of an Inception layer, a combination of 1x1, 3x3, and 5x5 Convolutional layer concatenated into a single output vector forming the input to the next stage, as seen in Figure 3. The internal layers pick and choose which filter size will be relevant to learn the required information, i.e. images with small features require a lower filter size. InceptionV3 makes a number of upgrades increasing the accuracy and reducing the computation complexity of the Inception network including factorizing 5x5 convolutions to two 3x3 convolutions, widening filter banks, implementing factorized 7x7 convolutions, and including BatchNorm in Auxiliary Classifiers. This flexibility improves the accuracy and efficiency of the model, achieving a 94.49% top-5 accuracy in ImageNet. InceptionV3 takes a 299x299 RGB image as input, and thus has greater potential for detail extraction than VGG-16.

Figure 2. VGG-16 Network Architecture

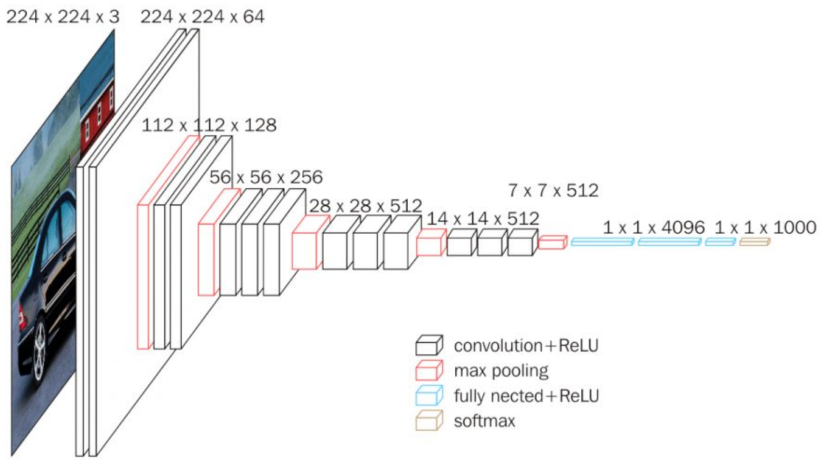
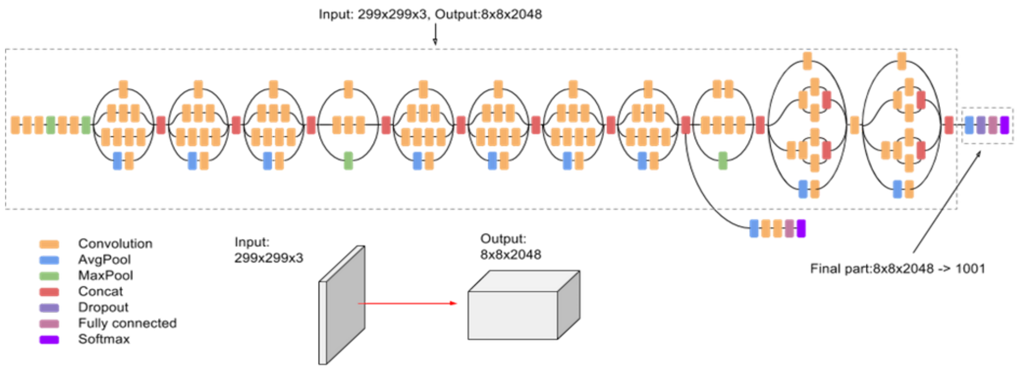


Figure 3. InceptionV3 Network Architecture



To improve our baseline, in the second stage, we studied a publicly available model created by Google for educational means which is trainable on a small subset of the MS-COCO dataset [10] (6000 images). This model implements InceptionV3 (Figure 3) with pre-trained ImageNet weights for object detection and a CNN Encoder and GRU-based RNN decoder inspired by the *Show, Attend and Tell* paper [] that introduced the effective implementation of soft and hard attention in Image Captioning models. For our customized model, we used the InceptionV3 model for features extraction of the input images.

At the last stage of the evolution of our model, we added the pre-trained contextualized GloVe 300d embedding, and implemented it in our network. For the text generation task, we also used GRU instead of LSTM, as used in the baseline model. GRU had a similar accuracy as LSTM but it is comparably faster. For a 5 epochs training LSTM spent `3647s` but with GRU it only took `2980s`.

To describe the data flow in all of these three models, we have the following steps: 1) download the data and unzip, 2) extract the features of images (with either models, VGG or Inception). Note that the last layer of the feature extraction model (softmax) is removed, so we can receive the features (hidden states) instead of the categorical classifications (what the last layer does). 3) load the captions and clean their text, 4) fit the captions to the tokenizer, 5) unpack GloVe embedding, 6) create the model, which has a sequence generation function and a few dense layers to concatenate the two inputs (images features and captions). Figures 4 and 5 illustrate the model summary for this final step. The left branch receives the textual inputs from the captions and the right branch receives the image features. After the embedding, RNN and dropout layers, the two inputs are merged in one dense layer (dimension 256) and then decoded to the final layer with the dimension of our vocabulary size (7597).

Figure 4. Baseline Model Summary

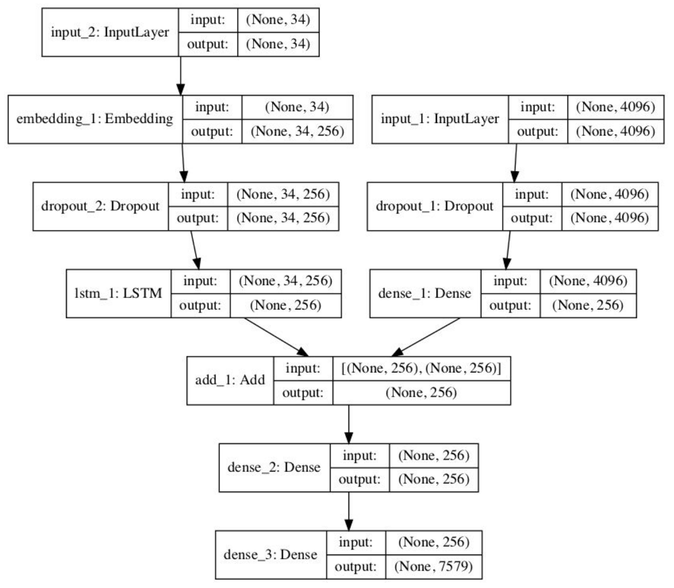
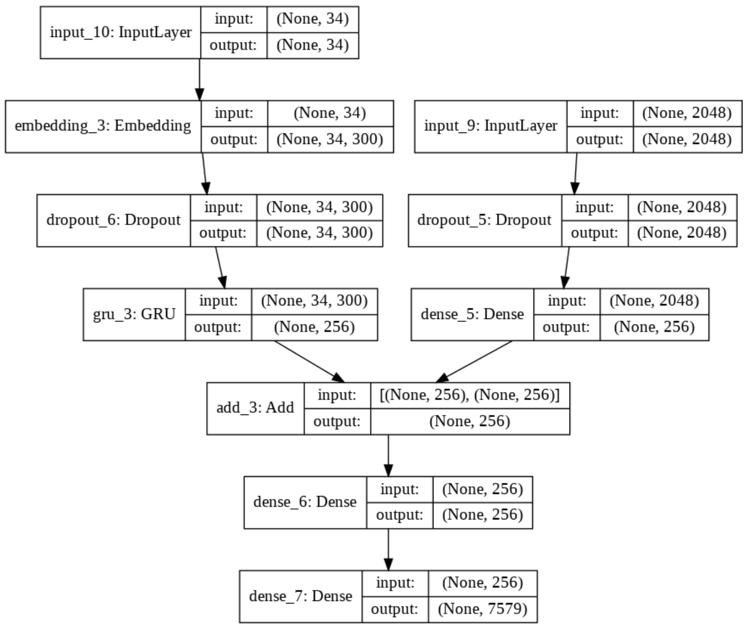

input_2: InputLayer | input: (None, 34) | output: (None, 34)
embedding_1: Embedding | input: (None, 34) | output: (None, 34, 256)
input_1: InputLayer | input: (None, 4096) | output: (None, 4096)
dropout_2: Dropout | input: (None, 34, 256) | output: (None, 34, 256)
dropout_1: Dropout | input: (None, 4096) | output: (None, 4096)
lstm_1: LSTM | input: (None, 34, 256) | output: (None, 256)
dense_1: Dense | input: (None, 4096) | output: (None, 256)
add_1: Add | input: [(None, 256), (None, 256)] | output: (None, 256)
dense_2: Dense | input: (None, 256) | output: (None, 256)
dense_3: Dense | input: (None, 256) | output: (None, 7579)

Figure 5. Augmented RNN with GloVe Model Summary



Results

We evaluated the success of our models using the Bilingual Evaluation Understudy or BLEU scores, on the Flicker dataset. This approach counts matching n-grams in the candidate translation to n-grams in the reference text, where a perfect score, i.e. an exact match, is 1.0. For perspective, the current highest BLEU score on the COCO dataset is 0.417. The loss function for our evaluation is the categorical cross-entropy.

At every stage of our project we ran the models for at least 20 epochs. We tried different optimizers, such as Adam, SGD, RMSprop, with different learning rates in the range of 1e-1 to 1e-5. Training after 20 epochs did not decrease the loss dramatically and quite reverse it started to increase the loss for more than 30 epochs. The best optimizer in our experiments was also the default Adam (lr=0.001). The lowest loss achieved on Flickr 8k dataset belongs to the improved model with GloVe with loss of 2.3 and dropout of 0.125 after 20 epochs of training. We noticed that decreasing the dropout from 0.5 to lower values results in lower loss. The main downside of decreasing the dropout is overfitting of the model, so it cannot perform as accurate on new data. Table 1, shows a summary of the results for training our three models for 5 epochs on both Flickr 8k and 30k.

Model	**Loss After 5 Epochs**	**Bleu-4 Score After 5 Epochs**
Baseline on Flickr 8k	3.3799	0.074
InceptionV3 on Flickr 8k	3.1567	*no improvement*
InceptionV3 with Glove on Flickr 8k	**2.9053**	**0.069**
Baseline on Flickr 30k	4.0411	0.080
InceptionV3 on Flickr 30k	3.9471	*no improvement*
InceptionV3 with Glove on Flickr 30k	**4.0777**	**0.071**

Table 1. Results of Training with Three Different Models on Flickr 8k and 30k

Discussion

The biggest challenge in this project was our limited access to high speed GPU and large RAM capacity. This was particularly problematic with working on larger datasets, deeper models and training for more epochs. Due to this limitation, we had a very limited capability for fine-tuning our model or exploring other helpful strategies for improving our model.

Another challenge in our project relates to overfitting, where the model performs well in the training phase but falls short in tests with new data. Overfitting has multiple causes: small dataset, low dropout and longer training (more than 10 epochs). Due to the computational limits we could not use very large datasets. Higher dropout will also result in higher loss and underfitting. Therefore, we only tried to limit the training to the acceptable range of 5 to 10 epochs.

Related Work

Show and Tell [1] is a very influential work in the image captioning field. We aimed to choose this approach as our baseline model, but due to the complexity of the model and our limited computational resources this option did not seem realistic. Another educational work described in [2] helped us learn about the image captioning task, its available datasets and the existing models. CutMix [3] is an augmentation method for image captioning that has been tested on MS-COCO. Augmentation techniques can improve the performance of feature extraction models and result in more accurate text generation. An attention mechanism is proposed in [4] that targets multiple regions in an image for object detection. This approach uses ResNet baseline and is tested on MS-COCO dataset. State of the art models implement object-semantics aligned pre-training using object tags in images as anchor points in a method coined OSCAR [15].

Conclusions

In this project we selected a baseline model for image captioning and improved the model with InceptionV3 and GloVe 300d embedding. The loss of our augmented model is lower than the baseline model. The BLEU scores are also slightly higher. The performance of our model, however, is limited by the small training datasets that we used and also our limited access to high speed GPUs. Our observations, described in the discussion section, will help us detect the sources of shortcomings and improve our model in the future.

Future Work

For the future work, we will implement EfficientNet-B7 as our object detection model with pre-trained weights on ImageNet. First introduced in Tan and Le 2019 [14], EfficientNet is a convolution neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient, a method known as compound scaling. EfficientNet reaches high levels of accuracy with significantly less data than competing models. We also consider other options such as Fast R-CNN [8] and HOG as possible alternatives.

To solve the overfitting problem, we can try training our model on a larger or a more specific dataset. The MSCOCO dataset has approximately 150,000 image-caption pairs, and the ImageNet dataset has several million pairs. Both datasets include a wide variety of detailed scenes and objects. We can also try generating a dataset with reduced class variety - for example, if we were able to reduce the classes of images to animals outdoors, then select all pairs in a large dataset like ImageNet, we may be able to train our model to generate effective captions on this smaller subset of images without demanding enormous computational resources.

References

1. Vinyals, Oriol, et al. "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge." *IEEE transactions on pattern analysis and machine intelligence* 39.4 (2016): 652-663.
2. Mullachery, Vikram, and Vishal Motwani. "Image Captioning." *arXiv preprint arXiv:1805.09137* (2018).
3. Yun, Sangdoo, et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features." *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
4. Anderson, Peter, et al. "Bottom-up and top-down attention for image captioning and visual question answering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
5. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

6. Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
7. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Communications of the ACM* 60.6 (2017): 84-90.
8. Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2015.
9. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
10. Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." *European conference on computer vision*. Springer, Cham, 2014.
11. Rashtchian, Cyrus, et al. "Collecting image annotations using amazon's mechanical turk." *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. 2010.
12. Young, Peter, et al. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions." *Transactions of the Association for Computational Linguistics* 2 (2014): 67-78.
13. How to Develop a Deep Learning Photo Caption Generator from Scratch, by Jason Brownlee on June 27, 2019 in Deep Learning for Natural Language Processing.
14. Tan, Mingxing, and Quoc V. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946 (2019).
15. Xiujn Li et al. "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks." arXiv preprint arXiv:2004.06165 (2020).
16. Szegedy et al. "Rethinking the Inception Architecture for Computer Vision." arXiv preprint arXiv:1512.00567v3 (2015).