The American University in Cairo

The School of Sciences and Engineering

CSCE 4930 – Practical Deep Machine Learning

Dr. Mohamed Moustafa

Final Project

“Image Captioning”

Submitted by:

Ahmed Ibrahim … 900153478

Amr Gouhar … 900153482

Abstract—This paper describes our experiments for image captioning with the MS COCO Dataset. Our experiments were based on a tutorial, which is referenced, that we were trying to enhance. The full description of the project is explained in the paper.

Keywords—image, caption, deep learning, neural networks

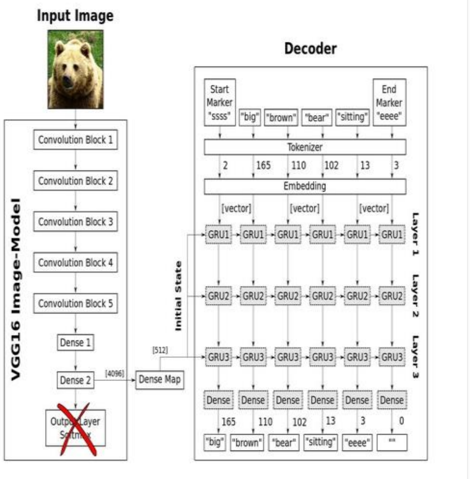
I. Introduction

The problem we were trying to tackle using a machine learning model in this project is image captioning; that is, given an image, the model should be able to generate a caption that best describes that image. To solve this problem, we used the COCO Dataset which contains many images for training, validation and testing. Our model is a neural network model that uses both convolutional neural networks and recurrent neural networks. The details for the model will be explained later.

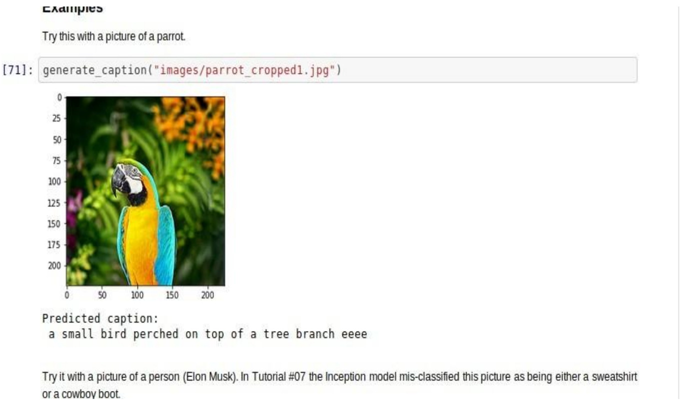
.

II. Literature review

This problem of image captioning is not new. Many people tried working on it. One of the proposed solutions was using the convolutional network model (VGG16) along with three layers of recurrent units (GRUs) [1]. The following figure shows the architecture for that model.



The accuracy of that model was not provided; however, it gave pretty good predictions for pictures that did not belong to the training set as it is shown in the following figure.



Examples

Try this with a picture of a parrot.

```
[71]: generate_caption("images/parrot_cropped1.jpg")
```

```
Predicted caption:
 a small bird perched on top of a tree branch eeee
```

Try it with a picture of a person (Elon Musk). In Tutorial #07 the Inception model mis-classified this picture as being either a sweatshirt or a cowboy boot.

In our project, we followed the footsteps of this model. This is going to be explained in the experiment section.

III. Data Collection

In this section we describe how the data is gathered for the MS COCO captions dataset. For images, we use the dataset collected by Microsoft COCO. These images are split into training, validation and testing sets. The images were gathered by searching for pairs of 80 object categories and various scene types on Flickr. The goal of the MS COCO image collection process was to gather images containing multiple objects in their natural context. Given the visual complexity of most images in the dataset, they pose an interesting and difficult challenge for image captioning [3].

For generating a dataset of image captions, the same training, validation and testing sets were used as in the original MS COCO dataset. Two datasets were collected. The first dataset MS COCO c5 contains five reference captions for every image in the MS COCO training, validation and testing datasets. The second dataset MS COCO c40 contains 40 reference sentences for a randomly chosen 5,000 images from the MS COCO testing dataset. MS COCO was created since many automatic evaluation metrics achieve higher correlation with human judgement when given more reference sentences [3].

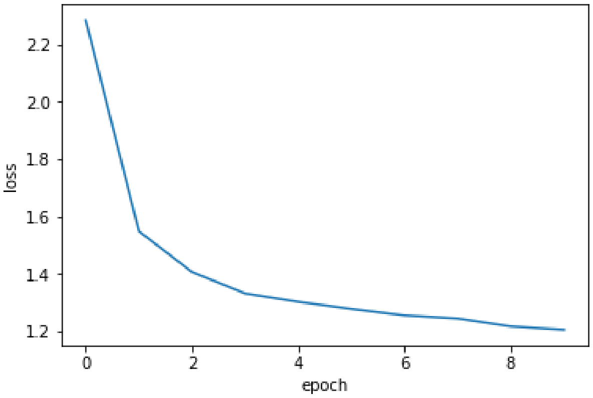
The number of captions gathered is 413,915 captions for 82,783 images in training, 202,520 captions for 40,504 images in validation and 379,249 captions for 40,775 images in testing including 179,189 for MS COCO c5 and 200,060 for MS COCO c40. For each testing image, they collected one additional caption to compute the scores of human performance for comparing scores of machine generated captions. The total number of collected captions is 1,026,459. We plan to collect captions for the MS COCO 2015 dataset when it is released, which should approximately double the size of the caption dataset [3].

IV. Experiments and Results

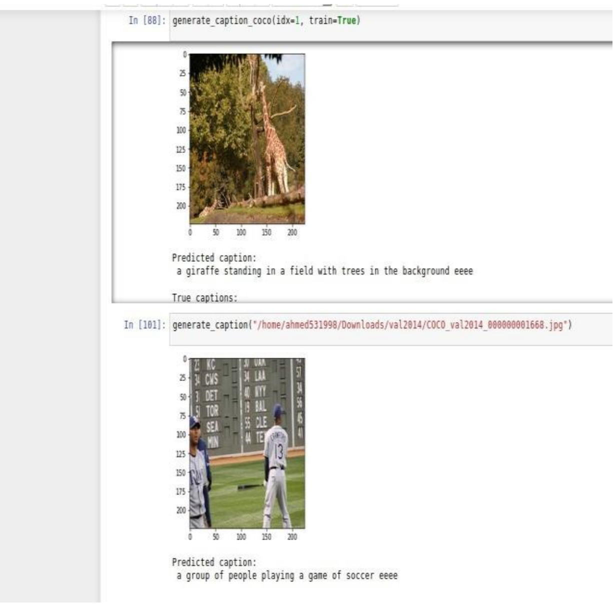
Our model consisted of the following: the convolutional network and the recurrent network. For the convolutional network, we used the already trained VGG16 model and so we did not train that. We only extracted the features from the final fully connected layer and fed that into our recurrent model. Based on research, we found that using more than three recurrent layers does not help with the accuracy and it just causes an overfitting [2] and so in the models that we tried, we did not use more than three recurrent layers.

As for the hyperparameters, we just used a learning rate of 1e-3 which is, based on practice and convention, a suitable learning rate. We used it with the RMSProp optimizer. We also used the sparse cross entropy loss to calculate the loss.

For the recurrent layers, we started by 3 layers, each with 512 units, but we noticed that the loss started to decrease for around 3 epochs but then it started increasing for the remaining epochs. So, we decided to reduce it to 2 layers and also to reduce the number of units in each layer to 50. This gave us better results as it is shown in the following figure since the loss kept decreasing.



As it is evident from the graph, we did not run it for more than 10 epochs because, unfortunately, the machine kept on crashing all the time. But during these 10 epochs of training, the training loss decreased to almost 1.20 which is a good value given the conditions of training that we faced. Therefore, we stuck to this model. Also, a sample prediction is shown in the following figure.



```
In [88]: generate_caption_coco(idx=1, train=True)
```

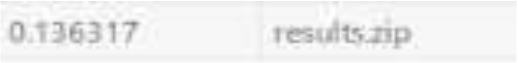
```
Predicted caption:
 a giraffe standing in a field with trees in the background eeee

True captions:
```

```
In [101]: generate_caption("/home/ahmed531998/Downloads/val2014/COCO_val2014_000000001668.jpg")
```

```
Predicted caption:
 a group of people playing a game of soccer eeee
```

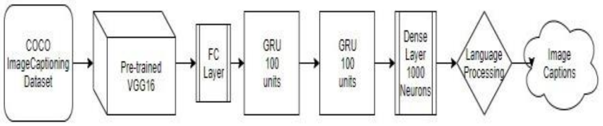
The score that we got when we submitted it to Codalab was (0.1363).



0.136317 results.zip

V. Final architecture

We used the pre-trained VGG16 as our CNN which we feed the input to, and then the output from the final FC layer is fed to our 2-layer GRU RNN with 50 units at each unit. The output is then fed to the Dense layer of 1000 neurons signifying the 1000 words used, then using language processing the caption is created and stored for submission.



VI. Used evaluation metric

The results were submitted on CodaLab for evaluation. The reference sentences for the testing set are kept private to reduce the risk of overfitting. Numerous evaluation metrics are computed on MS COCO. These include BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, METEOR and CIDEr-D [2].

VII. Useful Applications

Image Captioning will be useful in fields where text is most used and with the use of this, you can generate text from images. As in, use the information directly from any particular image in a textual format automatically.

There are many NLP applications right now, which extract insights from a given text data or an essay etc. The same benefits can be obtained by people who would benefit from automated insights from images.

Image captioning would serve as a huge help for visually impaired people. Many applications can be developed in that space for their aid.

Another use is in Social Media. Platforms like Facebook can infer directly from the image, where you are (beach, cafe etc), what you wear (color) and more importantly what you're doing.

VIII. Limitations

There was low computational power and insufficient resources. Constant Kernel Panics despite using batch sizes of 32, 20, 16 and 8 eventually, as well as an electricity cut. All required resetting the experiment several times and yielded very low results at the end.

We did not have much time to try more models and also we there were technical issues with the machines and so the results are not the best since the working conditions were not the best.

IX. Future Work

We'd suggest using the pre-trained ResNet instead of VGG16, as well as replacing all the FC layers with 1-D CNNs. Performing Data Augmentation on the input data by applying different simple filters to the data is also expected to increase the accuracy.

Using an ensemble of networks is also expected to increase the accuracy as well as creating better environment for a better training. This could be done using better machines.

Going through more training epochs would also increase the accuracy as we didn't have enough time to train the machine due to constant interruptions either by power cuts or processor crashes.

However, we were not able to do any of these suggestions due to low computational power and insufficient resources. This is, of course, in addition to the time constraints.

X. References

[1]https://github.com/Hvass-Labs/TensorFlow-

Tutorials/blob/master/22_Image_Captioning.ipy

nb [2] https://arxiv.org/pdf/1805.09137.pdf

[3] https://github.com/tylin/coco-caption
[4] https://github.com/amaiasalvador/imcap_keras