Introduction

Enabling the computing devices to automatically generate descriptions for images is always an interesting and challenging topic. It could have huge variety of applications as auto-captioning for internet uploaded images, live surveillance for abnormal behaviour, personal daily activity analysis, and help visually impaired people understand the content of images.
The parties may concern for this project are
- SNS companies
- Search engine service companies
- Surveillance and security companies
- Intelligent home devices and product companies
- Social welfare organizations

This project aims to apply image caption generation models, which combines recent advances in computer vision and machine translation to produce realistic image captions using neural networks. It would be built as an end-to-end system for this problem, with raw image input, it could output a description contains not only all the objects or scene with properties in this image, but also the relationships and/or movement in a readable sentence.
After successfully established and validated of the proposed model, it would be optimized by varying the encoding structure and obtained the encoding method with best accuracy.

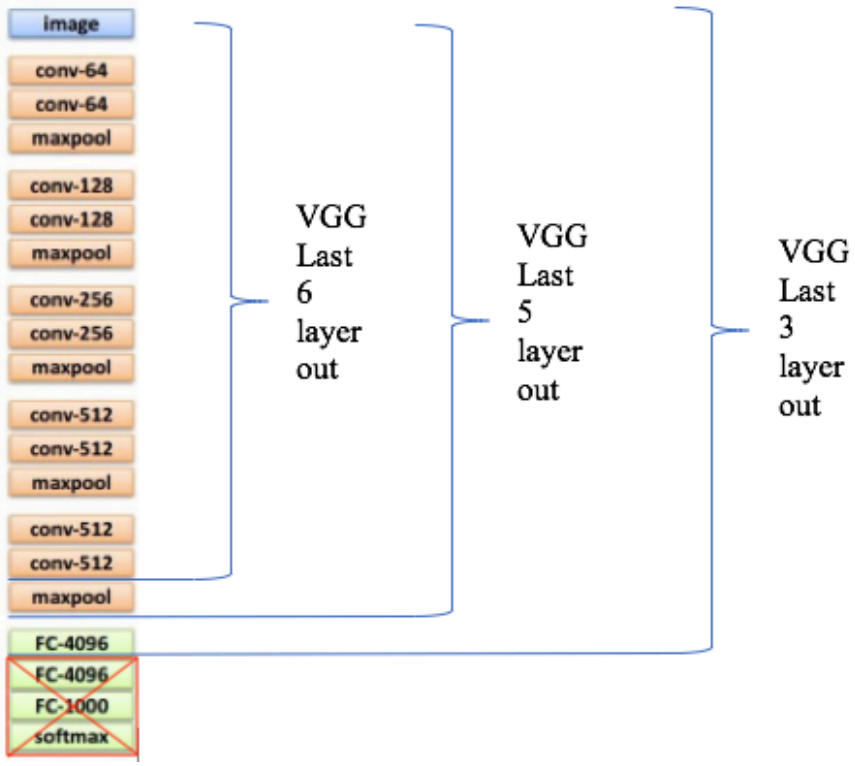
Experiment

General Description
During this project, I'll try to establish the image captioning model using Keras. Train and test would be based on Flickr 8K dataset. This dataset includes 6000 training images and 1000 test images with a full text descriptions for each image.
For the variation of encoder experiment session, I'll try truncation of last 3, 5, and 6 layers of VGG to get image training features, and the optimal one would be obtained based on the metric of BLEU and Per-Word accuracy.

Model
The model consists of 3 main parts, a VGG-16 image encoder, a GRU partial caption encoder and a 2-stack-GRU next-word decoder.

1. VGG-16 Image Encoder

   VGG-16 was originally created for image classification problem in ImageNet. Therefore, we need to drop out the last few layers which are highly related for classification problem. With the truncation of VGG, we could get the encoded features for each image.
   dimensions:
   - (14, 14, 512) for last-6-layer-out
   - (7, 7, 512) for last-5-layer-out
   - (4096,) for last-3-layer-out).



The features would be then passed to a dense layer and repeated to match the dimension with partial caption encoder, which would be elaborated in next section

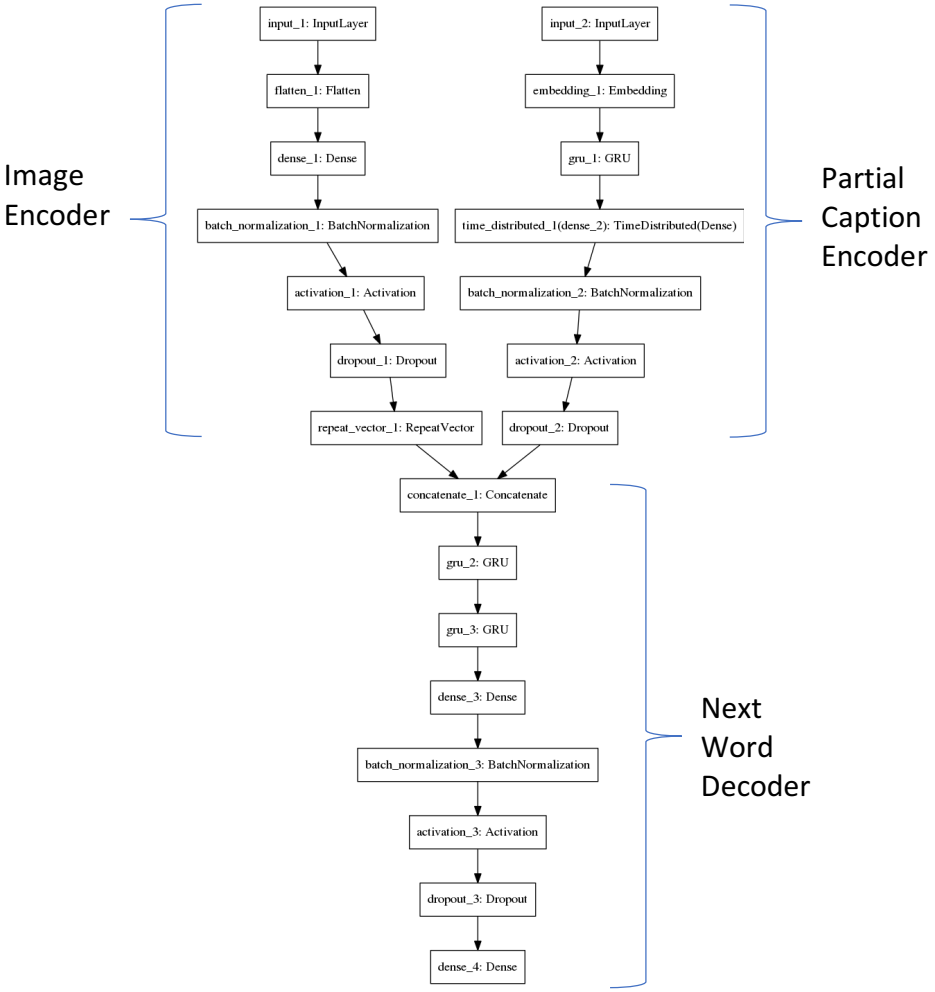
2. GRU partial caption encoder

   Partial caption encoder in this model is composed by 3 parts, a GloVe embedding layer (50d), a GRU layer and a dense layer with batch-normalization and dropout. With padded to full max length for each partial caption, the input would be encoded as a dimension (30, 256) tensor.

3. 2-stack-GRU Next-word encoder

   To make a prediction of output of the next word, I established a 2-stack GRU decoder. The decoder model consists of 4 layers, 2 stacked GRU layers and 2 dense layers. The first dense layer contains batch-normalization and dropout, and the last dense layer is vocabulary sized with “softmax” activation for prediction.
   With concatenation of image and partial caption encoded features as input, the model could output a probability distribution in vocabulary for next word. (vocabulary size: 3896 for all the Flicker8K training set descriptions)

4. Full model



where input_1 is the image features processed by VGG with truncation, input_2 is the padded sequence of partial caption, and dense_4 is the output with dimension of size of vocabulary.

Training

The system uses pre-trained VGG-16 model with weights from ImageNet. Therefore, the image encoding part would not be considered as trainable parameters. Each image's VGG features would be pre-processed and saved in a features.pkl file for faster training and testing. The rest of model is trained with 6000 flicker images and their captions in form of batches:

[
Features: [VGG images_1 features, [startseq, 0, ……..…, 0]], Label: [This];
Features: [VGG images_1 features, [startseq, This, 0, ……., 0]], Label: [is];
Features: [VGG images_1 features, [startseq, This, is, 0, ……, 0]], Label: [a];
Features: [VGG images_1 features, [startseq, This, is, a, 0, ……., 0]], Label: [dog];
Features: [VGG images_1 features, [startseq, This, is, a, dog, 0, …, 0]], Label: [endseq];

Features: [VGG images_2 features, [startseq, 0, ……..…, 0]], Label: [Man];
Features: [VGG images_2 features, [startseq, Man, 0, ……..…., 0]], Label: [run];
…
]

Since the computing device on my Azure virtual machine has 112GB memory and 2xTesla GPU, I chose as much as 300 images per batch to train. This configuration could help for fast training because all samples in a batch could run parallel. Besides, the large batch size would significantly reduce noise for gradient descent and obtain smooth training curve.

Test and Evaluation

During test procedure, initially, image features with a startseq token would be feed into the model and MAP method to sample the next word from the vocabulary distribution. Next, the new partial caption, [startseq, first predicted word], and same image features would be feed again into the network to predict the second most probable word in this sentence. This procedure would run iteratively until we got the endseq token or meet the max length of the description.

The evaluation is based 2 different metrics, BLEU and per word match.

Per word match measures the probability of recall of words in the actual description. It simply counts the number of words appeared both in actual description and the predicted caption and divided by the actual description length.

BLEU metric is the abbreviation of bilingual evaluation understudy. It is a modified n-gram precision with brevity penalty. BLEU firstly calculates the precision of n-gram (n words in sequence) in form of weighted log likelihood with truncation of repeated n-gram. And next, the term would be penalized by the sentence length in form of:

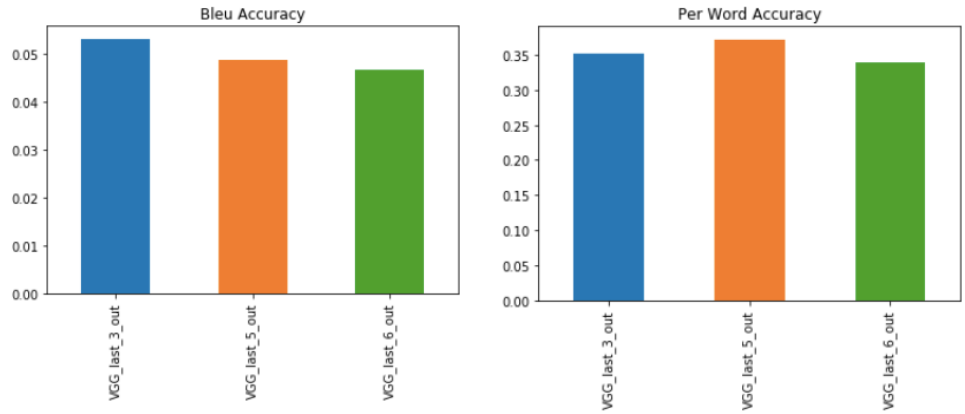
$$BP = \begin{cases} 1 & if \quad c > r \\ e^{1-r/c} & if \quad c \leq r \end{cases}$$

Therefore, the final BLEU metric is in form of:

$$Bleu = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n)$$

Results

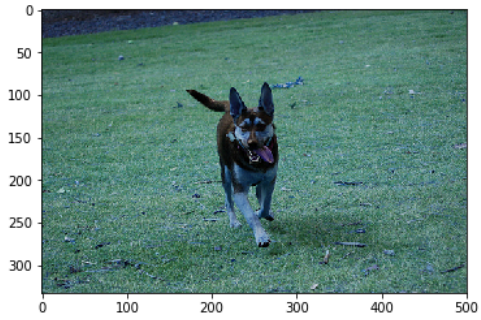
After training on 6000 images and test on 100 sample images for 3 models, I got the result of bleu accuracy and per-word accuracy for each encoding method as follows:



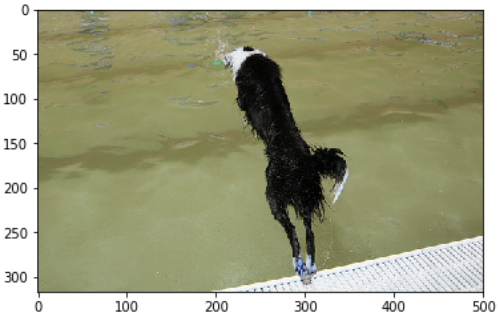
From the test accuracy result, we could conclude that the VGG with last 3 layers' truncation (output feature dimension (4096,)) performs best on the BLEU metric and VGG with last 5 layers' truncation (output feature dimension (7, 7, 512)) performs best on the Per-Word metric.

Precisely captioned image samples:

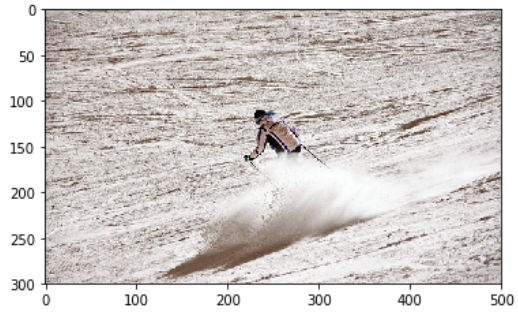
Predicted: dog runs in the grass



Predicted: black and white dog jumps in the water



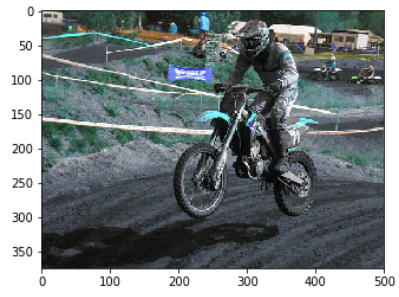
Predicted: lone skier skiing down snowy mountain



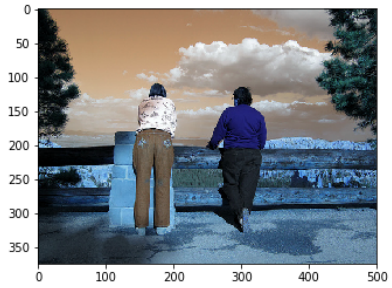
Somewhat relatedly captioned image samples:

Predicted: man rides down the sidewalk next to another man

Predicted: biker riding on marked by tree

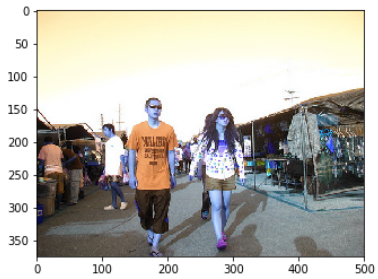


Predicted: guy and girl are standing on railing near the ocean



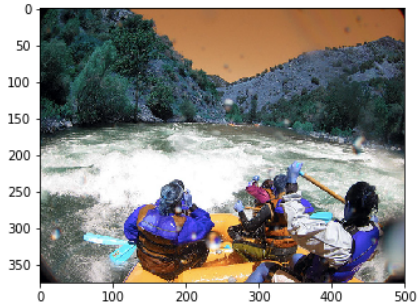
Non-relatedly captioned image samples:

```
Predicted: woman carrying silver is standing outside carpet
```



```
Predicted: back is standing on two sling of red celebrating
```

```
Predicted: group of people are sitting in bleachers and the two boys are on the ground surrounded by body of water
```



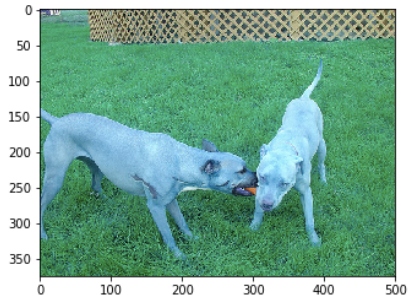
Discussion

Since BLEU metric is more applicable to sentence evaluation, we should judge the model based more on this evaluation method. According to the result from previous sections, we may draw a conclusion that encoding with last 3 layer truncated VGG produces better test accuracy.
Higher dimension of model layers in neural network would induce higher nonlinear transformation of features for images. Therefore, VGG with only fewer layers truncated could keep image encoded with less noisy and easier to classified features. However, on the other hand, the deeper layers' features may closer to the original task domain of pre-trained VGG (classification problem). At the end, the compromised solution of our task suggests that VGG with last 3-layer truncation may be the best encoding method.

Future Development

1. The evaluation metric is highly biased on the partially descripted human caption. Sometimes, the predicted caption is totally deviated from the actual caption. This would induce an evaluation with low score on BLEU and Per-Word. However, we may not totally accept the evaluation and simply predicate the caption is wrong because the predicted caption and the actual one focused on different scene of the image. Example:

```
Actual: golden dog with blue object in its mouth stands next to another golden dog in front of green fence in grassy
field
Predicted: two dogs are fighting in the grass
```



Therefore, we need to collect more candidate human captions for each image for comprehensive description. The evaluation score would be more acceptable.

2. In BLEU and Per-Word metric, different word predicted from actual one would be evaluated as false. This “hard” predication may lead to an underestimation of sentence that contains word similar to the actual one. We need to create a new “soft” evolution metric with ability to measure the similarity distance of each word and its context in a sentence. For example, we could change the n-gram precision in BLEU to the n-gram distance of words in embedded space.
3. To further improve the accuracy of the model, we need more training data to enrich the scene, movements, objects and vocabulary.