PES
UNIVERSITY

Project Presentation
(Final - ESA)

Project Title : Image Caption Generator

Project Team :

Kritika Kapoor	PES1201701868
Shubha M	PES1201701540
Shrutiya M	PES1201700160

Project Abstract and Scope

Image Caption Generator:

1. Image Captioning refers to the process of generating textual description from an image – based on the objects and actions in the image.
2. Image captioning has various applications such as recommendations in editing applications, usage in virtual assistants, for image indexing, for visually impaired persons, for social media, and several other natural language processing applications.
3. Generally, a captioning model is a combination of two separate architecture that is CNN (Convolutional Neural Networks)& RNN (Recurrent Neural Networks) and in this case LSTM (Long Short Term Memory), which is a special kind of RNN that includes a memory cell, in order to maintain the information for a longer period of time.

Many systems use Vanilla RNN for this problem statement, but using LSTM we maintain the meaning of the caption predicted so far by remembering it, and use that to produce the next word. Hence it is an improvement over the classic model.

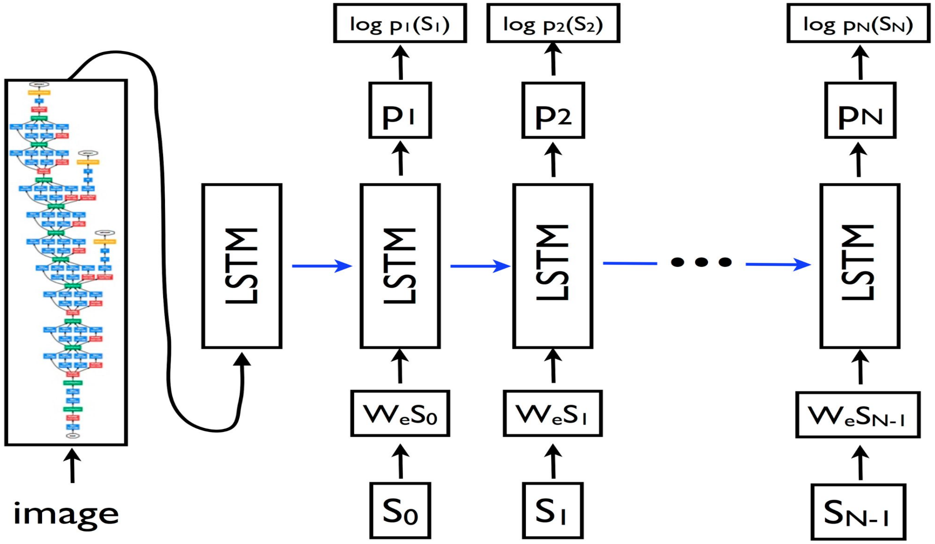
Project Scope (cont.d)

1. Image Captioning is one of the areas where Deep-Learning and Natural Language Processing meet. The idea comes from the inspiration of mimicking the remarkable human ability to compress huge amounts of salient visual information into descriptive language.
2. It’s the most basic step to other complicated deep learning applications in areas such as web development, it’s good practice to provide a description for any image that appears on the page so that an image can be read or heard as opposed to just seen. This makes web content accessible.
3. It can be used for Video processing techniques as the basic processing of a video begins with an image. (Video summarization, Scene extraction etc)

Solution Architecture

Our Model comprises of two basic components:

1. Convolutional Neural Network: This is used for extracting the most important features of the input image, breaking down an image of size 224*224*3 into a vector of 1*4096 recognizable features.
2. Recurrent Neural Network: Input to this component of the model are the features, in which we implement LSTM cells, an improvised concept of RNN involving memory. Input to the cell at time step t will be the caption predicted at time step t-1 and the weights are accordingly adjusted to produce the most probable word keeping the embeddings into consideration.



Model

1. We take the image embedding from the VGG-16 model and use it to train the rest of our model.We pre-computed the 4,096 dimensional features to speed up training.

2. Writing them into a npy file, we make our work easier by using them to train the RNN.

3. To transform words into a fixed-length representation suitable for LSTM input, we use an embedding layer that learns to map words to 256 dimensional textual features (or word-embeddings). Word-embeddings help us represent our words as vectors, where similar word-vectors are semantically similar.

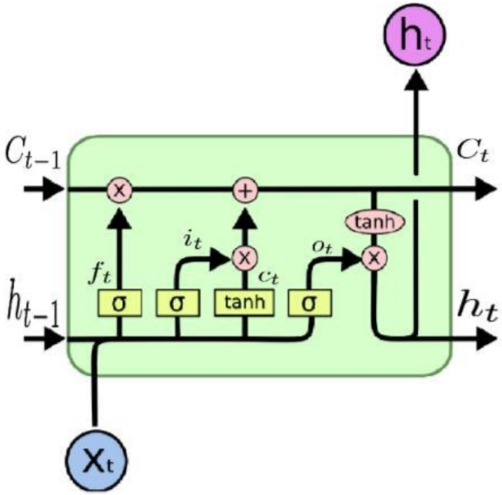
4. Our custom RNN model consists of a BasicLSTMCell implemented using Tensorflow, used to generate a word at every time step, looped till the maximum number of words that constitute a caption.

5. From the available captions, vocabulary is formed by setting a threshold on the frequency of the words. Each word is mapped into a 256 dimension vector. Also, the image is mapped into a word space to provide appropriate input to the LSTM cell.

6. To predict the next word, the embeddings of the previous word are passed to the LSTM and finally in the end these features are encoded back into words Caption is generated using a naive greedy approach.

Math used

Using the Basic LSTM cell implemented by tensorflow, we make use of the following architecture:



Forget gate represented by f_t
Input gate represented by i_t
Output gate represented by o_t

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

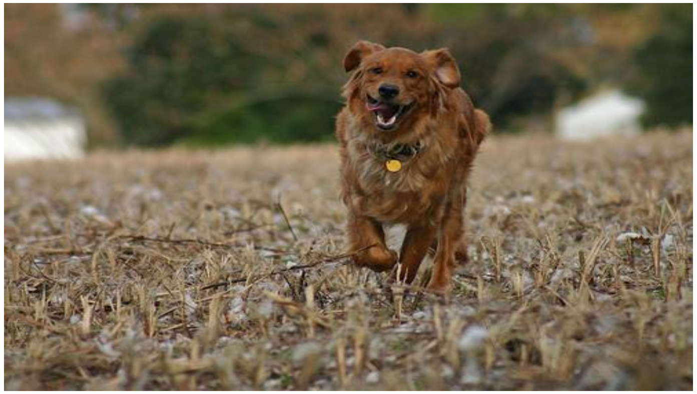
We use a batch size of 128 and number of units in LSTM cells 256.
Weights that are learnt by our model - (Vocabulary size = 996 words)
Word embedding(996 * 256), Word embedding bias (256 * 1)
Image embedding(4096 * 256), Image embedding bias (256 * 1)
Word encoding(256 * 996), Word encoding bias(996 * 1)

Model Description(cont.d)

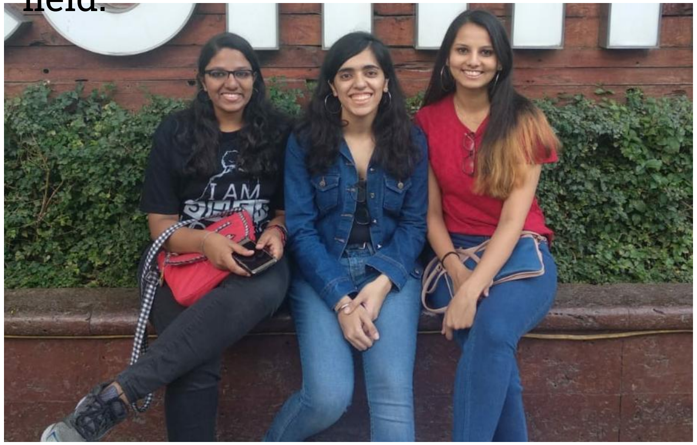
Activation Function: Softmax
Loss Function Used: Cross Entropy
Optimizer : Adam optimizer
Testing : We currently have a model that gives the probability of a word appearing next in a caption, given the image and all previous words.An image is given as input to the model and iteratively outputs the next most probable word, building up a single caption. (Naive Greedy Search).

Thus, the caption generator gives a useful framework for learning to map from images to human-level image captions. By training on large numbers of image-caption pairs, the model learns to capture relevant semantic information from visual features.

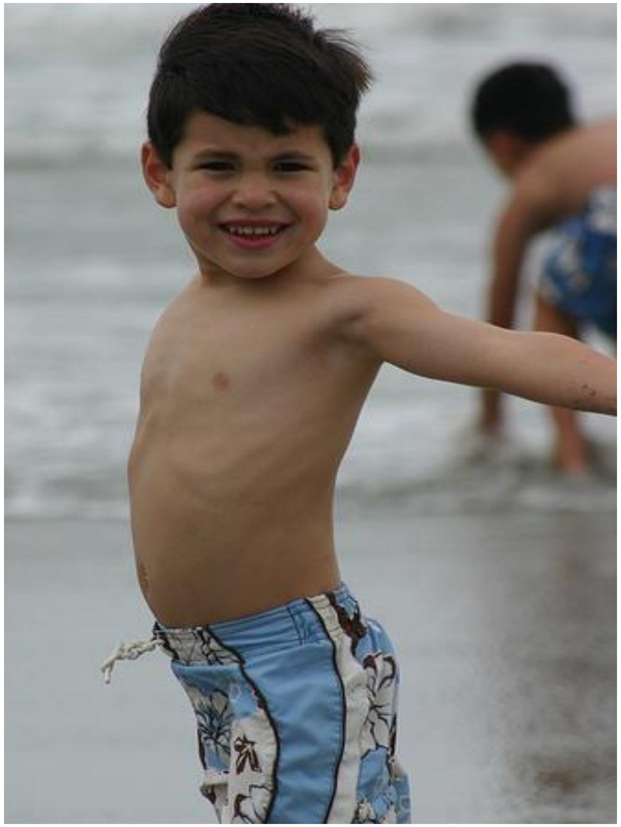
Examples for captions generated



A brown dog runs through a grassy field.



Two women and a man are smiling and walking in a park .



A boy in swimming trunks walking along the .

Constraints, Assumptions & Dependencies

1. As of now, due to limited computational resources, we trained it on Flickr8k dataset consisting of 8091 images and 5 captions for every image. We were not able to get features embedding for Flickr30k due to limited computational capacities.

2. One of the fundamental assumptions is that similar images are likely to share similar and correlated annotations.

3. Mundane captions are generated and not selectively focussing on areas of interest. Can be overcome by using visual attention mechanism.

4. Dependencies : Tensorflow 1.x, Numpy, scipy(1.14) (imread, imresize), keras.

Future work plan

1. As of now, due to limited computational resources, we trained it on Flickr8k dataset consisting of 8091 images and 5 captions for every image. We would like to improve the model by training it on Flickr30k dataset, consisting of 30k images or COCO dataset which is 42.7 GB huge.

2. We have chosen the basic LSTM cell as our processing unit in our Recurrent Neural Network, we would try and experiment with a customized RNN cell to see how it affects the network.

3. Using Attention mechanism for better captioning.

4. Using Beam Search for better captioning.

5. Extending the image captioning concept to Video summarization.

References

1. Vinyals, O., Toshev, A., Bengio, S. and Erhan, D., 2016. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, *39*(4), pp.652-663.

2. Deep Visual-Semantic Alignments for Generating Image Descriptions
Andrej Karpathy, Li Fei-Fei; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3128-3137

3. Guiding the Long-Short Term Memory Model for Image Caption Generation
Xu Jia, Efstratios Gavves, Basura Fernando, Tinne Tuytelaars; The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2407-2415

Thank You