Image Caption Generator using pre-trained CNN and word2vec models

Dasari Sai Praneeth
NIT Calicut B190689EE
saipraneeth_b190689ee@nitc.ac.in

Kammula Siva Naga Manikanta
NIT Calicut B190567EE
sivanagamanikanta_b190567ee@nitc.ac.in

Dinakar Chennupati
NIT Calicut B190904EE
dinakar_b190904ee@nitc.ac.in

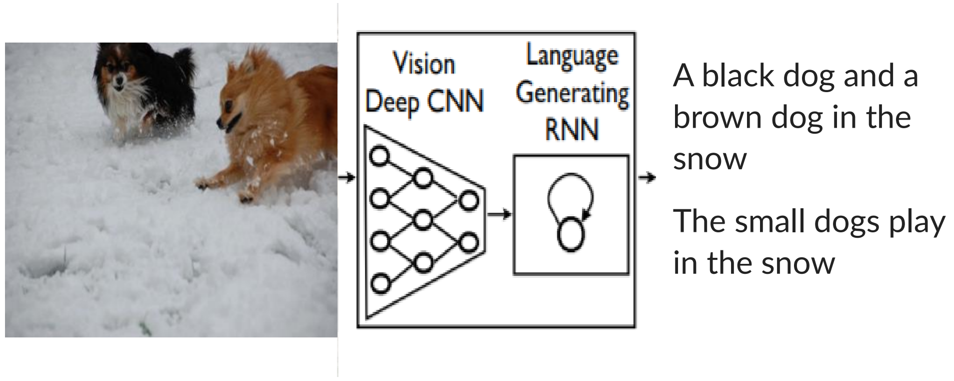
Abstract:

Recent advances in Computer vision and Natural language Processing manifested the capability of world of image exploration. With current state-of-the-art, the neural network models are able to digest the content hidden in the image and express it in natural language courtesy to the advances in NLP. In this Image Caption project we're going to explore how well our model defines the content of the image by utilising InceptionV3 model & VGG16 as Image Encoders, Glove6B model as Text encoder and Greedy & Beam Search for captioning the images. BELU evaluation policy is going to be adapted for individual image that gives insights to the performance of our model.

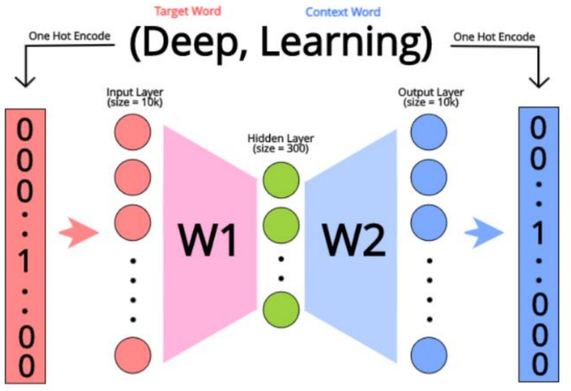
1. Introduction

Generating descriptions for images is one of the gruelling tasks present in the field of deep learning out there. As humans, when we are given an image, we can perceive the significant details present in the image and at the same time we are capable of omitting insignificant ones. Now, given the deep learning network our image and requesting to generate description requires discerning image and understanding natural language is no simple assignment. Whatever may be the depth of the problem, this task equally finds its importance in many applications in our day-to-day life. For instance, given smartphones visually impaired people can benefit remarkably by knowing things that present around them, also identifying correct scenarios for self-driving cars which pushes autonomous technology to next step and last but not least CCTV cameras to notice any unusual activities going around.

The story begins with CNN architecture[1] which is known to lead the race of discerning image, providing mind boggling computer vision. And at the same rate another architecture known as RNN which initially understood sequence processing, later advanced to generate natural language which can be seen in google translation model. Further issues with vanishing gradients led to another RNN based LSTM model[2] which is capable of memorizing large time series data and so on.

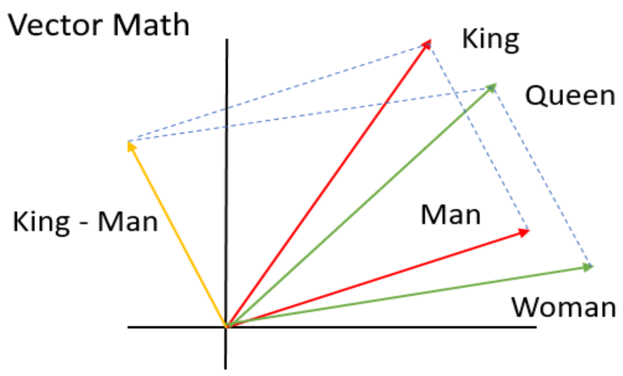

Vision
Deep CNN
Language
Generating
RNN
A black dog and a brown dog in the snow
The small dogs play in the snow

Previously in deep learning tasks like machine translation [3] which is model to fathom conversion of languages from one to another is piece of puzzle in the black hole deep learning that still got room for innovation. Approach to this bewildering problem in words in pretty simple and plain. Using RNN as 'encoder' for feeding input through the pipeline and also using same recurrent neural network state-of-the-art model, but this time as 'decoder' to generate required results.



Word encoder and decoder used as one of the interface of Machine translation

In essence, the approach for generating descriptions for images got the same simple approach but this time we add CNN as 'encoder' to plunge the input features along with partial captions into the input pipeline. Elaborating this approach, we use some well pre-trained models that are best known in object recognition tasks to extract the image features i.e., encoding images in some well numbered feature vectors contingent to different models present out there. For instance, InceptionV3 when it gets dumped with images while dropping its last two layers, we get produces 2048dimensional vector. And 4096 numbered feature vectors for VGG16 model respectively. Now, uprooting vectors from



Glove6b Model: Featuring how word represented as vectors are related.

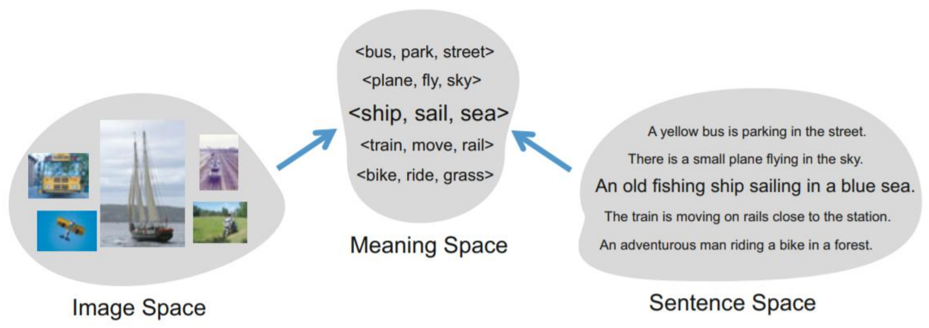
words can be done either from scratch using some well-known functions like Tokenizer in Keras and at the same time with pre-trained models like Glove6b[4] developed in Oxford University. The advantage of using pre-trained one for word2vec is that it provides universal context for words which are closely related.

So, we begin our recipe implementing above scenario with the Flick8k dataset[5]. We vaguely maximized the whole above discussed process to fine tune the model and expect some good generated outcomes that are well within the boundaries of good state-of-the-art results.

2. LITERATURE SURVEY

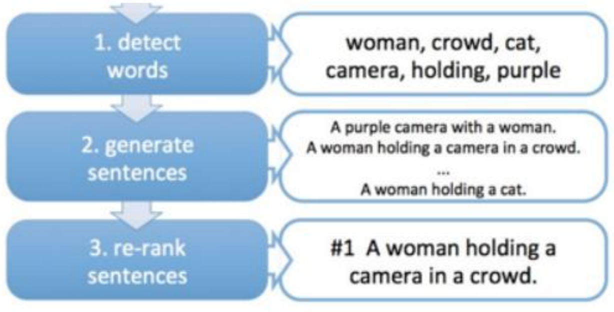
A vast and diverse amount of research had been done on this staggering project of image captioning. As improvements in object classification and generative neural networks got advanced, this project got the same pace in its improvement and development. Increased interest in the field of describing images based on significant details present in them led to innovation of many models to tackle this enthusiastic problem.

One of the well-known papers provided by highlighting the details about this captioning approach is ‘Every Picture Tells a Story: Generating Sentences from Images’ paper proposed by Ali Farhadi et al[6]. This paper tries to wrestle the problem with triplet representation space namely object, action, scene. In essence, images and texts are widely scrutinized in this meaning space. Similarity of images and texts in the meaning space will result in high score and then asserts that caption to that image.

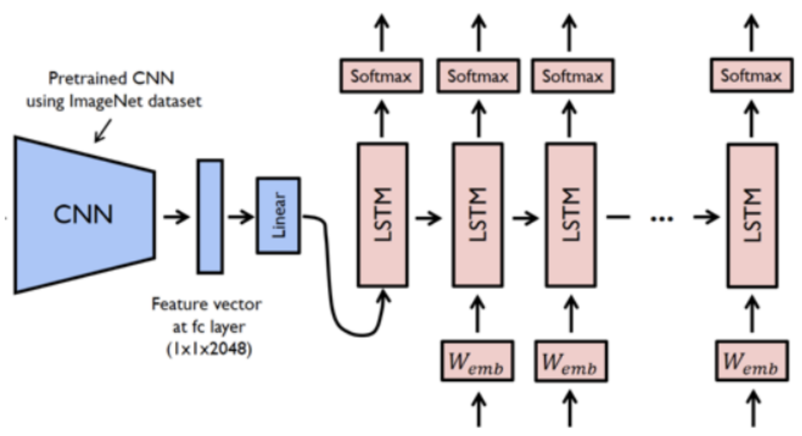

<bus, park, street>
<plane, fly, sky>
<ship, sail, sea>
<train, move, rail>
<bike, ride, grass>
Meaning Space
Image Space
A yellow bus is parking in the street.
There is a small plane flying in the sky.
An old fishing ship sailing in a blue sea.
The train is moving on rails close to the station.
An adventurous man riding a bike in a forest.
Sentence Space

Triplet Representation Space: Object, Action and Scene

In ‘From Captions to Visual Concepts and Back’ paper[7] where the experiment is done on Microsoft COCO dataset achieved state-of-the results by presenting ‘multi-modal’ point of view. The concept is pretty much straightforward: Input the images by extracting feature vectors but when dealing with descriptions describing those images the intuitive work they had done is better explained in three steps. Simply detect words, then generate sentences but most importantly re-rank sentences.

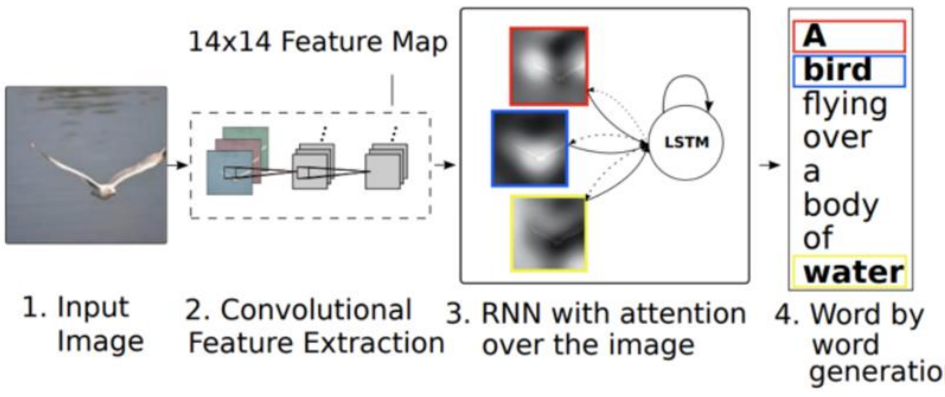

1. detect words
woman, crowd, cat, camera, holding, purple
2. generate sentences
A purple camera with a woman.
A woman holding a camera in a crowd.
...
A woman holding a cat.
3. re-rank sentences
#1 A woman holding a camera in a crowd.

Another paper proposed by Google AI researchers ‘Show and Tell: A Neural Image Caption Generator’[8] provided their recipe by proclaiming that their work is inspired by advances in machine translation which provided significant insights in improvement of sequence-to-sequence learning which paved the success path for their Neural Image Captioning Model. As a whole, they’ve provided encoder-decoder framework for generating captions to the images.



Architecture of Show and Tell NIC using encoder-decoder framework

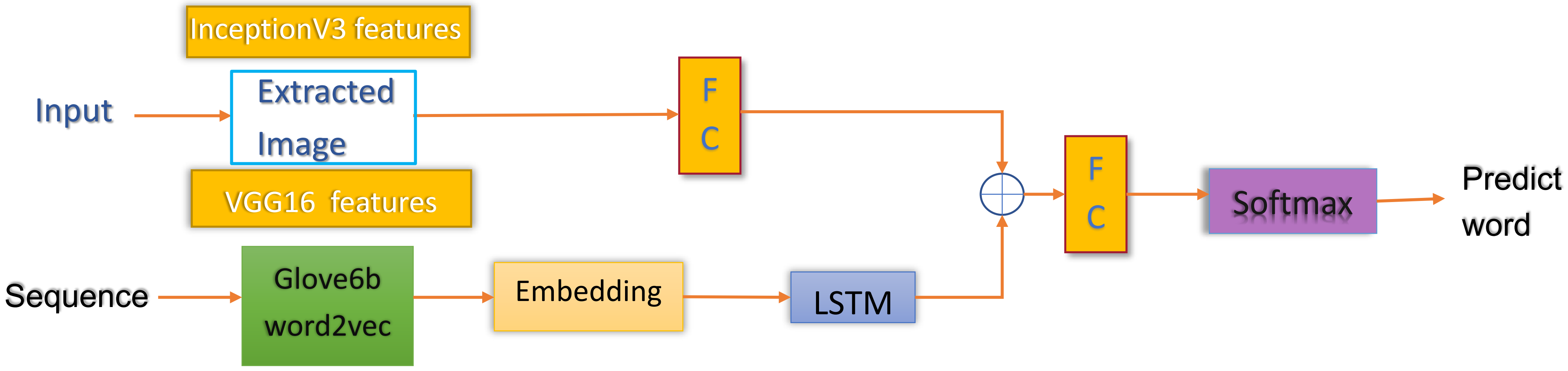
‘Show, Attend and Tell: Attention Mechanism and Image Captioning’ paper proposed by Kelvin Xu et al.[9] dealt with attention model. Well, this attention mechanism is an improved version of encoder-decoder based framework which also generated well state-of-the-art results in field of image captioning.


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Bird view of Attention Model

3. MODEL

In a nutshell, our CNN-LSTM word embedding model architecture looks as shown in figure. We've used InceptionV3 model to encode images into 2048 dimensional feature vector & VGG16 model to extract 4096 dimensional feature vector, tokenizer to encode captions into 200 featured dimensional vector. For generating captions we synthesized Greedy and Beam Search.



Architecture of our Model (Note: we considered InceptionV3 and VGG16 features separately and FC stands for fully connected layer)

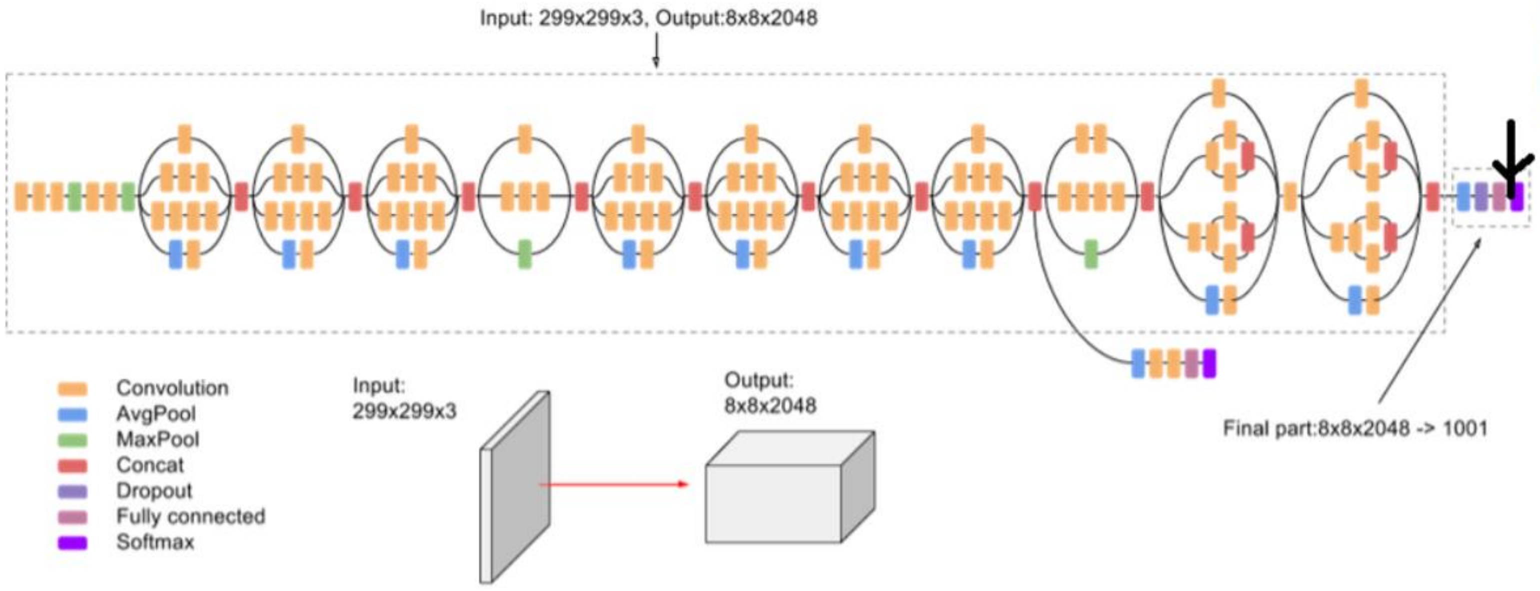
In summary, we have: One of two inputs as partial caption vector and another one as Image feature vector respectively.

Output is an n appropriate word, next in the sequence of partial caption provided in the input 1 (or in probability terms we say **conditioned** on image vector and the partial caption)

3.2 InceptionV3 Model

InceptionV3 neural network model[10] is well known object classification model which achieved around 93.7% top-5 accuracy when trained more than a million images to classify 1000 object categories from the store of animals to electronic gadgets. Since this model is well incorporated with the features of over a million images, it will be a good starting point for our network to encode our image to get accurate features from our image dataset. This process of propping pre-trained model for the basis of another task of similar objective is called as Transfer Learning[11].

Nevertheless, since we're dealing with object classification, we just used the model to encode the images i.e., convert images into accurate feature vectors, we've excluded last softmax layer. This is concept that comes under the topic of automatic feature engineering.



At this point we extract the 2048 - length vector by removing the last softmax layer which performs 1000-class classification

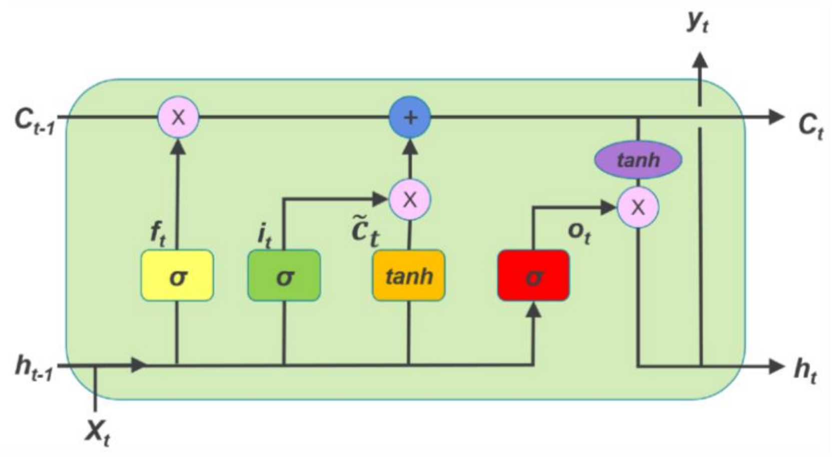
Hence, as prop to our model we omitted softmax layer and able to draw a 2048 length feature vector(bottle neck features) for every image in our flickr8k dataset.

3.3 LSTM based text generator

Apparently, a sentence can be represented as words. The time step t is simply an index of tth word in the sentence which represents the position of each word. For instance, if our sentence encompasses of T words then the time step of first word is t = 1, the second word is t = 2 and for the last word is t = T. We added START and END tokens to each word to indicate start and end of the sentence which are first and last time steps respectively.

Every single word of a sentence is represented as a vector. Word to vectors conversion might be adopted from scratch but pretrained word vector model Glove by Pennigton et al. is used to map word to 200 dimensional vector, since generally the retrained word vector will achieve higher performance for specific task.

Recurrent neural networks suffer from vanishing gradient problem which makes them vulnerable when tackling long sequences. Hence, LSTM[2] which uses gate concept to retrieve essential information is preferred.



LSTM Gate Cell with three gates

Mathematical equations describing LSTM Model are as follows:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$

$$m_t = i_t \odot \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) + f_t \odot m_{t-1}$$

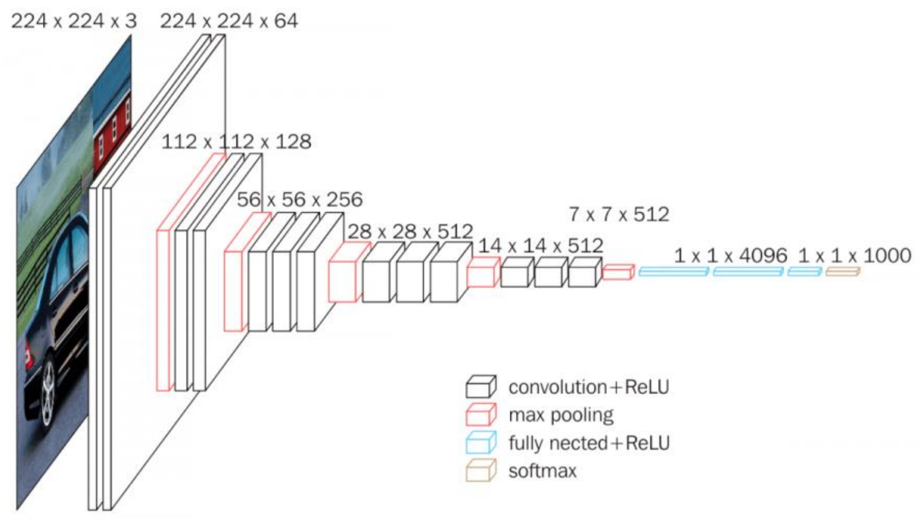
$$h_t = o_t \odot \tanh(m_t)$$

Where y_t is out of the LSTM gate, x_t is current input vector at time t, and W_{ix}, W_{ih}, b_i representing weights and bias emphasizing quantity of x_t to enter into memory cell.h_{t-1}, h_t represents the hidden layer states at time steps t-1 and time t respectively. W_{fx}, W_{fh}, b_f are weights, bias associated with forget gate. A forget gate is the one that controls how much information to be omitted through the LSTM. W_{ox}, W_{oh}, b_o are terms associated with output cell that outputs current latest information from the gate. Note that , $\odot$ represents element-wise multiplication. σ represents the sigmoid function, acts as filtering point by puffing output in between 0 and 1.

Long story short, gate controls how much information should flow (excluded or included) through the network coming from the previous timestep. In the case of output of gate close to 0, the word features almost are not fed, but output close to 1 results in more importance given to those specific word features.

3.4 VGG16

VGG16[12] is a convolutional neural network devised by Simonyan and Zisserman from esteemed Oxford University in the paper "Very deep convolutional networks for large-scale image recognition'. Similar to InceptionV3 model the VGG16 model is trained over 14 million images used to classify images into 1000 different categories and achieved over 90.7% top-5 accuracy.



Architecture of VGG16 Neural Network Model

This manifests the harness power of VGG16 in image classification. But our problem deals with extracting image features i.e., mapping images to vectors so we omitted the last layer same as we had done with InceptionV3 model.

4. Experiment

The experiment is basically intended to train the model by providing images that are proposed to be trained along with partial captions with the pattern of predicting the next word in the sequence. So, each training image is considered to be several data points fed to our model.

For InceptionV3, input 1 represents encoded 2048dimensional image vector and input 2 is 34 sequence length 200 dimensional encoded caption vector.

For VGG16, input 1 is encoded 4096dimensional image vector and input 2 is 34 sequence length 200 dimensional encoded description vector.

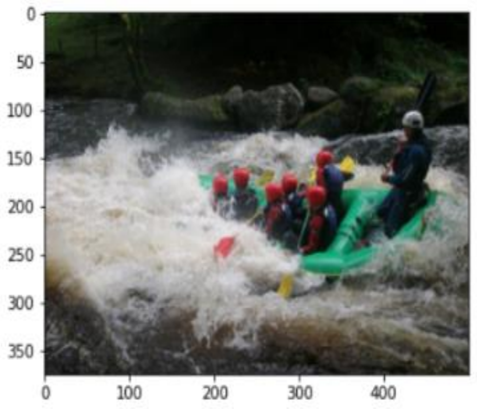
4.1. Dataset

A diverse number of datasets are available to train and generate captions for images. Enumerating, we have Flickr8k, Flickr30k, Microsoft COCO datasets and so on. As we dealing with computational sensitive gpu's we're going to stick with Flick8k dataset which is small out of all datasets provided for image captioning methods and will serve as a good starting in exploring the models designed to generate captions for unseen images.

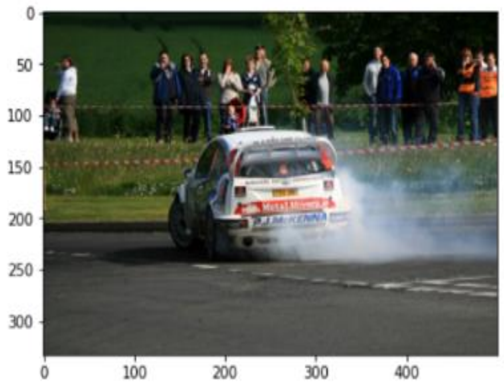
Elaborating Flickr8k dataset[5], it is provided with 8000 images with 5 captions descriptions each image i.e., 40,000 captions in total. Images of different scenarios are incorporated and captioned with various captions describing semantic and syntactic meanings of the respective images.

4.2 Implementation in Generation

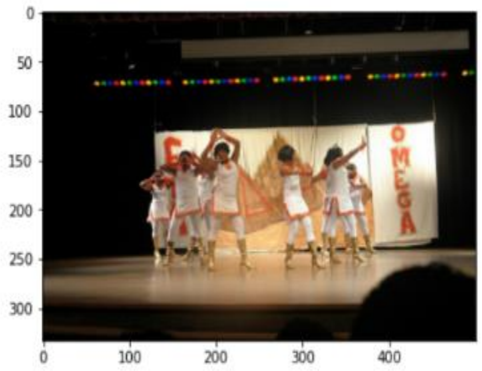
For captioning the test images, we used greedy search and beam search, however, beam search is computationally expensive. Hence, we generated descriptions using greedy search for most of the test images. Some interesting results and patterns can be pointed from the inference of test images. Below some of the well captioned images by the model followed by average predictions which explicitly informs about our model's accuracy dealing with some detailed images which also points the fact of smaller image dataset and could be well improved with huge captioned datasets but computationally expensive all the same.



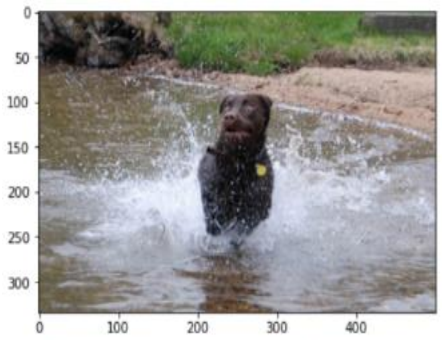
Greedy: a group of people rafting down a river



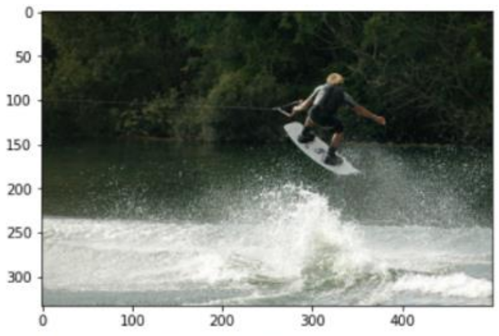
Greedy: a race car spins down a hill



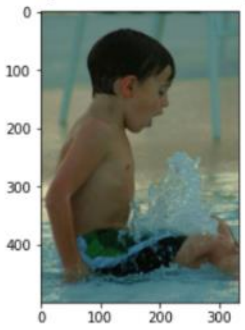
Greedy: a group of children are playing on a school stage



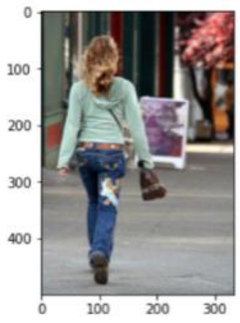
Greedy: a dog is running through the water



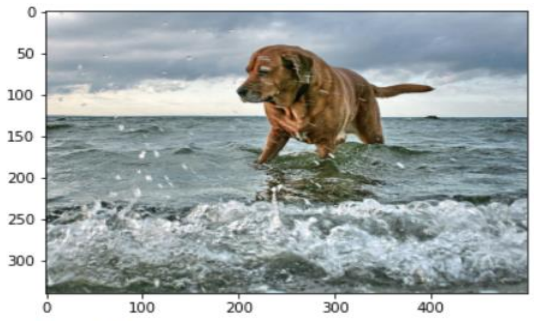
Greedy: a man paddles a kayak through rapids



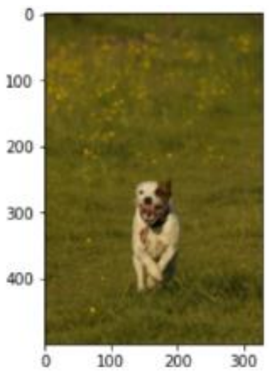
Greedy: a boy in a swimming pool



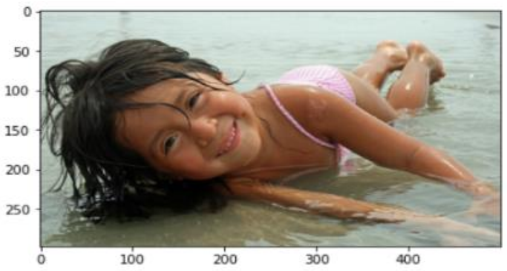
Greedy: a woman in a red vest is walking down the street



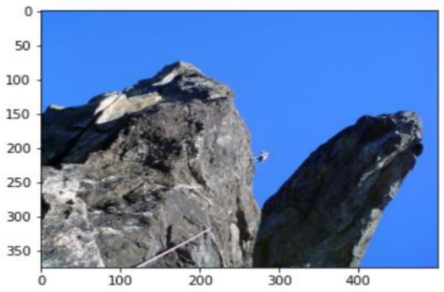
Greedy: a brown dog is running through the water



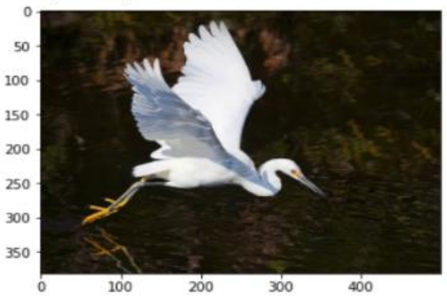
Greedy: a white dog with brown spots is running through a field



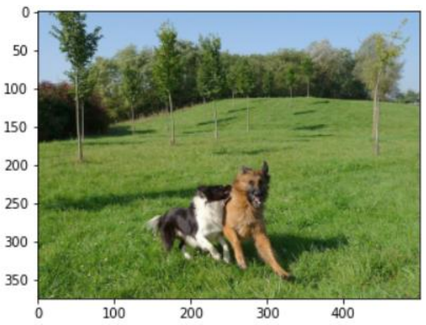
Greedy: a young girl in a red bathing suit is laying on the beach



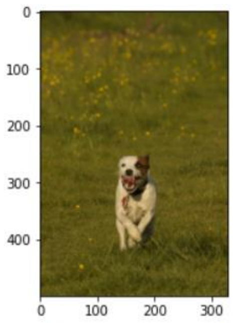
Greedy: a man in a red helmet climbing a cliff



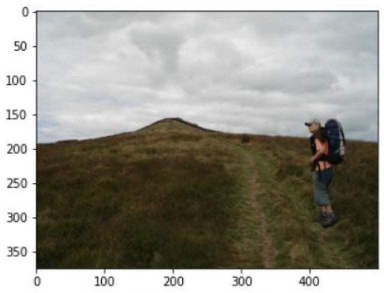
Greedy: a white crane flying over the water



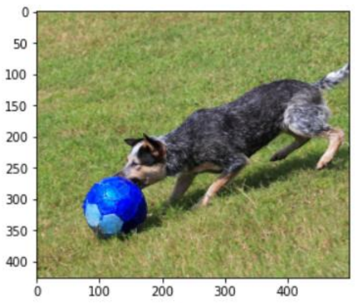
Greedy: a brown and white dog is running through a field of grass



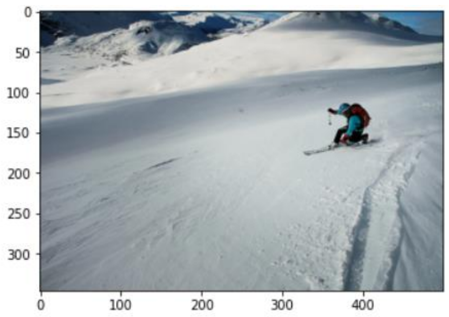
Greedy: a white dog with brown spots is running through a field



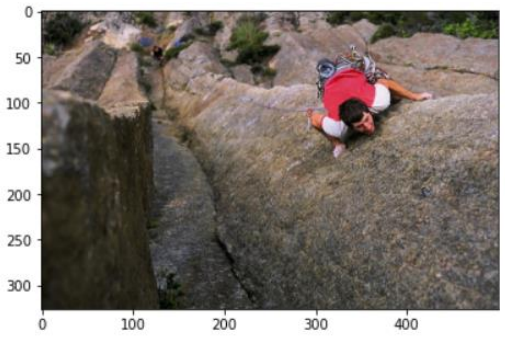
Greedy: a man is hiking up a hill with a mountain in the background



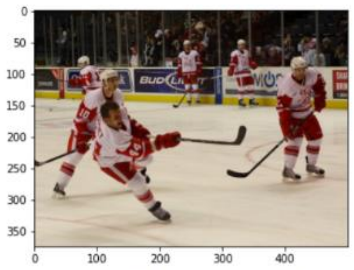
Greedy: a black dog is playing with a soccer ball



Greedy: a skier is skiing down a snowy hill

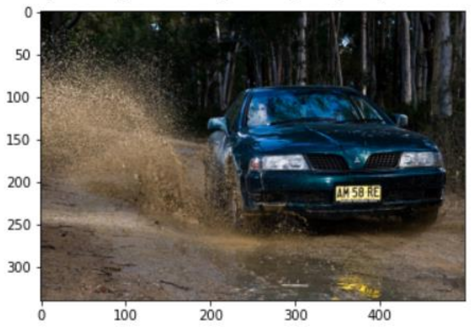


Greedy: a man in a red shirt is climbing a large rock

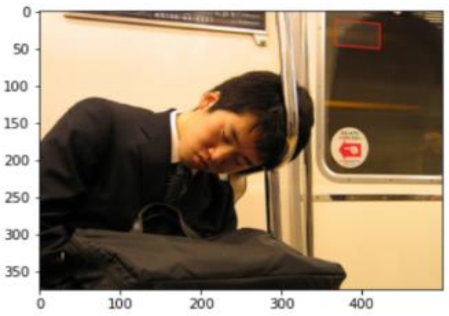


Greedy: a hockey player in a red and white uniform is walking along the ice

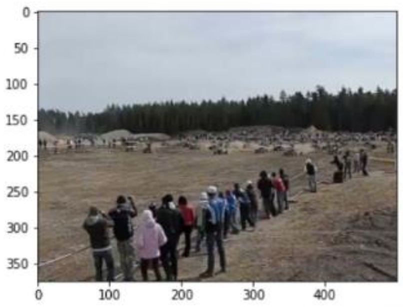
- *Well captioned images*



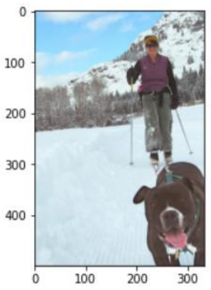
Greedy: a yellow car is driving through the water



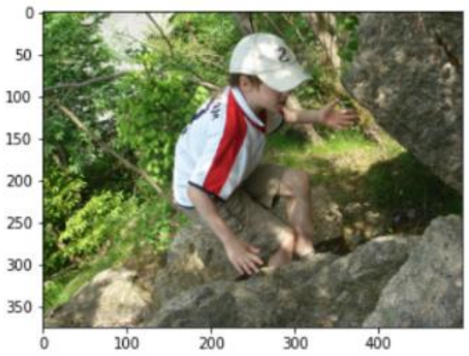
Greedv: a man in a neon costume is sitting on a train



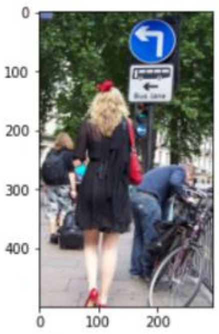
Greedy: a group of people are standing on a dirt road near a river



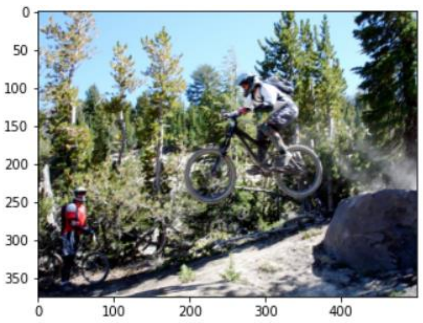
Greedy: a man in a red jacket is standing on a snowy hill



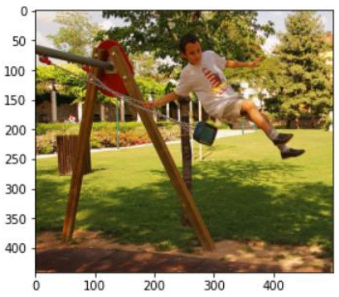
Greedy: a man in a red shirt is climbing a rock face



Greedy: a woman in a black coat is walking down

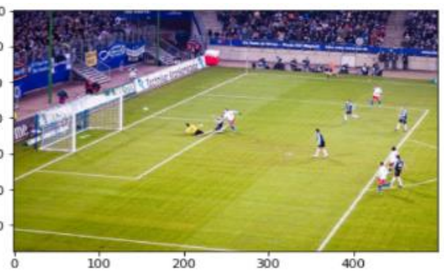


Greedy: a man in a blue shirt is biking on a rocky trail

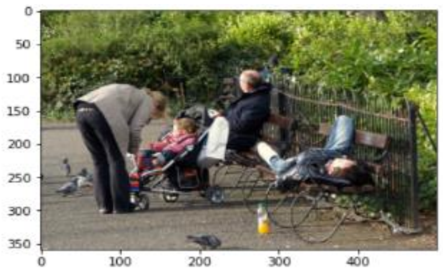


Greedy: a girl plays on a swing

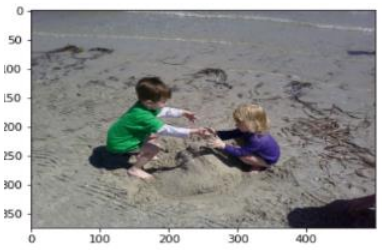
- *Describes with minor errors*



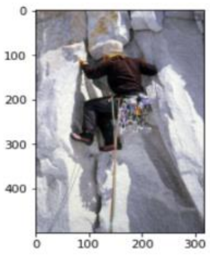
eedy: a man in a white shirt is playing golf



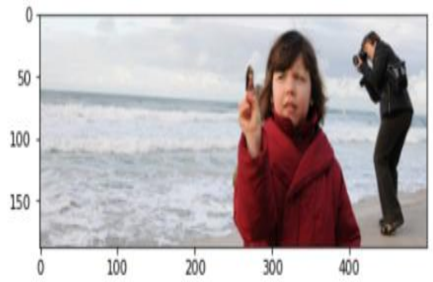
Greedy: a man in a blue shirt is standing next to a truck



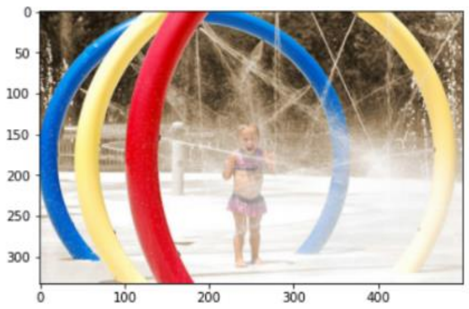
reedy: a group of people are standing on a rock overlooking the water



Greedy: a man in a red shirt is standing on a snowy mountain



Greedy: a man is standing on the beach with a fishing pole in his hand



Greedy: a boy is jumping off of a red slide

- *Unrelated to the images*

4.3 Training Details

For InceptionV3 model, we found 30-35 epochs are good enough to generate some well captioned images. Batch size is chosen as 3 picture per batch and we proceeded with default 0.001 learning rate. Also, it must be perceived that decreasing the learning rate over some 20 epochs makes gradient updates more dynamic and efficient.

For VGG16 model, the interesting thing that we observed is that it starts to overfit way earlier than InceptionV3 Model and we trained it for just 3 epochs and started to observe it overfitting. This makes sense because VGG16 deals with feature vectors of size 4096 vector which is double the InceptionV3's feature vector.

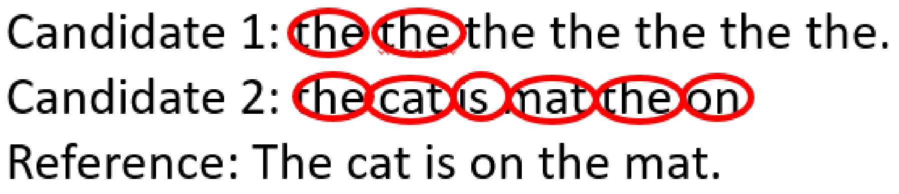
During training, cross entropy loss was chosen as the loss function. To make the model converge faster, Adam annealing policy was adopted. ReLU is chosen as the activation function to minimize vanishing gradient problem. In total we used 2 hidden with 256 units for InceptionV3 and 512 units for VGG16 as more hidden layers increase overfitting. We find that 30 epochs for InceptionV3 and 3 epochs for VGG16 are good enough to fetch compelling results.

4.4 Evaluation Metrics

While dealing problems involving Natural Language Processing tasks, there is never have been a standard and fixed evaluation policy is adopted. In contrast there are multiple algorithms developed from the inferences of similarity of sentences. Few of them are Consensus-based Image Description Evaluation(CIDEr), Metric for evaluation of Translation with Explicit Ordering(METEOR), Recall-Oriented Understudy for Gisting Evaluation(ROUGE) and BLEU stands for bilingual evaluation policy[13]. For our model we used BELU[13] evaluation algorithm since it's been in wide use lately.

- BLEU

"**BLEU** (**bilingual evaluation understudy**) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU." - Wikipedia



Candidate 1: the the the the the the the.
Candidate 2: the cat is mat the on
Reference: The cat is on the mat.

BELU Metric: Comparing generated captions with reference

Basically BELU compares the generated text's words with reference sentence words no matter at what positions they are in. Here we present our models BELU score for both VGG16 and InceptionV3. According to our approach InceptionV3 performed better than VGG16 in captioning images. Our model's BLEU score was way better than many research paper models with the task of captioning images. Although our model's score is not the best in state-of-the-art results, it's well within the boundaries of compelling BELU results.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
VGG16+LSTM	0.520439	0.336081	0.242425	0.1302
InceptionV3+LSTM	0.5112	0.3320	0.2472	0.13467

5. Conclusion

All in all, we are able to present a model that is thoroughly trained and overwhelmingly & computationally expensive (owing to the quality of gpu's available) caption generator. Although our model isn't with the disguise of captivating state-of-the-art neural networks owing to the fact of smaller dataset and computationally expensive task of the problem, we generated some effective captions from our model that met well with context of images mingled with both semantic and syntactic understandings of the captions. At the same note, our BELU scores are well within the range of the state-of-the-art results.

6. Future Work

Further tuning to our model could be done by providing large quantity of datasets like flikr30k, Microsoft Coco dataset. Another experiment can be employed by extracting image features from well-known pre-trained CNN models like Resnet, Xception, Googlenet models. Instead of using Glove6b for word2vec, we can also use Tokenizer class from keras module to tokenize the captions from scratch and related results should be noted. As a fun concept we could use fusion approach where we could merge InceptionV3 and VGG16 features to train our model.

References

[1] Keiron O'Shea and Ryan Nash 'An Introduction to Convolutional Neural Netowrks', 2015

[2] Sepp Hochreiter, JurgenSchemidhuber, Corso Elvezia 'Long Short-term Memory', 1997

[3] Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Shikha Jain 'Machine translation using deep learning: An overview', 2017

[4] Jeffrey Pennigton, Richard Socher, Christopher D. Manning 'Glove: Global Vectors for Word Represenation'

[5] https://en.wikipedia.org/wiki/Flickr

[6] Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in ECCV, 2010.

[7] Hao Fang, Saurabh Gupta, Forrest landola, Rupresh Srivastava, Li Deng, Piotr Dollar, Jianfeng Gao 'From Captions to Visual Concepts and Back', 2015

[8]Oriol Vinyals, Alexander Tosheve, Samy Bengio, Dumitru Erhan "Show and Tell: A Neural Image Caption Generator", in CVPR 2015

[9] Kelvin Xu, Jimmy Ba, Ryan kiros, Kyunghyun Cho 'Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.

[10] keras.io/api/applications/inceptionv3

[11] machinelearningmastery.com/transfer-learning-for-deep-learning

[12] keras.io/api/applications/vgg/#vgg16-function

[13] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu 'BLEU: a method for Automatic Evalation of Machine Translation'

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. IEEE Trans. Pattern Anal. Mach. Intell., 39(4):664–676, Apr. 2017.