Graphic Era Deemed to be University

Dehradun, Uttarakhand

A Mini Project Report

on

IMAGE CAPTIONING

Submitted by:

SARTHAK SAHAI

B. Tech CSE

Semester III

2017000

Department of Computer Science and Engineering

FEBRUARY,2022

PROBLEM STATEMENT:

Using natural language to automatically describe the content of photographs is a fundamental and difficult task. Building models that can create captions for a image has become viable thanks to advances in processing power and the availability of large datasets. Humans, on the other hand, can simply explain the environments in which they find themselves. It's natural for someone to describe an enormous quantity of information about a image in a single glance. Although much progress has been made in computer vision, tasks such as object recognition, action classification, image classification, attribute classification, and scene recognition are now possible, allowing a computer to describe an image in the form of a human-like sentence is still a relatively new task.

INTRODUCTION:

Caption creation is a fascinating artificial intelligence challenge that involves generating a descriptive text for a given image. It uses two computer vision approaches to comprehend the image's content, as well as a language model from the field of natural language processing to convert the image's comprehension into words in the correct order. Image captioning has a variety of uses, including recommendations in editing software, use in virtual assistants, image indexing, accessibility for visually impaired people, social media, and a variety of other natural language processing applications. On examples of this problem, deep learning approaches have recently obtained state-of-the-art results. Deep learning models have been shown to be capable of achieving optimal outcomes in the realm of caption generating challenges. A single end-to-end model can be defined to predict a caption given a photo, rather than requiring sophisticated data preparation or a pipeline of separately designed models. These findings demonstrate that our suggested model outperforms traditional models in terms of image captioning in performance evaluation. The limitations of neural networks are determined mostly by the amount of memory available on the GPUs used to train the network as well as the duration of training time it is allowed. Our network takes around seven days to train on GTX 1050 4GB and GTX 760 GPUs. According to our results, our results can be improved by utilizing faster and larger GPUs and more exhaustive datasets.

OBJECTIVE:

Image captioning seeks to construct a sentence description for an image automatically. Our project model will take an image as input and output an English sentence that describes the image's contents. In recent years, it has gained a lot of study attention in the field of cognitive computing. The challenge is difficult because it combines concepts from both the computer vision and natural language processing fields. We have developed a model using the concepts of a Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model and build a working model of Image caption generator by implementing CNN with LSTM.

MOTIVATION:

We must first comprehend the significance of this issue in real-world circumstances. Let's look at a couple scenarios when a solution to this problem could be quite valuable.

- Self-driving automobiles — Automatic driving is one of the most difficult difficulties and captioning the area around the car can help the self-driving system.
- Aid for the blind — We can develop a product for the blind that will lead them on the roads without the need for anyone else's assistance. This can be accomplished by first turning the scene to text, then the text to speech. Both are now well-known Deep Learning applications.
- CCTV cameras are now ubiquitous, but if we can provide appropriate captions in addition to watching the world, we can trigger alarms as soon as criminal activity is detected somewhere. This is likely to help minimize crime and/or accidents.

DATASET:

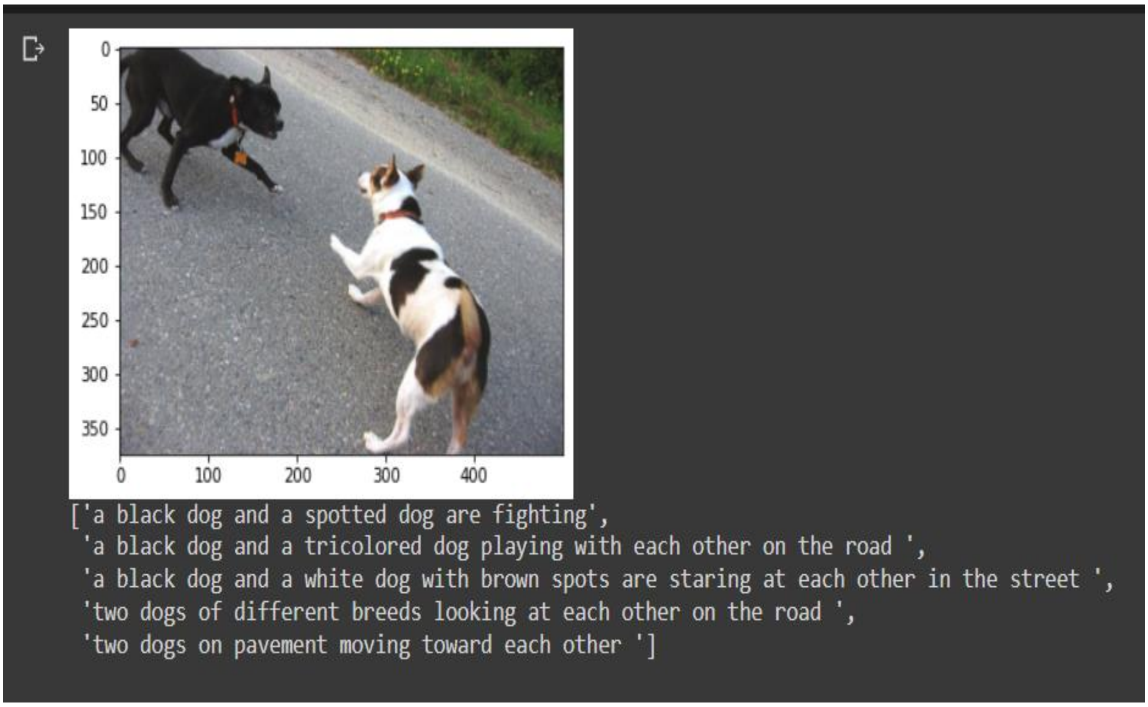
Flickr8k is a nice starting dataset because it is small and can be trained with a CPU on low-end laptops/desktops.

Each image in the Flickr8k dataset is accompanied by five different descriptions that describe the entities and events depicted in the image. The dataset captures some of the linguistic variability that can be used to describe the same image by connecting each image with different, independently produced words.

The following is the structure of our dataset: The 8000 photos are contained in Flick8k/ Flick8k Dataset/.

Flick8k Text/ Flickr8k.token.txt: This file contains the image id as well as the five captions.

The training image ids are stored in Flickr8k.trainImages.txt.



```
['a black dog and a spotted dog are fighting',
 'a black dog and a tricolored dog playing with each other on the road ',
 'a black dog and a white dog with brown spots are staring at each other in the street ',
 'two dogs of different breeds looking at each other on the road ',
 'two dogs on pavement moving toward each other ']
```

The test image ids are stored in Flickr8k.testImages.txt.

SYSTEM ARCHITECTURE:

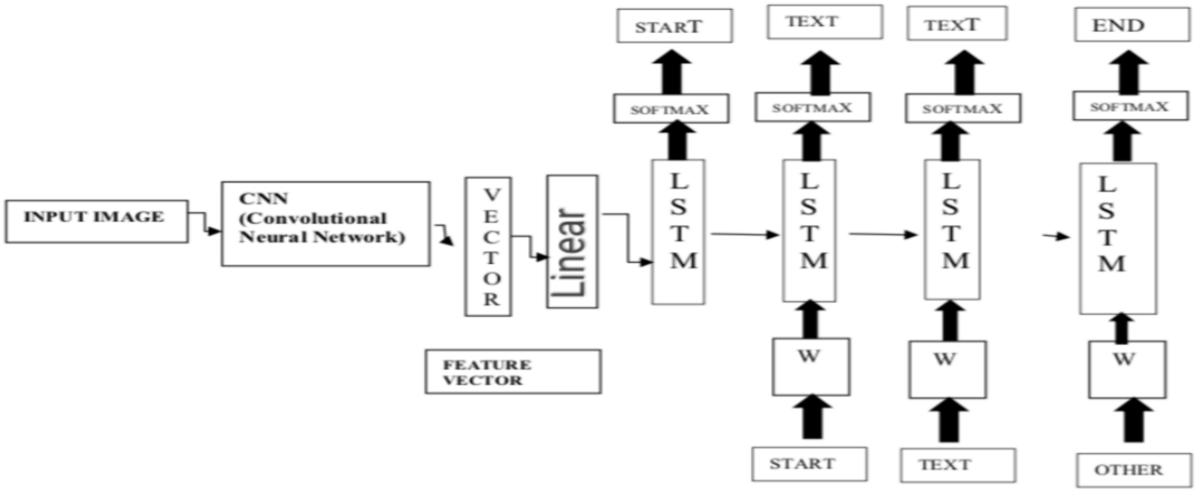


Figure 1: Proposed Model of Image Caption Generator

The proposed Image Caption Generator model is depicted in Figure 1 above. The input image is presented in this model, and then as indicated in the illustration, a convolutional neural network is utilized to build a dense feature vector. This dense vector, also known as an embedding, can be utilized as an input to various algorithms and creates appropriate captions for a given image as an output.

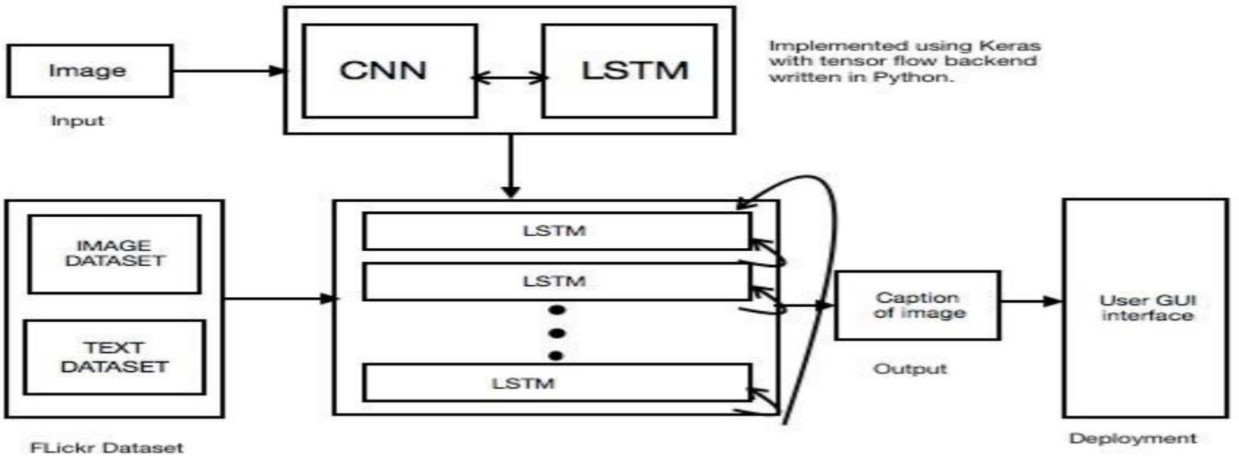


Figure 2: System Architecture of Image Caption Generator

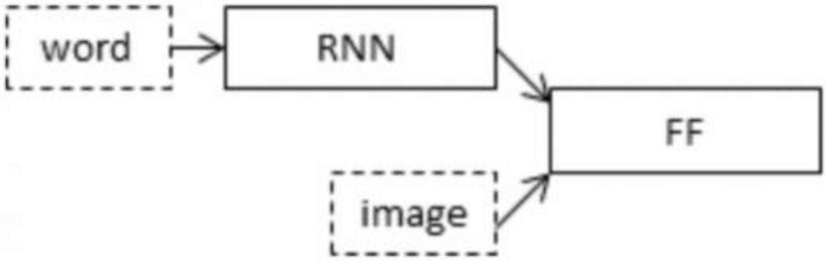
This embedding becomes a representation of the image for an image caption generator, and it is utilized as the LSTM's initial state for creating relevant captions for the image. Figure 2 depicts the system architecture of our system. The following is a diagram of our suggested system design.

METHODOLOGY:

To encode text sequences of variable lengths, our model will use CNN as the 'image model' and RNN/LSTM as the 'language model.' To make a final prediction, the vectors generated from both encodings are mixed and processed by a Dense layer.

We'll use a merge architecture to keep the image out of the RNN/LSTM, allowing us to train the image-handling and language-handling parts of the neural network separately, utilizing images and words from distinct training sets.

Before each prediction, we can integrate a distinct representation of the image with the final RNN state in our merge model.



The above diagram is a visual representation of our approach.

The blending of image features with text encodings at a later level in the architecture is beneficial, since it can result in higher-quality captions with fewer layers than the standard inject architecture (CNN as encoder and RNN as a decoder).

Transfer learning will be used to encode our image features. We can employ a variety of models, including VGG-16, InceptionV3, ResNet, and others.

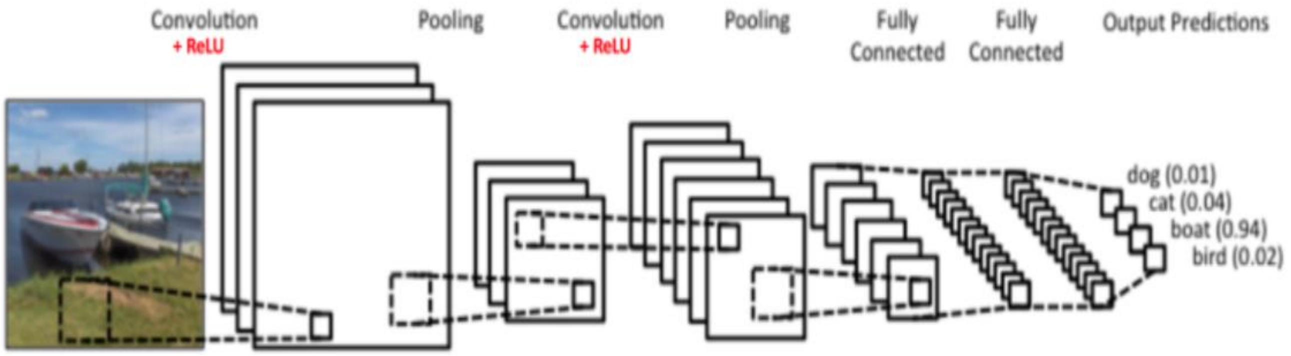
We'll utilize the inceptionV3 model, which has the fewest training parameters of the three and outperforms them all.

We'll map each word to a 200-dimensional vector to encode our text sequence. A pre-trained Glove model will be used for this. After the input layer, a separate layer called the embedding layer will be used to map the data.

ALGORITHMS:

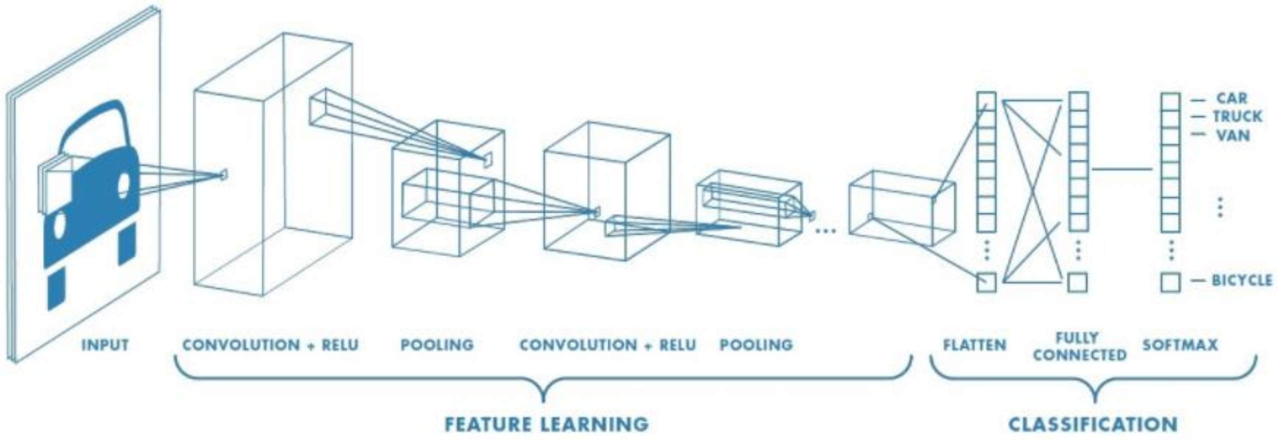
CONVOLUTIONAL NEURAL NETWORK:

Convolutional neural networks are customized deep neural networks that process data in the form of a 2D matrix as input. CNNs perform well with images and may be represented as a two-dimensional matrix. CNN is a powerful tool for image classification and identification. It can tell whether a image is of a bird, a plane, or Superman, for example. By scanning an image from left to right and top to bottom, important features can be recovered, and the features can then be merged to classify images. It can handle photos that have been rotated, resized, then rotated again, as well as changes in perspective.



A SIMPLE CONVET ARCHITECTURE

To train and evaluate deep learning CNN models, each input image will be passed through a sequence of convolution layers using filters (Kernals), Pooling, fully connected layers (FC), and the SoftMax function to classify an object with probabilistic values ranging from 0 to 1. The flow of CNN to process an input image and classify objects based on values is depicted in the diagram below.



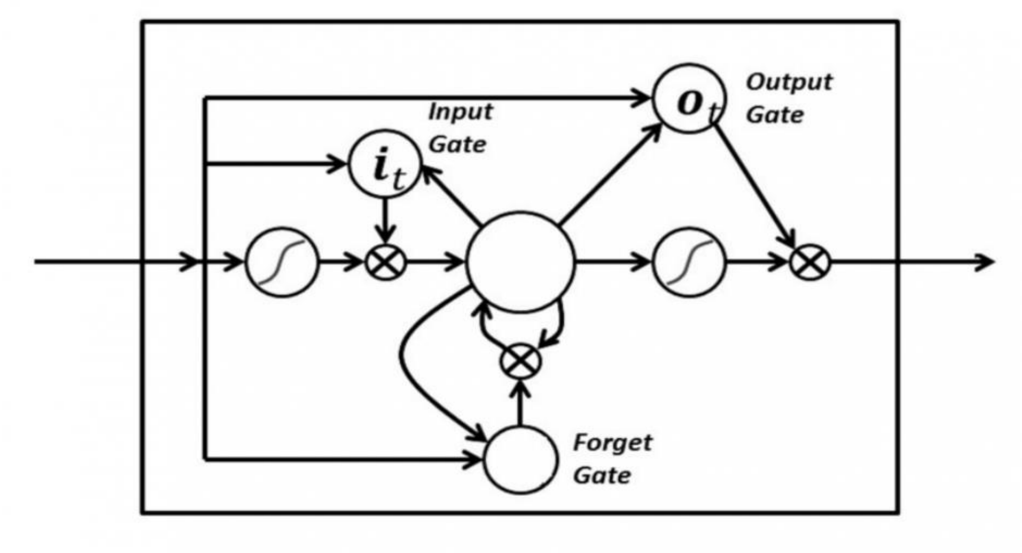
Long Short-Term Memory:

LSTMs are a sort of RNN (recurrent neural network) that excels at sequence prediction. Based on the previous paragraph, we can guess what the next words will be. By addressing the restrictions of RNN, it has proven to be more effective than regular RNN. LSTM can keep track of useful data throughout the processing and eliminate irrelevant data.

RNNs can recall inputs for a long time because to LSTMs. This is because LSTMs store information in a memory similar to that of a computer. The LSTM has the ability to read, write, and delete data from its memory.

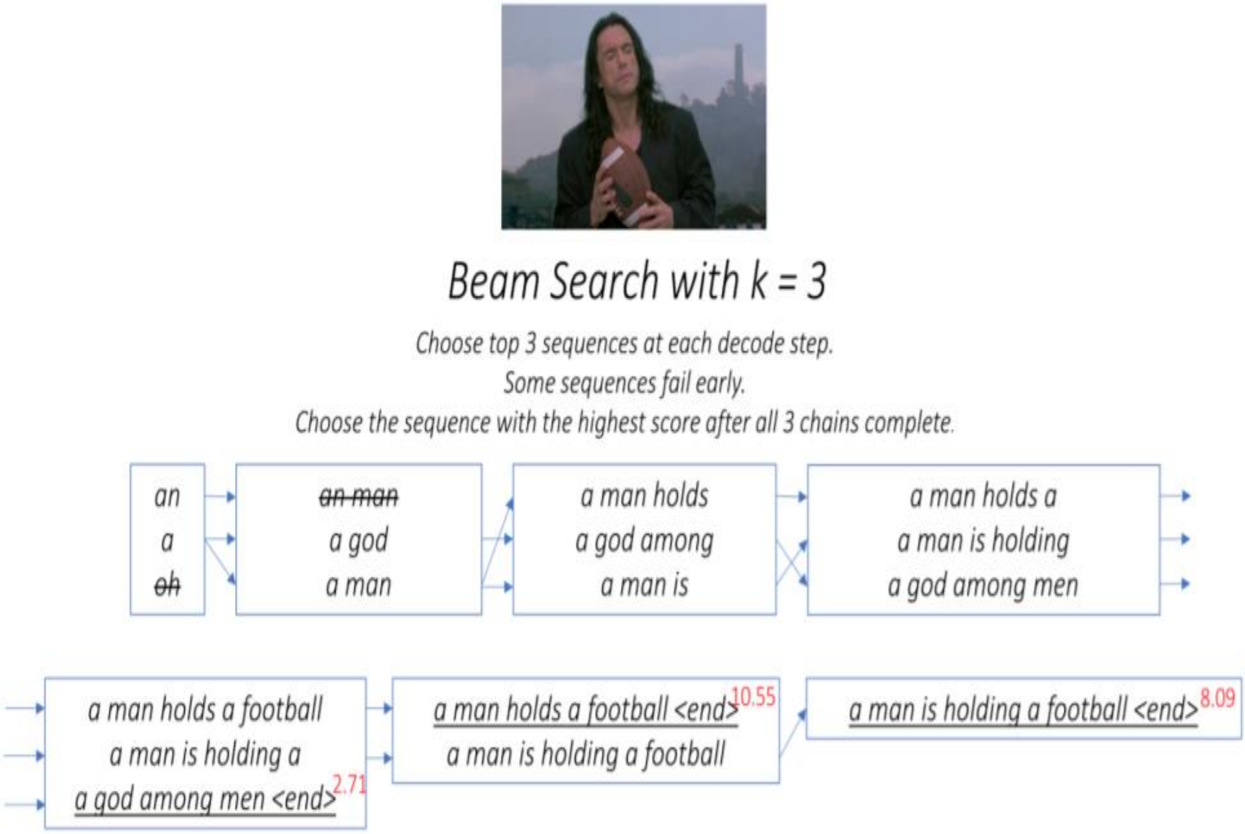
This memory can be thought of as a gated cell, with gated indicating that the cell selects whether or not to store or erase information (i.e., whether or not to open the gates) based on the value it gives to the data. Weights, which are also learned by the algorithm, are used to allocate importance. This basically implies that it learns what information is important over time and what information is not.

There are three gates in an LSTM: input, forget, and output. These gates determine whether fresh input should be allowed (input gate), whether it should be deleted because it isn't important (forget gate), or whether it should have an impact on the output at the current timestep (impact gate) (output gate).



BEAM SEARCH:

Beam Search is presented as the final step in the Show and Tell paper [to generate a sentence with the highest likelihood of occurrence given the input image. The algorithm is a best-first search algorithm that iteratively considers the set of the k best sentences up to time t as candidates for generating sentences of size t + 1, keeping only the best k of them because this better approximates the probability of getting the global maximum


Beam Search with k = 3

Choose top 3 sequences at each decode step.
Some sequences fail early.
Choose the sequence with the highest score after all 3 chains complete.

as stated in the paper. We experimented with beam search sizes ranging from 1 to 5 and found that beam sizes 3 and 4 have the best assessment. As a result, we decided on 3 as the beam search size.

The beam search size was set at 3. A decent demonstration diagram may be found in Figure, which was pulled from a GitHub source. This diagram illustrates how beam search avoids selecting the highest probability word at each step (local maximum) in favor of selecting the sequence of words with the highest overall probability score (global maximum).

MODEL BUILDING AND TRAINING:

As you can see from our method, we chose to use the InceptionV3 network, which was pre-trained on the ImageNet dataset, for transfer learning.

It's important to note that we don't need to classify the images here; all we need to do is extract an image vector. As a result, the softmax layer is removed from the inceptionV3 model.

We need to pre-process our input before feeding it into the model because we're using InceptionV3. As a result, we create a preprocess function that resizes the photos to (299 x 299) and feeds it to Keras preprocess input() function.

INCEPTION v3 MODEL:

By changing earlier Inception architectures, Inception v3 primarily focuses on burning less processing power. Inception Networks (GoogLeNet/Inception v1) have been shown to be more computationally efficient than VGGNet, both in terms of the amount of parameters created by the network and the cost incurred (memory and other resources). Several strategies for improving the network have been proposed in an Inception v3 model to loosen the constraints for easier model adaption.

GLOVE EMBEDDING:

Word vectors convert words into a vector space in which like words are grouped together and distinct words are separated. Glove has an advantage over Word2Vec in that it does not rely just on the local context of words to generate word vectors, but also takes into account global word co-occurrence.

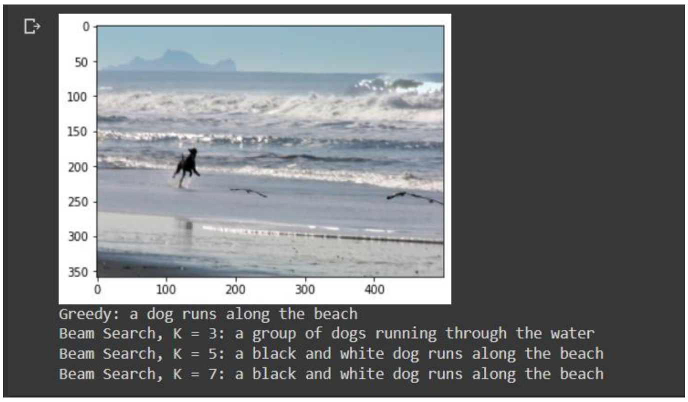
Glove's main idea is that the co-occurrence matrix can be used to infer semantic associations between words. We'll use Glove to map all of the words in our 38-word caption to a 200-dimension vector for our model.

GloVe has the benefit that, unlike Word2vec, it does not rely solely on local statistics (local context information of words) to generate word vectors, but also incorporates worldwide statistics (word co-occurrence).

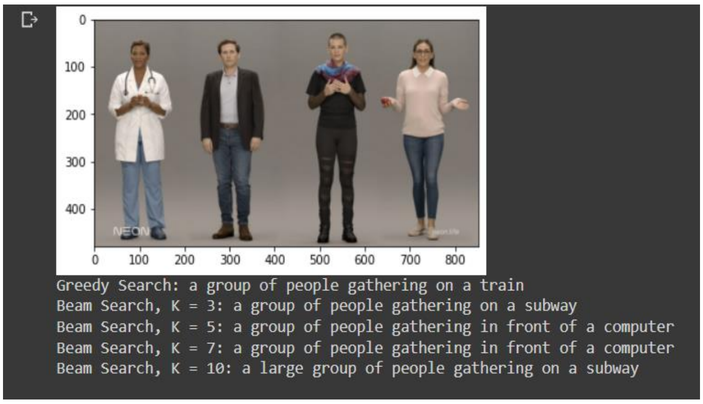
The primary idea behind the GloVe word embedding is to use statistics to derive the link between the words. The co-occurrence matrix, unlike the occurrence matrix, informs you how frequently a specific word pair appears together. A pair of words that occur together is represented by each value in the co-occurrence matrix.

RESULT:

After the data has been eaten by the pipeline, the output will be in an encoded format that must be reversed back into English words before it can be understood by humans. Another significant point is that the RNN network's output is a series of likelihoods of words (likelihoods). In RNN, selecting the highest likelihood word at each decode stage tends to produce a sub-optimal output. Instead, we've used Beam Search, which was first proposed in and is a popular method for determining the best path for decoding natural language sentences. Below are some examples of captions generated from test photographs.

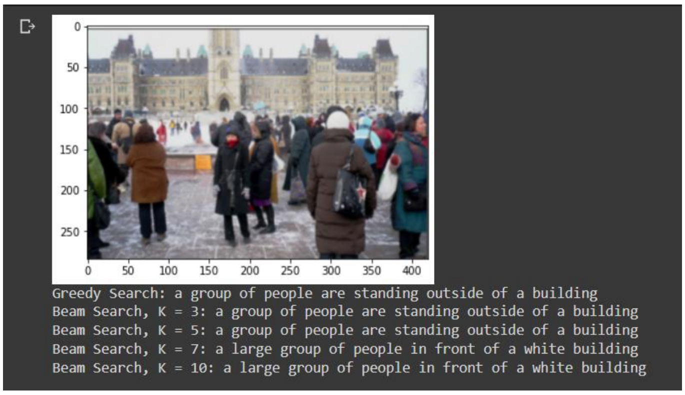


```
Greedy: a dog runs along the beach
Beam Search, K = 3: a group of dogs running through the water
Beam Search, K = 5: a black and white dog runs along the beach
Beam Search, K = 7: a black and white dog runs along the beach
```

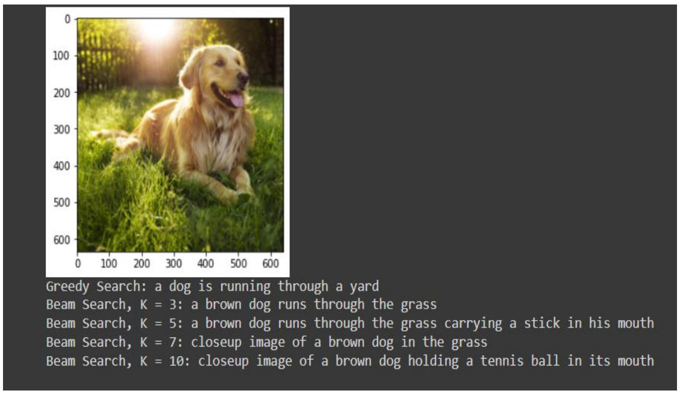


```
Greedy Search: a group of people gathering on a train
Beam Search, K = 3: a group of people gathering on a subway
Beam Search, K = 5: a group of people gathering in front of a computer
Beam Search, K = 7: a group of people gathering in front of a computer
Beam Search, K = 10: a large group of people gathering on a subway
```

We can see that we accurately explained what happened in the image in this section. You'll also notice that the captions supplied by Beam Search are far superior to those generated by Greedy Search.



```
Greedy Search: a group of people are standing outside of a building
Beam Search, K = 3: a group of people are standing outside of a building
Beam Search, K = 5: a group of people are standing outside of a building
Beam Search, K = 7: a large group of people in front of a white building
Beam Search, K = 10: a large group of people in front of a white building
```



```
Greedy Search: a dog is running through a yard
Beam Search, K = 3: a brown dog runs through the grass
Beam Search, K = 5: a brown dog runs through the grass carrying a stick in his mouth
Beam Search, K = 7: closeup image of a brown dog in the grass
Beam Search, K = 10: closeup image of a brown dog holding a tennis ball in its mouth
```

CONCLUSION:

Automatic image captioning is still in its infancy, and there are a slew of ongoing research initiatives aimed at improving image feature extraction and generating semantically superior sentences. Due to restricted processing resources, we were able to finish all we indicated in the project proposal, but we had to use a smaller dataset (Flickr8k). If given more time, there may be room for improvement.

If given more time, there may be room for improvement. First and foremost, we employed a pre-trained CNN network without fine-tuning as part of our workflow, therefore the network does not respond to this specific training dataset. As a result, by testing with several CNN pre-trained networks and allowing for fine-tuning.

Another way to improve is to use a mix of Flickr8k, Flickr30k, and MSCOCO to train. The more diversified the training dataset the network has seen, the more accurate the output will be in general.