Image Captioning

Minor Project(Final Report)

Harshal Bhardwaj 18001003039
Harshit Garg 18001003040

Introduction

Image Captioning is the process of generating textual description of an image. It uses both Natural Language Processing and Computer Vision to generate the captions.

A young boy is playing basketball. 	Two dogs play in the grass. 	A dog swims in the water. 
A group of people walking down a street. 	A group of women dressed in formal attire. 	Two children play in the water. 

Motivation

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.

> Self driving cars —

Automatic driving is one of the biggest challenges and if we can properly caption the scene around the car, it can give a boost to the self driving system

> Automatic Captioning –

can help, make Google Image Search as good as Google Search, as then every image could be first converted into a caption and then search can be performed based on the caption.

>CCTV cameras –

(Closed-circuit television cameras) are everywhere today, but along with viewing the world, if we can also generate relevant captions, then we can raise alarms as soon as there is some malicious activity going on somewhere. This could probably help reduce some crime and/or accidents.

DATA COLLECTION

There are many open source datasets available for this problem, like Flickr 8k (containing8k images), Flickr 30k (containing 30k images), MS COCO (containing 180k images), etc.

But for the purpose of this case study, I have used the Flickr 8k dataset which you can download from kaggle. Also training a model with a large number of images may not be feasible on a system which is not a very high end PC/Laptop.

This dataset contains 8000 images each with 5 captions (as we have already seen in the Introduction section that an image can have multiple captions, all being relevant simultaneously).

These images are bifurcated as follows:

- Training Set — 6000 images
- Dev Set — 1000 images
- Test Set — 1000 images

Major Libraries and modes used….

Libraries

- Pandas
- Numpy
- Matplotlib
- Tensorflow
- Keras

Models

- Convolutional neural network(CNN)
- Recurrent Neural network(RNN)

CNN

Convolutional Neural Networks, or CNNs, were designed to map image data to an output variable

A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution.

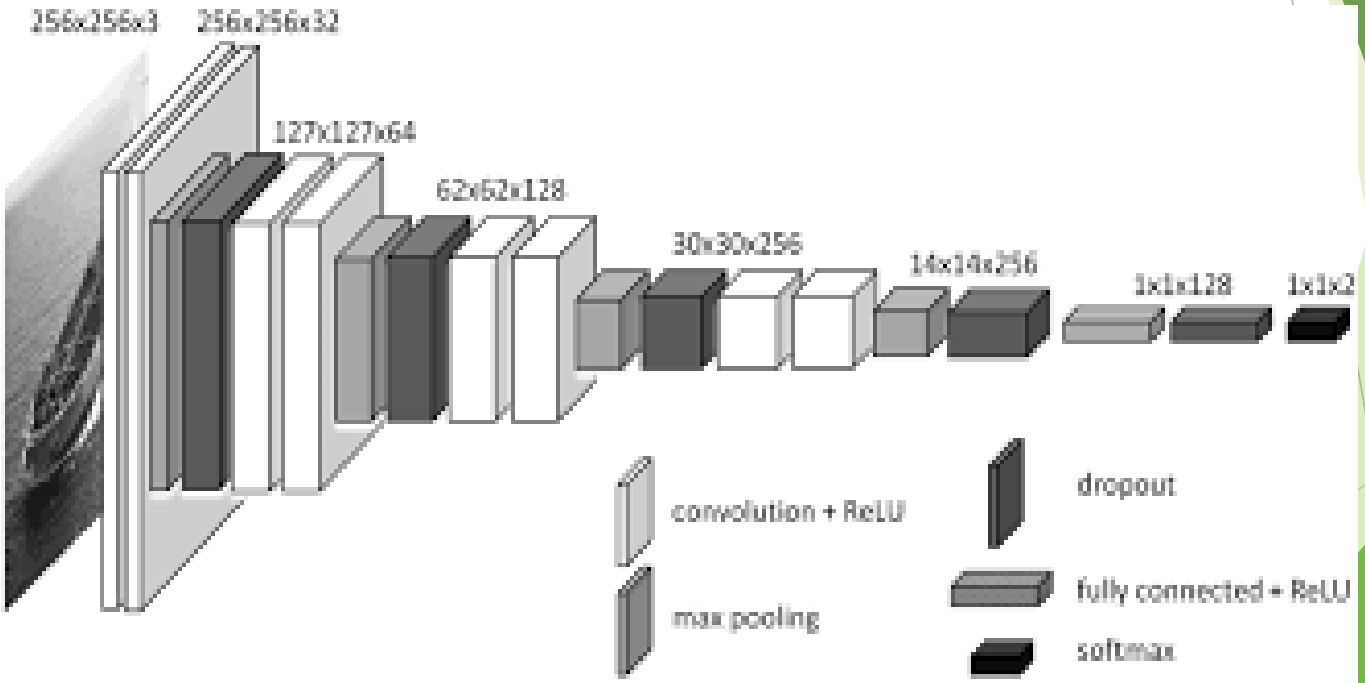
RNN

RNNs in general and LSTMs in particular have received the most success when working with sequences of words and paragraphs, generally called natural language processing.

This includes both sequences of text and sequences of spoken language represented as a time series. They are also used as generative models that require a sequence output, not only with text, but on applications such as generating handwriting.

Use RNNs For:

- Text data
- Speech data
- Classification prediction problems
- Regression prediction problems
- Generative models


256x256x3
256x256x32
127x127x64
62x62x128
30x30x256
14x14x256
1x1x128
1x1x2
convolution + ReLU
max pooling
dropout
fully connected + ReLU
softmax

STEPS :-

- **1. DATA COLLECTION :-**

- Data is collected in the form of csv file from Kaggle and captions are
- read from csv file.
- And a dictionary is created to map all the 5
- captions to the image.

- **2. DATA CLEANING:-**

- Convert all words to lowercase.

- Remove all punctuation.

- Remove all words that are one character or less in length (e.g. ‘a’).

- Remove all words with numbers in them.

- **3. GENERATING VOCABULARY:-**

- creating a set of all unique words and a freaquency map of all words is created and all the words
- with frequency less than a minimum threshold is removed.

4. Prepare Training and test data

Caption of giving training data is linked to images

5.Feature Extraction:

Transfer learning using RESNET50 model is used to extract features.

6. LSTM(long short term memory):

Used for caption generation.

7. Training of dataset

8. Prediction

Concepts Used

1. Encoder
2. Decoder
3. Attention
4. Transfer Learning
5. Beam Search

Encoder

The Encoder encodes the input image with 3 color channels into a smaller image with "learned" channels.

This smaller encoded image is a summary representation of all that's useful in the original image Model will progressively create smaller and smaller representations of the original image, and each subsequent representation is more "learned", with a greater number of channels. The final encoding produced by our ResNet-101 encoder has a size of 14x14 with 2048 channels, i.e., a 2048, 14, 14 size tensor.

Decoder

The Decoder's job is to look at the encoded image and generate a caption word by word.

Since it's generating a sequence, it would need to be a Recurrent Neural Network (RNN). We will use an LSTM.
we want the Decoder to be able to look at different parts of the image at different points in the sequence. For example, while generating the word football in a man holds a football, the Decoder would know to focus on – you guessed it – the football!

Beam Search

- At the first decode step, consider the top k candidates.
- Generate k second words for each of these k first words
- Choose the top k [first word, second word] combinations considering additive scores.
- For each of these k second words, choose k third words, choose the top k [first word, second word, third word] combinations.
- Repeat at each decode step.
- After k sequences terminate, choose the sequence with the best overall score.

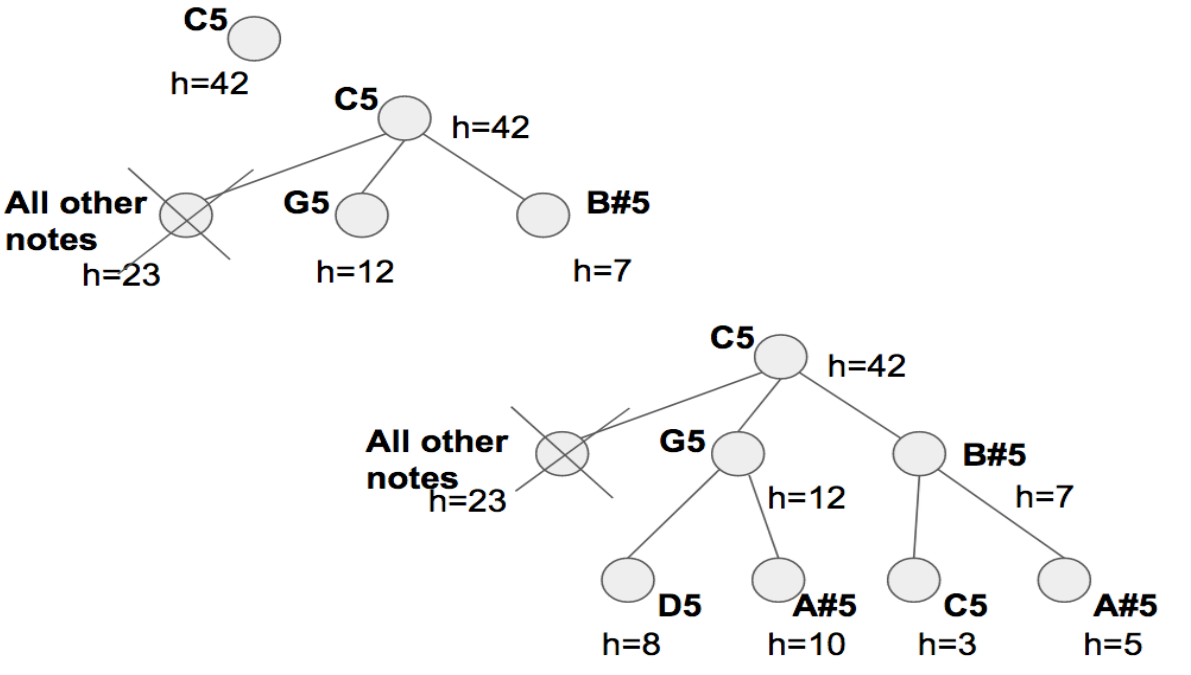
Attention

The Attention network computes these weights

Intuitively, how would you estimate the importance of a certain part of an image? You would need to be aware of the sequence you have generated so far, so you can look at the image and decide what needs describing next. For example, after you mention a man, it is logical to declare that he is holding a football.

This is exactly what the Attention mechanism does – it considers the sequence generated thus far, and attends to the part of the image that needs describing next.

Beam Search


C5
h=42
C5
h=42
All other notes
h=23
G5
h=12
B#5
h=7
C5
h=42
All other notes
h=23
G5
h=12
B#5
h=7
D5
h=8
A#5
h=10
C5
h=3
A#5
h=5

Implementation

We will need three inputs

Images:- images fed to the model must be a Float tensor of dimension N, 3, 256, 256, and must be normalized by the aforesaid mean and standard deviation. N is the batch size.

Caption:-Captions are both the target and the inputs of the Decoder as each word is used to generate the next word.

Caption Length:-Since the captions are padded, we would need to keep track of the lengths of each caption.

We only process a sequence upto its length and don't waste computation

Therefore, caption lengths fed to the model must be an Int tensor of dimension N.

Prediction

A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A skateboarder does a trick on a ramp.

A dog is jumping to catch a frisbee.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

A little girl in a pink hat is blowing bubbles.

A refrigerator filled with lots of food and drinks.

A herd of elephants walking across a dry grass field.

A close up of a cat laying on a couch.

A red motorcycle parked on the side of the road.

A yellow school bus parked in a parking lot.