Image Caption Generator

Deon Saji & Mayank Singla
111901022 & 111901030
Bachelor of Technology
Computer Science and Engineering
IIT Palakkad, 2021

Preface

This report is the outcome of the internship program with UST Global. The tenure of this internship program was from June 2021 to July 2021. The main objective of the internship was to understand and familiarize us with topics related to Natural Language Processing. We were allowed to work in the field of Image Captioning and in this report, we have mentioned all our learnings and researches done during the internship phase. In this report, we have added different sections explaining Problem Definition, Brief idea of the topics, Approach, Results, Conclusion, and Future Improvements. We have explained our project in simple words hoping that the readers find it easy to understand and get an essence of the topics. Though we have tried our best to keep this report free from errors, we apologize for any errors that are found.

Acknowledgment

First, we would like to thank our mentor, Hanish Sargiya. He was available during this internship period giving us correct directions at each stage. He was ready to provide resources and clear all our doubts which we had since we had no prior knowledge of these topics. Online meetings and discussions that were scheduled frequently really helped us in working as a team. Next, we would like to thank Mr. Ashok G Nair, who provided all the support and guidance which was important for the successful completion of the project. His comments at the final presentation motivated us to take related project topics in the future.

We would also like to thank our primary point of contact of the company, Gopika Bindu. She helped us in getting connected with the mentors and other company officials. She helped us a lot in smoothly taking the internship to move forward.

We are extremely thankful to the UST team for doing their job in the best possible way.

Abstract

This report describes the internship we spent at UST Global. The main work is to understand the present state-of-the-art techniques applied in Image Captioning and to tweak some parameters of the state-of-the-art technology to improve its performance. The more challenging and at the same time the most interesting part of the task is that there is no limit to extending this task. There are a lot of variables that can be adjusted to get better performance or worse performance. Even the slightest mistake would steer off the road that is intended to take. The present state-of-the-art technique to efficiently caption an image was by Andrej Karpathy. The idea was to predict the next word in the sequence given the image and sequence of words. This idea closely relates to the concept of conditional probability. Initially, that state-of-the-art technology is used and a deep learning model is developed and even trained on the Flickr 8k dataset. Later the architecture of the deep learning model, transfer learning of the deep learning model, and metric of the deep learning model is tweaked to get better performance. There were some improvements made on the course of 8 intermediate models, hours of training for each model, millions of parameters in each model. Considering the technological infrastructure available and the time constraint of 8 weeks, the result of the work is pretty convincing.

A perfect understanding of transfer learning can be drawn from this work. There are multiple areas where transfer learning is applied, to enhance the performance of the model. As transfer learning requires no additional training time, the efficiency is increased compared with the side effects it has.

Introduction

Computer vision is one of the areas that has been advancing rapidly thanks to deep learning. Tasks that earlier seemed very difficult for machines are now easily done by them. Today deep learning computer vision helps in self-driving cars and figuring out different objects around.

Image classification/recognition, object detection, and localization are the three important computer vision problems. This helps to make machines look at objects and arrive at conclusions. Unlike humans, machines use numbers to interpret each image. We need to train them to understand different numbers and this is challenging.

Image Captioning is the process of generating a textual description of an image. It uses both **Natural Language Processing** and **Computer Vision** to generate the captions. Image modeling comes under the Computer Vision task and Language modeling comes under the NLP task. We use an encoder-decoder architecture in which a **deep convolutional neural network (CNN)** is trained to classify the image (**encoder**) and a **Recurrent neural network** is used to decode the caption(**decoder**).

Applications

- Help blind people by generating captions of images of the view they see.
- Search photos by searching their generated captions. (Ex. Google Photos)
- Provision of captions to provide HTML header and “alt” attribute to improve Search Engine scoring of the page for search terms related to the content of the movie or image(in Web Development).
- Use in virtual assistants.
- In Social Media (Ex. Facebook auto detects friends from images of posts we share and notify those friends)

Problem definition

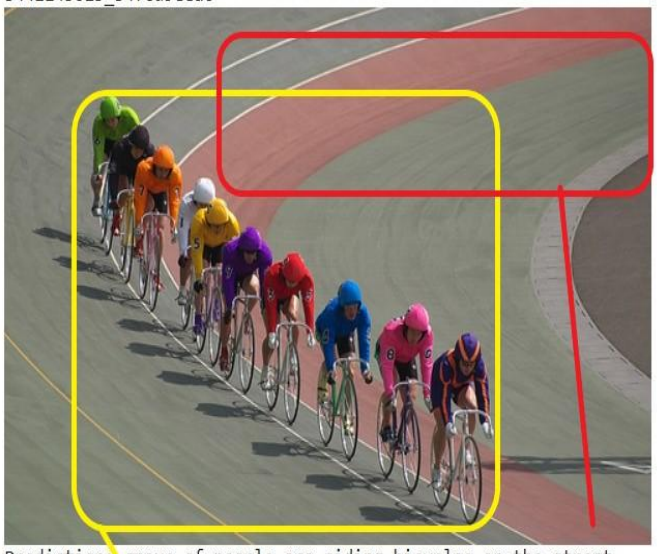
The problem is to caption images which require a computer vision system to detect and localize images and then describe the image. Training on a set of images and their captions and using the attained knowledge to correctly predict new images, accounting for syntax as well as the semantics of the sentence. Input to the model is images and their corresponding captions(here five captions describing each image are input). And the output of the model is text describing the corresponding input image. Flickr 8k dataset was used for this project considering that a huge dataset could be computationally expensive. The dataset consists of 8,000 images, each with 5 different captions. This dataset is split into 6000, 2000, 1000 images each for the training dataset, dev dataset, and test dataset respectively.

Observations

Prediction: little boy in blue shorts jumps into pool

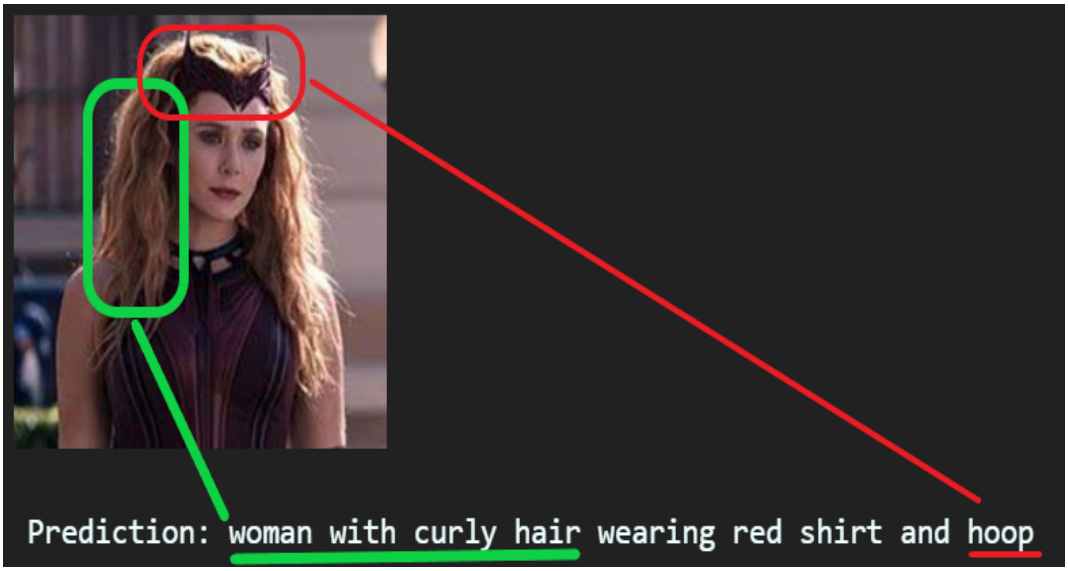
Actual Description: boy holding large foam wand waves it near swimming pool
Actual Description: boy plays with foam noodle toy by pool
Actual Description: boy plays with noodle by the pool
Actual Description: child with water noodle
Actual Description: little boy wearing blue shorts is holding yellow noodle over pool

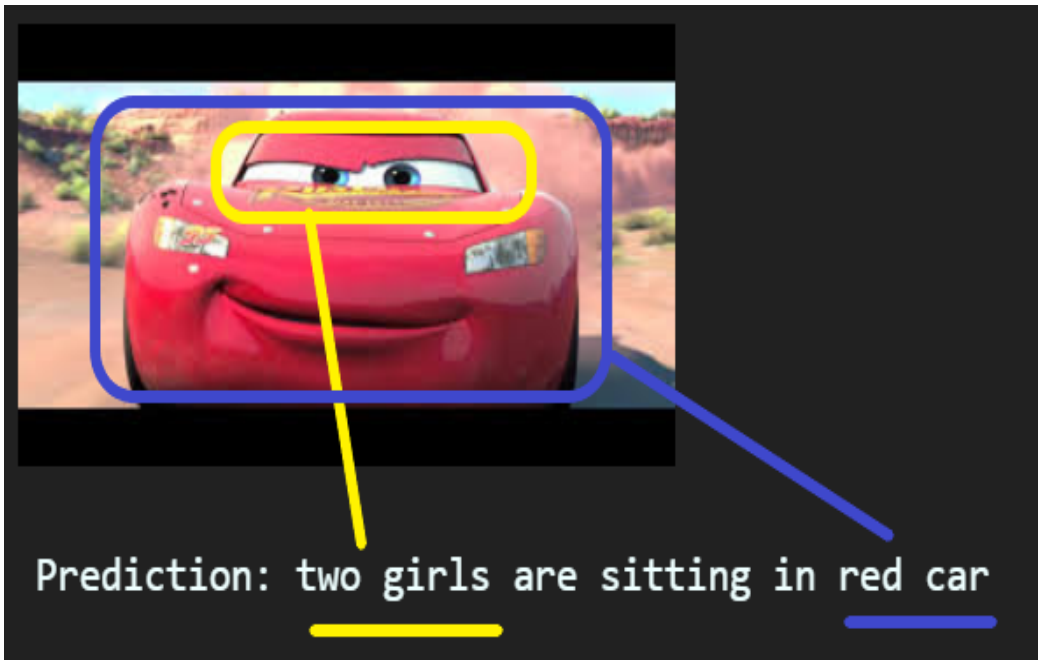
Inception Model
3441145615_b4fcd9eea0



Prediction: group of people are riding bicycles on the street

Actual Description: group of cyclers race around track
Actual Description: group of people in colored outfits ride bikes around track
Actual Description: bicyclists stay in line as each wear different color suits
Actual Description: multiple bicyclists wearing different colored shirts and helmets riding around track
Actual Description: ten cyclists in different colors are racing around bend in the track


Prediction: woman with curly hair wearing red shirt and hoop


Prediction: two girls are sitting in red car

ResNet Model
236730743_0d4fd8de5a

Prediction: man paddles canoe on the water

ResNet Model
3040033126_9f4b88261b

Prediction: the dog is walking through the water

ResNet Model
544576742_283b65fa0d

Prediction: man in red shirt is rock climbing

Prediction: man in black shirt and cap is holding up cup

ResNet Model
3485425825_c2f3446e73

Prediction: man in blue shirt and beard pants is riding bike

ResNet Model
3619416477_9d18580a14

Prediction: white dog is running through shallow water

ResNet Model
2706766641_a9df81969d

Prediction: baby in red shirt is sitting on wooden floor

ResNet Model
2891617125_f939f604c7

Prediction: man on motorcycle rides on dirt road

ResNet Model
3052196390_c59dd24ca8

Prediction: girl in swimsuit is pointing at the water fountain

ResNet Model
3710520638_866d542a80

Prediction: black dog is running through the water

ResNet Model
3227148358_f152303584

Prediction: bird descending water

ResNet Model
2208067635_39a03834ca

Prediction: two girls are playing in the leaves

ResNet Model
150387174_24825cf871

Prediction: man in helmet riding bike down the road

ResNet Model
143688283_a96ded20f1

Prediction: boy in red shirt is climbing rock wall

ResNet Model
3670907052_c827593564

Prediction: man riding bicycle down dirt hill

ResNet Model
2340206885_58754a799a

Prediction: two dogs are playing in the snow

ResNet Model
3568197730_a071d7595b

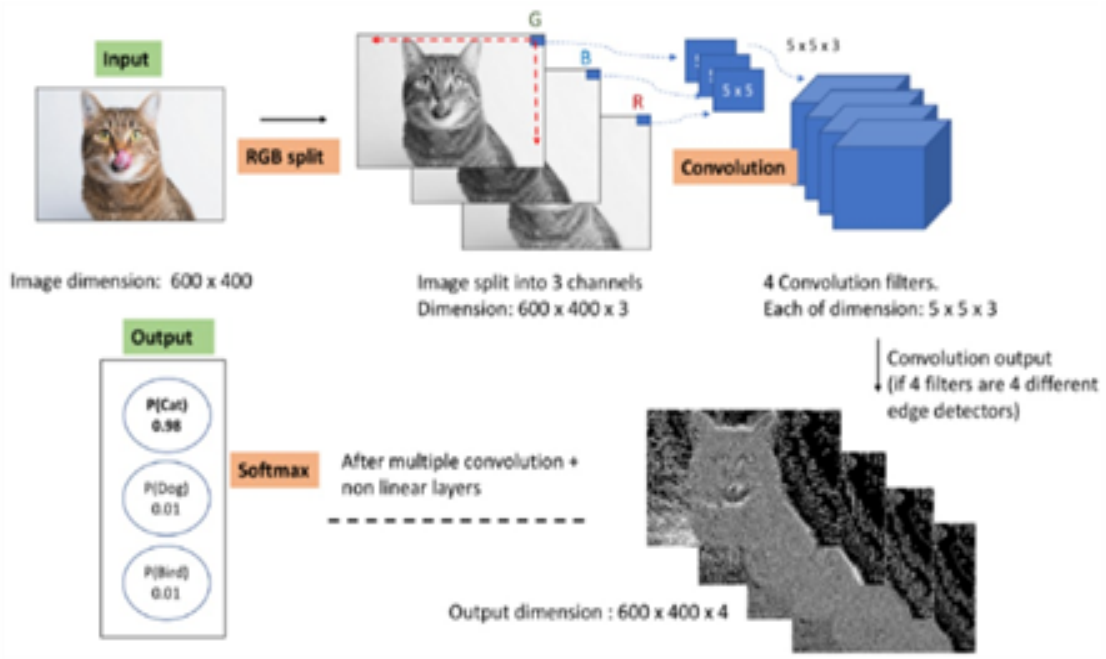
Prediction: boy in red shirt is jumping off of playground equipment

Approach

Convolutional Neural Networks

The machine must be able to learn patterns like vertical, horizontal edges, round shapes, and other patterns of an image. Convolution, Pooling, and RELU are the three operations that are performed multiple times on an input matrix of pixels representing an image.

Convolution is done by convolving with n filters which are then treated with pooling layers like max-pool / average-pooling and then with nonlinear activation functions like RELU. Finally, the fully connected layer is passed to a softmax layer which predicts the probability of each object in the given image.

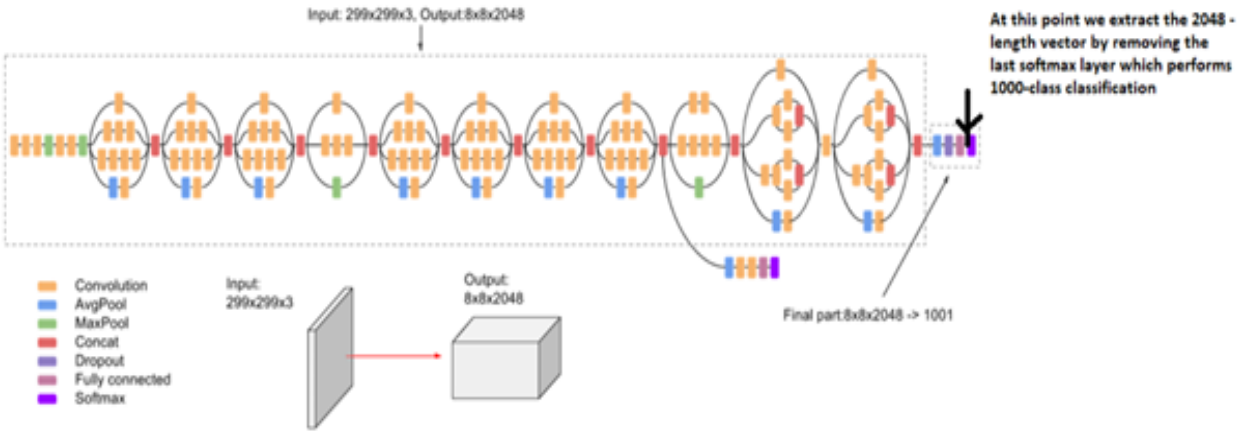


Some of the classic networks to extract features and implement the above are AlexNet, VGG, LeNet-5, Inception, ResNet.

Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

Source

In this project, we have used two networks - InceptionV3 and ResNet152 which are CNN models trained on the imagenet dataset. The pre-trained weights used for these networks were already trained on 1000 different classes.



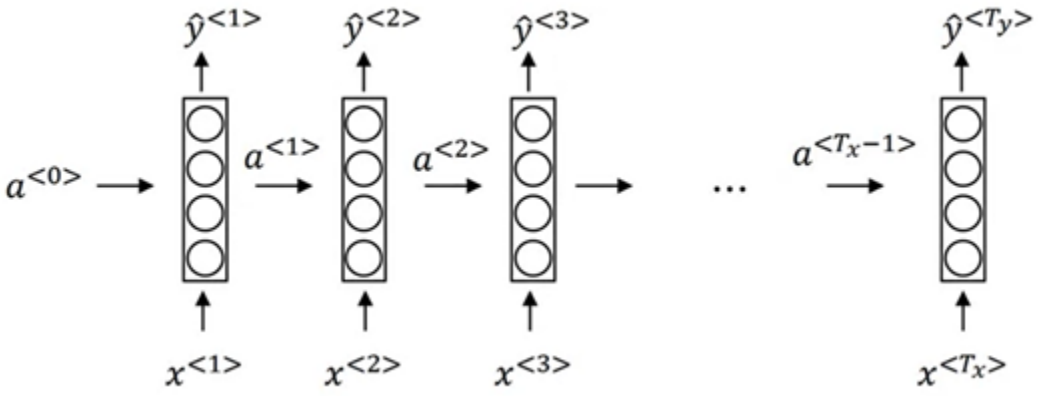
Normally, CNN's last layer is the softmax layer, which assigns the probability that each object might be in the image. It works on the principles of conditional probability. But if we remove that softmax layer from CNN(because we want the feature vector of an image), we can feed CNN's rich encoding of the image into the decoder (language generation RNN) designed to produce phrases.

Inceptionv3 - 2048 length feature vector

Resnet 152 V2 - 2048 length feature vector

Recurrent Neural Networks (RNN)

The second step in this project is to generate captions. Earlier in the data preprocessing part, each caption is appended with a ‘start’ token at the beginning of the sentence and an ‘end’ token at the end of the sentence. It is important to know the start and end of each sentence. In image captioning, the input is an image and the output is a sequence of words, so we use sequence models such as Recurrent Neural Networks to solve this problem

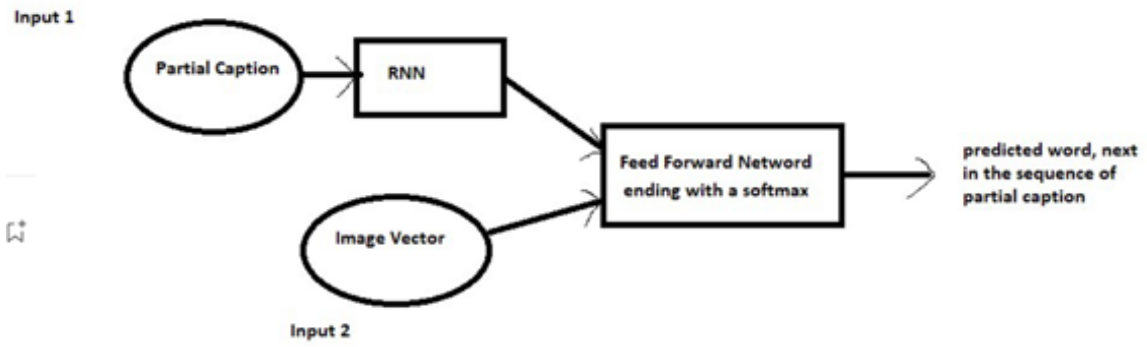


Language Modelling

We construct a vocabulary of words from the training dataset. Each word is assigned an index corresponding to its position in the list.

In Image Captioning we have two inputs, feature vector of the image+ partial caption.

	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq



Each prediction happens this way, first, the ‘start’ token along with the image is given as input to the model, the model predicts the next most probable word given the image and first word. Now, the first two words (start along with the first predicted word) are passed as input to the model, the model now predicts the next most probable word given the image and first two words. This process is repeated till the end token is observed.

First step prediction **max(p(y1|startseq))-**

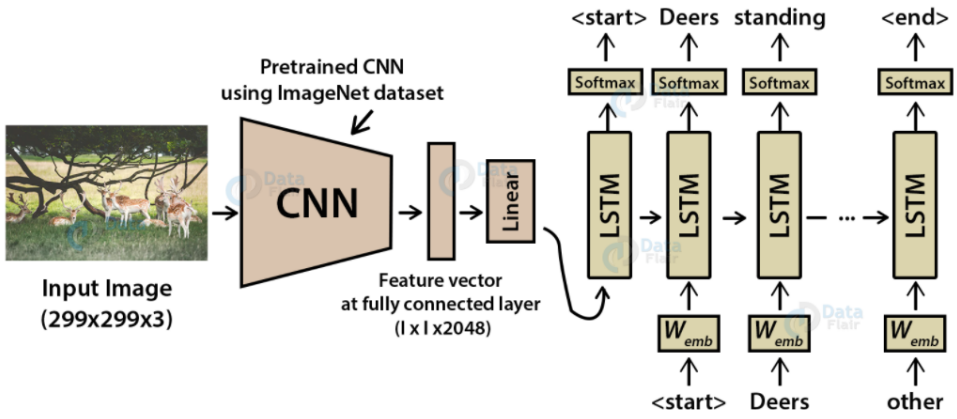
Second step **max(p(y2|startseq the))**

Third step **p(y3| startseq the black)**

……

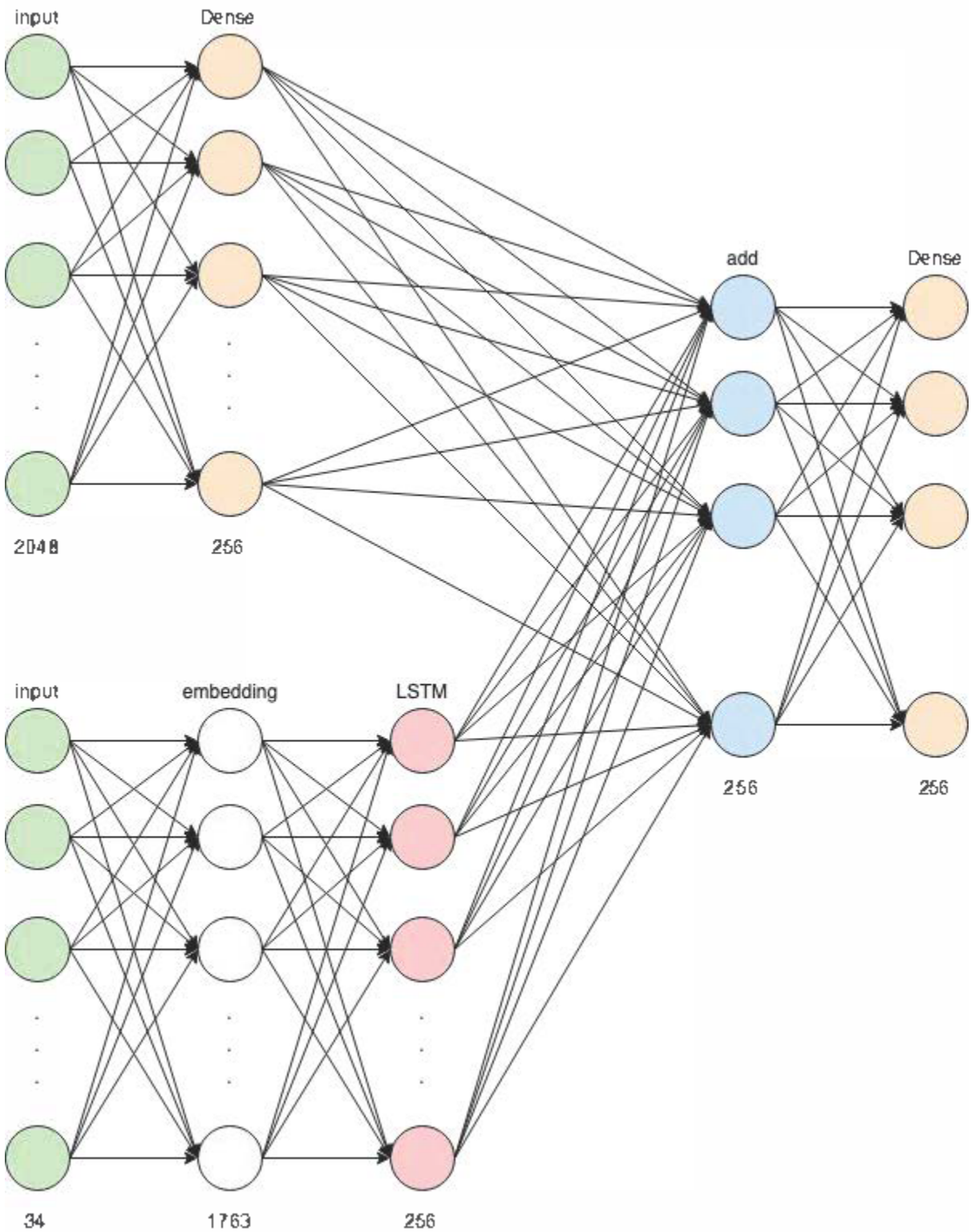
nth step p(yn| startseq the black cat…… grass)

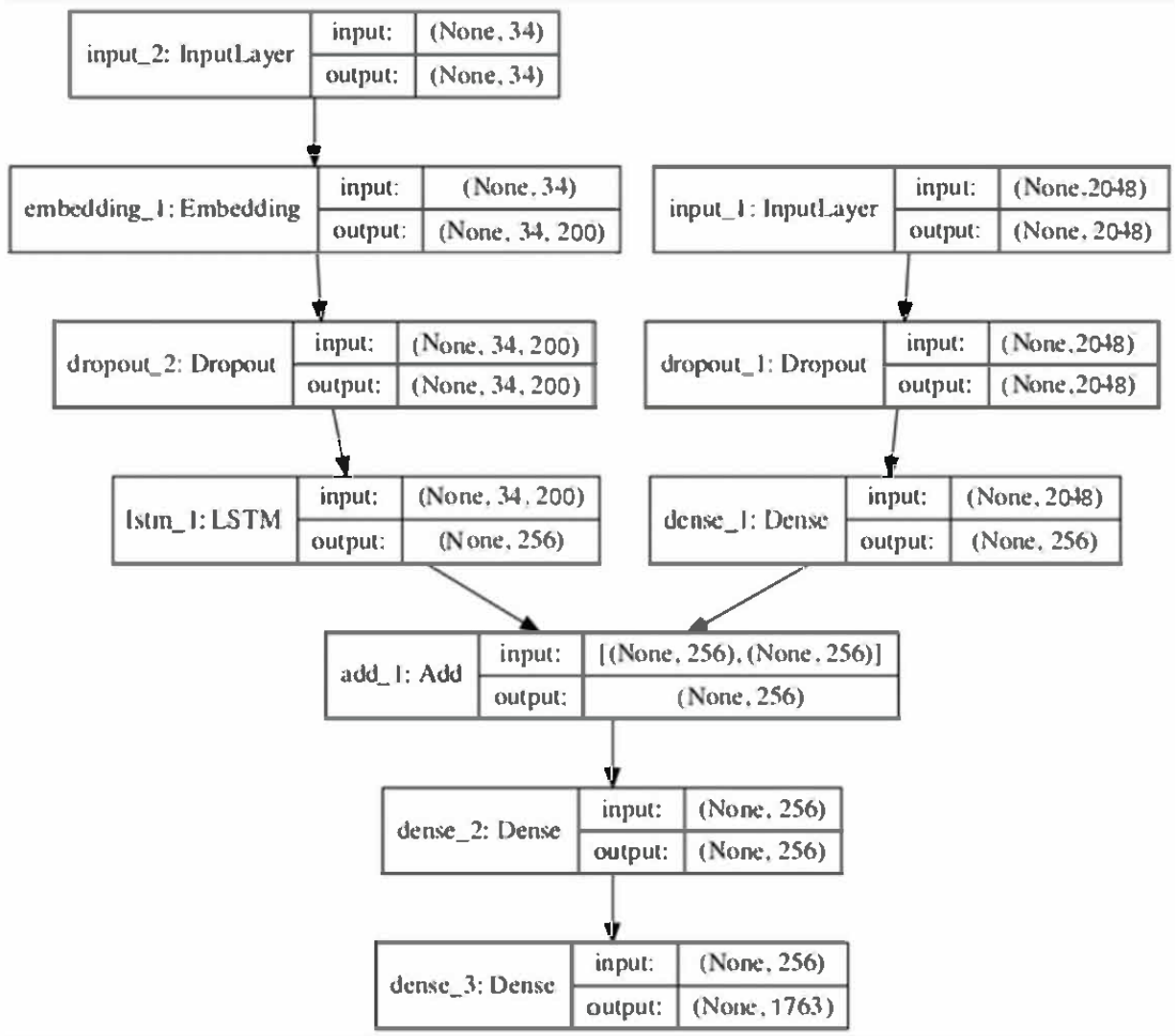
In this project, we are using **LSTM** (**Long short term memory**) which is responsible for generating the image captions. It is a type of RNN which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN(vanishing gradient descent) which had short-term memory. LSTM can carry out relevant information throughout the processing of input



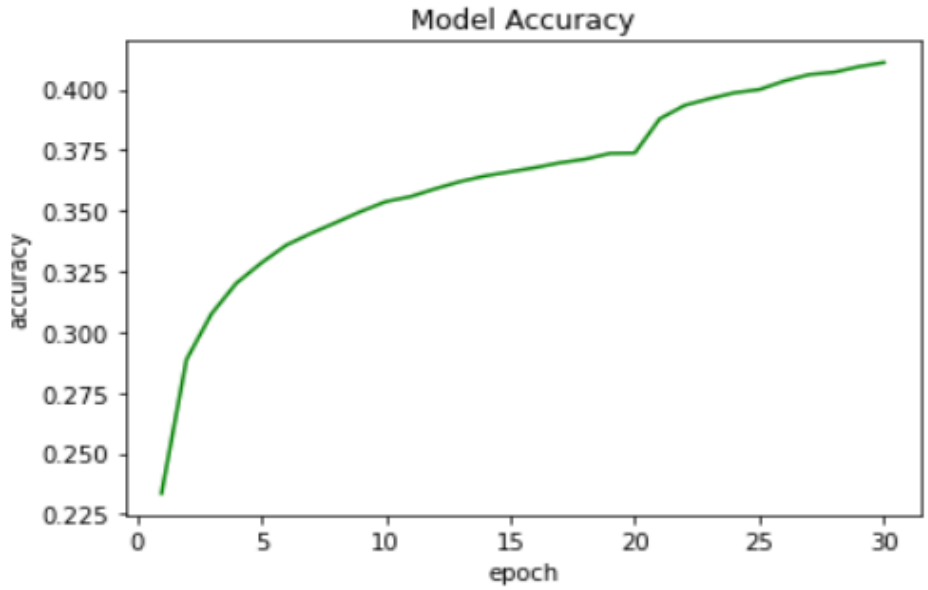
In this project, we are using pre-trained **Glove** embeddings to feed to LSTM cells. The gloVe contains vector representation of many words, in particular, GloVe-6B-200d vector representation is used in this work. The embedding layer is used as part of the language model transfer learning, to get vector representation of words. The model consists of a couple of dropout layers, a couple of dense layers, an embedding layer, and an LSTM layer. **Categorical Cross-Entropy loss is used along with Adam optimizer to train the model.** The training was done for **thirty epochs** with a **dropout of 0.5.**

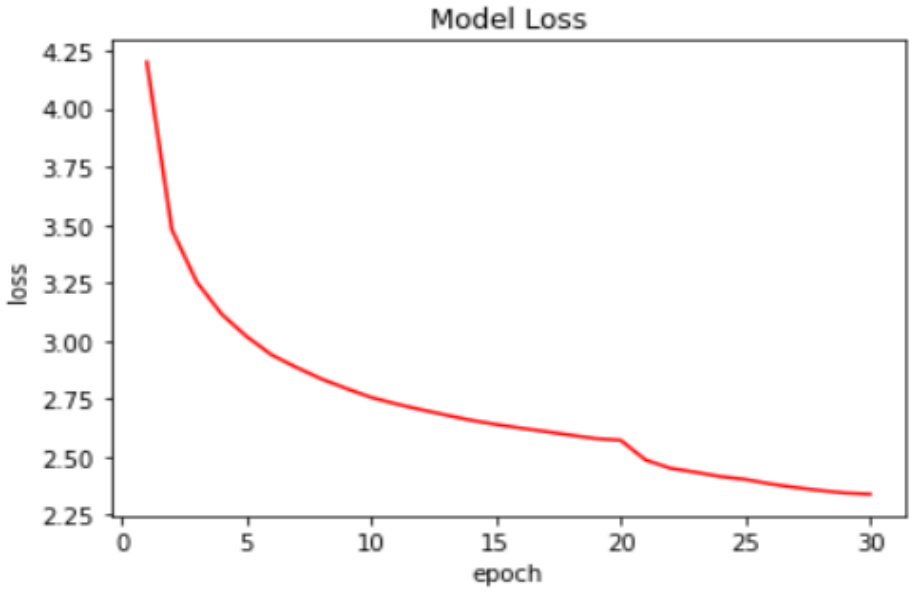
Model Plot


input
Dense
2048
256
input
embedding
LSTM
34
1763
256
add
Dense
256
256


input_2: InputLayer
input: (None, 34)
output: (None, 34)
embedding_1: Embedding
input: (None, 34)
output: (None, 34, 200)
input_1: InputLayer
input: (None,2048)
output: (None, 2048)
dropout_2: Dropout
input: (None, 34, 200)
output: (None, 34, 200)
dropout_1: Dropout
input: (None,2048)
output: (None,2048)
lstm_1: LSTM
input: (None, 34, 200)
output: (None, 256)
dense_1: Dense
input: (None, 2048)
output: (None, 256)
add_1: Add
input: [(None, 256), (None, 256)]
output: (None, 256)
dense_2: Dense
input: (None, 256)
output: (None, 256)
dense_3: Dense
input: (None, 256)
output: (None, 1763)

Results





LOSS ON TRAIN DATASET: 2.1425817012786865
ACCURACY ON TRAIN DATASET: 0.44499143958091736
LOSS ON DEV DATASET: 3.849219799041748
ACCURACY ON DEV DATASET: 0.32388588786125183
LOSS ON TEST DATASET: 3.7788126468658447
ACCURACY ON TEST DATASET: 0.32825222611427307

Learning and Outcomes

Theoretical learning involves understanding the present state of the art model on image captioning. Convolution is a revolutionary idea, instead of having populated neurons in each layer, there is a convolution matrix, which is convolved with the output from the previous layer to generate an output of the current layer. This convolution is element-wise multiplication with padding and stride. Another major theoretical learning was as a part of learning transfer learning models used in the current task like InceptionV3, InceptionResNetV2, ResNetV2. These models have hundreds of layers. Along with these layers, imagenet weights are used as transfer learning to generate feature vectors from images.

Practical learning was from implementing the model and later reiterating it. Major learning was on how to use the Keras API, to implement various functionalities and techniques required to solve the image captioning task.

Dropout is mainly used to avoid overfitting. As the model went on doing better on the training dataset, overfitting became a significant problem. But Keras has a nice way of dealing with overfitting by adding additional dropout layers in between the architecture.

Also, we learned about the generator function. Again Keras has a nice way of dealing with generator functions. If the dataset is large then computing it on any memory device would be difficult. So, it is better to load the data on demand. The generator function helps in achieving this. Thus enabling the loading of much larger datasets.

We also learned how to improve the accuracy of our model and then analyze those improvements using some metrics and their results on different test images.

Summary

A Deep Learning model was implemented to solve the task of image captioning, many techniques were implemented keeping in mind the efficiency of the model.

The first part was a more learning segment. There was a lot of reading. Our mentor assigned us quite a lot of reading jobs. But that reading enlightened us in the field of deep learning.

The second part was about implementing the main model. The main model consisted of implementing the solution for image captioning.

The third part was iterating on the transfer learning part, to ultimately choose InceptionV3 and ResNet152V2 . First, the implementation used was the basic InceptionV3 model for transfer learning.

Improvements

- Due to the time constraint, later improvements could not be made. Also, the idea of using larger datasets like Flickr30k or MS COCO was dropped due to a lack of infrastructure requirements.
- Training on a large dataset (Currently trained on Flickr_8k dataset with only 8k images, can try Flickr_30k dataset or MSCOCO dataset with 120k images), also note training on large dataset may take more than weeks.
- Adding more layers to our LSTM model.
- Training for more epochs and trying different learning rates.