MANAV RACHNA
vidyanatariksha

MANAV RACHNA UNIVERSITY
(FORMELY MANAV RACHNA COLLEGE OF ENGEENERING
NAAC ACCREDITED 'A' GRADE INSTITUTION)
DECLARED AS STATE PRIVATE UNIVERSITY UNDER SECTION 2F OF THE UGC ACT,1956

REPORT

Image Caption Generator using Deep Learning

By-

Abhishta Singh 2K19CSUN04001

Bhavvya Ratra 2K19CSUN04007

Isha Kaushik 2K19CSUN04011

B.Tech.CSE-6DSML

Submitted To-

Ms Priyanka Gupta

Contents

Abstract -
Delineating the contents and objects in the image using natural language is the objective and typical task but has a major impact. For instance, it is a boon for visually impaired people who are unable to comprehend visuals. It can also provide more detailed information to various governmental investigation agencies regarding the crime scene images and footage. The project includes Convolutional Neural Network (CNN) followed by Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU). This paper summarizes with the methods used in order to create much efficient and accurate model, providing with the commonly used dataset i.e. Flickr 8K Dataset containing about 8091 images.

Introduction -
Image Captioning models require three primary components. They don't have any standard names yet, they can be named as Image Feature Encoder, Sequence Decoder and Sentence Generator. One of the customary deep learning architectures for image captioning is called the "inject" architecture that bridges up the Image Feature Encoder to the Sequence Decoder. For our model, we have used LSTM, CNN and architectures of CNN i.e. VGG16-Net

- **CNN**

CNN or Convolutional Neural Network can be defined as the type of Neural Network which allows to extract higher representation for the

image content. It takes the image's raw pixel data, trains the model and then extracts the features automatically for better classification.

- **LSTM**

LSTM or Long-Short Term Memory is a type of Recurrent Neural Network (RNN) in Deep Learning that has been specifically developed for the use of handling sequential prediction problems. For instance, Text/Image/Handwriting Generation, Stock Market Prediction, Text Translation, etc. They contain neurons to perform computation, and they are often referred to as memory cells or simply cells.

- **VGG16**

VGG16 is a CNN Architecture, which was used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. It is still one of the best vision architecture to date. It is an extensively used CNN architecture used for ImageNet, a large visual database project used in visual object recognition software research.

- **VGG19**

VGG19 is a variant of the VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layers, 5 MaxPool layers and 1 SoftMax layer). VGG19 has 19.6 billion FLOPs. Compared with VGG16, VGG19 is slightly better but requests more memory. The VGG16 model is composed of convolutions layers, max pooling layers, and fully connected layers. The total is 16 layers with 5 blocks and each block with a max pooling layer.

- **InceptionV3 -**

InceptionV3 is a convolutional neural network for accommodating in image analysis and object detection, and got its start as a module for GoogleNet. It is the third edition of Google's Inception Convolutional Neural Network, and was introduced during the ImageNet Recognition Challenge. The design of InceptionV3 was aspired to allow deeper networks while also keeping the number of parameters from growing too large: it has "under 25 million parameters", compared against 60 million for AlexNet.

Motivation -

The idea behind this project was to generate captions automatically of the objects seen in an image. Also, in order to solve the vanishing gradient problem faced by standard recurrent neural networks, GRU is also applied on the image dataset. This will help in understanding the dataset and getting more accurate upshots. The eminence of this project for the society is for natural language processing, for social media, for image indexing and many more.

Problem Statement -

The problem statement that made us get moving was to develop a system for users, which can automatically generate the description of an image with the use of CNN along with LSTM and to apply GRU simultaneously for better upshots/results.

Objective -

The main objective of the project is to create a model based on Deep Learning using CNN, LSTM & GRU. Also, to compare the results and the better one will be used to create a Web Application working model.

Literature Review -

- Manish Raypurkar, Abhishek Supe, Pratik Bhumkar, Pravin Borse, Dr. Shabnam Sayyad composed their work on Deep Learning Based Image Caption Generator in 2021 at International Research Journal of Engineering and Technology, proposing Deep Learning based on Convolutional Neural Network (CNN) & LSTM which is a type of RNN and the dataset used in this model is Flickr 8K. Conclusion that came out was that the proposed model produces high quality captions.

- Chaoyang Wang , Ziwei Zhou and Liang Xu composed their work on An Integrative Review of Image Captioning Research in 2020 at ISCME, proposing Deep Learning based on the SVM classifier & CRF used in pattern recognition. The dataset that has been used was MSCOCO. This paper introduces some existing image captioning methods and analyzes their principles. The datasets and evaluation indexes needed in this field are introduced.Although the existing image captioning algorithms have improved the prediction effect to a certain extent, they do not realize the function of generating specific description statements according to specific situations.

- Murk Chohan, Adil Khan, Muhammad Saleem Mahar, Saif Hassan, Abdul Ghafoor, Mehmood Khan composed their work on Image Captioning using Deep Learning in 2020 at

International Journal of Advanced Computer Science and Applications, in which, for a text description of specific tasks like in medical or traffic movement description their own dedicated datasets are created. Datasets used are MSCOCO and Flickr 8k & 30k. Conclusion that came out was that CNN is the best-suited model for image content extraction. RNN & LSTM are used for language generation. LSTM has performed better than RNN.

- Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares composed their work on Image Captioning: Transforming Objects into Words in 2020 at Yahoo Research San Francisco, in which, for object detection and feature extraction, RCNN is used as base CNN. Region Proposed Network (RPN) generates bounding boxes for object proposals. Datasets used are MS-COCO 2014 captions dataset containing 113k training images. Conclusion that came out was that The Object Relation Transformer has been adapted for the image caption generator task that encodes detected images. At present, the model only takes into account geometric information in the encoder phase.

- B.Krishnakumar,K.Kousalya, S.Gokul,R.Karthikeyan, D.Kaviyarasu composed their work on Image Caption Generator Using Deep Learning in 2020 at IJAST, proposed Deep Learning based Convolution Neural Networks to identify objects in the images using OpenCv. Detected Images converted into audio using GTTP and then converted to text using the Long Short Term Memory network. They used the Pre-trained model VGG16 as a baseline model. Conclusion that came out was that the proposed Model successfully trained to generate captions of images using CNN technique, model is depends on data and

used small data set. The model generates captions by using Keras Framework used in Jupyter notebook.

- Lakshminarasimhan Srinivasan, Dinesh Sreekanthan, Amutha A.L composed their work on Image Captioning - A Deep Learning Approach in 2018 at International Journal of Applied Engineering Research, in which CNN and LSTM are used. Flickr 8K Dataset is used in the proposed model. The Training Dataset has 6000 images and the test dataset has 1000 images. Conclusion that came out was that the authors have implemented a deep learning approach for the captioning of images using Keras and Tensorflow at backend.

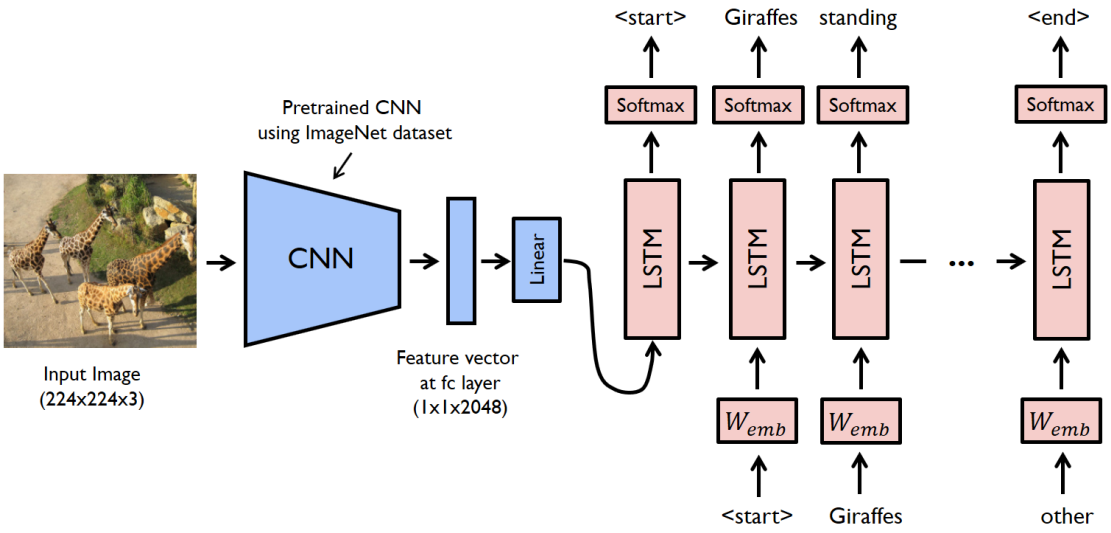
- R. Subhash composed their work on Automatic Image Captioning using CNN & LSTM in 2019 at Journal of Physics Conference, proposed Deep Learning based Convolution Neural Networks and natural language processing NLP technique. Reasonable sentences are framed. Dataset used was MSCOCO with Pycharm IDE. Conclusion that came out was that the proposed Model having CNN whose output is paired with LSTM helps to generate descriptive captions for the image.

- Pranay Mathur, Aman Gill, Ayush Yadav, Anurag Mishra, Nand Kumar Bansode composed their work on Camera2Caption: A real time image caption generator in 2017 at International Conference on Computational Intelligence in Data Science, proposed Deep Learning & Deep Reinforcement Learning laid by computer vision. Dataset used was MSCOCO. Conclusion that

came out was that the proposed Model based on Deep Learning & generate high quality caption by using Tensorflow by Google.

Methodology -

In order to train the LSTM Model, we have predefined our target and label text. For an instance, suppose the caption is "Black dog and spotted dog are fighting", following will be the label and target text:Label - [<start>, Black, dog, and, spotted, dog, are, fighting .]Target - [Black dog and spotted dog are fighting ., <end>]This is done in order to make the model recognize the start and end of our labeled sequence.



Source: Google

Experimental Setup -

- **Libraries used:**

  For VGG16-Net:

  Keras (to import VGG16, load_img, img_to_array, preprocess_input, Model), OS, IPython to display Images, String, Pickle to import dump, Numpy, Keras Layers for LSTM, Keras Tensorflow, Keras Callbacks, Matplotlib.

  For VGG19-Net:

  Keras (to import VGG16, load_img, img_to_array, preprocess_input, Model), OS, IPython to display Images, String, Pickle to import dump, Numpy, Keras Layers for LSTM, Keras Tensorflow, Keras Callbacks, Matplotlib.

- **Platform-**

  The platform that we used for VGG16 and VGG19 models are Google Colab Notebook and Jupyter Notebook respectively. Among both the notebooks. It’s obvious that the Jupyter Notebook works faster than Google Colab Notebook, since the Laptop's GPU is directly involved in the Jupyter Notebook.

- **Dataset-**

Table I: Flickr8k Dataset

Flickr8k Image Dataset (Flickr8k_Dataset)	
Total Images	8092
Training Data (images)	6000
Validation Data (images)	1000
Testing Data (images)	1000
Flickr8k Text Dataset (Flickr8k_text)	
Flickr8k token (text)	40460
Flickr8k lemmatized token (text)	40460

The dataset that we used for our project is Flickr8K Dataset in which total images areb 8092. For training of the data, we have used 6000 images, for validation-1000 images and for testing of the image data, we used 1000 images.

In this dataset, Flickr8K token file is used containing the text vocabulary of size 40460 and the Flickr8K lemmatized token contains 40460 of the vocabulary size of the text.

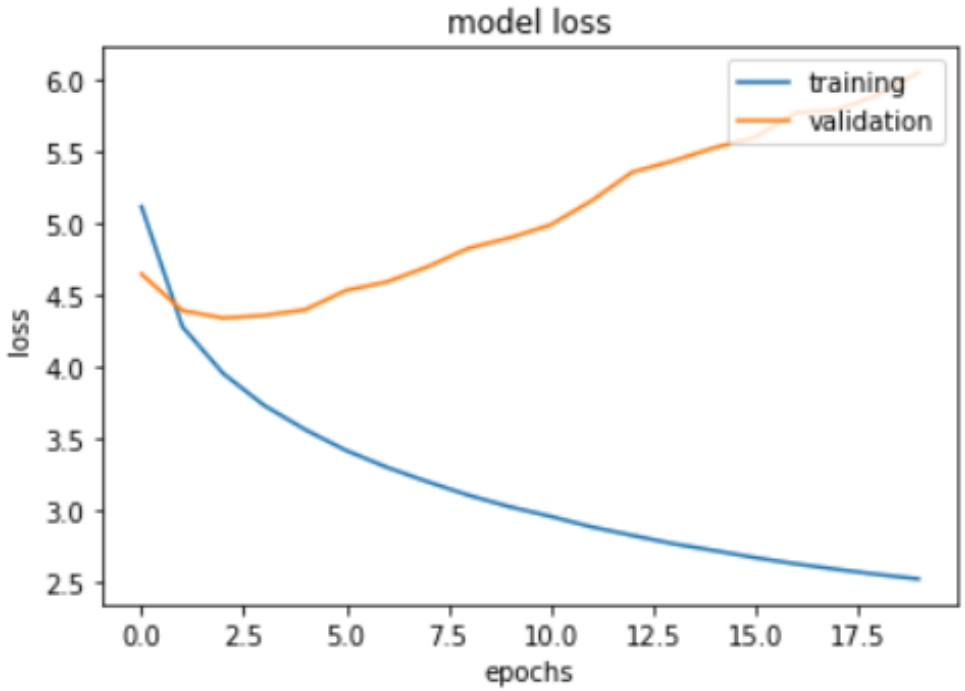
Results & Conclusion -

i	Xi		Yi
1	Image Feature Vector	startseq	A
2	Image Feature Vector	startseq A	girl
3	Image Feature Vector	startseq A girl	going
4	Image Feature Vector	startseq A girl going	into
5	Image Feature Vector	startseq A girl going into	a
6	Image Feature Vector	startseq A girl going into a	wooden
7	Image Feature Vector	startseq A girl going into a wooden	building
8	Image Feature Vector	startseq A girl going into a wooden building	endseq

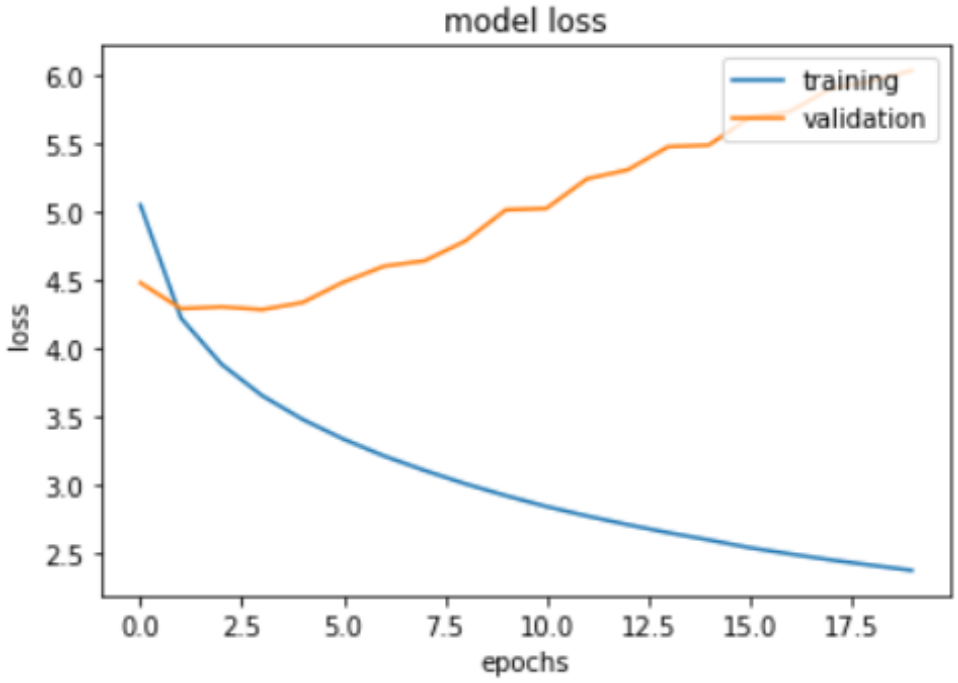
In our whole model, based on image pre-processing, training and testing, we found that the captions that are already present and the model that predicted the caption have a bit more similarity but many errors too. In order to be more accurate, more images are needed to be pre-processed, trained and tested. Since we used the Flickr 8K dataset containing 8091 images to be precise, it is clearly seen that for more accuracy of the model, more thousands of images are needed. For future research, the Flickr 31K dataset containing approximately 31,000 images can be used.

```
Model: "model_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 fc1 (Dense)                 (None, 4096)              102764544

 fc2 (Dense)                 (None, 4096)              16781312

=================================================================
Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0
_________________________________________________________________
None
```

VGG-16 Model Summary



VGG-16 Graphical Representation



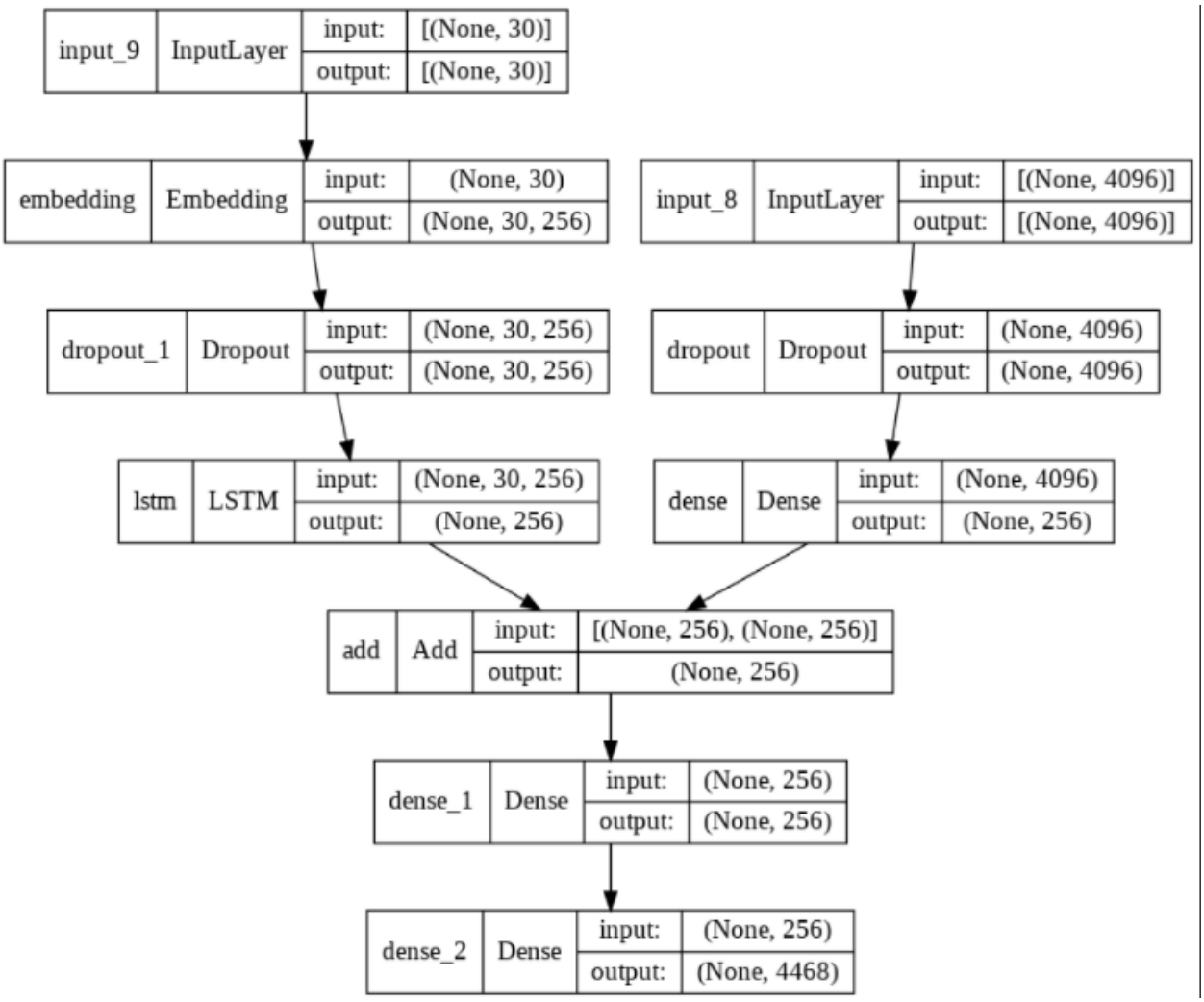
VGG-19 Graphical Representation

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312

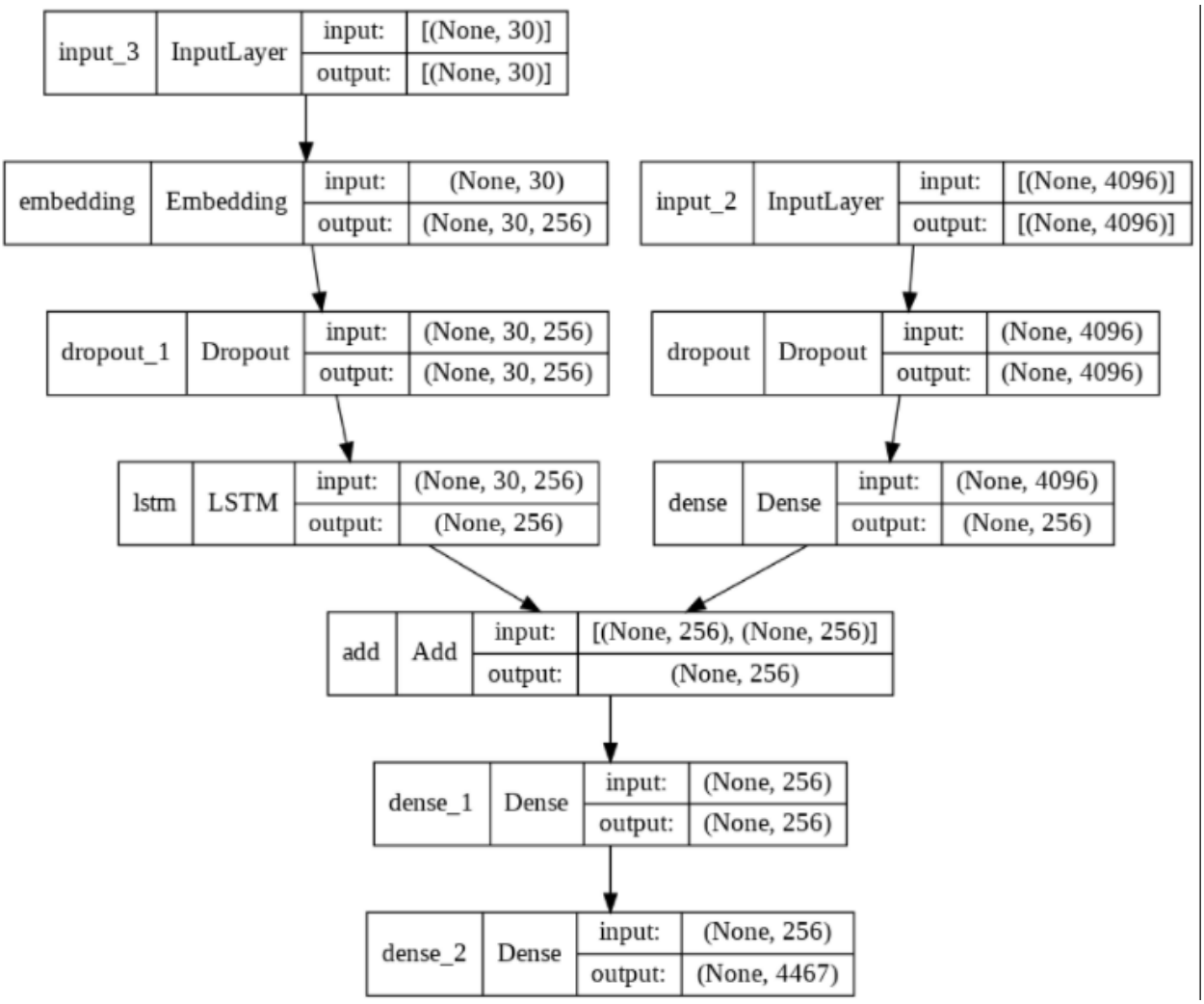
Total params: 139,570,240
Trainable params: 139,570,240
Non-trainable params: 0

VGG-19 Model Summary

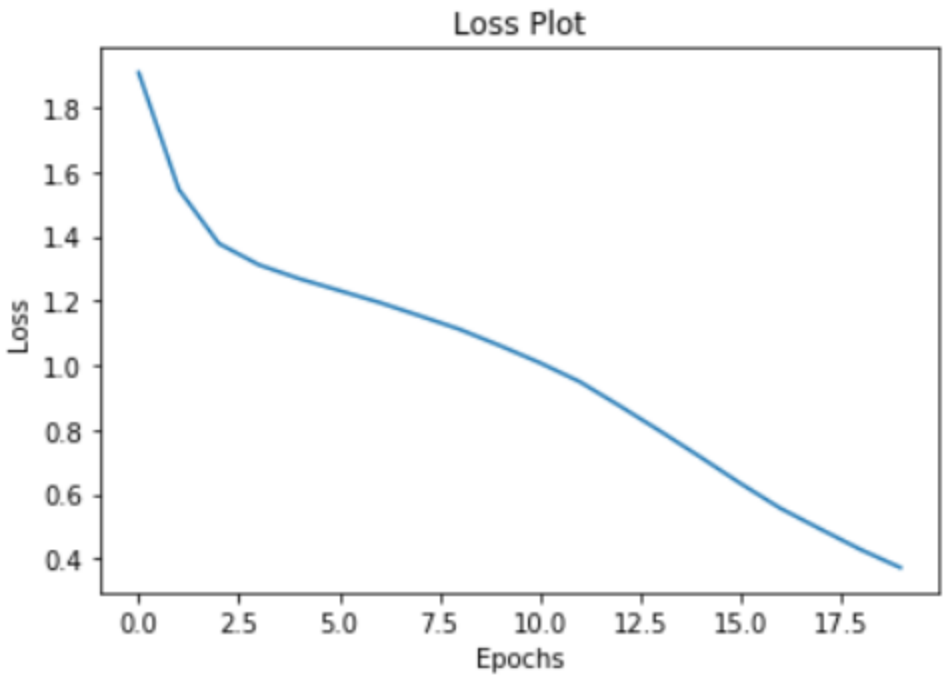
CNNs	Layers	#param	Done By
VGG-16 + LSTM	15	134	Abhishta
VGG-19 + LSTM	18	139	Bhavvya



VGG-16 Model Flowchart



VGG-16 Model Flowchart



InceptionV3 Graphical Representation

```
Real Caption: 1925434818 2949a8f6d8 jpg 1       some children playing in a pit full of colorful balls
Prediction Caption: 7e8df9a2ea jpg 1    a little girl holds a <unk> truck bench holding a little girl in a blue water sit along
the woods in a blue leaves in a brown dog plays on an object in an older brother on the
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:211: RuntimeWarning: Glyph 9 missing from current
font.
  font.set_text(s, 0.0, flags=flags)
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:15: UserWarning: Tight layout not applied. tight_layout cannot
make axes width small enough to accommodate all axes decorations
  from ipykernel import kernelapp as app
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:180: RuntimeWarning: Glyph 9 missing from current
font.
  font.set_text(s, 0, flags=flags)
```

InceptionV3 Result Accuracy

Future Work-

For future work, in order to improve the model, we can use GTTI-API for the visually impared section of the society. This will help them describe the contents of the image present in it.

Since, Deep-belief networks are used to recognize, cluster and generate images, video sequences and motion-capture data, therefore we can use it to make the model more esteemed.

Reference -

(1)https://towardsdatascience.com/

(2)https://kaggle.com/adityajn105/flickr8k

(3)https://irjet.net/archives/V8/i3/IRJET-V8I392.pdf

(4)https://iopscience.iop.org/article/10.1088/1742-6596/1748/4/042060/pdf

(5)https://thesai.org/Downloads/Volume11No5/Paper_37Image_Captioning_using_Deep_Learning.pdf

(6)https://proceedings.neurips.cc/paper/2019/file/680390c55bbd9ce416d1d69a9ab4760d-Paper.pdf

(7)http://sersc.org/journals/index.php/IJAST/article/view/5927/3650

(8)https://iopscience.iop.org/article/10.1088/1742-6596/1748/4/042060/pdf

(9)https://www.ripublication.com/ijaer18/ijaerv13n9_102.pdf

(10)https://www.researchgate.net/publication/337311437_Automatic_Image_Captioning_Using_Convolution_Neural_Networks_and_LSTM

(11)[*https://www.irjet.net/archives/V6/i3/IRJET-V6I31335.pdf*](https://www.irjet.net/archives/V6/i3/IRJET-V6I31335.pdf)

(12)[*https://www.kaggle.com/shweta2407/vgg16-and-lstm-image-caption-generator*](https://www.kaggle.com/shweta2407/vgg16-and-lstm-image-caption-generator)

(13)[*https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615*](https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615)

(14)[*https://arxiv.org/ftp/arxiv/papers/2009/2009.02565.pdf*](https://arxiv.org/ftp/arxiv/papers/2009/2009.02565.pdf)