A MAJOR PROJECT

on

IMAGE CAPTION GENERATOR USING DEEP LEARNING

Submitted in partial fulfilment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

MALINENI APARNA

O180825

Under the guidance of

Mrs. S. NAGAMANI, Assistant Professor

Department of Computer Science and Engineering

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

ONGOLE CAMPUS

2023-2024

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is certify that the major project entitled on **'IMAGE CAPTION GENERATOR'**, being submitted by **MALINENI APARNA (O180825)** in partial fulfilment of the requirement for the award of the degree of the Bachelor Of Technology in computer science and Engineering in Dr. APJ Abdul Kalam ,RGUKT-AP IIIT Ongole is a record of bonafide work carried out by them under my guidance and supervision during the academic year 2023-24.

The results presented in this project have been verified and found to be satisfactory. The report hasn't been submitted previously in part or in full to this or any other university or institution for the award of any degree.

Mrs. S. Nagamani
Assistant Professor,
Department of CSE,
RGUKT Ongole.

Mr. B Sampath Babu(P.h.D)
Head of Department,
Department of CSE,
RGUKT Ongole

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is certify that the project entitled on **'IMAGE CAPTION GENERATOR'**, being submitted by **MALINENI APARNA (O180825)** in partial fulfilment of the requirements for the award of the degree of the Bachelor Of Technology in computer science and Engineering in Dr. APJ Abdul Kalam ,RGUKT-AP IIIT Ongole is a record of bonafide work carried out by them under my guidance and supervision during the academic year 2023-24.

Mrs. S Nagamani
Assistant Professor,
Department of CSE,
RGUKT Ongole.

Mr. B Sampath Babu(P.h.D)
Head of Department,
Department of CSE,
RGUKT Ongole

APPROVAL SHEET

This report entitled "**IMAGE CAPTION GENERATOR**" submitted by MALINENI APARNA (O180825) is **Mrs.S NAGAMANI**, Assistant Professor Department of CSE, RGUKT Ongole approved for the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING**.

Examiner

_____

_____

_____

_____

Supervisor

_____

_____

_____

_____

Chairman

_____

Date: _____

Place:

ACKNOWLEDGEMENT

It is our privilege and pleasure to express a profound sense of respect, gratitude and Indebtedness to our guide **Mrs S NAGAMANI, Assistant Professor**, Dept. of Computer Science and Engineering, Rajiv Gandhi University of Knowledge Technologies, for her indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work. We express our sincere gratitude to **B Sampath Babu, Assistant Professor & Head of Department of Computer Science and Engineering**, Rajiv Gandhi University of Knowledge Technologies, for his suggestions, motivations and co-operation for the, successful completion of the work. We extend our sincere thanks to **Prof. B Jayarami Reddy**, **Director**, Rajiv Gandhi University of Knowledge Technologies for his encouragement. And also we thank each individual of the RGUKT, Ongole campus for their impeccable support for our project.

With Sincere Regards,

M. Aparna - O180825

Date: _____

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in out submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature**:** _____ M. Aparna– O180825

Date:

ABSTRACT

In this project, we systematically analyse a deep neural networks based image caption generation method. With an image as the input, the method can output an English sentence describing the content in the image. We analyse three components of the method convolutional neural network (CNN), recurrent neural network (RNN) and sentence generation. The context of a photograph is automatically described in English. When a picture is captioned, the computer learns to interpret the visual information of the image using one or more phrases. The ability to analyse the state, properties, and relationship between these objects is required for the meaningful description generation process of high-level picture semantics. Using CNN -LSTM architectural models on the captioning of a graphical image, we hope to detect things and inform people via text messages in this research. To correctly identify the items, the input image is first reduced to grayscale and then processed by a Convolution Neural Network (CNN).

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
DL	Deep Learning
FC	Fully Connected
LRCN	Long-term Recurrent Convolutional Network

CHAPTER 1
INTRODUCTION

1.1 MOTIVATION:

Training computers to be able to automatically generate descriptive captions for images is currently a very hot topic in Computer Vision and Machine Learning. This task is a combination of image scene understanding, feature extraction, and translation of visual representations into natural languages. This project shows some great promises such as building assistive technologies for visually impaired people and help automating caption tasks on the internet

1.2 PROBLEM DEFINITION:

Our approach is based on two basic models: CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory). CNN is utilized as an encoder in the derived application to extract features from the snapshot or image, and LSTM is used as a decoder to organize the words and generate captions. Image captioning can help with a variety of things, such as assisting the visionless with text-to-speech through real-time input about the scenario over a camera feed, and increasing social medical leisure by restructuring captions for photos in social feeds as well as spoken messages.

1.3 OBJECTIVE OF THE PROJECT:

The project aims to work on one of the ways to context a photograph in simple English sentences using Deep Learning (DL). The objective is to train a CNN-RNN (Convolutional Neural Network — Recurrent Neural Network) model for automatically generating image captions.

CHAPTER 2
LITERATURE SURVEY

Literature survey is the most important step in the software development process. Before developing the tool, it is necessary to determine the time factor, economy, and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool, they programmers need a lot of external support. This support can be obtained from senior programmers, books, or websites. Before building the system, the above considerations are taken into account for developing the proposed system.

The major part of the project development sector considers and fully surveys all the required needs for developing the project. For every project, a Literature survey is the most important sector in the software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, manpower, economy, and company strength. To improve and tailor the user experience on its products, photos use image classification. Intraclass variation, occlusion, deformation, size variation, perspective variation, and lighting are all frequent issues in computer vision that are represented by the picture classification problem. Methods that work well for picture classification are likely to work well for otherimportant computer vision tasks like detection, localization, and segmentation as well.

Image captioning is a great illustration of this. Given an image, the image captioning challenge is to generate a sentence description of the image. The picture captioning problem is comparable to the image classification problemin that it expects more detail and has a bigger universe of possibilities. Image classification is used as a black box system in modern picture captioning systems, therefore greater image classification leads to better captioned. The image captioning problem is intriguing in and of itself because it brings together two significant AI fields: computer vision and natural language processing. An image captioning system demonstrates that it understands both image semantics and natural language.

Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what all the necessary software is needed to proceed with the next step such as developing the tools, and the associated operations. To construct an image sentence, image classification is a key stage in the object recognition and picture analysis process. The final output of the image categorization phase might be a statement. To date, a variety of image captioning techniques have been presented. Several studies have been carried out in attempt to determine the best image captioning technique. It's difficult to pick one approach as the finest of them allbecause the results and accuracy are dependent on a variety of circumstances.

In order to achieve the most accurate results, traditional approaches have been constantly modified as well as new image captioning techniques invented during the previous few decades. Each caption generator technique has its own set of benefits and drawbacks. The focus of the research today is on combining the desired qualities of various techniques in order to boost efficiency. Many high-level tasks, such as image classification, object detection, and, more recently, semantic segmentation, have recently been proven to obtain outstanding results using convolutional neural networks with many layers. A two-stage technique is frequently used, especially for semantic segmentation. Convolutional networks are trained in this way to offer good local pixel-wise data for the second stage, which is often a more global graphical analysis model.

We will use Long short-term memory (LSTM), which is a subset of RNNs, to tackle the problem of Vanishing Gradient. The main goal of LSTM is to solve the problem of Vanishing Gradients. The unique feature of LSTM is that it can keep data values for long periods, allowing it to address the vanishing gradientproblem. When compared to applying RNN, the results revealed that using a mixture of LSTM generated better outcomes. CNNs employ multilayer convolution to accomplish feature engineering and integrate these features internally, unlike traditional image recognition algorithms. It also employs the pooling and fully connected (FC) layers, as well as SoftMax.

CHAPTER 3

METHODOLOGIES AND ANALYSIS

3.1 **EXISTING SYSTEM**:

Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe them in a natural language like English. In this Python based project, we will have implemented the caption generator using CNN (Convolutional Neural Networks) and LSTM (Long Short term memory). The image features will be extracted from Xception which is a CNN model trained on the COCO dataset and then we feed the features into the LSTM model which will be responsible for generating the image captions. Convolutional neural networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easy. It can handle the images that have been translated, rotated, scaled and changes in perspective

3.2 **PROPOSED SYSTEM:**

We have written data pre-processing scripts to process raw input data (both images and captions) into proper format; A pre-trained Convolutional Neural Network architecture as an encoder to extract and encode image features in to a higher dimensional vector space; An LSTM-based Recurrent Neural Network as a decoder to convert encoded features to natural language descriptions; Attention mechanism which allows the decoder to see features from a specifically highlighted region of the input image to improve the overall performance. Development of automatically describing an image with more than natural language sentences which leads to faster information transfer.

3.3 **REQUIREMENTS SPECIFICATION :**

Programming Languages : PYTHON
Modules : Numpy, Pandas, CNN, RNN, Keras, TensorFlow,Pytorch
Operating system : WINDOWS 10 PRO any OS that runs python
IDE editor : VISUAL STUDIO CODE
Tools : Anaconda, Jupyter Notebook, Google Colab
RAM required: 4GB and above

3.4 WORKING EXPLANATION:

1. A user uploads an image that they want to generate a caption for.
2. A gray-scale image is processed through CNN to identify the objects.
3. CNN scans images left-right, and top-bottom, and extracts important image features.
4. By applying various layers like Convolutional, Pooling, Fully Connected, and thus using activation function, we successfully extracted features of every image.
5. It is then converted to LSTM.
6. Using the LSTM layer, we try to predict what the next word could be.
7. The application proceeds to generates a sentence describing the image.

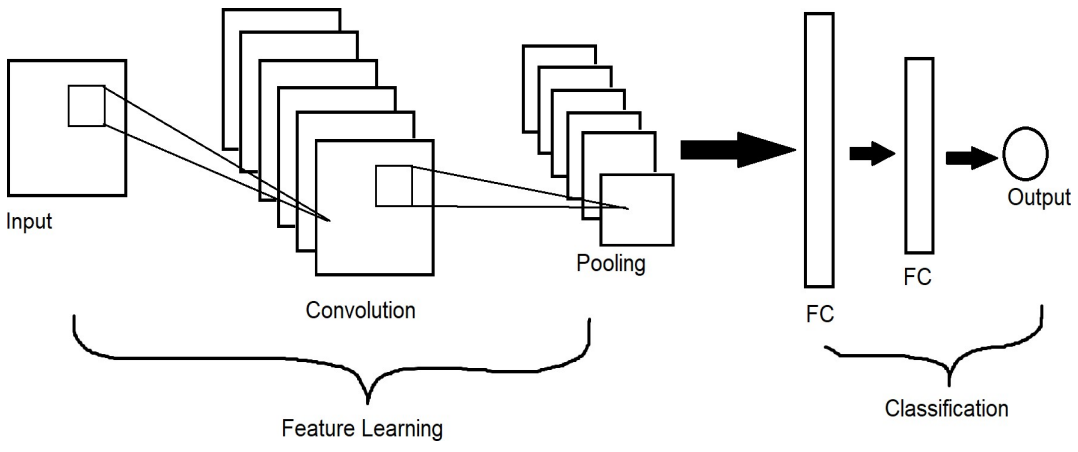
3.5 INTRODUCTION TO TECHNOLOGIES USED:

Algorithms:

- Convolutional Neural Network
- Long Short-Term Memory

3.5.1.Overview on CNN:

CNN is a subfield of Deep learning and specialized deep neural networks used to recognize and classify images. It processes the data represented as 2D matrix-like images. CNN can deal with scaled, translated, and rotated imagery. It analyses the visual imagery by scanning them from left to right and top to bottom and extracting relevant features. Finally, it combines all the parts for image classification.



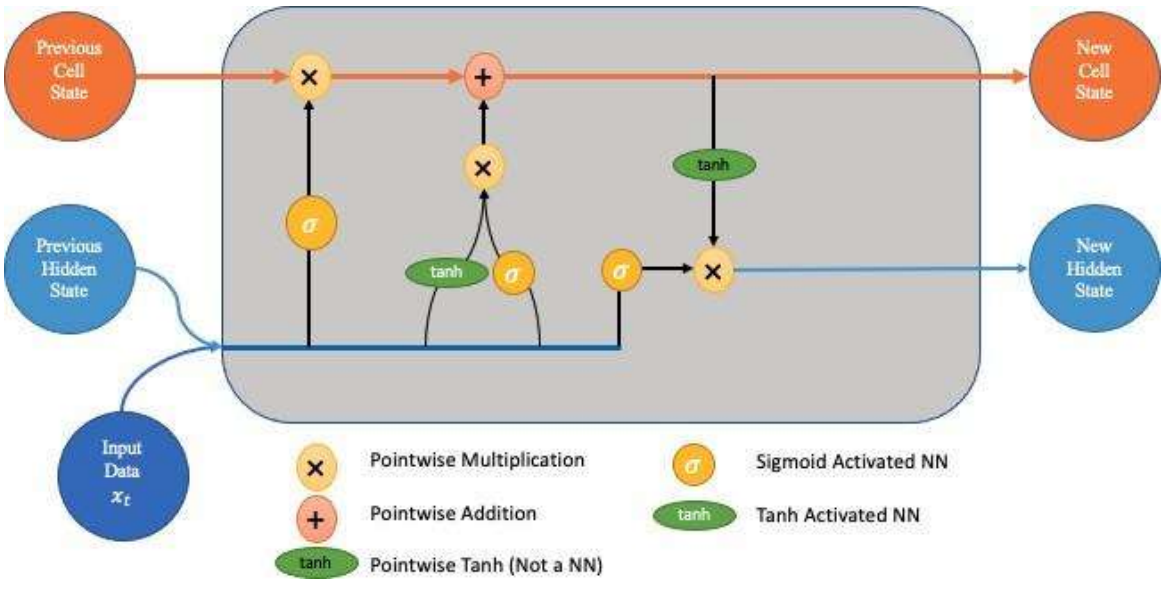
3.5.1 CNN

Some advantages of CNN are:

- It works well for both supervised and unsupervised learning.
- Easy to understand and fast to implement.
- It has the highest accuracy among all algorithms that predicts images. Little dependence on pre-processing, decreasing the need for human effort to develop its functionalities.

3.5.2.Overview of LSTM:

Being a type of RNN (recurrent neural network), LSTM (Long short-term memory) is capable of working with sequence prediction problems. It is mostly used for the next word prediction purposes, as in Google search our system is showing the next word based on the previous text. Throughout the processing of inputs, LSTM is used to carry out the relevant information and to discard non-relevant information.



3.5.2. LSTM

Some advantages of LSTM are:

- Provides us with a large range of parameters such as learning rates, and input and output biases.
- The complexity to update each weight is reduced to O (1) with LSTMs.

3.5.3.CNN - LSTM Architecture Model

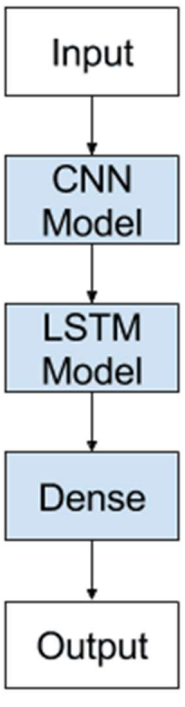
The CNN LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction.

CNN-LSTMs were developed for visual time series prediction problems and the application of generating textual descriptions from sequence of image (e.g., videos) Specifically, the problem of

- Activity Recognition: Generating a textual description of activity demonstrated in a sequence of images.
- Image Description: Generating a textual description of a single image.
- Video Description: Generating a textual description of a sequence of images.

This architecture was originally referred to as a Long-term Recurrent Convolutional Network (LRCN) model, although we will use the more generic name “CNN LSTM”

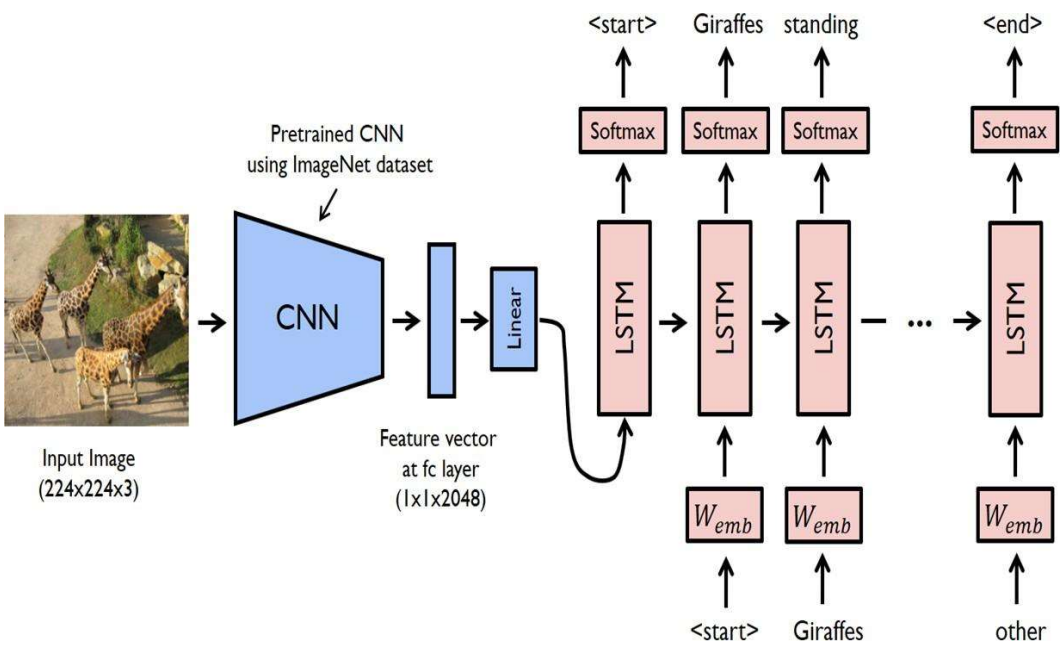
- CNN is used for extracting features from the image. We will use the pre-trained model Xception.
- LSTM will use the information from CNN to help generate a description of the image.



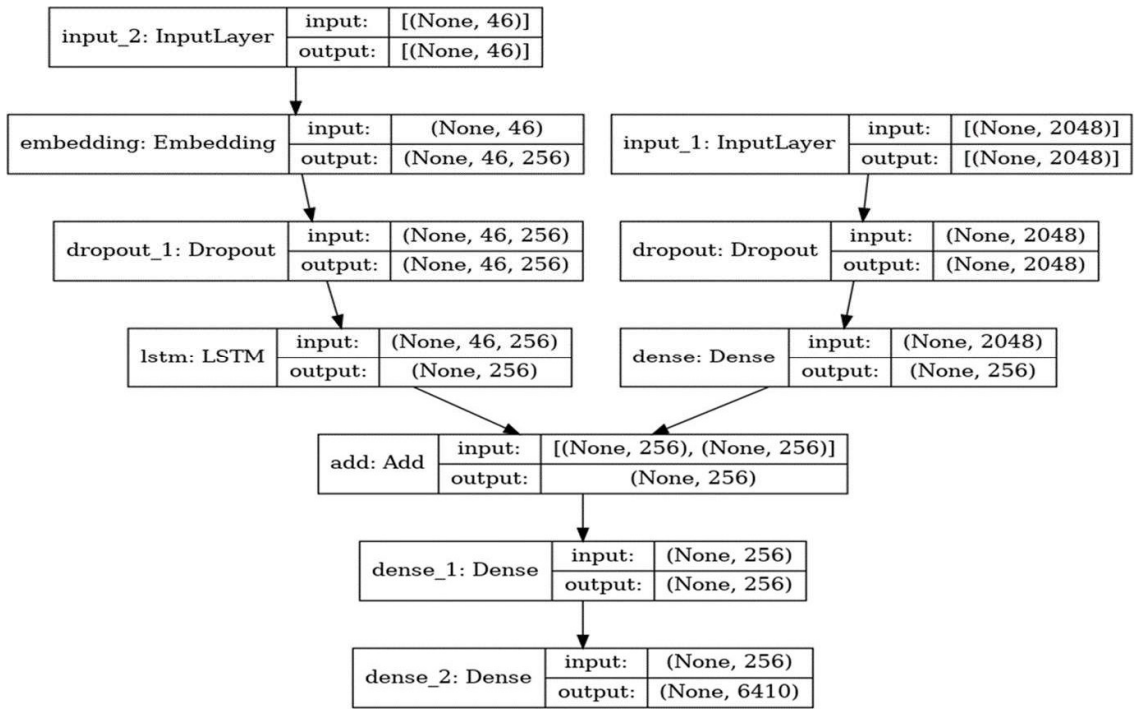
3.5.3. CNN-LSTM MODEL

Methodology

- Import Libraries
- Upload COCO dataset (Data Preprocessing)
- Apply CNN to identify the objects in the image.
- Preprocess and tokenize the captions.
- Use LSTM to predict the next word of the sentence.
- Make a Data Generator
- View Images with caption.



3.5.4. System Architecture



3.5.5. Workflow Diagram

3.6 OVERALL DESCRIPTION

Automatically image captioning is far from mature and there are a lot of ongoing research projects aiming for more accurate image feature extraction and semantically better sentence generation. We successfully completed what we mentioned in the project proposal, but used a smaller dataset due to limited computational power. There can be potential improvements if given more time. First of all, we directly used pre-trained CNN network as part of our pipeline without fine-tuning, so the network does not adapt to this specific training dataset. Thus, by experimenting with different CNN pre-trained networks and enabling fine-tuning, we expect to achieve a slightly higher BLEU4 score. Another potential improvement is by training on a combination of Flickr8k, Flickr30k, and MSCOCO. In general, the more diverse training dataset the network has seen, the output will be. We all agree this project ignite sour interesting application Machine Learning knowledge in Computer Vision and expects to explore more in the future.

CHAPTER-4

DESIGN

4.1 UML DIAGRAMS:

UML is the short form of Unified Modelling Language. UML is a standardized general purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The important goal for UML is to create a common modelling language for the sake of Object Oriented Software engineering. In its current form UML consists of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Activity Diagram:

An activity diagram in UML (Unified Modeling Language) is used to visualize the flow of activities or processes within a system. It is often used to model the workflow, business processes, and complex algorithms..

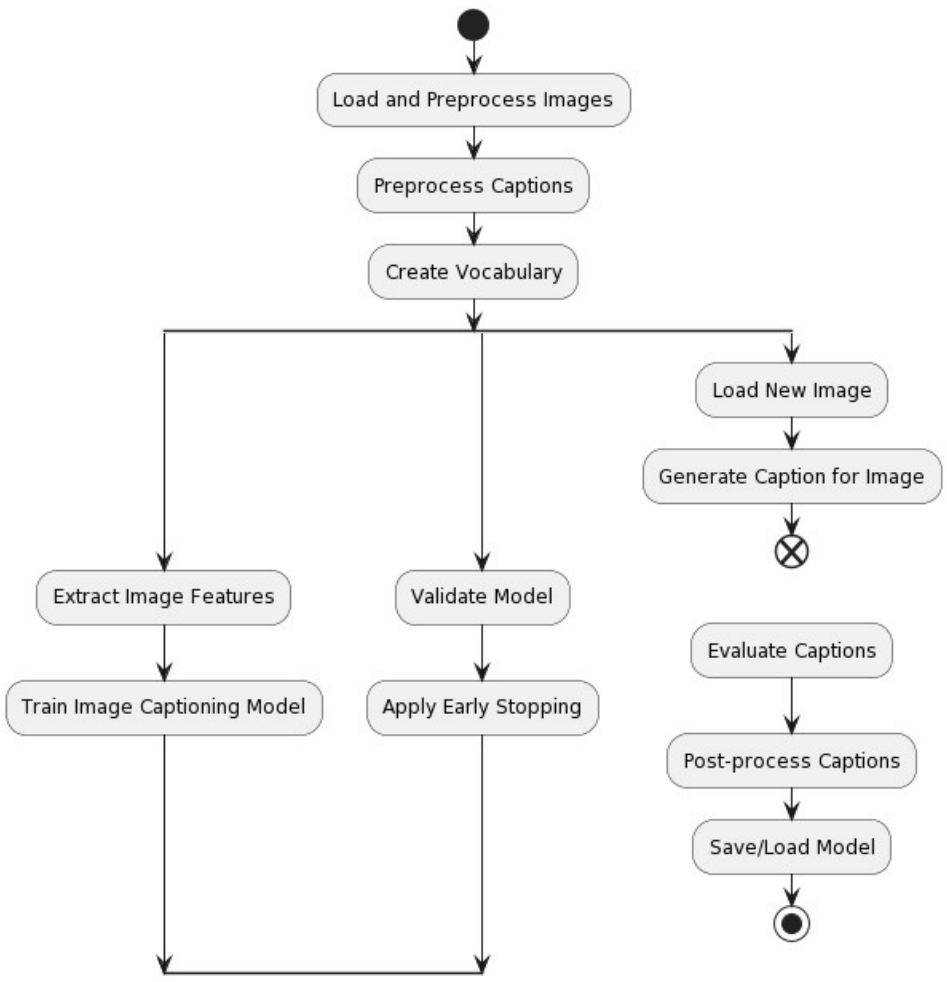


Fig 4.1: Activity Diagram

Class Diagram:

A class diagram is a visual representation of class objects in a model system, categorized by class types. Each class type is represented as a rectangle with three compartments for the class name, attributes, and operations. To understand a class diagram, we must first define what a class is.

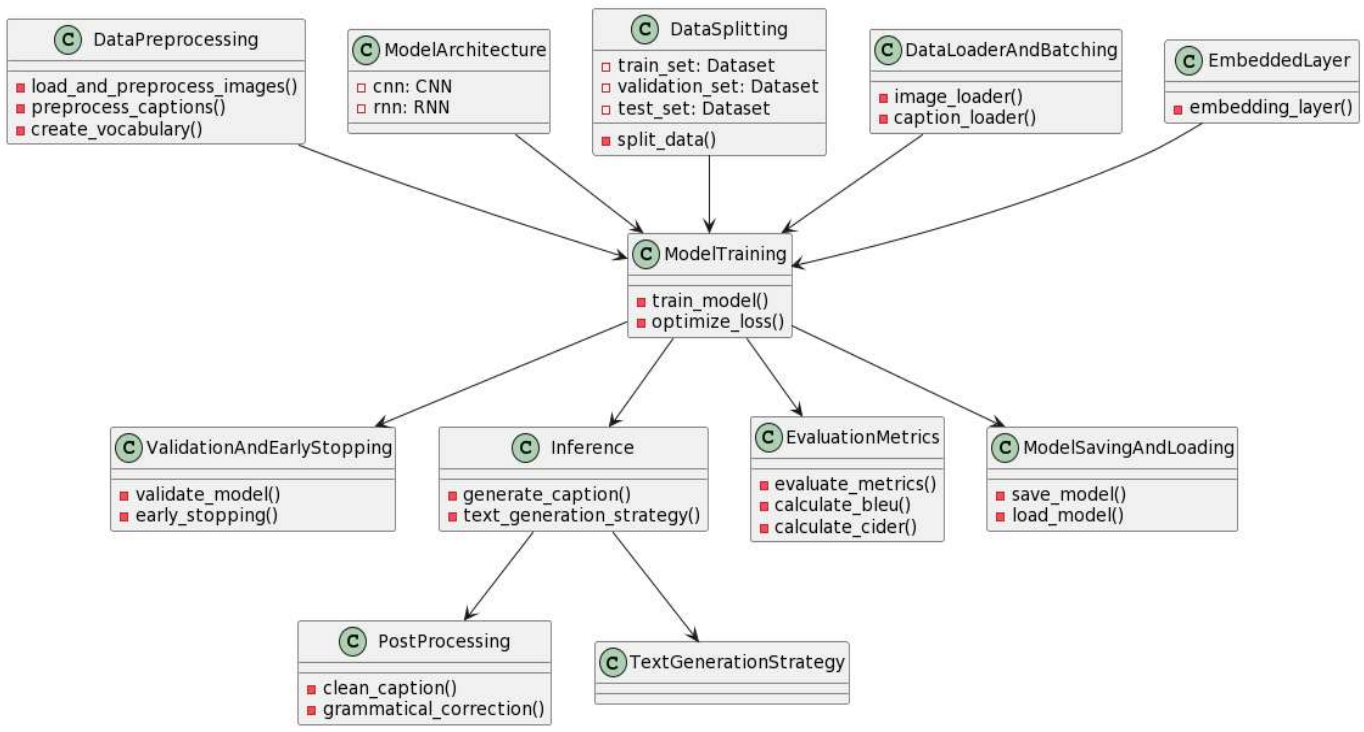


Fig 4.2: Class Diagram

Sequence Diagram:

A sequence diagram is a Unified Modelling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

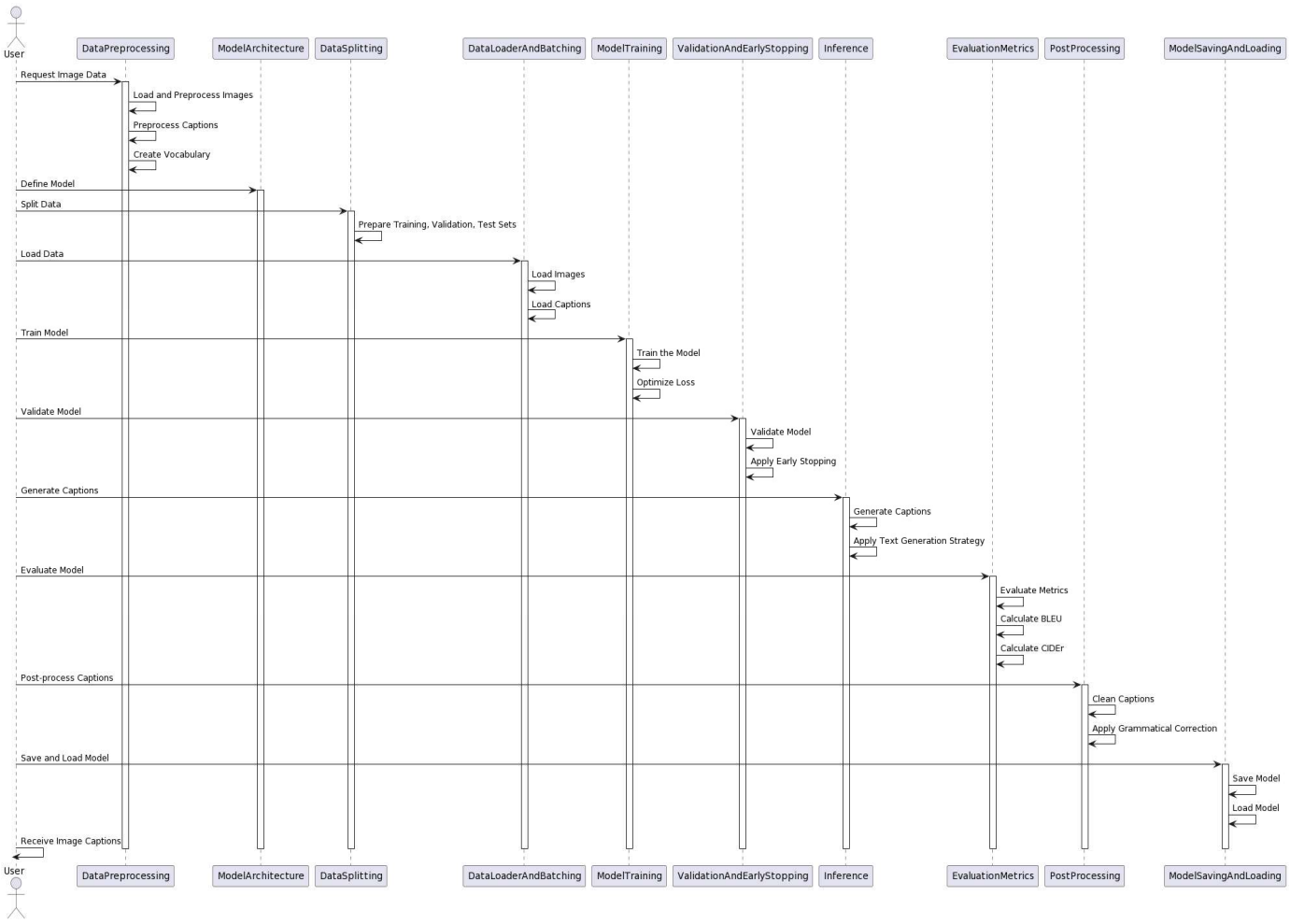


Fig 4.3: Sequence Diagram

CHAPTER-5
IMPLEMENTATION

5.1 SAMPLE CODE

```
# importing basic libraries
import os
import zipfile
import numpy as np
import random
import pickle
from math import ceil
from collections import defaultdict
from tqdm.notebook import tqdm
from PIL import Image
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline

# extracting text/caption data from downloaded zip file to new folder Flickr8k_text
zip_ref = zipfile.ZipFile('Flickr8k_text/Flickr8k_text.zip' ,'r')
zip_ref.extractall('Flickr8k_text')
zip_ref.close()

# extracting image data from downloaded zip file
zip_ref = zipfile.ZipFile('Flickr8k_Dataset.zip', 'r')
zip_ref.extractall()
zip_ref.close()

# Checking the image dataset
img_dir = 'Flicker8k_Dataset'
print('Total Images -- ',len(os.listdir(img_dir)))
photos = os.listdir(img_dir)
# print(os.listdir(img_dir))

img_numb = 4781
# displaying some images from the dataset
plt.subplots(ncols = 2, nrows = 2, figsize = (8,10))
for i, img in enumerate(os.listdir(img  dir)[img  numb : img  numb + 4]):
  plt.subplot(2,2, int(i+1))
  img = mpimg.imread(img_dir + "/" + img)
  plt.imshow(img)
def load  clean  descriptions(file, photos):
    descriptions = {}
    for line in file.split("\n"):
        words = line.split("\t")
        if len(words)<1 :
            continue
```

```
        image, image_caption = words[0], words[1:]
        if image in photos:
            if image not in descriptions:
                descriptions[image] = []
            desc = '<start> ' + " ".join(image_caption) + ' <end>'
            descriptions[image].append(desc)
    return descriptions

description = load_clean_descriptions(desc_txt, photos)
print('Type of Description variable', type(description))
print('Length of Description variable', len(description))
print('Keys of Description variable', description.keys())

# Deep learning framework for building and training models
import tensorflow as tf

## Pre-trained model for image feature extraction
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.models import Model

## Tokenizer class for captions tokenization
from tensorflow.keras.preprocessing.text import Tokenizer
## Function for padding sequences to a specific length
from tensorflow.keras.preprocessing.sequence import pad_sequences

## Class for defining Keras models
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, concatenate, Bidirectional, Dot, Activation,
RepeatVector, Multiply, Lambda

# For checking score
from nltk.translate.bleu_score import corpus_bleu
```

Extracting Features:

```
# Initialize an empty dictionary to store image features
image_features = {}

# Define the directory path where images are located
img_dir = 'Flicker8k_Dataset'

# Loop through each image in the directory
for img_name in tqdm(os.listdir(img_dir)):

    # Load the image from file
    img_path = os.path.join(img_dir, img_name)
    image = load_img(img_path, target_size=(224, 224))
```

```
# Convert image pixels to a numpy array
image = img_to_array(image)

# Reshape the data for the model
image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
# Preprocess the image for ResNet50
image = preprocess_input(image)

# Extract features using the pre-trained ResNet50 model
image_feature = model.predict(image, verbose=0) #silent

# Get the image ID by removing the file extension
image_id = img_name.split('.')[0]

# Store the extracted feature in the dictionary with the image ID as the key
image_features[image_id] = image_feature
```

Choosing Random Image from dataset

```
# choosing a random image from the Image Dataset to display
imgName = images[random.randint(1, len(os.listdir(img_dir)))]
# concatenating the Image Dataset and Image paths
imgPath  = img_dir+ '/'+ imgName

# Check if a file exists in Directory
if  os.path.exists(imgPath):
  # printing the absolute path of the image.
  file_path = os.path.abspath(imgPath)
  print(file_path)

  # Checking and displaying captions for Image
  if imgName in cleaned_description:
    # printing multiple captions one by one
    for i in range(len(cleaned_description[imgName])):
      print(i+1,'--',cleaned_description[imgName][i])
  # if caption not available
  else:
    print('Caption Available')

  # displaying image
  img = mpimg.imread(imgPath)
  plt.imshow(img)
```

Creating and Training Model

```
# Encoder model
inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.5)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)
fe2_projected = RepeatVector(max_caption_length)(fe2)
fe2_projected = Bidirectional(LSTM(256, return_sequences=True))(fe2_projected)
```

```
# Sequence feature layers
inputs2 = Input(shape=(max_caption_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.5)(se1)
se3 = Bidirectional(LSTM(256, return_sequences=True))(se2)

# Apply attention mechanism using Dot product
attention = Dot(axes=[2, 2])([fe2_projected, se3])  # Calculate attention scores

# Softmax attention scores
attention_scores = Activation('softmax')(attention)

# Apply attention scores to sequence embeddings
attention_context = Lambda(lambda x: tf.einsum('ijk,ijl->ikl', x[0], x[1]))([attention_scores, se3])

# Sum the attended sequence embeddings along the time axis
context_vector = tf.reduce_sum(attention_context, axis=1)

# Decoder model
decoder_input = concatenate([context_vector, fe2], axis=-1)
decoder1 = Dense(256, activation='relu')(decoder_input)
outputs = Dense(vocab_size, activation='softmax')(decoder1)

# Create the model
model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam')

# Visualize the model
plot_model(model, show_shapes=True)
```

Caption Generation

```
img_path = 'Flicker8k_Dataset/' + str(os.listdir('Flicker8k_Dataset')[random.randint(1,8091)])
print(generate_cap(img_path))
Image.open(img_path)
```

5.2 IMPLIMENTED SCREENSHOTS

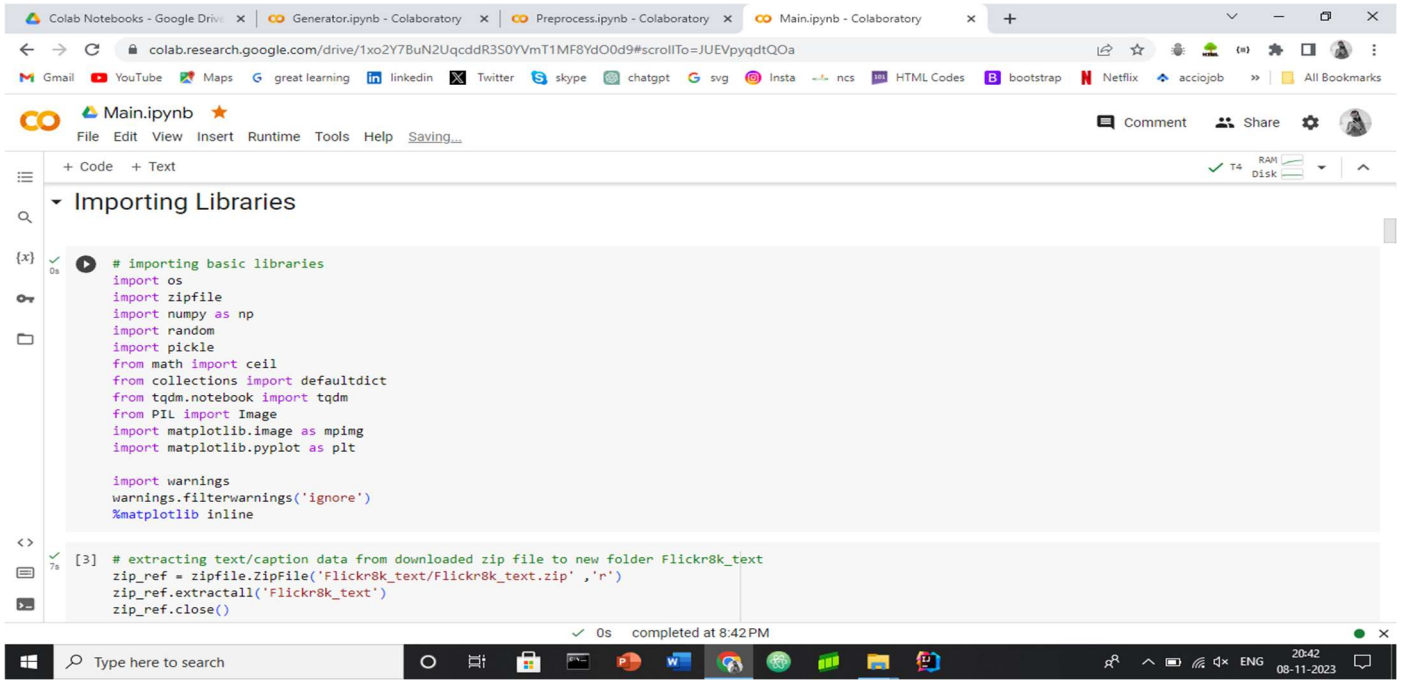


Fig 1:Import Liabraies

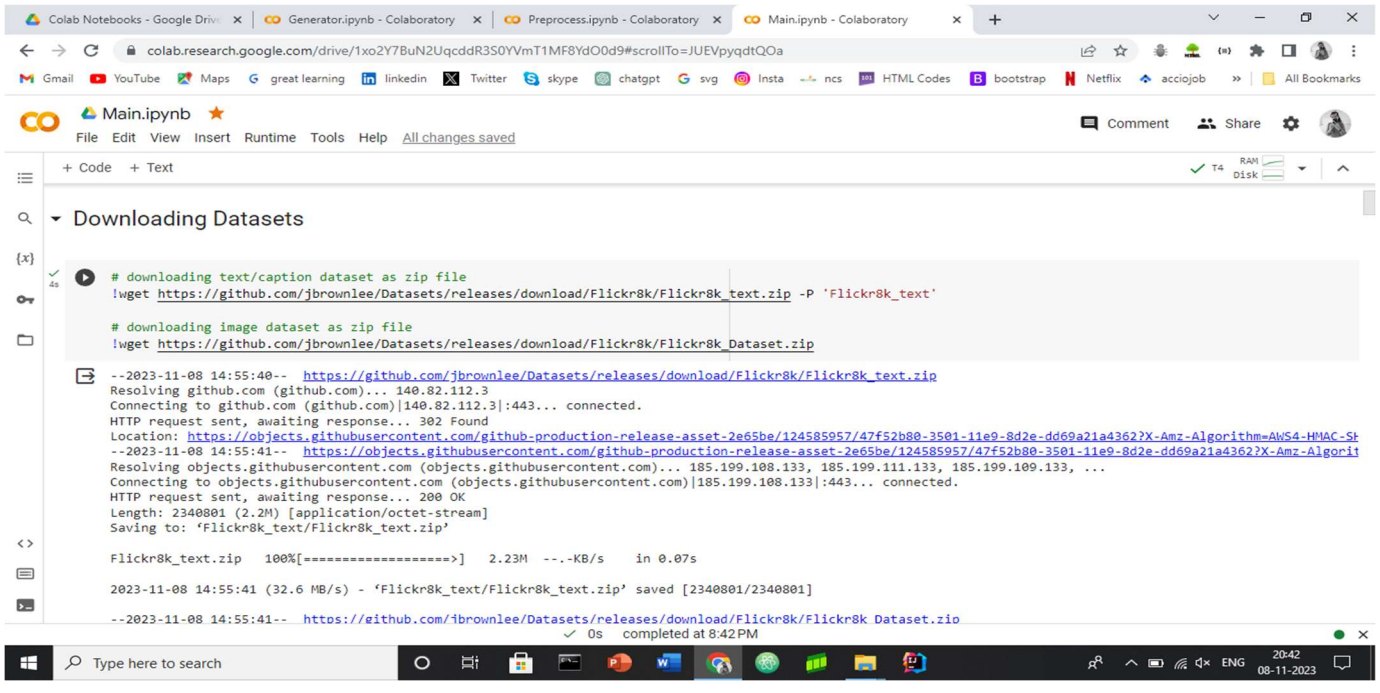


Fig 2: Load dataset & unzip

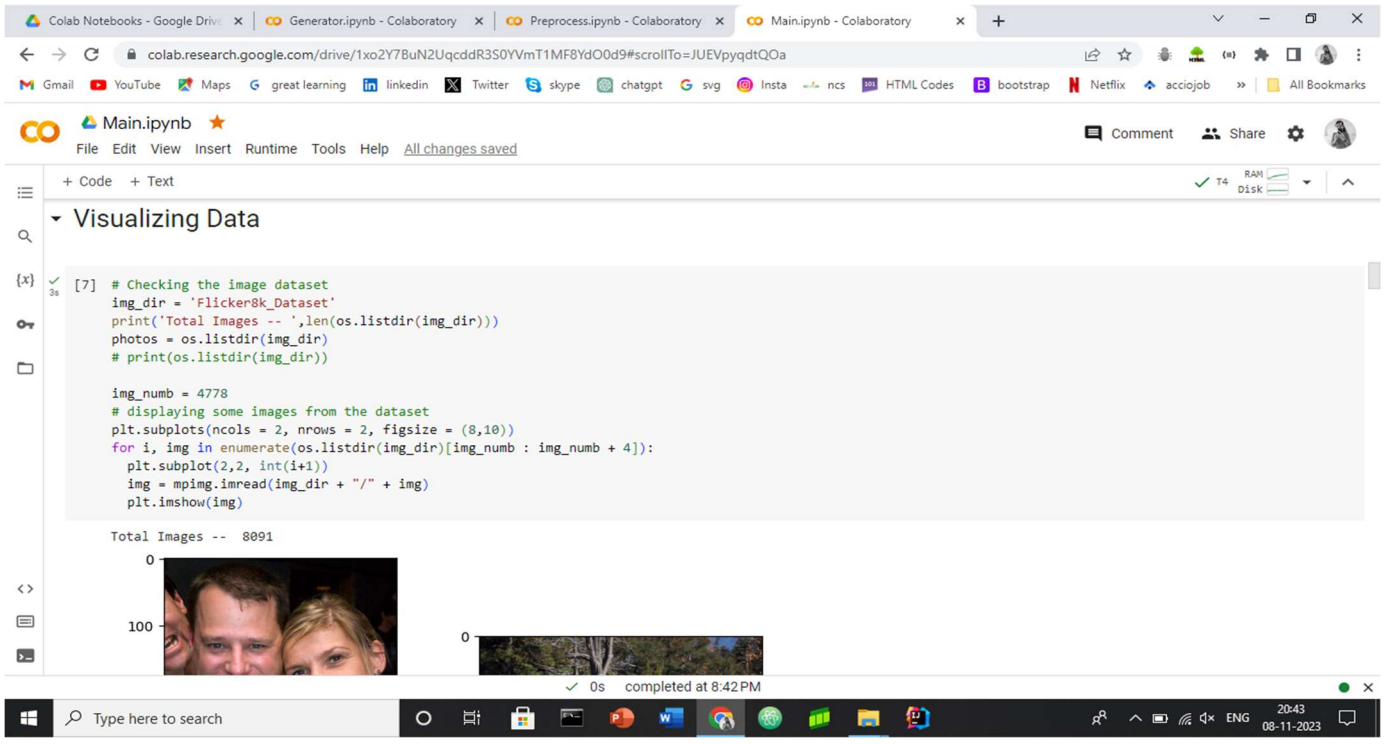


Fig 3: Loading Images

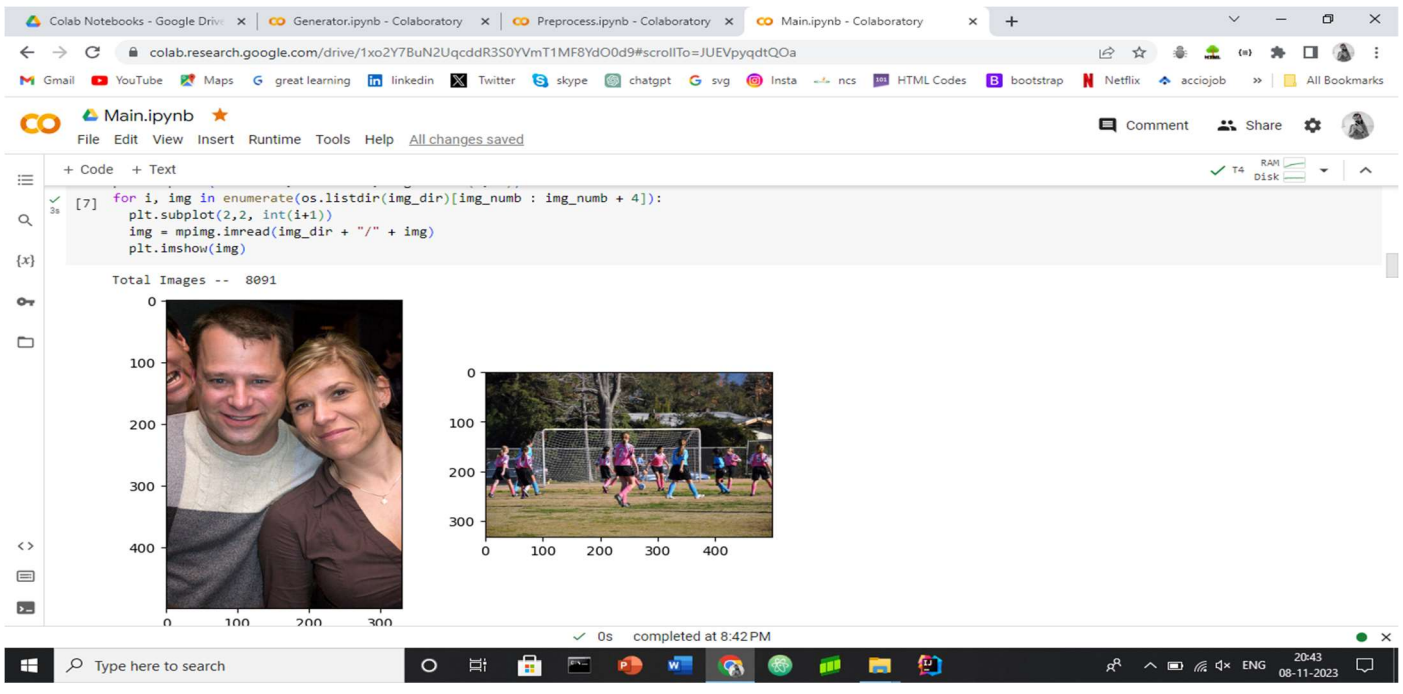


Fig 4: Extracting Image Features

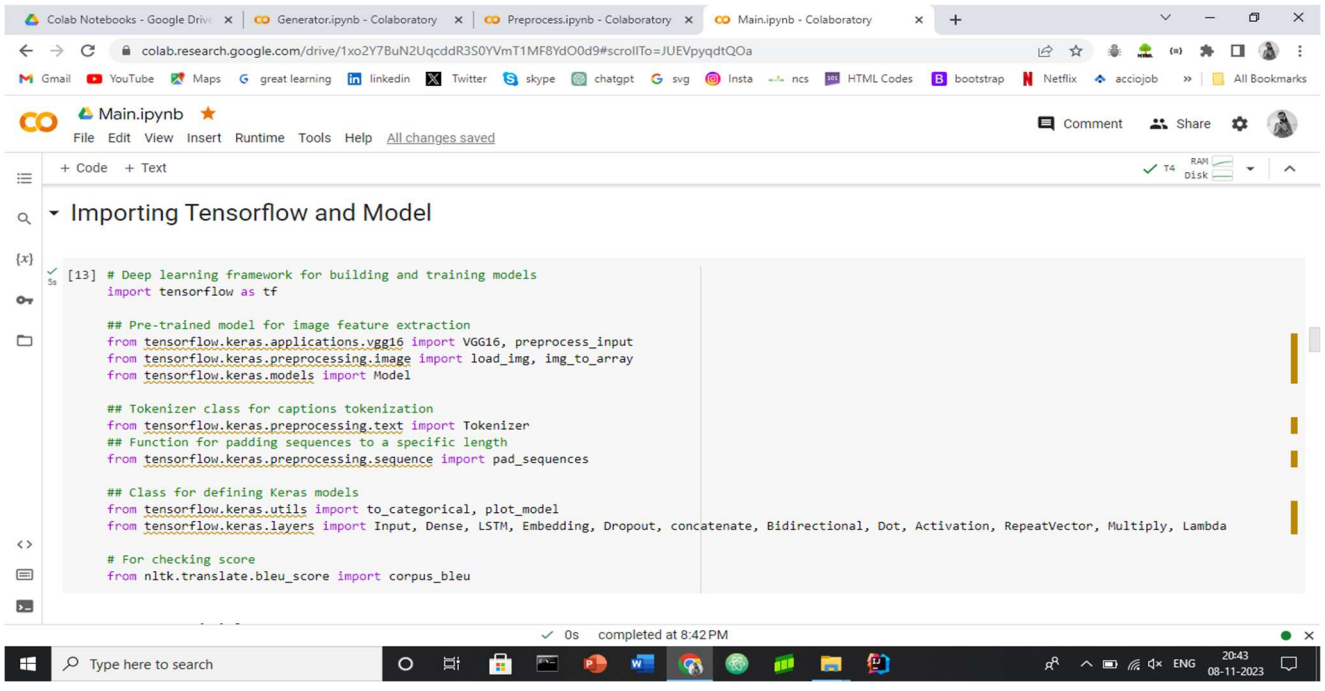


Fig 5: RNN Model

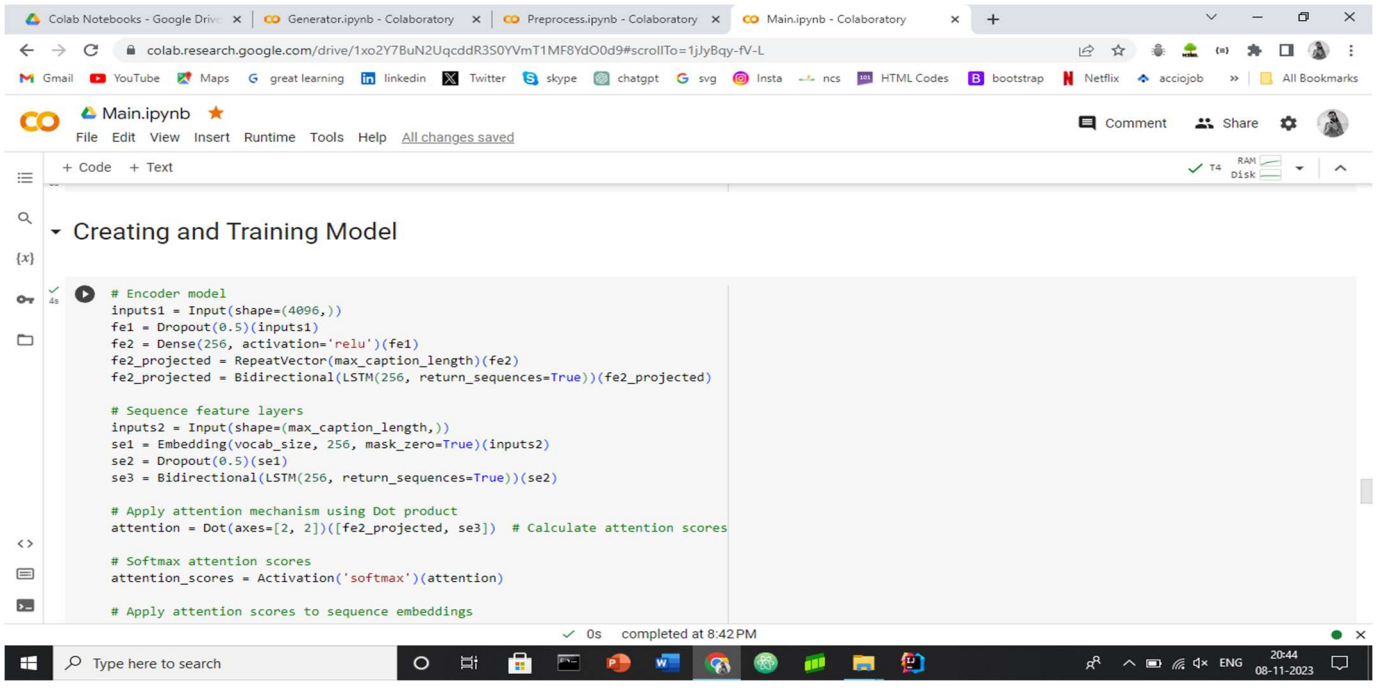


Fig 6: Training Model

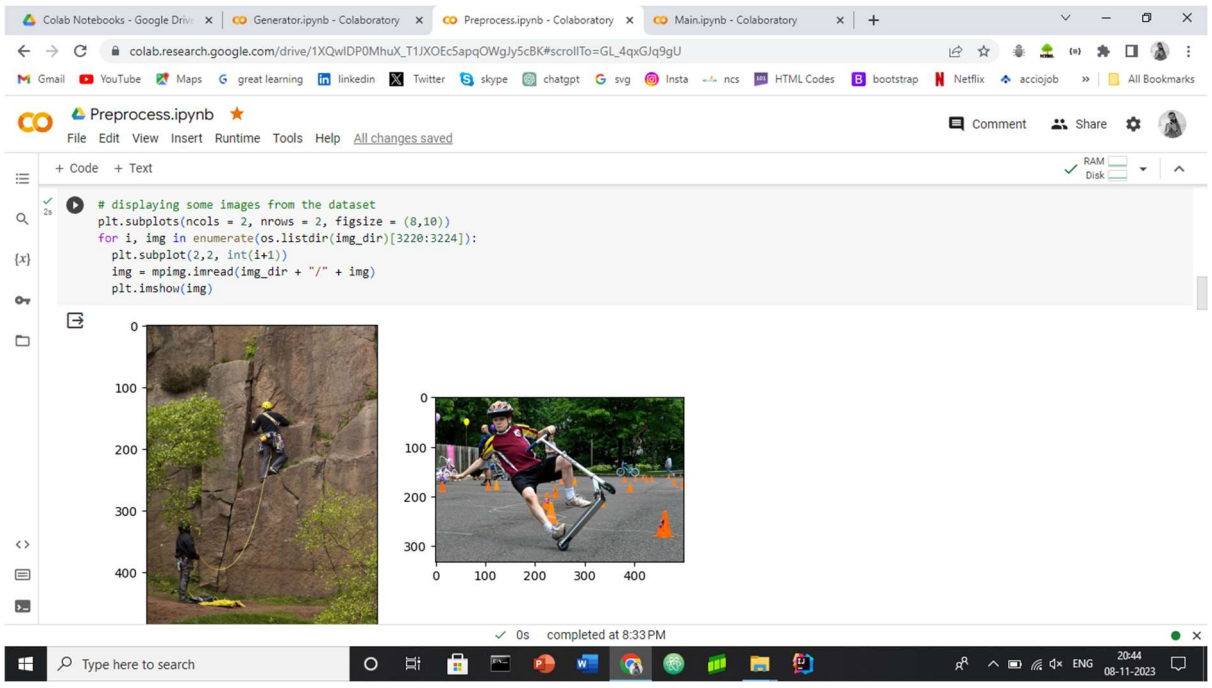


Fig 7: Output for image input

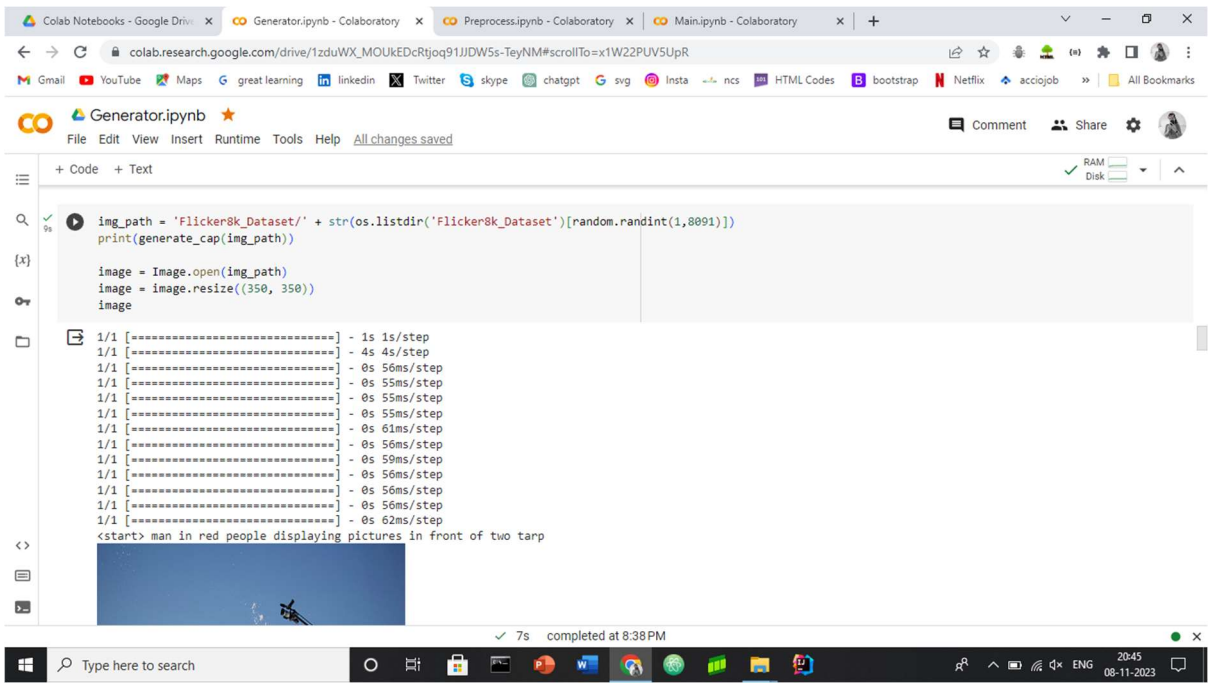


Fig 8: Load annotations

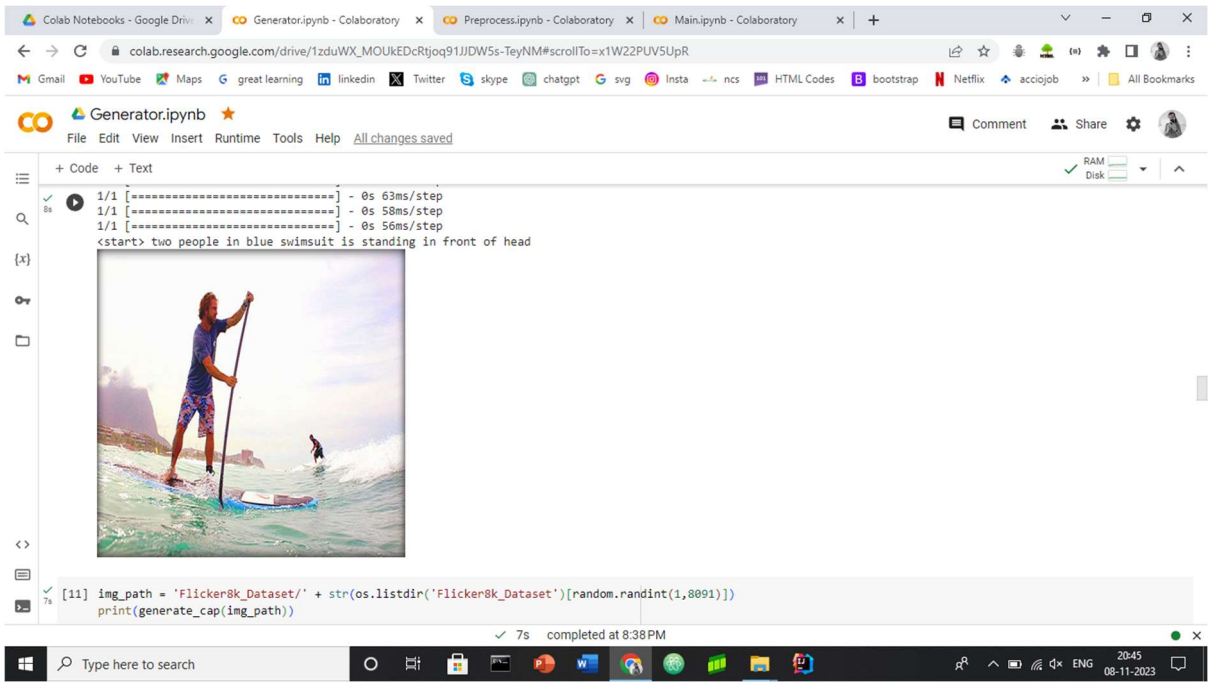


Fig9: Encoder

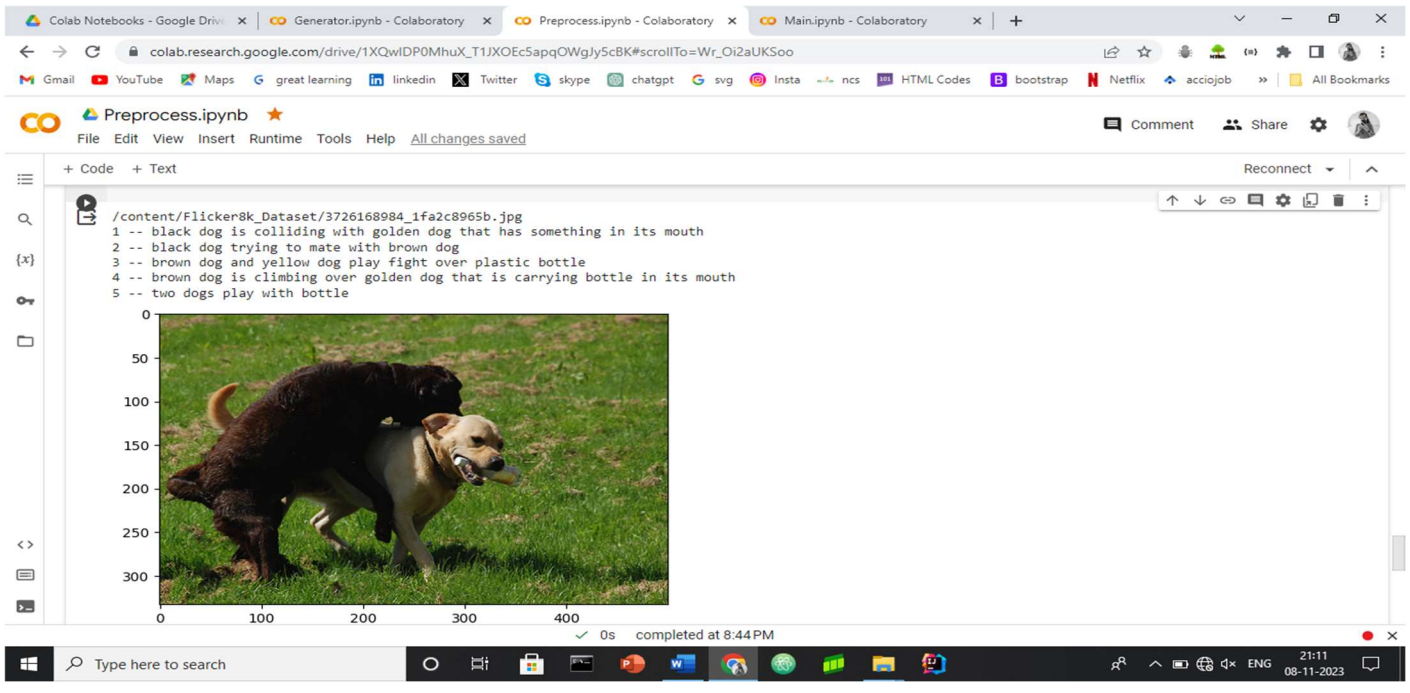


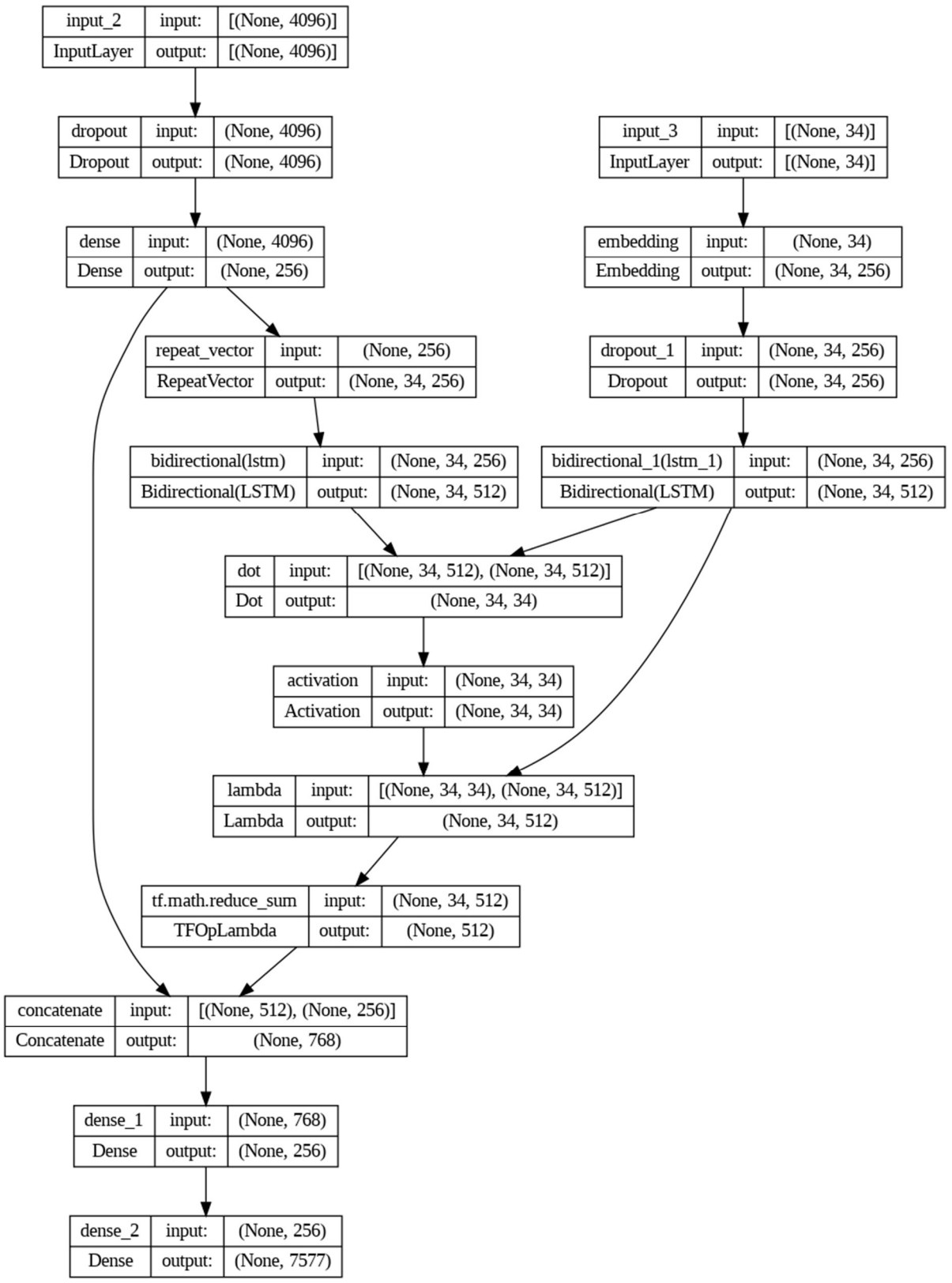
Fig 10:Generating Multiple captions for Image

IMAGE FEATURE EXTRACTION

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_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: 134260544 (512.16 MB)
Trainable params: 134260544 (512.16 MB)
Non-trainable params: 0 (0.00 Byte)

Creating and Training Model:



CHAPTER 6
CONCLUSION

6.1 RESULTS

Automatically image captioning is far from mature and there are a lot of ongoing research projects aiming for more accurate image feature extraction and semantically better sentence generation. We successfully completed what we mentioned in the project proposal, but used a smaller dataset due to limited computational power. There can be potential improvements if given more time. First of all, we directly used pre-trained CNN network as part of our pipeline without fine-tuning, so the network does not adapt to this specific training dataset. Thus, by experimenting with different CNN pre-trained networks and enabling fine-tuning, we expect to achieve a slightly higher BLEU4 score. Another potential improvement is by training on a combination of Flickr8k, Flickr30k, and MSCOCO. In general, the more diverse training dataset the network has seen, the more accurate the output will be. We all agree this projectignitesour interest in application of Machine Learning knowledge in Computer Vision and expects to explore more in the future.

6.2 FUTURE SCOPE

The future scope tries to generate captions in UI interface. In future scope ,we try to generate more accuracy and better results for it.

- In proposed system, I implemented Automatically generating random images captions
- For future scope, we need to add UI interface for generating automatically images captions
- generating audio for the randomly occurring captions.
- The CNN-LSTM model was created to automatically generate captions for the input images.
- I'd like to train our model on a larger dataset with a greater number of photographs in the future. The captions generated should be in a range of languages.
- Larger datasets and alternative CNN architectures, such as LeNet, AlexNet, GoogLeNet, ResNet, and others, were used to train and evaluate the model.

CHAPTER - 7

REFERENCES

7.1 REFERENCES

1. .[Gupta and Mannem] Ankush Gupta and Prashanth Mannem. From image annotation to image description. In Neural information processing.
2. Shuang Bai and Shan An (2018): A survey on automatic imagecaption generation.
3. D.Bahdanau, K.Cho, and Y.Bengio(2015): Neural machine translationby jointly learning to align and translate.
4. K.Xu, J.Ba, K.Cho, and R.Salakhutdinov (2018): Show attend and tell:Neural image caption generator with visual attention.
5. M.Pedersoli, T.Lucas, C.Schmid, and J.Verbeek (2017): Areas ofattention for image captioning.
6. H.R.Tavakoli,R.Shetty, B.Ali, and J.Laaksonen(2017): Paying attentionto descriptions generated by image captioning models.
7. A.Matthews, L.Xie, and X.He(2018): Sem Style-learning to generatestylized image captions using unaligned text.
8. C.Park, B.Kim, and G.kim(2018): Towards personalized image captioning via multimodal memory networks.
9. X.Chen, Ma Lin, W.Jiang, J.Yao, and W.Liu(June 2018): RegularizingRNNs for caption generation by reconstructing the past with the present.
10. T.Yao, Y.Pan, Y.Li, Z.Qiu, and T.Mei (June 2016): Boosting imagecaptioning with attributes.
11. Deep Learning in big data Analytics: A comparative study-ScientificFigure on ResarchGate (2021).
12. D.Bahdanau, K.Cho, and Y.Bengio(2015): Neural machine translationby jointly learning to align and translate.
13. K.Xu, J.Ba, K.Cho, and R.Salakhutdinov (2018): Show attend and tell: Neural image caption generator with visual attention.
14. M.Pedersoli, T.Lucas, C.Schmid, and J.Verbeek (2017): Areas of attention for image captioning.
15. H.R.Tavakoli,R.Shetty, B.Ali, and J.Laaksonen(2017): Paying attention to descriptions generated by image captioning models.
16. A.Matthews, L.Xie, and X.He(2018): Sem Style-learning to generate stylized image captions using unaligned text.
17. C.Park, B.Kim, and G.kim(2018): Towards personalized image captioning via multimodal memory networks.

18. X.Chen, Ma Lin, W.Jiang, J.Yao, and W.Liu(June 2018): Regularizing RNNs for caption generation by reconstructing the past with the present.

19. T.Yao, Y.Pan, Y.Li, Z.Qiu, and T.Mei (June 2016): Boosting image captioning with attributes.

20. Deep Learning in big data Analytics: A comparative study-Scientific Figure onResarchGate (2021).

21. Kaustav et al. (June 2016): A Facial Expression Recognition System to predict Emotions.