Group_171_DL_Assignment_2_Set_4

September 20, 2021

0.1 1. Import Libraries / Dataset

[119]:

```
import numpy as np
import pandas as pd
import cv2
import os
from glob import glob
from pickle import dump, load
import glob
import pickle
import random
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import imagesize
import time

import tensorflow as tf
from keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.resnet50 import preprocess_input

from keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical


from keras.models import Model
import os
import collections

from tensorflow.keras.utils import plot_model
from keras.layers.merge import add
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Dense, Flatten,Input, Convolution2D,␣
 ↪Dropout, LSTM, GRU, TimeDistributed, Embedding, Bidirectional, Activation,␣
 ↪RepeatVector,Concatenate
from tensorflow.keras.regularizers import l2
from tensorflow.keras import regularizers

import keras
```

[70]:
```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

```
Mounted at /content/drive
```

[71]:
```
pickle_file = 'drive/MyDrive/set_4.pkl'
images_path = 'drive/MyDrive/Image_captioning_Dataset/Flicker8k_Dataset'
google_test_image = 'drive/MyDrive/Image_captioning_Dataset/Flicker8k_Dataset/
 ↪test/test_image1.jpg'
```

[72]:
```
# Read the data from the pickle file
with open(pickle_file, 'rb') as fid:
     image_caption_data = pickle.load(fid)
```

[73]:
```
## Dataframe created based on the pickle file
image_caption_info_df = pd.DataFrame(columns=['image_name', 'caption_id',␣
 ↪'image_caption'])
```

[74]:
```
for line in image_caption_data:
  line = line.strip()
  image_row = [(line.split('\t'))[0].split('#')[0], (line.split('\t'))[0].
 ↪split('#')[1], (line.split('\t'))[1].strip('.')]
  image_row_dict = {'image_name' : (line.split('\t'))[0].split('#')[0],␣
 ↪'caption_id' : (line.split('\t'))[0].split('#')[1], 'image_caption' : (line.
 ↪split('\t'))[1].strip('.')}
  image_caption_info_df = image_caption_info_df.append(image_row_dict,␣
 ↪ignore_index=True)
```

[75]:
```
image_caption_info_df.head(10)
```

[75]:
```
                  image_name  …
image_caption
0  3312779887_7682db7827.jpg  …  A snowboarder do a trick off of a yellow
pyramid
1  2766926202_4201bf2bf9.jpg  …      Two man be play with glow stick and
sparkler
2   244760301_5809214866.jpg  …             Several hiker walk along a rocky
path
3    97105139_fae46fe8ef.jpg  …  Two person with head covering stand in a
sandy…
4  2646046871_c3a5dbb971.jpg  …  A child jump in the air with his or her shirt
…
5  3122606953_a979dd3d33.jpg  …               Two black dog walk through the
snow
```

```
6  3457604528_302396c08c.jpg  …           A child be run through the grassy
field
7  2745663684_650f84e1e6.jpg  …  a young man skateboard on a street wear a
blac…
8  3308997740_91765ecdcc.jpg  …  A girl with a beanie stand in front of a
windo…
9  2619454551_c4bb726a85.jpg  …              A bright colored bird and a small
dog

[10 rows x 3 columns]
```

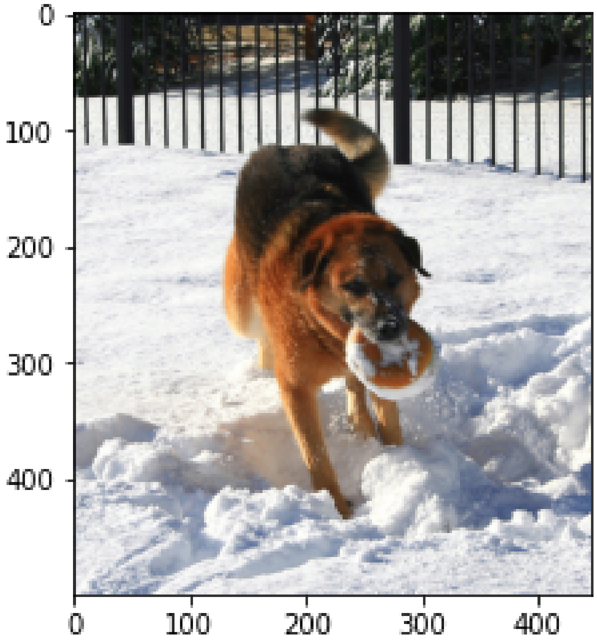
[76]:

```
# Removing file names with *.jpg.1 from pickle file data's dataframe
image_caption_info_df =␣
↪image_caption_info_df[image_caption_info_df['image_name'].str.contains(".jpg.
↪1")==False]
```

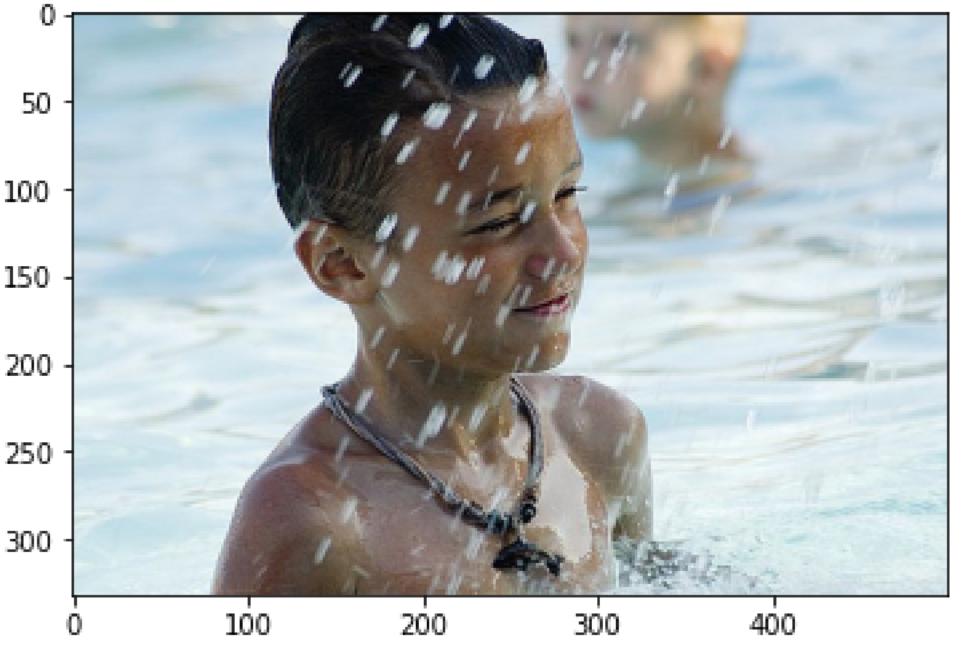
0.2 2. Data Visualization and augmentation

[77]:

```
# from PIL import Image
# import imagesize
# import matplotlib.image as mpimg

random_list = random.sample(range(1,25000), 2)
for i in random_list:
  img = mpimg.imread(images_path + '/' + image_caption_info_df.
↪iloc[i]['image_name'])
  plt.imshow(img)
  plt.show()
  print(image_caption_info_df.iloc[i]['image_caption'])
  print(' ')
```



```
A dog with a Frisbee in the snow
```



```
A young boy with a necklace on in the water
```

[78]:

```
# Creating the List of images in the dataset folder
list_of_images = []
for i in image_caption_info_df['image_name']:
  path = images_path+'/'+i
  if path in list_of_images:
    continue
  else:
    list_of_images.append(path)

print(len(list_of_images))
```

```
8019
```

0.3 3. Model Building

[79]:

```
## Using Pretrained         Resnet-50 model         trained         on ImageNet␣
 ↪dataset as base model
base_model = ResNet50(weights='imagenet', include_top=True)
```

[80]:

```
# from keras.models import Model
last = base_model.layers[-2].output
model_emb = Model(inputs = base_model.input, outputs = last)
model_emb.summary()
```

```
Model: "model_2"

--------------------------------------------------------------------------------
------------------
Layer (type)                    Output Shape         Param #     Connected to
================================================================================
==================
input_3 (InputLayer)            [(None, 224, 224, 3) 0

--------------------------------------------------------------------------------
------------------
conv1_pad (ZeroPadding2D)       (None, 230, 230, 3)  0           input_3[0][0]

--------------------------------------------------------------------------------
------------------
conv1_conv (Conv2D)             (None, 112, 112, 64) 9472        conv1_pad[0][0]

--------------------------------------------------------------------------------
------------------
conv1_bn (BatchNormalization)   (None, 112, 112, 64) 256
conv1_conv[0][0]

--------------------------------------------------------------------------------
------------------
conv1_relu (Activation)         (None, 112, 112, 64) 0           conv1_bn[0][0]

--------------------------------------------------------------------------------
------------------
```

```
pool1_pad (ZeroPadding2D)       (None, 114, 114, 64) 0
conv1_relu[0][0]
--------------------------------------------------------------------------------
------------------
pool1_pool (MaxPooling2D)       (None, 56, 56, 64)   0           pool1_pad[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_1_conv (Conv2D)    (None, 56, 56, 64)   4160
pool1_pool[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_1_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block1_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_1_relu (Activation (None, 56, 56, 64)   0
conv2_block1_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_2_conv (Conv2D)    (None, 56, 56, 64)   36928
conv2_block1_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_2_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block1_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_2_relu (Activation (None, 56, 56, 64)   0
conv2_block1_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_0_conv (Conv2D)    (None, 56, 56, 256)  16640
pool1_pool[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_3_conv (Conv2D)    (None, 56, 56, 256)  16640
conv2_block1_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_0_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block1_0_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_3_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block1_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_add (Add)          (None, 56, 56, 256)  0
```

```
conv2_block1_0_bn[0][0]
conv2_block1_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block1_out (Activation)   (None, 56, 56, 256)  0
conv2_block1_add[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_1_conv (Conv2D)    (None, 56, 56, 64)   16448
conv2_block1_out[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_1_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block2_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_1_relu (Activation (None, 56, 56, 64)   0
conv2_block2_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_2_conv (Conv2D)    (None, 56, 56, 64)   36928
conv2_block2_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_2_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block2_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_2_relu (Activation (None, 56, 56, 64)   0
conv2_block2_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_3_conv (Conv2D)    (None, 56, 56, 256)  16640
conv2_block2_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_3_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block2_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_add (Add)          (None, 56, 56, 256)  0
conv2_block1_out[0][0]
conv2_block2_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block2_out (Activation)   (None, 56, 56, 256)  0
conv2_block2_add[0][0]
--------------------------------------------------------------------------------
```

```
------------------
conv2_block3_1_conv (Conv2D)    (None, 56, 56, 64)   16448
conv2_block2_out[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_1_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block3_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_1_relu (Activation (None, 56, 56, 64)   0
conv2_block3_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_2_conv (Conv2D)    (None, 56, 56, 64)   36928
conv2_block3_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_2_bn (BatchNormali (None, 56, 56, 64)   256
conv2_block3_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_2_relu (Activation (None, 56, 56, 64)   0
conv2_block3_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_3_conv (Conv2D)    (None, 56, 56, 256)  16640
conv2_block3_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_3_bn (BatchNormali (None, 56, 56, 256)  1024
conv2_block3_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_add (Add)          (None, 56, 56, 256)  0
conv2_block2_out[0][0]
conv2_block3_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv2_block3_out (Activation)   (None, 56, 56, 256)  0
conv2_block3_add[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_1_conv (Conv2D)    (None, 28, 28, 128)  32896
conv2_block3_out[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block1_1_conv[0][0]
```

```
--------------------------------------------------------------------------------
------------------
conv3_block1_1_relu (Activation (None, 28, 28, 128)  0
conv3_block1_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block1_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block1_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_2_relu (Activation (None, 28, 28, 128)  0
conv3_block1_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_0_conv (Conv2D)    (None, 28, 28, 512)  131584
conv2_block3_out[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block1_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_0_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block1_0_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block1_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_add (Add)          (None, 28, 28, 512)  0
conv3_block1_0_bn[0][0]
conv3_block1_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block1_out (Activation)   (None, 28, 28, 512)  0
conv3_block1_add[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_1_conv (Conv2D)    (None, 28, 28, 128)  65664
conv3_block1_out[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block2_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_1_relu (Activation (None, 28, 28, 128)  0
conv3_block2_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block2_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block2_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_2_relu (Activation (None, 28, 28, 128)  0
conv3_block2_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block2_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block2_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_add (Add)          (None, 28, 28, 512)  0
conv3_block1_out[0][0]
conv3_block2_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block2_out (Activation)   (None, 28, 28, 512)  0
conv3_block2_add[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_1_conv (Conv2D)    (None, 28, 28, 128)  65664
conv3_block2_out[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block3_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_1_relu (Activation (None, 28, 28, 128)  0
conv3_block3_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
```

```
conv3_block3_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block3_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block3_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_2_relu (Activation (None, 28, 28, 128)  0
conv3_block3_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block3_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block3_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_add (Add)          (None, 28, 28, 512)  0
conv3_block2_out[0][0]
conv3_block3_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block3_out (Activation)   (None, 28, 28, 512)  0
conv3_block3_add[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_1_conv (Conv2D)    (None, 28, 28, 128)  65664
conv3_block3_out[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_1_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block4_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_1_relu (Activation (None, 28, 28, 128)  0
conv3_block4_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_2_conv (Conv2D)    (None, 28, 28, 128)  147584
conv3_block4_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_2_bn (BatchNormali (None, 28, 28, 128)  512
conv3_block4_2_conv[0][0]
--------------------------------------------------------------------------------
```

```
------------------
conv3_block4_2_relu (Activation (None, 28, 28, 128)  0
conv3_block4_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_3_conv (Conv2D)    (None, 28, 28, 512)  66048
conv3_block4_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_3_bn (BatchNormali (None, 28, 28, 512)  2048
conv3_block4_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_add (Add)          (None, 28, 28, 512)  0
conv3_block3_out[0][0]
conv3_block4_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv3_block4_out (Activation)   (None, 28, 28, 512)  0
conv3_block4_add[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_1_conv (Conv2D)    (None, 14, 14, 256)  131328
conv3_block4_out[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block1_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_1_relu (Activation (None, 14, 14, 256)  0
conv4_block1_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block1_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block1_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_2_relu (Activation (None, 14, 14, 256)  0
conv4_block1_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_0_conv (Conv2D)    (None, 14, 14, 1024) 525312
conv3_block4_out[0][0]
```

```
--------------------------------------------------------------------------------
------------------
conv4_block1_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block1_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_0_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block1_0_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block1_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_add (Add)          (None, 14, 14, 1024) 0
conv4_block1_0_bn[0][0]
conv4_block1_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block1_out (Activation)   (None, 14, 14, 1024) 0
conv4_block1_add[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block1_out[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block2_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_1_relu (Activation (None, 14, 14, 256)  0
conv4_block2_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block2_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block2_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_2_relu (Activation (None, 14, 14, 256)  0
conv4_block2_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
```

```
conv4_block2_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block2_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_add (Add)          (None, 14, 14, 1024) 0
conv4_block1_out[0][0]
conv4_block2_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block2_out (Activation)   (None, 14, 14, 1024) 0
conv4_block2_add[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block2_out[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block3_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_1_relu (Activation (None, 14, 14, 256)  0
conv4_block3_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block3_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block3_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_2_relu (Activation (None, 14, 14, 256)  0
conv4_block3_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block3_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block3_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
```

```
conv4_block3_add (Add)          (None, 14, 14, 1024) 0
conv4_block2_out[0][0]
conv4_block3_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block3_out (Activation)   (None, 14, 14, 1024) 0
conv4_block3_add[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block3_out[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block4_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_1_relu (Activation (None, 14, 14, 256)  0
conv4_block4_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block4_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block4_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_2_relu (Activation (None, 14, 14, 256)  0
conv4_block4_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block4_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block4_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_add (Add)          (None, 14, 14, 1024) 0
conv4_block3_out[0][0]
conv4_block4_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block4_out (Activation)   (None, 14, 14, 1024) 0
conv4_block4_add[0][0]
```

```
--------------------------------------------------------------------------------
------------------
conv4_block5_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block4_out[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block5_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_1_relu (Activation (None, 14, 14, 256)  0
conv4_block5_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block5_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block5_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_2_relu (Activation (None, 14, 14, 256)  0
conv4_block5_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block5_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block5_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_add (Add)          (None, 14, 14, 1024) 0
conv4_block4_out[0][0]
conv4_block5_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block5_out (Activation)   (None, 14, 14, 1024) 0
conv4_block5_add[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_1_conv (Conv2D)    (None, 14, 14, 256)  262400
conv4_block5_out[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_1_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block6_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_1_relu (Activation (None, 14, 14, 256)  0
conv4_block6_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_2_conv (Conv2D)    (None, 14, 14, 256)  590080
conv4_block6_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_2_bn (BatchNormali (None, 14, 14, 256)  1024
conv4_block6_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_2_relu (Activation (None, 14, 14, 256)  0
conv4_block6_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_3_conv (Conv2D)    (None, 14, 14, 1024) 263168
conv4_block6_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_3_bn (BatchNormali (None, 14, 14, 1024) 4096
conv4_block6_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_add (Add)          (None, 14, 14, 1024) 0
conv4_block5_out[0][0]
conv4_block6_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv4_block6_out (Activation)   (None, 14, 14, 1024) 0
conv4_block6_add[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_1_conv (Conv2D)    (None, 7, 7, 512)    524800
conv4_block6_out[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_1_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block1_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_1_relu (Activation (None, 7, 7, 512)    0
conv5_block1_1_bn[0][0]
--------------------------------------------------------------------------------
------------------
```

```
conv5_block1_2_conv (Conv2D)    (None, 7, 7, 512)    2359808
conv5_block1_1_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_2_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block1_2_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_2_relu (Activation (None, 7, 7, 512)    0
conv5_block1_2_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_0_conv (Conv2D)    (None, 7, 7, 2048)   2099200
conv4_block6_out[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624
conv5_block1_2_relu[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_0_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block1_0_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_3_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block1_3_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_add (Add)          (None, 7, 7, 2048)   0
conv5_block1_0_bn[0][0]
conv5_block1_3_bn[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block1_out (Activation)   (None, 7, 7, 2048)   0
conv5_block1_add[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block2_1_conv (Conv2D)    (None, 7, 7, 512)    1049088
conv5_block1_out[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block2_1_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block2_1_conv[0][0]
--------------------------------------------------------------------------------
------------------
conv5_block2_1_relu (Activation (None, 7, 7, 512)    0
conv5_block2_1_bn[0][0]
--------------------------------------------------------------------------------
```

```
------------------
conv5_block2_2_conv (Conv2D)    (None, 7, 7, 512)    2359808
conv5_block2_1_relu[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block2_2_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block2_2_conv[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block2_2_relu (Activation (None, 7, 7, 512)    0
conv5_block2_2_bn[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block2_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624
conv5_block2_2_relu[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block2_3_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block2_3_conv[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block2_add (Add)          (None, 7, 7, 2048)   0
conv5_block1_out[0][0]
conv5_block2_3_bn[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block2_out (Activation)   (None, 7, 7, 2048)   0
conv5_block2_add[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_1_conv (Conv2D)    (None, 7, 7, 512)    1049088
conv5_block2_out[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_1_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block3_1_conv[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_1_relu (Activation (None, 7, 7, 512)    0
conv5_block3_1_bn[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_2_conv (Conv2D)    (None, 7, 7, 512)    2359808
conv5_block3_1_relu[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_2_bn (BatchNormali (None, 7, 7, 512)    2048
conv5_block3_2_conv[0][0]
```

```
--------------------------------------------------------------------------------
------------------
conv5_block3_2_relu (Activation (None, 7, 7, 512)    0
conv5_block3_2_bn[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624
conv5_block3_2_relu[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192
conv5_block3_3_conv[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_add (Add)          (None, 7, 7, 2048)   0
conv5_block2_out[0][0]
conv5_block3_3_bn[0][0]

--------------------------------------------------------------------------------
------------------
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0
conv5_block3_add[0][0]

--------------------------------------------------------------------------------
------------------
avg_pool (GlobalAveragePooling2 (None, 2048)         0
conv5_block3_out[0][0]
================================================================================
==================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120

--------------------------------------------------------------------------------
------------------
```

0.3.1 Feature Extraction

[81]:

```python
# Limiting to 1500 images for extracting the features as choosing more images causing colab environment to crash while array manipulation.

images_features = {}
count = 0
for i in list_of_images:
  img = cv2.imread(i)
  img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
  img = cv2.resize(img, (224, 224))

  img = img.reshape(1, 224, 224, 3)
  pred = model_emb.predict(img).reshape(2048,)
  im_name = i.split('/')[-1]
  img_name = im_name
  images_features[img_name] = pred

  count += 1
  limit = 1500
  if count > limit:
    break
  elif count % 50 == 0:
    print('Count of Images processed : ', count)
print("Extraction Completed")
```

```
Count of Images processed :  50
Count of Images processed :  100
Count of Images processed :  150
Count of Images processed :  200
Count of Images processed :  250
Count of Images processed :  300
Count of Images processed :  350
Count of Images processed :  400
Count of Images processed :  450
Count of Images processed :  500
Count of Images processed :  550
Count of Images processed :  600
Count of Images processed :  650
Count of Images processed :  700
Count of Images processed :  750
Count of Images processed :  800
Count of Images processed :  850
Count of Images processed :  900
Count of Images processed :  950
Count of Images processed :  1000
Count of Images processed :  1050
Count of Images processed :  1100
Count of Images processed :  1150
Count of Images processed :  1200
Count of Images processed :  1250
Count of Images processed :  1300
Count of Images processed :  1350
Count of Images processed :  1400
Count of Images processed :  1450
Count of Images processed :  1500
Extraction Completed
```

[82]:

```
# Preparing Image name - Captions Dictionary using collections.
#import collections
```

```
# Mapping  all "Captions" to it's respective "Image Name".
captions_dict = collections.defaultdict(list)

for file_name in image_caption_info_df['image_name'].unique():
  if file_name in list(images_features.keys()):
    image_path = file_name
    df = image_caption_info_df.loc[image_caption_info_df['image_name'] ==␣
 ↪file_name]
    for ind in df.index:
      caption = df['image_caption'][ind]
      captions_dict[image_path].append(caption)
```

[83]:
```
len(captions_dict)
```

[83]:
```
1501
```

[84]:
```
## Sample Visualization of Images from dictionary along with it's captions.
# import matplotlib.pyplot as plt
for i in range(5):
  plt.figure()
  img_name = list_of_images[i]
  img = mpimg.imread(list_of_images[i])
  plt.xlabel(captions_dict[img_name.split('/')[-1]])
  plt.imshow(img)
  plt.show()
```

['A snowboarder do a trick off of a yellow pyramid ', 'A snowboarder hang upside down from his board during a maneuver in front of a crowd of person ']

['Two man be play with glow stick and sparkler ', 'Two child wear glow necklace play with sparkler while stand in water ', 'Two child play with firework in shallow water', 'Two person be stand in shallow water , wave sparkler around']

['Several hiker walk along a rocky path ', 'A group of man wear similar clothing backpack through the countryside ', 'People walk', 'A group of person in matching outfit hike up a trail with one person lag behind ']

['Two person with head covering stand in a sandy field ', 'Two man in robe wave at an approach jeep travel through the sand ', 'Two man in keffiyahs stand next to car in the desert and wave at a pass vehicle ']

['A child jump in the air with his or her shirt fly open ', 'A boy in an open Hawaiian shirt be make the Longhorn symbol with his hand ', 'A child in Hawaiian clothing jump and pose in a low cut yard nearby a fence and building ']

[85]:

```python
## Sample Visualization of Images from dictionary along with it's captions based on the 'feature'

for file_name in images_features.keys():
  plt.figure()
  img_name = images_path +'/'+ file_name
  img = mpimg.imread(img_name)
  plt.xlabel(captions_dict[img_name.split('/')[-1]])
  plt.imshow(img)
  plt.show()
  break
```

```
['A snowboarder do a trick off of a yellow pyramid ', 'A snowboarder hang upside down from his board during a maneuver in front of a crowd of person ']
```

[86]:

```
## Appending the 'captions' with 'startseq' & 'endseq' for processing.
def preprocessed(txt):
  modified = txt.lower()
  modified = 'startseq ' + modified + ' endseq'
  return modified


for k, v in captions_dict.items():
  for vv in v:
    captions_dict[k][v.index(vv)] = preprocessed(vv)
```

[87]:

```
# Preparing the count of words from the dictionary

count_words = {}
for k, vv in captions_dict.items():
  for v in vv:
    for word in v.split():
      if word not in count_words:
        count_words[word] = 0
      else:
        count_words[word] += 1

print(len(count_words))
```

```
2678
```

[88]:

```
# Preparing the dictionary of words from 'count of words'
THRESH = -1
count = 1
new_dict = {}
for k, v in count_words.items():
  if count_words[k] > THRESH:
    new_dict[k] = count
    count += 1
```

[89]:

```
##Addind string 'OUT' in the word dictionary to mark the end of dictionary.
new_dict['<OUT>'] = len(new_dict)
```

[90]:

```
# Saving the new dictionary for future reference
from pickle import dump
dump(new_dict,open('new_dict1500.p','wb'))
print('Saved new_dict.p')
```

```
Saved new_dict.p
```

[91]:

```
# Backing up caption-image dictionary
captions_backup = captions_dict.copy()
captions_dict = captions_backup.copy()
```

[92]:

```
## Mapping image with it's rspective captions list for the model usage purpose.

for k, vv in captions_dict.items():
  for v in vv:
    encoded = []
    for word in v.split():
      if word not in new_dict:
        encoded.append(new_dict['<OUT>'])
      else:
        encoded.append(new_dict[word])
    captions_dict[k][vv.index(v)] = encoded
```

[33]:

```
# from keras.preprocessing.sequence import pad_sequences
# from tensorflow.keras.utils import to_categorical
```

0.4 Building 'Generator' Function

[93]:

```
## Finding the maximum number of captions for the image
MAX_LEN = 0
for k, vv in captions_dict.items():
  for v in vv:
    if len(v) > MAX_LEN:
      MAX_LEN = len(v)
```

[94]:

```
## Preparing input data to train the model
Batch_size = 5000
VOCAB_SIZE = len(new_dict)

def generator(photo, caption):
  n_samples = 0
  X = []
  y_in = []
  y_out = []
  for k, vv in caption.items():
    for v in vv:
      for i in range(1, len(v)):
        X.append(photo[k])
        in_seq= [v[:i]]
        out_seq = v[i]
        in_seq = pad_sequences(in_seq, maxlen=MAX_LEN, padding='post',␣
↪truncating='post')[0]
        out_seq = to_categorical([out_seq], num_classes=VOCAB_SIZE)[0]
        y_in.append(in_seq)
        y_out.append(out_seq)
  return X, y_in, y_out
```

[95]:

```
## Creating the data generator using the dictionary and extracted image features
# from keras.preprocessing.sequence import pad_sequences
# from tensorflow.keras.utils import to_categorical

captions_dict = dict(captions_dict)
X, y_in, y_out = generator(images_features, captions_dict)
```

[96]:

```
## Describing the length of inputs available for the model training
print('X length:',len(X))
print('y_in length:',len(y_in))
print('y_out length:',len(y_out))
```

```
X length: 60903
y_in length: 60903
y_out length: 60903
```

[97]:

```
## Converting the input data X to numpy array.
X = np.array(X)
```

[98]:

```
y_in = np.array(y_in, dtype='float64')
```

[99]:

```
y_out = np.array(y_out, dtype='float64')
```

[100]:

```
## Input shapes
X.shape, y_in.shape, y_out.shape
```

[100]:

```
((60903, 2048), (60903, 35), (60903, 2679))
```

[101]:

```
## Sample Numpy X input data
X[710]
```

[101]:

```
array([0.0459689 , 0.00814397, 0.08347236, …, 0.40034485, 0.24291696,
       0.0465971 ], dtype=float32)
```

[102]:

```
## Sample y_in data
y_in[2]
```

```
[102]: array([1., 2., 3., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
              0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
              0.])
```

0.5 4. Model Compilation

```
[116]: ## 5 layered        GRU layer        model, with tanh activation function, L2␣
        ↪regularization and a dropout layer
       ## Using Adam optimizer with learning rate of 0.001. With learning rates 0.1 ,␣
        ↪0.01 , the loss values were heavy and the accuracy of the network are very␣
        ↪low, compared to the learning rate 0.001.
       ## Hence choosing 0.001 as the learnig rate.

       embedding_size = 128
       max_len = MAX_LEN
       vocab_size = len(new_dict)

       image_model = Sequential()
       # image_model.add(Dropout(0.1))
       image_model.add(Dense(embedding_size, input_shape=(2048,), activation='relu'))
       image_model.add(RepeatVector(max_len))

       image_model.summary()

       language_model = Sequential()

       language_model.add(Embedding(input_dim=vocab_size, output_dim=embedding_size,␣
        ↪input_length=max_len))
       language_model.add(GRU(256, return_sequences=True))
       language_model.add(Dropout(0.1))
       language_model.add(TimeDistributed(Dense(embedding_size)))

       language_model.summary()

       print()
       print('Combining the image and language models')
       conca = Concatenate()([image_model.output, language_model.output])
       x = GRU(128, return_sequences=True)(conca)
       x = GRU(512, return_sequences=True)(x)
       x = GRU(512, return_sequences=True)(x)
       x = GRU(512, return_sequences=True)(x)
       x = GRU(512, return_sequences=False)(x)
       x = Dense(vocab_size)(x)
       out = Activation('softmax')(x)
       model = Model(inputs=[image_model.input, language_model.input], outputs = out)
```

```
model.compile(loss='categorical_crossentropy', optimizer = Adam(learning_rate =␣
 ↪0.001), metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_17"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 128)               262272

_________________________________________________________________
repeat_vector_11 (RepeatVect (None, 35, 128)           0
=================================================================
Total params: 262,272
Trainable params: 262,272
Non-trainable params: 0

_________________________________________________________________
Model: "sequential_18"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, 35, 128)           342912

_________________________________________________________________
gru_15 (GRU)                 (None, 35, 256)           296448

_________________________________________________________________
dropout_10 (Dropout)         (None, 35, 256)           0

_________________________________________________________________
time_distributed_5 (TimeDist (None, 35, 128)           32896
=================================================================
Total params: 672,256
Trainable params: 672,256
Non-trainable params: 0

_________________________________________________________________

Combining the image and language models
Model: "model_4"

________________________________________________________________________________
__________________
Layer (type)                    Output Shape         Param #     Connected to
================================================================================
==================
embedding_5_input (InputLayer)  [(None, 35)]         0

________________________________________________________________________________
__________________
embedding_5 (Embedding)         (None, 35, 128)      342912
embedding_5_input[0][0]

________________________________________________________________________________
__________________
dense_18_input (InputLayer)     [(None, 2048)]       0
```

```
--------------------------------------------------------------------------------
------------------
gru_15 (GRU)                    (None, 35, 256)      296448
embedding_5[0][0]

--------------------------------------------------------------------------------
------------------
dense_18 (Dense)                (None, 128)          262272
dense_18_input[0][0]

--------------------------------------------------------------------------------
------------------
dropout_10 (Dropout)            (None, 35, 256)      0           gru_15[0][0]

--------------------------------------------------------------------------------
------------------
repeat_vector_11 (RepeatVector) (None, 35, 128)      0           dense_18[0][0]

--------------------------------------------------------------------------------
------------------
time_distributed_5 (TimeDistrib (None, 35, 128)      32896
dropout_10[0][0]

--------------------------------------------------------------------------------
------------------
concatenate_5 (Concatenate)     (None, 35, 256)      0
repeat_vector_11[0][0]
time_distributed_5[0][0]

--------------------------------------------------------------------------------
------------------
gru_16 (GRU)                    (None, 35, 128)      148224
concatenate_5[0][0]

--------------------------------------------------------------------------------
------------------
gru_17 (GRU)                    (None, 35, 512)      986112      gru_16[0][0]

--------------------------------------------------------------------------------
------------------
gru_18 (GRU)                    (None, 35, 512)      1575936     gru_17[0][0]

--------------------------------------------------------------------------------
------------------
gru_19 (GRU)                    (None, 35, 512)      1575936     gru_18[0][0]

--------------------------------------------------------------------------------
------------------
gru_20 (GRU)                    (None, 512)          1575936     gru_19[0][0]

--------------------------------------------------------------------------------
------------------
dense_20 (Dense)                (None, 2679)         1374327     gru_20[0][0]

--------------------------------------------------------------------------------
------------------
activation_2 (Activation)       (None, 2679)         0           dense_20[0][0]
================================================================================
==================
Total params: 8,170,999
```

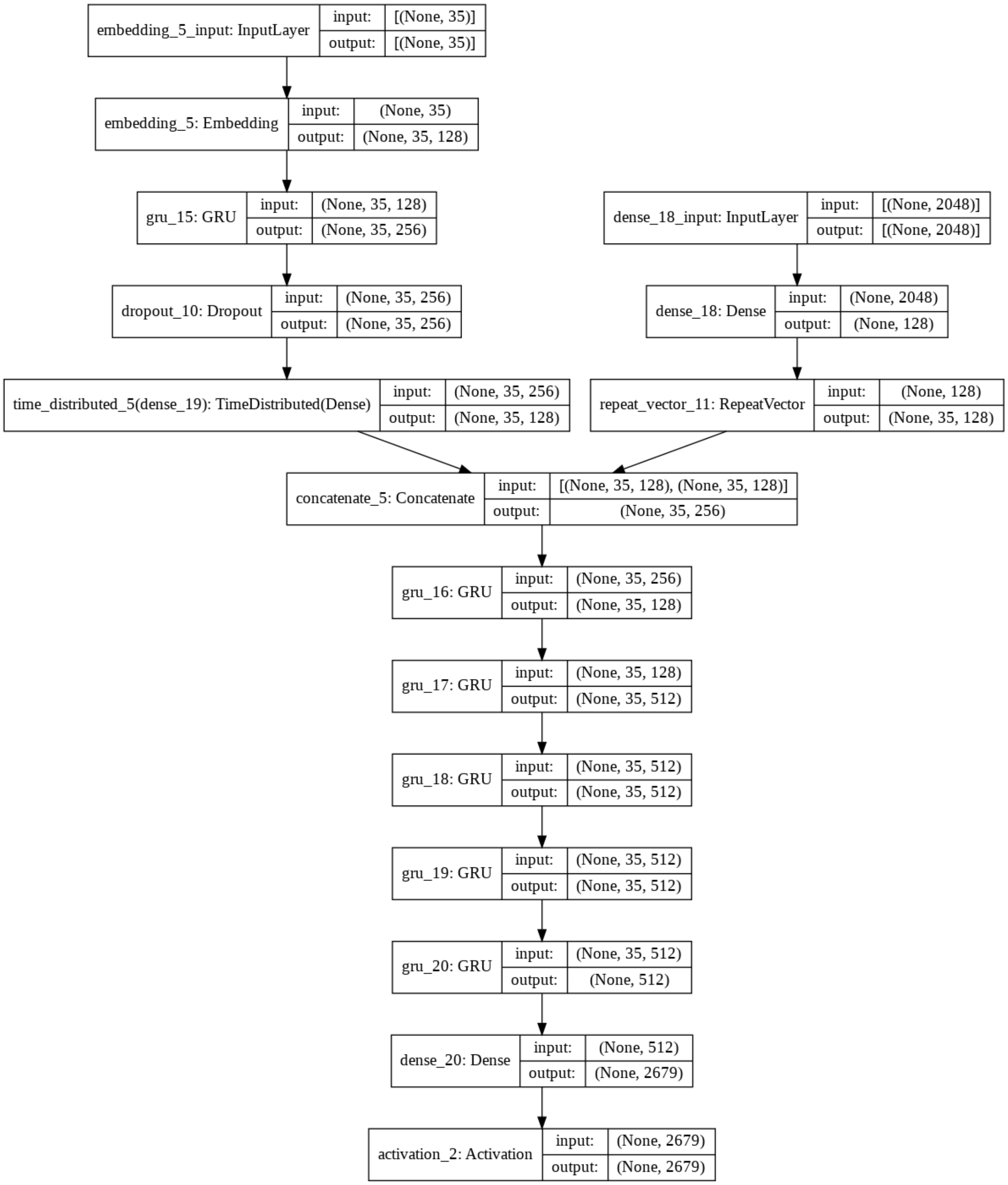
```
Trainable params: 8,170,999
Non-trainable params: 0
--------------------------------------------------------------------------------
------------------
```

0.6 5. Model Training

[117]:

```
# from keras.utils import plot_model
plot_model(model,to_file='model.png', show_shapes=True)
```

[117]:


embedding_5_input: InputLayer | input: [(None, 35)] | output: [(None, 35)]
embedding_5: Embedding | input: (None, 35) | output: (None, 35, 128)
gru_15: GRU | input: (None, 35, 128) | output: (None, 35, 256)
dense_18_input: InputLayer | input: [(None, 2048)] | output: [(None, 2048)]
dropout_10: Dropout | input: (None, 35, 256) | output: (None, 35, 256)
dense_18: Dense | input: (None, 2048) | output: (None, 128)
time_distributed_5(dense_19): TimeDistributed(Dense) | input: (None, 35, 256) | output: (None, 35, 128)
repeat_vector_11: RepeatVector | input: (None, 128) | output: (None, 35, 128)
concatenate_5: Concatenate | input: [(None, 35, 128), (None, 35, 128)] | output: (None, 35, 256)
gru_16: GRU | input: (None, 35, 256) | output: (None, 35, 128)
gru_17: GRU | input: (None, 35, 128) | output: (None, 35, 512)
gru_18: GRU | input: (None, 35, 512) | output: (None, 35, 512)
gru_19: GRU | input: (None, 35, 512) | output: (None, 35, 512)
gru_20: GRU | input: (None, 35, 512) | output: (None, 512)
dense_20: Dense | input: (None, 512) | output: (None, 2679)
activation_2: Activation | input: (None, 2679) | output: (None, 2679)

[120]:

```
class TimeHistory(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.times = []

    def on_epoch_begin(self, epoch, logs={}):
        self.epoch_time_start = time.time()

    def on_epoch_end(self, epoch, logs={}):
        self.times.append(time.time() - self.epoch_time_start)
```

[121]:

```
## Training the model

total_timetaken = []
time_callback = TimeHistory()

history = model.fit([X, y_in], y_out, batch_size=256, epochs=5,␣
 ↪callbacks=[time_callback])
total_timetaken.append(time_callback.times)
print('Training Done')
model_1_history = history
```

```
Epoch 1/5
238/238 [==============================] - 1845s 8s/step - loss: 5.3231 -
accuracy: 0.1470
Epoch 2/5
238/238 [==============================] - 1835s 8s/step - loss: 5.2621 -
accuracy: 0.1475
Epoch 3/5
238/238 [==============================] - 1832s 8s/step - loss: 5.2653 -
accuracy: 0.1475
Epoch 4/5
238/238 [==============================] - 1835s 8s/step - loss: 5.2655 -
accuracy: 0.1482
Epoch 5/5
238/238 [==============================] - 1851s 8s/step - loss: 5.2649 -
accuracy: 0.1467
Training Done
```

[122]:

```
## Creating the Inverse dictionary for works and it's corresponding numbers.
## This is used to retrieve the word against the number predicted based on␣
 ↪Probabilistic model

inv_dict = {v:k for k, v in new_dict.items()}
```

[123]:

```
## Saving the inverse dictionary for future reference
from pickle import dump
```

```
dump(inv_dict,open('inv_dict1500.p','wb'))
```

[124]:

```
## Saving the model.h5 file for offline usage
model.save('trainedmodel1500.h5')
```

[125]:

```
## Saving the model's weights
model.save_weights('mine_model_weights.h5')
```

[126]:

```
## Saving new_dict in the form of numpy dictionary
np.save('vocab.npy', new_dict)
```

[56]:

```
## Function for accessing images from the test data

def getImage(x):

    test_img_path = list_of_images[x]
    test_img = cv2.imread(test_img_path)
    test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)
    test_img = cv2.resize(test_img, (224,224))
    test_img = np.reshape(test_img, (1,224,224,3))

    return test_img
```

[134]:

```
history.history.keys()
```

[134]:

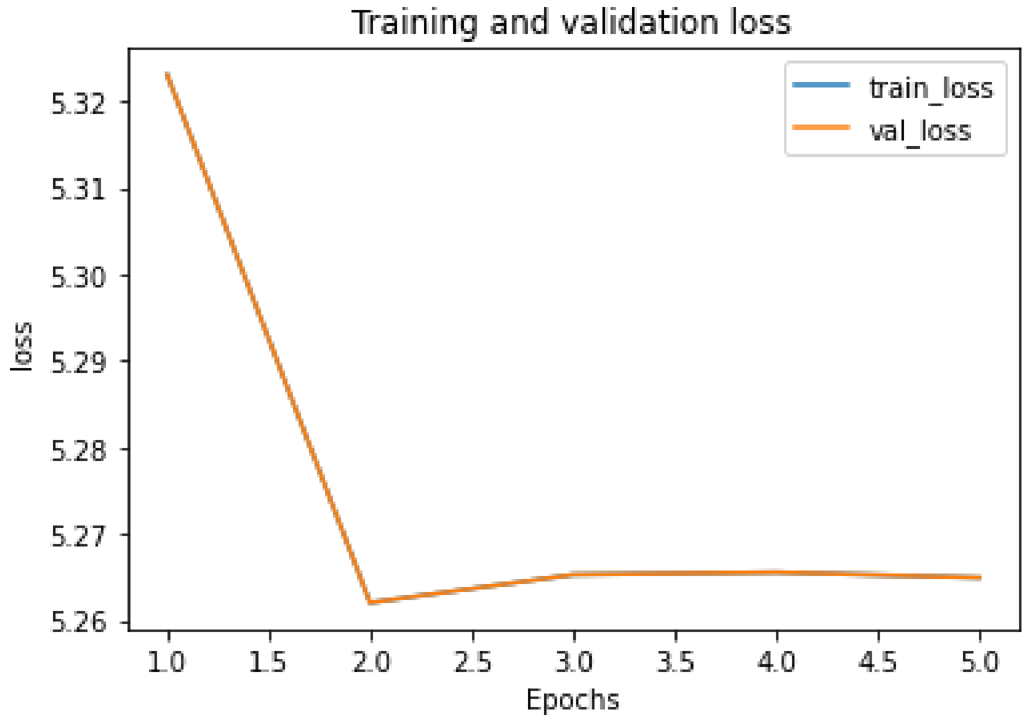
```
dict_keys(['loss', 'accuracy'])
```

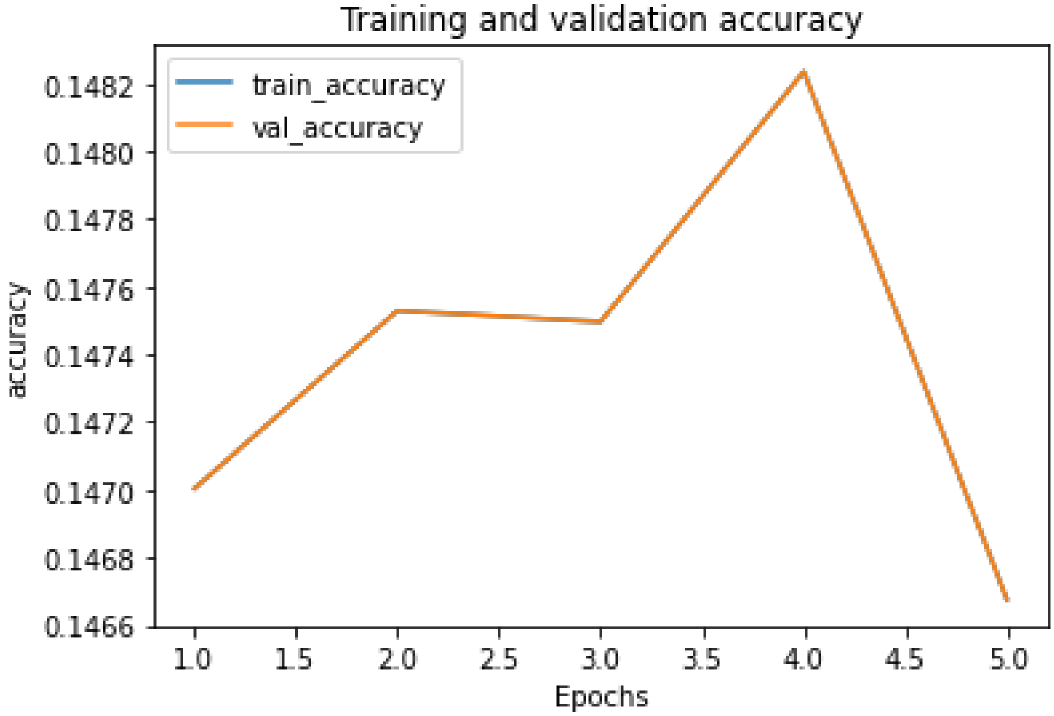
[135]:

```
##          Plotting the loss         and         accuracy history         graphs

def plot_loss_acc_graph(history):

  metrics = ['loss', 'accuracy']
  for metric in metrics:
    train_metrics = history.history[metric]
    val_metrics = history.history[metric]
    epochs = range(1, len(train_metrics) + 1)
    plt.plot(epochs, train_metrics)
    plt.plot(epochs, val_metrics)
    plt.title('Training and validation '+ metric)
    plt.xlabel("Epochs")
    plt.ylabel(metric)
    plt.legend(["train_"+metric, 'val_'+metric])
    plt.show()
```

[136]:

```
plot_loss_acc_graph(model_1_history)
print("Total Time taken for Training 'model_1' model : ",␣
 ↪sum(total_timetaken[0]))
```


Training and validation loss
train_loss
val_loss
loss
5.32
5.31
5.30
5.29
5.28
5.27
5.26
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Epochs


Training and validation accuracy
train_accuracy
val_accuracy
accuracy
0.1482
0.1480
0.1478
0.1476
0.1474
0.1472
0.1470
0.1468
0.1466
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Epochs

```
Total Time taken for Training 'model_1' model :  9197.200782775879
```

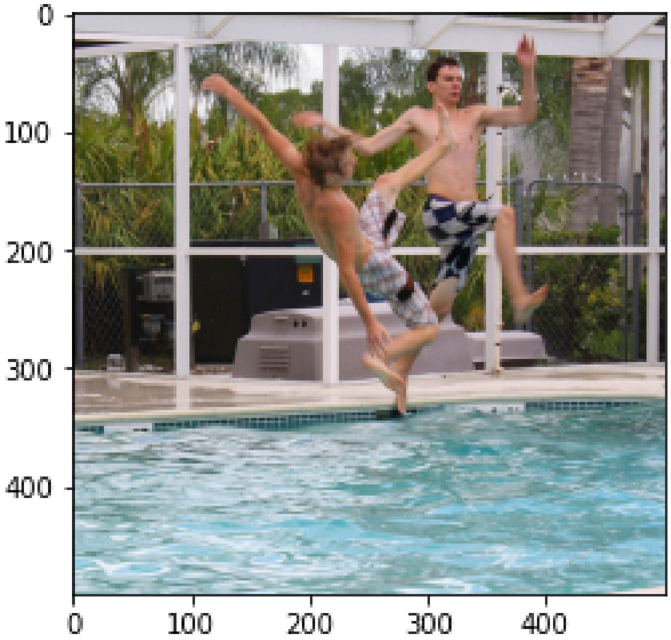
0.7 6. Model Evaluation

[60]:

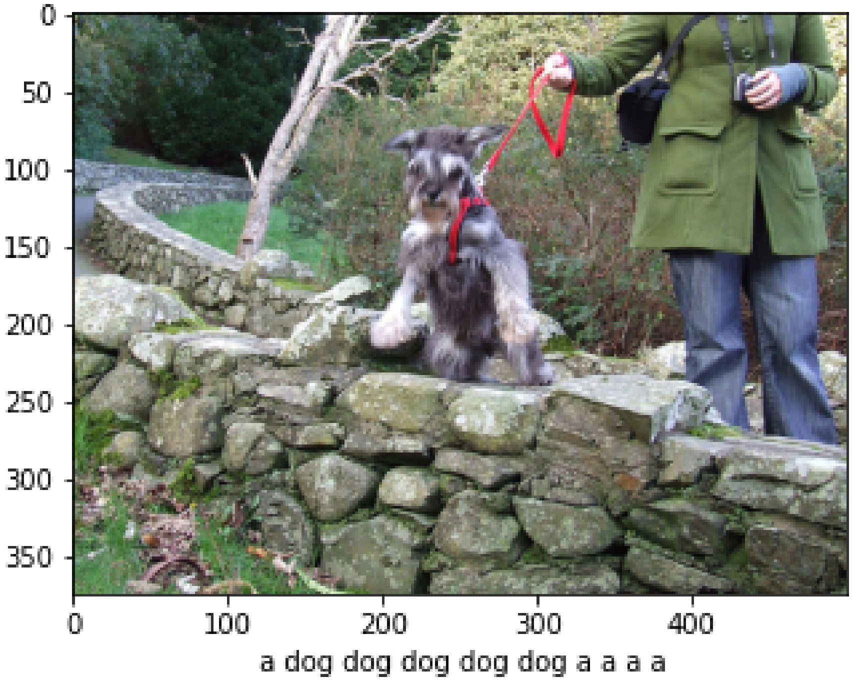
```
## Predicting captions for 5 random images in range 1500 to 6000
for i in range(5):

    no = np.random.randint(0,1500,(1,1))[0,0]
    test_feature = model_emb.predict(getImage(no)).reshape(1,2048)
    test_img_path = list_of_images[no]
    test_img = cv2.imread(test_img_path)
    test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)

    text_inp = ['startseq']
    count = 0
    caption = ''
    while count < 25:
        count += 1

        encoded = []
        for i in text_inp:
            encoded.append(new_dict[i])

        encoded = [encoded]
        encoded = pad_sequences(encoded, padding='post', truncating='post',␣
 ↪maxlen=MAX_LEN)
        prediction = np.argmax(model.predict([test_feature, encoded]))
        sampled_word = inv_dict[prediction]

        if sampled_word == 'endseq':
            break
        caption = caption + ' ' + sampled_word

        text_inp.append(sampled_word)

    plt.figure()
    plt.imshow(test_img)
    plt.xlabel(caption)
```

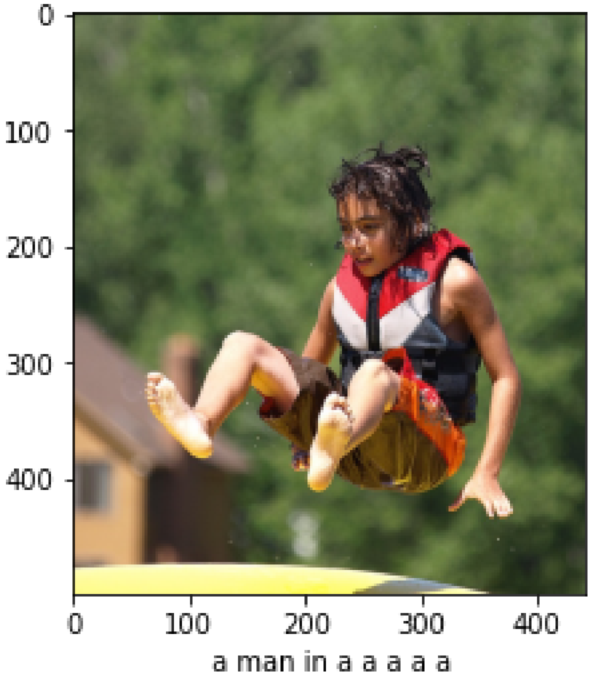
```
Predicting captions for 5 random images in range 1500 to 6000
```

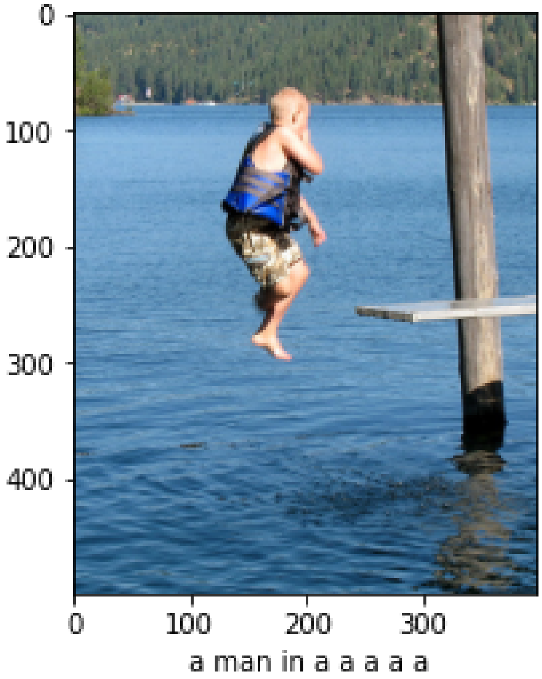
a man in a a a a



a man in a a a a a


0
50
100
150
200
250
300
350
0
100
200
300
400
a dog dog dog dog dog a a a a


0
100
200
300
400
0
100
200
300
400
a man in a a a a a


0
100
200
300
400
0
100
200
300
a man in a a a a a

```
[ ]:
```