```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.
import kagglehub
adityajn105_flickr8k_path = kagglehub.dataset_download('adityajn105/flickr8k')

print('Data source import complete.')
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/adityajn105/flickr8k?dataset_version_number=1...
100%|██████████| 1.04G/1.04G [00:17<00:00, 62.0MB/s]Extracting files...

Data source import complete.
```

Captioning Model - Using MobileNetV3Large and Bidirectional LSTM

> **Definition of the problem :**

In a variety of sectors, including media, e-commerce, and healthcare, it is becoming more and more important to comprehend the content of images. Assume you manage a sizable e-commerce platform and wish to enhance the searchability and accessibility of your products by having photos automatically generate descriptions rather than requiring you to write captions by hand. This would enable effective catalog management by saving time and lowering human error. For instance, your clients might use photographs to locate particular products, but it's difficult to tell the details without captions.

> **The solution :**

We can create an AI program that can automatically provide insightful captions for photos using computer vision and deep learning models. Accurately labeling and describing photographs will enable improved search capabilities, enhance user experience, and lower operating expenses. For example, by training a model such as MobileNetV3Large, the algorithm may provide captions for product photos, such as "A red sleeveless dress" or "Black leather sneakers," making product filtering and search on the site simple.

Importing necessary libraries

```
import os
import random
import pickle
import numpy as np
from tqdm.notebook import tqdm

import warnings
warnings.filterwarnings('ignore')

from PIL import Image
import matplotlib.pyplot as plt
from wordcloud import WordCloud

import nltk
import tensorflow as tf
from nltk.corpus import stopwords
from tensorflow.keras.models import Model,load_model
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.applications import MobileNetV3Large
from tensorflow.keras.utils import to_categorical, plot_model
```

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.applications.mobilenet_v3  import preprocess_input
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add,Bidirectional
from tensorflow.keras.layers import BatchNormalization, RepeatVector, Dot, Activation,Concatenate
from tensorflow.keras.optimizers import Adam


pip install gtts
```

```
Collecting gtts
  Downloading gTTS-2.5.3-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: requests<3,>=2.27 in /opt/conda/lib/python3.10/site-packages (from gtts) (2.32.3)
Requirement already satisfied: click<8.2,>=7.1 in /opt/conda/lib/python3.10/site-packages (from gtts) (8.1.7)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.10/site-packages (from requests<3,
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from requests<3,>=2.27->gtts
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests<3,>=2.27
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages (from requests<3,>=2.27
Downloading gTTS-2.5.3-py3-none-any.whl (29 kB)
Installing collected packages: gtts
Successfully installed gtts-2.5.3
Note: you may need to restart the kernel to use updated packages.
```

```
from gtts import gTTS
from IPython.display import Audio, display
```

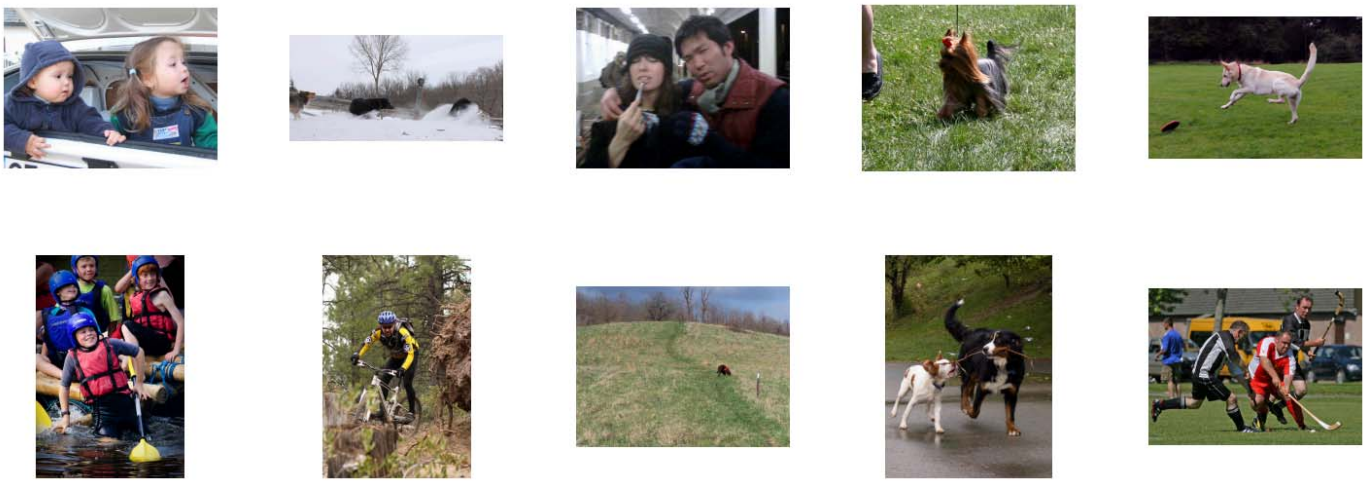
Download the Dataset

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("adityajn105/flickr8k")

print("Path to dataset files:", path)
```

```
Path to dataset files: /kaggle/input/flickr8k
```

```
BASE_DIR = '/kaggle/input/flickr8k'
WORKING_DIR = '/kaggle/working'


IMAGES_DIR = os.path.join(BASE_DIR, 'Images')
# List of image filenames
image_filenames = os.listdir(IMAGES_DIR)

print("Number of images in the dataset:",len(image_filenames))

# Display 10 images
import random
random.seed(42)
image_filenames = random.sample(image_filenames, 10)
fig, axes = plt.subplots(2, 5, figsize=(15, 6)) #10 image in 2 rows and 5 columns
fig.tight_layout(pad=2.0)

for i, ax in enumerate(axes.flat):
    # Load the image
    img_path = os.path.join(IMAGES_DIR, image_filenames[i])
    img = Image.open(img_path)
    # Display the image
    ax.imshow(img)
    ax.axis('off')
plt.show()
```

```
Number of images in the dataset: 8091
```



```
def print_image_shapes(image_dir, image_filenames):
    random.seed(42)
    image_filenames = random.sample(image_filenames, 10)
    for img_name in image_filenames[:10]:
        img_path = os.path.join(image_dir, img_name)
        with Image.open(img_path) as img:
            print(f"Image ID: {img_name}, Shape: {img.size}")

# Call the function
print_image_shapes(IMAGES_DIR, image_filenames)
```

```
Image ID: 3321063116_4e5deeac83.jpg, Shape: (500, 247)
Image ID: 1417637704_572b4d6557.jpg, Shape: (500, 375)
Image ID: 2860400846_2c1026a573.jpg, Shape: (500, 333)
Image ID: 3710971182_cb01c97d15.jpg, Shape: (500, 368)
Image ID: 3474985008_0a827cd340.jpg, Shape: (333, 500)
Image ID: 3003011417_79b49ff384.jpg, Shape: (333, 500)
Image ID: 1688699579_2f72328c7e.jpg, Shape: (375, 500)
Image ID: 3134586018_ae03ba20a0.jpg, Shape: (500, 376)
Image ID: 763577068_4b96ed768b.jpg, Shape: (500, 388)
Image ID: 448590900_db83c42006.jpg, Shape: (500, 375)
```

```
with open(os.path.join(BASE_DIR, 'captions.txt'), 'r') as f:
    next(f)
    # read all captions file
    captions_doc = f.read()

# To check the first 1000 characters of the captions file
print(captions_doc[:1000-35])
```

```
1000268201_693b08cb0e.jpg,A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg,A girl going into a wooden building .
1000268201_693b08cb0e.jpg,A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg,A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg,A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg,A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg,A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg,A black dog and a white dog with brown spots are staring at each other in the street .
1001773457_577c3a7d70.jpg,Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg,Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg,A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl
```

Pre-processing the captions

```
# create mapping of image to captions

mapping = {}
for line in tqdm(captions_doc.split('\n')):
    # split the line by comma
    tokens = line.split(',')
    if len(line) < 2:
        continue
    image_id, caption = tokens[0], tokens[1:]
    # remove extension from image ID
    image_id = image_id.split('.')[0]
    # convert caption list to string
    caption = " ".join(caption)
    # create list if image ID will added to the first time
    if image_id not in mapping:
        mapping[image_id] = []
    # store the caption if the image ID already existing in the mapping dict
    mapping[image_id].append(caption)
```

```
# after preparing the captions , we will display a sample of the data with captions

def display_images_with_mapping(directory, mapping, num_images=3):

    img_names = os.listdir(directory)[:num_images]

    fig, axes = plt.subplots(num_images, 1, figsize=(10, num_images * 3))

    for i, img_name in enumerate(img_names):
        # Extract image ID by removing file extension
        image_id = img_name.split('.')[0]
        # Ensure the image ID is in the mapping dictionary
        if image_id in mapping:
            # Load the image
            img_path = os.path.join(directory, img_name)
            img = plt.imread(img_path)
        # Plot the image in the corresponding row
            axes[i].imshow(img)
            axes[i].axis('off')
            # Get the corresponding captions from the mapping
            captions = mapping[image_id][:5]
            caption_text = "\n".join(captions)
            axes[i].set_title(caption_text, fontsize=12)

    # Adjust layout
    plt.tight_layout()
    plt.show()

# Directory containing images
image_directory = os.path.join(BASE_DIR, 'Images')
# Call the function to display images with captions
display_images_with_mapping(image_directory, mapping)
```

A dog in a snowy area .
A spotted dog catching a ball on a snowy field .
A white and brown spotted dog runs along the snow to catch a ball .
A white dog is running fast on a trail covered by snow .
The brown and white dog is playing in the snow .

A black and white dog is playing with a ball on a lawn .
A black and white dog jumps up as a ball is thrown over his head .
dog jumping looking up and small white ball in air passed by
The black and white dog is attempting to catch a ball in the garden .
The dog jumps up waiting to catch something being thrown .

A child in denim playing hopscotch .
Closeup of a pair of feet in mid jump while playing hopscotch .
Someone is playing hopscotch on a chalked out grid on the ground .
The kid is playing hopscotch .
The legs of a child playing hopscotch .

```
def clean_captions(mapping):
    for key, captions in mapping.items():
        for i in range(len(captions)):
            # take one caption at a time
            caption = captions[i]
            # convert to lowercase
            caption = caption.lower()
            # delete digits, special chars, etc
            caption = caption.replace('[^A-Za-z]', '')
            # delete additional spaces
            caption = caption.replace('\s+', ' ')
            # add start and end tags to the caption ---> important in text generation
            caption = 'startseq ' + " ".join([word for word in caption.split() if len(word)>1]) + ' endseq'
            captions[i] = caption
```

The reason behind startseq and endseq is,

startseq : Will act as our first word when feature extracted image vector is fed to decoder. It will kick-start the caption generation process.

enseq : This will tell the decoder when to stop. We will stop predicting word as soon as endseq appears or we have predicted all words from train dictionary whichever comes first.

```
# before cleaning the captions
mapping['1000268201_693b08cb0e']
```

```
['A child in a pink dress is climbing up a set of stairs in an entry way .',
 'A girl going into a wooden building .',
 'A little girl climbing into a wooden playhouse .',
 'A little girl climbing the stairs to her playhouse .',
 'A little girl in a pink dress going into a wooden cabin .']
```

```
# after cleaning the captions
clean_captions(mapping)

mapping['1000268201_693b08cb0e']
```

```
['startseq child in pink dress is climbing up set of stairs in an entry way endseq',
 'startseq girl going into wooden building endseq',
 'startseq little girl climbing into wooden playhouse endseq',
 'startseq little girl climbing the stairs to her playhouse endseq',
 'startseq little girl in pink dress going into wooden cabin endseq']
```

```
all_captions = []

for key in mapping:
    for caption in mapping[key]:
        all_captions.append(caption)

all_captions[:10]
```

```
['startseq child in pink dress is climbing up set of stairs in an entry way endseq',
 'startseq girl going into wooden building endseq',
 'startseq little girl climbing into wooden playhouse endseq',
 'startseq little girl climbing the stairs to her playhouse endseq',
 'startseq little girl in pink dress going into wooden cabin endseq',
 'startseq black dog and spotted dog are fighting endseq',
 'startseq black dog and tri-colored dog playing with each other on the road endseq',
 'startseq black dog and white dog with brown spots are staring at each other in the street endseq',
 'startseq two dogs of different breeds looking at each other on the road endseq',
 'startseq two dogs on pavement moving toward each other endseq']
```

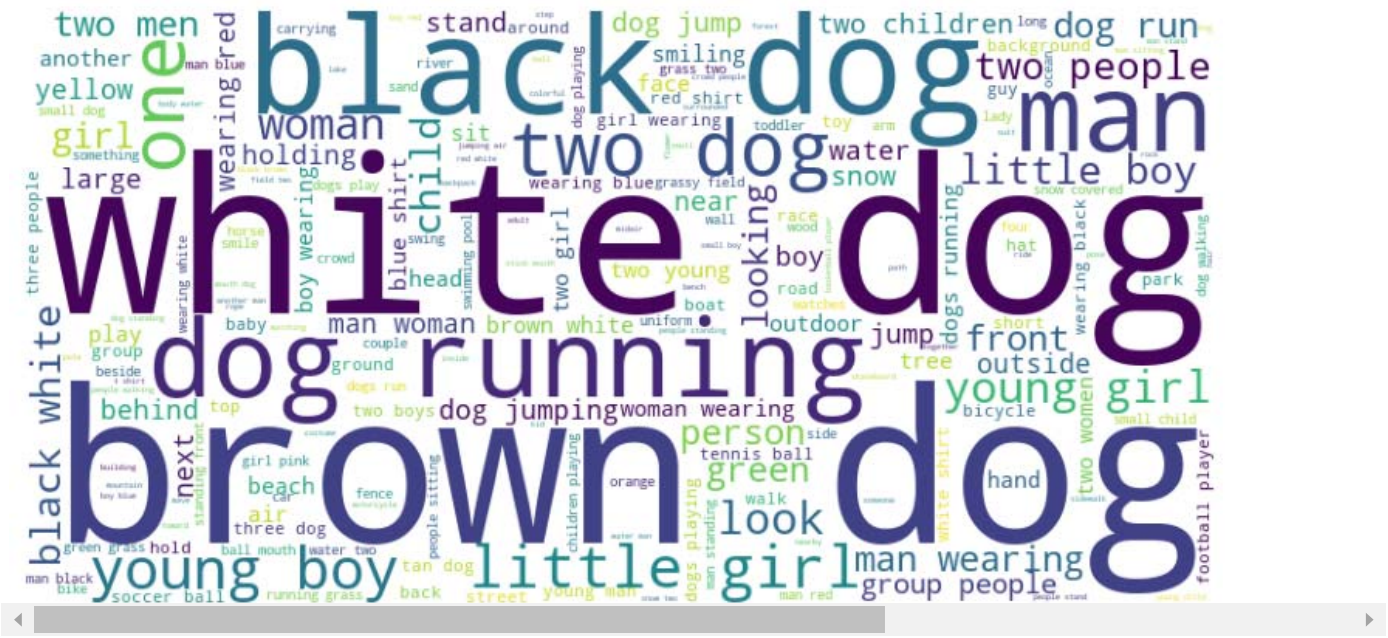
```
len(all_captions)
```

```
40455
```

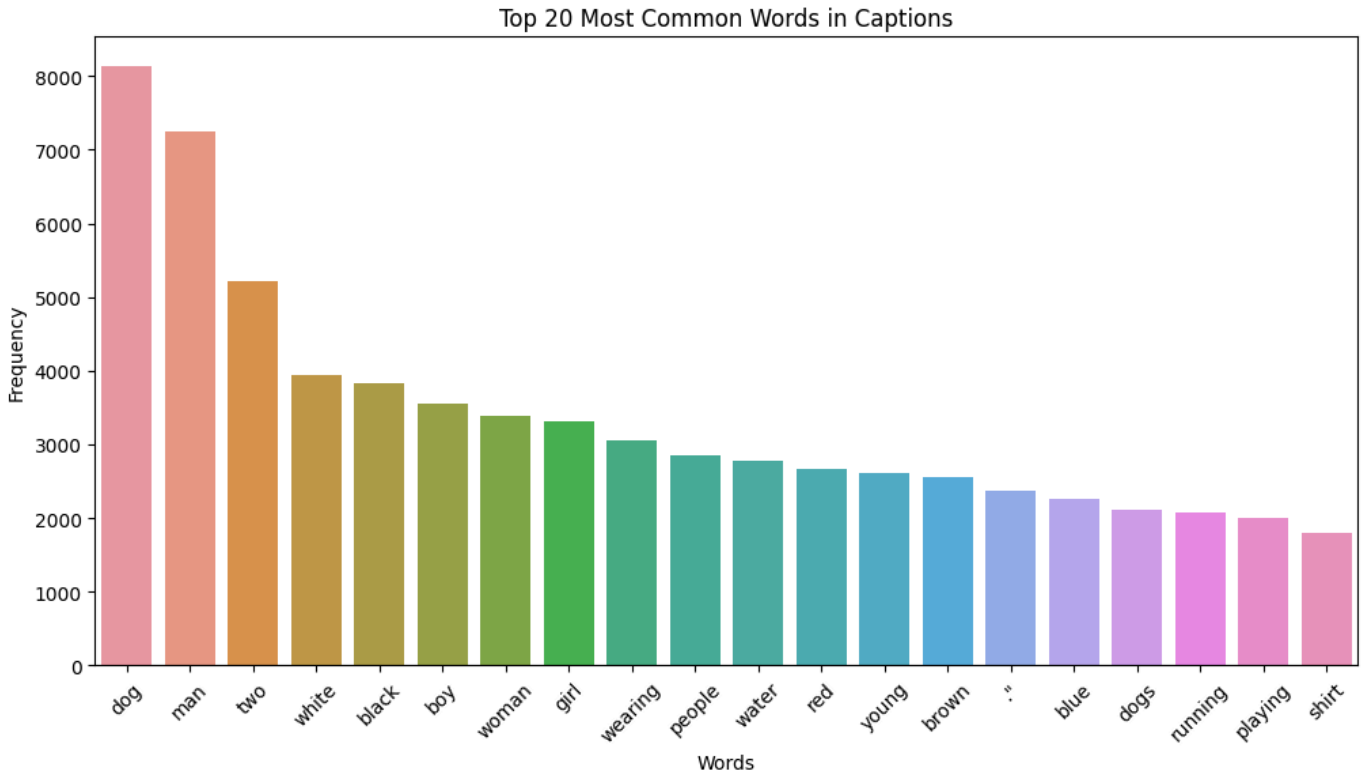
```
# to know the most frequant words in your dataset

nltk.download('stopwords')
# Remove 'startseq' and 'endseq' from captions
filtered_captions = [caption.replace('startseq', '').replace('endseq', '') for caption in all_captions]

# Combine all captions into one string
all_text = " ".join(filtered_captions)

# Remove stopwords
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in all_text.split() if word not in stop_words]
```

```
# Create a word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(" ".join(filtered_words))

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```




```
# Analyze the frequency of words in captions
from collections import Counter
import seaborn as sns

word_counts = Counter(filtered_words)
common_words = word_counts.most_common(20)

# Create a bar plot for the most common words
words, counts = zip(*common_words)
plt.figure(figsize=(12, 6))
sns.barplot(x=list(words), y=list(counts))
plt.title('Top 20 Most Common Words in Captions')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```




```
# Tokenize the captions to convert them into sequences of integers
tokenizer = Tokenizer()
tokenizer.fit_on_texts(all_captions)
vocab_size = len(tokenizer.word_index) + 1
vocab_size
```

```
8485
```

```
tokenizer.index_word
```

```
{1: 'startseq',
 2: 'endseq',
 3: 'in',
 4: 'the',
 5: 'on',
 6: 'is',
 7: 'and',
 8: 'dog',
 9: 'with',
 10: 'man',
 11: 'of',
 12: 'two',
 13: 'white',
 14: 'black',
 15: 'boy',
 16: 'are',
 17: 'woman',
 18: 'girl',
 19: 'to',
 20: 'wearing',
 21: 'at',
 22: 'people',
 23: 'water',
 24: 'red',
 25: 'young',
```

```
        26: 'brown',
        27: 'an',
        28: 'his',
        29: 'blue',
        30: 'dogs',
        31: 'running',
        32: 'through',
        33: 'playing',
        34: 'while',
        35: 'shirt',
        36: 'down',
        37: 'standing',
        38: 'ball',
        39: 'little',
        40: 'grass',
        41: 'snow',
        42: 'child',
        43: 'person',
        44: 'jumping',
        45: 'over',
        46: 'three',
        47: 'front',
        48: 'sitting',
        49: 'holding',
        50: 'up',
        51: 'field',
        52: 'small',
        53: 'by',
        54: 'a',
        55: 'large',
        56: 'green',
        57: 'one',
```

```
print(all_captions[5])
print(tokenizer.texts_to_sequences([all_captions[5]])[0])
```

```
startseq black dog and spotted dog are fighting endseq
[1, 14, 8, 7, 843, 8, 16, 343, 2]
```

```
with open(os.path.join(WORKING_DIR, 'tokenizer.pkl'), 'wb') as f:
    pickle.dump(tokenizer, f)

max_length = max(len(caption.split()) for caption in all_captions)
max_length
```

```
35
```

```
# prepare a list of image IDs
image_ids = list(mapping.keys())

image_ids[:5]
```

```
['1000268201_693b08cb0e',
 '1001773457_577c3a7d70',
 '1002674143_1b742ab4b8',
 '1003163366_44323f5815',
 '1007129816_e794419615']
```

Load the Model and Extract features

To extract useful characteristics from photos, we load the MobileNetV3Large model in this phase. The model has already been pre-trained on the ImageNet dataset. We alter the model to obtain the second-to-last layer rather than using it for classification. The main visual information of the image is represented by a feature vector that this layer provides; that is, we wish to extract the image's characteristics only for the purpose of using them to predict the image's captions.

Feature Extraction: The captioning model uses the extracted feature vectors as input, giving it a condensed representation of the image's content. These attributes are crucial for producing meaningful captions because they capture significant elements of the image, such as objects, colors, and forms.

Preprocessing steps:

- Every picture in the dataset should be loaded.
- To guarantee compatibility with the model, resize and preprocess it.
- Extrapolate significant visual characteristics from MobileNetV3Large's.
- For further usage in the image captioning model, save the extracted feature vectors.

```
# load MobileNetV3Large model
model = MobileNetV3Large(weights='imagenet', include_top=True)
model = Model(inputs=model.inputs, outputs=model.layers[-2].output)

# summarize the model for checking the last layer
model.summary()
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v3/weights_mobilenet_v3
22661472/22661472 ━━━━━━━━━━━━━━━━━━━━ **0s** 0us/step
Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, None, None, 3)	0	-
rescaling (Rescaling)	(None, None, None, 3)	0	input_layer[0][0]
conv (Conv2D)	(None, None, None, 16)	432	rescaling[0][0]
conv_bn (BatchNormalizatio…	(None, None, None, 16)	64	conv[0][0]
activation (Activation)	(None, None, None, 16)	0	conv_bn[0][0]
expanded_conv_dept… (DepthwiseConv2D)	(None, None, None, 16)	144	activation[0][0]
expanded_conv_dept… (BatchNormalizatio…	(None, None, None, 16)	64	expanded_conv_de…
re_lu (ReLU)	(None, None, None, 16)	0	expanded_conv_de…
expanded_conv_proj… (Conv2D)	(None, None, None, 16)	256	re_lu[0][0]
expanded_conv_proj… (BatchNormalizatio…	(None, None, None, 16)	64	expanded_conv_pr…
expanded_conv_add (Add)	(None, None, None, 16)	0	activation[0][0], expanded_conv_pr…
expanded_conv_1_ex… (Conv2D)	(None, None, None, 64)	1,024	expanded_conv_ad…
expanded_conv_1_ex… (BatchNormalizatio…	(None, None, None, 64)	256	expanded_conv_1_…
re_lu_1 (ReLU)	(None, None, None, 64)	0	expanded_conv_1_…
expanded_conv_1_de… (ZeroPadding2D)	(None, None, None, 64)	0	re_lu_1[0][0]
expanded_conv_1_de… (DepthwiseConv2D)	(None, None, None, 64)	576	expanded_conv_1_…
expanded_conv_1_de… (BatchNormalizatio…	(None, None, None, 64)	256	expanded_conv_1_…
re_lu_2 (ReLU)	(None, None, None, 64)	0	expanded_conv_1_…
expanded_conv_1_pr… (Conv2D)	(None, None, None, 24)	1,536	re_lu_2[0][0]
expanded_conv_1_pr… (BatchNormalizatio…	(None, None, None, 24)	96	expanded_conv_1_…
expanded_conv_2_ex… (Conv2D)	(None, None, None, 72)	1,728	expanded_conv_1_…
expanded_conv_2_ex… (BatchNormalizatio…	(None, None, None, 72)	288	expanded_conv_2_…
re_lu_3 (ReLU)	(None, None, None, 72)	0	expanded_conv_2_…

expanded_conv_2_de… (DepthwiseConv2D)	(None, None, None, 72)	648	re_lu_3[0][0]
expanded_conv_2_de… (BatchNormalizatio…	(None, None, None, 72)	288	expanded_conv_2_…
re_lu_4 (ReLU)	(None, None, None, 72)	0	expanded_conv_2_…
expanded_conv_2_pr… (Conv2D)	(None, None, None, 24)	1,728	re_lu_4[0][0]
expanded_conv_2_pr… (BatchNormalizatio…	(None, None, None, 24)	96	expanded_conv_2_…
expanded_conv_2_add (Add)	(None, None, None, 24)	0	expanded_conv_1_… expanded_conv_2_…
expanded_conv_3_ex… (Conv2D)	(None, None, None, 72)	1,728	expanded_conv_2_…
expanded_conv_3_ex… (BatchNormalizatio…	(None, None, None, 72)	288	expanded_conv_3_…
re_lu_5 (ReLU)	(None, None, None, 72)	0	expanded_conv_3_…
expanded_conv_3_de… (ZeroPadding2D)	(None, None, None, 72)	0	re_lu_5[0][0]
expanded_conv_3_de… (DepthwiseConv2D)	(None, None, None, 72)	1,800	expanded_conv_3_…
expanded_conv_3_de… (BatchNormalizatio…	(None, None, None, 72)	288	expanded_conv_3_…
re_lu_6 (ReLU)	(None, None, None, 72)	0	expanded_conv_3_…
expanded_conv_3_sq… (GlobalAveragePool…	(None, 1, 1, 72)	0	re_lu_6[0][0]
expanded_conv_3_sq… (Conv2D)	(None, 1, 1, 24)	1,752	expanded_conv_3_…
expanded_conv_3_sq… (ReLU)	(None, 1, 1, 24)	0	expanded_conv_3_…
expanded_conv_3_sq… (Conv2D)	(None, 1, 1, 72)	1,800	expanded_conv_3_…
add (Add)	(None, 1, 1, 72)	0	expanded_conv_3_…
re_lu_7 (ReLU)	(None, 1, 1, 72)	0	add[0][0]
multiply (Multiply)	(None, 1, 1, 72)	0	re_lu_7[0][0]
expanded_conv_3_sq… (Multiply)	(None, None, None, 72)	0	re_lu_6[0][0], multiply[0][0]
expanded_conv_3_pr… (Conv2D)	(None, None, None, 40)	2,880	expanded_conv_3_…
expanded_conv_3_pr… (BatchNormalizatio…	(None, None, None, 40)	160	expanded_conv_3_…
expanded_conv_4_ex… (Conv2D)	(None, None, None, 120)	4,800	expanded_conv_3_…
expanded_conv_4_ex… (BatchNormalizatio…	(None, None, None, 120)	480	expanded_conv_4_…
re_lu_8 (ReLU)	(None, None, None, 120)	0	expanded_conv_4_…

expanded_conv_4_de… (DepthwiseConv2D)	(None, None, None, 120)	3,000	re_lu_8[0][0]
expanded_conv_4_de… (BatchNormalizatio…	(None, None, None, 120)	480	expanded_conv_4_…
re_lu_9 (ReLU)	(None, None, None, 120)	0	expanded_conv_4_…
expanded_conv_4_sq… (GlobalAveragePool…	(None, 1, 1, 120)	0	re_lu_9[0][0]
expanded_conv_4_sq… (Conv2D)	(None, 1, 1, 32)	3,872	expanded_conv_4_…
expanded_conv_4_sq… (ReLU)	(None, 1, 1, 32)	0	expanded_conv_4_…
expanded_conv_4_sq… (Conv2D)	(None, 1, 1, 120)	3,960	expanded_conv_4_…
add_1 (Add)	(None, 1, 1, 120)	0	expanded_conv_4_…
re_lu_10 (ReLU)	(None, 1, 1, 120)	0	add_1[0][0]
multiply_1 (Multiply)	(None, 1, 1, 120)	0	re_lu_10[0][0]
expanded_conv_4_sq… (Multiply)	(None, None, None, 120)	0	re_lu_9[0][0], multiply_1[0][0]
expanded_conv_4_pr… (Conv2D)	(None, None, None, 40)	4,800	expanded_conv_4_…
expanded_conv_4_pr… (BatchNormalizatio…	(None, None, None, 40)	160	expanded_conv_4_…
expanded_conv_4_add (Add)	(None, None, None, 40)	0	expanded_conv_3_… expanded_conv_4_…
expanded_conv_5_ex… (Conv2D)	(None, None, None, 120)	4,800	expanded_conv_4_…
expanded_conv_5_ex… (BatchNormalizatio…	(None, None, None, 120)	480	expanded_conv_5_…
re_lu_11 (ReLU)	(None, None, None, 120)	0	expanded_conv_5_…
expanded_conv_5_de… (DepthwiseConv2D)	(None, None, None, 120)	3,000	re_lu_11[0][0]
expanded_conv_5_de… (BatchNormalizatio…	(None, None, None, 120)	480	expanded_conv_5_…
re_lu_12 (ReLU)	(None, None, None, 120)	0	expanded_conv_5_…
expanded_conv_5_sq… (GlobalAveragePool…	(None, 1, 1, 120)	0	re_lu_12[0][0]
expanded_conv_5_sq… (Conv2D)	(None, 1, 1, 32)	3,872	expanded_conv_5_…
expanded_conv_5_sq… (ReLU)	(None, 1, 1, 32)	0	expanded_conv_5_…
expanded_conv_5_sq… (Conv2D)	(None, 1, 1, 120)	3,960	expanded_conv_5_…
add_2 (Add)	(None, 1, 1, 120)	0	expanded_conv_5_…
re_lu_13 (ReLU)	(None, 1, 1, 120)	0	add_2[0][0]

multiply_2 (Multiply)	(None, 1, 1, 120)	0	re_lu_13[0][0]
expanded_conv_5_sq… (Multiply)	(None, None, None, 120)	0	re_lu_12[0][0], multiply_2[0][0]
expanded_conv_5_pr… (Conv2D)	(None, None, None, 40)	4,800	expanded_conv_5_…
expanded_conv_5_pr… (BatchNormalizatio…	(None, None, None, 40)	160	expanded_conv_5_…
expanded_conv_5_add (Add)	(None, None, None, 40)	0	expanded_conv_4_… expanded_conv_5_…
expanded_conv_6_ex… (Conv2D)	(None, None, None, 240)	9,600	expanded_conv_5_…
expanded_conv_6_ex… (BatchNormalizatio…	(None, None, None, 240)	960	expanded_conv_6_…
activation_1 (Activation)	(None, None, None, 240)	0	expanded_conv_6_…
expanded_conv_6_de… (ZeroPadding2D)	(None, None, None, 240)	0	activation_1[0][…
expanded_conv_6_de… (DepthwiseConv2D)	(None, None, None, 240)	2,160	expanded_conv_6_…
expanded_conv_6_de… (BatchNormalizatio…	(None, None, None, 240)	960	expanded_conv_6_…
activation_2 (Activation)	(None, None, None, 240)	0	expanded_conv_6_…
expanded_conv_6_pr… (Conv2D)	(None, None, None, 80)	19,200	activation_2[0][…
expanded_conv_6_pr… (BatchNormalizatio…	(None, None, None, 80)	320	expanded_conv_6_…
expanded_conv_7_ex… (Conv2D)	(None, None, None, 200)	16,000	expanded_conv_6_…
expanded_conv_7_ex… (BatchNormalizatio…	(None, None, None, 200)	800	expanded_conv_7_…
activation_3 (Activation)	(None, None, None, 200)	0	expanded_conv_7_…
expanded_conv_7_de… (DepthwiseConv2D)	(None, None, None, 200)	1,800	activation_3[0][…
expanded_conv_7_de… (BatchNormalizatio…	(None, None, None, 200)	800	expanded_conv_7_…
activation_4 (Activation)	(None, None, None, 200)	0	expanded_conv_7_…
expanded_conv_7_pr… (Conv2D)	(None, None, None, 80)	16,000	activation_4[0][…
expanded_conv_7_pr… (BatchNormalizatio…	(None, None, None, 80)	320	expanded_conv_7_…
expanded_conv_7_add (Add)	(None, None, None, 80)	0	expanded_conv_6_… expanded_conv_7_…
expanded_conv_8_ex… (Conv2D)	(None, None, None, 184)	14,720	expanded_conv_7_…
expanded_conv_8_ex… (BatchNormalizatio…	(None, None, None, 184)	736	expanded_conv_8_…

activation_5 (Activation)	(None, None, None, 184)	0	expanded_conv_8_…
expanded_conv_8_de… (DepthwiseConv2D)	(None, None, None, 184)	1,656	activation_5[0][…
expanded_conv_8_de… (BatchNormalizatio…	(None, None, None, 184)	736	expanded_conv_8_…
activation_6 (Activation)	(None, None, None, 184)	0	expanded_conv_8_…
expanded_conv_8_pr… (Conv2D)	(None, None, None, 80)	14,720	activation_6[0][…
expanded_conv_8_pr… (BatchNormalizatio…	(None, None, None, 80)	320	expanded_conv_8_…
expanded_conv_8_add (Add)	(None, None, None, 80)	0	expanded_conv_7_… expanded_conv_8_…
expanded_conv_9_ex… (Conv2D)	(None, None, None, 184)	14,720	expanded_conv_8_…
expanded_conv_9_ex… (BatchNormalizatio…	(None, None, None, 184)	736	expanded_conv_9_…
activation_7 (Activation)	(None, None, None, 184)	0	expanded_conv_9_…
expanded_conv_9_de… (DepthwiseConv2D)	(None, None, None, 184)	1,656	activation_7[0][…
expanded_conv_9_de… (BatchNormalizatio…	(None, None, None, 184)	736	expanded_conv_9_…
activation_8 (Activation)	(None, None, None, 184)	0	expanded_conv_9_…
expanded_conv_9_pr… (Conv2D)	(None, None, None, 80)	14,720	activation_8[0][…
expanded_conv_9_pr… (BatchNormalizatio…	(None, None, None, 80)	320	expanded_conv_9_…
expanded_conv_9_add (Add)	(None, None, None, 80)	0	expanded_conv_8_… expanded_conv_9_…
expanded_conv_10_e… (Conv2D)	(None, None, None, 480)	38,400	expanded_conv_9_…
expanded_conv_10_e… (BatchNormalizatio…	(None, None, None, 480)	1,920	expanded_conv_10…
activation_9 (Activation)	(None, None, None, 480)	0	expanded_conv_10…
expanded_conv_10_d… (DepthwiseConv2D)	(None, None, None, 480)	4,320	activation_9[0][…
expanded_conv_10_d… (BatchNormalizatio…	(None, None, None, 480)	1,920	expanded_conv_10…
activation_10 (Activation)	(None, None, None, 480)	0	expanded_conv_10…
expanded_conv_10_s… (GlobalAveragePool…	(None, 1, 1, 480)	0	activation_10[0]…
expanded_conv_10_s… (Conv2D)	(None, 1, 1, 120)	57,720	expanded_conv_10…
expanded_conv_10_s… (ReLU)	(None, 1, 1, 120)	0	expanded_conv_10…

(ReLU)			
expanded_conv_10_s… (Conv2D)	(None, 1, 1, 480)	58,080	expanded_conv_10…
add_3 (Add)	(None, 1, 1, 480)	0	expanded_conv_10…
re_lu_14 (ReLU)	(None, 1, 1, 480)	0	add_3[0][0]
multiply_3 (Multiply)	(None, 1, 1, 480)	0	re_lu_14[0][0]
expanded_conv_10_s… (Multiply)	(None, None, None, 480)	0	activation_10[0]… multiply_3[0][0]
expanded_conv_10_p… (Conv2D)	(None, None, None, 112)	53,760	expanded_conv_10…
expanded_conv_10_p… (BatchNormalizatio…	(None, None, None, 112)	448	expanded_conv_10…
expanded_conv_11_e… (Conv2D)	(None, None, None, 672)	75,264	expanded_conv_10…
expanded_conv_11_e… (BatchNormalizatio…	(None, None, None, 672)	2,688	expanded_conv_11…
activation_11 (Activation)	(None, None, None, 672)	0	expanded_conv_11…
expanded_conv_11_d… (DepthwiseConv2D)	(None, None, None, 672)	6,048	activation_11[0]…
expanded_conv_11_d… (BatchNormalizatio…	(None, None, None, 672)	2,688	expanded_conv_11…
activation_12 (Activation)	(None, None, None, 672)	0	expanded_conv_11…
expanded_conv_11_s… (GlobalAveragePool…	(None, 1, 1, 672)	0	activation_12[0]…
expanded_conv_11_s… (Conv2D)	(None, 1, 1, 168)	113,064	expanded_conv_11…
expanded_conv_11_s… (ReLU)	(None, 1, 1, 168)	0	expanded_conv_11…
expanded_conv_11_s… (Conv2D)	(None, 1, 1, 672)	113,568	expanded_conv_11…
add_4 (Add)	(None, 1, 1, 672)	0	expanded_conv_11…
re_lu_15 (ReLU)	(None, 1, 1, 672)	0	add_4[0][0]
multiply_4 (Multiply)	(None, 1, 1, 672)	0	re_lu_15[0][0]
expanded_conv_11_s… (Multiply)	(None, None, None, 672)	0	activation_12[0]… multiply_4[0][0]
expanded_conv_11_p… (Conv2D)	(None, None, None, 112)	75,264	expanded_conv_11…
expanded_conv_11_p… (BatchNormalizatio…	(None, None, None, 112)	448	expanded_conv_11…
expanded_conv_11_a… (Add)	(None, None, None, 112)	0	expanded_conv_10… expanded_conv_11…
expanded_conv_12_e… (Conv2D)	(None, None, None, 672)	75,264	expanded_conv_11…
expanded_conv_12_e… (BatchNormalizatio…	(None, None, None, 672)	2,688	expanded_conv_12…

activation_13 (Activation)	(None, None, None, 672)	0	expanded_conv_12…
expanded_conv_12_d… (ZeroPadding2D)	(None, None, None, 672)	0	activation_13[0]…
expanded_conv_12_d… (DepthwiseConv2D)	(None, None, None, 672)	16,800	expanded_conv_12…
expanded_conv_12_d… (BatchNormalizatio…	(None, None, None, 672)	2,688	expanded_conv_12…
activation_14 (Activation)	(None, None, None, 672)	0	expanded_conv_12…
expanded_conv_12_s… (GlobalAveragePool…	(None, 1, 1, 672)	0	activation_14[0]…
expanded_conv_12_s… (Conv2D)	(None, 1, 1, 168)	113,064	expanded_conv_12…
expanded_conv_12_s… (ReLU)	(None, 1, 1, 168)	0	expanded_conv_12…
expanded_conv_12_s… (Conv2D)	(None, 1, 1, 672)	113,568	expanded_conv_12…
add_5 (Add)	(None, 1, 1, 672)	0	expanded_conv_12…
re_lu_16 (ReLU)	(None, 1, 1, 672)	0	add_5[0][0]
multiply_5 (Multiply)	(None, 1, 1, 672)	0	re_lu_16[0][0]
expanded_conv_12_s… (Multiply)	(None, None, None, 672)	0	activation_14[0]… multiply_5[0][0]
expanded_conv_12_p… (Conv2D)	(None, None, None, 160)	107,520	expanded_conv_12…
expanded_conv_12_p… (BatchNormalizatio…	(None, None, None, 160)	640	expanded_conv_12…
expanded_conv_13_e… (Conv2D)	(None, None, None, 960)	153,600	expanded_conv_12…
expanded_conv_13_e… (BatchNormalizatio…	(None, None, None, 960)	3,840	expanded_conv_13…
activation_15 (Activation)	(None, None, None, 960)	0	expanded_conv_13…
expanded_conv_13_d… (DepthwiseConv2D)	(None, None, None, 960)	24,000	activation_15[0]…
expanded_conv_13_d… (BatchNormalizatio…	(None, None, None, 960)	3,840	expanded_conv_13…
activation_16 (Activation)	(None, None, None, 960)	0	expanded_conv_13…
expanded_conv_13_s… (GlobalAveragePool…	(None, 1, 1, 960)	0	activation_16[0]…
expanded_conv_13_s… (Conv2D)	(None, 1, 1, 240)	230,640	expanded_conv_13…
expanded_conv_13_s… (ReLU)	(None, 1, 1, 240)	0	expanded_conv_13…
expanded_conv_13_s… (Conv2D)	(None, 1, 1, 960)	231,360	expanded_conv_13…
add_6 (Add)	(None, 1, 1, 960)	0	expanded_conv_13…

re_lu_17 (ReLU)	(None, 1, 1, 960)	0	add_6[0][0]
multiply_6 (Multiply)	(None, 1, 1, 960)	0	re_lu_17[0][0]
expanded_conv_13_s… (Multiply)	(None, None, None, 960)	0	activation_16[0]… multiply_6[0][0]
expanded_conv_13_p… (Conv2D)	(None, None, None, 160)	153,600	expanded_conv_13…
expanded_conv_13_p… (BatchNormalizatio…	(None, None, None, 160)	640	expanded_conv_13…
expanded_conv_13_a… (Add)	(None, None, None, 160)	0	expanded_conv_12… expanded_conv_13…
expanded_conv_14_e… (Conv2D)	(None, None, None, 960)	153,600	expanded_conv_13…
expanded_conv_14_e… (BatchNormalizatio…	(None, None, None, 960)	3,840	expanded_conv_14…
activation_17 (Activation)	(None, None, None, 960)	0	expanded_conv_14…
expanded_conv_14_d… (DepthwiseConv2D)	(None, None, None, 960)	24,000	activation_17[0]…
expanded_conv_14_d… (BatchNormalizatio…	(None, None, None, 960)	3,840	expanded_conv_14…
activation_18 (Activation)	(None, None, None, 960)	0	expanded_conv_14…
expanded_conv_14_s… (GlobalAveragePool…	(None, 1, 1, 960)	0	activation_18[0]…
expanded_conv_14_s… (Conv2D)	(None, 1, 1, 240)	230,640	expanded_conv_14…
expanded_conv_14_s… (ReLU)	(None, 1, 1, 240)	0	expanded_conv_14…
expanded_conv_14_s… (Conv2D)	(None, 1, 1, 960)	231,360	expanded_conv_14…
add_7 (Add)	(None, 1, 1, 960)	0	expanded_conv_14…
re_lu_18 (ReLU)	(None, 1, 1, 960)	0	add_7[0][0]
multiply_7 (Multiply)	(None, 1, 1, 960)	0	re_lu_18[0][0]
expanded_conv_14_s… (Multiply)	(None, None, None, 960)	0	activation_18[0]… multiply_7[0][0]
expanded_conv_14_p…	(None, None,	153,600	expanded_conv_14…

```
print("Number of layers in MobileNetV3Large model - ",len(model.layers))
# model.layers
```

```
Number of layers in MobileNetV3Large model -  193
```

```
features = {}
directory = os.path.join(BASE_DIR, 'Images')

for img_name in tqdm(os.listdir(directory)):

    img_path = os.path.join(directory, img_name)
    # load the image and resize it to 224x224
    image = load_img(img_path, target_size=(224, 224))
    # Convert image to a numpy array
    image = img_to_array(image)

    # Expand dimensions to match the model's expected input shape (1, 224, 224, 3)
    image = np.expand_dims(image, axis=0)

    # Preprocess the image for MobileNetV3Large
    image = preprocess_input(image)

    # Extract features from the second-to-last fully connected layer
    feature = model.predict(image, verbose=0)

    # Get the image ID (filename without extension)
    image_id = img_name.split('.')[0]

    # Store the extracted features in the dictionary
    features[image_id] = feature
```

```
  0%|          | 0/8091 [00:00<?, ?it/s]
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1730616841.767664     111 service.cc:145] XLA service 0x7e14ec001e40 initialized for platform CUDA (this
I0000 00:00:1730616841.767709     111 service.cc:153]   StreamExecutor device (0): Tesla T4, Compute Capability 7.5
I0000 00:00:1730616841.767713     111 service.cc:153]   StreamExecutor device (1): Tesla T4, Compute Capability 7.5
I0000 00:00:1730616846.209683     111 device_compiler.h:188] Compiled cluster using XLA!  This line is logged at mos
```

```
# features that are extraced
temp = list(features.values())[:1]
print(temp)
```

```
[array([[-1.67093503e+00,  1.21567309e+00, -1.46605730e+00,
         -3.25958766e-02,  6.19656205e-01, -8.93524468e-01,
          8.20110857e-01,  1.43384552e+00,  8.47923398e-01,
         -1.21743572e+00,  9.41929519e-01, -9.85750377e-01,
         -9.34345365e-01, -1.27735898e-01, -1.28913462e+00,
         -9.26899850e-01, -1.77125537e+00, -3.85095239e-01,
         -1.75614405e+00, -2.06362009e-01,  2.02754736e-01,
         -1.27294016e+00, -1.27112556e+00, -7.03813910e-01,
          1.74907163e-01, -3.57186437e-01, -9.61246341e-02,
         -4.34576690e-01, -9.26136494e-01,  3.72002006e-01,
          1.23263991e+00, -1.94780886e-01, -1.02877975e+00,
          8.46320093e-02, -1.10996962e+00, -1.37878966e+00,
         -3.63099426e-01, -3.40528429e-01, -1.09874666e-01,
         -4.54239190e-01, -6.35593772e-01, -8.78795981e-01,
         -1.41274440e+00, -9.05199945e-01, -4.98752117e-01,
         -2.07583213e+00, -7.65039742e-01, -1.87640572e+00,
          1.09213424e+00, -4.41057682e-01, -7.44278133e-01,
         -2.03125983e-01, -1.55820286e+00, -1.20645535e+00,
         -4.23264861e-01,  1.21565618e-01, -1.67366576e+00,
         -7.50194252e-01, -5.99760294e-01, -7.01360345e-01,
         -3.33959985e+00, -8.55771840e-01, -1.20013642e+00,
         -2.77375293e+00, -1.62814283e+00, -2.36343241e+00,
         -1.42202154e-01, -9.42902923e-01, -2.59135753e-01,
         -5.87473325e-02, -1.16516590e+00,  7.70915329e-01,
```

```
               -1.10695386e+00,  2.65711606e-01, -1.60384141e-02,
               -6.67130232e-01, -4.89737511e-01,  5.50859928e-01,
                1.38336375e-01,  6.06464148e-01,  3.09881210e-01,
                1.18998647e+00,  8.03180695e-01, -4.22755599e-01,
                7.39964485e-01, -1.61712155e-01,  1.40070164e+00,
               -2.20526910e+00, -1.73831964e+00, -7.56731093e-01,
               -4.94187623e-01, -2.67673969e-01, -7.16326416e-01,
               -1.63308668e+00, -1.08523655e+00, -1.13233817e+00,
               -1.48368680e+00, -6.78054810e-01, -2.68361926e+00,
                1.11522198e+00, -1.24755430e+00,  4.81797636e-01,
               -1.23916817e+00, -1.25399709e+00,  8.00386488e-01,
               -1.37585986e+00, -3.58840203e+00,  9.03422356e-01,
                1.83092093e+00, -4.65156883e-01, -7.32361972e-01,
                1.26200497e-01,  1.09713066e+00,  2.82317489e-01,
                4.16582882e-01, -1.30975991e-01, -5.13838649e-01,
               -3.79627571e-02, -9.42826033e-01, -4.41743165e-01,
                1.14498758e+00,  3.63165557e-01, -1.70535278e+00,
               -3.45722049e-01,  1.31234610e+00, -8.19970429e-01,
               -1.19447708e-02,  1.12535954e+00, -1.60582483e+00,
                9.78747606e-01,  9.21175957e-01, -3.14316154e-01,
                2.12277102e+00, -3.08193803e-01, -6.98406696e-02,
               -2.24534854e-01, -1.51027250e+00, -1.61787999e+00,
               -1.01021481e+00, -3.97055000e-01, -3.93685967e-01,
                1.06600113e-02, -9.83721137e-01, -3.34558666e-01,
                1.87380099e+00,  1.36295408e-01, -2.29582638e-01,
               -2.32551765e+00, -1.43930483e+00, -2.12434220e+00,
               -1.92295837e+00,  1.89599466e+00, -3.82352918e-01,
               -7.43405461e-01,  2.94519424e-01,  7.17111111e-01,
                2.54719853e+00, -7.64353871e-01,  3.16187382e-01,
                5.61387253e+00,  7.54691601e-01,  6.18610334e+00,
                5.18578339e+00,  6.19148111e+00,  3.34430456e+00,
                2.36511874e+00,  5.65858316e+00,  6.35105181e+00,
                3.14609385e+00,  4.39192390e+00,  4.13854074e+00,
               2.92573881e+00,  7.01858997e+00,  8.66832447e+00
```

```
# store features in pickle file ! ...to use it instead of the last code of feature extraction again

pickle.dump(features, open(os.path.join(WORKING_DIR, 'features.pkl'), 'wb'))


# load features from pickle

with open(os.path.join(WORKING_DIR, 'features.pkl'), 'rb') as f:
    features = pickle.load(f)
```

Model Architecture and Training on data

To prevent memory issues, especially when working with large datasets, you can use a data generator function to generate batches of data on the fly. This way, only a small portion of the dataset is loaded into memory at a time, which helps prevent memory crashes. This is done by using the `data_generator()` function.

```
from tensorflow.keras.preprocessing.sequence import pad_sequences

def data_generator(data_keys, mapping, features, tokenizer, max_length, vocab_size, batch_size):
    X1, X2, y = list(), list(), list()
    n = 0

    while 1:
        for key in data_keys:
            n += 1
            captions = mapping[key]

            for caption in captions:
                seq = tokenizer.texts_to_sequences([caption])[0]

                for i in range(1, len(seq)):
```

```
                in_seq, out_seq = seq[:i], seq[i]

                # Ensure padding is applied properly here
                in_seq = pad_sequences([in_seq], maxlen=max_length, padding='post')[0]
                out_seq = to_categorical([out_seq], num_classes=vocab_size)[0]

                X1.append(features[key][0])
                X2.append(in_seq)
                y.append(out_seq)

            if n == batch_size:
                yield {"image": np.array(X1), "text": np.array(X2)}, np.array(y)
                X1, X2, y = list(), list(), list()
                n = 0
```

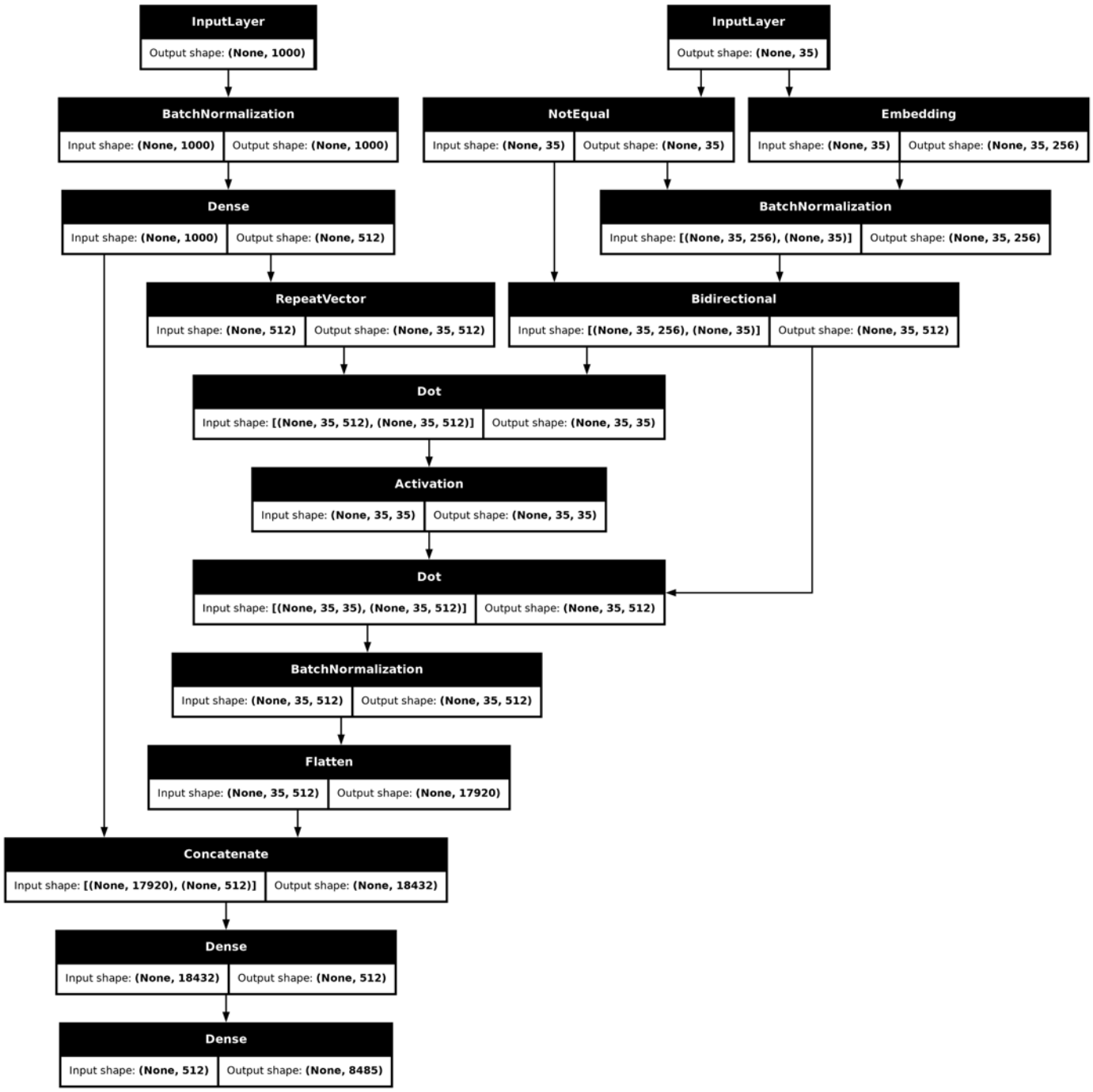
This code defines an encoder-decoder model architecture for the image captioning task, featuring two input layers:

- one for image features with a shape of (1000,), extracted from a pre-trained model, and
- another for textual captions with a shape defined by max_length.

The image features undergo Batch Normalization, followed by a Dense layer with 512 units and ReLU activation for feature transformation, and a Repeat Vector to match the sequence length of the captions. The text input is processed through an Embedding layer that converts integer sequences into dense vectors, followed by Batch Normalization and a Bidirectional LSTM layer to capture sequential dependencies in the captions. An attention mechanism computes the attention scores between the projected image features and the text features, producing a context vector that encapsulates relevant information for generating the caption. This context vector is then concatenated with the image features, processed through another Dense layer, and finally passed to a softmax output layer that predicts the next word in the caption based on the combined features.

The model is compiled using categorical cross-entropy loss and the Adam optimizer, with a learning rate of 5e-4 and gradient clipping set to 5.0, to ensure efficient training and evaluation.

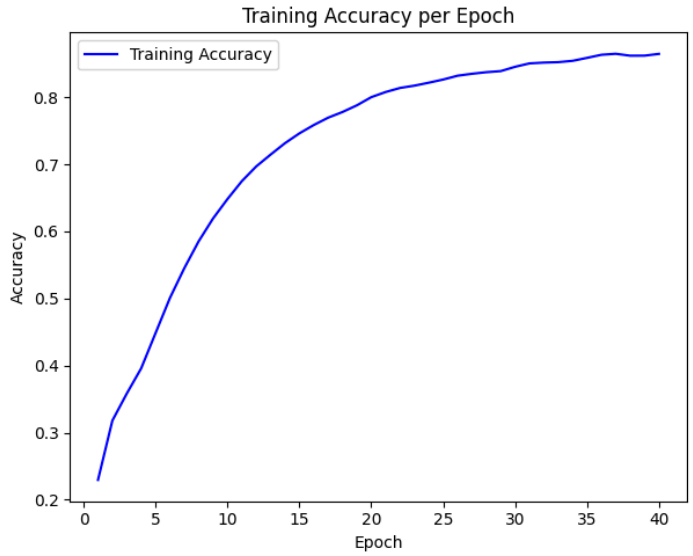
```
#Model Architecture
inputs1 = Input(shape=(1000,),name='image')
fe1 = BatchNormalization()(inputs1)
fe2 = Dense(512, activation='relu')(fe1)
fe2_projected = RepeatVector(max_length)(fe2)

inputs2 = Input(shape=(max_length,), name='text')
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = BatchNormalization()(se1)
se3 = Bidirectional(LSTM(256, return_sequences=True))(se2)

attention = Dot(axes=[2, 2])([fe2_projected, se3])
attention = Activation('softmax')(attention)
context_vector = Dot(axes=[1, 1])([attention, se3])
context_vector = BatchNormalization()(context_vector)


context_vector = tf.keras.layers.Flatten()(context_vector)
decoder1 = Concatenate()([context_vector, fe2])
decoder2 = Dense(512, activation='relu')(decoder1)
outputs = Dense(vocab_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)
optimizer = Adam(learning_rate=5e-4, clipvalue=5.0)
model.compile(loss="categorical_crossentropy", optimizer=optimizer, metrics=['accuracy'])


plot_model(model, show_shapes=True)
```

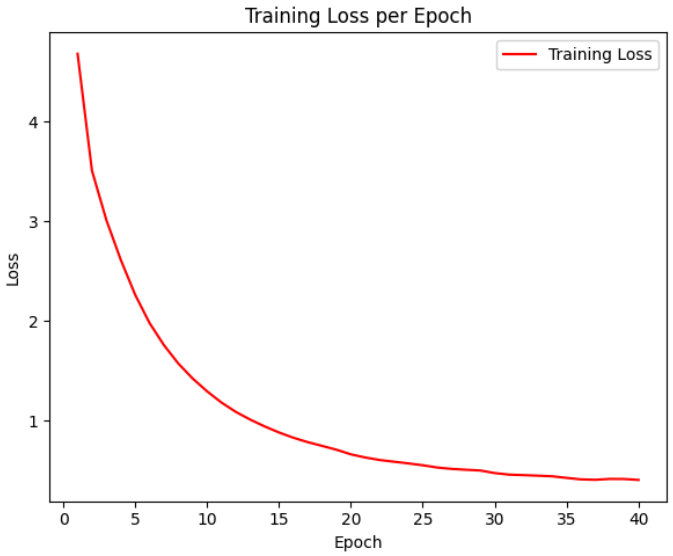



```
# Model training from the data generator
# epochs = 40
# batch_size = 32
```

```
# steps = len(image_ids) // batch_size
# for i in range(epochs):
#     generator = data_generator(image_ids, mapping, features, tokenizer, max_length, vocab_size, batch_size)
#     model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)


# Initialize lists to store accuracy and loss values
accuracy_list = []
loss_list = []

# Model training from the data generator
# Training loop
epochs = 40
batch_size = 32
steps = len(image_ids) // batch_size

for i in range(epochs):
    generator = data_generator(image_ids, mapping, features, tokenizer, max_length, vocab_size, batch_size)
    history = model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)

    # Append the accuracy and loss from this epoch
    accuracy_list.append(history.history['accuracy'][0])
    loss_list.append(history.history['loss'][0])

# Plotting accuracy and loss
plt.figure(figsize=(12, 5))

# Plot accuracy
plt.subplot(1, 2, 1)
plt.plot(range(1, epochs + 1), accuracy_list, label='Training Accuracy', color='blue')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Training Accuracy per Epoch')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(range(1, epochs + 1), loss_list, label='Training Loss', color='red')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training Loss per Epoch')
plt.legend()

plt.tight_layout()
plt.show()
```

```
252/252 ━━━━━━━━━━━━━━━━━━━━ 70s 252ms/step - accuracy: 0.1819 - loss: 5.4240
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.3089 - loss: 3.6321
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 248ms/step - accuracy: 0.3503 - loss: 3.1025
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 248ms/step - accuracy: 0.3862 - loss: 2.6930
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 248ms/step - accuracy: 0.4350 - loss: 2.3351
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 248ms/step - accuracy: 0.4883 - loss: 2.0389
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 247ms/step - accuracy: 0.5341 - loss: 1.8079
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 248ms/step - accuracy: 0.5753 - loss: 1.6185
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 248ms/step - accuracy: 0.6100 - loss: 1.4677
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.6392 - loss: 1.3365
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.6678 - loss: 1.2189
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.6921 - loss: 1.1136
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 251ms/step - accuracy: 0.7100 - loss: 1.0322
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.7254 - loss: 0.9676
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.7400 - loss: 0.9084
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 251ms/step - accuracy: 0.7530 - loss: 0.8512
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.7634 - loss: 0.8099
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.7743 - loss: 0.7619
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.7853 - loss: 0.7218
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.7992 - loss: 0.6698
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.8050 - loss: 0.6426
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.8095 - loss: 0.6217
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.8148 - loss: 0.5967
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.8200 - loss: 0.5779
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 247ms/step - accuracy: 0.8250 - loss: 0.5604
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 249ms/step - accuracy: 0.8307 - loss: 0.5382
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 251ms/step - accuracy: 0.8338 - loss: 0.5252
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 254ms/step - accuracy: 0.8369 - loss: 0.5123
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 253ms/step - accuracy: 0.8386 - loss: 0.5058
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 254ms/step - accuracy: 0.8426 - loss: 0.4862
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 255ms/step - accuracy: 0.8484 - loss: 0.4676
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 252ms/step - accuracy: 0.8513 - loss: 0.4599
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 251ms/step - accuracy: 0.8507 - loss: 0.4574
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.8531 - loss: 0.4474
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.8556 - loss: 0.4371
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 251ms/step - accuracy: 0.8613 - loss: 0.4202
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.8638 - loss: 0.4143
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 251ms/step - accuracy: 0.8624 - loss: 0.4174
252/252 ━━━━━━━━━━━━━━━━━━━━ 63s 250ms/step - accuracy: 0.8631 - loss: 0.4135
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 253ms/step - accuracy: 0.8642 - loss: 0.4100
```





The accuracy of this model is around **87%** which is comparitively better than most of the models.

```
# save the best model
model.save('best.keras')

model.save('best.h5')
```

Generating the Caption

The `idx_to_word` function converts an integer index into its corresponding word using the tokenizer. It works by scanning the tokenizer's `word_index` dictionary, which maps each word to an integer. When it finds a match between the input integer and a dictionary index, it returns the associated word. If no match is found, the function returns `None`. This function is crucial for decoding the model's predicted sequences during image captioning, translating numerical predictions into readable text.

```
def idx_to_word(integer, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == integer:
            return word
    return None
```

The `predict_caption` function generates a caption for a given image using a trained model. It begins by initializing the caption with the token `'startseq'`, indicating the beginning of the caption. The function then iteratively predicts each word up to a maximum length, `max_length`. In each iteration, the input sequence (starting with only `'startseq'`) is converted into integer tokens by the tokenizer and padded to maintain a consistent shape for the model input. Using the image features and encoded sequence, the model predicts the probability distribution for the next word. The word with the highest probability is then identified and converted back to a word via `idx_to_word`. If a valid word is generated, it is appended to the caption sequence. This process continues until an `'endseq'` token is produced or no valid word remains. The function returns the complete caption, capturing the model's learned description of the image.

```
# generate caption for an image

def predict_caption(model, image, tokenizer, max_length):
    # add start tag for generation process
    in_text = 'startseq'
    for i in range(max_length):
        # encode input sequence
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        # pad the sequence
        sequence = pad_sequences([sequence], max_length,padding='post')

        # predict next word in the sentence
        ypred = model.predict([image, sequence], verbose=0)
        # get index with high probability of the next word
        ypred = np.argmax(ypred)
        # convert index to word (look at the function above)
        word = idx_to_word(ypred, tokenizer)
        # stop if word not found
        if word is None:
            break
        # append word as input for generating next word
        in_text += " " + word
        # stop if we reach end tag
        if word == 'endseq':
            break

    return in_text
```

The `generate_caption` function takes an image file name, extracts its ID, and loads the image. It then generates a predicted caption with `predict_caption`, which includes the special tokens `startseq` and `endseq`. These tokens are removed to create a clean caption for display. Using the Google Text-to-Speech (gTTS) library, the caption text is converted to audio and saved as an MP3 file. An audio player is then shown to allow automatic playback of the caption, while the image is displayed via `matplotlib`.

```
def generate_caption(image_name):
    # Extract image ID
    image_id = image_name.split('.')[0]

    # Load the image file
    img_path = os.path.join(BASE_DIR, "Images", image_name)
    image = Image.open(img_path)

    # Generate the predicted caption
    prediction = predict_caption(model, features[image_id], tokenizer, max_length)

    # Remove the 'startseq' and 'endseq' tags from the caption for display and audio
    caption_text = prediction.replace('startseq', '').replace('endseq', '').strip()

    # Display the predicted caption
    print("Predicted Caption:", caption_text)

    # Step 1: Generate the audio for the caption using gTTS
    tts = gTTS(caption_text, lang='en',slow=False)
    audio_path = "predicted_caption.mp3"
    tts.save(audio_path)

    # Step 2: Display the audio player before the image
    print("Playing the predicted caption audio:")
    display(Audio(audio_path, autoplay=True))

    # Step 3: Display the image
    plt.axis('off')
    plt.imshow(image)
    plt.show()

    return caption_text


# choose random images to get the captions of them

images_path = os.listdir(os.path.join(BASE_DIR,'Images'))
image_name_idx = np.random.randint(0,8000,1)

for i in image_name_idx:
    cap = generate_caption(images_path[i])
```

```
Predicted Caption: two girls in pink coats drink from soda bottles and lean on tree
Playing the predicted caption audio:
```

0:00 / 0:04

```
# choose random images to get the captions of them

images_path = os.listdir(os.path.join(BASE_DIR,'Images'))
image_name_idx = np.random.randint(0,8000,1)

for i in image_name_idx:
    print(images_path[i])
    cap = generate_caption(images_path[i])
```

```
3445296377_1e5082b44b.jpg
Predicted Caption: girl in pink snowsuit splashes into dirty water
Playing the predicted caption audio:
```

0:00 / 0:03

```
from nltk.translate.bleu_score import sentence_bleu

def calculate_bleu_score(image_name, model, tokenizer, max_length):
```

```
#     def preprocess_image(image_path):
#         img = Image.open(image_path)
#         img = img.resize((224, 224))
#         img = img_to_array(img)
#         img = np.expand_dims(img, axis=0)
#         img = preprocess_input(img)
#         return img

#     def generate_caption(model, tokenizer, image, max_length):
#         in_text = 'startseq'
#         for i in range(max_length):
#             sequence = tokenizer.texts_to_sequences([in_text])[0]
#             sequence = pad_sequences([sequence], maxlen=max_length)
#             yhat = model.predict([image, sequence], verbose=0)
#             yhat = np.argmax(yhat)
#             word = tokenizer.index_word[yhat]
#             if word is None:
#                 break
#             in_text += ' ' + word
#             if word == 'endseq':
#                 break
#         final_caption = in_text.split()
#         final_caption = final_caption[1:-1]
#         final_caption = ' '.join(final_caption)
#         return final_caption

    # Generate the predicted caption
    predicted_caption = generate_caption(image_name)

    # Extract image ID by removing file extension
    image_id = image_path.split('.')[0]

    # Get the reference captions from the mapping
    reference_captions = mapping[image_id]
    print(reference_captions)

    # Tokenize the predicted caption and reference captions
    predicted_tokens = predicted_caption.split()
    reference_tokens = [ref.split() for ref in reference_captions]

    # Calculate the BLEU score
    bleu_score = sentence_bleu(reference_tokens, predicted_tokens)

    print("Predicted Caption:", predicted_caption)
    print("BLEU Score:", bleu_score)
    return bleu_score



image_path ='3445296377_1e5082b44b.jpg'
model = load_model('best.keras')
tokenizer = pickle.load(open('tokenizer.pkl', 'rb'))
max_length = 35  # Set the maximum length of the caption

calculate_bleu_score(image_path, model, tokenizer, max_length)
```

```
Predicted Caption: girl in pink snowsuit splashes into dirty water
Playing the predicted caption audio:
```

0:00 / 0:03

```
['startseq girl in pink snowsuit splashes into dirty flood water next to picnic table endseq', 'startseq little girl
Predicted Caption: girl in pink snowsuit splashes into dirty water
BLEU Score: 0.767279645960659
0.767279645960659
```

image.png

```
image_path = '479807465_cf42f39d00.jpg'
generate_caption(image_path)
```

```
Predicted Caption: two boys lifting dirt bike
Playing the predicted caption audio:
```

0:00 / 0:02

```
'two boys lifting dirt bike'
```

```
    # choose random images to get the captions of them

    images_path = os.listdir(os.path.join(BASE_DIR,'Images'))
    image_name_idx = np.random.randint(0,8000,4)

    for i in image_name_idx:
        cap = generate_caption(images_path[i])
```

```
Predicted Caption: young man is walking down some steps by bright colored flags
Playing the predicted caption audio:
```

0:00 / 0:04

```
Predicted Caption: brown dog is running
Playing the predicted caption audio:
```

0:00 / 0:01

```
Predicted Caption: two boys play with large sticks in the yard
Playing the predicted caption audio:
```

0:00 / 0:03

```
Predicted Caption: girl is playing in the water on an aqua beach
Playing the predicted caption audio:
```

0:00 / 0:03

Captioning Model 2 - Using DenseNet201 and LSTM

> Import the necessary libraries, download the dataset, and pre-process the captions. Till here follow the same steps as done for teh previous model since the dataset and preprocessing for the captions is same. So no need to repeat it.

```
import os
import pickle
import numpy as np
import pandas as pd
import seaborn as sns
import random
from tqdm.notebook import tqdm

import warnings
warnings.filterwarnings('ignore')

from PIL import Image
import matplotlib.pyplot as plt
from wordcloud import WordCloud

import nltk
import tensorflow as tf
from nltk.corpus import stopwords
from tensorflow.keras.models import Model,load_model
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.applications import MobileNetV3Large
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.applications.mobilenet_v3  import preprocess_input
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add,Bidirectional
from tensorflow.keras.layers import BatchNormalization, RepeatVector, Dot, Activation,Concatenate
from tensorflow.keras.optimizers import Adam


from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from tensorflow.keras.utils import Sequence
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, GlobalAveragePooling2D, Activation, Dropout, Flatten, Dense, I
from tensorflow.keras.layers import Embedding, LSTM, add, Concatenate, Reshape, concatenate, Bidirectional

from tensorflow.keras.applications import VGG16, ResNet50, DenseNet201
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
from textwrap import wrap
```

Feature extraction using model(DenseNet201)

```
model2 = DenseNet201()
model2 = Model(inputs=model2.input, outputs=model2.layers[-2].output)

# summarize the model
model2.summary()
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/densenet/densenet201_weights_tf_d
82524592/82524592 ━━━━━━━━━━━━━━━━━━━━ 1s 0us/step
Model: "functional_5"

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0	-
zero_padding2d (ZeroPadding2D)	(None, 230, 230, 3)	0	input_layer_1[0]…
conv1_conv (Conv2D)	(None, 112, 112, 64)	9,408	zero_padding2d[0…
conv1_bn (BatchNormalizatio…	(None, 112, 112, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None, 112, 112, 64)	0	conv1_bn[0][0]
zero_padding2d_1 (ZeroPadding2D)	(None, 114, 114, 64)	0	conv1_relu[0][0]
pool1 (MaxPooling2D)	(None, 56, 56, 64)	0	zero_padding2d_1…
conv2_block1_0_bn (BatchNormalizatio…	(None, 56, 56, 64)	256	pool1[0][0]
conv2_block1_0_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1_0_b…
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 128)	8,192	conv2_block1_0_r…
conv2_block1_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block1_1_c…
conv2_block1_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block1_1_b…
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block1_1_r…
conv2_block1_concat (Concatenate)	(None, 56, 56, 96)	0	pool1[0][0], conv2_block1_2_c…
conv2_block2_0_bn (BatchNormalizatio…	(None, 56, 56, 96)	384	conv2_block1_con…
conv2_block2_0_relu (Activation)	(None, 56, 56, 96)	0	conv2_block2_0_b…
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 128)	12,288	conv2_block2_0_r…
conv2_block2_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block2_1_c…
conv2_block2_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block2_1_b…
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block2_1_r…
conv2_block2_concat (Concatenate)	(None, 56, 56, 128)	0	conv2_block1_con… conv2_block2_2_c…
conv2_block3_0_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block2_con…
conv2_block3_0_relu (Activation)	(None, 56, 56, 128)	0	conv2_block3_0_b…

conv2_block3_1_conv (Conv2D)	(None, 56, 56, 128)	16,384	conv2_block3_0_r…
conv2_block3_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block3_1_c…
conv2_block3_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block3_1_b…
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block3_1_r…
conv2_block3_concat (Concatenate)	(None, 56, 56, 160)	0	conv2_block2_con… conv2_block3_2_c…
conv2_block4_0_bn (BatchNormalizatio…	(None, 56, 56, 160)	640	conv2_block3_con…
conv2_block4_0_relu (Activation)	(None, 56, 56, 160)	0	conv2_block4_0_b…
conv2_block4_1_conv (Conv2D)	(None, 56, 56, 128)	20,480	conv2_block4_0_r…
conv2_block4_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block4_1_c…
conv2_block4_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block4_1_b…
conv2_block4_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block4_1_r…
conv2_block4_concat (Concatenate)	(None, 56, 56, 192)	0	conv2_block3_con… conv2_block4_2_c…
conv2_block5_0_bn (BatchNormalizatio…	(None, 56, 56, 192)	768	conv2_block4_con…
conv2_block5_0_relu (Activation)	(None, 56, 56, 192)	0	conv2_block5_0_b…
conv2_block5_1_conv (Conv2D)	(None, 56, 56, 128)	24,576	conv2_block5_0_r…
conv2_block5_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block5_1_c…
conv2_block5_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block5_1_b…
conv2_block5_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block5_1_r…
conv2_block5_concat (Concatenate)	(None, 56, 56, 224)	0	conv2_block4_con… conv2_block5_2_c…
conv2_block6_0_bn (BatchNormalizatio…	(None, 56, 56, 224)	896	conv2_block5_con…
conv2_block6_0_relu (Activation)	(None, 56, 56, 224)	0	conv2_block6_0_b…
conv2_block6_1_conv (Conv2D)	(None, 56, 56, 128)	28,672	conv2_block6_0_r…
conv2_block6_1_bn (BatchNormalizatio…	(None, 56, 56, 128)	512	conv2_block6_1_c…
conv2_block6_1_relu (Activation)	(None, 56, 56, 128)	0	conv2_block6_1_b…
conv2_block6_2_conv (Conv2D)	(None, 56, 56, 32)	36,864	conv2_block6_1_r…

conv2_block6_concat (Concatenate)	(None, 56, 56, 256)	0	conv2_block5_con… conv2_block6_2_c…
pool2_bn (BatchNormalizatio…	(None, 56, 56, 256)	1,024	conv2_block6_con…
pool2_relu (Activation)	(None, 56, 56, 256)	0	pool2_bn[0][0]
pool2_conv (Conv2D)	(None, 56, 56, 128)	32,768	pool2_relu[0][0]
pool2_pool (AveragePooling2D)	(None, 28, 28, 128)	0	pool2_conv[0][0]
conv3_block1_0_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	pool2_pool[0][0]
conv3_block1_0_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_0_b…
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128)	16,384	conv3_block1_0_r…
conv3_block1_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block1_1_c…
conv3_block1_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_1_b…
conv3_block1_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block1_1_r…
conv3_block1_concat (Concatenate)	(None, 28, 28, 160)	0	pool2_pool[0][0], conv3_block1_2_c…
conv3_block2_0_bn (BatchNormalizatio…	(None, 28, 28, 160)	640	conv3_block1_con…
conv3_block2_0_relu (Activation)	(None, 28, 28, 160)	0	conv3_block2_0_b…
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	20,480	conv3_block2_0_r…
conv3_block2_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block2_1_c…
conv3_block2_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_1_b…
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block2_1_r…
conv3_block2_concat (Concatenate)	(None, 28, 28, 192)	0	conv3_block1_con… conv3_block2_2_c…
conv3_block3_0_bn (BatchNormalizatio…	(None, 28, 28, 192)	768	conv3_block2_con…
conv3_block3_0_relu (Activation)	(None, 28, 28, 192)	0	conv3_block3_0_b…
conv3_block3_1_conv (Conv2D)	(None, 28, 28, 128)	24,576	conv3_block3_0_r…
conv3_block3_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block3_1_c…
conv3_block3_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block3_1_b…
conv3_block3_2_conv	(None, 28, 28,	36,864	conv3_block3_1_r…

(Conv2D)	32)		
conv3_block3_concat (Concatenate)	(None, 28, 28, 224)	0	conv3_block2_con… conv3_block3_2_c…
conv3_block4_0_bn (BatchNormalizatio…	(None, 28, 28, 224)	896	conv3_block3_con…
conv3_block4_0_relu (Activation)	(None, 28, 28, 224)	0	conv3_block4_0_b…
conv3_block4_1_conv (Conv2D)	(None, 28, 28, 128)	28,672	conv3_block4_0_r…
conv3_block4_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block4_1_c…
conv3_block4_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block4_1_b…
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block4_1_r…
conv3_block4_concat (Concatenate)	(None, 28, 28, 256)	0	conv3_block3_con… conv3_block4_2_c…
conv3_block5_0_bn (BatchNormalizatio…	(None, 28, 28, 256)	1,024	conv3_block4_con…
conv3_block5_0_relu (Activation)	(None, 28, 28, 256)	0	conv3_block5_0_b…
conv3_block5_1_conv (Conv2D)	(None, 28, 28, 128)	32,768	conv3_block5_0_r…
conv3_block5_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block5_1_c…
conv3_block5_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block5_1_b…
conv3_block5_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block5_1_r…
conv3_block5_concat (Concatenate)	(None, 28, 28, 288)	0	conv3_block4_con… conv3_block5_2_c…
conv3_block6_0_bn (BatchNormalizatio…	(None, 28, 28, 288)	1,152	conv3_block5_con…
conv3_block6_0_relu (Activation)	(None, 28, 28, 288)	0	conv3_block6_0_b…
conv3_block6_1_conv (Conv2D)	(None, 28, 28, 128)	36,864	conv3_block6_0_r…
conv3_block6_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block6_1_c…
conv3_block6_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block6_1_b…
conv3_block6_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block6_1_r…
conv3_block6_concat (Concatenate)	(None, 28, 28, 320)	0	conv3_block5_con… conv3_block6_2_c…
conv3_block7_0_bn (BatchNormalizatio…	(None, 28, 28, 320)	1,280	conv3_block6_con…
conv3_block7_0_relu (Activation)	(None, 28, 28, 320)	0	conv3_block7_0_b…
conv3_block7_1_conv	(None, 28, 28,	40,960	conv3_block7_0_r…

(Conv2D)	128)		
conv3_block7_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block7_1_c…
conv3_block7_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block7_1_b…
conv3_block7_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block7_1_r…
conv3_block7_concat (Concatenate)	(None, 28, 28, 352)	0	conv3_block6_con… conv3_block7_2_c…
conv3_block8_0_bn (BatchNormalizatio…	(None, 28, 28, 352)	1,408	conv3_block7_con…
conv3_block8_0_relu (Activation)	(None, 28, 28, 352)	0	conv3_block8_0_b…
conv3_block8_1_conv (Conv2D)	(None, 28, 28, 128)	45,056	conv3_block8_0_r…
conv3_block8_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block8_1_c…
conv3_block8_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block8_1_b…
conv3_block8_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block8_1_r…
conv3_block8_concat (Concatenate)	(None, 28, 28, 384)	0	conv3_block7_con… conv3_block8_2_c…
conv3_block9_0_bn (BatchNormalizatio…	(None, 28, 28, 384)	1,536	conv3_block8_con…
conv3_block9_0_relu (Activation)	(None, 28, 28, 384)	0	conv3_block9_0_b…
conv3_block9_1_conv (Conv2D)	(None, 28, 28, 128)	49,152	conv3_block9_0_r…
conv3_block9_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block9_1_c…
conv3_block9_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block9_1_b…
conv3_block9_2_conv (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block9_1_r…
conv3_block9_concat (Concatenate)	(None, 28, 28, 416)	0	conv3_block8_con… conv3_block9_2_c…
conv3_block10_0_bn (BatchNormalizatio…	(None, 28, 28, 416)	1,664	conv3_block9_con…
conv3_block10_0_re… (Activation)	(None, 28, 28, 416)	0	conv3_block10_0_…
conv3_block10_1_co… (Conv2D)	(None, 28, 28, 128)	53,248	conv3_block10_0_…
conv3_block10_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block10_1_…
conv3_block10_1_re… (Activation)	(None, 28, 28, 128)	0	conv3_block10_1_…
conv3_block10_2_co… (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block10_1_…
conv3_block10_conc…	(None, 28, 28,	0	conv3_block9_con…

conv3_block10_conc… (Concatenate)	(None, 28, 28, 448)	0	conv3_block9_con… conv3_block10_2_…
conv3_block11_0_bn (BatchNormalizatio…	(None, 28, 28, 448)	1,792	conv3_block10_co…
conv3_block11_0_re… (Activation)	(None, 28, 28, 448)	0	conv3_block11_0_…
conv3_block11_1_co… (Conv2D)	(None, 28, 28, 128)	57,344	conv3_block11_0_…
conv3_block11_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block11_1_…
conv3_block11_1_re… (Activation)	(None, 28, 28, 128)	0	conv3_block11_1_…
conv3_block11_2_co… (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block11_1_…
conv3_block11_conc… (Concatenate)	(None, 28, 28, 480)	0	conv3_block10_co… conv3_block11_2_…
conv3_block12_0_bn (BatchNormalizatio…	(None, 28, 28, 480)	1,920	conv3_block11_co…
conv3_block12_0_re… (Activation)	(None, 28, 28, 480)	0	conv3_block12_0_…
conv3_block12_1_co… (Conv2D)	(None, 28, 28, 128)	61,440	conv3_block12_0_…
conv3_block12_1_bn (BatchNormalizatio…	(None, 28, 28, 128)	512	conv3_block12_1_…
conv3_block12_1_re… (Activation)	(None, 28, 28, 128)	0	conv3_block12_1_…
conv3_block12_2_co… (Conv2D)	(None, 28, 28, 32)	36,864	conv3_block12_1_…
conv3_block12_conc… (Concatenate)	(None, 28, 28, 512)	0	conv3_block11_co… conv3_block12_2_…
pool3_bn (BatchNormalizatio…	(None, 28, 28, 512)	2,048	conv3_block12_co…
pool3_relu (Activation)	(None, 28, 28, 512)	0	pool3_bn[0][0]
pool3_conv (Conv2D)	(None, 28, 28, 256)	131,072	pool3_relu[0][0]
pool3_pool (AveragePooling2D)	(None, 14, 14, 256)	0	pool3_conv[0][0]
conv4_block1_0_bn (BatchNormalizatio…	(None, 14, 14, 256)	1,024	pool3_pool[0][0]
conv4_block1_0_relu (Activation)	(None, 14, 14, 256)	0	conv4_block1_0_b…
conv4_block1_1_conv (Conv2D)	(None, 14, 14, 128)	32,768	conv4_block1_0_r…
conv4_block1_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block1_1_c…
conv4_block1_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block1_1_b…
conv4_block1_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block1_1_r…

conv4_block1_concat (Concatenate)	(None, 14, 14, 288)	0	pool3_pool[0][0], conv4_block1_2_c…
conv4_block2_0_bn (BatchNormalizatio…	(None, 14, 14, 288)	1,152	conv4_block1_con…
conv4_block2_0_relu (Activation)	(None, 14, 14, 288)	0	conv4_block2_0_b…
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 128)	36,864	conv4_block2_0_r…
conv4_block2_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block2_1_c…
conv4_block2_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block2_1_b…
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block2_1_r…
conv4_block2_concat (Concatenate)	(None, 14, 14, 320)	0	conv4_block1_con… conv4_block2_2_c…
conv4_block3_0_bn (BatchNormalizatio…	(None, 14, 14, 320)	1,280	conv4_block2_con…
conv4_block3_0_relu (Activation)	(None, 14, 14, 320)	0	conv4_block3_0_b…
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 128)	40,960	conv4_block3_0_r…
conv4_block3_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block3_1_c…
conv4_block3_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block3_1_b…
conv4_block3_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block3_1_r…
conv4_block3_concat (Concatenate)	(None, 14, 14, 352)	0	conv4_block2_con… conv4_block3_2_c…
conv4_block4_0_bn (BatchNormalizatio…	(None, 14, 14, 352)	1,408	conv4_block3_con…
conv4_block4_0_relu (Activation)	(None, 14, 14, 352)	0	conv4_block4_0_b…
conv4_block4_1_conv (Conv2D)	(None, 14, 14, 128)	45,056	conv4_block4_0_r…
conv4_block4_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block4_1_c…
conv4_block4_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block4_1_b…
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block4_1_r…
conv4_block4_concat (Concatenate)	(None, 14, 14, 384)	0	conv4_block3_con… conv4_block4_2_c…
conv4_block5_0_bn (BatchNormalizatio…	(None, 14, 14, 384)	1,536	conv4_block4_con…
conv4_block5_0_relu (Activation)	(None, 14, 14, 384)	0	conv4_block5_0_b…
conv4_block5_1_conv (Conv2D)	(None, 14, 14, 128)	49,152	conv4_block5_0_r…

conv4_block5_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block5_1_c…
conv4_block5_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block5_1_b…
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block5_1_r…
conv4_block5_concat (Concatenate)	(None, 14, 14, 416)	0	conv4_block4_con… conv4_block5_2_c…
conv4_block6_0_bn (BatchNormalizatio…	(None, 14, 14, 416)	1,664	conv4_block5_con…
conv4_block6_0_relu (Activation)	(None, 14, 14, 416)	0	conv4_block6_0_b…
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 128)	53,248	conv4_block6_0_r…
conv4_block6_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block6_1_c…
conv4_block6_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block6_1_b…
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block6_1_r…
conv4_block6_concat (Concatenate)	(None, 14, 14, 448)	0	conv4_block5_con… conv4_block6_2_c…
conv4_block7_0_bn (BatchNormalizatio…	(None, 14, 14, 448)	1,792	conv4_block6_con…
conv4_block7_0_relu (Activation)	(None, 14, 14, 448)	0	conv4_block7_0_b…
conv4_block7_1_conv (Conv2D)	(None, 14, 14, 128)	57,344	conv4_block7_0_r…
conv4_block7_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block7_1_c…
conv4_block7_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block7_1_b…
conv4_block7_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block7_1_r…
conv4_block7_concat (Concatenate)	(None, 14, 14, 480)	0	conv4_block6_con… conv4_block7_2_c…
conv4_block8_0_bn (BatchNormalizatio…	(None, 14, 14, 480)	1,920	conv4_block7_con…
conv4_block8_0_relu (Activation)	(None, 14, 14, 480)	0	conv4_block8_0_b…
conv4_block8_1_conv (Conv2D)	(None, 14, 14, 128)	61,440	conv4_block8_0_r…
conv4_block8_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block8_1_c…
conv4_block8_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block8_1_b…
conv4_block8_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block8_1_r…
conv4_block8_concat (Concatenate)	(None, 14, 14, 512)	0	conv4_block7_con… conv4_block8_2_c…

conv4_block9_0_bn (BatchNormalizatio…	(None, 14, 14, 512)	2,048	conv4_block8_con…
conv4_block9_0_relu (Activation)	(None, 14, 14, 512)	0	conv4_block9_0_b…
conv4_block9_1_conv (Conv2D)	(None, 14, 14, 128)	65,536	conv4_block9_0_r…
conv4_block9_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block9_1_c…
conv4_block9_1_relu (Activation)	(None, 14, 14, 128)	0	conv4_block9_1_b…
conv4_block9_2_conv (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block9_1_r…
conv4_block9_concat (Concatenate)	(None, 14, 14, 544)	0	conv4_block8_con… conv4_block9_2_c…
conv4_block10_0_bn (BatchNormalizatio…	(None, 14, 14, 544)	2,176	conv4_block9_con…
conv4_block10_0_re… (Activation)	(None, 14, 14, 544)	0	conv4_block10_0_…
conv4_block10_1_co… (Conv2D)	(None, 14, 14, 128)	69,632	conv4_block10_0_…
conv4_block10_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block10_1_…
conv4_block10_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block10_1_…
conv4_block10_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block10_1_…
conv4_block10_conc… (Concatenate)	(None, 14, 14, 576)	0	conv4_block9_con… conv4_block10_2_…
conv4_block11_0_bn (BatchNormalizatio…	(None, 14, 14, 576)	2,304	conv4_block10_co…
conv4_block11_0_re… (Activation)	(None, 14, 14, 576)	0	conv4_block11_0_…
conv4_block11_1_co… (Conv2D)	(None, 14, 14, 128)	73,728	conv4_block11_0_…
conv4_block11_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block11_1_…
conv4_block11_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block11_1_…
conv4_block11_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block11_1_…
conv4_block11_conc… (Concatenate)	(None, 14, 14, 608)	0	conv4_block10_co… conv4_block11_2_…
conv4_block12_0_bn (BatchNormalizatio…	(None, 14, 14, 608)	2,432	conv4_block11_co…
conv4_block12_0_re… (Activation)	(None, 14, 14, 608)	0	conv4_block12_0_…
conv4_block12_1_co… (Conv2D)	(None, 14, 14, 128)	77,824	conv4_block12_0_…
conv4_block12_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block12_1_…

conv4_block12_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block12_1_…
conv4_block12_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block12_1_…
conv4_block12_conc… (Concatenate)	(None, 14, 14, 640)	0	conv4_block11_co… conv4_block12_2_…
conv4_block13_0_bn (BatchNormalizatio…	(None, 14, 14, 640)	2,560	conv4_block12_co…
conv4_block13_0_re… (Activation)	(None, 14, 14, 640)	0	conv4_block13_0_…
conv4_block13_1_co… (Conv2D)	(None, 14, 14, 128)	81,920	conv4_block13_0_…
conv4_block13_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block13_1_…
conv4_block13_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block13_1_…
conv4_block13_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block13_1_…
conv4_block13_conc… (Concatenate)	(None, 14, 14, 672)	0	conv4_block12_co… conv4_block13_2_…
conv4_block14_0_bn (BatchNormalizatio…	(None, 14, 14, 672)	2,688	conv4_block13_co…
conv4_block14_0_re… (Activation)	(None, 14, 14, 672)	0	conv4_block14_0_…
conv4_block14_1_co… (Conv2D)	(None, 14, 14, 128)	86,016	conv4_block14_0_…
conv4_block14_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block14_1_…
conv4_block14_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block14_1_…
conv4_block14_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block14_1_…
conv4_block14_conc… (Concatenate)	(None, 14, 14, 704)	0	conv4_block13_co… conv4_block14_2_…
conv4_block15_0_bn (BatchNormalizatio…	(None, 14, 14, 704)	2,816	conv4_block14_co…
conv4_block15_0_re… (Activation)	(None, 14, 14, 704)	0	conv4_block15_0_…
conv4_block15_1_co… (Conv2D)	(None, 14, 14, 128)	90,112	conv4_block15_0_…
conv4_block15_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block15_1_…
conv4_block15_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block15_1_…
conv4_block15_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block15_1_…
conv4_block15_conc… (Concatenate)	(None, 14, 14, 736)	0	conv4_block14_co… conv4_block15_2_…
conv4_block16_0_bn	(None, 14, 14,	2,944	conv4_block15_co…

(BatchNormalizatio…	736)		
conv4_block16_0_re… (Activation)	(None, 14, 14, 736)	0	conv4_block16_0_…
conv4_block16_1_co… (Conv2D)	(None, 14, 14, 128)	94,208	conv4_block16_0_…
conv4_block16_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block16_1_…
conv4_block16_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block16_1_…
conv4_block16_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block16_1_…
conv4_block16_conc… (Concatenate)	(None, 14, 14, 768)	0	conv4_block15_co… conv4_block16_2_…
conv4_block17_0_bn (BatchNormalizatio…	(None, 14, 14, 768)	3,072	conv4_block16_co…
conv4_block17_0_re… (Activation)	(None, 14, 14, 768)	0	conv4_block17_0_…
conv4_block17_1_co… (Conv2D)	(None, 14, 14, 128)	98,304	conv4_block17_0_…
conv4_block17_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block17_1_…
conv4_block17_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block17_1_…
conv4_block17_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block17_1_…
conv4_block17_conc… (Concatenate)	(None, 14, 14, 800)	0	conv4_block16_co… conv4_block17_2_…
conv4_block18_0_bn (BatchNormalizatio…	(None, 14, 14, 800)	3,200	conv4_block17_co…
conv4_block18_0_re… (Activation)	(None, 14, 14, 800)	0	conv4_block18_0_…
conv4_block18_1_co… (Conv2D)	(None, 14, 14, 128)	102,400	conv4_block18_0_…
conv4_block18_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block18_1_…
conv4_block18_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block18_1_…
conv4_block18_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block18_1_…
conv4_block18_conc… (Concatenate)	(None, 14, 14, 832)	0	conv4_block17_co… conv4_block18_2_…
conv4_block19_0_bn (BatchNormalizatio…	(None, 14, 14, 832)	3,328	conv4_block18_co…
conv4_block19_0_re… (Activation)	(None, 14, 14, 832)	0	conv4_block19_0_…
conv4_block19_1_co… (Conv2D)	(None, 14, 14, 128)	106,496	conv4_block19_0_…
conv4_block19_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block19_1_…
conv4_block19_1_re…	(None, 14, 14,	0	conv4_block19_1_…

(Activation)	128)		
conv4_block19_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block19_1_…
conv4_block19_conc… (Concatenate)	(None, 14, 14, 864)	0	conv4_block18_co… conv4_block19_2_…
conv4_block20_0_bn (BatchNormalizatio…	(None, 14, 14, 864)	3,456	conv4_block19_co…
conv4_block20_0_re… (Activation)	(None, 14, 14, 864)	0	conv4_block20_0_…
conv4_block20_1_co… (Conv2D)	(None, 14, 14, 128)	110,592	conv4_block20_0_…
conv4_block20_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block20_1_…
conv4_block20_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block20_1_…
conv4_block20_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block20_1_…
conv4_block20_conc… (Concatenate)	(None, 14, 14, 896)	0	conv4_block19_co… conv4_block20_2_…
conv4_block21_0_bn (BatchNormalizatio…	(None, 14, 14, 896)	3,584	conv4_block20_co…
conv4_block21_0_re… (Activation)	(None, 14, 14, 896)	0	conv4_block21_0_…
conv4_block21_1_co… (Conv2D)	(None, 14, 14, 128)	114,688	conv4_block21_0_…
conv4_block21_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block21_1_…
conv4_block21_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block21_1_…
conv4_block21_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block21_1_…
conv4_block21_conc… (Concatenate)	(None, 14, 14, 928)	0	conv4_block20_co… conv4_block21_2_…
conv4_block22_0_bn (BatchNormalizatio…	(None, 14, 14, 928)	3,712	conv4_block21_co…
conv4_block22_0_re… (Activation)	(None, 14, 14, 928)	0	conv4_block22_0_…
conv4_block22_1_co… (Conv2D)	(None, 14, 14, 128)	118,784	conv4_block22_0_…
conv4_block22_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block22_1_…
conv4_block22_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block22_1_…
conv4_block22_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block22_1_…
conv4_block22_conc… (Concatenate)	(None, 14, 14, 960)	0	conv4_block21_co… conv4_block22_2_…
conv4_block23_0_bn (BatchNormalizatio…	(None, 14, 14, 960)	3,840	conv4_block22_co…
conv4_block23_0_re…	(None, 14, 14	0	conv4_block23_0_

(Activation)	960)	0	
conv4_block23_1_co… (Conv2D)	(None, 14, 14, 128)	122,880	conv4_block23_0_…
conv4_block23_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block23_1_…
conv4_block23_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block23_1_…
conv4_block23_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block23_1_…
conv4_block23_conc… (Concatenate)	(None, 14, 14, 992)	0	conv4_block22_co… conv4_block23_2_…
conv4_block24_0_bn (BatchNormalizatio…	(None, 14, 14, 992)	3,968	conv4_block23_co…
conv4_block24_0_re… (Activation)	(None, 14, 14, 992)	0	conv4_block24_0_…
conv4_block24_1_co… (Conv2D)	(None, 14, 14, 128)	126,976	conv4_block24_0_…
conv4_block24_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block24_1_…
conv4_block24_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block24_1_…
conv4_block24_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block24_1_…
conv4_block24_conc… (Concatenate)	(None, 14, 14, 1024)	0	conv4_block23_co… conv4_block24_2_…
conv4_block25_0_bn (BatchNormalizatio…	(None, 14, 14, 1024)	4,096	conv4_block24_co…
conv4_block25_0_re… (Activation)	(None, 14, 14, 1024)	0	conv4_block25_0_…
conv4_block25_1_co… (Conv2D)	(None, 14, 14, 128)	131,072	conv4_block25_0_…
conv4_block25_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block25_1_…
conv4_block25_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block25_1_…
conv4_block25_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block25_1_…
conv4_block25_conc… (Concatenate)	(None, 14, 14, 1056)	0	conv4_block24_co… conv4_block25_2_…
conv4_block26_0_bn (BatchNormalizatio…	(None, 14, 14, 1056)	4,224	conv4_block25_co…
conv4_block26_0_re… (Activation)	(None, 14, 14, 1056)	0	conv4_block26_0_…
conv4_block26_1_co… (Conv2D)	(None, 14, 14, 128)	135,168	conv4_block26_0_…
conv4_block26_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block26_1_…
conv4_block26_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block26_1_…

conv4_block26_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block26_1_…
conv4_block26_conc… (Concatenate)	(None, 14, 14, 1088)	0	conv4_block25_co… conv4_block26_2_…
conv4_block27_0_bn (BatchNormalizatio…	(None, 14, 14, 1088)	4,352	conv4_block26_co…
conv4_block27_0_re… (Activation)	(None, 14, 14, 1088)	0	conv4_block27_0_…
conv4_block27_1_co… (Conv2D)	(None, 14, 14, 128)	139,264	conv4_block27_0_…
conv4_block27_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block27_1_…
conv4_block27_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block27_1_…
conv4_block27_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block27_1_…
conv4_block27_conc… (Concatenate)	(None, 14, 14, 1120)	0	conv4_block26_co… conv4_block27_2_…
conv4_block28_0_bn (BatchNormalizatio…	(None, 14, 14, 1120)	4,480	conv4_block27_co…
conv4_block28_0_re… (Activation)	(None, 14, 14, 1120)	0	conv4_block28_0_…
conv4_block28_1_co… (Conv2D)	(None, 14, 14, 128)	143,360	conv4_block28_0_…
conv4_block28_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block28_1_…
conv4_block28_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block28_1_…
conv4_block28_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block28_1_…
conv4_block28_conc… (Concatenate)	(None, 14, 14, 1152)	0	conv4_block27_co… conv4_block28_2_…
conv4_block29_0_bn (BatchNormalizatio…	(None, 14, 14, 1152)	4,608	conv4_block28_co…
conv4_block29_0_re… (Activation)	(None, 14, 14, 1152)	0	conv4_block29_0_…
conv4_block29_1_co… (Conv2D)	(None, 14, 14, 128)	147,456	conv4_block29_0_…
conv4_block29_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block29_1_…
conv4_block29_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block29_1_…
conv4_block29_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block29_1_…
conv4_block29_conc… (Concatenate)	(None, 14, 14, 1184)	0	conv4_block28_co… conv4_block29_2_…
conv4_block30_0_bn (BatchNormalizatio…	(None, 14, 14, 1184)	4,736	conv4_block29_co…
conv4_block30_0_re… (Activation)	(None, 14, 14, 1184)	0	conv4_block30_0_…

conv4_block30_1_co… (Conv2D)	(None, 14, 14, 128)	151,552	conv4_block30_0_…
conv4_block30_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block30_1_…
conv4_block30_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block30_1_…
conv4_block30_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block30_1_…
conv4_block30_conc… (Concatenate)	(None, 14, 14, 1216)	0	conv4_block29_co… conv4_block30_2_…
conv4_block31_0_bn (BatchNormalizatio…	(None, 14, 14, 1216)	4,864	conv4_block30_co…
conv4_block31_0_re… (Activation)	(None, 14, 14, 1216)	0	conv4_block31_0_…
conv4_block31_1_co… (Conv2D)	(None, 14, 14, 128)	155,648	conv4_block31_0_…
conv4_block31_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block31_1_…
conv4_block31_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block31_1_…
conv4_block31_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block31_1_…
conv4_block31_conc… (Concatenate)	(None, 14, 14, 1248)	0	conv4_block30_co… conv4_block31_2_…
conv4_block32_0_bn (BatchNormalizatio…	(None, 14, 14, 1248)	4,992	conv4_block31_co…
conv4_block32_0_re… (Activation)	(None, 14, 14, 1248)	0	conv4_block32_0_…
conv4_block32_1_co… (Conv2D)	(None, 14, 14, 128)	159,744	conv4_block32_0_…
conv4_block32_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block32_1_…
conv4_block32_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block32_1_…
conv4_block32_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block32_1_…
conv4_block32_conc… (Concatenate)	(None, 14, 14, 1280)	0	conv4_block31_co… conv4_block32_2_…
conv4_block33_0_bn (BatchNormalizatio…	(None, 14, 14, 1280)	5,120	conv4_block32_co…
conv4_block33_0_re… (Activation)	(None, 14, 14, 1280)	0	conv4_block33_0_…
conv4_block33_1_co… (Conv2D)	(None, 14, 14, 128)	163,840	conv4_block33_0_…
conv4_block33_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block33_1_…
conv4_block33_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block33_1_…
conv4_block33_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block33_1_…

conv4_block33_conc… (Concatenate)	(None, 14, 14, 1312)	0	conv4_block32_co… conv4_block33_2_…
conv4_block34_0_bn (BatchNormalizatio…	(None, 14, 14, 1312)	5,248	conv4_block33_co…
conv4_block34_0_re… (Activation)	(None, 14, 14, 1312)	0	conv4_block34_0_…
conv4_block34_1_co… (Conv2D)	(None, 14, 14, 128)	167,936	conv4_block34_0_…
conv4_block34_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block34_1_…
conv4_block34_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block34_1_…
conv4_block34_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block34_1_…
conv4_block34_conc… (Concatenate)	(None, 14, 14, 1344)	0	conv4_block33_co… conv4_block34_2_…
conv4_block35_0_bn (BatchNormalizatio…	(None, 14, 14, 1344)	5,376	conv4_block34_co…
conv4_block35_0_re… (Activation)	(None, 14, 14, 1344)	0	conv4_block35_0_…
conv4_block35_1_co… (Conv2D)	(None, 14, 14, 128)	172,032	conv4_block35_0_…
conv4_block35_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block35_1_…
conv4_block35_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block35_1_…
conv4_block35_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block35_1_…
conv4_block35_conc… (Concatenate)	(None, 14, 14, 1376)	0	conv4_block34_co… conv4_block35_2_…
conv4_block36_0_bn (BatchNormalizatio…	(None, 14, 14, 1376)	5,504	conv4_block35_co…
conv4_block36_0_re… (Activation)	(None, 14, 14, 1376)	0	conv4_block36_0_…
conv4_block36_1_co… (Conv2D)	(None, 14, 14, 128)	176,128	conv4_block36_0_…
conv4_block36_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block36_1_…
conv4_block36_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block36_1_…
conv4_block36_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block36_1_…
conv4_block36_conc… (Concatenate)	(None, 14, 14, 1408)	0	conv4_block35_co… conv4_block36_2_…
conv4_block37_0_bn (BatchNormalizatio…	(None, 14, 14, 1408)	5,632	conv4_block36_co…
conv4_block37_0_re… (Activation)	(None, 14, 14, 1408)	0	conv4_block37_0_…
conv4_block37_1_co… (Conv2D)	(None, 14, 14, 128)	180,224	conv4_block37_0_…

conv4_block37_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block37_1_…
conv4_block37_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block37_1_…
conv4_block37_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block37_1_…
conv4_block37_conc… (Concatenate)	(None, 14, 14, 1440)	0	conv4_block36_co… conv4_block37_2_…
conv4_block38_0_bn (BatchNormalizatio…	(None, 14, 14, 1440)	5,760	conv4_block37_co…
conv4_block38_0_re… (Activation)	(None, 14, 14, 1440)	0	conv4_block38_0_…
conv4_block38_1_co… (Conv2D)	(None, 14, 14, 128)	184,320	conv4_block38_0_…
conv4_block38_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block38_1_…
conv4_block38_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block38_1_…
conv4_block38_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block38_1_…
conv4_block38_conc… (Concatenate)	(None, 14, 14, 1472)	0	conv4_block37_co… conv4_block38_2_…
conv4_block39_0_bn (BatchNormalizatio…	(None, 14, 14, 1472)	5,888	conv4_block38_co…
conv4_block39_0_re… (Activation)	(None, 14, 14, 1472)	0	conv4_block39_0_…
conv4_block39_1_co… (Conv2D)	(None, 14, 14, 128)	188,416	conv4_block39_0_…
conv4_block39_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block39_1_…
conv4_block39_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block39_1_…
conv4_block39_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block39_1_…
conv4_block39_conc… (Concatenate)	(None, 14, 14, 1504)	0	conv4_block38_co… conv4_block39_2_…
conv4_block40_0_bn (BatchNormalizatio…	(None, 14, 14, 1504)	6,016	conv4_block39_co…
conv4_block40_0_re… (Activation)	(None, 14, 14, 1504)	0	conv4_block40_0_…
conv4_block40_1_co… (Conv2D)	(None, 14, 14, 128)	192,512	conv4_block40_0_…
conv4_block40_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block40_1_…
conv4_block40_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block40_1_…
conv4_block40_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block40_1_…
conv4_block40_conc… (Concatenate)	(None, 14, 14, 1536)	0	conv4_block39_co… conv4_block40_2_…

(Concatenate)	1536)		conv4_block40_2_…
conv4_block41_0_bn (BatchNormalizatio…	(None, 14, 14, 1536)	6,144	conv4_block40_co…
conv4_block41_0_re… (Activation)	(None, 14, 14, 1536)	0	conv4_block41_0_…
conv4_block41_1_co… (Conv2D)	(None, 14, 14, 128)	196,608	conv4_block41_0_…
conv4_block41_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block41_1_…
conv4_block41_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block41_1_…
conv4_block41_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block41_1_…
conv4_block41_conc… (Concatenate)	(None, 14, 14, 1568)	0	conv4_block40_co… conv4_block41_2_…
conv4_block42_0_bn (BatchNormalizatio…	(None, 14, 14, 1568)	6,272	conv4_block41_co…
conv4_block42_0_re… (Activation)	(None, 14, 14, 1568)	0	conv4_block42_0_…
conv4_block42_1_co… (Conv2D)	(None, 14, 14, 128)	200,704	conv4_block42_0_…
conv4_block42_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block42_1_…
conv4_block42_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block42_1_…
conv4_block42_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block42_1_…
conv4_block42_conc… (Concatenate)	(None, 14, 14, 1600)	0	conv4_block41_co… conv4_block42_2_…
conv4_block43_0_bn (BatchNormalizatio…	(None, 14, 14, 1600)	6,400	conv4_block42_co…
conv4_block43_0_re… (Activation)	(None, 14, 14, 1600)	0	conv4_block43_0_…
conv4_block43_1_co… (Conv2D)	(None, 14, 14, 128)	204,800	conv4_block43_0_…
conv4_block43_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block43_1_…
conv4_block43_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block43_1_…
conv4_block43_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block43_1_…
conv4_block43_conc… (Concatenate)	(None, 14, 14, 1632)	0	conv4_block42_co… conv4_block43_2_…
conv4_block44_0_bn (BatchNormalizatio…	(None, 14, 14, 1632)	6,528	conv4_block43_co…
conv4_block44_0_re… (Activation)	(None, 14, 14, 1632)	0	conv4_block44_0_…
conv4_block44_1_co… (Conv2D)	(None, 14, 14, 128)	208,896	conv4_block44_0_…
conv4_block44_1_bn	(None, 14, 14,	512	conv4_block44_1_…

(BatchNormalizatio…	128)		
conv4_block44_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block44_1_…
conv4_block44_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block44_1_…
conv4_block44_conc… (Concatenate)	(None, 14, 14, 1664)	0	conv4_block43_co… conv4_block44_2_…
conv4_block45_0_bn (BatchNormalizatio…	(None, 14, 14, 1664)	6,656	conv4_block44_co…
conv4_block45_0_re… (Activation)	(None, 14, 14, 1664)	0	conv4_block45_0_…
conv4_block45_1_co… (Conv2D)	(None, 14, 14, 128)	212,992	conv4_block45_0_…
conv4_block45_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block45_1_…
conv4_block45_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block45_1_…
conv4_block45_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block45_1_…
conv4_block45_conc… (Concatenate)	(None, 14, 14, 1696)	0	conv4_block44_co… conv4_block45_2_…
conv4_block46_0_bn (BatchNormalizatio…	(None, 14, 14, 1696)	6,784	conv4_block45_co…
conv4_block46_0_re… (Activation)	(None, 14, 14, 1696)	0	conv4_block46_0_…
conv4_block46_1_co… (Conv2D)	(None, 14, 14, 128)	217,088	conv4_block46_0_…
conv4_block46_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block46_1_…
conv4_block46_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block46_1_…
conv4_block46_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block46_1_…
conv4_block46_conc… (Concatenate)	(None, 14, 14, 1728)	0	conv4_block45_co… conv4_block46_2_…
conv4_block47_0_bn (BatchNormalizatio…	(None, 14, 14, 1728)	6,912	conv4_block46_co…
conv4_block47_0_re… (Activation)	(None, 14, 14, 1728)	0	conv4_block47_0_…
conv4_block47_1_co… (Conv2D)	(None, 14, 14, 128)	221,184	conv4_block47_0_…
conv4_block47_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block47_1_…
conv4_block47_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block47_1_…
conv4_block47_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block47_1_…
conv4_block47_conc… (Concatenate)	(None, 14, 14, 1760)	0	conv4_block46_co… conv4_block47_2_…
conv4_block48_0_bn	(None, 14, 14,	7,040	conv4_block47_co…

(BatchNormalizatio…	1760)		
conv4_block48_0_re… (Activation)	(None, 14, 14, 1760)	0	conv4_block48_0_…
conv4_block48_1_co… (Conv2D)	(None, 14, 14, 128)	225,280	conv4_block48_0_…
conv4_block48_1_bn (BatchNormalizatio…	(None, 14, 14, 128)	512	conv4_block48_1_…
conv4_block48_1_re… (Activation)	(None, 14, 14, 128)	0	conv4_block48_1_…
conv4_block48_2_co… (Conv2D)	(None, 14, 14, 32)	36,864	conv4_block48_1_…
conv4_block48_conc… (Concatenate)	(None, 14, 14, 1792)	0	conv4_block47_co… conv4_block48_2_…
pool4_bn (BatchNormalizatio…	(None, 14, 14, 1792)	7,168	conv4_block48_co…
pool4_relu (Activation)	(None, 14, 14, 1792)	0	pool4_bn[0][0]
pool4_conv (Conv2D)	(None, 14, 14, 896)	1,605,632	pool4_relu[0][0]
pool4_pool (AveragePooling2D)	(None, 7, 7, 896)	0	pool4_conv[0][0]
conv5_block1_0_bn (BatchNormalizatio…	(None, 7, 7, 896)	3,584	pool4_pool[0][0]
conv5_block1_0_relu (Activation)	(None, 7, 7, 896)	0	conv5_block1_0_b…
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 128)	114,688	conv5_block1_0_r…
conv5_block1_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block1_1_c…
conv5_block1_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block1_1_b…
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block1_1_r…
conv5_block1_concat (Concatenate)	(None, 7, 7, 928)	0	pool4_pool[0][0], conv5_block1_2_c…
conv5_block2_0_bn (BatchNormalizatio…	(None, 7, 7, 928)	3,712	conv5_block1_con…
conv5_block2_0_relu (Activation)	(None, 7, 7, 928)	0	conv5_block2_0_b…
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 128)	118,784	conv5_block2_0_r…
conv5_block2_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block2_1_c…
conv5_block2_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block2_1_b…
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block2_1_r…
conv5_block2_concat (Concatenate)	(None, 7, 7, 960)	0	conv5_block1_con… conv5_block2_2_c…

conv5_block3_0_bn (BatchNormalizatio…	(None, 7, 7, 960)	3,840	conv5_block2_con…
conv5_block3_0_relu (Activation)	(None, 7, 7, 960)	0	conv5_block3_0_b…
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 128)	122,880	conv5_block3_0_r…
conv5_block3_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block3_1_c…
conv5_block3_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block3_1_b…
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block3_1_r…
conv5_block3_concat (Concatenate)	(None, 7, 7, 992)	0	conv5_block2_con… conv5_block3_2_c…
conv5_block4_0_bn (BatchNormalizatio…	(None, 7, 7, 992)	3,968	conv5_block3_con…
conv5_block4_0_relu (Activation)	(None, 7, 7, 992)	0	conv5_block4_0_b…
conv5_block4_1_conv (Conv2D)	(None, 7, 7, 128)	126,976	conv5_block4_0_r…
conv5_block4_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block4_1_c…
conv5_block4_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block4_1_b…
conv5_block4_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block4_1_r…
conv5_block4_concat (Concatenate)	(None, 7, 7, 1024)	0	conv5_block3_con… conv5_block4_2_c…
conv5_block5_0_bn (BatchNormalizatio…	(None, 7, 7, 1024)	4,096	conv5_block4_con…
conv5_block5_0_relu (Activation)	(None, 7, 7, 1024)	0	conv5_block5_0_b…
conv5_block5_1_conv (Conv2D)	(None, 7, 7, 128)	131,072	conv5_block5_0_r…
conv5_block5_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block5_1_c…
conv5_block5_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block5_1_b…
conv5_block5_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block5_1_r…
conv5_block5_concat (Concatenate)	(None, 7, 7, 1056)	0	conv5_block4_con… conv5_block5_2_c…
conv5_block6_0_bn (BatchNormalizatio…	(None, 7, 7, 1056)	4,224	conv5_block5_con…
conv5_block6_0_relu (Activation)	(None, 7, 7, 1056)	0	conv5_block6_0_b…
conv5_block6_1_conv (Conv2D)	(None, 7, 7, 128)	135,168	conv5_block6_0_r…
conv5_block6_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block6_1_c…

conv5_block6_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block6_1_b…
conv5_block6_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block6_1_r…
conv5_block6_concat (Concatenate)	(None, 7, 7, 1088)	0	conv5_block5_con… conv5_block6_2_c…
conv5_block7_0_bn (BatchNormalizatio…	(None, 7, 7, 1088)	4,352	conv5_block6_con…
conv5_block7_0_relu (Activation)	(None, 7, 7, 1088)	0	conv5_block7_0_b…
conv5_block7_1_conv (Conv2D)	(None, 7, 7, 128)	139,264	conv5_block7_0_r…
conv5_block7_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block7_1_c…
conv5_block7_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block7_1_b…
conv5_block7_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block7_1_r…
conv5_block7_concat (Concatenate)	(None, 7, 7, 1120)	0	conv5_block6_con… conv5_block7_2_c…
conv5_block8_0_bn (BatchNormalizatio…	(None, 7, 7, 1120)	4,480	conv5_block7_con…
conv5_block8_0_relu (Activation)	(None, 7, 7, 1120)	0	conv5_block8_0_b…
conv5_block8_1_conv (Conv2D)	(None, 7, 7, 128)	143,360	conv5_block8_0_r…
conv5_block8_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block8_1_c…
conv5_block8_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block8_1_b…
conv5_block8_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block8_1_r…
conv5_block8_concat (Concatenate)	(None, 7, 7, 1152)	0	conv5_block7_con… conv5_block8_2_c…
conv5_block9_0_bn (BatchNormalizatio…	(None, 7, 7, 1152)	4,608	conv5_block8_con…
conv5_block9_0_relu (Activation)	(None, 7, 7, 1152)	0	conv5_block9_0_b…
conv5_block9_1_conv (Conv2D)	(None, 7, 7, 128)	147,456	conv5_block9_0_r…
conv5_block9_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block9_1_c…
conv5_block9_1_relu (Activation)	(None, 7, 7, 128)	0	conv5_block9_1_b…
conv5_block9_2_conv (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block9_1_r…
conv5_block9_concat (Concatenate)	(None, 7, 7, 1184)	0	conv5_block8_con… conv5_block9_2_c…
conv5_block10_0_bn (BatchNormalizatio…	(None, 7, 7, 1184)	4,736	conv5_block9_con…

conv5_block10_0_re… (Activation)	(None, 7, 7, 1184)	0	conv5_block10_0_…
conv5_block10_1_co… (Conv2D)	(None, 7, 7, 128)	151,552	conv5_block10_0_…
conv5_block10_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block10_1_…
conv5_block10_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block10_1_…
conv5_block10_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block10_1_…
conv5_block10_conc… (Concatenate)	(None, 7, 7, 1216)	0	conv5_block9_con… conv5_block10_2_…
conv5_block11_0_bn (BatchNormalizatio…	(None, 7, 7, 1216)	4,864	conv5_block10_co…
conv5_block11_0_re… (Activation)	(None, 7, 7, 1216)	0	conv5_block11_0_…
conv5_block11_1_co… (Conv2D)	(None, 7, 7, 128)	155,648	conv5_block11_0_…
conv5_block11_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block11_1_…
conv5_block11_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block11_1_…
conv5_block11_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block11_1_…
conv5_block11_conc… (Concatenate)	(None, 7, 7, 1248)	0	conv5_block10_co… conv5_block11_2_…
conv5_block12_0_bn (BatchNormalizatio…	(None, 7, 7, 1248)	4,992	conv5_block11_co…
conv5_block12_0_re… (Activation)	(None, 7, 7, 1248)	0	conv5_block12_0_…
conv5_block12_1_co… (Conv2D)	(None, 7, 7, 128)	159,744	conv5_block12_0_…
conv5_block12_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block12_1_…
conv5_block12_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block12_1_…
conv5_block12_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block12_1_…
conv5_block12_conc… (Concatenate)	(None, 7, 7, 1280)	0	conv5_block11_co… conv5_block12_2_…
conv5_block13_0_bn (BatchNormalizatio…	(None, 7, 7, 1280)	5,120	conv5_block12_co…
conv5_block13_0_re… (Activation)	(None, 7, 7, 1280)	0	conv5_block13_0_…
conv5_block13_1_co… (Conv2D)	(None, 7, 7, 128)	163,840	conv5_block13_0_…
conv5_block13_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block13_1_…
conv5_block13_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block13_1_…

conv5_block13_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block13_1_…
conv5_block13_conc… (Concatenate)	(None, 7, 7, 1312)	0	conv5_block12_co… conv5_block13_2_…
conv5_block14_0_bn (BatchNormalizatio…	(None, 7, 7, 1312)	5,248	conv5_block13_co…
conv5_block14_0_re… (Activation)	(None, 7, 7, 1312)	0	conv5_block14_0_…
conv5_block14_1_co… (Conv2D)	(None, 7, 7, 128)	167,936	conv5_block14_0_…
conv5_block14_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block14_1_…
conv5_block14_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block14_1_…
conv5_block14_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block14_1_…
conv5_block14_conc… (Concatenate)	(None, 7, 7, 1344)	0	conv5_block13_co… conv5_block14_2_…
conv5_block15_0_bn (BatchNormalizatio…	(None, 7, 7, 1344)	5,376	conv5_block14_co…
conv5_block15_0_re… (Activation)	(None, 7, 7, 1344)	0	conv5_block15_0_…
conv5_block15_1_co… (Conv2D)	(None, 7, 7, 128)	172,032	conv5_block15_0_…
conv5_block15_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block15_1_…
conv5_block15_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block15_1_…
conv5_block15_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block15_1_…
conv5_block15_conc… (Concatenate)	(None, 7, 7, 1376)	0	conv5_block14_co… conv5_block15_2_…
conv5_block16_0_bn (BatchNormalizatio…	(None, 7, 7, 1376)	5,504	conv5_block15_co…
conv5_block16_0_re… (Activation)	(None, 7, 7, 1376)	0	conv5_block16_0_…
conv5_block16_1_co… (Conv2D)	(None, 7, 7, 128)	176,128	conv5_block16_0_…
conv5_block16_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block16_1_…
conv5_block16_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block16_1_…
conv5_block16_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block16_1_…
conv5_block16_conc… (Concatenate)	(None, 7, 7, 1408)	0	conv5_block15_co… conv5_block16_2_…
conv5_block17_0_bn (BatchNormalizatio…	(None, 7, 7, 1408)	5,632	conv5_block16_co…
conv5_block17_0_re… (Activation)	(None, 7, 7, 1408)	0	conv5_block17_0_…

conv5_block17_1_co… (Conv2D)	(None, 7, 7, 128)	180,224	conv5_block17_0_…
conv5_block17_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block17_1_…
conv5_block17_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block17_1_…
conv5_block17_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block17_1_…
conv5_block17_conc… (Concatenate)	(None, 7, 7, 1440)	0	conv5_block16_co… conv5_block17_2_…
conv5_block18_0_bn (BatchNormalizatio…	(None, 7, 7, 1440)	5,760	conv5_block17_co…
conv5_block18_0_re… (Activation)	(None, 7, 7, 1440)	0	conv5_block18_0_…
conv5_block18_1_co… (Conv2D)	(None, 7, 7, 128)	184,320	conv5_block18_0_…
conv5_block18_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block18_1_…
conv5_block18_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block18_1_…
conv5_block18_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block18_1_…
conv5_block18_conc… (Concatenate)	(None, 7, 7, 1472)	0	conv5_block17_co… conv5_block18_2_…
conv5_block19_0_bn (BatchNormalizatio…	(None, 7, 7, 1472)	5,888	conv5_block18_co…
conv5_block19_0_re… (Activation)	(None, 7, 7, 1472)	0	conv5_block19_0_…
conv5_block19_1_co… (Conv2D)	(None, 7, 7, 128)	188,416	conv5_block19_0_…
conv5_block19_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block19_1_…
conv5_block19_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block19_1_…
conv5_block19_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block19_1_…
conv5_block19_conc… (Concatenate)	(None, 7, 7, 1504)	0	conv5_block18_co… conv5_block19_2_…
conv5_block20_0_bn (BatchNormalizatio…	(None, 7, 7, 1504)	6,016	conv5_block19_co…
conv5_block20_0_re… (Activation)	(None, 7, 7, 1504)	0	conv5_block20_0_…
conv5_block20_1_co… (Conv2D)	(None, 7, 7, 128)	192,512	conv5_block20_0_…
conv5_block20_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block20_1_…
conv5_block20_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block20_1_…
conv5_block20_2_co…	(None, 7, 7, 32)	36,864	conv5_block20_1_…

(Conv2D)			
conv5_block20_conc… (Concatenate)	(None, 7, 7, 1536)	0	conv5_block19_co… conv5_block20_2_…
conv5_block21_0_bn (BatchNormalizatio…	(None, 7, 7, 1536)	6,144	conv5_block20_co…
conv5_block21_0_re… (Activation)	(None, 7, 7, 1536)	0	conv5_block21_0_…
conv5_block21_1_co… (Conv2D)	(None, 7, 7, 128)	196,608	conv5_block21_0_…
conv5_block21_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block21_1_…
conv5_block21_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block21_1_…
conv5_block21_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block21_1_…
conv5_block21_conc… (Concatenate)	(None, 7, 7, 1568)	0	conv5_block20_co… conv5_block21_2_…
conv5_block22_0_bn (BatchNormalizatio…	(None, 7, 7, 1568)	6,272	conv5_block21_co…
conv5_block22_0_re… (Activation)	(None, 7, 7, 1568)	0	conv5_block22_0_…
conv5_block22_1_co… (Conv2D)	(None, 7, 7, 128)	200,704	conv5_block22_0_…
conv5_block22_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block22_1_…
conv5_block22_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block22_1_…
conv5_block22_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block22_1_…
conv5_block22_conc… (Concatenate)	(None, 7, 7, 1600)	0	conv5_block21_co… conv5_block22_2_…
conv5_block23_0_bn (BatchNormalizatio…	(None, 7, 7, 1600)	6,400	conv5_block22_co…
conv5_block23_0_re… (Activation)	(None, 7, 7, 1600)	0	conv5_block23_0_…
conv5_block23_1_co… (Conv2D)	(None, 7, 7, 128)	204,800	conv5_block23_0_…
conv5_block23_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block23_1_…
conv5_block23_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block23_1_…
conv5_block23_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block23_1_…
conv5_block23_conc… (Concatenate)	(None, 7, 7, 1632)	0	conv5_block22_co… conv5_block23_2_…
conv5_block24_0_bn (BatchNormalizatio…	(None, 7, 7, 1632)	6,528	conv5_block23_co…
conv5_block24_0_re… (Activation)	(None, 7, 7, 1632)	0	conv5_block24_0_…
conv5_block24_1_co…	(None, 7, 7, 128)	208,896	conv5_block24_0_…

(Conv2D)			
conv5_block24_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block24_1_…
conv5_block24_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block24_1_…
conv5_block24_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block24_1_…
conv5_block24_conc… (Concatenate)	(None, 7, 7, 1664)	0	conv5_block23_co… conv5_block24_2_…
conv5_block25_0_bn (BatchNormalizatio…	(None, 7, 7, 1664)	6,656	conv5_block24_co…
conv5_block25_0_re… (Activation)	(None, 7, 7, 1664)	0	conv5_block25_0_…
conv5_block25_1_co… (Conv2D)	(None, 7, 7, 128)	212,992	conv5_block25_0_…
conv5_block25_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block25_1_…
conv5_block25_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block25_1_…
conv5_block25_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block25_1_…
conv5_block25_conc… (Concatenate)	(None, 7, 7, 1696)	0	conv5_block24_co… conv5_block25_2_…
conv5_block26_0_bn (BatchNormalizatio…	(None, 7, 7, 1696)	6,784	conv5_block25_co…
conv5_block26_0_re… (Activation)	(None, 7, 7, 1696)	0	conv5_block26_0_…
conv5_block26_1_co… (Conv2D)	(None, 7, 7, 128)	217,088	conv5_block26_0_…
conv5_block26_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block26_1_…
conv5_block26_1_re… (Activation)	(None, 7, 7, 128)	0	conv5_block26_1_…
conv5_block26_2_co… (Conv2D)	(None, 7, 7, 32)	36,864	conv5_block26_1_…
conv5_block26_conc… (Concatenate)	(None, 7, 7, 1728)	0	conv5_block25_co… conv5_block26_2_…
conv5_block27_0_bn (BatchNormalizatio…	(None, 7, 7, 1728)	6,912	conv5_block26_co…
conv5_block27_0_re… (Activation)	(None, 7, 7, 1728)	0	conv5_block27_0_…
conv5_block27_1_co… (Conv2D)	(None, 7, 7, 128)	221,184	conv5_block27_0_…
conv5_block27_1_bn (BatchNormalizatio…	(None, 7, 7, 128)	512	conv5_block27_1_…

```
print("Number of layers in DenseNet201:", len(model2.layers))
```

```
Number of layers in DenseNet201: 708
```

```
features2 = {}
directory = os.path.join(BASE_DIR, 'Images')

for img_name in tqdm(os.listdir(directory)):

    img_path = os.path.join(directory, img_name)
    # load the image and resize it to 224x224
    image = load_img(img_path, target_size=(224, 224))
    # Convert image to a numpy array
    image = img_to_array(image)

    # Expand dimensions to match the model's expected input shape (1, 224, 224, 3)
    image = np.expand_dims(image, axis=0)

    # Preprocess the image for MobileNetV3Large
    image = preprocess_input(image)

    # Extract features from the second-to-last fully connected layer
    feature = model2.predict(image, verbose=0)

    # Get the image ID (filename without extension)
    image_id = img_name.split('.')[0]

    # Store the extracted features in the dictionary
    features2[image_id] = feature
```

```
  0%|          | 0/8091 [00:00<?, ?it/s]
```

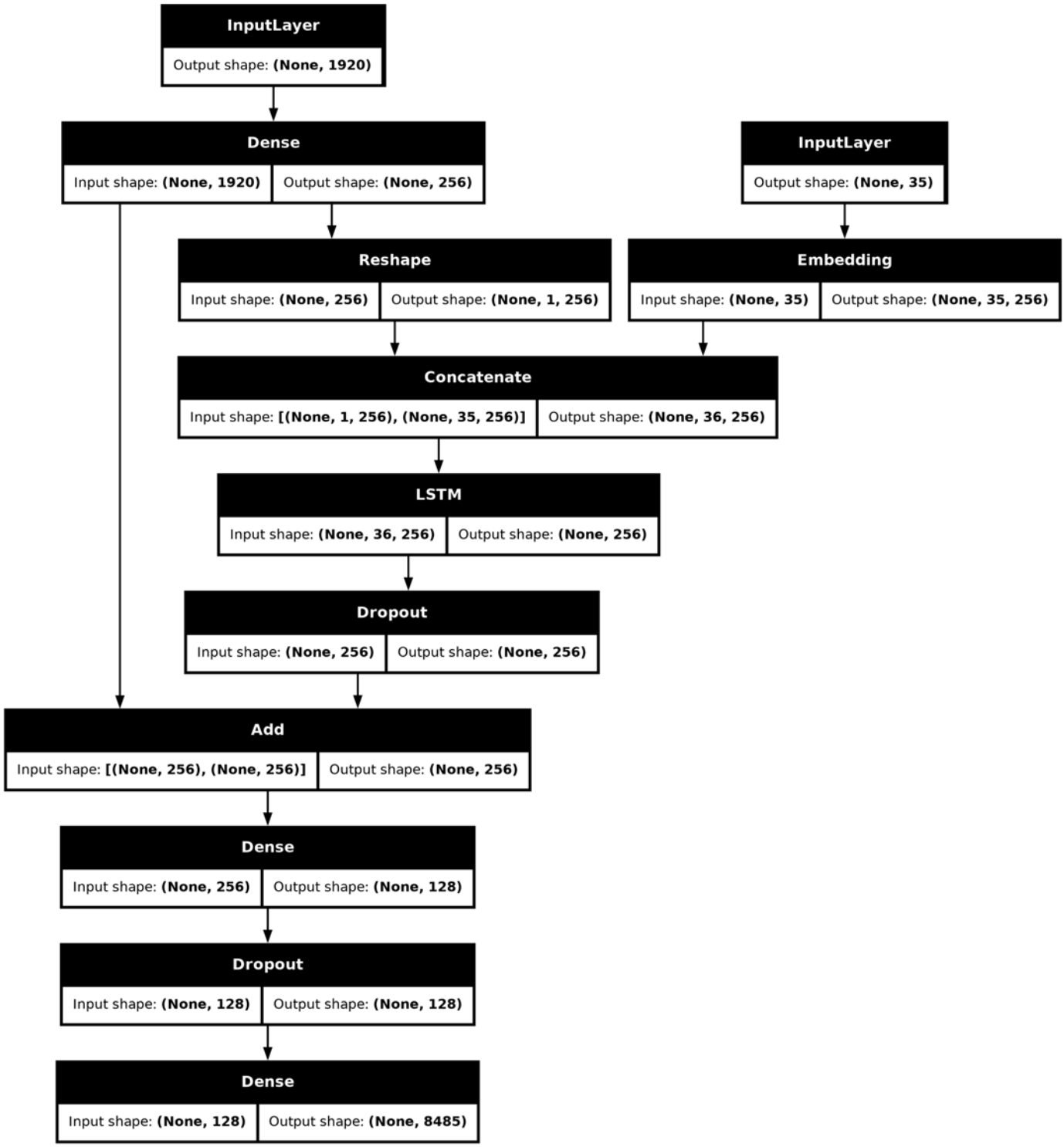
```
# Store the features
pickle.dump(features2, open(os.path.join(WORKING_DIR, 'features2.pkl'), 'wb'))
```

```
# load features from pickle
with open(os.path.join(WORKING_DIR, 'features2.pkl'), 'rb') as f:
    features2 = pickle.load(f)
```

Model Architecture and training

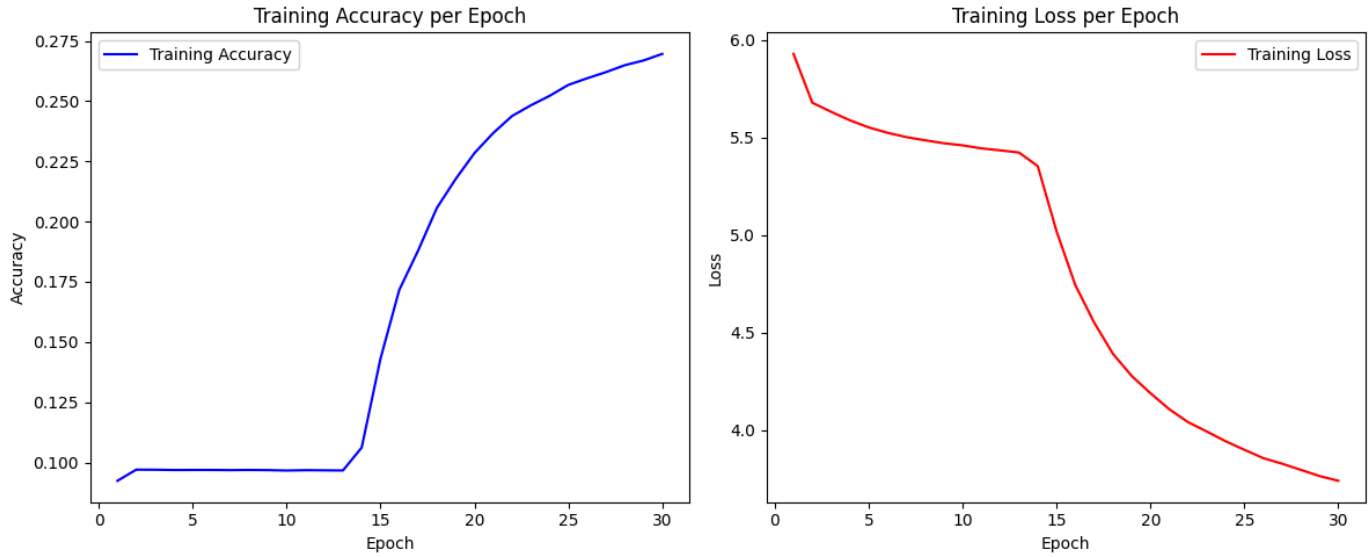
```
from tensorflow.keras.preprocessing.sequence import pad_sequences

def data_generator(data_keys, mapping, features, tokenizer, max_length, vocab_size, batch_size):
    X1, X2, y = list(), list(), list()
    n = 0

    while 1:
        for key in data_keys:
            n += 1
            captions = mapping[key]

            for caption in captions:
                seq = tokenizer.texts_to_sequences([caption])[0]

                for i in range(1, len(seq)):
                    in_seq, out_seq = seq[:i], seq[i]

                    # Ensure padding is applied properly here
                    in_seq = pad_sequences([in_seq], maxlen=max_length, padding='post')[0]
                    out_seq = to_categorical([out_seq], num_classes=vocab_size)[0]
```

```
                    X1.append(features[key][0])
                    X2.append(in_seq)
                    y.append(out_seq)

            if n == batch_size:
                yield {"image": np.array(X1), "text": np.array(X2)}, np.array(y)
                X1, X2, y = list(), list(), list()
                n = 0


# Model Architecture
input1 = Input(shape=(1920,), name='image') # Assign name 'image' to input1
img_features = Dense(256, activation='relu')(input1)
img_features_reshaped = Reshape((1, 256), input_shape=(256,))(img_features)

input2 = Input(shape=(max_length,), name='text') # Assign name 'text' to input2
sentence_features = Embedding(vocab_size, 256, mask_zero=False)(input2)
merged = concatenate([img_features_reshaped,sentence_features],axis=1)
sentence_features = LSTM(256)(merged)
x = Dropout(0.5)(sentence_features)
x = add([x, img_features])
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(vocab_size, activation='softmax')(x)

model2 = Model(inputs=[input1,input2], outputs=output)
model2.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])


plot_model(model2, show_shapes=True)
```


InputLayer
Output shape: (None, 1920)
Dense
Input shape: (None, 1920)
Output shape: (None, 256)
InputLayer
Output shape: (None, 35)
Reshape
Input shape: (None, 256)
Output shape: (None, 1, 256)
Embedding
Input shape: (None, 35)
Output shape: (None, 35, 256)
Concatenate
Input shape: [(None, 1, 256), (None, 35, 256)]
Output shape: (None, 36, 256)
LSTM
Input shape: (None, 36, 256)
Output shape: (None, 256)
Dropout
Input shape: (None, 256)
Output shape: (None, 256)
Add
Input shape: [(None, 256), (None, 256)]
Output shape: (None, 256)
Dense
Input shape: (None, 256)
Output shape: (None, 128)
Dropout
Input shape: (None, 128)
Output shape: (None, 128)
Dense
Input shape: (None, 128)
Output shape: (None, 8485)

```
# Initialize lists to store accuracy and loss values
accuracy_list2 = []
loss_list2 = []

# Model training from the data generator
# Training loop
epochs = 30
batch_size = 32
steps = len(image_ids) // batch_size

for i in range(epochs):
    generator = data_generator(image_ids, mapping, features2, tokenizer, max_length, vocab_size, batch_size)
    history = model2.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)

    # Append the accuracy and loss from this epoch
    accuracy_list2.append(history.history['accuracy'][0])
    loss_list2.append(history.history['loss'][0])

# Plotting accuracy and loss
plt.figure(figsize=(12, 5))

# Plot accuracy
plt.subplot(1, 2, 1)
plt.plot(range(1, epochs + 1), accuracy_list2, label='Training Accuracy', color='blue')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Training Accuracy per Epoch')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(range(1, epochs + 1), loss_list2, label='Training Loss', color='red')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training Loss per Epoch')
plt.legend()

plt.tight_layout()
plt.show()
```

```
252/252 ━━━━━━━━━━━━━━━━━━━━ 65s 246ms/step - accuracy: 0.0813 - loss: 6.2747
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 244ms/step - accuracy: 0.0965 - loss: 5.6733
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 245ms/step - accuracy: 0.0965 - loss: 5.6215
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 245ms/step - accuracy: 0.0963 - loss: 5.5823
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 245ms/step - accuracy: 0.0964 - loss: 5.5443
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 246ms/step - accuracy: 0.0965 - loss: 5.5154
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 246ms/step - accuracy: 0.0963 - loss: 5.4958
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 245ms/step - accuracy: 0.0965 - loss: 5.4807
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 245ms/step - accuracy: 0.0964 - loss: 5.4663
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 246ms/step - accuracy: 0.0963 - loss: 5.4576
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 253ms/step - accuracy: 0.0963 - loss: 5.4430
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 253ms/step - accuracy: 0.0963 - loss: 5.4383
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 247ms/step - accuracy: 0.0963 - loss: 5.4326
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 254ms/step - accuracy: 0.0989 - loss: 5.4031
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 247ms/step - accuracy: 0.1339 - loss: 5.1101
252/252 ━━━━━━━━━━━━━━━━━━━━ 62s 246ms/step - accuracy: 0.1677 - loss: 4.7933
252/252 ━━━━━━━━━━━━━━━━━━━━ 65s 257ms/step - accuracy: 0.1857 - loss: 4.5835
252/252 ━━━━━━━━━━━━━━━━━━━━ 64s 255ms/step - accuracy: 0.2031 - loss: 4.4124
252/252 ━━━━━━━━━━━━━━━━━━━━ 70s 277ms/step - accuracy: 0.2158 - loss: 4.2889
252/252 ━━━━━━━━━━━━━━━━━━━━ 71s 282ms/step - accuracy: 0.2282 - loss: 4.1943
252/252 ━━━━━━━━━━━━━━━━━━━━ 71s 283ms/step - accuracy: 0.2366 - loss: 4.1083
252/252 ━━━━━━━━━━━━━━━━━━━━ 77s 306ms/step - accuracy: 0.2438 - loss: 4.0387
252/252 ━━━━━━━━━━━━━━━━━━━━ 77s 304ms/step - accuracy: 0.2485 - loss: 3.9896
252/252 ━━━━━━━━━━━━━━━━━━━━ 85s 340ms/step - accuracy: 0.2525 - loss: 3.9375
252/252 ━━━━━━━━━━━━━━━━━━━━ 91s 361ms/step - accuracy: 0.2574 - loss: 3.8946
252/252 ━━━━━━━━━━━━━━━━━━━━ 99s 393ms/step - accuracy: 0.2599 - loss: 3.8506
252/252 ━━━━━━━━━━━━━━━━━━━━ 101s 399ms/step - accuracy: 0.2622 - loss: 3.8200
252/252 ━━━━━━━━━━━━━━━━━━━━ 94s 374ms/step - accuracy: 0.2656 - loss: 3.7908
252/252 ━━━━━━━━━━━━━━━━━━━━ 98s 389ms/step - accuracy: 0.2678 - loss: 3.7594
252/252 ━━━━━━━━━━━━━━━━━━━━ 96s 382ms/step - accuracy: 0.2707 - loss: 3.7379
```




```
# save the best model
model2.save('best2.keras')

model2.save('best2.h5')
```

Generating the caption

```
def idx_to_word(integer, tokenizer):
    for word, index in tokenizer.word_index.items()
```

```
            if index == integer:
                return word
        return None


    # generate caption for an image

    def predict_caption(model2, image, tokenizer, max_length):
        # add start tag for generation process
        in_text = 'startseq'
        for i in range(max_length):
            # encode input sequence
            sequence = tokenizer.texts_to_sequences([in_text])[0]
            # pad the sequence
            sequence = pad_sequences([sequence], max_length,padding='post')

            # predict next word in the sentence
            ypred = model2.predict([image, sequence], verbose=0)
            # get index with high probability of the next word
            ypred = np.argmax(ypred)
            # convert index to word (look at the function above)
            word = idx_to_word(ypred, tokenizer)
            # stop if word not found
            if word is None:
                break
            # append word as input for generating next word
            in_text += " " + word
            # stop if we reach end tag
            if word == 'endseq':
                break

        return in_text


    def generate_caption2(image_name):
        # Extract image ID
        image_id = image_name.split('.')[0]

        # Load the image file
        img_path = os.path.join(BASE_DIR, "Images", image_name)
        image = Image.open(img_path)
```