Automated Image Description Generat... Draft saved

File Edit View Run Add-ons Help

Share Save Version 2

Run All Code Draft Session (1h:35m) HDD CPU RAM GPU

```
import os
import pickle
import numpy as np
from tqdm.notebook import tqdm
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Model
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add
```

+ Code + Markdown

[5]:

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

[6]:

```
model = VGG16()

model = Model(inputs=model.inputs, outputs=model.layers[-2].output)
```

Console

Automated Image Description Generat... Draft saved

File Edit View Run Add-ons Help

Share Save Version 2

Run All Code Draft Session (1h:37m) HDD CPU RAM GPU

```
print(model.summary())
```

```
2022-09-08 07:28:21.862195: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:21.987391: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:21.988312: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:21.989625: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep
Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 AVX512F FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2022-09-08 07:28:21.989926: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:21.990635: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:21.991259: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:24.266852: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:24.267720: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:24.268371: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-09-08 07:28:24.268971: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/de
vice:GPU:0 with 15401 MB memory:  -> device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:00:04.0, compute capability: 6.0
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
553467904/553467096 [==============================] - 3s 0us/step
553476096/553467096 [==============================] - 3s 0us/step
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
```

Console

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
=================================================================
Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0
_________________________________________________________________
None
```

[7]:

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

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

    img_path = directory + '/' + img_name
    image = load_img(img_path, target_size=(224, 224))

    image = img_to_array(image)

    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))

    image = preprocess_input(image)

    feature = model.predict(image, verbose=0)

    image_id = img_name.split('.')[0]

    features[image_id] = feature
```

Loading widget...

```
2022-09-08 07:28:29.209678: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
2022-09-08 07:28:30.180525: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005
```

[8]:

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

[9]:

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

[10]:
```
with open(os.path.join(BASE_DIR, 'captions.txt'), 'r') as f:
    next(f)
    captions_doc = f.read()
```

[11]:
```
mapping = {}

for line in tqdm(captions_doc.split('\n')):

    tokens = line.split(',')
    if len(line) < 2:
        continue
    image_id, caption = tokens[0], tokens[1:]

    image_id = image_id.split('.')[0]

    caption = " ".join(caption)

    if image_id not in mapping:
        mapping[image_id] = []

    mapping[image_id].append(caption)
```

Loading widget...

[12]:
```
len(mapping)
```

[12]: 8091

[13]:
```
def clean(mapping):
    for key, captions in mapping.items():
        for i in range(len(captions)):
```

[13]:
```
def clean(mapping):
    for key, captions in mapping.items():
        for i in range(len(captions)):

         caption = captions[i]

        caption = caption.lower()

        caption = caption.replace('[^A-Za-z]', '')

        caption = caption.replace('\s+', ' ')

        caption = 'startseq ' + " ".join([word for word in caption.split() if len(word)>1]) + ' endseq'
        captions[i] = caption
```

[14]:
```
mapping['1000268201_693b08cb0e']
```

[14]:
```
['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 .']
```

[15]:
```
clean(mapping)
```

[16]:
```
mapping['1000268201_693b08cb0e']
```

[16]:
```
['A child in a pink dress is climbing up a set of stairs in an entry way .',
 'A girl going into a wooden building .',
```

[16]:
```
['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 .',
 'startseq little girl in pink dress going into wooden cabin endseq']
```

[17]:
```
all_captions = []
for key in mapping:
    for caption in mapping[key]:
        all_captions.append(caption)
```

[18]:
```
len(all_captions)
```

[18]:
```
40455
```

[19]:
```
all_captions[:10]
```

[19]:
```
['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 .',
 'startseq little girl in pink dress going into wooden cabin endseq',
 'A black dog and a spotted dog are fighting',
 'A black dog and a tri-colored dog playing with each other on the road .',
 'A black dog and a white dog with brown spots are staring at each other in the street .',
 'Two dogs of different breeds looking at each other on the road .',
 'startseq two dogs on pavement moving toward each other endseq']
```

[20]:
```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(all_captions)
vocab_size = len(tokenizer.word_index) + 1
```

[21]:
```
vocab_size
```

[21]: 8496

+ Code + Markdown

[22]:
```
max_length = max(len(caption.split()) for caption in all_captions)
max_length
```

[22]: 35

[23]:
```
image_ids = list(mapping.keys())
split = int(len(image_ids) * 0.90)
train = image_ids[:split]
test = image_ids[split:]
```

[24]:
```
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)):
```

```
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]

                in_seq = pad_sequences([in_seq], maxlen=max_length) [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:
                    X1, X2, y = np.array(X1), np.array(X2), np.array(y)
                    yield [X1, X2], y
                    X1, X2, y = list(), list(), list()
                    n = 0
```

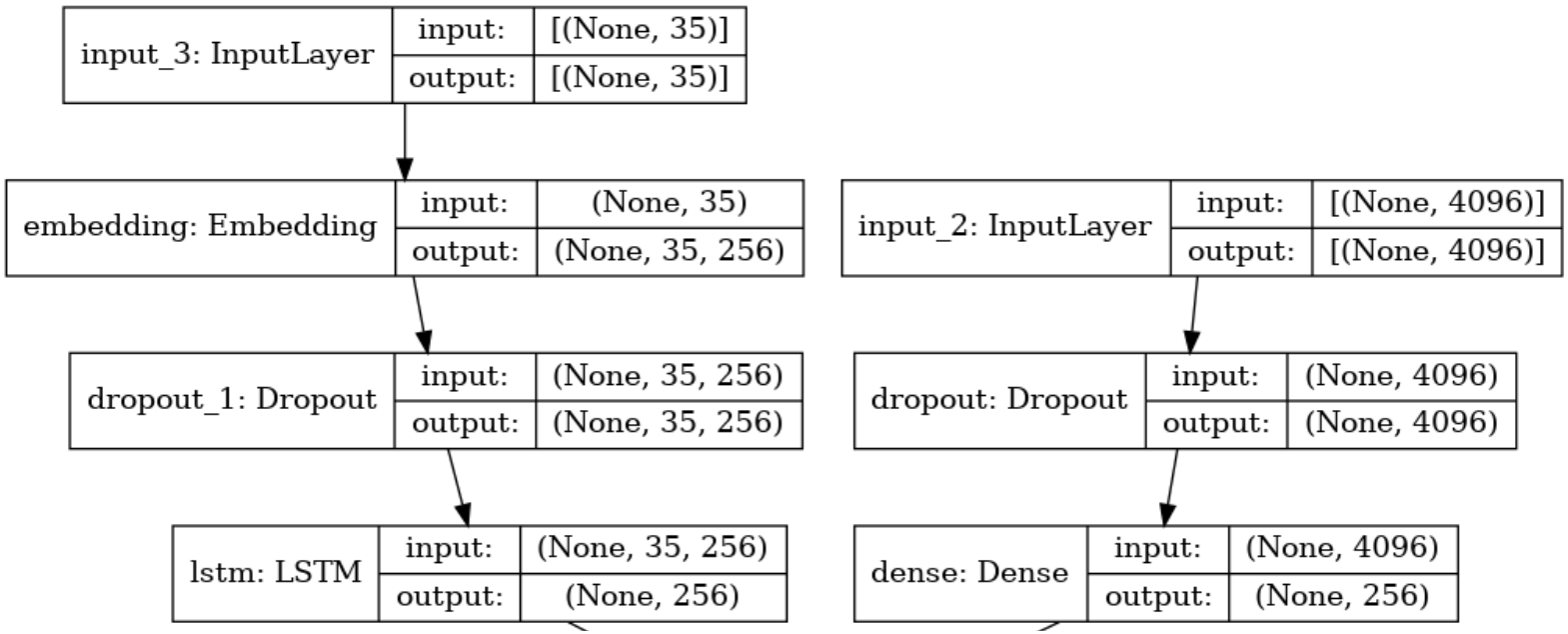
[25]:

```
inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.4)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)

inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.4)(se1)
```

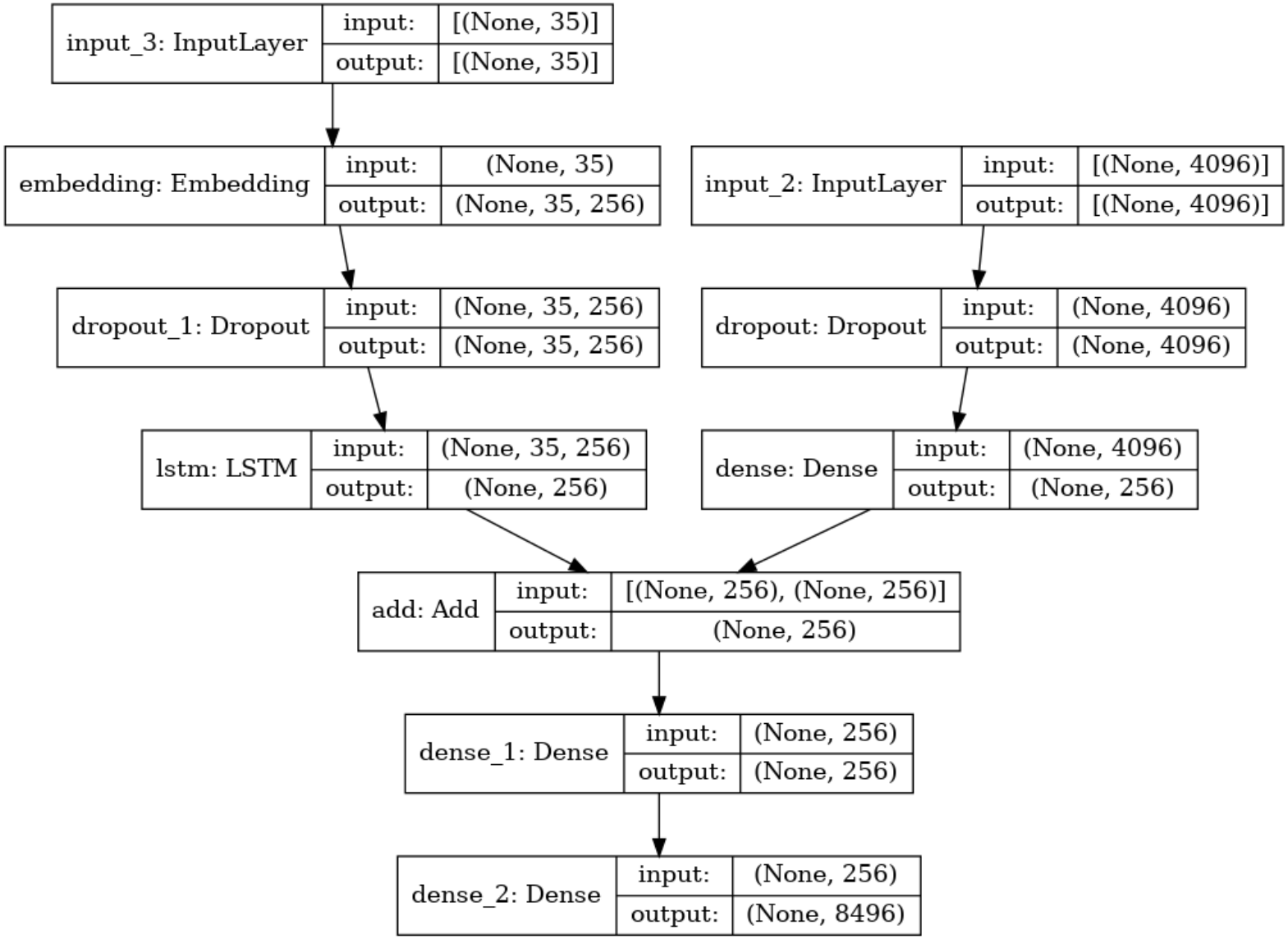
[25]:

```
inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.4)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)

inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.4)(se1)
se3 = LSTM(256)(se2)


decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
outputs = Dense(vocab_size, activation='softmax')(decoder2)

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


plot_model(model, show_shapes=True)
```

[25]:



[25]:


input_3: InputLayer
input: [(None, 35)]
output: [(None, 35)]
embedding: Embedding
input: (None, 35)
output: (None, 35, 256)
input_2: InputLayer
input: [(None, 4096)]
output: [(None, 4096)]
dropout_1: Dropout
input: (None, 35, 256)
output: (None, 35, 256)
dropout: Dropout
input: (None, 4096)
output: (None, 4096)
lstm: LSTM
input: (None, 35, 256)
output: (None, 256)
dense: Dense
input: (None, 4096)
output: (None, 256)
add: Add
input: [(None, 256), (None, 256)]
output: (None, 256)
dense_1: Dense
input: (None, 256)
output: (None, 256)
dense_2: Dense
input: (None, 256)
output: (None, 8496)

[26]:

```
epochs = 20
batch_size = 32
steps = len(train) // batch_size

for i in range(epochs):

    generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch_size)

    model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)
```

```
227/227 [==============================] - 66s 275ms/step - loss: 5.2801
227/227 [==============================] - 59s 258ms/step - loss: 3.9776
227/227 [==============================] - 59s 257ms/step - loss: 3.5063
227/227 [==============================] - 60s 262ms/step - loss: 3.2249
227/227 [==============================] - 60s 262ms/step - loss: 3.0192
227/227 [==============================] - 62s 272ms/step - loss: 2.8675
227/227 [==============================] - 60s 266ms/step - loss: 2.7504
227/227 [==============================] - 61s 268ms/step - loss: 2.6537
227/227 [==============================] - 61s 269ms/step - loss: 2.5675
227/227 [==============================] - 61s 269ms/step - loss: 2.4947
227/227 [==============================] - 61s 268ms/step - loss: 2.4321
227/227 [==============================] - 61s 267ms/step - loss: 2.3756
227/227 [==============================] - 60s 263ms/step - loss: 2.3294
227/227 [==============================] - 62s 273ms/step - loss: 2.2835
227/227 [==============================] - 62s 270ms/step - loss: 2.2398
227/227 [==============================] - 61s 269ms/step - loss: 2.1960
227/227 [==============================] - 58s 255ms/step - loss: 2.1586
227/227 [==============================] - 62s 274ms/step - loss: 2.1264
227/227 [==============================] - 62s 270ms/step - loss: 2.0964
227/227 [==============================] - 63s 276ms/step - loss: 2.0669
```

[27]:

```
model.save(WORKING_DIR+'/best_model.h5')
```

```
/opt/conda/lib/python3.7/site-packages/keras/utils/generic_utils.py:497: CustomMaskWarning: Custom mask layers require a config and must override get_config. When loading, the
custom mask layer must be passed to the custom_objects argument.
  category=CustomMaskWarning)
```

[28]:

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

[29]:

```
def predict_caption(model, image, tokenizer, max_length):

    in_text = 'startseq'

    for i in range(max_length):

        sequence = tokenizer.texts_to_sequences([in_text])[0]

        sequence = pad_sequences([sequence], max_length)

        yhat = model.predict([image, sequence], verbose=0)

        yhat = np.argmax(yhat)

        word = idx_to_word(yhat, tokenizer)

        if word is None:
            break

        in_text += " " + word

        if word == 'endseq':
            break
    return in_text
```

[30]:

```
from nltk.translate.bleu_score import corpus_bleu

actual, predicted = list(), list()

for key in tqdm(test):

    captions = mapping[key]

    y_pred = predict_caption(model, features[key], tokenizer, max_length)

    actual_captions = [caption.split() for caption in captions]
    y_pred = y_pred.split()
```

[30]:

```
from nltk.translate.bleu_score import corpus_bleu

actual, predicted = list(), list()

for key in tqdm(test):

    captions = mapping[key]

    y_pred = predict_caption(model, features[key], tokenizer, max_length)

    actual_captions = [caption.split() for caption in captions]
    y_pred = y_pred.split()

    actual.append(actual_captions)
    predicted.append(y_pred)

    print("BLEU-1: %f" % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
    print("BLEU-2: %f" % corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
```

Loading widget...

```
/opt/conda/lib/python3.7/site-packages/nltk/translate/bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/opt/conda/lib/python3.7/site-packages/nltk/translate/bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/opt/conda/lib/python3.7/site-packages/nltk/translate/bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
BLEU-1: 0.444444
BLEU-2: 0.000000
BLEU-1: 0.450000
BLEU-2: 0.158114
BLEU-1: 0.416500
BLEU-2: 0.194800
BLEU-1: 0.428480
BLEU-2: 0.195319
```

```
BLEU-1: 0.534344
BLEU-2: 0.286435
BLEU-1: 0.533040
BLEU-2: 0.286089
BLEU-1: 0.533275
BLEU-2: 0.285321
BLEU-1: 0.534722
BLEU-2: 0.285373
BLEU-1: 0.531732
BLEU-2: 0.282818
BLEU-1: 0.531117
BLEU-2: 0.281856
BLEU-1: 0.531646
BLEU-2: 0.283183
BLEU-1: 0.533501
BLEU-2: 0.284365
BLEU-1: 0.532835
BLEU-2: 0.284004
BLEU-1: 0.533443
BLEU-2: 0.283258
BLEU-1: 0.532298
BLEU-2: 0.281546
BLEU-1: 0.530364
BLEU-2: 0.279650
BLEU-1: 0.531727
BLEU-2: 0.281413
BLEU-1: 0.531002
BLEU-2: 0.279737
BLEU-1: 0.529042
BLEU-2: 0.278226
BLEU-1: 0.528727
BLEU-2: 0.277408
BLEU-1: 0.528868
BLEU-2: 0.277087
BLEU-1: 0.527438
BLEU-2: 0.275324
BLEU-1: 0.527273
BLEU-2: 0.274453
BLEU-1: 0.526316
BLEU-2: 0.273179
BLEU-1: 0.526158
BLEU-2: 0.272349
BLEU-1: 0.526746
BLEU-2: 0.272512
BLEU-1: 0.525037
BLEU-2: 0.270851
BLEU-1: 0.524470
BLEU-2: 0.271175
BLEU-1: 0.521300
BLEU-2: 0.268735
BLEU-1: 0.521117
BLEU-2: 0.269815
BLEU-1: 0.522760
BLEU-2: 0.270908
BLEU-1: 0.520760
```

```
BLEU-1: 0.518462
BLEU-2: 0.271548
BLEU-1: 0.516392
BLEU-2: 0.269938
BLEU-1: 0.516590
BLEU-2: 0.269738
BLEU-1: 0.516159
BLEU-2: 0.268673
BLEU-1: 0.515425
BLEU-2: 0.267649
BLEU-1: 0.516233
BLEU-2: 0.268758
BLEU-1: 0.516771
BLEU-2: 0.270235
BLEU-1: 0.516668
BLEU-2: 0.270031
BLEU-1: 0.517174
BLEU-2: 0.270652
BLEU-1: 0.516774
BLEU-2: 0.269817
BLEU-1: 0.518809
BLEU-2: 0.270266
BLEU-1: 0.518094
BLEU-2: 0.269261
BLEU-1: 0.517022
BLEU-2: 0.268807
BLEU-1: 0.517780
BLEU-2: 0.269836
BLEU-1: 0.518786
BLEU-2: 0.269414
BLEU-1: 0.520174
BLEU-2: 0.269867
BLEU-1: 0.519709
BLEU-2: 0.269770
BLEU-1: 0.521051
BLEU-2: 0.271262
BLEU-1: 0.521481
BLEU-2: 0.271030
BLEU-1: 0.519013
BLEU-2: 0.269123
BLEU-1: 0.520334
BLEU-2: 0.269373
BLEU-1: 0.521905
BLEU-2: 0.269627
BLEU-1: 0.521512
BLEU-2: 0.268875
BLEU-1: 0.520258
BLEU-2: 0.267647
BLEU-1: 0.520700
BLEU-2: 0.267611
BLEU-1: 0.522547
BLEU-2: 0.269361
BLEU-1: 0.523257
BLEU-2: 0.269479
BLEU-1: 0.522796
```

```
BLEU-1: 0.523181
BLEU-2: 0.270366
BLEU-1: 0.522356
BLEU-2: 0.269844
BLEU-1: 0.521570
BLEU-2: 0.269168
BLEU-1: 0.521494
BLEU-2: 0.269430
BLEU-1: 0.521236
BLEU-2: 0.268843
BLEU-1: 0.521709
BLEU-2: 0.268654
BLEU-1: 0.521466
BLEU-2: 0.268178
BLEU-1: 0.520153
BLEU-2: 0.267225
BLEU-1: 0.521137
BLEU-2: 0.268604
BLEU-1: 0.522445
BLEU-2: 0.270323
BLEU-1: 0.522876
BLEU-2: 0.271020
BLEU-1: 0.522603
BLEU-2: 0.270700
BLEU-1: 0.522541
BLEU-2: 0.270691
BLEU-1: 0.522805
BLEU-2: 0.271387
BLEU-1: 0.523567
BLEU-2: 0.271798
BLEU-1: 0.522459
BLEU-2: 0.270954
BLEU-1: 0.522002
BLEU-2: 0.270797
BLEU-1: 0.521594
BLEU-2: 0.270242
BLEU-1: 0.521855
BLEU-2: 0.270218
BLEU-1: 0.522086
BLEU-2: 0.269988
BLEU-1: 0.521466
BLEU-2: 0.269394
BLEU-1: 0.522712
BLEU-2: 0.269627
BLEU-1: 0.523309
BLEU-2: 0.270141
BLEU-1: 0.523404
BLEU-2: 0.269776
BLEU-1: 0.523157
BLEU-2: 0.269576
BLEU-1: 0.523577
BLEU-2: 0.270339
BLEU-1: 0.522668
BLEU-2: 0.269527
BLEU-1: 0.522272
```

```
BLEU-1: 0.515670
BLEU-2: 0.261098
```

+ Code + Markdown

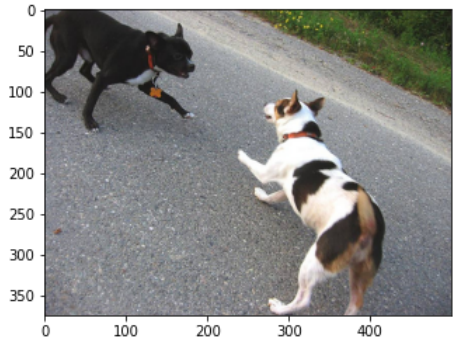
[31]:

```
from PIL import Image
import matplotlib.pyplot as plt
def generate_caption(image_name):

    image_id = image_name.split('.')[0]
    img_path = os.path.join(BASE_DIR, "Images", image_name)
    image = Image.open(img_path)
    captions = mapping[image_id]
    print('---------------------Actual---------------------')
    for caption in captions:
        print(caption)

    y_pred = predict_caption(model, features[image_id], tokenizer, max_length)
    print('--------------------Predicted--------------------')
    print(y_pred)
    plt.imshow(image)
```

[32]:

```
generate_caption("1001773457_577c3a7d70.jpg")
```

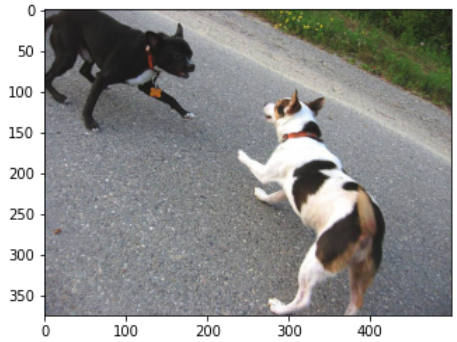
```
---------------------Actual---------------------
A black dog and a spotted dog are fighting
A black dog and a tri-colored dog playing with each other on the road .
A black dog and a white dog with brown spots are staring at each other in the street .
Two dogs of different breeds looking at each other on the road .
startseq two dogs on pavement moving toward each other endseq
--------------------Predicted--------------------
startseq two dogs play with each other on the grass endseq
```



Console

[32]:
```
generate_caption("1001773457_577c3a7d70.jpg")
```

```
--------------------Actual--------------------
A black dog and a spotted dog are fighting
A black dog and a tri-colored dog playing with each other on the road .
A black dog and a white dog with brown spots are staring at each other in the street .
Two dogs of different breeds looking at each other on the road .
startseq two dogs on pavement moving toward each other endseq
--------------------Predicted--------------------
startseq two dogs play with each other on the grass endseq
```



[33]:
```
generate_caption("1002674143_1b742ab4b8.jpg")
```

```
--------------------Actual--------------------
A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
A little girl is sitting in front of a large painted rainbow .
A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
There is a girl with pigtails sitting in front of a rainbow painting .
startseq young girl with pigtails painting outside in the grass endseq
--------------------Predicted--------------------
startseq young girl with pigtails painting in the air painting wave endseq
```

[33]:
```
generate_caption("1002674143_1b742ab4b8.jpg")
```

```
--------------------Actual--------------------
A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
A little girl is sitting in front of a large painted rainbow .
A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
There is a girl with pigtails sitting in front of a rainbow painting .
startseq young girl with pigtails painting outside in the grass endseq
--------------------Predicted--------------------
startseq young girl with pigtails painting in the air painting wave endseq
```

+ Code + Markdown

[34]:
```
generate_caption("101669240_b2d3e7f17b.jpg")
```

```
--------------------Actual--------------------
A man in a hat is displaying pictures next to a skier in a blue hat .
A man skis past another man displaying paintings in the snow .
A person wearing skis looking at framed pictures set up in the snow .
A skier looks at framed pictures in the snow next to trees .
startseq man on skis looking at artwork for sale in the snow endseq
--------------------Predicted--------------------
startseq the skier is displaying pictures in the snow endseq
```

startseq young girl with pigtails painting outside in the grass endseq
--------------------Predicted--------------------
startseq young girl with pigtails painting in the air painting wave endseq

[34]:
```
generate_caption("101669240_b2d3e7f17b.jpg")
```

--------------------Actual--------------------
A man in a hat is displaying pictures next to a skier in a blue hat .
A man skis past another man displaying paintings in the snow .
A person wearing skis looking at framed pictures set up in the snow .
A skier looks at framed pictures in the snow next to trees .
startseq man on skis looking at artwork for sale in the snow endseq
--------------------Predicted--------------------
startseq the skier is displaying pictures in the snow endseq