

```
In [1]: import cv2  
from glob import glob  
import matplotlib.pyplot as plt  
from os import listdir
```

```
In [2]: #First step is to read images from the Dataset. In this project i am using Flickr-8k Data set  
#In Flickr-8k data set 6000 images are for training,1000 images are for validation and 1000 images are for testing
```

```
In [3]: images_data=glob('C:/Users/mahid/M.Tech Project/Data Set/Images/*')
```

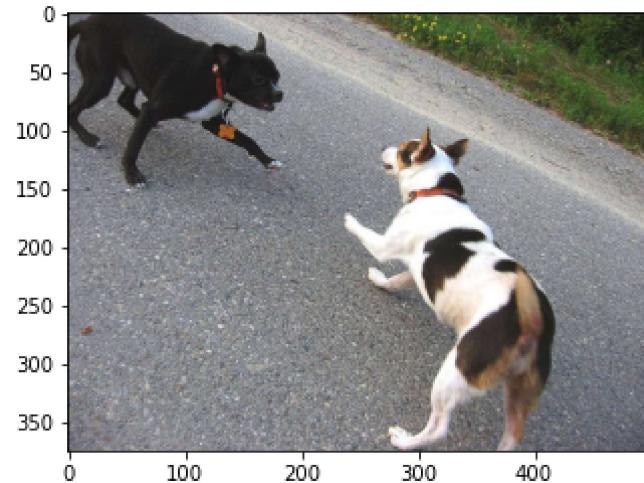
```
In [4]: count=0  
for i in images_data:  
    if(count<10):  
        print(i)  
    count+=1
```

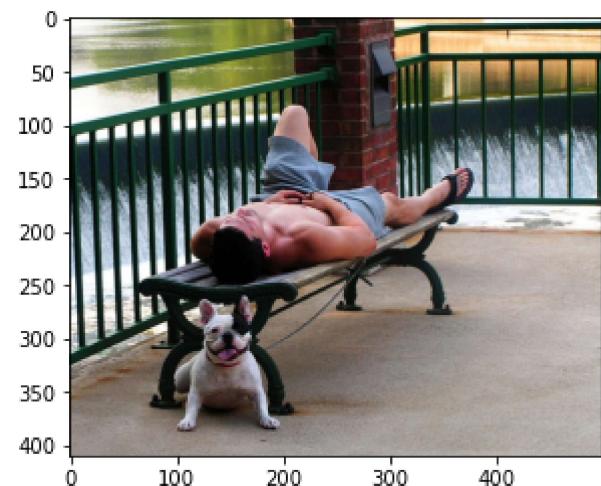
```
C:/Users/mahid/M.Tech Project/Data Set/Images\1000268201_693b08cb0e.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1001773457_577c3a7d70.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1002674143_1b742ab4b8.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1003163366_44323f5815.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1007129816_e794419615.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1007320043_627395c3d8.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1009434119_febe49276a.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1012212859_01547e3f17.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1015118661_980735411b.jpg  
C:/Users/mahid/M.Tech Project/Data Set/Images\1015584366_dfcec3c85a.jpg
```

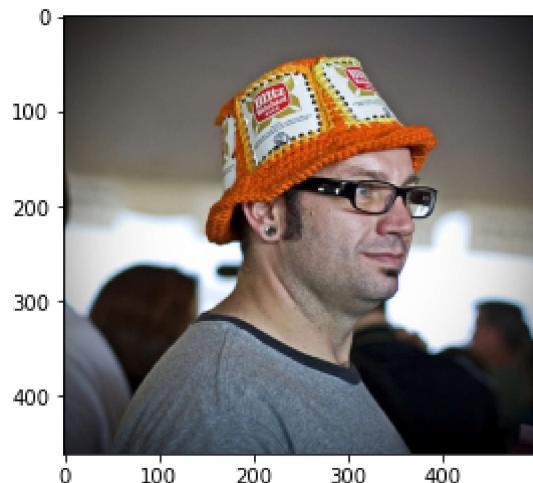
```
In [5]: image_shape=cv2.imread(images_data[0])  
print(image_shape.shape)
```

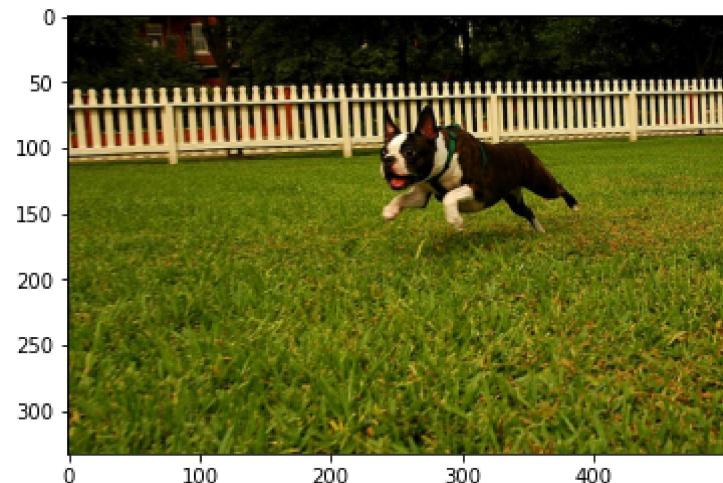
```
(500, 375, 3)
```

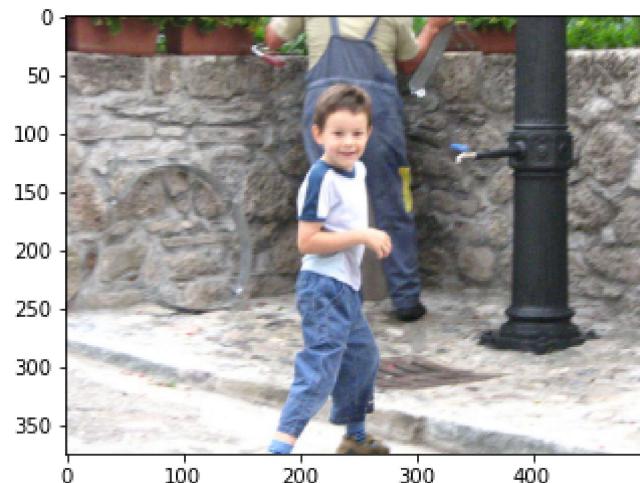
```
In [6]: for i in range(10):  
    image=cv2.imread(images_data[i])  
    image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)  
    plt.figure()  
    plt.imshow(image)
```











```
In [7]: #Now we need to pre process the text data as well but giving unique numbers to each unique word in data  
#Because Neural networks can work only on numerical values
```

```
In [8]: captions_data_path = 'C:/Users/mahid/M.Tech Project/Data Set/Flickr8k_text/Flickr8k.token.txt'
```

```
In [9]: #In captions document the format is Image name #no Caption of that Image name  
#for each image there are 5 different captions indicating similar meaning to the image  
#EX: 1000268201_693b08cb0e.jpg#0 A child in a pink dress is climbing up a set of stairs in an entry way .
```

```
In [10]: captions_data = open(captions_data_path, 'rb').read()
captions_data=captions_data.decode('utf-8').split('\n')
```

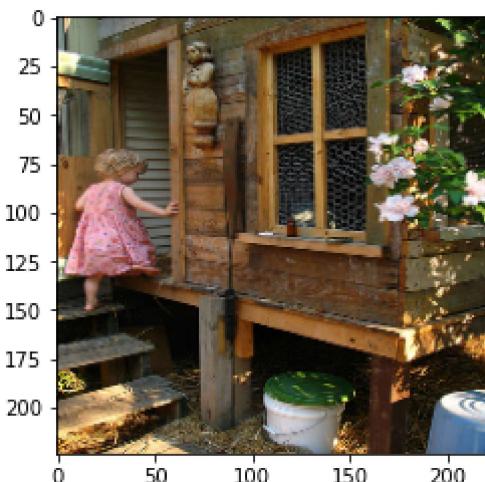
```
In [11]: len(captions_data)
```

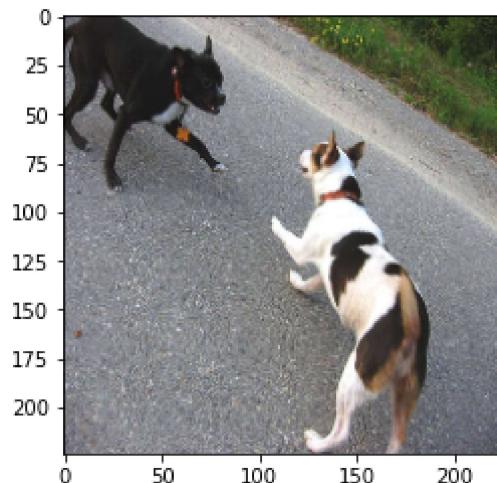
```
Out[11]: 40456
```

```
In [12]: captions_data[0]
```

```
Out[12]: '1000268201_693b08cb0e.jpg#0\tA child in a pink dress is climbing up a set of stairs in an entry way .'
```

```
In [13]: for i in range(2):
    image=cv2.imread(images_data[i])
    image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
    image=cv2.resize(image,(224,224))
    image=image.reshape(224,224,3)
    plt.figure()
    plt.imshow(image)
```





```
In [14]:  
for i in range(10):  
    print(captions_data[i])
```

1000268201_693b08cb0e.jpg#0	A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg#1	A girl going into a wooden building .
1000268201_693b08cb0e.jpg#2	A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg#3	A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg#4	A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg#0	A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg#1	A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg#2	A black dog and a white dog with brown spots are staring at each other in the street .
1001773457_577c3a7d70.jpg#3	Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg#4	Two dogs on pavement moving toward each other .

```
In [15]:  
from pickle import dump  
from tensorflow.keras.applications import ResNet50  
from keras.models import Model
```

```
In [16]:  
model=ResNet50(include_top=True,weights='imagenet')  
lastlayer=model.layers[-2].output  
model=Model(inputs=model.input,outputs=lastlayer)  
print(model.summary())
```

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
<hr/>			
input_1 (InputLayer)	[(None, 224, 224, 3 0)]		[]
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3) 0		['input_1[0][0]']
conv1_conv (Conv2D)	(None, 112, 112, 64 9472)		['conv1_pad[0][0]']
conv1_bn (BatchNormalization)	(None, 112, 112, 64 256)		['conv1_conv[0][0]']
conv1_relu (Activation)	(None, 112, 112, 64 0)		['conv1_bn[0][0]']
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64 0)		['conv1_relu[0][0]']
pool1_pool (MaxPooling2D)	(None, 56, 56, 64) 0		['pool1_pad[0][0]']
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64) 4160		['pool1_pool[0][0]']
conv2_block1_1_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block1_1_conv[0][0]']
conv2_block1_1_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block1_1_bn[0][0]']
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64) 36928		['conv2_block1_1_relu[0][0]']
conv2_block1_2_bn (BatchNormal ization)	(None, 56, 56, 64) 256		['conv2_block1_2_conv[0][0]']
conv2_block1_2_relu (Activatio n)	(None, 56, 56, 64) 0		['conv2_block1_2_bn[0][0]']
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256) 16640		['pool1_pool[0][0]']
conv2_block1_3_conv (Conv2D)	(None, 56, 56, 256) 16640		['conv2_block1_2_relu[0][0]']
conv2_block1_0_bn (BatchNormal ization)	(None, 56, 56, 256) 1024		['conv2_block1_0_conv[0][0]']

conv2_block1_3_bn (BatchNormal ization)	(None, 56, 56, 256)	1024	['conv2_block1_3_conv[0][0]']
conv2_block1_add (Add)	(None, 56, 56, 256)	0	['conv2_block1_0_bn[0][0]', 'conv2_block1_3_bn[0][0]']
conv2_block1_out (Activation)	(None, 56, 56, 256)	0	['conv2_block1_add[0][0]']
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 64)	16448	['conv2_block1_out[0][0]']
conv2_block2_1_bn (BatchNormal ization)	(None, 56, 56, 64)	256	['conv2_block2_1_conv[0][0]']
conv2_block2_1_relu (Activatio n)	(None, 56, 56, 64)	0	['conv2_block2_1_bn[0][0]']
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 64)	36928	['conv2_block2_1_relu[0][0]']
conv2_block2_2_bn (BatchNormal ization)	(None, 56, 56, 64)	256	['conv2_block2_2_conv[0][0]']
conv2_block2_2_relu (Activatio n)	(None, 56, 56, 64)	0	['conv2_block2_2_bn[0][0]']
conv2_block2_3_conv (Conv2D)	(None, 56, 56, 256)	16640	['conv2_block2_2_relu[0][0]']
conv2_block2_3_bn (BatchNormal ization)	(None, 56, 56, 256)	1024	['conv2_block2_3_conv[0][0]']
conv2_block2_add (Add)	(None, 56, 56, 256)	0	['conv2_block1_out[0][0]', 'conv2_block2_3_bn[0][0]']
conv2_block2_out (Activation)	(None, 56, 56, 256)	0	['conv2_block2_add[0][0]']
conv2_block3_1_conv (Conv2D)	(None, 56, 56, 64)	16448	['conv2_block2_out[0][0]']
conv2_block3_1_bn (BatchNormal ization)	(None, 56, 56, 64)	256	['conv2_block3_1_conv[0][0]']
conv2_block3_1_relu (Activatio n)	(None, 56, 56, 64)	0	['conv2_block3_1_bn[0][0]']
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 64)	36928	['conv2_block3_1_relu[0][0]']

conv2_block3_2_bn (BatchNormal ization)	(None, 56, 56, 64) 256	['conv2_block3_2_conv[0][0]']
conv2_block3_2_relu (Activatio n)	(None, 56, 56, 64) 0	['conv2_block3_2_bn[0][0]']
conv2_block3_3_conv (Conv2D)	(None, 56, 56, 256) 16640	['conv2_block3_2_relu[0][0]']
conv2_block3_3_bn (BatchNormal ization)	(None, 56, 56, 256) 1024	['conv2_block3_3_conv[0][0]']
conv2_block3_add (Add)	(None, 56, 56, 256) 0	['conv2_block2_out[0][0]', 'conv2_block3_3_bn[0][0]']
conv2_block3_out (Activation)	(None, 56, 56, 256) 0	['conv2_block3_add[0][0]']
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128) 32896	['conv2_block3_out[0][0]']
conv3_block1_1_bn (BatchNormal ization)	(None, 28, 28, 128) 512	['conv3_block1_1_conv[0][0]']
conv3_block1_1_relu (Activatio n)	(None, 28, 28, 128) 0	['conv3_block1_1_bn[0][0]']
conv3_block1_2_conv (Conv2D)	(None, 28, 28, 128) 147584	['conv3_block1_1_relu[0][0]']
conv3_block1_2_bn (BatchNormal ization)	(None, 28, 28, 128) 512	['conv3_block1_2_conv[0][0]']
conv3_block1_2_relu (Activatio n)	(None, 28, 28, 128) 0	['conv3_block1_2_bn[0][0]']
conv3_block1_0_conv (Conv2D)	(None, 28, 28, 512) 131584	['conv2_block3_out[0][0]']
conv3_block1_3_conv (Conv2D)	(None, 28, 28, 512) 66048	['conv3_block1_2_relu[0][0]']
conv3_block1_0_bn (BatchNormal ization)	(None, 28, 28, 512) 2048	['conv3_block1_0_conv[0][0]']
conv3_block1_3_bn (BatchNormal ization)	(None, 28, 28, 512) 2048	['conv3_block1_3_conv[0][0]']
conv3_block1_add (Add)	(None, 28, 28, 512) 0	['conv3_block1_0_bn[0][0]', 'conv3_block1_3_bn[0][0]']

conv3_block1_out (Activation)	(None, 28, 28, 512)	0	['conv3_block1_add[0][0]']
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	65664	['conv3_block1_out[0][0]']
conv3_block2_1_bn (BatchNormal ization)	(None, 28, 28, 128)	512	['conv3_block2_1_conv[0][0]']
conv3_block2_1_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block2_1_bn[0][0]']
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 128)	147584	['conv3_block2_1_relu[0][0]']
conv3_block2_2_bn (BatchNormal ization)	(None, 28, 28, 128)	512	['conv3_block2_2_conv[0][0]']
conv3_block2_2_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block2_2_bn[0][0]']
conv3_block2_3_conv (Conv2D)	(None, 28, 28, 512)	66048	['conv3_block2_2_relu[0][0]']
conv3_block2_3_bn (BatchNormal ization)	(None, 28, 28, 512)	2048	['conv3_block2_3_conv[0][0]']
conv3_block2_add (Add)	(None, 28, 28, 512)	0	['conv3_block1_out[0][0]', 'conv3_block2_3_bn[0][0]']
conv3_block2_out (Activation)	(None, 28, 28, 512)	0	['conv3_block2_add[0][0]']
conv3_block3_1_conv (Conv2D)	(None, 28, 28, 128)	65664	['conv3_block2_out[0][0]']
conv3_block3_1_bn (BatchNormal ization)	(None, 28, 28, 128)	512	['conv3_block3_1_conv[0][0]']
conv3_block3_1_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block3_1_bn[0][0]']
conv3_block3_2_conv (Conv2D)	(None, 28, 28, 128)	147584	['conv3_block3_1_relu[0][0]']
conv3_block3_2_bn (BatchNormal ization)	(None, 28, 28, 128)	512	['conv3_block3_2_conv[0][0]']
conv3_block3_2_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block3_2_bn[0][0]']

conv3_block3_3_conv (Conv2D)	(None, 28, 28, 512)	66048	['conv3_block3_2_relu[0][0]']
conv3_block3_3_bn (BatchNormal ization)	(None, 28, 28, 512)	2048	['conv3_block3_3_conv[0][0]']
conv3_block3_add (Add)	(None, 28, 28, 512)	0	['conv3_block2_out[0][0]', 'conv3_block3_3_bn[0][0]']
conv3_block3_out (Activation)	(None, 28, 28, 512)	0	['conv3_block3_add[0][0]']
conv3_block4_1_conv (Conv2D)	(None, 28, 28, 128)	65664	['conv3_block3_out[0][0]']
conv3_block4_1_bn (BatchNormal ization)	(None, 28, 28, 128)	512	['conv3_block4_1_conv[0][0]']
conv3_block4_1_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block4_1_bn[0][0]']
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 128)	147584	['conv3_block4_1_relu[0][0]']
conv3_block4_2_bn (BatchNormal ization)	(None, 28, 28, 128)	512	['conv3_block4_2_conv[0][0]']
conv3_block4_2_relu (Activatio n)	(None, 28, 28, 128)	0	['conv3_block4_2_bn[0][0]']
conv3_block4_3_conv (Conv2D)	(None, 28, 28, 512)	66048	['conv3_block4_2_relu[0][0]']
conv3_block4_3_bn (BatchNormal ization)	(None, 28, 28, 512)	2048	['conv3_block4_3_conv[0][0]']
conv3_block4_add (Add)	(None, 28, 28, 512)	0	['conv3_block3_out[0][0]', 'conv3_block4_3_bn[0][0]']
conv3_block4_out (Activation)	(None, 28, 28, 512)	0	['conv3_block4_add[0][0]']
conv4_block1_1_conv (Conv2D)	(None, 14, 14, 256)	131328	['conv3_block4_out[0][0]']
conv4_block1_1_bn (BatchNormal ization)	(None, 14, 14, 256)	1024	['conv4_block1_1_conv[0][0]']
conv4_block1_1_relu (Activatio n)	(None, 14, 14, 256)	0	['conv4_block1_1_bn[0][0]']

conv4_block1_2_conv (Conv2D)	(None, 14, 14, 256)	590080	['conv4_block1_1_relu[0][0]']
conv4_block1_2_bn (BatchNormal ization)	(None, 14, 14, 256)	1024	['conv4_block1_2_conv[0][0]']
conv4_block1_2_relu (Activatio n)	(None, 14, 14, 256)	0	['conv4_block1_2_bn[0][0]']
conv4_block1_0_conv (Conv2D)	(None, 14, 14, 1024)	525312	['conv3_block4_out[0][0]']
conv4_block1_3_conv (Conv2D)	(None, 14, 14, 1024)	263168	['conv4_block1_2_relu[0][0]']
conv4_block1_0_bn (BatchNormal ization)	(None, 14, 14, 1024)	4096	['conv4_block1_0_conv[0][0]']
conv4_block1_3_bn (BatchNormal ization)	(None, 14, 14, 1024)	4096	['conv4_block1_3_conv[0][0]']
conv4_block1_add (Add)	(None, 14, 14, 1024 0)	0	['conv4_block1_0_bn[0][0]', 'conv4_block1_3_bn[0][0]']
conv4_block1_out (Activation)	(None, 14, 14, 1024 0)	0	['conv4_block1_add[0][0]']
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 256)	262400	['conv4_block1_out[0][0]']
conv4_block2_1_bn (BatchNormal ization)	(None, 14, 14, 256)	1024	['conv4_block2_1_conv[0][0]']
conv4_block2_1_relu (Activatio n)	(None, 14, 14, 256)	0	['conv4_block2_1_bn[0][0]']
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 256)	590080	['conv4_block2_1_relu[0][0]']
conv4_block2_2_bn (BatchNormal ization)	(None, 14, 14, 256)	1024	['conv4_block2_2_conv[0][0]']
conv4_block2_2_relu (Activatio n)	(None, 14, 14, 256)	0	['conv4_block2_2_bn[0][0]']
conv4_block2_3_conv (Conv2D)	(None, 14, 14, 1024 263168)	263168	['conv4_block2_2_relu[0][0]']

)	
conv4_block2_3_bn (BatchNormal ization)	(None, 14, 14, 1024 4096)	['conv4_block2_3_conv[0][0]']
conv4_block2_add (Add)	(None, 14, 14, 1024 0)	['conv4_block1_out[0][0]', 'conv4_block2_3_bn[0][0]']
conv4_block2_out (Activation)	(None, 14, 14, 1024 0)	['conv4_block2_add[0][0]']
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 256) 262400	['conv4_block2_out[0][0]']
conv4_block3_1_bn (BatchNormal ization)	(None, 14, 14, 256) 1024	['conv4_block3_1_conv[0][0]']
conv4_block3_1_relu (Activatio n)	(None, 14, 14, 256) 0	['conv4_block3_1_bn[0][0]']
conv4_block3_2_conv (Conv2D)	(None, 14, 14, 256) 590080	['conv4_block3_1_relu[0][0]']
conv4_block3_2_bn (BatchNormal ization)	(None, 14, 14, 256) 1024	['conv4_block3_2_conv[0][0]']
conv4_block3_2_relu (Activatio n)	(None, 14, 14, 256) 0	['conv4_block3_2_bn[0][0]']
conv4_block3_3_conv (Conv2D)	(None, 14, 14, 1024 263168)	['conv4_block3_2_relu[0][0]']
conv4_block3_3_bn (BatchNormal ization)	(None, 14, 14, 1024 4096)	['conv4_block3_3_conv[0][0]']
conv4_block3_add (Add)	(None, 14, 14, 1024 0)	['conv4_block2_out[0][0]', 'conv4_block3_3_bn[0][0]']
conv4_block3_out (Activation)	(None, 14, 14, 1024 0)	['conv4_block3_add[0][0]']
conv4_block4_1_conv (Conv2D)	(None, 14, 14, 256) 262400	['conv4_block3_out[0][0]']
conv4_block4_1_bn (BatchNormal ization)	(None, 14, 14, 256) 1024	['conv4_block4_1_conv[0][0]']

conv4_block4_1_relu (Activation)	(None, 14, 14, 256) 0		['conv4_block4_1_bn[0][0]']
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 256) 590080		['conv4_block4_1_relu[0][0]']
conv4_block4_2_bn (BatchNormal ization)	(None, 14, 14, 256) 1024		['conv4_block4_2_conv[0][0]']
conv4_block4_2_relu (Activatio n)	(None, 14, 14, 256) 0		['conv4_block4_2_bn[0][0]']
conv4_block4_3_conv (Conv2D)	(None, 14, 14, 1024 263168)		['conv4_block4_2_relu[0][0]']
conv4_block4_3_bn (BatchNormal ization)	(None, 14, 14, 1024 4096)		['conv4_block4_3_conv[0][0]']
conv4_block4_add (Add)	(None, 14, 14, 1024 0)		['conv4_block3_out[0][0]', 'conv4_block4_3_bn[0][0]']
conv4_block4_out (Activation)	(None, 14, 14, 1024 0)		['conv4_block4_add[0][0]']
conv4_block5_1_conv (Conv2D)	(None, 14, 14, 256) 262400		['conv4_block4_out[0][0]']
conv4_block5_1_bn (BatchNormal ization)	(None, 14, 14, 256) 1024		['conv4_block5_1_conv[0][0]']
conv4_block5_1_relu (Activatio n)	(None, 14, 14, 256) 0		['conv4_block5_1_bn[0][0]']
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 256) 590080		['conv4_block5_1_relu[0][0]']
conv4_block5_2_bn (BatchNormal ization)	(None, 14, 14, 256) 1024		['conv4_block5_2_conv[0][0]']
conv4_block5_2_relu (Activatio n)	(None, 14, 14, 256) 0		['conv4_block5_2_bn[0][0]']
conv4_block5_3_conv (Conv2D)	(None, 14, 14, 1024 263168)		['conv4_block5_2_relu[0][0]']
conv4_block5_3_bn (BatchNormal ization)	(None, 14, 14, 1024 4096)		['conv4_block5_3_conv[0][0]']

conv4_block5_add (Add)	(None, 14, 14, 1024 0)		['conv4_block4_out[0][0]', 'conv4_block5_3_bn[0][0]']
conv4_block5_out (Activation)	(None, 14, 14, 1024 0)		['conv4_block5_add[0][0]']
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 256) 262400		['conv4_block5_out[0][0]']
conv4_block6_1_bn (BatchNormal ization)	(None, 14, 14, 256) 1024		['conv4_block6_1_conv[0][0]']
conv4_block6_1_relu (Activatio n)	(None, 14, 14, 256) 0		['conv4_block6_1_bn[0][0]']
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 256) 590080		['conv4_block6_1_relu[0][0]']
conv4_block6_2_bn (BatchNormal ization)	(None, 14, 14, 256) 1024		['conv4_block6_2_conv[0][0]']
conv4_block6_2_relu (Activatio n)	(None, 14, 14, 256) 0		['conv4_block6_2_bn[0][0]']
conv4_block6_3_conv (Conv2D)	(None, 14, 14, 1024 263168)		['conv4_block6_2_relu[0][0]']
conv4_block6_3_bn (BatchNormal ization)	(None, 14, 14, 1024 4096)		['conv4_block6_3_conv[0][0]']
conv4_block6_add (Add)	(None, 14, 14, 1024 0)		['conv4_block5_out[0][0]', 'conv4_block6_3_bn[0][0]']
conv4_block6_out (Activation)	(None, 14, 14, 1024 0)		['conv4_block6_add[0][0]']
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 512) 524800		['conv4_block6_out[0][0]']
conv5_block1_1_bn (BatchNormal ization)	(None, 7, 7, 512) 2048		['conv5_block1_1_conv[0][0]']
conv5_block1_1_relu (Activatio n)	(None, 7, 7, 512) 0		['conv5_block1_1_bn[0][0]']
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 512) 2359808		['conv5_block1_1_relu[0][0]']

conv5_block1_2_bn (BatchNormal ization)	(None, 7, 7, 512)	2048	['conv5_block1_2_conv[0][0]']
conv5_block1_2_relu (Activatio n)	(None, 7, 7, 512)	0	['conv5_block1_2_bn[0][0]']
conv5_block1_0_conv (Conv2D)	(None, 7, 7, 2048)	2099200	['conv4_block6_out[0][0]']
conv5_block1_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	['conv5_block1_2_relu[0][0]']
conv5_block1_0_bn (BatchNormal ization)	(None, 7, 7, 2048)	8192	['conv5_block1_0_conv[0][0]']
conv5_block1_3_bn (BatchNormal ization)	(None, 7, 7, 2048)	8192	['conv5_block1_3_conv[0][0]']
conv5_block1_add (Add)	(None, 7, 7, 2048)	0	['conv5_block1_0_bn[0][0]', 'conv5_block1_3_bn[0][0]']
conv5_block1_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block1_add[0][0]']
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 512)	1049088	['conv5_block1_out[0][0]']
conv5_block2_1_bn (BatchNormal ization)	(None, 7, 7, 512)	2048	['conv5_block2_1_conv[0][0]']
conv5_block2_1_relu (Activatio n)	(None, 7, 7, 512)	0	['conv5_block2_1_bn[0][0]']
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 512)	2359808	['conv5_block2_1_relu[0][0]']
conv5_block2_2_bn (BatchNormal ization)	(None, 7, 7, 512)	2048	['conv5_block2_2_conv[0][0]']
conv5_block2_2_relu (Activatio n)	(None, 7, 7, 512)	0	['conv5_block2_2_bn[0][0]']
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	['conv5_block2_2_relu[0][0]']
conv5_block2_3_bn (BatchNormal ization)	(None, 7, 7, 2048)	8192	['conv5_block2_3_conv[0][0]']
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	['conv5_block1_out[0][0]', 'conv5_block2_3_bn[0][0]']

```
'conv5_block2_3_bn[0][0]']

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

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

conv5_block3_1_bn (BatchNormal  (None, 7, 7, 512) 2048   ['conv5_block3_1_conv[0][0]']
                   ization)

conv5_block3_1_relu (Activatio (None, 7, 7, 512) 0      ['conv5_block3_1_bn[0][0]']
n)

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

conv5_block3_2_bn (BatchNormal (None, 7, 7, 512) 2048   ['conv5_block3_2_conv[0][0]']
                   ization)

conv5_block3_2_relu (Activatio (None, 7, 7, 512) 0      ['conv5_block3_2_bn[0][0]']
n)

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

conv5_block3_3_bn (BatchNormal (None, 7, 7, 2048) 8192   ['conv5_block3_3_conv[0][0]']
                   ization)

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

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

avg_pool (GlobalAveragePooling (None, 2048) 0      ['conv5_block3_out[0][0]']
2D)

=====
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
```

None

In [20]:

```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from tensorflow.keras.applications.resnet50 import preprocess_input
```

In [21]:

```
def extract_features(directory):
    count=0
    features = dict()
    for name in listdir(directory):
        filename = directory + '/' + name
        image = load_img(filename, target_size=(224, 224))
        image = img_to_array(image)
        image = image.reshape(1,224,224,3)
        image = preprocess_input(image)
        feature = model.predict(image)
        image_id = name.split('.')[0]
        features[image_id] = feature
        count=count+1
    if(count%500==0):
        print(count)
    return features

# extract features from all images
directory = 'C:/Users/mahid/M.Tech Project/Data Set/Images/'
features = extract_features(directory)
print('Extracted Features: %d' % len(features))
```

```
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
7500
8000
Extracted Features: 8091
```

In [22]:

```
dump(features, open('C:/Users/mahid/M.Tech Project/Resnet_Model/features.pkl','wb'))
```

In [1]:

```
import string
def mapCaptions(file):
    captions_map={}
    for i in file.split('\n'):
        img_data=i.split()
        if len(i)<2:
            continue
        image_id=img_data[0]
        image_caption=img_data[1:]
        image_id=image_id.split('.')[0]
        image_caption=' '.join(image_caption)
        if image_id not in captions_map:
            captions_map[image_id]=[]
        captions_map[image_id].append(image_caption)
    return captions_map
```

In [2]:

```
def cleanRawCaptions(descriptions):
    mytable=str.maketrans(' ', ' ', string.punctuation)
    for image_id,image_caption in descriptions.items():
        for i in range(len(image_caption)):
            caption=image_caption[i]
            caption=caption.split()
            caption=[word.lower() for word in caption]
            caption=[w.translate(mytable) for w in caption]
            caption=[word for word in caption if len(word)>1]
            caption=[word for word in caption if word.isalpha()]
            image_caption[i]=' '.join(caption)
```

In [3]:

```
def vocabularyExtraction(image_captions):
    list_words = set()
    for key in image_captions.keys():
        [list_words.update(d.split()) for d in image_captions[key]]
    return list_words

def countWords(captions_dict):
    word_count={}
    for image,captions in captions_dict.items():
        for caption in captions:
            for word in caption.split():
                if word not in word_count:
                    word_count[word] = 0
                else:
```

```

        word_count[word] += 1
    return(word_count)

# save descriptions to file, one per line
def save_descriptions(descriptions, filename):
    lines=[]
    for image_id,image_caption in descriptions.items():
        for caption in image_caption:
            lines.append(image_id + ' ' +caption)
    data='\n'.join(lines)
    file=open(filename, 'w')
    file.write(data)
    file.close()

```

In [4]:

```

file=open('C:/Users/mahid/M.Tech Project/Data Set/Flickr8k_text/Flickr8k.token.txt', 'r')
text=file.read()
file.close()
desc =mapCaptions(text)
print('Total images mapped with captions : %d' % len(desc))
cleanRawCaptions(desc)
vocabulary =vocabularyExtraction(desc)
wordCount=countWords(desc)
print('Vocabulary Size after cleaning : %d' % len(vocabulary))
saveDescriptions(desc,'C:/Users/mahid/M.Tech Project/Resnet_Model/descriptions.txt')

```

Total images mapped with captions : 8091
 Vocabulary Size after cleaning : 8762

In [5]:

```

sorted_list= sorted(word_count.items(), key=lambda x:x[1],reverse=True)
top_30_items={}
count=30
for word,counter in sorted_list:
    top_30_items[word]=counter
    if(count==0):
        break
    count=count-1
for word in top_30_items:
    print(word,top_30_items[word])

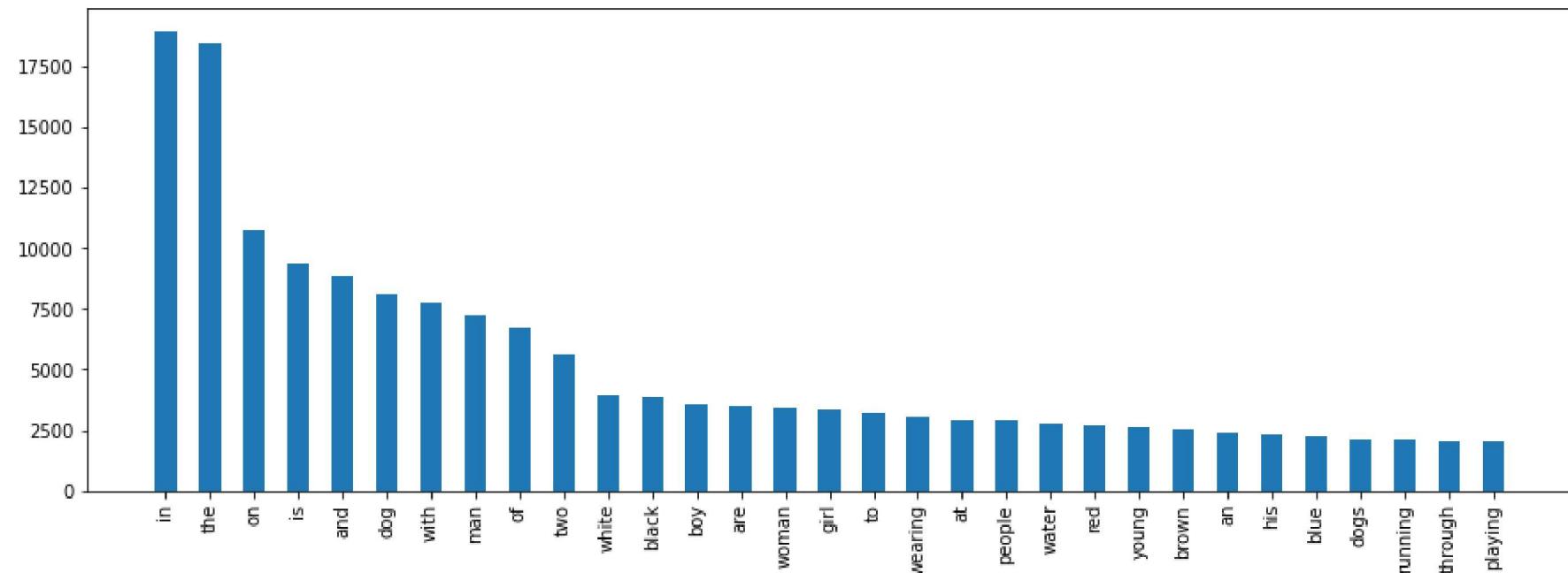
```

in 18969
 the 18415
 on 10743

is 9342
and 8851
dog 8130
with 7763
man 7265
of 6712
two 5638
white 3939
black 3831
boy 3580
are 3504
woman 3402
girl 3327
to 3171
wearing 3061
at 2914
people 2886
water 2782
red 2671
young 2629
brown 2560
an 2431
his 2356
blue 2267
dogs 2124
running 2072
through 2031
playing 2006

In [6]:

```
import matplotlib.pyplot as plt
keys=top_30_items.keys()
values=top_30_items.values()
plt.figure(figsize=(15,5))
plt.bar(keys,values,width=0.5)
plt.xticks(range(len(values)),keys,rotation=90)
plt.show()
```



```
In [6]: def load_set(filename):
    dataset=[]
    file=open(filename, 'r')
    text=file.read()
    file.close()
    for line in text.split('\n'):
        identifier = line.split('.')[0]
        dataset.append(identifier)
    return(dataset)
```

```
In [7]: filename = 'C:/Users/mahid/M.Tech Project/Data Set/Flickr8k_text/Flickr_8k.trainImages.txt'
train_dataset=load_set(filename)
print('Training Data Set Length is : %d' % len(train_dataset))
```

Training Data Set Length is : 6000

```
In [8]: def add_identifiers(filename,dataset):
    file=open(filename,'r')
    text=file.read()
    file.close()
```

```

captions={}
for line in text.split('\n'):
    image_data=line.split()
    image_id,image_caption=image_data[0],image_data[1:]
    if image_id in dataset:
        if image_id not in captions:
            captions[image_id]=[]
        modified= 'startseq '+' '.join(image_caption)+' endseq'
        captions[image_id].append(modified)
return captions

```

In [9]:

```

train_captions=add_identifiers('C:/Users/mahid/M.Tech Project/Resnet_Model/descriptions.txt',train_dataset)
print('Total image Captions read : train %d' % len(train_captions))

```

Total image Captions read : train 5999

In [10]:

```

for i in range(10):
    print(train_captions[train_dataset[i]])

```

['startseq black dog is running after white dog in the snow endseq', 'startseq black dog chasing brown dog through snow endseq',
 'startseq two dogs chase each other across the snowy ground endseq', 'startseq two dogs play together in the snow endseq', 'starts
 eq two dogs running through low lying body of water endseq'
 ['startseq little baby plays croquet endseq', 'startseq little girl plays croquet next to truck endseq', 'startseq the child is pl
 aying croquette by the truck endseq', 'startseq the kid is in front of car with put and ball endseq', 'startseq the little boy is
 playing with croquet hammer and ball beside the car endseq'
 ['startseq brown dog in the snow has something hot pink in its mouth endseq', 'startseq brown dog in the snow holding pink hat end
 seq', 'startseq brown dog is holding pink shirt in the snow endseq', 'startseq dog is carrying something pink in its mouth while w
 alking through the snow endseq', 'startseq dog with something pink in its mouth is looking forward endseq'
 ['startseq brown dog is running along beach endseq', 'startseq brown dog wearing black collar running across the beach endseq', 's
 tartseq dog walks on the sand near the water endseq', 'startseq brown dog running on the beach endseq', 'startseq the large brown
 dog is running on the beach by the ocean endseq'
 ['startseq black and white dog with red frisbee standing on sandy beach endseq', 'startseq dog drops red disc on beach endseq', 's
 tartseq dog with red frisbee flying in the air endseq', 'startseq dog catching red frisbee endseq', 'startseq the black dog is dro
 pping red disc on beach endseq'
 ['startseq cyclist wearing red helmet is riding on the pavement endseq', 'startseq girl is riding bike on the street while wearing
 red helmet endseq', 'startseq person on bike wearing red helmet riding down street endseq', 'startseq woman wears red helmet and b
 lue shirt as she goes for bike ride in the shade endseq', 'startseq person in blue shirt and red helmet riding bike down the road
 endseq'
 ['startseq man dressed in purple shirt and red bandanna smiles at the people watching him endseq', 'startseq man on the street we
 ring leather chaps and chainmail codpiece endseq', 'startseq man wearing purple shirt and black leather chaps poses for the camera
 endseq', 'startseq man dressed in leather chaps and purple shirt stands in front of onlookers endseq', 'startseq there is man in p
 urple shirt leather chaps and red bandanna standing near other men endseq']

```
[ 'startseq boy wearing red tshirt is running through woodland endseq', 'startseq child runs near some trees endseq', 'startseq young boy is dancing around endseq', 'startseq young boy with red short sleeved shirt and jeans runs by some trees endseq', 'startseq the little boy in the red shirt stops to smile for the camera endseq']
[ 'startseq girl in white dress endseq', 'startseq little girl in white is looking back at the camera while carrying water grenade endseq', 'startseq smiling young girl in braids is playing ball endseq', 'startseq young girl wearing white looks at the camera as she plays endseq', 'startseq the girl is holding green ball endseq']
[ 'startseq skier in yellow jacket is airborne above the mountains endseq', 'startseq skier jumps high in the air with view of the mountains endseq', 'startseq skiing man in fluorescent jacket jumps very high and it looks as though he is flying endseq', 'starts eq somone is high in the air doing ski jump endseq', 'startseq the skier in the green jacket and white pants appears to almost fly into the sky endseq']
```

In [11]:

```
from pickle import load
def load_features(filename, dataset):
    features={}
    image_features=load(open(filename, 'rb'))
    for i in dataset:
        if i in image_features:
            features[i]=image_features[i]
    return features
```

In [12]:

```
train_image_vector=load_features('C:/Users/mahid/M.Tech Project/Resnet_Model/features.pkl',train_dataset)
print('Total Features for images obtained : =%d' % len(train_image_vector))
```

Total Features for images obtained : =5999

In [13]:

```
c=0
for i,j in train_captions.items():
    print(str(i))
    print(str(train_captions[i]))
    c=c+1
    if(c==10):
        break
```

1000268201_693b08cb0e

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

1001773457_577c3a7d70

```
[ 'startseq black dog and spotted dog are fighting endseq', 'startseq black dog and tricolored dog playing with each other on the r oad endseq', 'startseq black dog and white dog with brown spots are staring at each other in the street endseq', 'startseq two dog s of different breeds looking at each other on the road endseq', 'startseq two dogs on pavement moving toward each other endseq']
```

1002674143_1b742ab4b8

['startseq little girl covered in paint sits in front of painted rainbow with her hands in bowl endseq', 'startseq little girl is sitting in front of large painted rainbow endseq', 'startseq small girl in the grass plays with fingerpaints in front of white canvas with rainbow on it endseq', 'startseq there is girl with pigtails sitting in front of rainbow painting endseq', 'startseq young girl with pigtails painting outside in the grass endseq']
1003163366_44323f5815

['startseq man lays on bench while his dog sits by him endseq', 'startseq man lays on the bench to which white dog is also tied endseq', 'startseq man sleeping on bench outside with white and black dog sitting next to him endseq', 'startseq shirtless man lies on park bench with his dog endseq', 'startseq man laying on bench holding leash of dog sitting on ground endseq']
1007129816_e794419615

['startseq man in an orange hat starring at something endseq', 'startseq man wears an orange hat and glasses endseq', 'startseq man with gauges and glasses is wearing blitz hat endseq', 'startseq man with glasses is wearing beer can crocheted hat endseq', 'startseq the man with pierced ears is wearing glasses and an orange hat endseq']
1007320043_627395c3d8

['startseq child playing on rope net endseq', 'startseq little girl climbing on red roping endseq', 'startseq little girl in pink climbs rope bridge at the park endseq', 'startseq small child grips onto the red ropes at the playground endseq', 'startseq the small child climbs on red ropes on playground endseq']
1009434119_febe49276a

['startseq black and white dog is running in grassy garden surrounded by white fence endseq', 'startseq black and white dog is running through the grass endseq', 'startseq boston terrier is running in the grass endseq', 'startseq boston terrier is running on lush green grass in front of white fence endseq', 'startseq dog runs on the green grass near wooden fence endseq']
1012212859_01547e3f17

['startseq dog shakes its head near the shore red ball next to it endseq', 'startseq white dog shakes on the edge of beach with an orange ball endseq', 'startseq dog with orange ball at feet stands on shore shaking off water endseq', 'startseq white dog playing with red ball on the shore near the water endseq', 'startseq white dog with brown ears standing near water with head turned to one side endseq']
1015118661_980735411b

['startseq boy smiles in front of stony wall in city endseq', 'startseq little boy is standing on the street while man in overalls is working on stone wall endseq', 'startseq young boy runs across the street endseq', 'startseq young child is walking on stone paved street with metal pole and man behind him endseq', 'startseq smiling boy in white shirt and blue jeans in front of rock wall with man in overalls behind him endseq']
1015584366_dfcec3c85a

['startseq black dog leaps over log endseq', 'startseq grey dog is leaping over fallen tree endseq', 'startseq large black dog leaps fallen log endseq', 'startseq mottled black and grey dog in blue collar jumping over fallen tree endseq', 'startseq the black dog jumped the tree stump endseq']

In [14]:

```
def split_captions(captions):
    caption=[]
    for i in captions.keys():
        [caption.append(j) for j in captions[i]]
    return caption
```

In [15]:

```
from keras.preprocessing.text import Tokenizer
```

```
In [16]: def token_convert(captions):
    lines=split_captions(captions)
    tokenizer=Tokenizer()
    tokenizer.fit_on_texts(lines)
    return tokenizer
```

```
In [17]: tokenizer=token_convert(train_captions)
vocabulary_size=len(tokenizer.word_index)+1
print('Vocabulary Size: %d' % vocabulary_size)
```

Vocabulary Size: 7578

```
In [18]: lines=split_captions(train_captions)
max_length=max(len(i.split()) for i in lines)
print('Maximum caption length is:%d' %max_length)
```

Maximum caption length is:34

```
In [19]: for i in range(20):
    print(lines[i])
```

```
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 tricolored 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
startseq little girl covered in paint sits in front of painted rainbow with her hands in bowl endseq
startseq little girl is sitting in front of large painted rainbow endseq
startseq small girl in the grass plays with fingerpaints in front of white canvas with rainbow on it endseq
startseq there is girl with pigtails sitting in front of rainbow painting endseq
startseq young girl with pigtails painting outside in the grass endseq
startseq man lays on bench while his dog sits by him endseq
startseq man lays on the bench to which white dog is also tied endseq
startseq man sleeping on bench outside with white and black dog sitting next to him endseq
```

```
startseq shirtless man lies on park bench with his dog endseq  
startseq man laying on bench holding leash of dog sitting on ground endseq
```

```
In [20]: from numpy import array
```

```
In [21]: def output_sequences(tokenizer,max_length,caption_list,image):  
    inp1=[]  
    inp2=[]  
    out=[]  
    for caption in caption_list:  
        sequence=tokenizer.texts_to_sequences([caption])[0]  
        for i in range(1,len(sequence)):  
            input_sequence,output_sequence=sequence[:i],sequence[i]  
            input_sequence=pad_sequences([input_sequence], maxlen=max_length)[0]  
            output_sequence=to_categorical([output_sequence], num_classes=vocabulary_size)[0]  
            inp1.append(image)  
            inp2.append(input_sequence)  
            out.append(output_sequence)  
    return array(inp1),array(inp2),array(out)
```

```
In [22]: def data_generator(tokenizer,max_length,captions,images):  
    while 1:  
        for key,desc in captions.items():  
            image=images[key][0]  
            input_img,input_seq,output_word=output_sequences(tokenizer,max_length,desc,image)  
            yield [[input_img, input_seq], output_word]
```

```
In [23]: import tensorflow  
from keras.preprocessing.sequence import pad_sequences  
from tensorflow.keras.utils import to_categorical  
from tensorflow.keras.utils import plot_model  
from keras.models import Model,Sequential  
from keras.layers import Input  
from keras.layers import Dense  
from keras.layers import LSTM  
from keras.layers import Embedding  
from keras.layers import Dropout  
from keras.layers.merge import add  
from keras.callbacks import ModelCheckpoint  
from keras.layers import Flatten,Convolution2D,TimeDistributed,Bidirectional,Activation,RepeatVector,Concatenate
```

```
from tensorflow.keras.utils import plot_model
from numpy import argmax
from keras.preprocessing.sequence import pad_sequences
from keras.models import load_model
from nltk.translate.bleu_score import corpus_bleu
```

In [28]:

```
def create_model(vocabulary_size,max_length):
    inputs1=Input(shape=(2048,))
    img_model_1=Dropout(0.5)(inputs1)
    img_model_2=Dense(128,activation='relu')(img_model_1)
    # sequence model
    inputs2=Input(shape=(max_length,))
    sequence_model_1=Embedding(vocabulary_size,128,mask_zero=True)(inputs2)
    sequence_model_2=Dropout(0.5)(sequence_model_1)
    sequence_model_3=LSTM(128)(sequence_model_2)
    # decoder model
    decoder1=add([img_model_2,sequence_model_3])
    decoder2=Dense(128,activation='relu')(decoder1)
    outputs=Dense(vocabulary_size,activation='softmax')(decoder2)
    # tie it together [image, seq] [word]
    model=Model(inputs=[inputs1,inputs2],outputs=outputs)
    model.compile(loss='categorical_crossentropy', optimizer='adam')
    # summarize model
    print(model.summary())
    return model
```

In [29]:

```
print(vocabulary_size)
print(max_length)
```

7578
34

In [31]:

```
model=create_model(vocabulary_size, max_length)
epochs=20
steps=len(train_captions)
for i in range(epochs):
    generator=data_generator(tokenizer,max_length,train_captions,train_image_vector)
    model.fit(generator,epochs=1,steps_per_epoch=steps)
    model.save('C:/Users/mahid/M.Tech Project/Resnet_Model/caption_model_'+str(i)+'.h5')
```

Model: "model_3"

Layer (type)	Output Shape	Param #	Connected to
<hr/>			
input_8 (InputLayer)	[(None, 34)]	0	[]
embedding_3 (Embedding)	(None, 34, 128)	969984	['input_8[0][0]']
input_7 (InputLayer)	[(None, 2048)]	0	[]
dropout_7 (Dropout)	(None, 34, 128)	0	['embedding_3[0][0]']
dense_9 (Dense)	(None, 128)	262272	['input_7[0][0]']
lstm_3 (LSTM)	(None, 128)	131584	['dropout_7[0][0]']
add_3 (Add)	(None, 128)	0	['dense_9[0][0]', 'lstm_3[0][0]']
dense_10 (Dense)	(None, 128)	16512	['add_3[0][0]']
dense_11 (Dense)	(None, 7578)	977562	['dense_10[0][0]']
<hr/>			
Total params:	2,357,914		
Trainable params:	2,357,914		
Non-trainable params:	0		

None

```
5999/5999 [=====] - 412s 68ms/step - loss: 4.6999
5999/5999 [=====] - 524s 87ms/step - loss: 3.9022
5999/5999 [=====] - 523s 87ms/step - loss: 3.6302
5999/5999 [=====] - 457s 76ms/step - loss: 3.4708
5999/5999 [=====] - 433s 72ms/step - loss: 3.3664
5999/5999 [=====] - 412s 69ms/step - loss: 3.2837
5999/5999 [=====] - 417s 69ms/step - loss: 3.2261
5999/5999 [=====] - 424s 71ms/step - loss: 3.1800
5999/5999 [=====] - 528s 88ms/step - loss: 3.1445
5999/5999 [=====] - 539s 90ms/step - loss: 3.1099
5999/5999 [=====] - 533s 89ms/step - loss: 3.0884
5999/5999 [=====] - 534s 89ms/step - loss: 3.0707
5999/5999 [=====] - 439s 73ms/step - loss: 3.0521
5999/5999 [=====] - 438s 73ms/step - loss: 3.0305
5999/5999 [=====] - 492s 82ms/step - loss: 3.0231
5999/5999 [=====] - 507s 84ms/step - loss: 3.0146
```

```
5999/5999 [=====] - 501s 84ms/step - loss: 2.9965
5999/5999 [=====] - 500s 83ms/step - loss: 2.9952
5999/5999 [=====] - 515s 86ms/step - loss: 2.9854
5999/5999 [=====] - 524s 87ms/step - loss: 2.9777
```

In [32]:

```
def word_mapping(number,tokenizer):
    for word,int_id in tokenizer.word_index.items():
        if int_id==number:
            return word
    return None
```

In [33]:

```
def caption_generate(model,tokenizer,image,max_length):
    caption='startseq'
    for i in range(max_length):
        sequence=tokenizer.texts_to_sequences([caption])[0]
        sequence=pad_sequences([sequence], maxlen=max_length)
        word_id=argmax(model.predict([image,sequence], verbose=0))
        word=word_mapping(word_id,tokenizer)
        if word is None:
            break
        caption += ' ' + word
        if word=='endseq':
            break
    return caption
```

In [34]:

```
def evaluate_model(model,descriptions,photos,tokenizer,max_length):
    actual=[]
    predicted=[]
    for key,desc_list in descriptions.items():
        pred=caption_generate(model,tokenizer,photos[key],max_length)
        act=[i.split() for i in desc_list]
        actual.append(act)
        predicted.append(pred.split())
    print('BLEU-1 Score: %f' % corpus_bleu(actual,predicted,weights=(1.0, 0, 0, 0)))
    print('BLEU-2 Score: %f' % corpus_bleu(actual,predicted,weights=(0.5, 0.5, 0, 0)))
    print('BLEU-3 Score: %f' % corpus_bleu(actual,predicted,weights=(0.33, 0.33, 0.33, 0)))
    print('BLEU-4 Score: %f' % corpus_bleu(actual,predicted,weights=(0.25, 0.25, 0.25, 0.25)))
```

In [35]:

```
filename = 'C:/Users/mahid/M.Tech Project/Data Set/Flickr8k_text/Flickr_8k.testImages.txt'
test_dataset = load_set(filename)
```

```
print('Dataset: %d' % len(test_dataset))
test_captions=add_identifiers('C:/Users/mahid/M.Tech Project/Resnet_Model/descriptions.txt',test_dataset)
print('Captions: test=%d' % len(test_captions))
test_features=load_features('C:/Users/mahid/M.Tech Project/Resnet_Model/features.pkl',test_dataset)
print('Photos: test=%d' % len(test_features))
for i in range(20):
    print("Model "+str(i)+" : ")
    filename='C:/Users/mahid/M.Tech Project/Resnet_Model/caption_model_'+str(i)+'.h5'
    model=load_model(filename)
    evaluate_model(model,test_captions,test_features,tokenizer,max_length)
```

Dataset: 1001
Captions: test=1000
Photos: test=1000
Model 0 :
BLEU-1 Score: 0.577783
BLEU-2 Score: 0.316408
BLEU-3 Score: 0.179528
BLEU-4 Score: 0.091401
Model 1 :
BLEU-1 Score: 0.565313
BLEU-2 Score: 0.314850
BLEU-3 Score: 0.186704
BLEU-4 Score: 0.100961
Model 2 :
BLEU-1 Score: 0.561332
BLEU-2 Score: 0.315540
BLEU-3 Score: 0.188081
BLEU-4 Score: 0.101612
Model 3 :
BLEU-1 Score: 0.566897
BLEU-2 Score: 0.317750
BLEU-3 Score: 0.189049
BLEU-4 Score: 0.105657
Model 4 :
BLEU-1 Score: 0.526355
BLEU-2 Score: 0.292087
BLEU-3 Score: 0.174097
BLEU-4 Score: 0.098369
Model 5 :
BLEU-1 Score: 0.493483
BLEU-2 Score: 0.270746
BLEU-3 Score: 0.157290
BLEU-4 Score: 0.085316

Model 6 :
BLEU-1 Score: 0.540192
BLEU-2 Score: 0.295842
BLEU-3 Score: 0.172854
BLEU-4 Score: 0.096122
Model 7 :
BLEU-1 Score: 0.543165
BLEU-2 Score: 0.301070
BLEU-3 Score: 0.177391
BLEU-4 Score: 0.099688
Model 8 :
BLEU-1 Score: 0.552535
BLEU-2 Score: 0.304719
BLEU-3 Score: 0.179835
BLEU-4 Score: 0.099554
Model 9 :
BLEU-1 Score: 0.547822
BLEU-2 Score: 0.300504
BLEU-3 Score: 0.176673
BLEU-4 Score: 0.100559
Model 10 :
BLEU-1 Score: 0.538199
BLEU-2 Score: 0.295056
BLEU-3 Score: 0.174768
BLEU-4 Score: 0.100190
Model 11 :
BLEU-1 Score: 0.524493
BLEU-2 Score: 0.286001
BLEU-3 Score: 0.169066
BLEU-4 Score: 0.093922
Model 12 :
BLEU-1 Score: 0.490771
BLEU-2 Score: 0.270756
BLEU-3 Score: 0.158838
BLEU-4 Score: 0.088547
Model 13 :
BLEU-1 Score: 0.545201
BLEU-2 Score: 0.295359
BLEU-3 Score: 0.174080
BLEU-4 Score: 0.097238
Model 14 :
BLEU-1 Score: 0.522232
BLEU-2 Score: 0.286351
BLEU-3 Score: 0.168651

BLEU-4 Score: 0.094575
Model 15 :
BLEU-1 Score: 0.529440
BLEU-2 Score: 0.289739
BLEU-3 Score: 0.169847
BLEU-4 Score: 0.093482
Model 16 :
BLEU-1 Score: 0.537868
BLEU-2 Score: 0.293744
BLEU-3 Score: 0.170875
BLEU-4 Score: 0.093140
Model 17 :
BLEU-1 Score: 0.528870
BLEU-2 Score: 0.284442
BLEU-3 Score: 0.165249
BLEU-4 Score: 0.092092
Model 18 :
BLEU-1 Score: 0.530353
BLEU-2 Score: 0.293163
BLEU-3 Score: 0.175838
BLEU-4 Score: 0.098891
Model 19 :
BLEU-1 Score: 0.456161
BLEU-2 Score: 0.241598
BLEU-3 Score: 0.139170
BLEU-4 Score: 0.074507