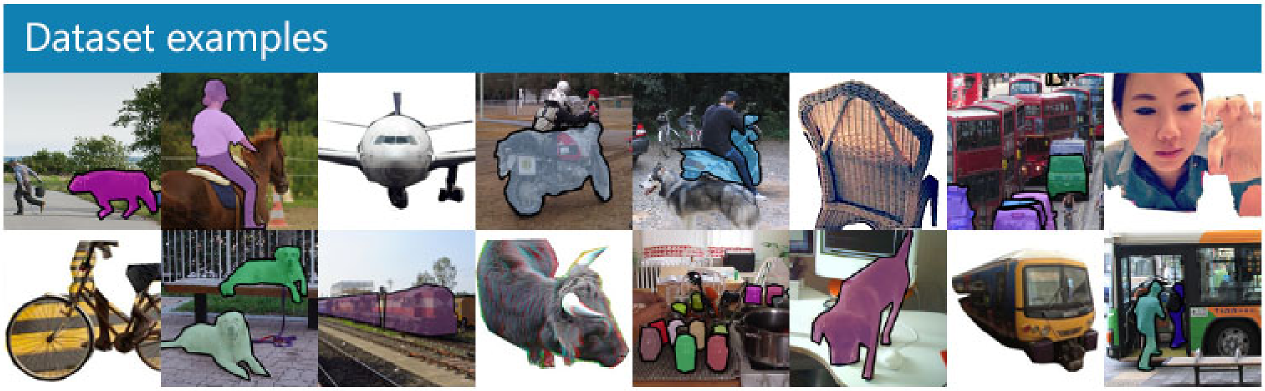
Computer Vision Nanodegree

Project: Image Captioning

The Microsoft **C**ommon **O**bjects in **CO**ntext (MS COCO) dataset is a large-scale dataset for scene understanding. The dataset is commonly used to train and benchmark object detection, segmentation, and captioning algorithms.



You can read more about the dataset on the website or in the research paper.

In this notebook, you will explore this dataset, in preparation for the project.

Step 1: Initialize the COCO API

We begin by initializing the COCO API that you will use to obtain the data.

In [1]:

```
import os
import sys
sys.path.append('/opt/cocoapi/PythonAPI')
from pycocotools.coco import COCO

# initialize COCO API for instance annotations
dataDir = 'opt/cocoapi'
dataType = 'val2014'
instances_annFile = os.path.join(dataDir, 'annotations/instances_{}.json'.format(dataType))
coco = COCO(instances_annFile)

# initialize COCO API for caption annotations
captions_annFile = os.path.join(dataDir, 'annotations/captions_{}.json'.format(dataType))
coco_caps = COCO(captions_annFile)

# get image ids
ids = list(coco.anns.keys())
```

```
loading annotations into memory...
Done (t=2.41s)
creating index...
index created!
loading annotations into memory...
Done (t=0.15s)
creating index...
index created!
```

In [2]:

```
list(coco.anns.values())[0]
```

Out[2]:

```
{'segmentation': [[239.97,
   260.24,
   222.04,
   270.49,
   199.84,
   253.41,
   213.5,
   227.79,
   259.62,
   200.46,
   274.13,
   202.17,
   277.55,
   210.71,
   249.37,
   253.41,
   237.41,
   264.51,
   242.54,
   261.95,
   228.87,
   271.34]],
 'area': 2765.1486500000005,
 'iscrowd': 0,
 'image_id': 558840,
 'bbox': [199.84, 200.46, 77.71, 70.88],
 'category_id': 58,
 'id': 156}
```

Step 2: Plot a Sample Image

Next, we plot a random image from the dataset, along with its five corresponding captions. Each time you run the code cell below, a different image is selected.

In the project, you will use this dataset to train your own model to generate captions from images!

In [3]:

```
import numpy as np
import skimage.io as io
import matplotlib.pyplot as plt
%matplotlib inline

# pick a random image and obtain the corresponding URL
ann_id = np.random.choice(ids)
img_id = coco.anns[ann_id]['image_id']
img = coco.loadImgs(img_id)[0]
url = img['coco_url']

# print URL and visualize corresponding image
print(url)
```

```python
I = io.imread(url)
plt.axis('off')
plt.imshow(I)
plt.show()

# load and display captions
annIds = coco_caps.getAnnIds(imgIds=img['id']);
anns = coco_caps.loadAnns(annIds)
coco_caps.showAnns(anns)
```

```
http://images.cocodataset.org/val2014/COCO_val2014_000000161962.jpg
```

```
A picture of a building and some grass.
A photo of a courtyard with a building in the background.
People are on the lawn enjoying the park situated in front of a tourist spot.
A castle-like building is in the background while the foreground is a green grass lawn, part of w
hich has been mowed.
A large public park stationed in front of a building.
```

Step 3: What's to Come!

In this project, you will use the dataset of image-caption pairs to train a CNN-RNN model to automatically generate images from captions. You'll learn more about how to design the architecture in the next notebook in the sequence (**1_Preliminaries.ipynb**).

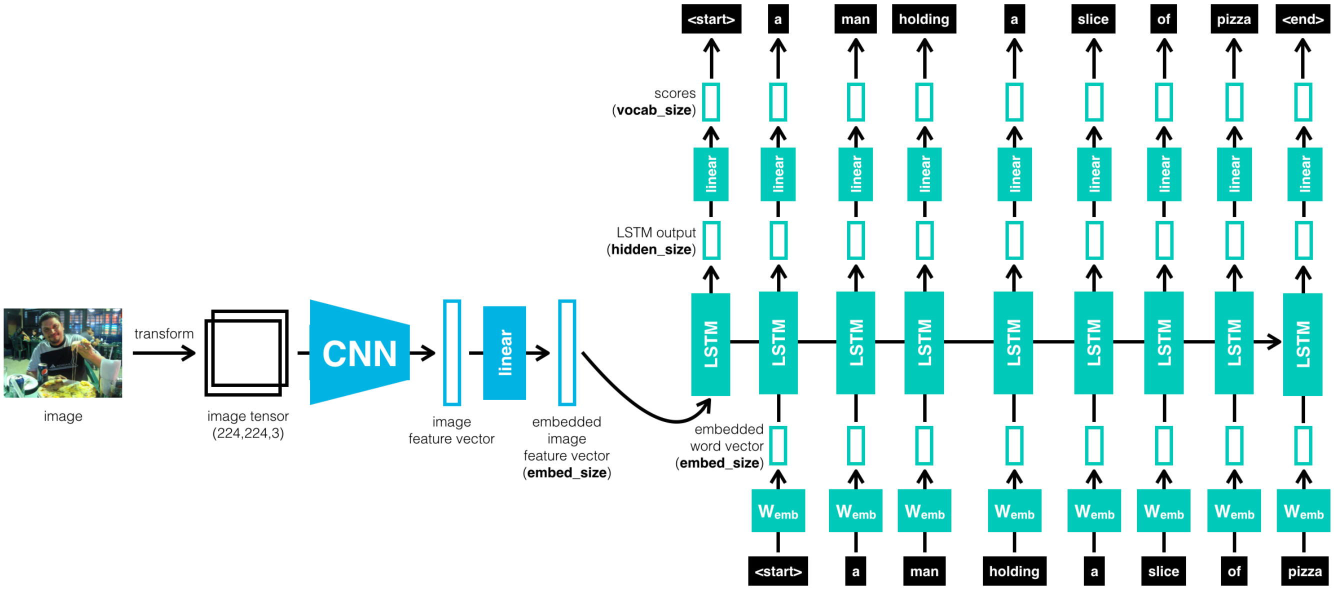


image
transform
image tensor
(224,224,3)
CNN
image
feature vector
linear
embedded
image
feature vector
(embed_size)
LSTM
embedded
word vector
(embed_size)
LSTM output
(hidden_size)
linear
scores
(vocab_size)
Wemb
<start>
a
man
holding
a
slice
of
pizza
<end>

```
In [ ]:
```