Document scope: Lab assignment – 4
Team: #5

CS5542-Big data Analytics and Applications

LAB ASSIGNMENT-4 REPORT

Submitted by: Koushik Katakam

Class ID: 10

Objective:

The main objective of this lab is to implement a bottom-up attention model to generate captions for an image.

Introduction:

Caption generation is the challenging artificial intelligence problem using NLP technique and computer vision. It requires the two pictures comprehension and language show from the field of Natural language processing. For sure, a depiction must catch the articles contained in a picture, yet it additionally should express how these items identify with one another, just as their characteristics and the exercises they are associated with.

Technologies used:

- Pycharm – A python IDE

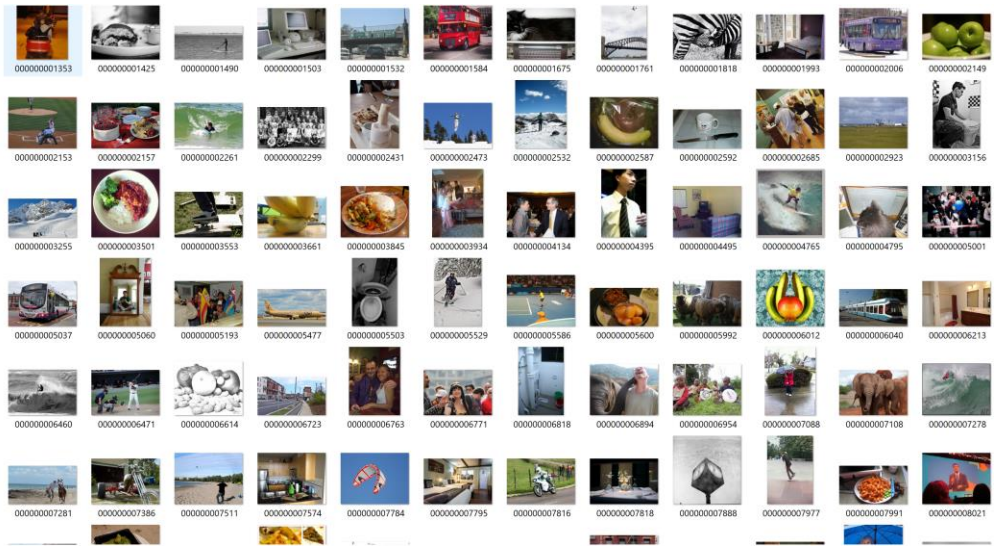
Libraries used:

- Tensorflow r1.0
- NLTK
- pandas
- MSCOCO images and captions.
- InceptionV4
- PIL

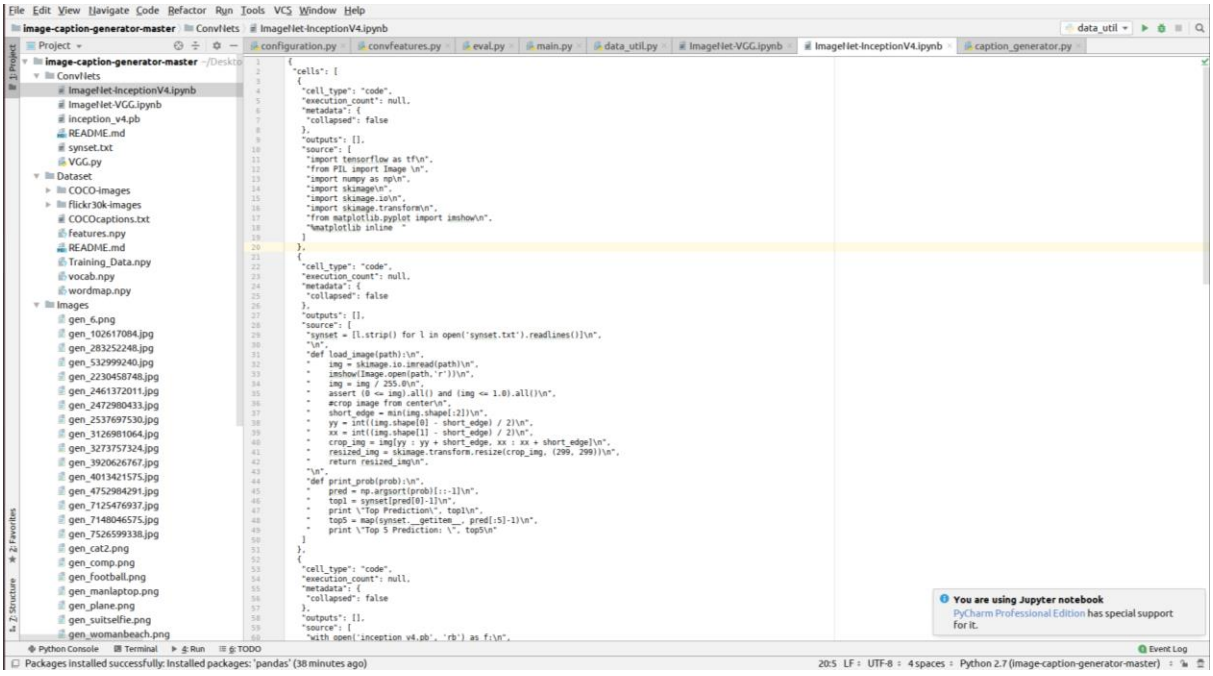
Screenshots of the obtained results:

Dataset: We have used MSCOCO dataset for training the model. The image dataset is as follows

Document scope: Lab assignment – 4
Team: #5

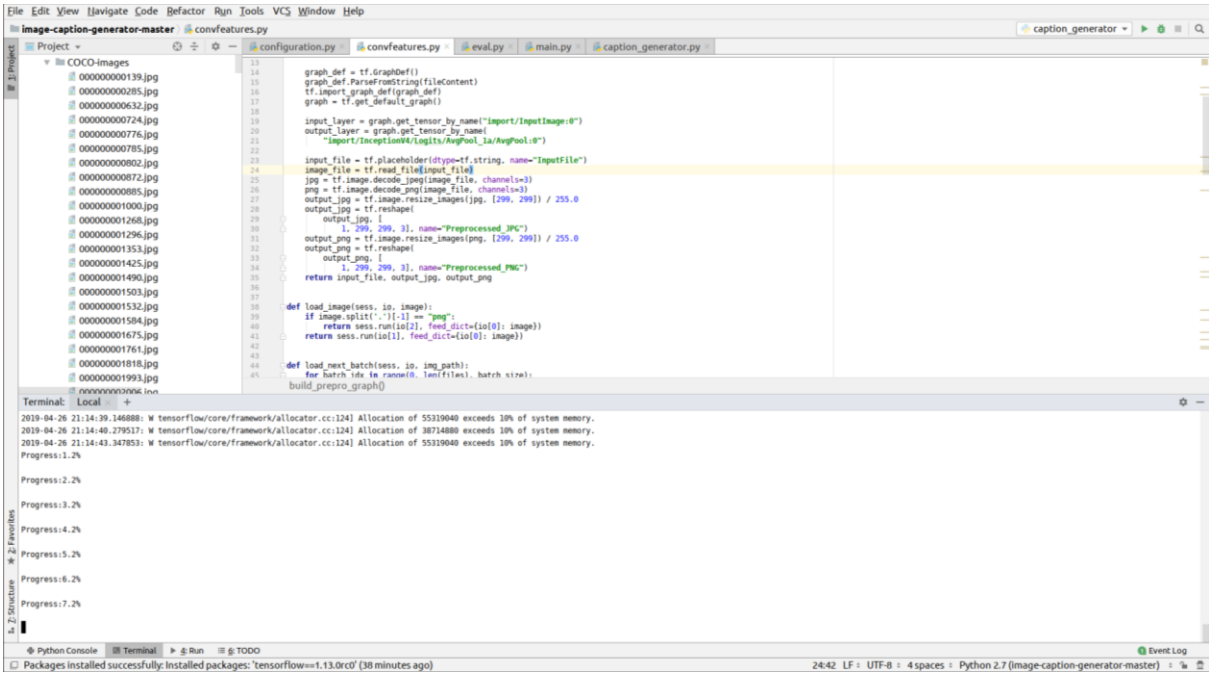


We have used inception model, an image recognition model that attained greater than 78.1% accuracy on the ImageNet dataset.



To implement attention-based model we have implemented various steps in the code.

Document scope: Lab assignment – 4
Team: #5



From this step we have obtained a features.npy file which extracted features of the images. Later this file, along with images is used to train the model and generate the captions.

```
def generate_vocab(df):
    global max_len, word_threshold, counter
    print "Generating Vocabulary"

    vocab = dict([w for w in counter.items() if w[1] >= word_threshold])
    vocab["<UNK>"] = len(counter) - len(vocab)
    vocab["<PAD>"] = df.caption.str.count("<PAD>").sum()
    vocab["<S>"] = df.caption.str.count("<S>").sum()
    vocab["</S>"] = df.caption.str.count("</S>").sum()
    wtoidx = {}
    wtoidx["<S>"] = 1
    wtoidx["</S>"] = 2
    wtoidx["<PAD>"] = 0
    wtoidx["<UNK>"] = 3
    print "Generating Word to Index and Index to Word"
    i = 4
    for word in vocab.keys():
        if word not in ["<S>", "</S>", "<PAD>", "<UNK>"]:
            wtoidx[word] = i
            i += 1
    print "Size of Vocabulary", len(vocab)
    return vocab, wtoidx
```

Document scope: Lab assignment – 4
Team: #5

Generating the captions:

```
    feat_path= Dataset/features.npy ,
    data_is_coco=False):
required_files = ["vocab", "wordmap", "Training_Data"]
generate = False
for fil in required_files:
    if not os.path.isfile('Dataset/' + fil + ".npy"):
        generate = True
        print "Required Files not present. Regenerating Data."
        break
if not generate:
    print "Dataset Present; Skipping Generation."
    return get_data(required_files)
global max_len, word_threshold, counter
max_len = ml
word_threshold = wt
print "Loading Caption Data", cap_path
if data_is_coco:
    # Prepare COCO captions in Flickr format
    cap_path = prepare_coco_captions(cap_path)
    # Load the COCO captions data
    with open(cap_path, 'r') as f:
        data = f.readlines()
    filenames = [caps.split('\t')[0].split('#')[0] for caps in data]
    captions = [caps.split('\t')[1] for caps in data]
    df = preprocess_coco_captions(filenames, captions)
else:
    with open(cap_path, 'r') as f:
        data = f.readlines()
    filenames = [caps.split('\t')[0].split('#')[0] for caps in data]
    captions = [caps.replace('\n', '').split('\t')[1] for caps in data]
    df = preprocess_flickr_captions(filenames, captions)
```

Caption generation and scores results:

```
Blue cumulative 2-gram: 0.000000
The hypothesis contains 0 counts of 2-gram overlaps.
Glue score for this sentence: 0.11764705882352941
Therefore the BLEU score evaluates to 0, independently of
  2) a white plate topped with meat , potatoes and vegetables . (p=0.000306)
how many N-gram overlaps of lower order it contains.
Blue cumulative 1-gram: 0.161348
Consider using lower n-gram order or use SmoothingFunction()
Blue cumulative 2-gram: 0.000000
  warnings.warn(_msg)
Glue score for this sentence: 0.09523809523809523
```

Document scope: Lab assignment – 4
Team: #5

```
Blue cumulative 2-gram: 0.000000
Glue score for this sentence: 0.09523809523809523
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)

Process finished with exit code 0
```

References:

- Show and Tell: A Neural Image Caption Generator -Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan
- https://arxiv.org/abs/1707.07998