

**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

# **CS5542 - BIGDATA ANALYTICS AND APPLICATIONS**

## **PROJECT REPORT 1**

### **CAPTION GENERATOR MODEL**

#### **1. Group Information:**

**Team #: 5**

**Team members:**

1. Koushik Katakam – 10
2. Saitejaswi Koppuravuri – 13
3. Venkata Lakshmi Korrapati – 14
4. Pavankumar Manchala – 16
5. Zakari Abdulmuhaymin Ahmad H – 29

#### **2. Abstract:**

Defining a caption for an image automatically has been booming in the arena of artificial intelligence which includes computer vision and Natural language processing. Using these two techniques a deep learning model must be developed in order to achieve the perfect caption generation for an image. Our project majorly focuses on providing the model which gives more accuracy by training and testing various models.

#### **3. Introduction:**

Caption generation is the challenging artificial intelligence problem using NLP technique and computer vision. It requires both images understanding and language model from the arena of Natural language processing. Indeed, a description must capture not only the objects contained in an image, but it also must express how these objects relate to each other as well as their attributes and the activities they are involved in. Our project has chosen the theme food where a model is trained with respect to food images and then testing is made with images and check the accuracy.

#### **4. Project Goals and Objective:**

##### **4.1 Motivation:**

Have you ever thought of generating a caption for an image? Yes, these days caption generation for an image has become an important task in the area of research of machine learning and Artificial Intelligence. No only captioning is a primary goal but predicting the objects in an image and express their

**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

relation in a process of natural language processing. The process of image captioning has been made little advanced on the advancement of neural networks.

#### 4.2 Significance/Uniqueness:

As this arena is emerging these days, there are quite a known number of applications which provide the image captioning for an image. Similar applications include Microsoft Seeing AI, Envision AI and couple more. Recently google has come up with an idea called Google Lookout especially for the disabled. But these applications have lacked a little accuracy and facing problems especially in particular lighting conditions. Especially, these applications in particular are developed for IOS.

#### 4.3 Objectives:

Our main motto is to make a visual world into an audible one. The main objective is to create a caption generator model for to understand images and text. Deep learning models are used in order to verify the best. Our major area of interest is on "FOOD". Various images of food are collected from the COCO Data set and train the model with these images. Finally, generating a perfect caption for an image.

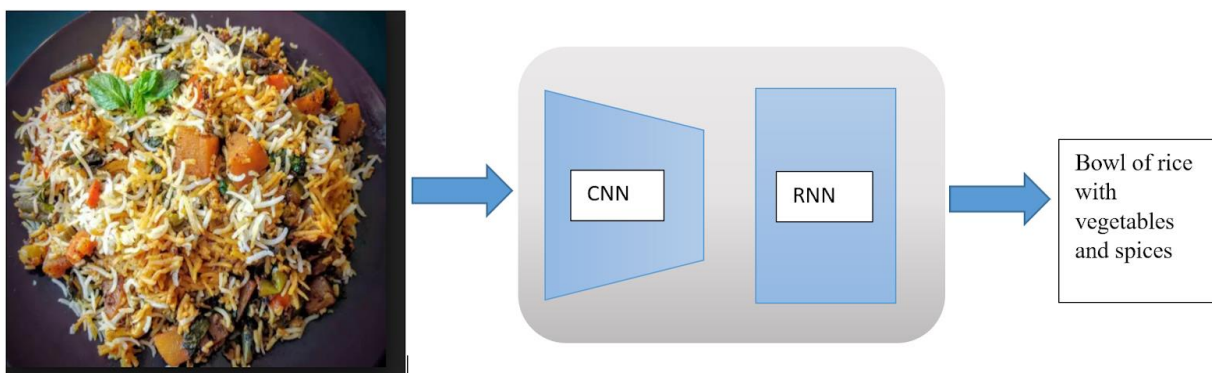
#### 4.4 System Features:

- The main feature is to generate a perfect caption for the image.
- Providing user, the captions with highest point of accuracy which can be done by training the implementing it with different models and choosing the correct optimizers when needed.
- Food is the major area of focus on which we are trying to collect images and train the model using Show, Attend and Tell methodology.

### 5. Proposed work:

#### 5.1 Proposed Models:

The workflow of the project is done using two different models namely CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Networks) models.



At present, we are with a simple workflow for the project i.e., when we give an image input to the model, it should process the image and predict it, resulting a text related to the image. For the most part, CNN was intended to outline information to a output variable. They were ended up being effective to the point, that they fit impeccably for a prediction problem including image information as an input. RNN were designed

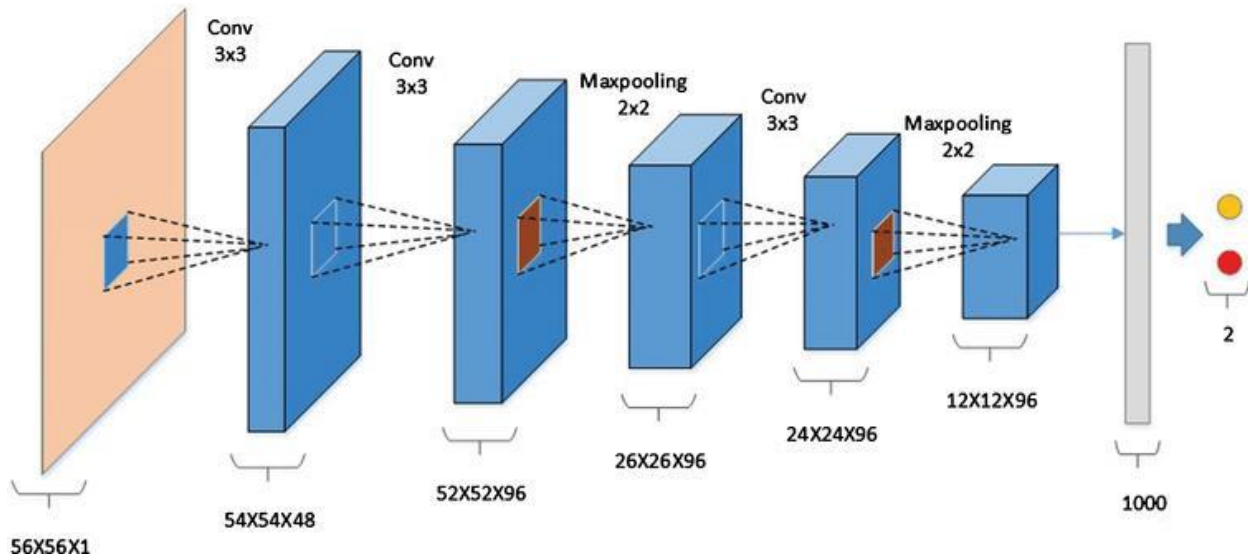
**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

to work with text analysis. Traditionally, RNN were difficult to train so in order to avoid this a combination of RNN model with LSTM (Long short-term memory) is used.

## 5.2 Network Diagram:



In fact, deep learning CNN models to train and test, each input image will go through a progression of convolution layers with filters, pooling layers and fully connected layer followed by applying a softmax function to classify an object with probabilistic values 0&1.

## Preprocessing of Image Captions Dataset using NLP

```

16 Chicken_count = 0
17 Coke_count = 0
18 Vegetable_count = 0
19
20 f = open('Extracted_tokens.txt','a')
21 fl = open('Extracted_urls.txt','a')
22 punctuations = "?!,.,,"
23 stop_words = set(stopwords.words('english'))
24 with open('SBU_captioned_photo_dataset_urls.txt') as file:
25     for line in file:
26         linecounter = linecounter + 1
27         tokens = word_tokenize(line)
28         for word in tokens:
29             if word in punctuations:
30                 tokens.remove(word)
31             if word in stop_words:
32                 tokens.remove(word)
33             #new_tokens.append(tokens)
34         for word in tokens:
35             lemmat_word = wordnet_lemmatizer.lemmatize(word, pos="v")
36             if 'Chicken' == lemmat_word:
37                 Chicken_count = Chicken_count + 1
38             f.writelines(line)
39             line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
40             with open('SBU_captioned_photo_dataset_urls.txt') as file:
41                 for line in file:
42                     for word in tokens:
43                         if word in punctuations:

```

Run: SIFT\_algorithm

File: "C:\Users\kiree\PycharmProjects\Lab1\venv\lib\site-packages\cv2\init.py", line 4, in <module>  
from .cv2 import \*

ImportError: DLL load failed: The specified module could not be found.

Process finished with exit code 1

Run: Run | TODO | Terminal | Python Console

Event Log

29:37 CRLF UTF-8 4 spaces

11:10 PM 2/22/2019

# Subject: Bigdata Analytics and Applications

## Document Scope: Project Report 1

### Team #: 5

Lab1 [C:\Users\kiree\PycharmProjects\Lab1] - ..word.py [Lab1] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

Project: Lab1 C:\Users\kiree\PycharmProjects\Lab1

- venv library root
- building\_1.jpg
- building\_2.JPG
- Extracted\_tokens.txt
- Extracted\_urls.txt
- NLPProcessing.py
- SBU\_captioned\_photo\_dataset\_captions.txt
- SBU\_captioned\_photo\_dataset\_urls.txt
- SIFT\_algorithm.py
- word.py

External Libraries

Scratches and Consoles

Run: SIFT\_algorithm x

```

31 if word in stop_words:
32     tokens.remove(word)
33 new_tokens = set(tokens)
34 for word in tokens:
35     lemmat_word = wordnet_lemmatizer.lemmatize(word, pos="v")
36     if 'Chicken' == lemmat_word:
37         Chicken_count = Chicken_count + 1
38         f.writelines(line)
39         line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
40         fl.writelines(line1)
41     if 'Coke' == lemmat_word:
42         Coke_count = Coke_count + 1
43         f.writelines(line)
44         line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
45         fl.writelines(line1)
46     if 'Vegetable' == lemmat_word:
47         Vegetable_count = Vegetable_count + 1
48         f.writelines(line)
49         line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
50         fl.writelines(line1)
51 objects = ('Chicken', 'Coke', 'Vegetable')
52 y_pos = np.arange(len(objects))
53 performance = [Chicken_count, Coke_count, Vegetable_count]
54 plot.bar(y_pos, performance, align='center', alpha=0.5)
55 with open('SBU_captioned_photo_dataset_urls.txt') as f:
56     for line in f:
57         for word in tokens:
58             if word in punctuations:

```

File "C:\Users\kiree\PycharmProjects\Lab1\venv\lib\site-packages\cv2\init.pyx", line 4, in <module>  
from .cv2 import \*  
ImportError: DLL load failed: The specified module could not be found.

Process finished with exit code 1

Run | TODO | Terminal | Python Console

Packages installed successfully: Installed packages: 'opencv-contrib-python' (today 9:46 PM)

29:37 CRLF UTF-8 4 spaces 11:10 PM 2/22/2019

Lab1 [C:\Users\kiree\PycharmProjects\Lab1] - ..word.py [Lab1] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

Project: Lab1 C:\Users\kiree\PycharmProjects\Lab1

- venv library root
- building\_1.jpg
- building\_2.JPG
- Extracted\_tokens.txt
- Extracted\_urls.txt
- NLPProcessing.py
- SBU\_captioned\_photo\_dataset\_captions.txt
- SBU\_captioned\_photo\_dataset\_urls.txt
- SIFT\_algorithm.py
- word.py

External Libraries

Scratches and Consoles

Run: SIFT\_algorithm x

```

40 line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
41 fl.writelines(line1)
42 if 'Coke' == lemmat_word:
43     Coke_count = Coke_count + 1
44     f.writelines(line)
45     line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
46     fl.writelines(line1)
47 if 'Vegetable' == lemmat_word:
48     Vegetable_count = Vegetable_count + 1
49     f.writelines(line)
50     line1 = linecache.getline('SBU_captioned_photo_dataset_urls.txt', linecounter)
51     fl.writelines(line1)
52 objects = ('Chicken', 'Coke', 'Vegetable')
53 y_pos = np.arange(len(objects))
54 performance = [Chicken_count, Coke_count, Vegetable_count]
55 plot.bar(y_pos, performance, align='center', alpha=0.5)
56 plot.xticks(y_pos, objects)
57 plot.show()

```

File "C:\Users\kiree\PycharmProjects\Lab1\venv\lib\site-packages\cv2\init.pyx", line 4, in <module>  
from .cv2 import \*  
ImportError: DLL load failed: The specified module could not be found.

Process finished with exit code 1

Run | TODO | Terminal | Python Console

Packages installed successfully: Installed packages: 'opencv-contrib-python' (today 9:46 PM)

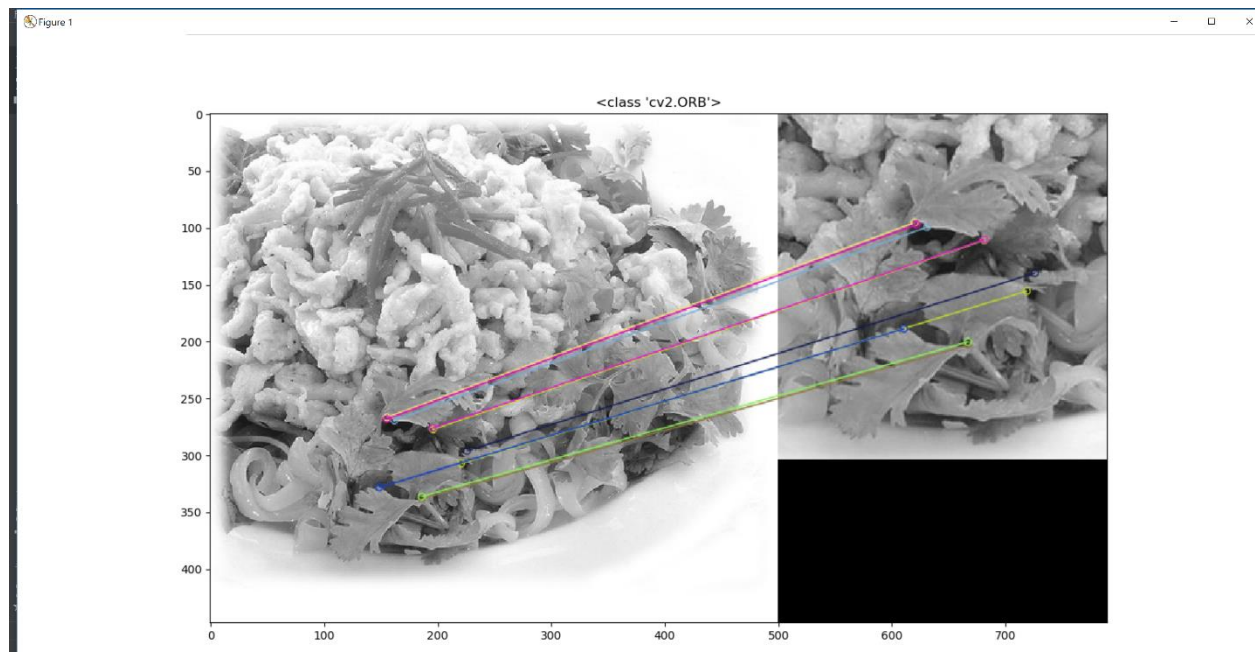
29:37 CRLF UTF-8 4 spaces 11:11 PM 2/22/2019

**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

### Feature Generation using SIFT:

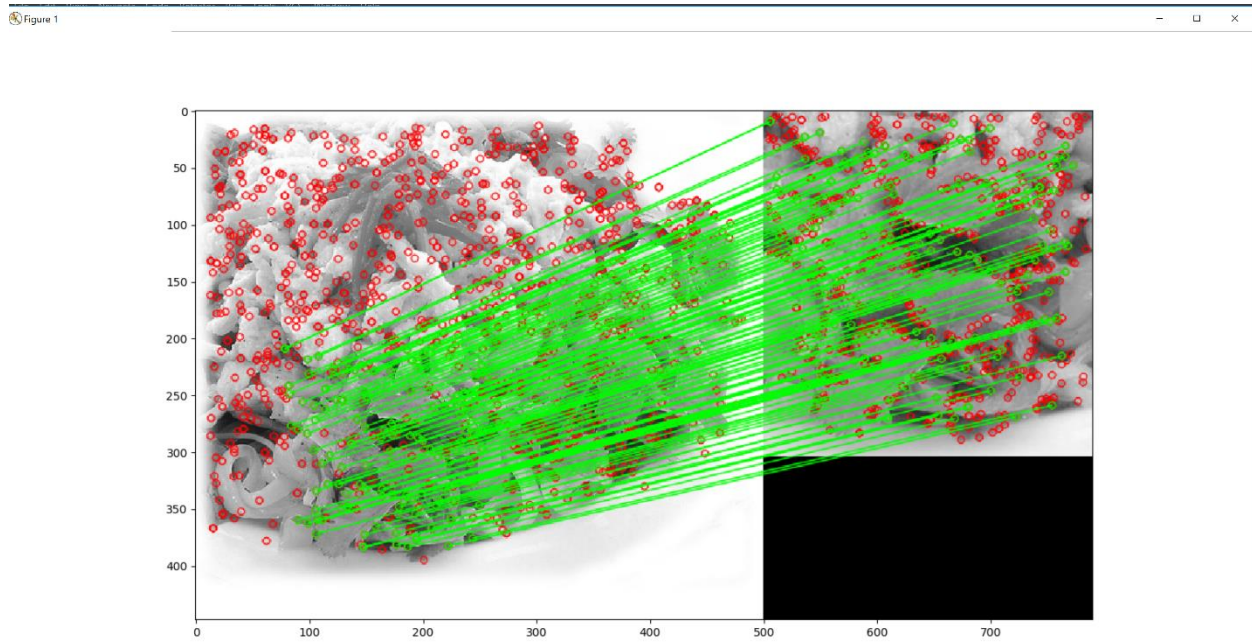
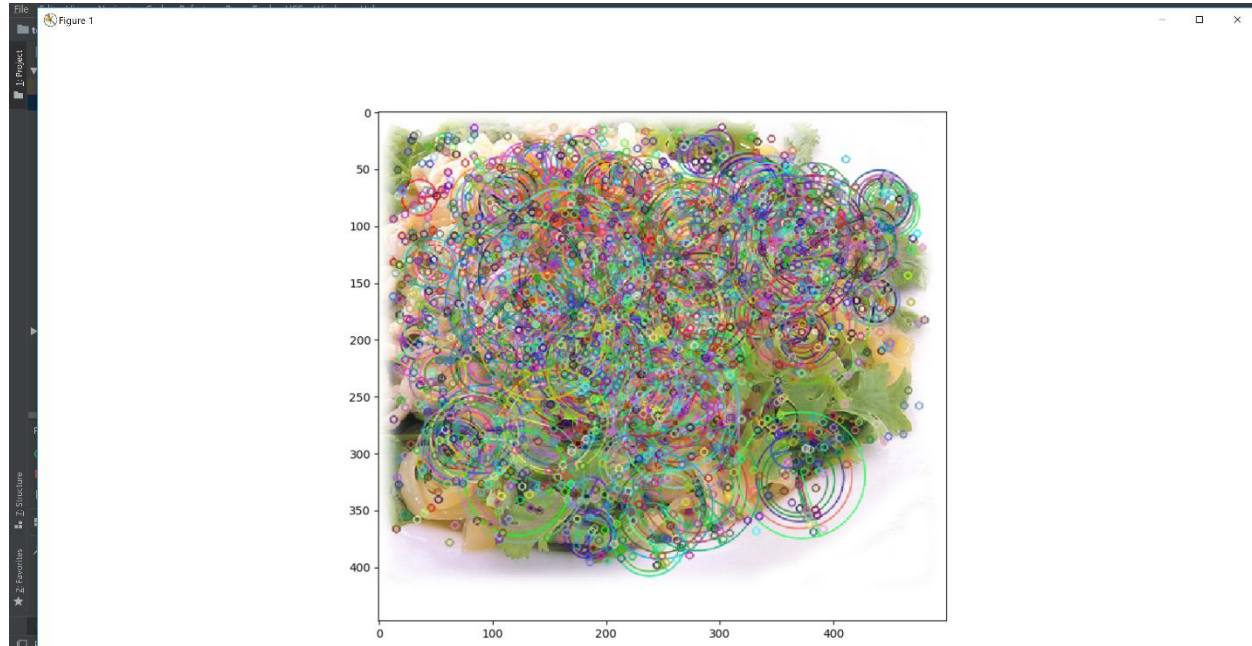




**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

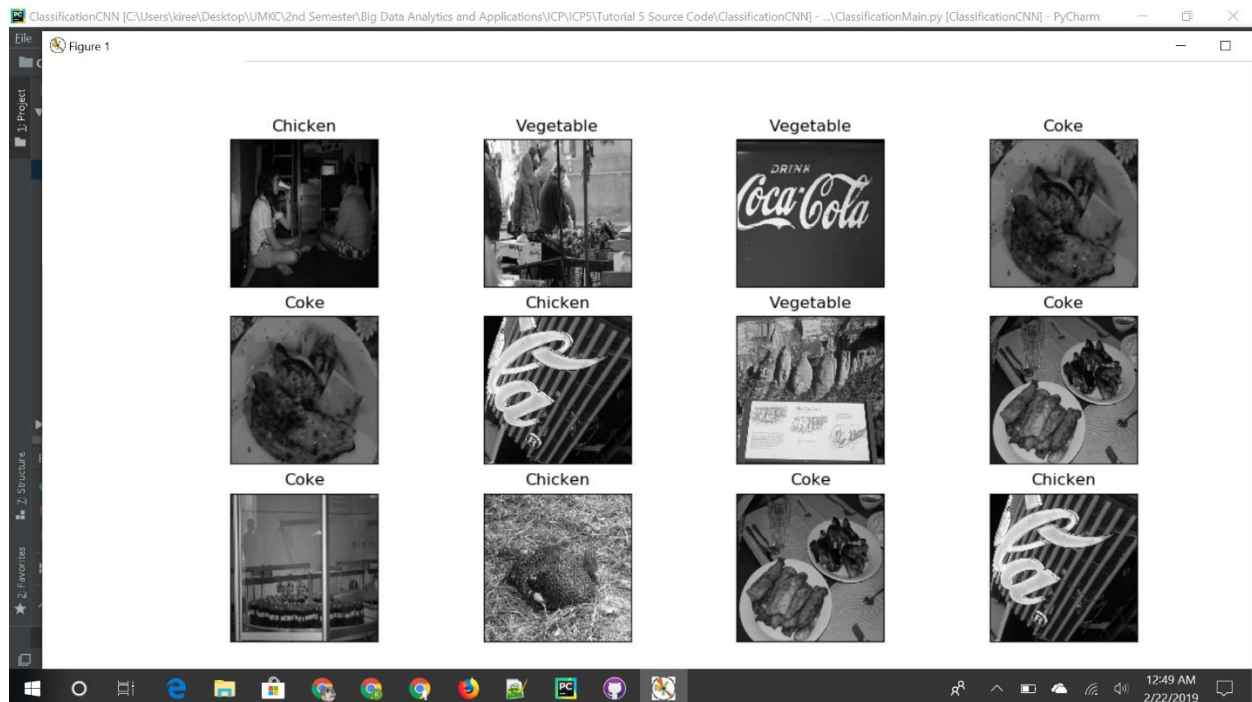


**Subject:** Bigdata Analytics and Applications

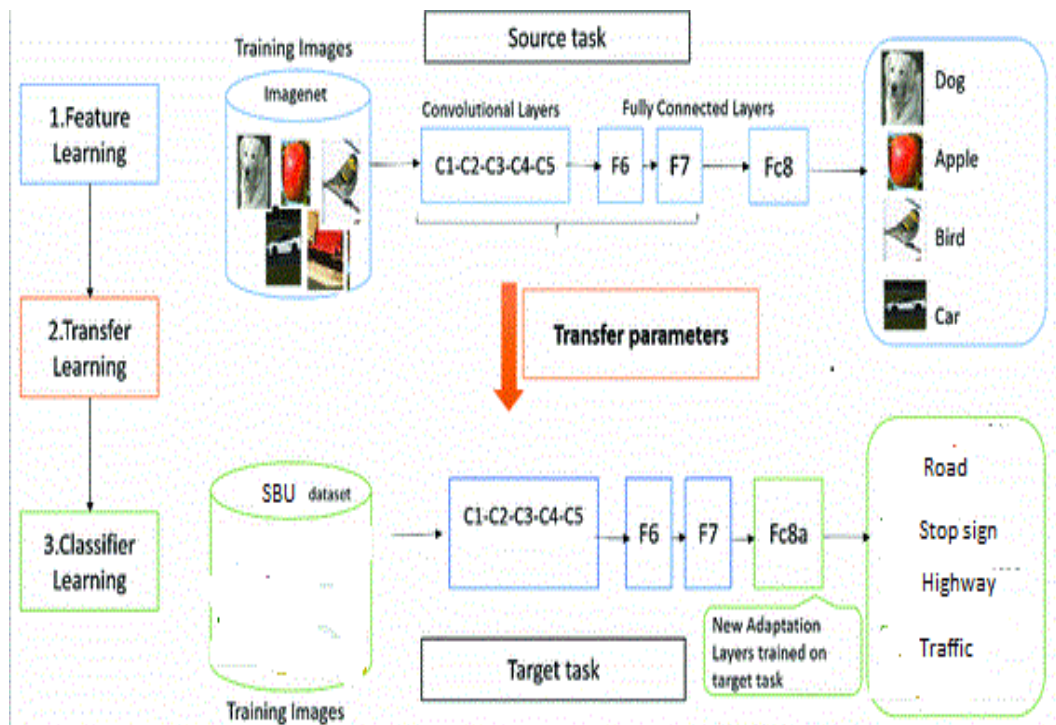
**Document Scope:** Project Report 1

**Team #:** 5

### Image classification using CNN model:



### Image classification using pretrained model



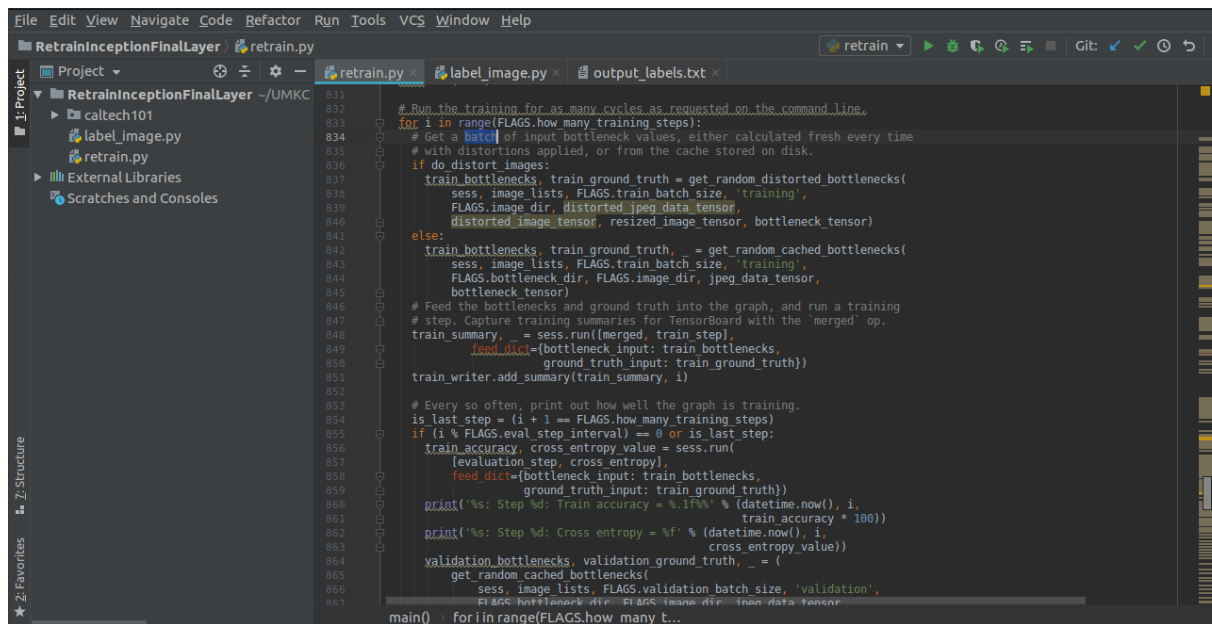




**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5



```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
RetrainInceptionFinalLayer / retrain.py
Project
RetrainInceptionFinalLayer ~/UMKC
  caltech101
  label_image.py
  retrain.py
External Libraries
Scratches and Consoles
retrain.py
831 # Run the training for as many cycles as requested on the command line.
832 for i in range(FLAGS.how_many_training_steps):
833     # Get a batch of input bottleneck values, either calculated fresh every time
834     # with distortions applied, or from the cache stored on disk.
835     if do_distort_images:
836         train_bottlenecks, train_ground_truth = get_random_distorted_bottlenecks(
837             sess, image_lists, FLAGS.train_batch_size, 'training',
838             FLAGS.image_dir, distorted_jpeg_data_tensor,
839             distorted_image_tensor, resized_image_tensor, bottleneck_tensor)
840     else:
841         train_bottlenecks, train_ground_truth, _ = get_random_cached_bottlenecks(
842             sess, image_lists, FLAGS.train_batch_size, 'training',
843             FLAGS.bottleneck_dir, FLAGS.image_dir, jpeg_data_tensor,
844             bottleneck_tensor)
845     # Feed the bottlenecks and ground truth into the graph, and run a training
846     # step. Capture training summaries for TensorBoard with the 'merged' op.
847     train_summary = sess.run([merged, train_step],
848                             feed_dict={bottleneck_input: train_bottlenecks,
849                                       ground_truth_input: train_ground_truth})
850     train_writer.add_summary(train_summary, i)
851
852     # Every so often, print out how well the graph is training.
853     is_last_step = (i + 1 == FLAGS.how_many_training_steps)
854     if (i % FLAGS.eval_step_interval) == 0 or is_last_step:
855         train_accuracy, cross_entropy_value = sess.run(
856             [evaluation_step, cross_entropy],
857             feed_dict={bottleneck_input: train_bottlenecks,
858                       ground_truth_input: train_ground_truth})
859         print('%s: Step %d: Train accuracy = %.1f%%' % (datetime.now(), i,
860                                                       train_accuracy * 100))
861         print('%s: Step %d: Cross entropy = %f' % (datetime.now(), i,
862                                                  cross_entropy_value))
863         validation_bottlenecks, validation_ground_truth, _ = (
864             get_random_cached_bottlenecks(
865                 sess, image_lists, FLAGS.validation_batch_size, 'validation',
866                 FLAGS.bottleneck_dir, FLAGS.image_dir, jpeg_data_tensor,
867                 validation_image_tensor, validation_bottleneck_tensor)
868             )
869         validation_accuracy, validation_cross_entropy = sess.run(
870             [evaluation_step, cross_entropy],
871             feed_dict={bottleneck_input: validation_bottlenecks,
872                       ground_truth_input: validation_ground_truth})
873         print('%s: Step %d: Validation accuracy = %.1f%% (cross entropy: %f)' %
874               (datetime.now(), i, validation_accuracy * 100, validation_cross_entropy))
875
876     main()
877 for i in range(FLAGS.how_many_training_steps):

```

**Github link:** [https://github.com/SaitejaswiK/CS5542\\_BigdataAnalyticsProject](https://github.com/SaitejaswiK/CS5542_BigdataAnalyticsProject)

## 7. Evaluation and results:

### 7.1 Datasets:

We have used SBU dataset and Flickr 8k dataset for implementation till date.

The links for the datasets are as follows:

SBU dataset: <http://www.cs.virginia.edu/~vicente/sbucaptions/>

Flickr\_8kdataset: [http://nlp.cs.illinois.edu/HockenmaierGroup/Framing\\_Image\\_Description/KCCA.html](http://nlp.cs.illinois.edu/HockenmaierGroup/Framing_Image_Description/KCCA.html)

### 7.2 System specifications:

- Ram - 8GB or more
- HDD/SSD - 256 GB
- Processor - i5 or more(windows)
- GPU - 2GB or more

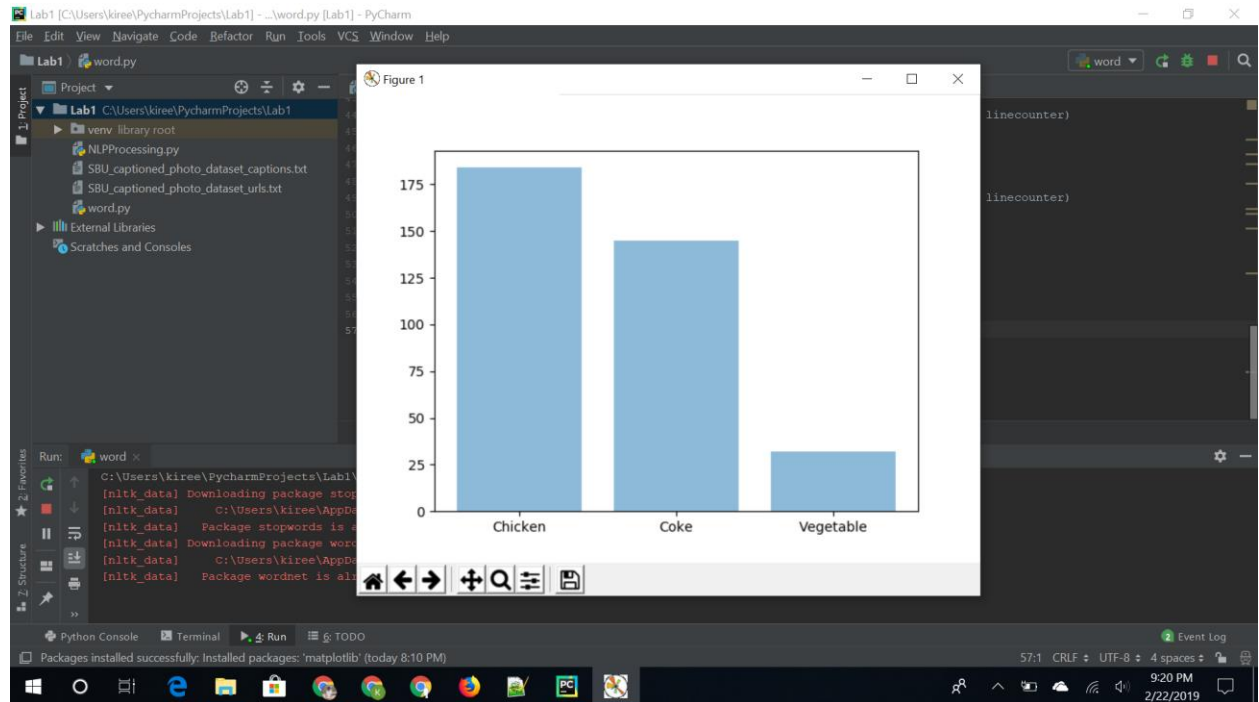
**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

### 7.3 Evaluation and Results:

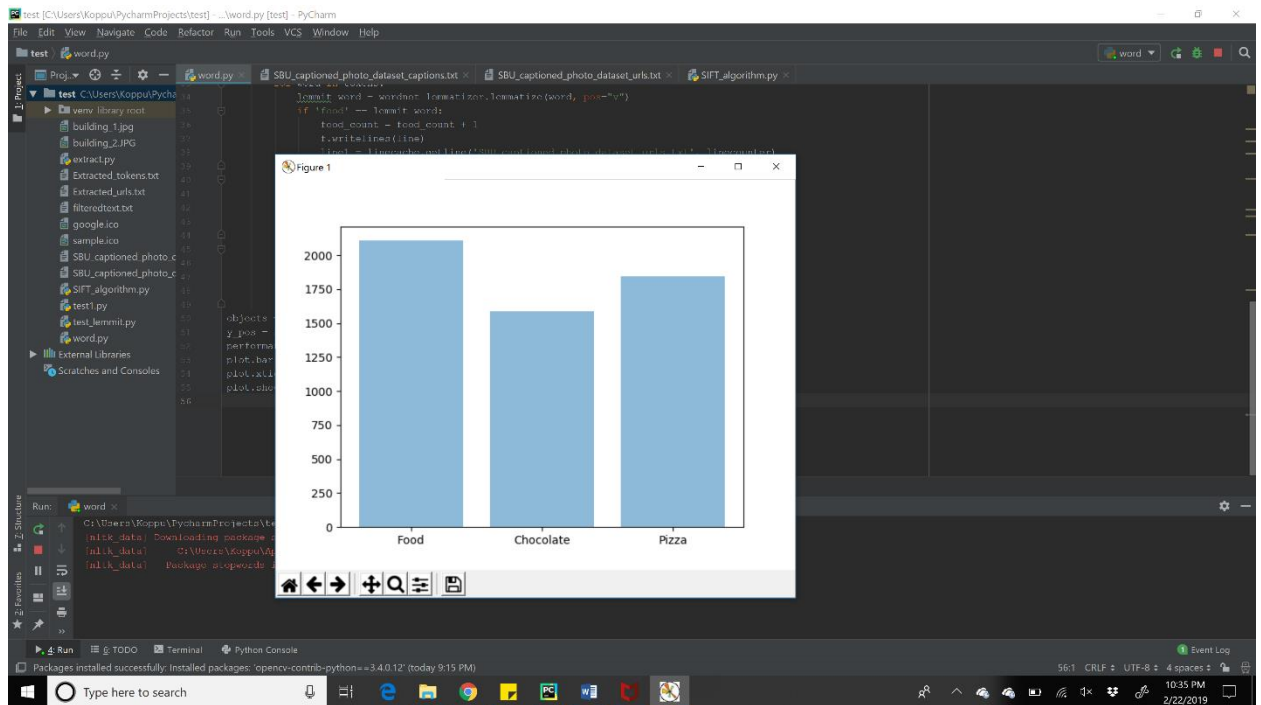
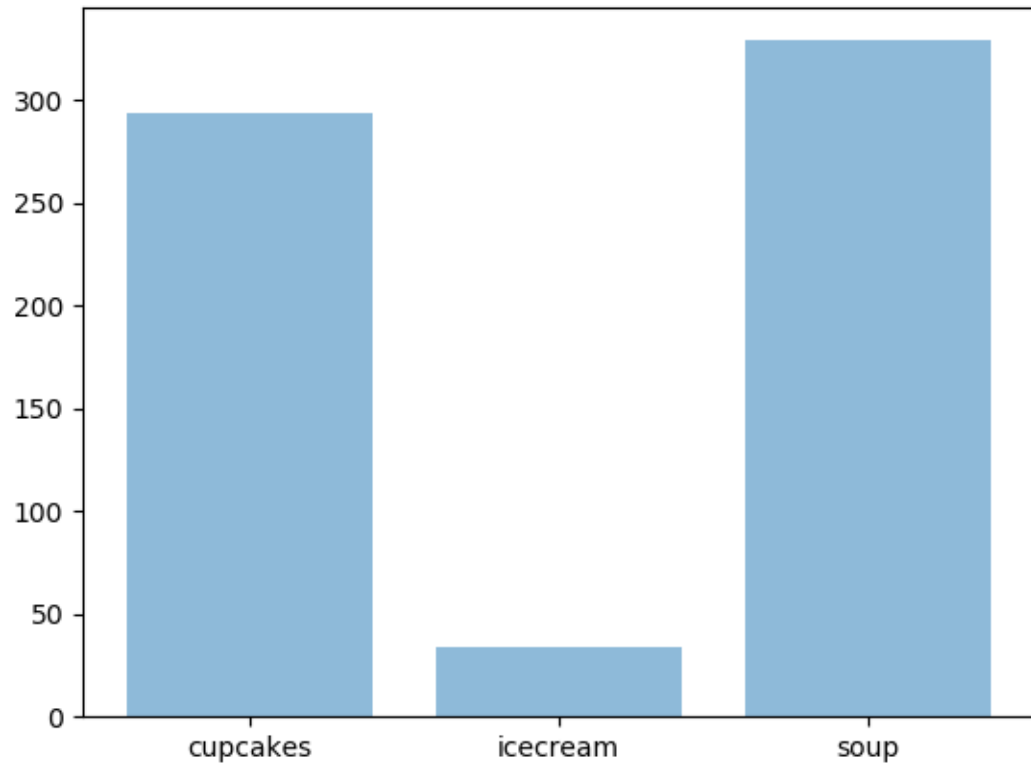
#### Dataset statistics:



**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5



**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

## 7.4 Hyperparameters:

- Number of Epochs - 400
- Batch Size - 60
- Activation Function - Relu
- Number of hidden layers and units - 3
- Weight Initialization - 32 , 64

## 7.5 Classification Accuracy based on CNN model and pretrained Models

```

1 import tensorflow as tf
2
3 image_path = 'data/flower_photos/roses/269037241_07fcaf56a_m.jpg'
4 image_path = 'caltech101/data/train/chicken/146960169_1c1c025a41.jpg'
5 # image_path = '676728-highwaybowl.jpg'
6
7 # Read in the image data
8 image_data = tf.gfile.FastGFile(image_path, 'rb').read()
9
10 # Loads label file, strips off carriage return
11 label_lines = [line.rstrip() for line
12               in tf.gfile.GFile("caltech101/data/output_labels.txt")]
13
14 # Unrolls graph from file
15 with tf.gfile.FastGFile("caltech101/data/output_graph.pb", 'rb') as f:
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Run: label\_image

```

C:\Users\kiree\PycharmProjects\test\venv\Scripts\python.exe "C:/Users/kiree/Desktop/UMKC/2nd Semester/Big Data Analytics and Applications/ICP/ICP5/Tutorial 5 Source Code/RetrainInceptionFinalLayer/1
WARNING:tensorflow:From C:/Users/kiree/Desktop/UMKC/2nd Semester/Big Data Analytics and Applications/ICP/ICP5/Tutorial 5 Source Code/RetrainInceptionFinalLayer/1
Instructions for updating:
Use tf.gfile.GFile.
2019-02-21 22:41:45.922733: W tensorflow/core/framework/op_def_util.cc:355] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef
2019-02-21 22:41:46.221505: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use
chicken (score = 0.97700)
coke (score = 0.01383)
vegetable (score = 0.00917)

Process finished with exit code 0

```

## Conclusion:

- To create a caption generator model
- To understand various food images, train and test accordingly.
- Calculate the accuracy between models.

## Related Papers:

<http://homepages.inf.ed.ac.uk/keller/papers/jair16.pdf>

<https://arxiv.org/pdf/1810.04020>

<https://arxiv.org/pdf/1805.09137>

[www.aclweb.org/anthology/P18-1238](http://www.aclweb.org/anthology/P18-1238)



**Subject:** Bigdata Analytics and Applications

**Document Scope:** Project Report 1

**Team #:** 5

**Show Attend and Tell Model:** Using this methodology we are trying to implement and check whether the model gives us the pretty quiet accuracy.

**Paper Link:** <http://proceedings.mlr.press/v37/xuc15.pdf>

**Bibliography:**

<http://proceedings.mlr.press/v37/xuc15.pdf>

<https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/>