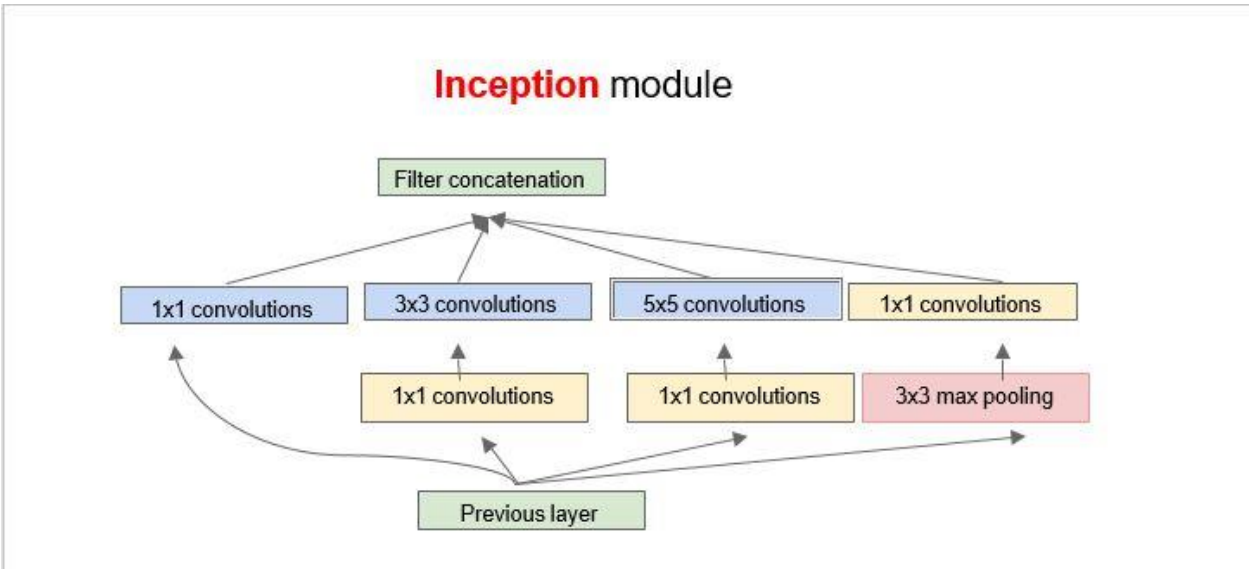
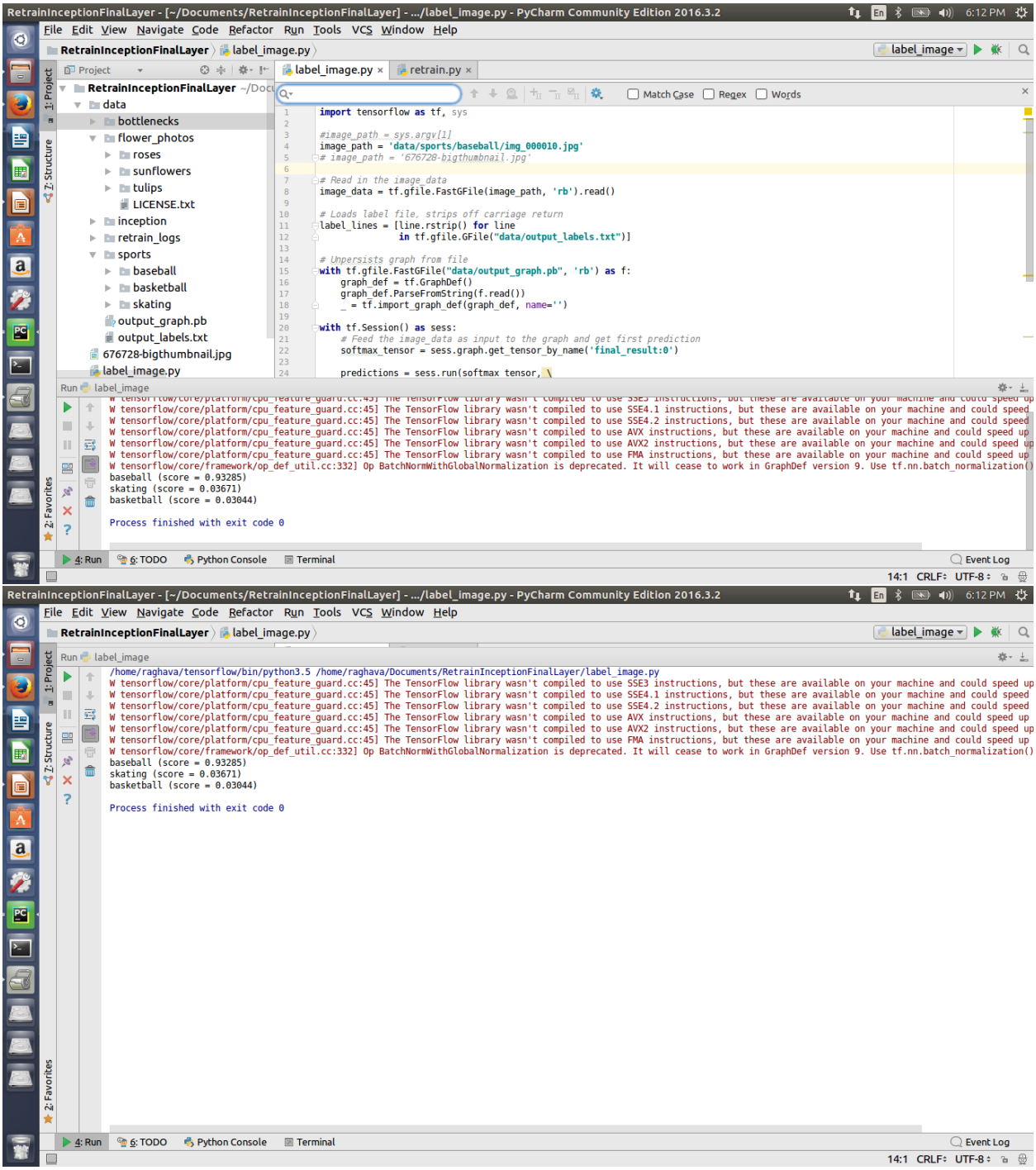
CS5542 Big Data Analytics and App
LAB ASSIGNMENT #10

RetrainInception

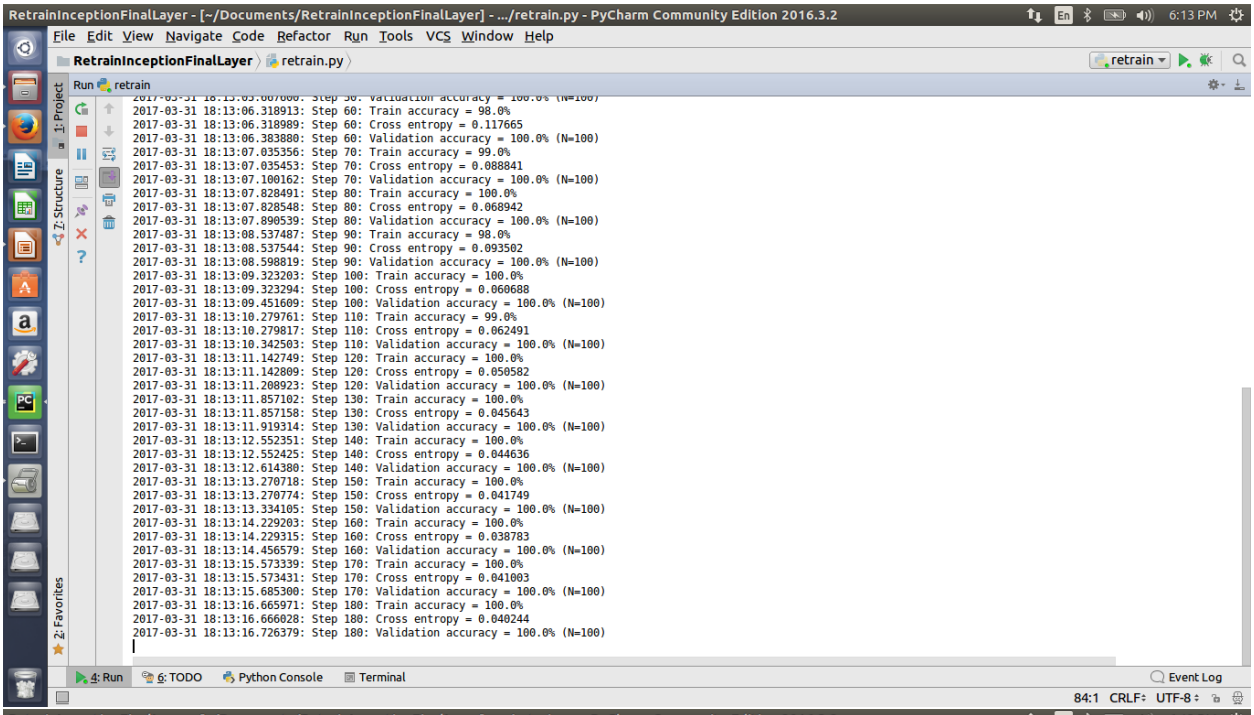
This program/model is developed by google. The google inception module is:

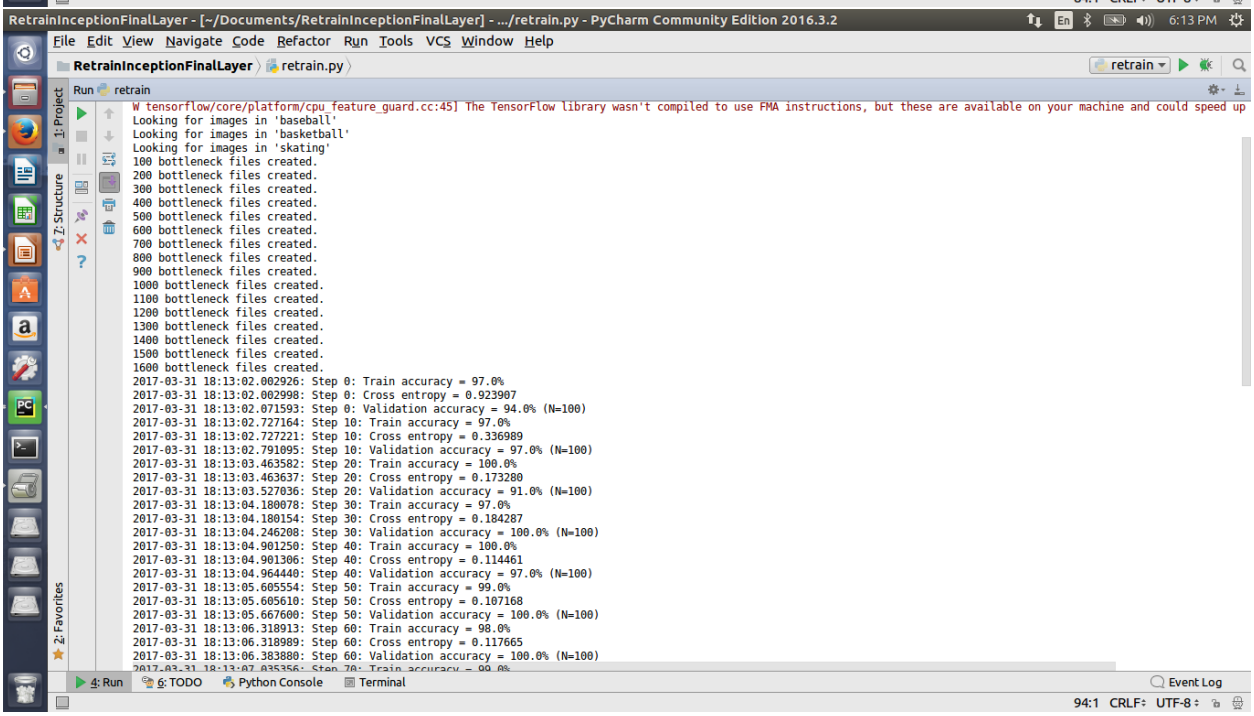


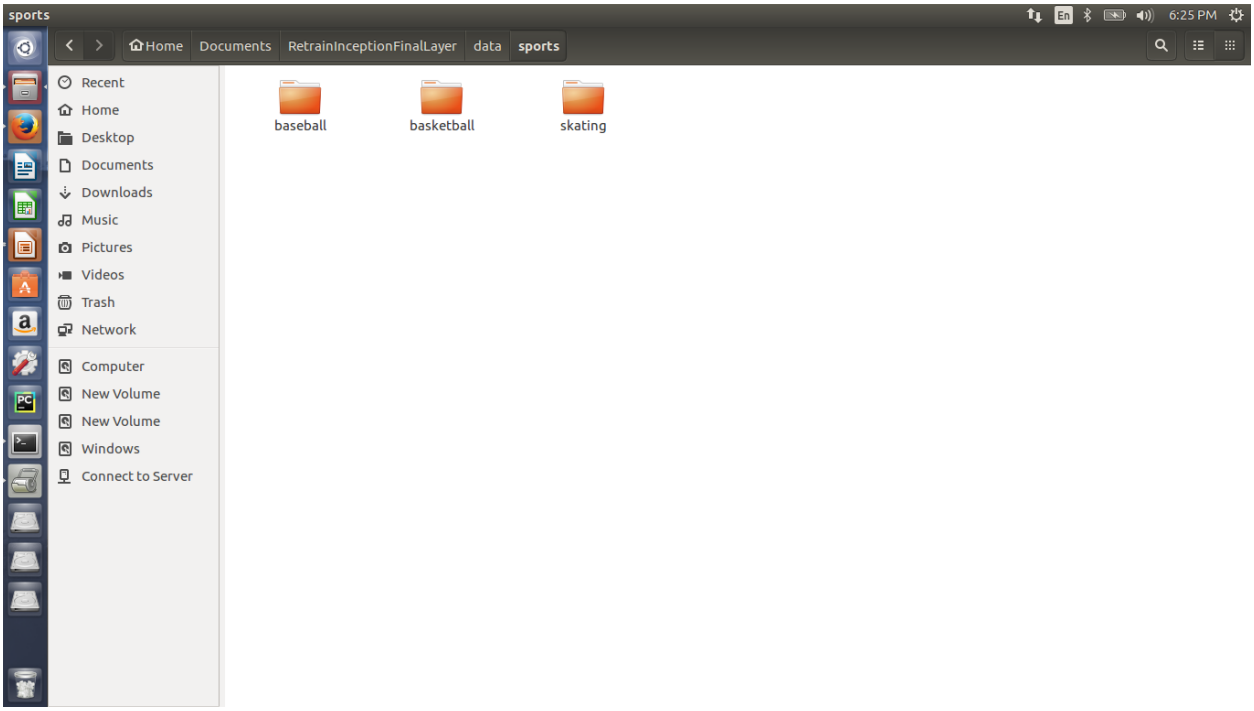
For this program, for the lab purpose sports image data set is taken, around 700 images were taken for this program to generate bottlenecks and the output for purpose of the other program. Three sets of images **baseball basketball skating** were taken, around a total of 700+ images were taken for this. The accuracy turned out to be 93.6% for some given image (as seen from the below screenshots). Here are the screenshots for the program and the data:


RetrainInceptionFinalLayer - [~/Documents/RetrainInceptionFinalLayer] - .../label_image.py - PyCharm Community Edition 2016.3.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
RetrainInceptionFinalLayer
label_image.py
label_image
Project
label_image.py
retrain.py
Match Case
Regex
Words
import tensorflow as tf, sys
#image_path = sys.argv[1]
image_path = 'data/sports/baseball/img_000010.jpg'
# image_path = '676728-bigthumbnail.jpg'
# Read in the image_data
image_data = tf.gfile.FastGFile(image_path, 'rb').read()
# Loads label file, strips off carriage return
label_lines = [line.rstrip() for line
in tf.gfile.GFile("data/output_labels.txt")]
Unpersists graph from file
with tf.gfile.FastGFile("data/output_graph.pb", 'rb') as f:
graph_def = tf.GraphDef()
graph_def.ParseFromString(f.read())
_ = tf.import_graph_def(graph_def, name='')
with tf.Session() as sess:
# Feed the image_data as input to the graph and get first prediction
softmax_tensor = sess.graph.get_tensor_by_name('final_result:0')
predictions = sess.run(softmax_tensor, \
data
bottlenecks
flower_photos
roses
sunflowers
tulips
LICENSE.txt
inception
retrain_logs
sports
baseball
basketball
skating
output_graph.pb
output_labels.txt
676728-bigthumbnail.jpg
label_image.py
Run label_image
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch_normalization()
baseball (score = 0.93285)
skating (score = 0.03671)
basketball (score = 0.03044)
Process finished with exit code 0
4: Run
6: TODO
Python Console
Terminal
Event Log
14:1 CRLF: UTF-8:
1: Project
7: Structure
2: Favorites

RetrainInceptionFinalLayer - [~/Documents/RetrainInceptionFinalLayer] - .../label_image.py - PyCharm Community Edition 2016.3.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
RetrainInceptionFinalLayer
label_image.py
label_image
Run label_image
/home/raghava/tensorflow/bin/python3.5 /home/raghava/Documents/RetrainInceptionFinalLayer/label_image.py
W tensorflow/core/framework/op_def_util.cc:332] Op BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch_normalization()
baseball (score = 0.93285)
skating (score = 0.03671)
basketball (score = 0.03044)
Process finished with exit code 0
4: Run
6: TODO
Python Console
Terminal
Event Log
14:1 CRLF: UTF-8:
1: Project
7: Structure
2: Favorites


RetrainInceptionFinalLayer - [~/Documents/RetrainInceptionFinalLayer] - .../retrain.py - PyCharm Community Edition 2016.3.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
RetrainInceptionFinalLayer retrain.py
retrain
Run retrain
2017-03-31 18:13:06.318913: Step 60: Train accuracy = 98.0%
2017-03-31 18:13:06.318989: Step 60: Cross entropy = 0.117665
2017-03-31 18:13:06.383880: Step 60: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:07.035356: Step 70: Train accuracy = 99.0%
2017-03-31 18:13:07.035453: Step 70: Cross entropy = 0.088841
2017-03-31 18:13:07.100162: Step 70: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:07.828491: Step 80: Train accuracy = 100.0%
2017-03-31 18:13:07.828548: Step 80: Cross entropy = 0.068942
2017-03-31 18:13:07.890539: Step 80: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:08.537487: Step 90: Train accuracy = 98.0%
2017-03-31 18:13:08.537544: Step 90: Cross entropy = 0.093502
2017-03-31 18:13:08.598819: Step 90: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:09.323203: Step 100: Train accuracy = 100.0%
2017-03-31 18:13:09.323294: Step 100: Cross entropy = 0.060688
2017-03-31 18:13:09.451609: Step 100: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:10.279761: Step 110: Train accuracy = 99.0%
2017-03-31 18:13:10.279817: Step 110: Cross entropy = 0.062491
2017-03-31 18:13:10.342503: Step 110: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:11.142749: Step 120: Train accuracy = 100.0%
2017-03-31 18:13:11.142809: Step 120: Cross entropy = 0.050582
2017-03-31 18:13:11.208923: Step 120: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:11.857102: Step 130: Train accuracy = 100.0%
2017-03-31 18:13:11.857158: Step 130: Cross entropy = 0.045643
2017-03-31 18:13:11.919314: Step 130: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:12.552351: Step 140: Train accuracy = 100.0%
2017-03-31 18:13:12.552425: Step 140: Cross entropy = 0.044636
2017-03-31 18:13:12.614380: Step 140: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:13.270718: Step 150: Train accuracy = 100.0%
2017-03-31 18:13:13.270774: Step 150: Cross entropy = 0.041749
2017-03-31 18:13:13.334105: Step 150: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:14.229203: Step 160: Train accuracy = 100.0%
2017-03-31 18:13:14.229315: Step 160: Cross entropy = 0.038783
2017-03-31 18:13:14.456579: Step 160: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:15.573339: Step 170: Train accuracy = 100.0%
2017-03-31 18:13:15.573431: Step 170: Cross entropy = 0.041003
2017-03-31 18:13:15.685300: Step 170: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:16.665971: Step 180: Train accuracy = 100.0%
2017-03-31 18:13:16.666028: Step 180: Cross entropy = 0.040244
2017-03-31 18:13:16.726379: Step 180: Validation accuracy = 100.0% (N=100)
4: Run 6: TODO Python Console Terminal Event Log
84:1 CRLF: UTF-8:


RetrainInceptionFinalLayer - [~/Documents/RetrainInceptionFinalLayer] - .../retrain.py - PyCharm Community Edition 2016.3.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
RetrainInceptionFinalLayer retrain.py
retrain
Run retrain
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up
Looking for images in 'baseball'
Looking for images in 'basketball'
Looking for images in 'skating'
100 bottleneck files created.
200 bottleneck files created.
300 bottleneck files created.
400 bottleneck files created.
500 bottleneck files created.
600 bottleneck files created.
700 bottleneck files created.
800 bottleneck files created.
900 bottleneck files created.
1000 bottleneck files created.
1100 bottleneck files created.
1200 bottleneck files created.
1300 bottleneck files created.
1400 bottleneck files created.
1500 bottleneck files created.
1600 bottleneck files created.
2017-03-31 18:13:02.002926: Step 0: Train accuracy = 97.0%
2017-03-31 18:13:02.002998: Step 0: Cross entropy = 0.923907
2017-03-31 18:13:02.071593: Step 0: Validation accuracy = 94.0% (N=100)
2017-03-31 18:13:02.727164: Step 10: Train accuracy = 97.0%
2017-03-31 18:13:02.727221: Step 10: Cross entropy = 0.336989
2017-03-31 18:13:02.791095: Step 10: Validation accuracy = 97.0% (N=100)
2017-03-31 18:13:03.463582: Step 20: Train accuracy = 100.0%
2017-03-31 18:13:03.463637: Step 20: Cross entropy = 0.173280
2017-03-31 18:13:03.527036: Step 20: Validation accuracy = 91.0% (N=100)
2017-03-31 18:13:04.180078: Step 30: Train accuracy = 97.0%
2017-03-31 18:13:04.180154: Step 30: Cross entropy = 0.184287
2017-03-31 18:13:04.246208: Step 30: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:04.901250: Step 40: Train accuracy = 100.0%
2017-03-31 18:13:04.901306: Step 40: Cross entropy = 0.114461
2017-03-31 18:13:04.964440: Step 40: Validation accuracy = 97.0% (N=100)
2017-03-31 18:13:05.605554: Step 50: Train accuracy = 99.0%
2017-03-31 18:13:05.605610: Step 50: Cross entropy = 0.107168
2017-03-31 18:13:05.667600: Step 50: Validation accuracy = 100.0% (N=100)
2017-03-31 18:13:06.318913: Step 60: Train accuracy = 98.0%
2017-03-31 18:13:06.318989: Step 60: Cross entropy = 0.117665
2017-03-31 18:13:06.383880: Step 60: Validation accuracy = 100.0% (N=100)
4: Run 6: TODO Python Console Terminal Event Log
94:1 CRLF: UTF-8:



TensorFlowCNNWeb

As part of this program, for display/predicting images via WEB UI, the **output.pb** and **outputlabel.txt** files were taken as input to the second program. Flask is installed through the pycharm terminal and run using the WEB Url provided. Here are the screenshots:

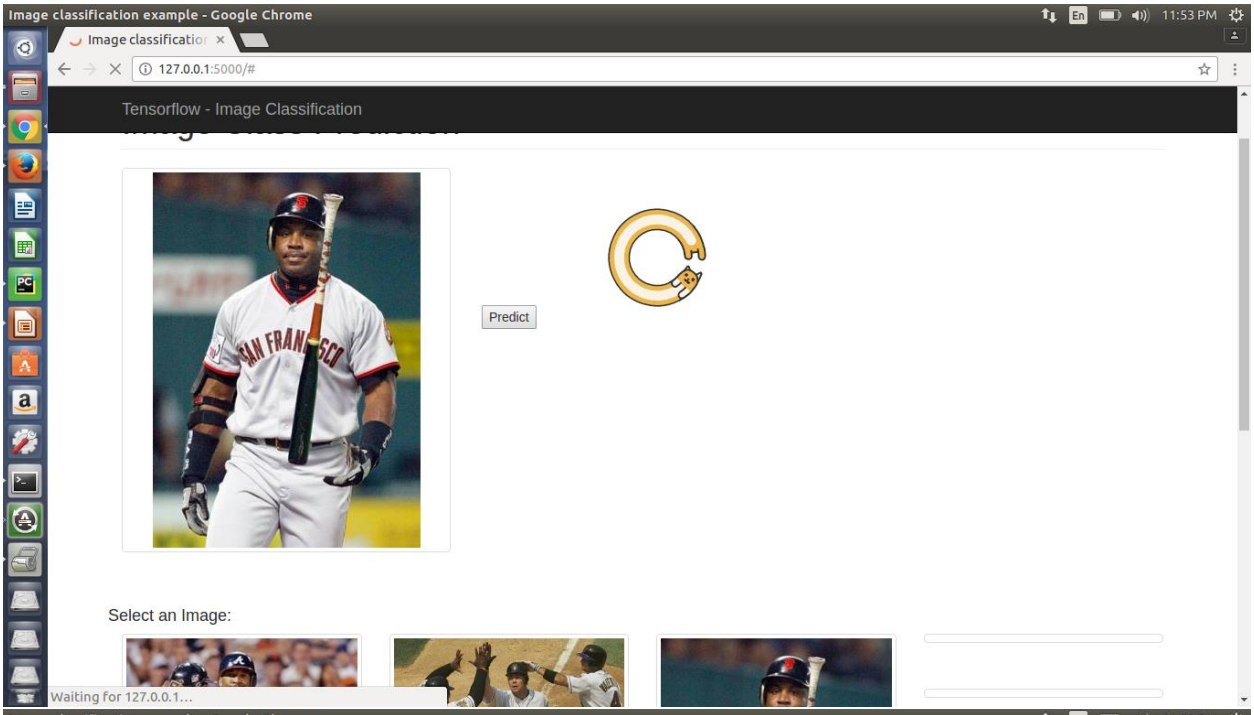


Image classification example - Google Chrome
11:53 PM
Image classificatio
127.0.0.1:5000/#
Tensorflow - Image Classification
Predict
Select an Image:
Waiting for 127.0.0.1...

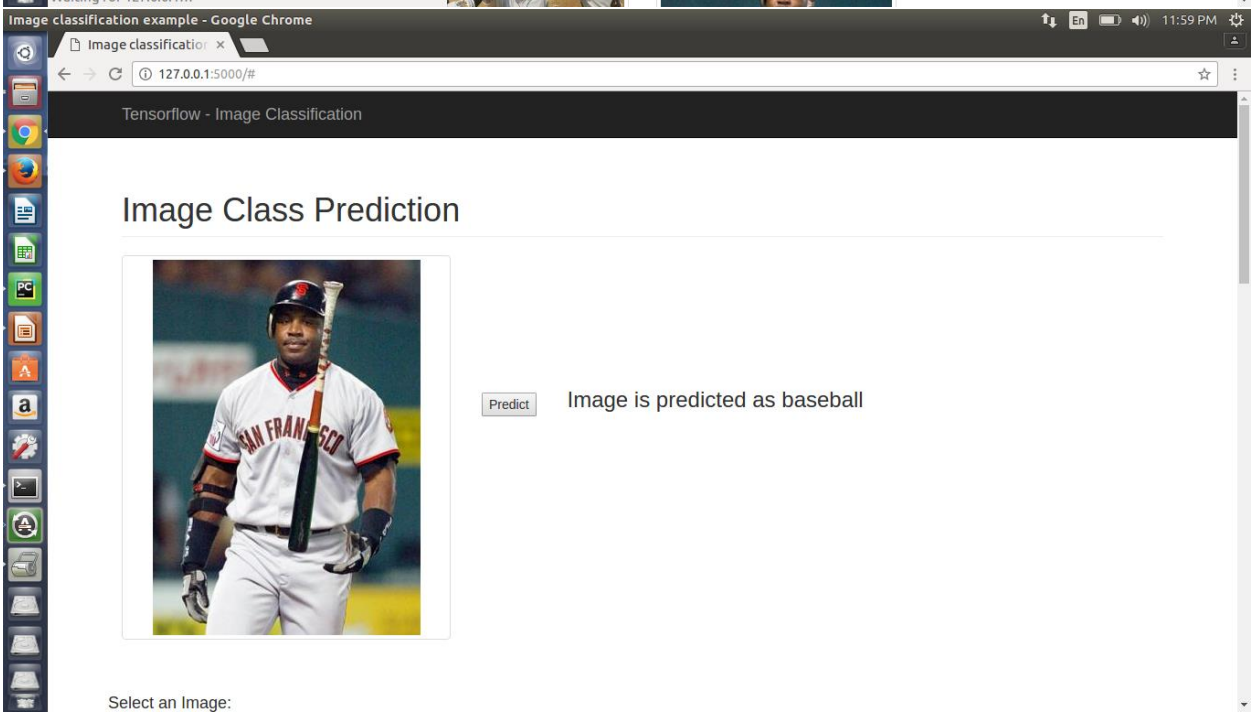
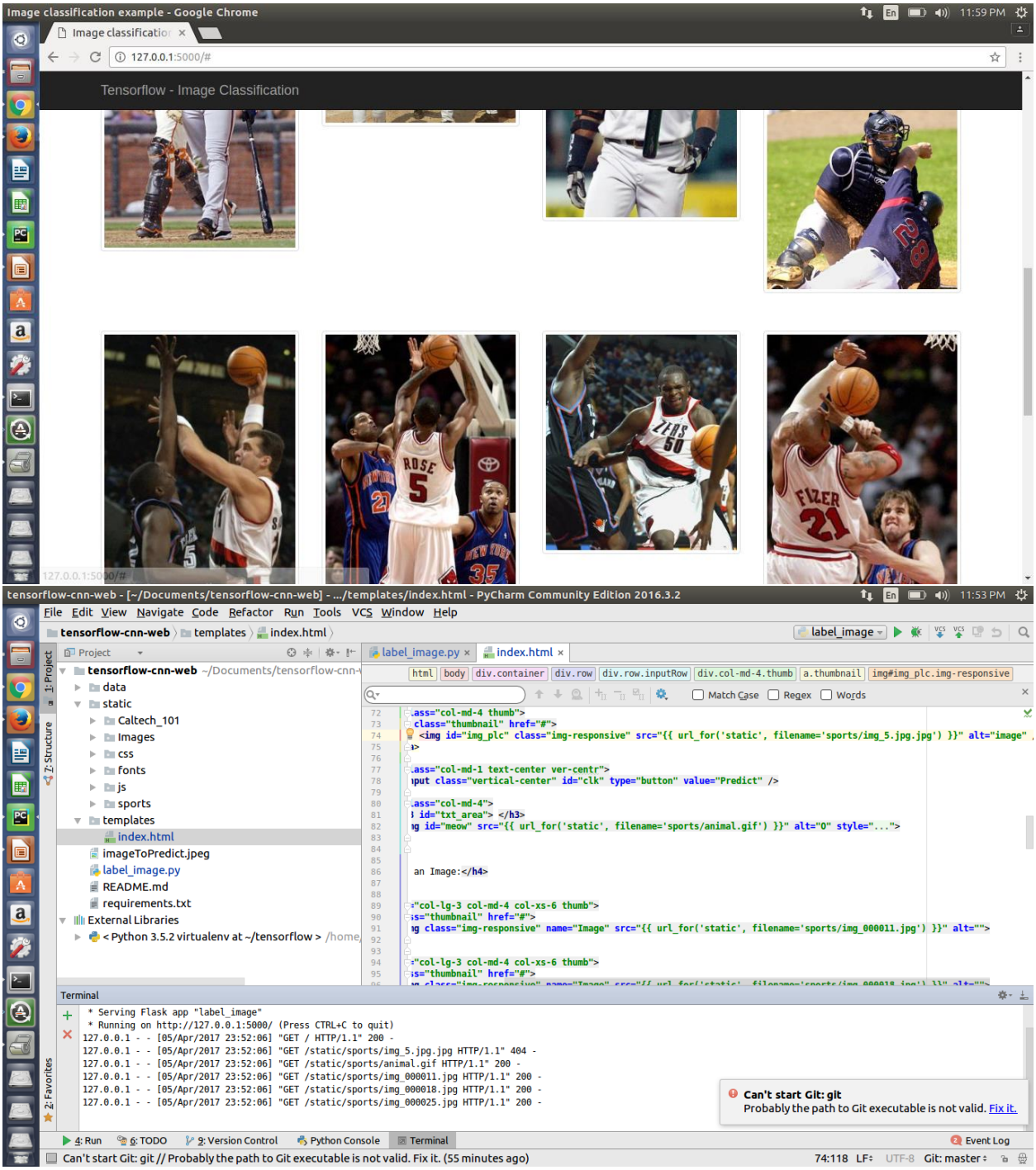


Image classification example - Google Chrome
11:59 PM
Image classificatio
127.0.0.1:5000/#
Tensorflow - Image Classification
Image Class Prediction
Predict
Image is predicted as baseball
Select an Image:



The image is predicted accurately as shown above in the screenshots.