



Bapi Reddy

[Follow](#)

Feb 19 · 4 min read

Complete your account registration. [Send verification email.](#)

Real time Image Classifier on Raspberry pi using Inception Framework

Hello every one today we are going to learn how to perform the Image classification on Raspberry pi, which you can use classify any set of images. Well lets start it out now, before diving deep here are learning points out of this tutorial

- Implementation of Tensorflow on Raspberry Pi
- Training the Google's Inception model for your own set of Images
- Porting the Classifier to Raspberry pi for real time classification

Here is objective in simple terms:

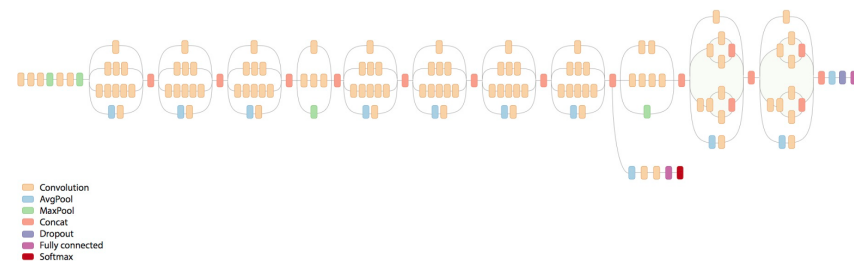
“Build a image classifier on Raspberry pi using Tensorflow”

Basic steps of how we are going to proceed

1. Collect the Training Image data for classification

2. Train the Inception image classifier using our new data
3. Porting the trained model to Raspberry pi
4. Create the Image classifier on Raspberry pi

Concept: The main concept we are going to use for building the classifier is called Transfer Learning using Inception. Well inception is a pre trained convolutional neural network model on 10,000,000 into thousand categories. Our use case here to train the inception on the images of DARTH Vader and Yoda. But Inception was not performed on these categories so we are going to use a concept called Transfer Learning



In the first layers, basic edge detection and shape detection of the image are performed so the weights doesn't change much in these layers for different images most of the difference occurs in the weights of last layers. So the concept of Transfer learning is that if we train just the last layers with our new set we can get pretty good accuracy for classifying images. Put it simply we are transferring the learning we

had from the all the previously trained 1,00,000 images into our new set of images.

Ok then lets get started with the steps.

Step 1: First download the sample images required for training set. I am searching google for images of darth vader there is this awesome extension which can download all the google search image results [here](#). Similarly for yoda images also. Step 1 is completed.

For Step 2, you need to have tensorflow setup on your local machine. If not you can follow it from [here](#). After tensor flow is set just clone the tensorflow repository to you local machine from official github repository. Now in order to train the image classifier we need to organise our image data so it easy for the training program to locate them.

First create a folder named “tf_files” inside the root directory and then create a another folder inside it regarding the image sets in this case i am naming it “star_wars” inside that we have one folder containing all the images of darth vader named “darth_vader” and another folder named Yoda containing all the images of yoda. Finally the file structure looks as follows

→*tf_file* →*star_wars* → *darth_vader, Yoda*

Now we have organised the image data, go to tensorflow directory in terminal and run the following command to initiate the training it

should take around 30minutes depending upon the number of images you have chosen

```
python tensorflow/examples/image_retraining/retrain.py \  
--bottleneck_dir=/tf_files/bottlenecks \  
--how_many_training_steps 500 \  
--model_dir=/tf_files/inception \  
--output_graph=/tf_files/retrained_graph.pb \  
--output_labels=/tf_files/retrained_labels.txt \  
--image_dir /tf_files/star_wars
```

bottlenecks folder used to cache the weights, output_graph denotes the retrained graph which we will be using on raspberry pi and the output_labels contains the labels of classification.

Note: If you face any issue like

TypeError: run() got an unexpected keyword argument 'argv'

then clone this repository

```
git clone -b r0.11 https://github.com/tensorflow/tensorflow.git
```

Moving on to Step 3, first install tensorflow on raspberry pi by following the steps in this [link](#). Now copy the retrained_graph.pb, retrained_labels.txt to your raspberry pi into a folder. Name the folder as tf_files. So the trained model is copied to raspberry pi.

For the final step, log into raspberry pi and create a simple python script named label_image.py as follows :

```
import tensorflow as tf
import sys

# change this as you see fit
image_path = sys.argv[1]

# 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("/home/pi/tf_files/retrained_labels.txt")]

# Unpersists graph from file
with
tf.gfile.FastGFile("/home/pi/tf_files/retrained_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, \
        {'DecodeJpeg/contents:0': image_data})

    # Sort to show labels of first prediction in order of
    confidence
```

```
top_k = predictions[0].argsort()[::-len(predictions[0]):]
[:::-1]

for node_id in top_k:
    human_string = label_lines[node_id]
    score = predictions[0][node_id]
    print('%s (score = %.5f)' % (human_string, score))
```

Now open up terminal and try the following command:

```
python label_image.py locationOfImage
```

Here locationOfImage example /home/pi/300093-darth-vader-lord-of-the-sith-002.jpg

The result should show as follows :

```
darth vader (score = 0.98963)
yoda (score = 0.01037)
```

Wrapping it up, you trained the image classifier for particular images on your local machine which gave a new cnn model. Now you have loaded this model into raspberry pi and classified a new image.

. . .

Footnotes : [Build a TensorFlow Image Classifier](#), [CodeLab by Google](#)

