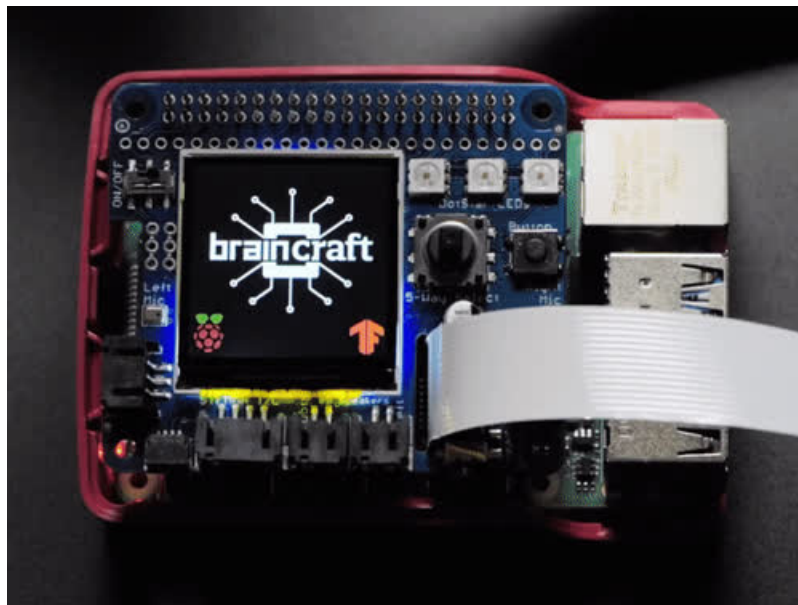


Running TensorFlow Lite Object Recognition on the Raspberry Pi 4

Created by Melissa LeBlanc-Williams



Last updated on 2021-06-23 09:43:57 PM EDT

Guide Contents

Guide Contents	2
Overview	3
Raspberry Pi 4 Computer & Camera	3
All-in One BrainCraft HAT	4
Display Output	4
Cooling It Down	5
Camera Extension Cable	6
Initial Setup	7
Enable Networking	7
Enable Interfaces	7
Update the Raspberry Pi	7
Upgrade Script	8
Display Setup	9
Option 1. BrainCraft HAT Setup	9
Option 2. PiTFT Setup	9
Camera Test	12
TensorFlow Lite 2 Setup	14
Install requirements	14
Virtual Environment	14
Install rpi-vision	14
Install TensorFlow 2.x	15
Running the Graphic Labeling Demo	16
Speech Output	18

Overview



Want to up your robotics game and give it the ability to detect objects? Maybe implement a security camera that can see and identify certain items? Now that the Raspberry Pi is fast enough to do machine learning, adding these features is fairly straightforward.

This guide will show you the steps to get TensorFlow 2 and TensorFlow Lite running on your Raspberry Pi 4 along with an object detection demo.

There are two main setup paths to choose from. The first option is with a PiTFT if you want to have a larger display. The second option is with the BrainCraft HAT, which has a built-in display and audio along several other components such as DotStar LEDs, a Joystick, and ports.

Raspberry Pi 4 Computer & Camera

To start with, you will need a Raspberry Pi 4. Since TensorFlow object detection is processing intensive, we recommend at least the 4GB model.

You really need a Pi 4 or better, TensorFlow vision recognition will not run on anything slower!

[Raspberry Pi 4 Model B - 4 GB RAM](#)

The Raspberry Pi 4 Model B is the newest Raspberry Pi computer made, and the Pi Foundation knows you can always make a good thing better! And what could make the Pi 4 better...

\$55.00

In Stock

Add to Cart

You will need a camera for the Raspberry Pi to see with.

[Raspberry Pi Camera Board v2 - 8 Megapixels](#)

Snap, snap! The Camera v2 is the new official camera board released by the Raspberry Pi Foundation!The Raspberry Pi Camera Board v2 is a high quality 8...

\$29.95

In Stock

Add to Cart

All-in One BrainCraft HAT

If you want to get a HAT that has everything you need besides the camera including display, sound, and cooling, you'll want to pick up the BrainCraft HAT.

Your browser does not support the video tag.

[Adafruit BrainCraft HAT - Machine Learning for Raspberry Pi 4](#)

The idea behind the BrainCraft HAT is that you'd be able to "craft brains" for Machine Learning on the EDGE, with Microcontrollers & Microcomputers. On ASK...

\$44.95

In Stock

Add to Cart

Display Output

You will also need a display so you can see what it's detecting. You can use any of our displays with the Raspberry Pi, but the 3.5" display is our biggest.

[PiTFT - Assembled 480x320 3.5" TFT+Touchscreen for Raspberry Pi](#)

Is this not the cutest, little display for the Raspberry Pi? It features a 3.5" display with 480x320 16-bit color pixels and a resistive touch overlay so is...

\$44.95

In Stock

Add to Cart

But our other PiTFT's will also work just fine

[PiTFT Plus 320x240 3.2" TFT + Resistive Touchscreen](#)

Is this not the cutest little display for the Raspberry Pi? It features a 3.2" display with 320x240 16-bit color pixels and a resistive touch overlay. The plate uses the high...

\$39.95

In Stock

Add to Cart

[PiTFT Plus Assembled 320x240 2.8" TFT + Resistive Touchscreen](#)

Is this not the cutest little display for the Raspberry Pi? It features a 2.8" display with 320x240 16-bit color pixels and a resistive touch overlay. The plate uses the high...

\$34.95

In Stock

Add to Cart

[Adafruit PiTFT 2.4" HAT Mini Kit - 320x240 TFT Touchscreen](#)

Is this not the cutest little display for the Raspberry Pi? It features a 2.4" display with 320x240 16-bit color pixels and a resistive touch overlay. The HAT uses the high speed...

\$34.95

In Stock

Add to Cart

[Adafruit PiTFT 2.2" HAT Mini Kit - 320x240 2.2" TFT - No Touch](#)

The cute PiTFT got even more adorable with this little primary display for Raspberry Pi in HAT form! It features a 2.2" display with 320x240 16-bit color pixels. The HAT uses the...

\$24.95

In Stock

Add to Cart

Cooling It Down

Since the Raspberry Pi 4 tends to run a little hot, we recommend the Pimoroni Fan SHIM.

[Pimoroni Fan SHIM for Raspberry Pi](#)

Looking for another way to keep your Raspberry Pi cool? Hook up Pimoroni's miniature...

\$11.95

In Stock

Add to Cart

Or this mini 5V fan

[Miniature 5V Cooling Fan for Raspberry Pi \(and Other Computers\)](#)

Looking for another way to keep your Raspberry Pi cool? Hook up this 5V Mini Cooling...

\$3.50

In Stock

Add to Cart

Or tall heatsink

Aluminum Heat Sink for Raspberry Pi 3 or 4 - 15 x 15 x 15mm

Looking for the best heat sink for your Raspberry Pi? Check out this nice and tall 15x15x15mm Heat...

\$1.95

In Stock

Add to Cart

In order to fit the fan/heatsink along with the display, you will need a GPIO stacking header.

GPIO Stacking Header for Pi A+/B+/Pi 2/Pi 3

Connect your own PCB to a Raspberry Pi B+ and stack on top with this normal-height female header with extra long pins. The female header part is about 8.5mm tall, good for small...

\$2.50

In Stock

Add to Cart

Camera Extension Cable

The flex cable that comes with the camera is a bit on the short side, so you may want a longer cable as well.

Flex Cable for Raspberry Pi Camera or Display - 24" / 610mm

This cable will let you swap out the stock 150mm long flex cable from a Raspberry Pi Camera (either 'classic' or 'NoIR' type) or Raspberry Pi Display for a...

\$2.95

In Stock

Add to Cart

Flex Cable for Raspberry Pi Camera or Display - 18" / 457mm

This cable will let you swap out the stock 150mm long flex cable from a Raspberry Pi Camera (either 'classic' or 'NoIR' type) or Raspberry Pi Display for a different...

\$2.50

In Stock

Add to Cart

Initial Setup

We're going to get started using the latest Raspbian Lite, we don't need the graphical system which takes up precious CPU cycles. Start by downloading the latest Raspberry Pi OS (formerly Raspbian) Lite operating system image.

<https://adafru.it/LAW>

<https://adafru.it/LAW>

Burn the operating system image to a Micro SD Card. Directions are available on [the Raspberry Pi site \(https://adafru.it/jd0\)](https://adafru.it/jd0).

Mount the SD Card on your computer. Go to the boot partition. [You want to create an empty file in that partition named ssh. This will tell Raspbian to enable SSH. \(https://adafru.it/vbC\)](https://adafru.it/vbC)

Enable Networking

For networking, you can use either a cable or WiFi. You can find more information about this in our [Raspberry Pi Network Setup \(https://adafru.it/dDN\)](https://adafru.it/dDN) guide.

Enable Interfaces

After properly unmounting ("ejecting") your card for you computer, go ahead and place the microSD card in the Pi and supply power to boot it up. It may take a few minutes before it's available.

On your computer, SSH into the card using the following:

```
ssh pi@raspberrypi.local
```

This is a great time to change your password.

```
passwd
```

Update the Raspberry Pi

Update the Pi

```
sudo apt-get update  
sudo apt-get upgrade
```

and

```
sudo apt-get install -y python3-pip
```

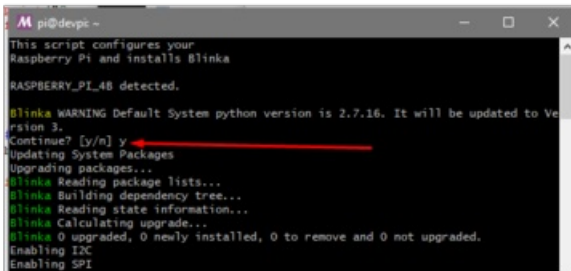
and

```
pip3 install --upgrade setuptools
```

Upgrade Script

We put together a script to easily make sure your Pi is correctly configured and install Blinka. Although Blinka isn't required for this to work, it's nice to have and this script also enables several other interfaces that are required for this project to work. It requires just a few commands to run. Most of it is installing the dependencies.

```
cd ~
sudo pip3 install --upgrade adafruit-python-shell
wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/raspi-
blinka.py
sudo python3 raspi-blinka.py
```

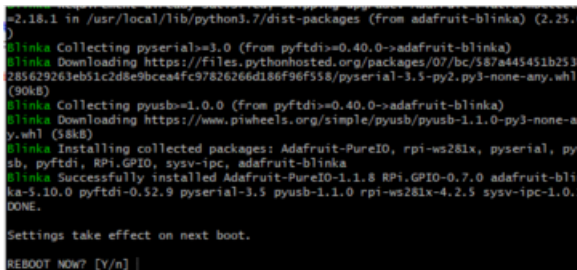


```
pi@devpi:~$
This script configures your
Raspberry Pi and installs Blinka

RASPBERRY_PI_4B detected.

Blinka WARNING Default System python version is 2.7.16. It will be updated to Ver
sion 3.
Continue? [y/n] y
Updating System Packages
Upgrading packages...
Blinka Reading package lists...
Blinka Building dependency tree...
Blinka Reading state information...
Blinka Calculating upgrade...
Blinka 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Enabling I2C
Enabling SPI
```

If your system default Python is Python 2 (which is likely on a first install), it will ask to confirm that you want to proceed. Choose **yes**.



```
=2.18.1 in /usr/local/lib/python3.7/dist-packages (from adafruit-blinka) (2.25.
)
Blinka Collecting pyserial>=3.0 (from pyftdi>=0.40.0->adafruit-blinka)
Blinka Downloading https://files.pythonhosted.org/packages/07/bc/587a445451b253
285629263eb51c2d8e9bcea4fc97826266d186f96f558/pyserial-3.5-py2.py3-none-any.whl
(90kB)
Blinka Collecting pyusb>=1.0.0 (from pyftdi>=0.40.0->adafruit-blinka)
Blinka Downloading https://www.piwheels.org/simple/pyusb/pyusb-1.1.0-py3-none-a
ny.whl (58kB)
Blinka Installing collected packages: Adafruit-PureIO, rpi-ws281x, pyserial, py
usb, pyftdi, RPi.GPIO, sysv-ipc, adafruit-blinka
Blinka Successfully installed Adafruit-PureIO-1.1.8 RPi.GPIO-0.7.0 adafruit-bli
nka-5.10.0 pyftdi-0.52.9 pyserial-3.5 pyusb-1.1.0 rpi-ws281x-4.2.5 sysv-ipc-1.0.
DONE.

Settings take effect on next boot.

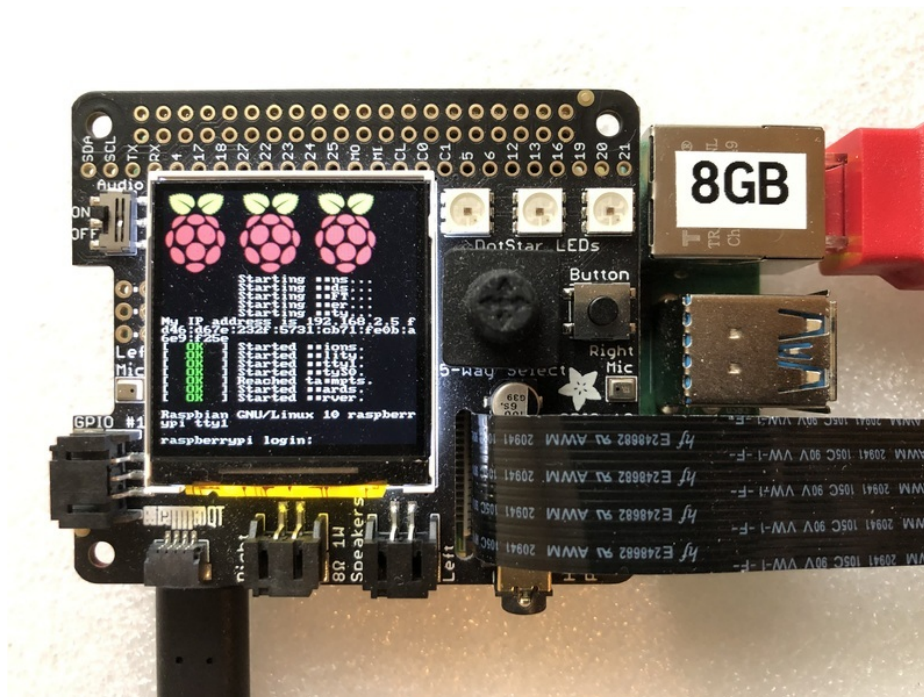
REBOOT NOW? [Y/n]
```

It may take a few minutes to run. When it finishes, it will ask you if you would like to reboot. Choose **yes**.

Display Setup

Option 1. BrainCraft HAT Setup

If you have the BrainCraft HAT, you'll first want to head over to the [Adafruit BrainCraft HAT - Easy Machine Learning for Raspberry Pi \(https://adafru.it/NLE\)](https://adafru.it/NLE) guide and go through the setup process there. Specifically, you'll want to follow the **Audio Setup**, **Fan Service Setup**, and **Display Module Install** pages. This will guide you through all the pieces needed to prepare the Pi.



Option 2. PiTFT Setup

If you have just a bare PiTFT, you'll want to install the PiTFT Drivers. There is a new installer script now, so it can be installed with just a few commands. First, start by installing a few dependencies and downloading the repo:

```
cd ~
sudo pip3 install --upgrade adafruit-python-shell click==7.0
sudo apt-get install -y git
git clone https://github.com/adafruit/Raspberry-Pi-Installer-Scripts.git
cd Raspberry-Pi-Installer-Scripts
```

Next, choose the install command based on your display. For the **2.4"**, **2.8"**, or **3.2" Resistive touchscreens**, use the following command:

```
sudo python3 adafruit-pitft.py --display=28r --rotation=90 --install-type=fbcpc
```

For the **3.5" Resistive touchscreen**, use the following command:

```
sudo python3 adafruit-pitft.py --display=35r --rotation=90 --install-type=fbcpc
```

For the **2.8" Capacitive touchscreen**, use the following command:

```
sudo python3 adafruit-pitft.py --display=28c --rotation=90 --install-type=fbcpc
```

When you get asked to **reboot**, reboot!

```
removing old section...
##### UPGRADING KERNEL #####
Updating packages...
Updating packages...
Installing Kernel Headers...
PITFT make: Entering directory '/usr/src/linux-headers-5.4.51-v7l+'
PITFT Building modules, stage 2.
PITFT MODPOST 2 modules
PITFT LD [M] /home/pi/Raspberry-Pi-Installer-Scripts/st7789_module/fb_st7789v.ko
PITFT LD [M] /home/pi/Raspberry-Pi-Installer-Scripts/st7789_module/st7789v_ada.ko
PITFT make: Leaving directory '/usr/src/linux-headers-5.4.51-v7l+'
PITFT Success!

Settings take effect on next boot.

REBOOT NOW? [Y/n] y
```

After it reboots, you should now see text on the display.

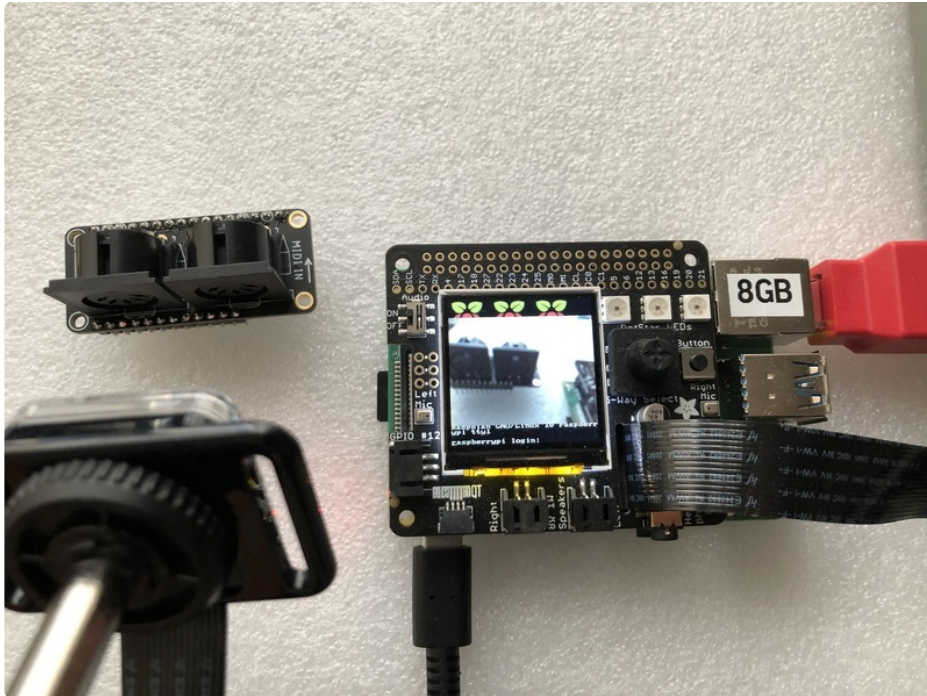


Camera Test

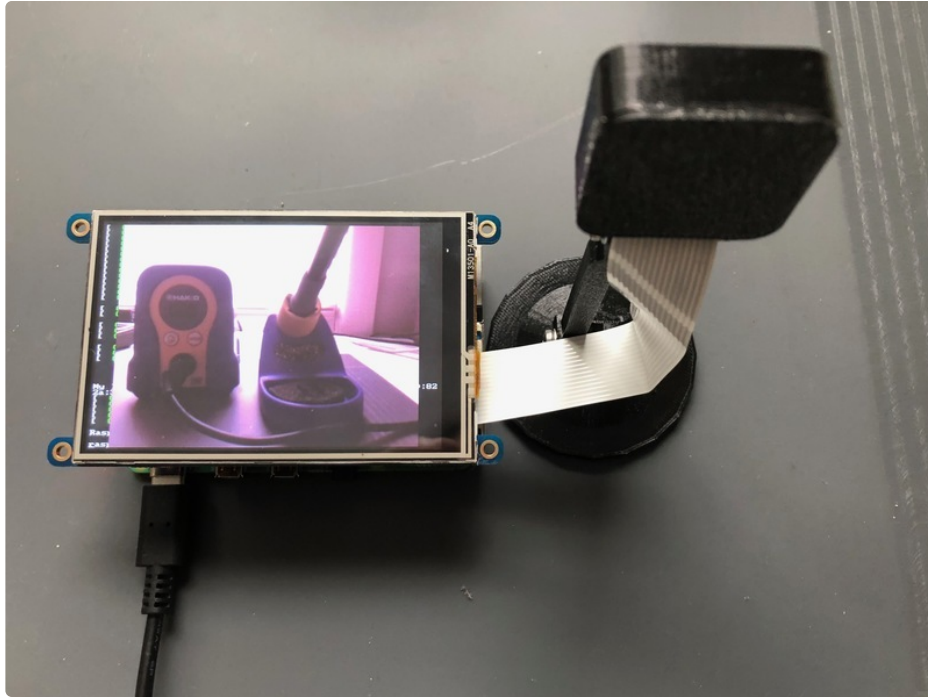
Now to do an initial test with the camera to make sure the hardware is working before we dive into TensorFlow. This should display what the camera sees on the display.

```
raspistill -t 0
```

Here's what it looks like on the BrainCraft HAT:



Here's what it looks like on the PiTFT:



If you don't see anything on your camera or get an error message, be sure you have the camera interface enabled in **raspi-config**. If it is still having issues, you will need to get that working before continuing. Unfortunately troubleshooting a camera setup can be complex and is beyond the scope of this guide.

TensorFlow Lite 2 Setup

There's a LOT of software to install, this can take up to an hour

Install requirements

There are a few packages that TensorFlow requires that need to be installed:

```
sudo apt-get install -y libatlas-base-dev libhdf5-dev libc-ares-dev libeigen3-dev build-essential libsdl-ttf2.0-0 python-pygame festival python3-h5py
```

Virtual Environment

There are a few dependency requirements to install TensorFlow inside the Python Environment:

```
pip3 install virtualenv Pillow numpy pygame
```

Install rpi-vision

Now to install an Adafruit fork of a program originally written by Leigh Johnson that uses the MobileNet V2 model to detect objects. This part will take a few minutes to complete.

```
cd ~
git clone --depth 1 https://github.com/adafruit/rpi-vision.git
cd rpi-vision
python3 -m virtualenv -p $(which python3) .venv
source .venv/bin/activate
```

```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ git clone --depth 1 https://github.com/adafruit/rpi-vision.git
Cloning into 'rpi-vision'...

remote: Enumerating objects: 99, done.
remote: Counting objects: 100% (99/99), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 99 (delta 19), reused 64 (delta 18), pack-reused 0
Unpacking objects: 100% (99/99), done.
Checking out files: 100% (80/80), done.
pi@raspberrypi:~ $ cd rpi-vision
pi@raspberrypi:~/rpi-vision $ python3 -m virtualenv -p $(which python3) .venv
Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/pi/rpi-vision/.venv/bin/python3
Also creating executable in /home/pi/rpi-vision/.venv/bin/python
Installing setuptools, pip, wheel...
done.
pi@raspberrypi:~/rpi-vision $ source .venv/bin/activate
(.venv) pi@raspberrypi:~/rpi-vision $
```

Install TensorFlow 2.x

You should now be inside a virtual environment. You can tell by the (.venv) on the left side of the command prompt. While in the virtual environment, you may download and install Tensorflow 2.3.1

wget https://raw.githubusercontent.com/PINTO0309/Tensorflow-bin/main/previous_versions/download_tensorflow-2.3.1-cp37-none-linux_armv7l.sh (<https://adafru.it/Taa>)

```
chmod a+x ./download_tensorflow-2.3.1-cp37-none-linux_armv7l.sh
```

```
./download_tensorflow-2.3.1-cp37-none-linux_armv7l.sh
```

```
pip3 install --upgrade setuptools
```

```
(.venv) pi@raspberrypi:~/rpi-vision $ wget https://raw.githubusercontent.com/PINTO0309/Tensorflow-bin/master/tensorflow-2.3.1-cp37-none-linux_armv7l_download.sh
--2020-10-08 16:24:10-- https://raw.githubusercontent.com/PINTO0309/Tensorflow-bin/master/tensorflow-2.3.1-cp37-none-linux_armv7l_download.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.68.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.68.133|:443.
.. connected.
HTTP request sent, awaiting response... 200 OK
Length: 372 [text/plain]
Saving to: 'tensorflow-2.3.1-cp37-none-linux_armv7l_download.sh'

tensorflow-2.3.1-cp 100%[=====>]      372  --.-KB/s   in 0s

2020-10-08 16:24:10 (6.41 MB/s) - 'tensorflow-2.3.1-cp37-none-linux_armv7l_download.sh'
saved [372/372]

(.venv) pi@raspberrypi:~/rpi-vision $ chmod a+x ./tensorflow-2.3.1-cp37-none-linux_armv7l_download.sh
(.venv) pi@raspberrypi:~/rpi-vision $ ./tensorflow-2.3.1-cp37-none-linux_armv7l_download.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  408      0  408    0     0    2071      0 --:--:-- --:--:-- --:--:--   2081
  0     0      0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0      0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 180M      0 180M    0     0  5866k      0 --:--:--  0:00:31 --:--:--  6335k
Download finished.
```

```
pip3 install ./tensorflow-2.3.1-cp37-none-linux_armv7l.whl
```

```
pip3 install -e .
```

```

Collecting gast>=0.2.0 (from tensorflow==2.0.0rc0)
  Using cached https://www.piwheels.org/simple/gast/gast-0.2.2-py3-none-any.whl
Collecting astor>=0.6.0 (from tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/d1/4f/950dfae467b384fc96bc6469de25d832534f6b44410
none-any.whl
Collecting protobuf>=3.6.1 (from tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/0d/2e/d4b1b67c264ce6722def110f2715461e9b4d4964795
y3-none-any.whl
Collecting tf-estimator-nightly<1.14.0.dev2019080602,>=1.14.0.dev2019080601 (from tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/21/28/f2a27a62943d5f041e4a6fd404b2d21cb7c59b2242a
-1.14.0.dev2019080601-py2.py3-none-any.whl
Collecting keras-applications>=1.0.8 (from tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/71/e3/19762fdfc62877ae9102edf6342d71b28bfd9dea3d
.0.8-py3-none-any.whl
Collecting absl-py>=0.7.0 (from tensorflow==2.0.0rc0)
  Using cached https://www.piwheels.org/simple/absl-py/absl_py-0.8.0-py3-none-any.whl
Requirement already satisfied: wheel>=0.26 in ./venv/lib/python3.7/site-packages (from tensorflow==2.0.
Collecting six>=1.10.0 (from tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe898238ce23f502a721c0ac0e
one-any.whl
Collecting wrapt>=1.11.1 (from tensorflow==2.0.0rc0)
  Using cached https://www.piwheels.org/simple/wrapt/wrapt-1.11.2-cp37m-linux_armv7l.whl
Collecting google-pasta>=0.1.6 (from tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/d0/33/376510eb8d6246f3c30545f416b2263eee461e40940
y3-none-any.whl
Collecting opt-einsum>=2.3.2 (from tensorflow==2.0.0rc0)
  Using cached https://www.piwheels.org/simple/opt-einsum/opt_einsum-3.0.1-py3-none-any.whl
Requirement already satisfied: setuptools>=41.0.0 in ./venv/lib/python3.7/site-packages (from tb-nightly
tensorflow==2.0.0rc0) (41.2.0)
Collecting werkzeug>=0.11.15 (from tb-nightly<1.15.0a20190807,>=1.15.0a20190806->tensorflow==2.0.0rc0)
  Using cached https://files.pythonhosted.org/packages/d1/ab/d3bed6b92042622d24decc7aad8877badf18aeca15

```

After this, go ahead and reboot the Pi.

```
sudo reboot
```

Running the Graphic Labeling Demo

Finally you are ready to run the detection software. First you want to run as **root** so that Python can access the Frame Buffer of the display.

```
sudo bash
```

Then activate the virtual environment again:

```
cd rpi-vision && . .venv/bin/activate
```

To run a program that will display the object it sees on screen type in the following:

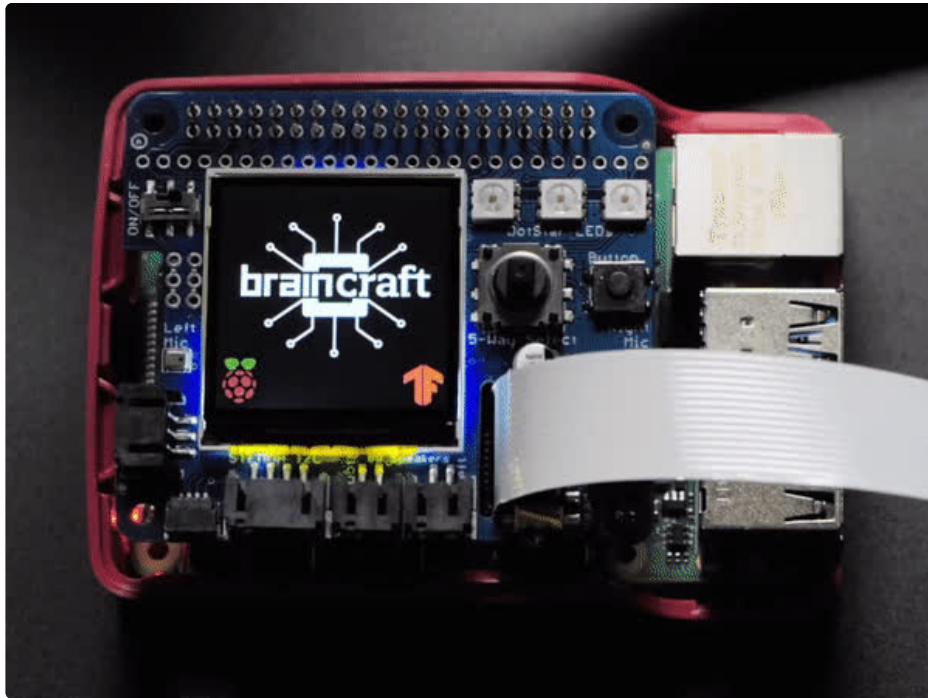
```
python3 tests/pitft_labeled_output.py --tflite
```

You should see a bunch of text scrolling in your SSH window.


```
INFO:root:[('n03063599', 'coffee_mug', 0.54296756), ('n07930864', 'cup', 0.04130427), ('n03584254', 'glass', 0.013251671), ('n03916031', 'perfume', 0.012419094)]
INFO:root:TFLite inference took 165 ms, 6.0 FPS
['coffee_mug', 'coffee_mug', 'coffee_mug', None, 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug']
Detected coffee_mug
INFO:root:[('n03063599', 'coffee_mug', 0.555453), ('n07930864', 'cup', 0.037044503), ('n03584254', 'glass', 0.012711463), ('n07892512', 'red_wine', 0.011942282)]
INFO:root:TFLite inference took 170 ms, 5.9 FPS
['coffee_mug', 'coffee_mug', None, 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug']
Detected coffee_mug
INFO:root:[('n03063599', 'coffee_mug', 0.52811885), ('n07930864', 'cup', 0.04920438), ('n03584254', 'glass', 0.013310642), ('n07920052', 'espresso', 0.012384634)]
INFO:root:TFLite inference took 167 ms, 6.0 FPS
['coffee_mug', None, 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug']
Detected coffee_mug
INFO:root:[('n03063599', 'coffee_mug', 0.53334224), ('n03584254', 'iPod', 0.038200967), ('n07930864', 'tissue', 0.013704391), ('n07892512', 'red_wine', 0.0112767955)]
INFO:root:TFLite inference took 163 ms, 6.1 FPS
[None, 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug']
Detected coffee_mug
INFO:root:[('n03063599', 'coffee_mug', 0.49518147), ('n07930864', 'cup', 0.039990723), ('n03584254', 'wine', 0.015094341), ('n02823750', 'beer_glass', 0.01345881)]
INFO:root:TFLite inference took 161 ms, 6.2 FPS
['coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug']
INFO:root:[('n03063599', 'coffee_mug', 0.53454936), ('n07930864', 'cup', 0.038659144), ('n03584254', 'glass', 0.014627116), ('n03642806', 'laptop', 0.012978077)]
INFO:root:TFLite inference took 165 ms, 6.0 FPS
['coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug', 'coffee_mug']
Detected coffee_mug
```

Now start holding up various items in front of the camera and it should display what it thinks it sees, which isn't actually what the item may be. Some items that it's pretty good about identifying are coffee mugs and animals.





Speech Output

As an added bonus, you can hook up a pair of headphones or a speaker to the Raspberry Pi and it will actually tell you what it is detecting. Make sure you don't have any HDMI cords plugged in though or the audio will go through the monitor.

