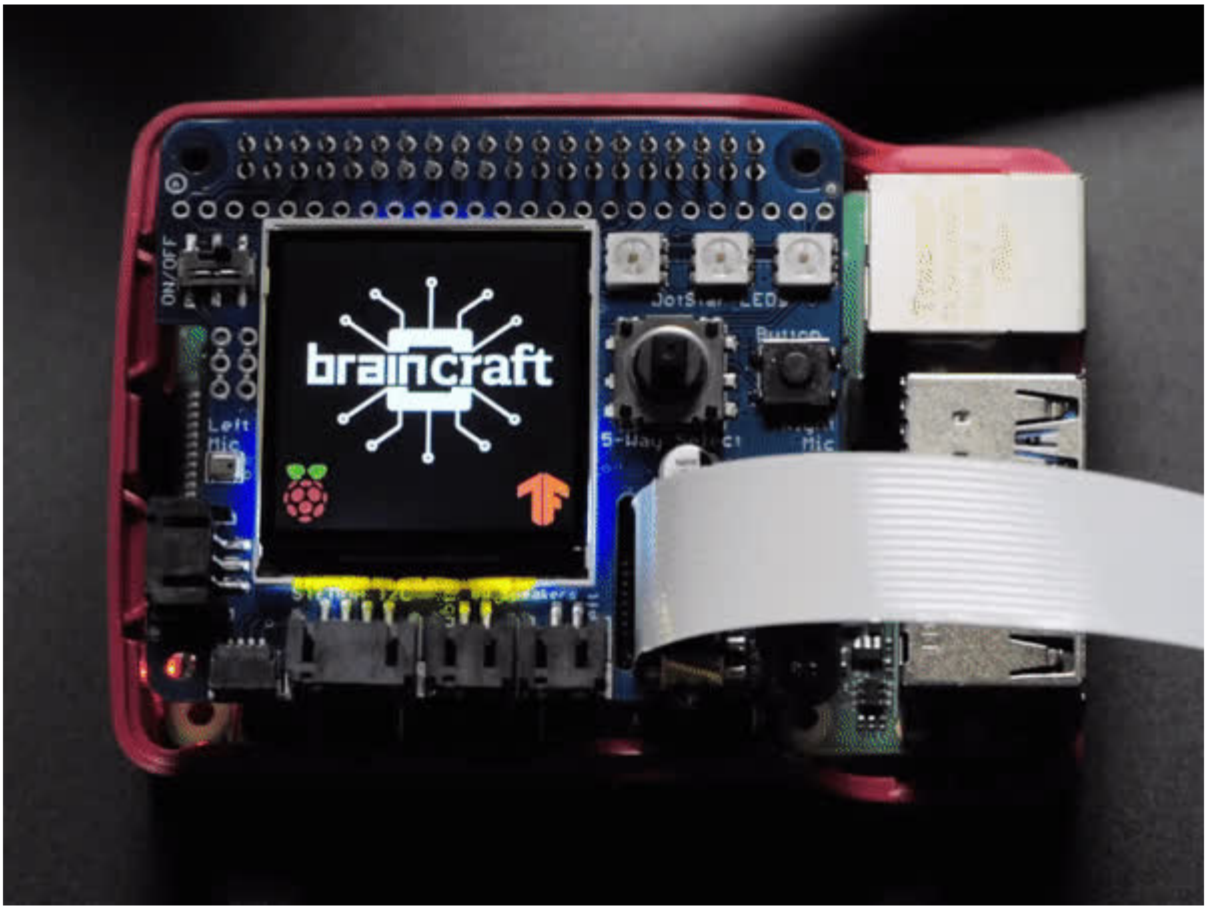
Running TensorFlow Lite Object Recognition on the Raspberry Pi 4 or Pi 5

Created by Melissa LeBlanc-Williams



https://learn.adafruit.com/running-tensorflow-lite-on-the-raspberry-pi-4

Last updated on 2025-08-11 11:44:28 AM EDT

Table of Contents

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.

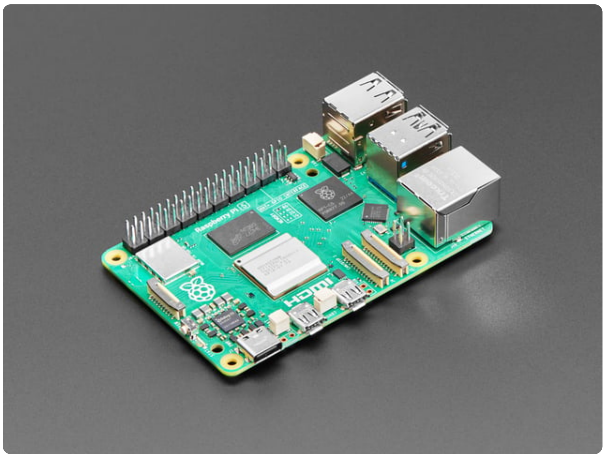
This guide will show you the steps to get TensorFlow 2 installed on your Raspberry Pi 4 or 5 and perform some object detection using the TensorFlow Lite Python Interpreter, which is faster than the full TensorFlow interpreter.

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 or 5 Computer & Camera

To start with, you will need a Raspberry Pi 4 or 5. Since TensorFlow object detection is processing intensive, you should use at least the 4GB model.

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



Raspberry Pi 5 - 4 GB RAM

The Raspberry Pi 5 is the newest Raspberry Pi computer, and the Pi Foundation knows you can always make a good thing better! And what could make the Pi 5 better than the...

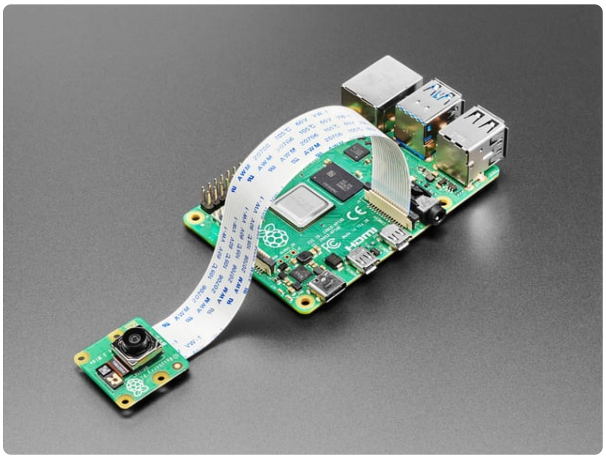
https://www.adafruit.com/product/5812

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...

https://www.adafruit.com/product/4296

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



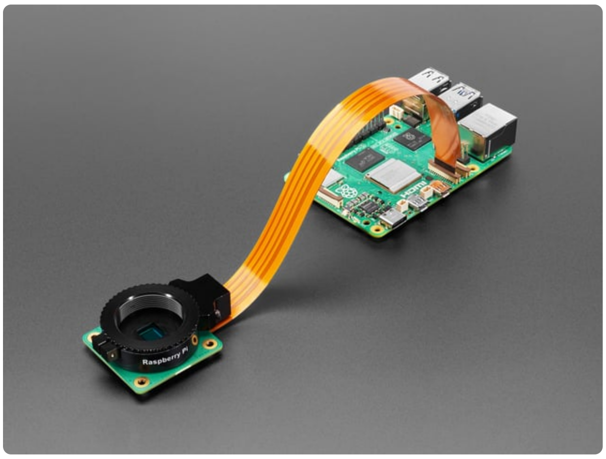
Raspberry Pi Camera Module 3 Standard

Raspberry Pi Camera Module 3 is a compact camera from Raspberry Pi. Featuring autofocus and a 12-megapixel sensor, and supported by Raspberry Pi's Picamera2 Python library, Camera...

https://www.adafruit.com/product/5657

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...
https://www.adafruit.com/product/3099

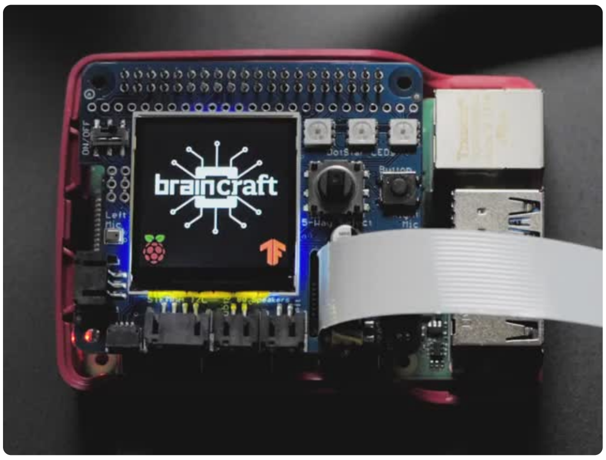
For the Raspberry Pi 5, you will also need a camera cable, which has a different size than the one that comes with the camera.



Raspberry Pi 5 FPC Camera Cable - 22-pin 0.5mm to 15-pin 1mm
This 200mm long camera cable is specifically designed to work with the Raspberry Pi 5 series. Just...
https://www.adafruit.com/product/5818

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.



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...
https://www.adafruit.com/product/4374

Display Output

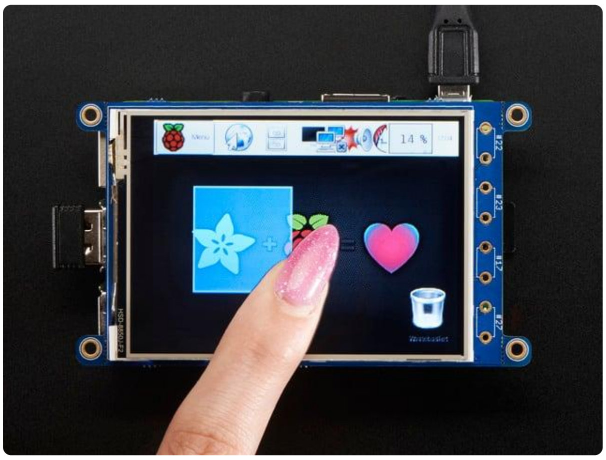
You will also need a display so you can see what it's detecting. You can use any of our displays with the Raspberry Pi, but the 3.5" display is Adafruit's biggest.

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...

https://www.adafruit.com/product/2097

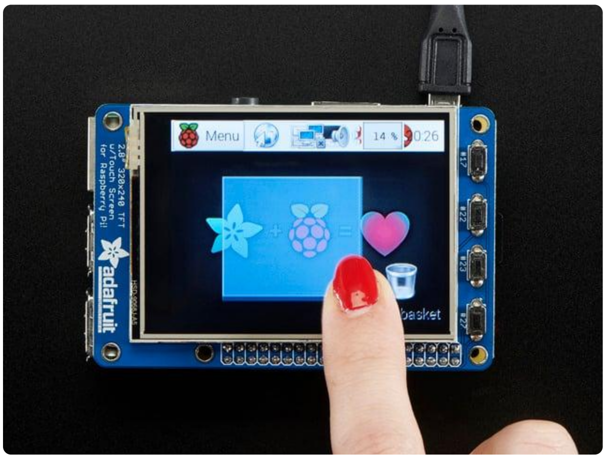
But our other PiTFT's will also work just fine



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...

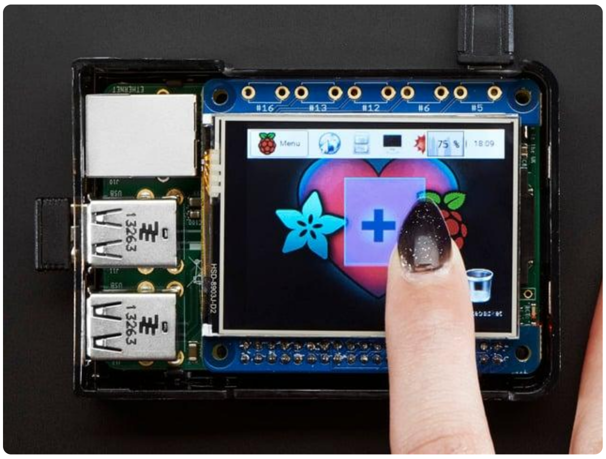
https://www.adafruit.com/product/2616



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...

https://www.adafruit.com/product/2298

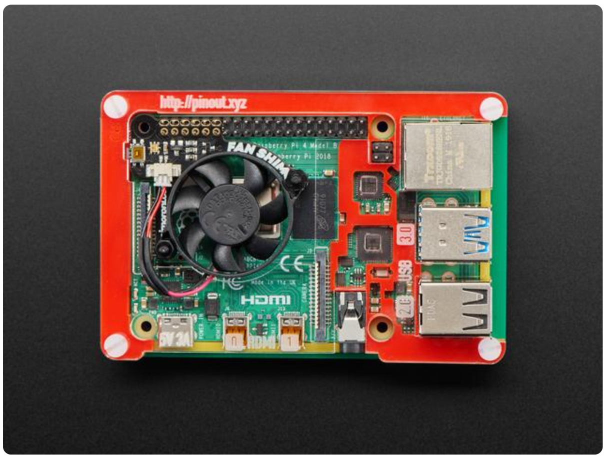


Adafruit PiTFT 2.4" HAT Mini Kit - 320x240 TFT Touchscreen
Is this not the cutest little display for the Raspberry Pi? It features a 2.4" display with 320x240 16-bit color pixels and a resistive touch overlay. The HAT uses the high speed...
https://www.adafruit.com/product/2455

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...
https://www.adafruit.com/product/2315

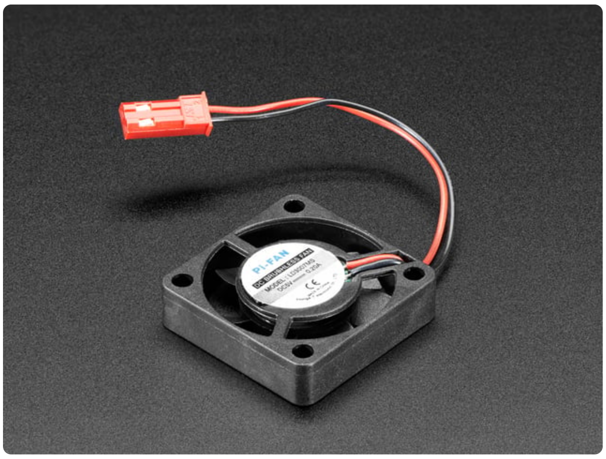
Cooling It Down

The Raspberry Pi 4 can run a little hot, especially when TensorFlow is doing a lot of data crunching. If you don't have the BrainCraft hat with the built-in fan, we recommend the Pimoroni Fan SHIM.



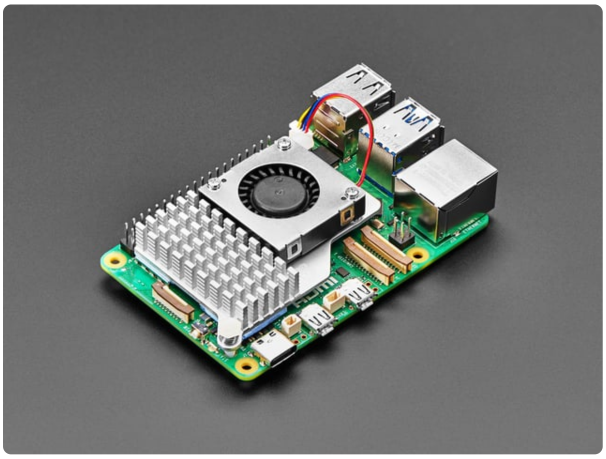
Pimoroni Fan SHIM for Raspberry Pi
Looking for another way to keep your Raspberry Pi cool? Hook up Pimoroni's miniature...
https://www.adafruit.com/product/4316

Or this mini 5V fan



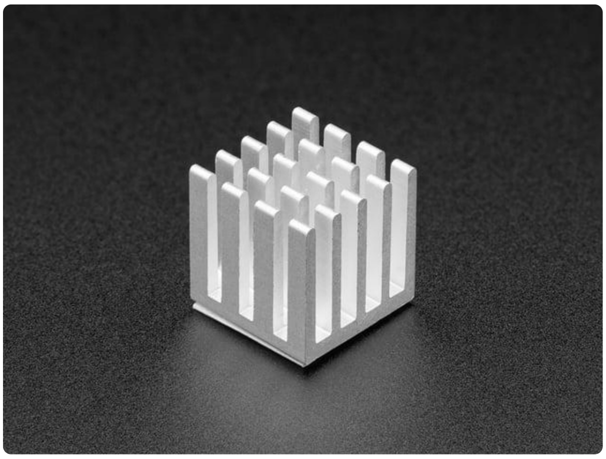
Miniature 5V Cooling Fan for Raspberry Pi (and Other Computers)
Looking for another way to keep your Raspberry Pi cool? Hook up this 5V Mini Cooling...
https://www.adafruit.com/product/3368

Or if you have the Pi 5, you can use an official cooler.



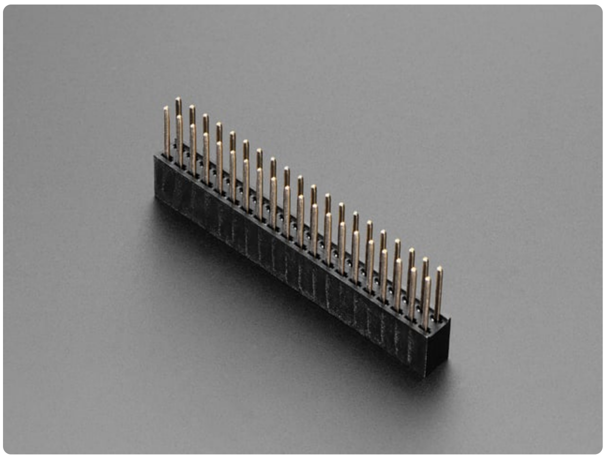
Official Raspberry Pi 5 Active Cooler
The Raspberry Pi 5 Active Cooler is compatible with the Raspberry Pi 5 and the
https://www.adafruit.com/product/5815

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...
https://www.adafruit.com/product/3082

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



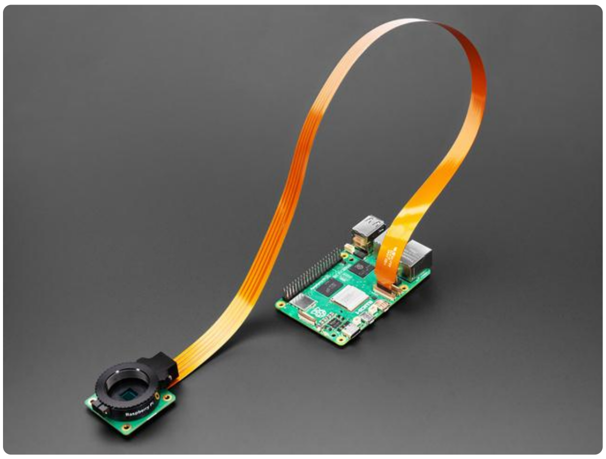
GPIO Stacking Header for Pi A+/B+/Pi 2/Pi 3

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...

https://www.adafruit.com/product/2223

Longer Camera Cable

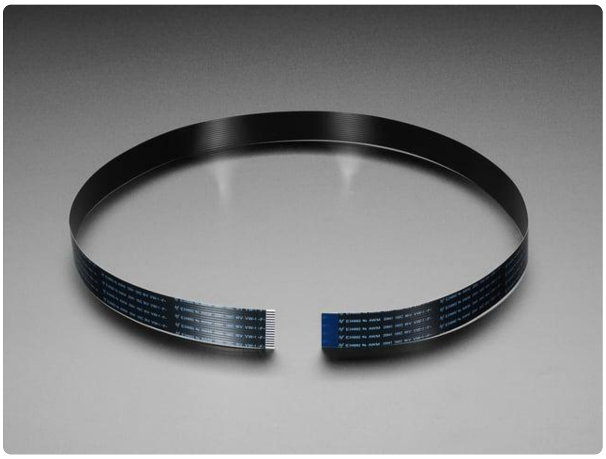
The flex cable that comes with the camera is a bit on the short side, so you may want a longer cable as well.



Raspberry Pi 5 FPC Camera Cable - 22-pin 0.5mm to 15-pin 1mm

This 500mm long camera cable is specifically designed to work with the Raspberry Pi 5 series. Just plug...

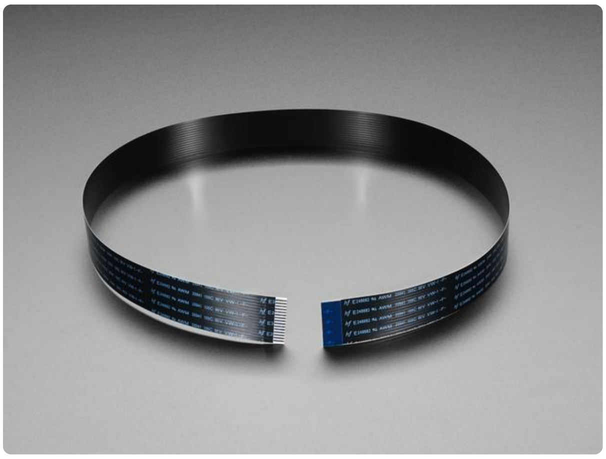
https://www.adafruit.com/product/5820



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...

https://www.adafruit.com/product/1731



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...

https://www.adafruit.com/product/1730

Initial Setup

This guide was originally written for Raspberry Pi OS Buster, with the original Picamera library. When Raspberry Pi released Bullseye, they did so with a promise that eventually there would be a new Picamera2 library that was built on top of the libcamera subsystem. While it was possible to install Picamera2, it was a very involved process that was not friendly to the average user and still highly experimental.

Nearly a year later, they started including Picamera2 with Raspberry Pi OS by default, so it was time for a guide update. The installation path that this updated guide uses is designed to make it as easy as possible for you to get a Raspberry Pi up and running with TensorFlow.

The overall installation path will be to using the 64-bit version of the Raspberry Pi Desktop because recent versions of TensorFlow are no longer compiling for the 32-bit OS and the QT OpenGL graphics drivers are installed by default on the Desktop version.

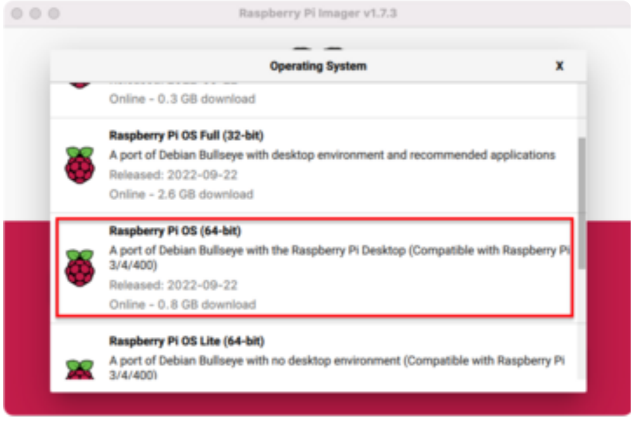
Image Installation

You're going to get started using the latest Raspbian Pi OS Desktop. Start by downloading and installing the latest Raspberry Pi Imager, which makes it super easy to customize all your settings before you even write the image to the SD Card.

Latest Raspberry Pi Imager Downloads Page

https://adafru.it/XMd

Use the imager to burn the operating system image to a Micro SD Card. Raspberry Pi has provided a quick video overview available to watch.



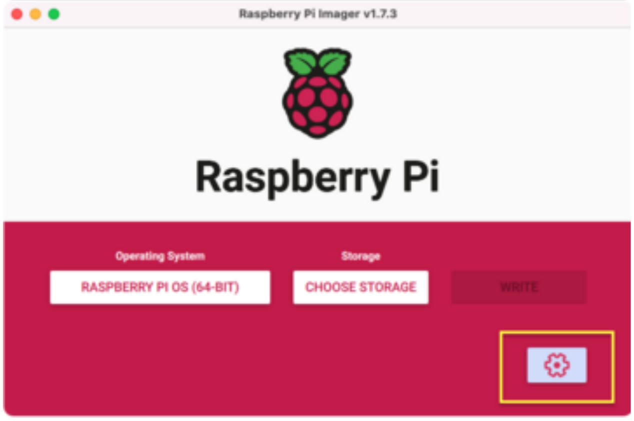
Online - 0.3 GB download

Raspberry Pi OS Full (32-bit)
A port of Debian Bullseye with desktop environment and recommended applications
Released: 2022-09-22
Online - 2.6 GB download

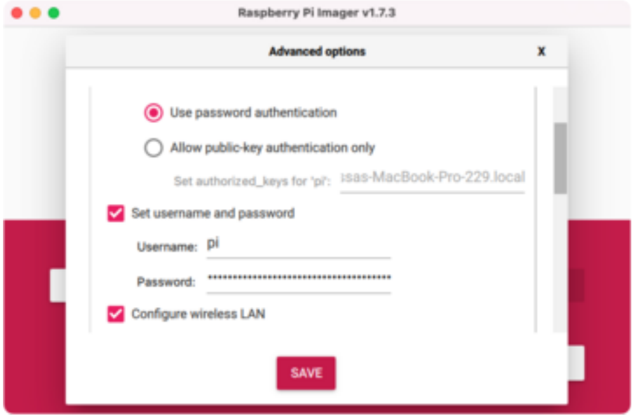
Raspberry Pi OS (64-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Compatible with Raspberry Pi 3/4/400)
Released: 2022-09-22
Online - 0.8 GB download

Raspberry Pi OS Lite (64-bit)
A port of Debian Bullseye with no desktop environment (Compatible with Raspberry Pi 3/4/400)

Select the 64-bit Raspberry Pi OS Bullseye Desktop image by going to **Raspberry Pi OS (Other) → Raspberry Pi OS (64-bit)**.



You can customize settings for your Pi by clicking the gear icon in the lower righthand corner after choosing your OS.



● Use password authentication
○ Allow public-key authentication only
Set authorized_keys for 'pi': ısas-MacBook-Pro-229.local
☑ Set username and password
Username: pi
Password: ••••••••••••••••••••••••••••••••••••
☑ Configure wireless LAN

Make sure to enable SSH, set a good username and password, set your WiFi credentials, and update the locale settings to your specific needs.

Log In

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, assuming you left the username as **pi** and the hostname as **raspberrypi.local**, SSH into the card using the following:

`ssh pi@raspberrypi.local`

If you changed your username or hostname settings in the imager, use those instead.

Update the Raspberry Pi

Update the Pi

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y python3-pip
sudo apt install --upgrade -y python3-setuptools
```

Setup Virtual Environment

If you are installing on the Bookworm version of Raspberry Pi OS, you will need to install your python modules in a virtual environment. You can find more information in the Python Virtual Environment Usage on Raspberry Pi (https://adafru.it/19a5) guide. To Install and activate the virtual environment, use the following commands:

```
sudo apt install python3.11-venv
python -m venv env --system-site-packages
```

You will need to activate the virtual environment every time the Pi is rebooted. To activate it:

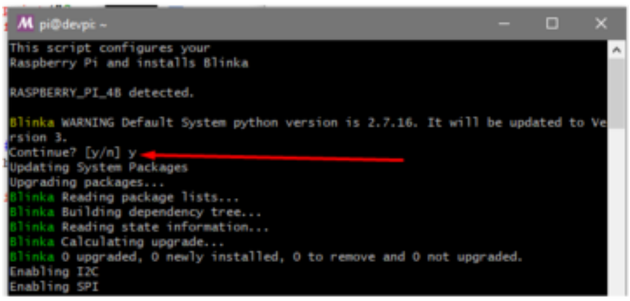
```
source env/bin/activate
```

If needed, you can use `deactivate`, but leave it active for now.

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

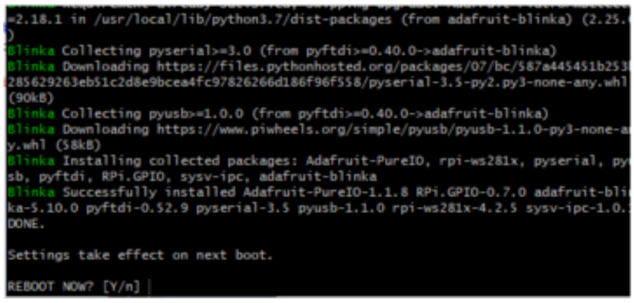


```
This script configures your
Raspberry Pi and installs Blinka

RASPBERRY_PI_4B detected.

Blinka WARNING Default System python version is 2.7.16. It will be updated to Ve
rsion 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 usually is no longer the case with the recent versions of Raspberry Pi OS), 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
y.whl (58kB)
Blinka Installing collected packages: Adafruit-PureIO, rpi-ws281x, pyserial, py
sb, pyftdi, RPi.GPIO, sysv-ipc, adafruit-blinka
Blinka Successfully installed Adafruit-PureIO-1.1.8 RPi.GPIO-0.7.0 adafruit-bli
ka-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**.

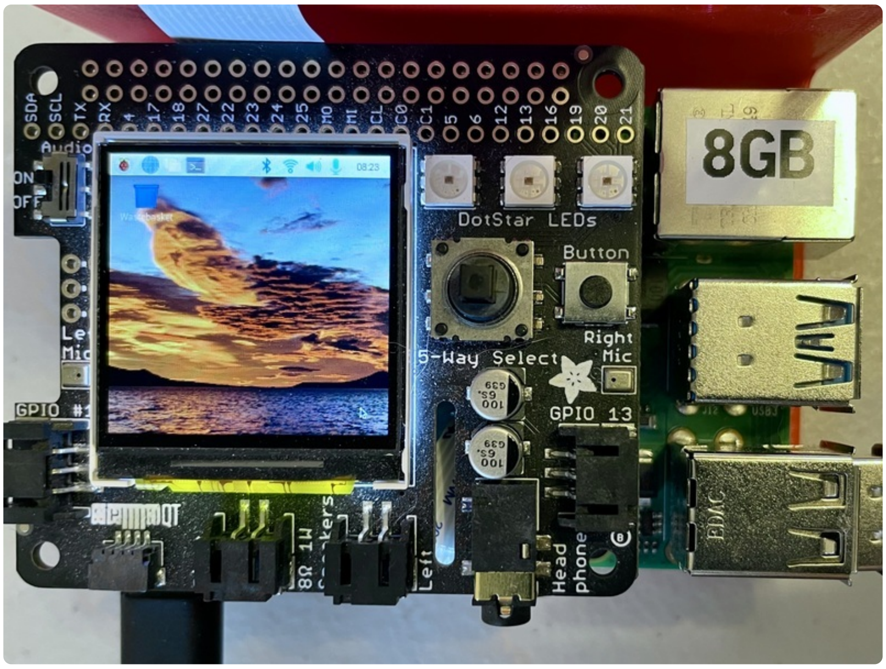
Display Setup

Driver Installation

The first step to setup the display is to install the necessary drivers. Depending on your exact hardware, you have a couple of options. After the driver setup, there are a couple of additional steps below.

Option 1. BrainCraft HAT Setup

If you have an Adafruit BrainCraft HAT, you'll first want to head over to the Adafruit BrainCraft HAT - Easy Machine Learning for Raspberry Pi (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
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 -E env PATH=$PATH python3 adafruit-pitft.py --display=28r --rotation=90 --
install-type=mirror
```

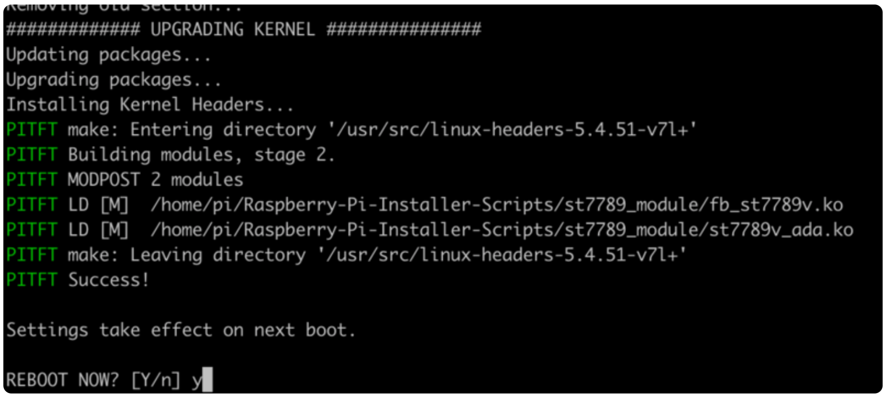
For the **3.5" Resistive touchscreen**, use the following command:

```
sudo -E env PATH=$PATH python3 adafruit-pitft.py --display=35r --rotation=90 --
install-type=mirror
```

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

```
sudo -E env PATH=$PATH python3 adafruit-pitft.py --display=28c --rotation=90 --
install-type=mirror
```

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

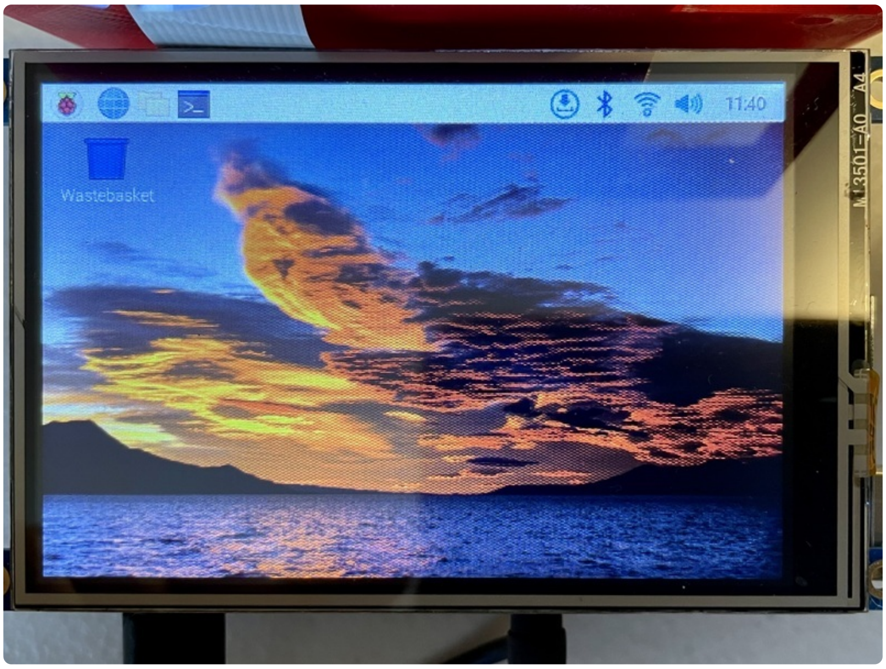


```
Removing old section...
############ UPGRADING KERNEL ##############
Updating packages...
Upgrading packages...
Installing Kernel Headers...
PITFT make: Entering directory '/usr/src/linux-headers-5.4.51-v7l+'
PITFT Building modules, stage 2.
PITFT MODPOST 2 modules
PITFT LD [M]  /home/pi/Raspberry-Pi-Installer-Scripts/st7789_module/fb_st7789v.ko
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 your desktop on the display.



Make the TFT the Default Display

With the Desktop Environment, you can set the default display by setting an environment variable. To set the TFT as the display where the window is drawn, there are a few ways to do this depending on the level of permanence that you would like to give it. Each way involves setting the **DISPLAY** environment variable to `:0`.

The first way is to prefix every command with `DISPLAY=:0`. This sets the enviroment variable until the command is done executing. So if you wanted to run **libcamera-hello**, you would type out:

```
DISPLAY=:0 libcamera-hello
```

The second way is to set the variable for the remainder of the SSH terminal session. To do this, just run the command every time you SSH into the Pi:

```
export DISPLAY=:0
```

The third way is to set it in a file that automatically runs each time you connect with SSH such as **.bashrc**. Of course this also assumes you continue using bash as your default shell. To make a backup of your current **.bashrc** file and add it, just run these commands:

```
cp ~/.bashrc ~/.bashrc.backup
echo "export DISPLAY=:0" &gt;&gt; ~/.bashrc
source ~/.bashrc
```

For the sake of simplicity, this guide will assume you are using one of the latter two methods. If you wanted to use the first method, just prefix it to any command that uses the camera such as the libcamera apps or any of the rpi-vision demos.

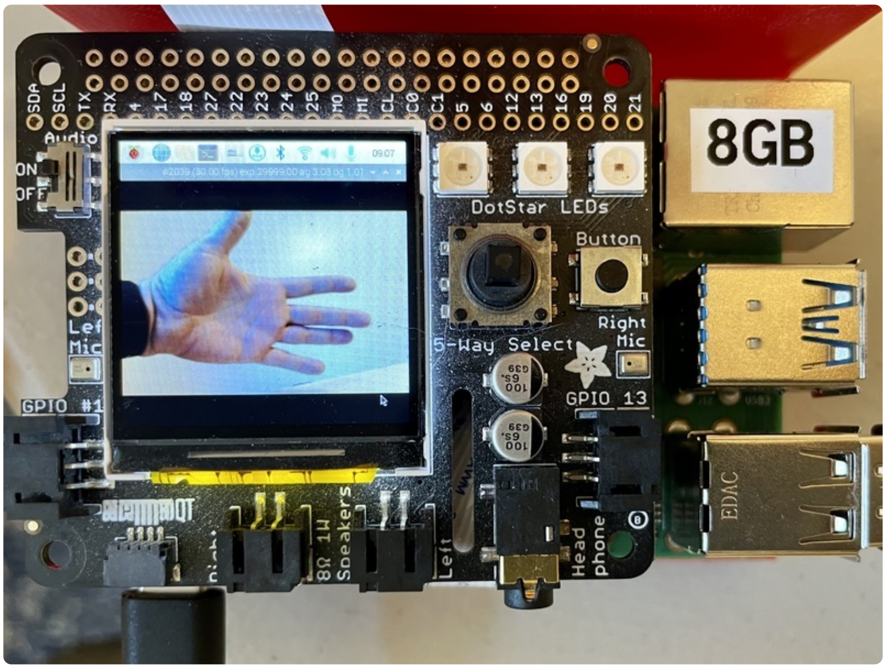
Camera Test

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

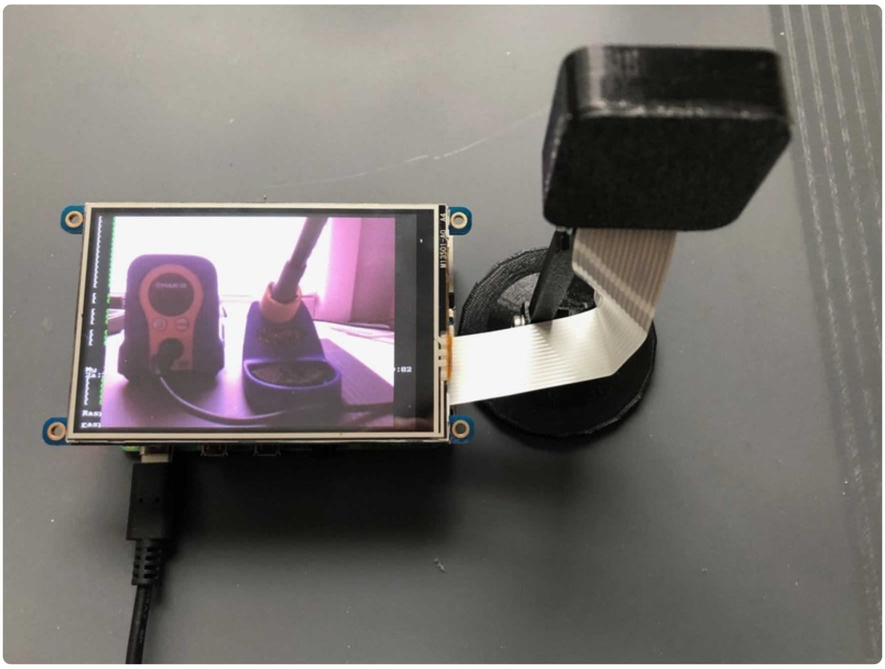
```
libcamera-hello -t 0
```

Press Ctrl+C to exit the test.

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



And on the PiTFT, it should look something like this:



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**. Also, be sure the camera cable is not inserted backwards. 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 2 Setup

Install requirements

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

```
sudo apt install -y python3-numpy python3-pillow python3-pygame
```

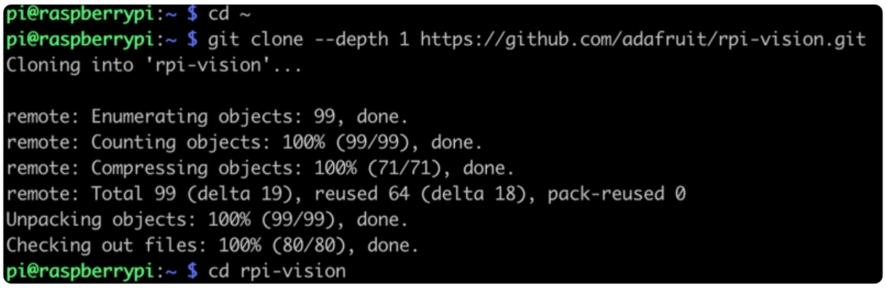
Install the Speech Output package

```
sudo apt install -y festival
```

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. While there is a MobileNet V3, it isn't as accurate as V2. This part will take a few minutes to complete.

```
cd ~
source env/bin/activate
git clone --depth 1 https://github.com/adafruit/rpi-vision.git
cd rpi-vision
pip3 install -e .
```



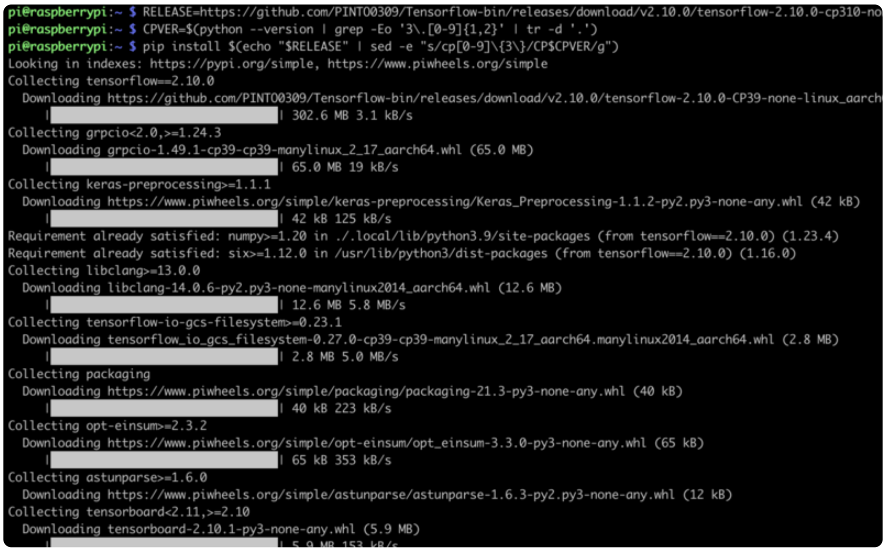
```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ git clone --depth 1 https://github.com/adafruit/rpi-vision.git
Cloning into 'rpi-vision'...

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
```

Install TensorFlow 2.x

Next you can download and install Tensorflow 2. There are a nice set of releases over on PINTO0309's GitHub Releases Page. (https://adafru.it/11sa) To install TensorFlow 2.15.0, run the following commands. This cryptic bit of shell code figures out the current version of CPython running on your Pi, substitutes that in, and then downloads and installs the appropriate release file.

```
RELEASE=https://github.com/PINTO0309/Tensorflow-bin/releases/download/v2.15.0.post1/
tensorflow-2.15.0.post1-cp311-none-linux_aarch64.whl
CPVER=$(python --version | grep -Eo '3\.[0-9]{1,2}' | tr -d '.')
pip install $(echo "$RELEASE" | sed -e "s/cp[0-9]\{3\}/CP$CPVER/g")
```



```
pi@raspberrypi:~ $ RELEASE=https://github.com/PINTO0309/Tensorflow-bin/releases/download/v2.10.0/tensorflow-2.10.0-cp310-no
pi@raspberrypi:~ $ CPVER=$(python --version | grep -Eo '3\.[0-9]{1,2}' | tr -d '.')
pi@raspberrypi:~ $ pip install $(echo "$RELEASE" | sed -e "s/cp[0-9]\{3\}/CP$CPVER/g")
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting tensorflow==2.10.0
  Downloading https://github.com/PINTO0309/Tensorflow-bin/releases/download/v2.10.0/tensorflow-2.10.0-CP39-none-linux_aarch
     |████████████████████████████████| 302.6 MB 3.1 kB/s
Collecting grpcio<2.0,>=1.24.3
  Downloading grpcio-1.49.1-cp39-cp39-manylinux_2_17_aarch64.whl (65.0 MB)
     |████████████████████████████████| 65.0 MB 19 kB/s
Collecting keras-preprocessing>=1.1.1
  Downloading https://www.piwheels.org/simple/keras-preprocessing/Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
     |████████████████████████████████| 42 kB 125 kB/s
Requirement already satisfied: numpy>=1.20 in ./.local/lib/python3.9/site-packages (from tensorflow==2.10.0) (1.23.4)
Requirement already satisfied: six>=1.12.0 in /usr/lib/python3/dist-packages (from tensorflow==2.10.0) (1.16.0)
Collecting libclang>=13.0.0
  Downloading libclang-14.0.6-py2.py3-none-manylinux2014_aarch64.whl (12.6 MB)
     |████████████████████████████████| 12.6 MB 5.8 MB/s
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.27.0-cp39-cp39-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (2.8 MB)
     |████████████████████████████████| 2.8 MB 5.0 MB/s
Collecting packaging
  Downloading https://www.piwheels.org/simple/packaging/packaging-21.3-py3-none-any.whl (40 kB)
     |████████████████████████████████| 40 kB 223 kB/s
Collecting opt-einsum>=2.3.2
  Downloading https://www.piwheels.org/simple/opt-einsum/opt_einsum-3.3.0-py3-none-any.whl (65 kB)
     |████████████████████████████████| 65 kB 353 kB/s
Collecting astunparse>=1.6.0
  Downloading https://www.piwheels.org/simple/astunparse/astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting tensorboard<2.11,>=2.10
  Downloading tensorboard-2.10.1-py3-none-any.whl (5.9 MB)
     |████████████████████████████████| 5.9 MB 153 kB/s
```

After this, go ahead and reboot the Pi.

```
sudo reboot
```

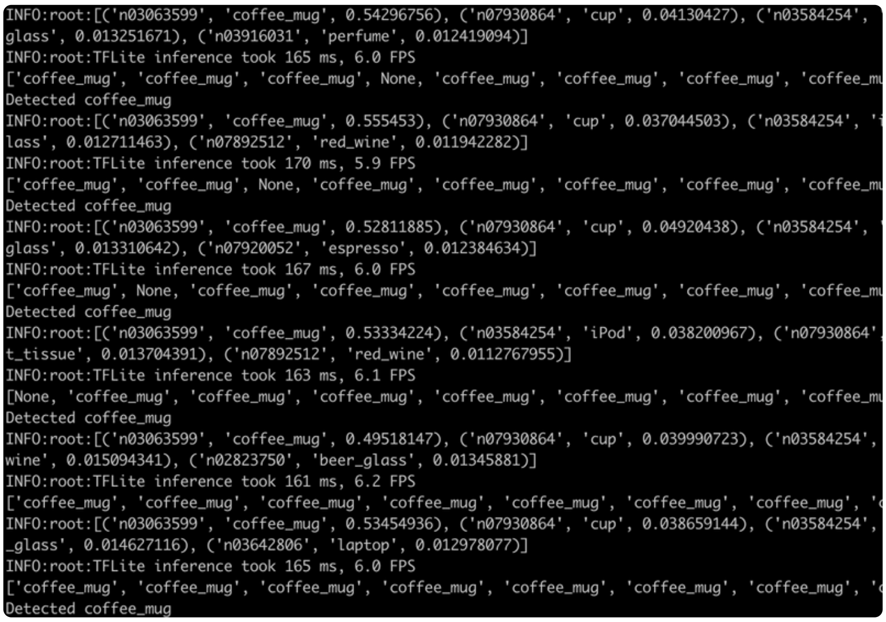
If you get a warning with "No module named 'rpi_vision'", this means Python couldn't load the module. Either the module or the Python binding likely wasn't installed correctly.

Running the Graphic Labeling Demo

Finally you are ready to run the detection software. To run a program that will display the object it sees on screen, type in the following:

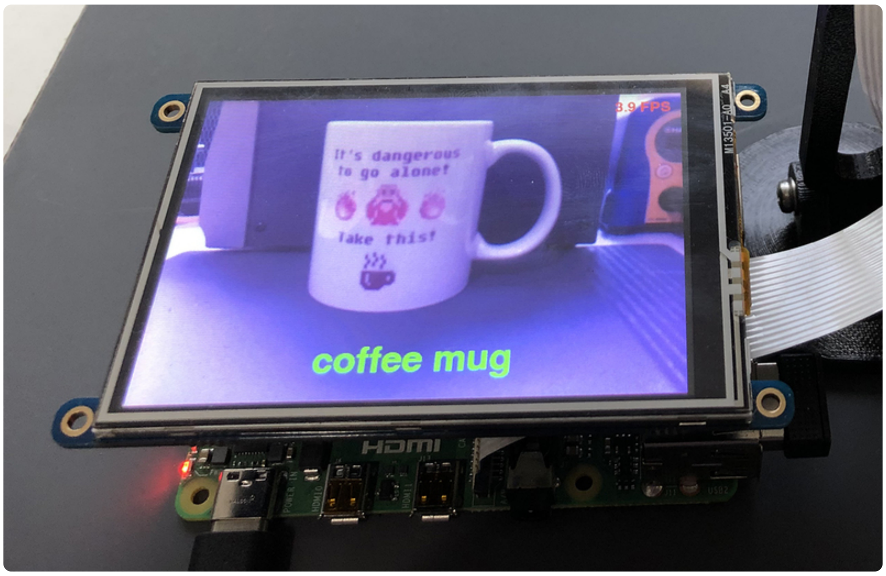
```
cd rpi-vision
python3 tests/pitft_labeled_output.py --tflite
```

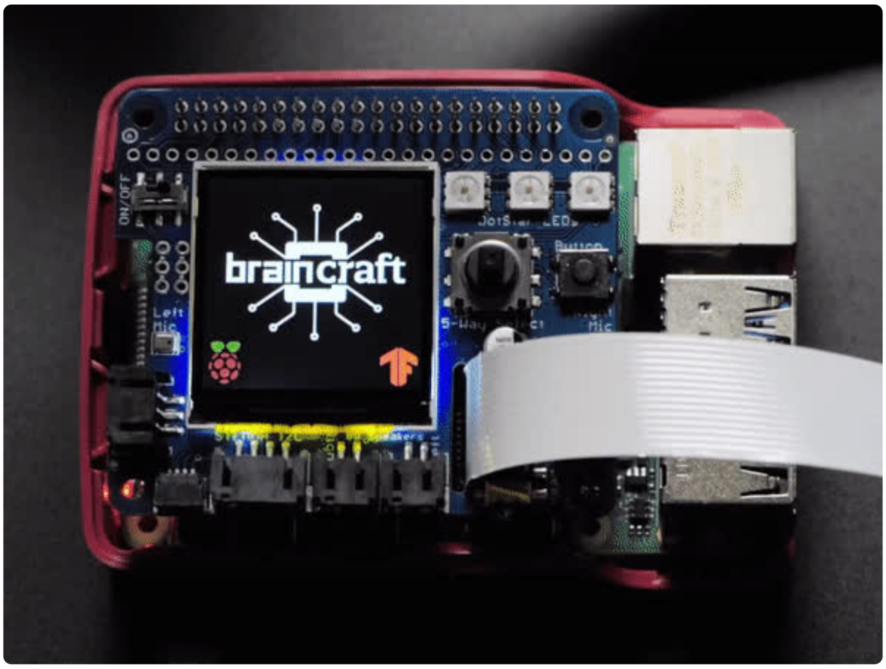
You should see a bunch of text scrolling in your SSH window.



```
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_mu
Detected coffee_mug
INFO:root:[('n03063599', 'coffee_mug', 0.555453), ('n07930864', 'cup', 0.037044503), ('n03584254', 'i
lass', 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_mu
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_mu
Detected coffee_mug
INFO:root:[('n03063599', 'coffee_mug', 0.53334224), ('n03584254', 'iPod', 0.038200967), ('n07930864',
t_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_mu
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', 'c
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', 'c
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.

If you don't hear anything, make sure the sound source you are expecting is selected. With the Raspberry Pi Desktop, you have a couple of ways to do this. If you have a mouse connected, you can right-click the speaker icon in the upper right-hand corner and choose your source.

If you prefer the command line, you can also run **raspi-config**, and go to **System Options → Audio** to select it.

If you still don't hear anything, make sure your volume is turned up by using **alsamixer**. For the BrainCraft HAT, this is described in more detail on the Audio Setup (https://adafru.it/Pf8) page.

Finally, you can test the text to speech from the command line by running the following command:

```
echo "This is a test" | festival --tts
```