



Raspberry Pi + Teachable Machine = Teachable Pi

Created by Andrew Reusch



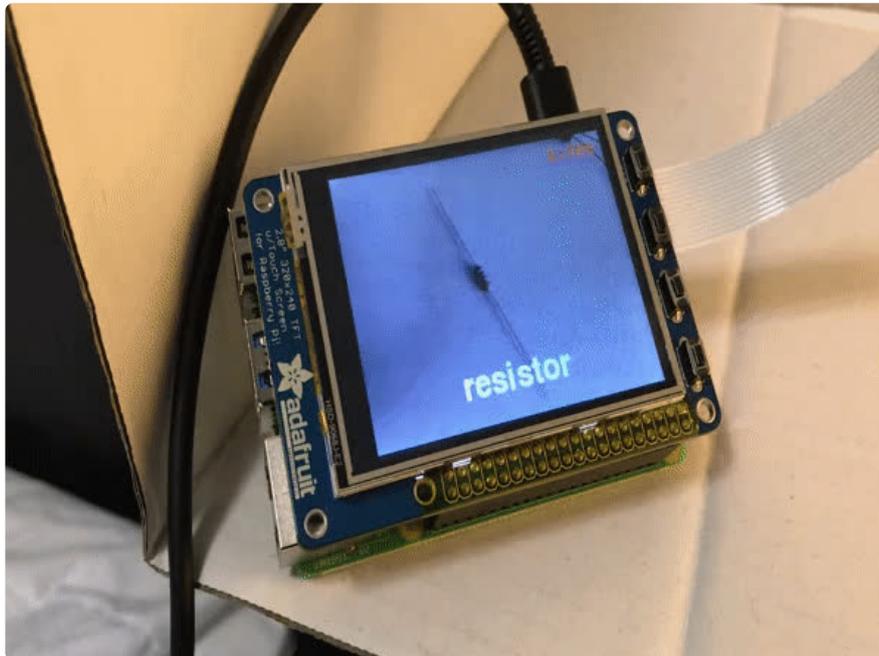
<https://learn.adafruit.com/teachable-machine-raspberry-pi-tensorflow-camera>

Last updated on 2025-07-23 01:00:37 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• You will need	
Visit Teachable Machine	3
<ul style="list-style-type: none">• Setting up Teachable Machine• Choosing Samples	
Create Categories	5
<ul style="list-style-type: none">• Adding Categories to your Teachable Machine project	
Use Computer Webcam	6
<ul style="list-style-type: none">• Capturing Samples• Before you start• Capturing the Samples	
Use Raspberry Pi Camera	8
<ul style="list-style-type: none">• Before you start• Setting up the Raspberry Pi• Launch the capture script• Record some data• Copy images to your desktop/laptop computer• Using Samba• Upload images to Teachable Machine• Using scp	
Training	12
<ul style="list-style-type: none">• Run Training• Previewing• Tweaking advanced settings	
Raspberry Pi Setup	15
<ul style="list-style-type: none">• Hardware• Stabilizing the camera• Software	
Running on the Pi	16
<ul style="list-style-type: none">• Exporting the SavedModel• Running the model on the Pi	

Overview



In our [previous demos](#), we got Google's Object Detection models running on a Raspberry Pi so that you could build your own smart camera! There's just one thing-- what if you want to recognize something else besides coffee cups and people? Until recently, this was challenging; either it required a complex Colaboratory notebook or a long series of local commands to set up a local TensorFlow environment.

Google recently released a tool called [Teachable Machine](#) that automates the whole process for you! You bring some example images, configure a few parameters on their web page, and voila! Google produces a custom model trained against your own objects.

This guide walks you through the process of configuring Teachable Machine, downloading the generated model to Raspberry Pi 4, and running the model on your Pi.

You will need

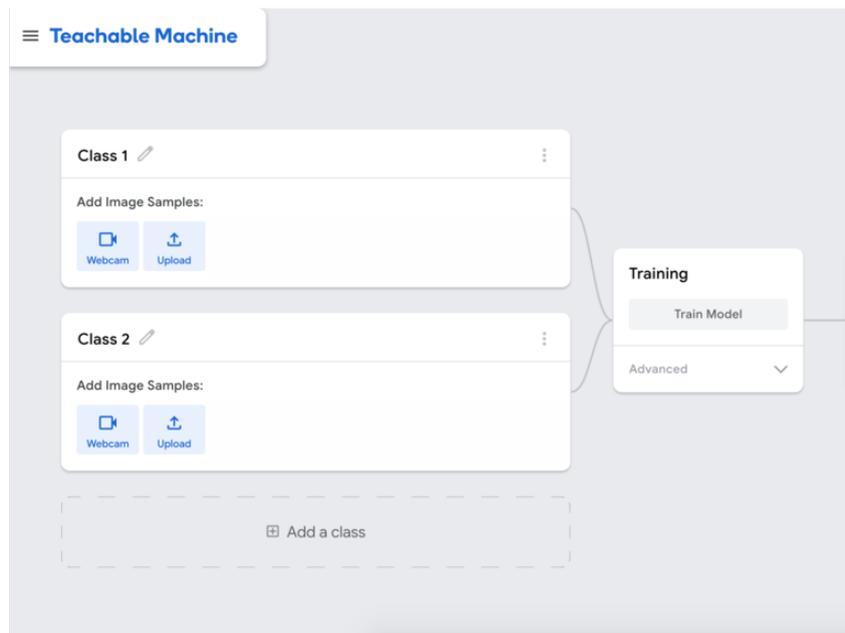
You'll need the same equipment as used in our [Running TensorFlow Lite Object Recognition on the Raspberry Pi 4](#) guide.

Visit Teachable Machine

Setting up Teachable Machine

To start with, visit the [Teachable Machine](#) page on your **desktop** or **laptop** (not the Pi-- it's too slow).

Click **Get Started**, then choose **Image Project** from the first screen, and you'll be presented with this empty project screen:



Now it's ready for you to input samples. Before you start, have a look at the guidelines for choosing and capturing samples below.

Choosing Samples

Now you need to find example images that include the objects you want to recognize. This can be the both the most important and most time-consuming part of retraining a machine learning model, so choose wisely!

Here are some tips for choosing example images:

- **Choose realistic images.** Your model won't behave predictably in scenarios it hasn't seen before.
- **Choose varied images.** It's really important that the only thing in common between the images is the thing you want to detect. As an example, if all of the images in one category have a blue background, the model might simply learn that a blue background indicates that category.
- **Choose representative images.** Make sure you include images that show all different angles of the object you'd like to recognize.

- **Include an "empty" category.** Try including a category that the machine can select when it isn't confident in any of the choices. This should have as much background variation as possible, and shouldn't include any objects you want to recognize.

Create Categories

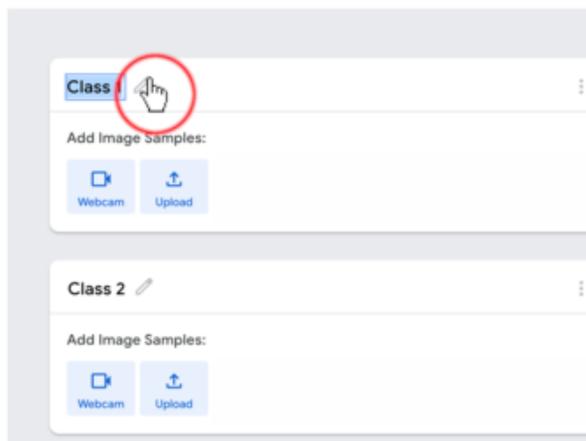
Now you need to create categories in the Teachable Machine UI to match the objects you've chosen. Usually, you should create one category for each distinct type of object you want to recognize--so, if you want to train your model to recognize different clothes, you might create categories for pants, dresses, shirts, and socks.

You can also try to train a smarter model, like one that looks for blue shirts or button-down shirts. But beware, this is advanced and we don't suggest you start with this.

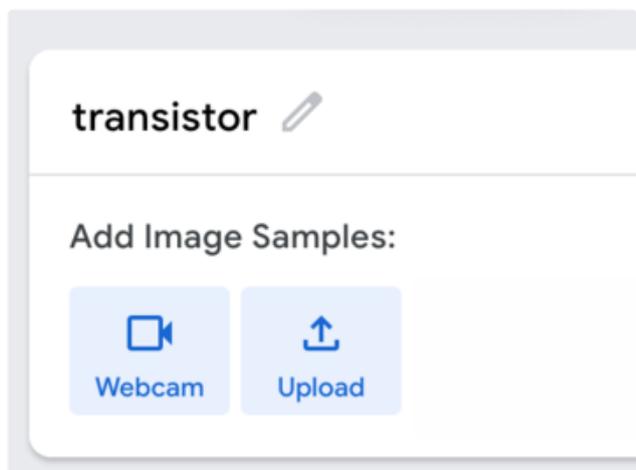
Also, be sure to create a nothing category--one that the machine can choose when it doesn't see anything. You'll need to find some example images that don't contain any of your chosen objects.

Adding Categories to your Teachable Machine project

Each box on the left of your page represents a different category. Change the name of each as follows:

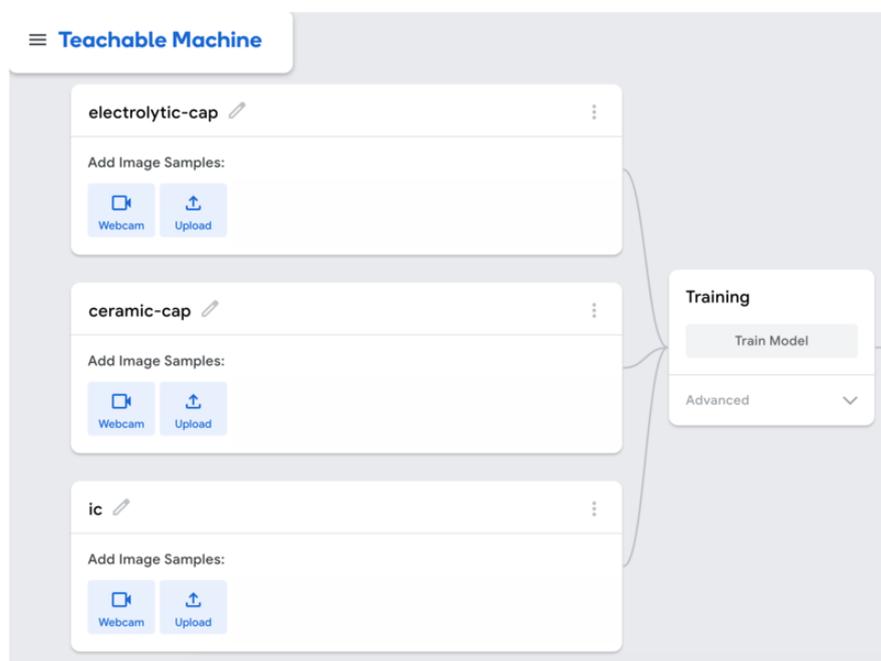


Click the pencil icon, then name the category.



Press **Enter** to accept the name. The box should look like this (e.g. for the transistor category).

Repeat these steps for each of the other objects you'd like to recognize. When you're finished, your screen should look like this:



Use Computer Webcam Capturing Samples

You can capture either from your computer's webcam or from the Pi's camera. When you use your computer's webcam, Teachable Machine captures the samples directly from the browser--super quick and easy! [When you use the Pi camera](#), you'll need to run a script to generate the samples, transfer them to your computer, and upload them.

For best results, you should capture samples using the camera you want to use. But, a MacBook Pro webcam worked well enough for us, and it's much easier to use. So, we

suggest you first try to use your computer webcam, and [use the Pi camera](#) if the model doesn't work well on the Pi.

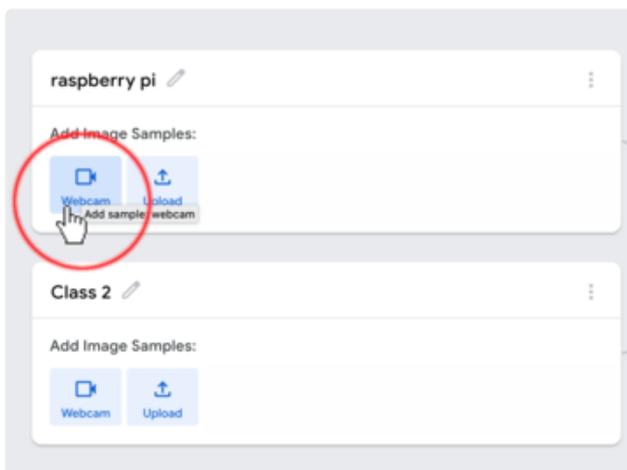
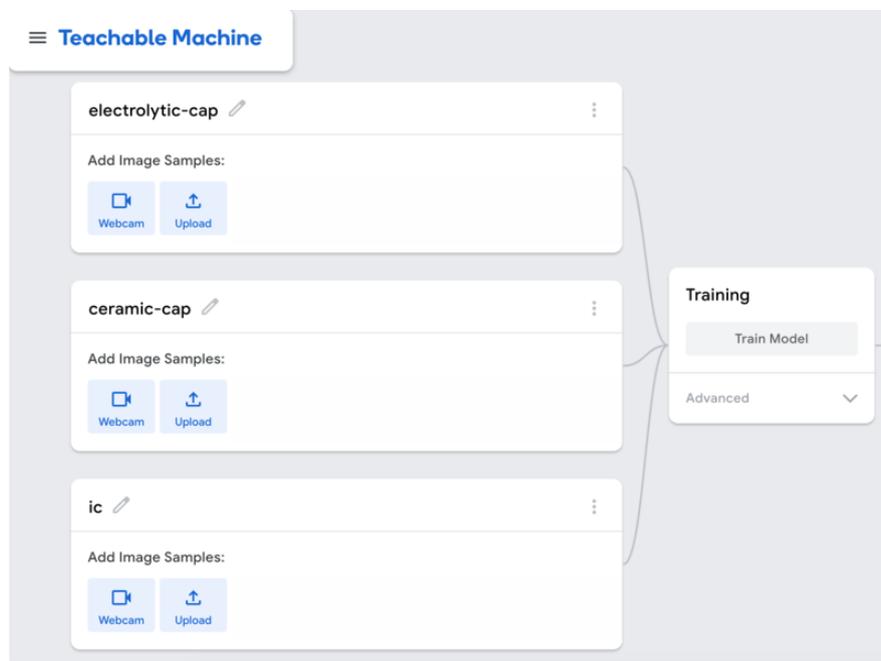
Before you start

Be sure you've followed the steps in [Create Categories](#) first.

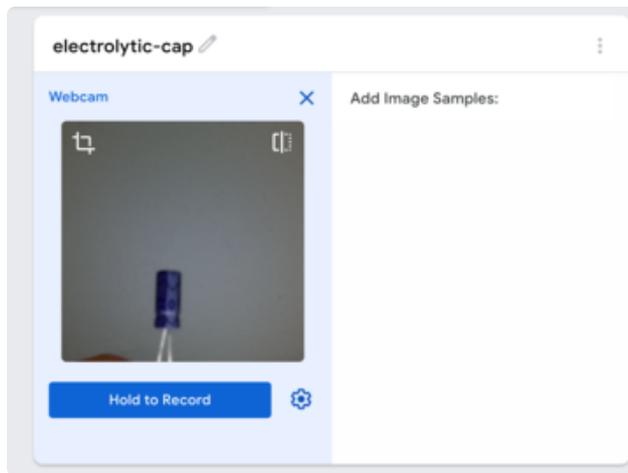
Though this method should work well enough, be warned that the model might not perform differently on the Pi because it uses a different camera. If this happens, you can try the [next method](#) to capture samples directly from the Pi camera.

Capturing the Samples

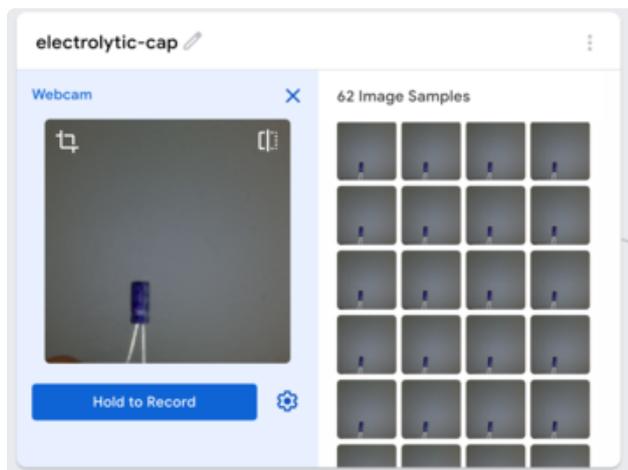
You should have Teachable Machine up on your laptop or desktop to a screen like this:



Now, click the Webcam button. You might need to allow the page access to your computer camera.



Click and hold the **Hold to Record** button to capture samples!



Samples will appear next to the webcam image.

Try to capture as many unique samples as you can. Remember:

- Try to capture around 200 samples per category.
- Capture different angles and light levels
- Make sure the samples are as representative of the object as they can be
- Try to move the camera around so that the only thing in common between all of the sample images for a given object is the object itself.

Use Raspberry Pi Camera Before you start

Be sure you've followed the steps in [Create Categories](#) first.

Setting up the Raspberry Pi

First, follow the instructions in [Raspberry Pi Setup](#). This will load the software onto the Pi.

Launch the capture script

SSH to the pi, or open a Terminal. Then, run these commands to set up the shell:

```
$ cd rpi-vision
$ sudo bash
# source .venv/bin/activate
```

Get ready to capture the first samples. Then, at the same shell, type:

```
# python3 tests/pitft_capture.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Ready. Press ENTER to toggle capture/standby. Press Ctrl+C to quit.

You may need to adjust the focal distance of your Pi camera by rotating the lens. The Raspberry Pi Camera Board v2 includes a handy adjustment wheel.



At this point, the Raspberry Pi display should look like this.

Record some data

At this point, the **STANDBY** text indicates that the Pi is not recording data. Make sure the camera is pointed at your object, then press **ENTER** to toggle between **STANDBY** and **RECORD** modes.



In **RECORD** mode, the script is capturing frames and saving them to the Pi's SD card. The display looks like this.

Press **ENTER** to toggle between **STANDBY** and **RECORD** modes.

Be sure you see the frame count increasing. This means that the Pi is saving data to the SD card.

When finished, press **CTRL + C** to stop the process.

Copy images to your desktop/laptop computer

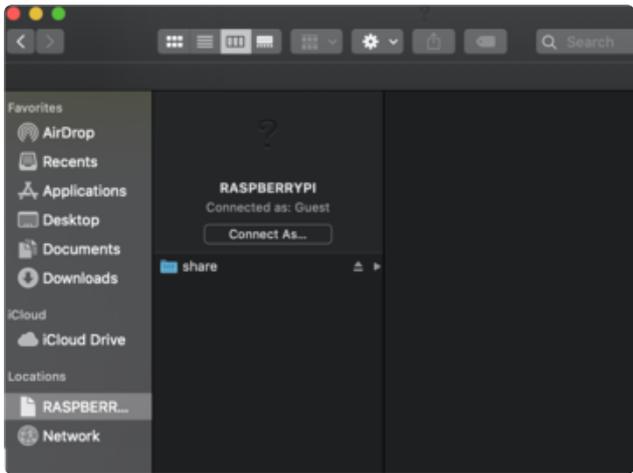
Here we show you how to use either `samba` or `scp` to copy the generated tar to your desktop/laptop for use with Teachable Machine.

Using Samba

The easiest way is to setup your Raspberry Pi as a file server using **Samba**. Follow [these steps](#).

Now, copy the generated tarball into your samba share:

```
$ cp ~/rpi_vision/⟨⟨OBJECT_NAME⟩⟩.tar /share
```



On your local desktop, look for the **RASPBERRYPI** machine on your network. Connect to the share and copy the tar to your local machine.

Using scp

On your local machine (i.e. not the raspberry pi), run:

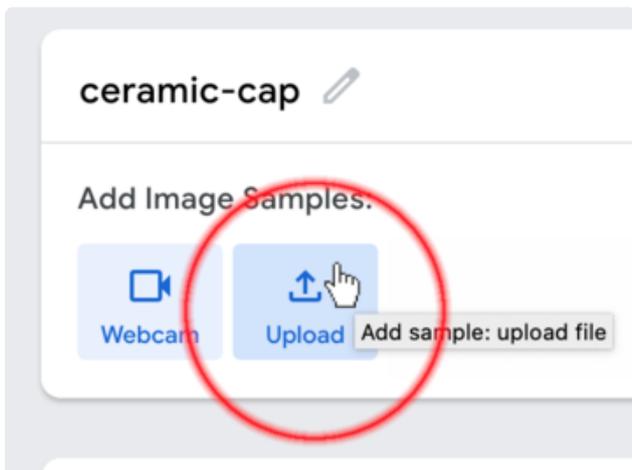
```
$ scp pi@raspberrypi.local:~/rpi_vision/<<OBJECT_NAME>>.tar .
```

Be sure to change **<<OBJECT_NAME>>.tar** to the actual path you specified to your **pitft_capture.py** command above.

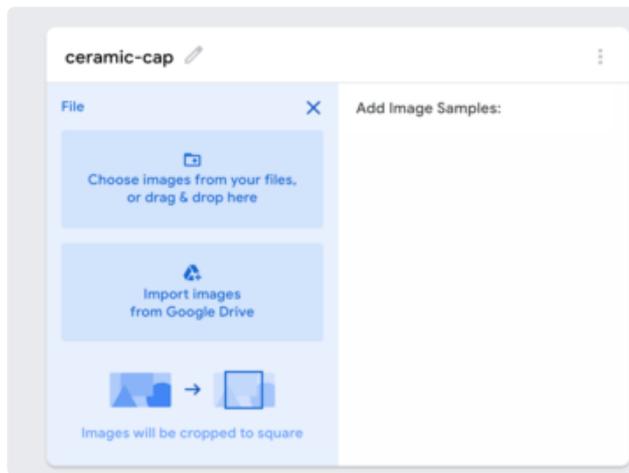
Upload images to Teachable Machine

Extract the tar by double-clicking it. On Windows, you may need a utility like [7-zip](#).

Now, in the Teachable Machine UI, follow these steps:

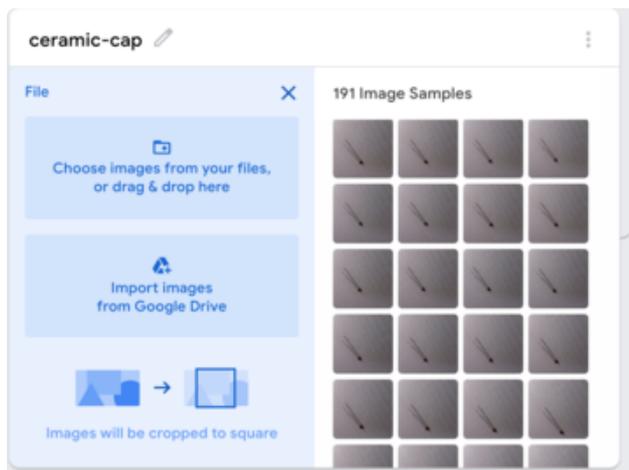


click the **Upload** button next to the class box these images belong to.



Now **drag-n-drop** the images from your file browser onto the dialog.

You can also click on the folder to find them from within the browser.

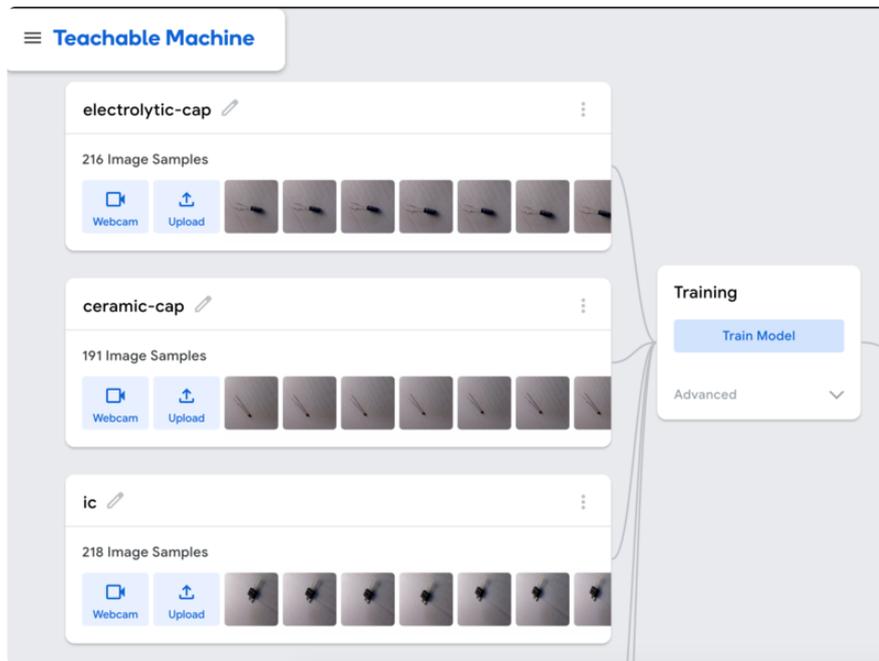


When done, the box looks like this.

Repeat this process for all categories in your project.

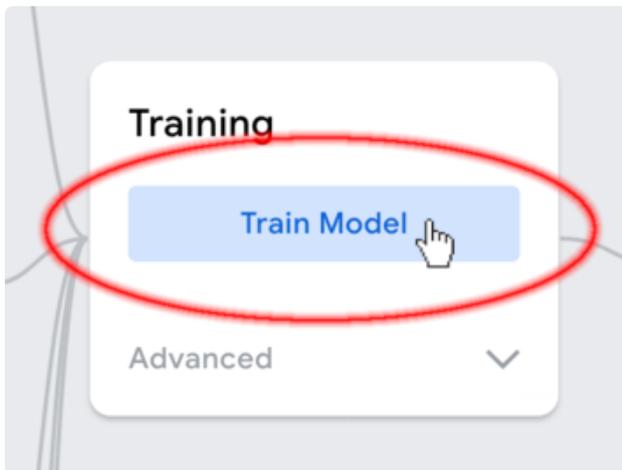
Training

When you've finished the last page (either [Use Computer Webcam](#) or [Use Raspberry Pi Camera](#)), your screen should look like the image below. Now you're ready to start training.



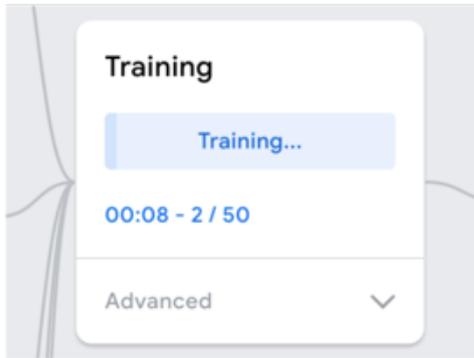
Run Training

Follow these steps to run the training process.



Now click **Train** and leave the tab open and visible. The training process happens in Javascript on your own machine, and will be slow or stalled if the browser window is left in the background.

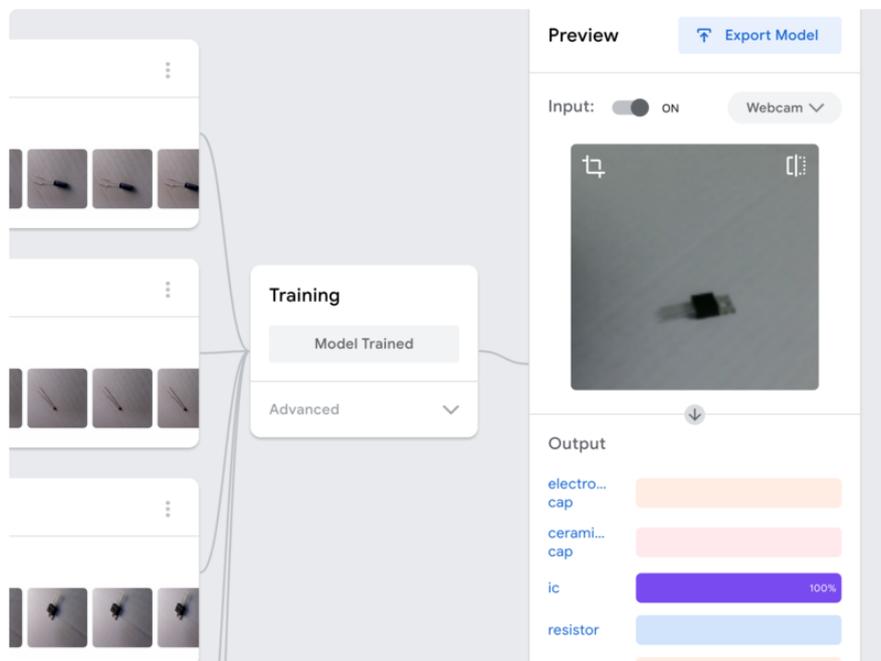
You may see a dialog telling you that the page is running too slowly. Don't click Kill Page! You'll need to start over again.



While training is in progress, you'll see a progress bar. Training should take around **5 minutes** or less using the default **50** epochs.

Previewing

After training, you can use your computer's webcam to try out the model. You can also upload images taken from the Pi.



Tweaking advanced settings

Sometimes your model may not perform as well as you like on the first run. There are a couple of things you can do in this situation:

- **Run for more epochs.** This is the easiest thing--just click the Advanced drop-down in the Training box, and change the number of epochs. Your training time will increase more-or-less linearly with the number of epochs. Try 100 or 200 to start.
- **Use different samples.** This will produce the most change in your results, but takes more time.

Raspberry Pi Setup

If you already captured images from the Raspberry Pi camera, you don't need to do this page twice.



Hardware

The camera and touchscreen display are attached to the Raspberry Pi. Plug in the camera first, then the touchscreen. Be sure the Pi is not powered while assembling hardware.

Here are the links to instructions for the hardware we use in this guide:

- [Camera Install Instructions](#)
- **Touchscreen Installation:** The [PiTFT](#) just needs to be attached to the 2x20 header so that the board is stacked directly on top of the Pi. The PiTFT module should fit snugly enough in the header port that you shouldn't need to attach any spacers.

Stabilizing the camera

Attach the camera to a larger, more stable object so that it doesn't bounce around on the cable. I taped the camera to the box it shipped in, like so:



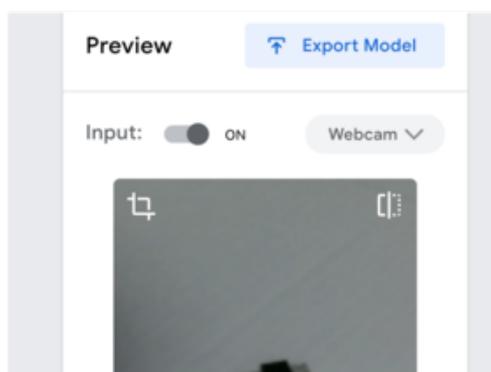
Software

Follow the [Initial Setup](#) and [TensorFlow Lite 2.0 Setup](#) steps. You can stop when you reach the **Running the Graphic Labelling Demo** section.

Running on the Pi

Exporting the SavedModel

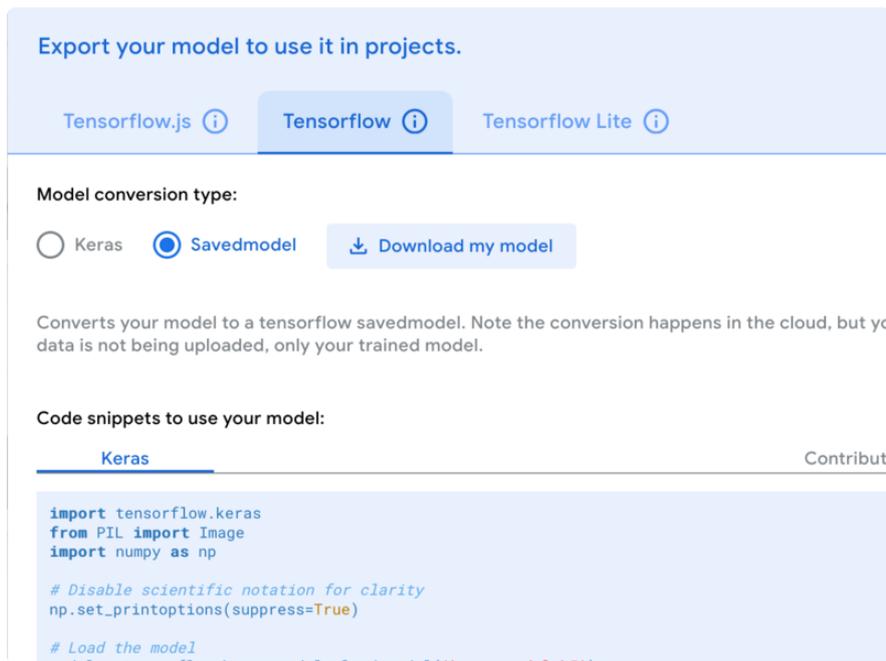
The first step is to export your shiny new model in **SavedModel** format. This makes it easy to transfer all of the files needed to the Pi, since it comes in a zip file.



Click **Export Model** in the Preview box.

In the dialog that appears, choose **TensorFlow** and be sure the **SavedModel** button is selected, as shown below.

Then, click **Download my model**. After a minute or two, your browser will save a `converted_savedmodel.zip` file.



Running the model on the Pi

When you clicked the **Download my model** button in the last step, your browser should have saved a file called `converted_savedmodel.zip`. Now you need to place this on your Raspberry Pi.

You can transfer the model to the Raspberry Pi using `scp`:

```
$ scp <<PATH_TO_DOWNLOADS>>/converted_savedmodel.zip pi@raspberrypi.local:~
```

NOTE: You might need to adjust the first path in the command above to point to your Downloads folder.

Now, SSH to your Pi and run:

```
$ cd rpi-vision
$ sudo bash
# source .venv/bin/activate
# python3 tests/pitft_teachablemachine.py ../converted_savedmodel.zip
```

It will take some time to load the model into memory. During this time, you'll see the BrainCraft logo appear on the display. Afterwards, point your rpi camera and try it out!

Be sure to follow these steps to get a file called `pitft_teachablemachine.py`. This file is not in the project GitHub repo, it is made from the training and part of the process noted above.

If you get an error that a module like `pygame` or `rpi_vision` isn't found, please use `pip` or `pip3` to load the module for Python use.

