**Face Recognition Using MathWorks on Raspberry Pi**

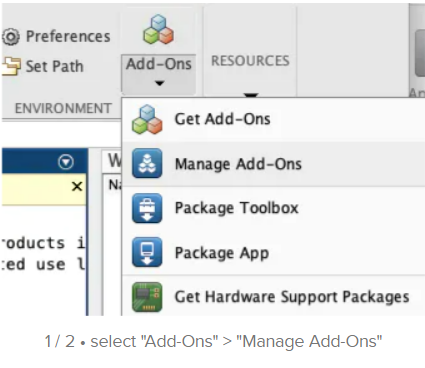
MATLAB is a powerful tool used by data scientists and artificial intelligence researchers, among others. Although there is a learning curve, it's worth diving in, because once you understand it well enough to get things done, you'll likely find it efficient and elegant. Their libraries and tools are hard to give up once you've had the chance to experience them. As a bonus, they offer support packages to generate and run code on a variety of platforms, including Raspberry Pi!

### Install the MATLAB Raspberry Pi support package.

First, install the support package in MATLAB.

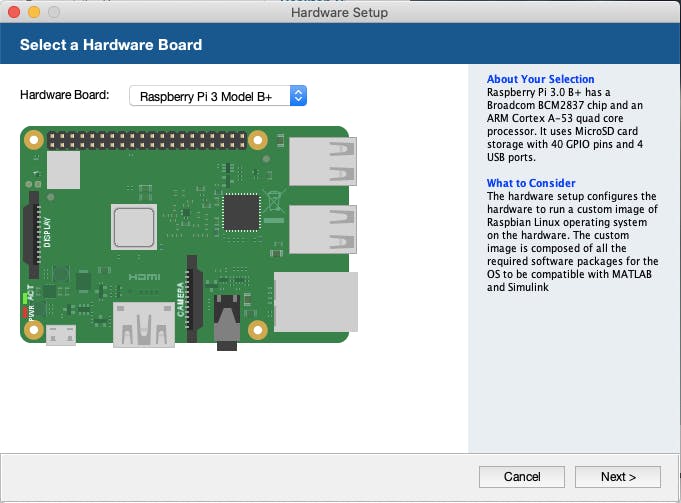
* Open MATLAB and on the home tab, select 'Add-Ons" > "Get Add-Ons."
* Search for "MATLAB Raspberry Pi support package" and select "Install."
* Accept the agreement.
* On the final screen of setup, choose "Setup now." That will open the board setup wizard.

Note: If you already closed it the installation wizard, you can find the image setup by selecting "Add-Ons" > "Manage Add-Ons" and then selecting the gear button (*setup*) next to the Raspberry Pi support package.



### Set up your pi to work with MATLAB.

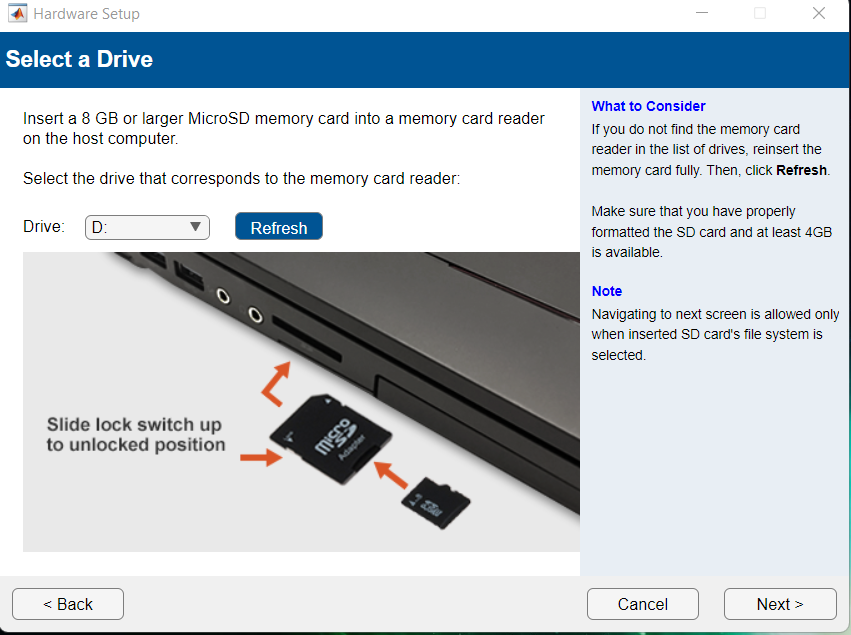
* On the first screen, select your type of Raspberry Pi. Click "next."
* Select "Setup hardware with MathWorks Raspian image."
* Download the deep learning image.
* Find your downloaded image and unzip it.
* Validate the image by using the "validate" button.



**Burn the image**

MATLAB was not able to burn the image on Mac, so I ended up simply using Balena Etcher to burn the image instead.

On the "Write Firmware" page, MATLAB checks the SD card to make sure it has been burned. I reinserted the SD card so that the burned image was visible to the program. That way I was able to select "Next" and skip to the next step.



### Set up the connection

There are four different connection options to connect your Pi to MATLAB:

**1. Connect to LAN or home network**

**2. Connect to wireless network**

**3. Connect directly to host computer**

**4. Manually enter network settings**

A screenshot of a computer

Description automatically generated with medium confidence

* In MATLAB, choose "**Connect to wireless network."** Fill in the SSID and password that both the board and PC are connected to (note - my Raspi 3 B+ is not able to connect to 2.4GHz Wi-Fi networks. Apparently, it's a known issue)
* Select "**Automatically get IP address**." Click "next."

Normally, MatLab sets the network connection up on your board automatically, but since we had to skip the previous imaging step, we'll have to set up the WIFI on the Pi manually. To set up WIFI on the Pi, you may wish to connect a keyboard, mouse and HDMI monitor, although you can also do it via SSH.

* Remove the SD card from your computer and insert it in your Raspberry Pi.
* Connect the Pi to a power supply of at least 5V 2A via the micro-USB.
* Connect the camera to the board by inserting the MIPI-CSI cable with blue side facing the pi's ethernet connectors, and the other blue side facing away from the camera lens.
* Click "**next**" in MATLAB to verify your connection.
* You should see the screen that says "Test Hardware Connection successful". If not, double check that your pi and PC running laptop are connected to the same network and try running "test connection" again.

### Connect your Pi to MATLAB

mypi = raspi

writeLED(mypi,'led0',1)  
writeLED(mypi,'led0',0)

cam=cameraboard(mypi, 'Resolution', '1280x720')

A screenshot of a computer

Description automatically generated with medium confidence

clear

## Code

%% set up your raspberry pi hardware peripherals access  
mypi = raspi();  
myCam = cameraboard(mypi)  
  
%% Image acquisiton & plotting  
  
mySnap = snapshot(myCam)  
imshow(mySnap)  
hold on  
  
%% Detect face and insert a rectangle around the face  
  
fD = vision.CascadeObjectDetector(); %% from computer vision toolbox  
bbox = step(fD, mySnap);  
imageOut = insertObjectAnnotation(mySnap, 'rectangle', bbox, 'Face');  
imshow(imageOut);  
  
%% Repeat in a loop to make it look like face detection in video  
flag = 1;  
  
while flag  
clearvars -except mypi myCam flag  
mySnap = snapshot(myCam);  
imshow(mySnap);  
hold on  
fD = vision.CascadeObjectDetector();  
bbox = step(fD, mySnap);  
imageOut = insertObjectAnnotation(mySnap, 'rectangle', bbox, 'Face');  
imshow(imageOut);  
title('Detected Face');  
drawnow %% refreshes the image window after every loop  
end