Raspberry Pi / Internet of Things

BootStrap

Objectives

This bootstrap is designed to help you be familiar with:

1) How to set up a Raspberry Pi along with peripheral units
2) How to connect to a Raspberry Pi remotely
3) How to use a camera connected to a Raspberry Pi.
4) How to detect motion using a camera feed using Tensorflow Lite

To finish this bootstrap and get the cash awards, you need to **finish the BootStrap Learning List and the Bootstrap Project** shown in this document.

Resources

1) Raspberry Pi imaging software: https://www.raspberrypi.com/software/
2) VNC Viewer download: https://www.realvnc.com/en/connect/download/viewer/
3) How to install a Raspberry Pi camera onto a Raspberry Pi: https://youtu.be/GImeVqHQzsE
4) Python Camera Library Documentation: https://datasheets.raspberrypi.com/camera/picamera2-manual.pdf
5) Raspberry Pi Camera Documentation: https://www.raspberrypi.com/documentation/accessories/camera.html
6) Raspberry Pi Camera Software Documentation: https://www.raspberrypi.com/documentation/computers/camera_software.html
7) MoveNet download: https://tfhub.dev/google/movenet/singlepose/lightning/3

Acquiring the Parts

You will need:

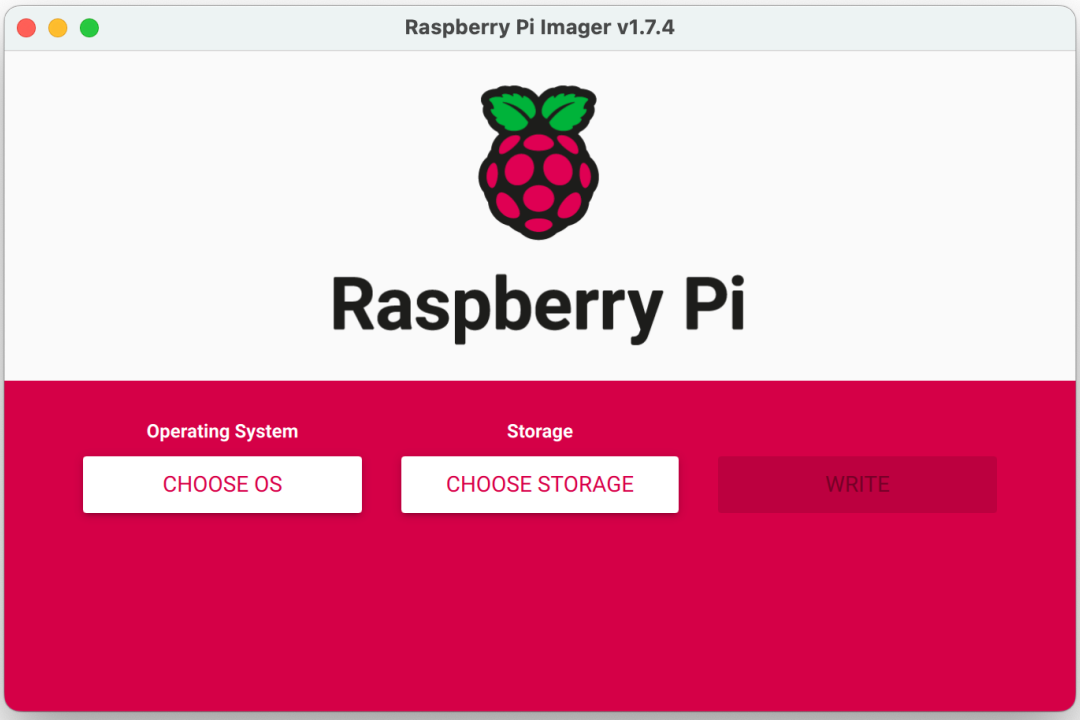
- A Raspberry Pi 4
- A 128 GB MicroSD Card
- A 5V 15W+ USB-C Power Supply
- A Raspberry Pi Camera
- A MicroSD Card Adapter
- For initial set up:
    - A Micro HDMI to HDMI cable
    - A monitor, mouse, and keyboard

If you live near our Irvine office, you can stop by to pick up the supplies up there. Otherwise, you will need to purchase the items yourself.

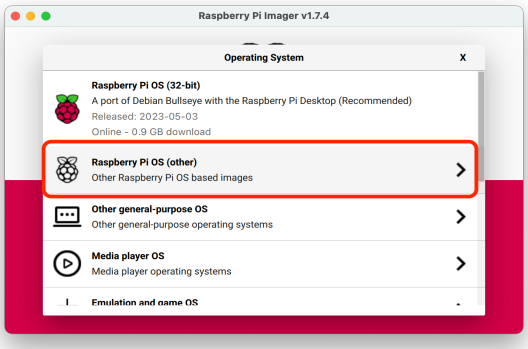
Bootstrap Instructions

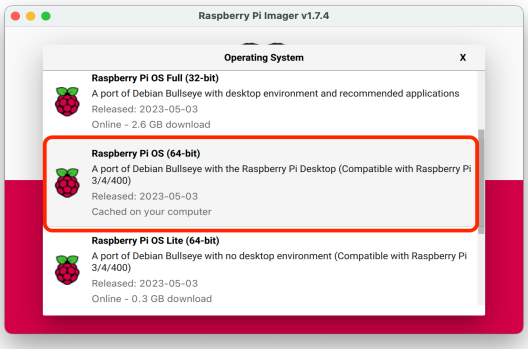
Installing the OS

Download the Raspberry Pi Imager software from https://www.raspberrypi.com/software/. The imager is available on MacOS, Windows and Ubuntu. Launch the Raspberry Pi Imager software.

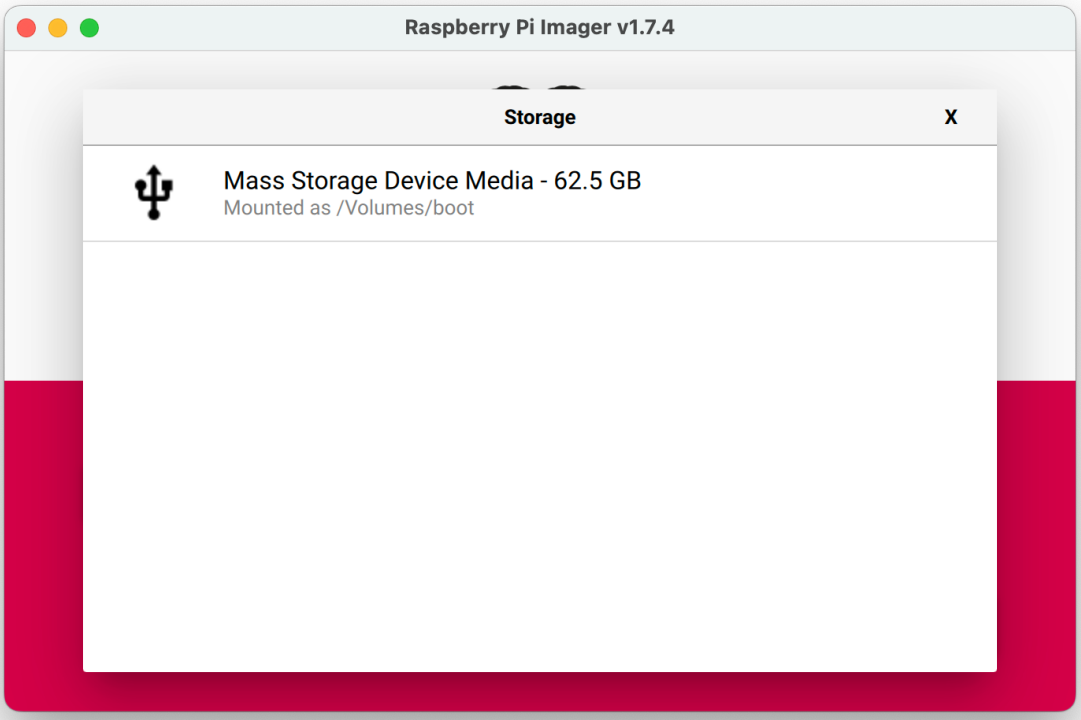


Under **Operating System**, select the **Choose OS** button. In the new window, select on **Raspberry Pi OS (other)** then select **Raspberry Pi OS (64-bit)**.



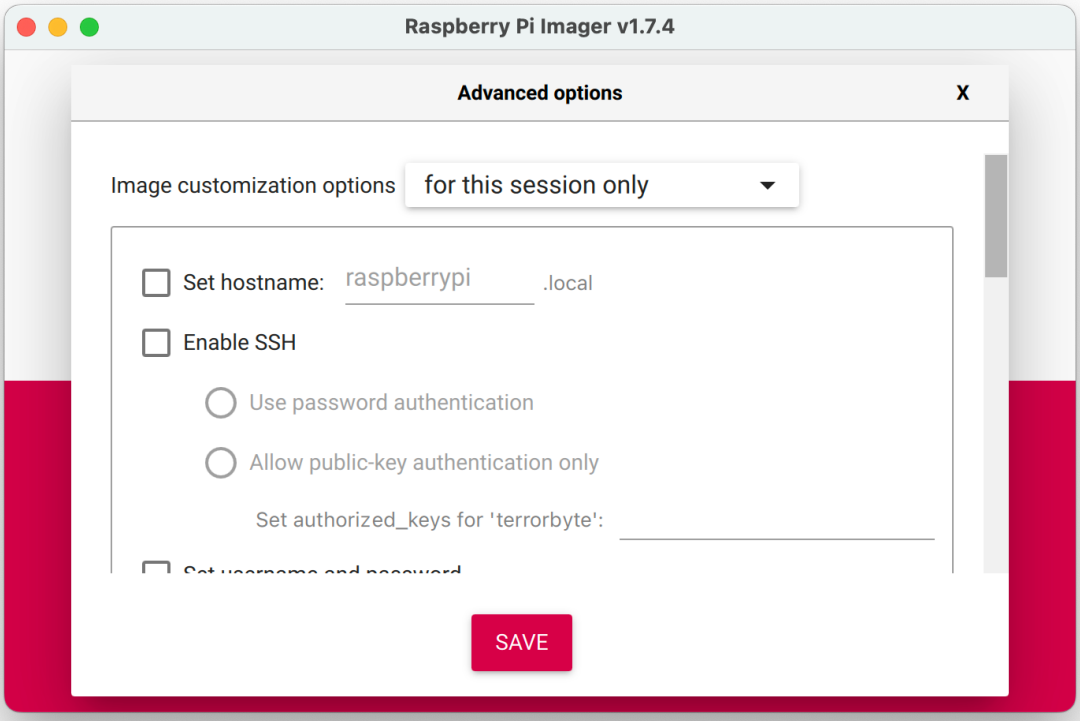


Connect the microsd card to your computer. Under **Storage,** select the **Choose Storage** button, then select the SD card from the list.

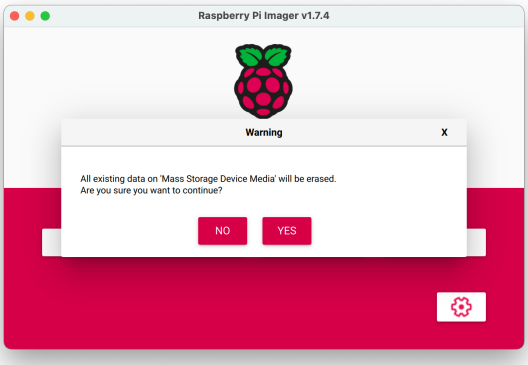


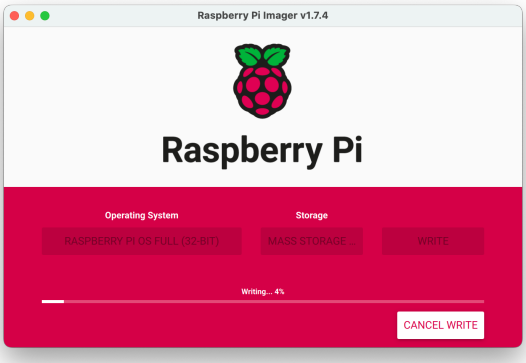
Click on the gear button on the bottom left of the window to enter the **Advanced Settings.** Set the following settings:

- Set the **hostname** to the project's name with "pi" in front ex: pi-yoga
- Turn on **Enable SSH** with **Use password authentication**
- Set the username to "pi" with the password "codingminds"
- Turn on **Configure wireless LAN** using the CodingMinds office wifi credentials, also set the **Wireless LAN country** to **US**
- Turn on **Set Local settings**, set the **Time zone** to **America/Los_An**gles and the **Keyboard layout** to **us**
- Click on the save button to save your changes.



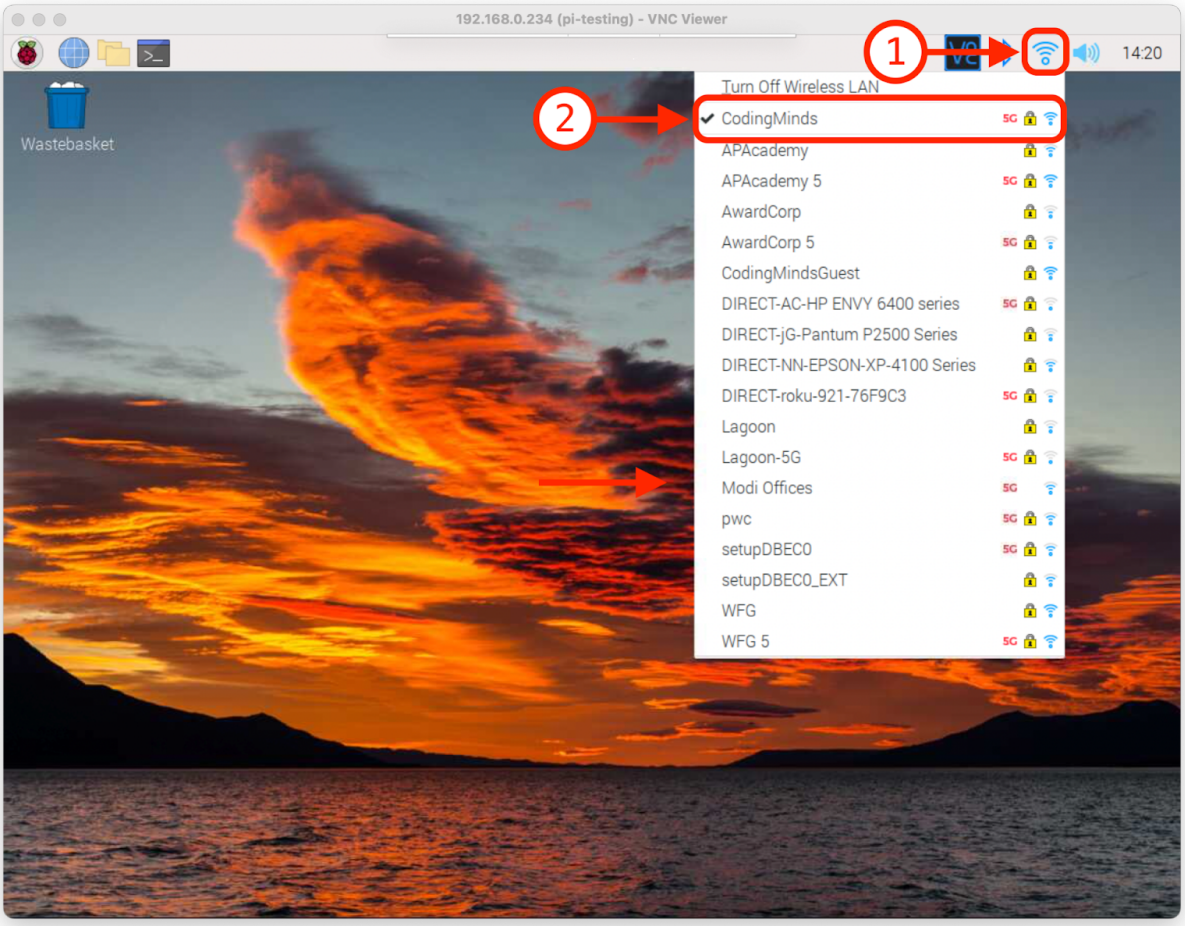
Click on the **Write** button to write the **Raspberry Pi OS (64-bit)** onto the SD card. A warning window appears warning the sd card will be erased and asking to to continue, select the **Yes** button to continue and override the sd card with the operating system. This step will take a few minutes.



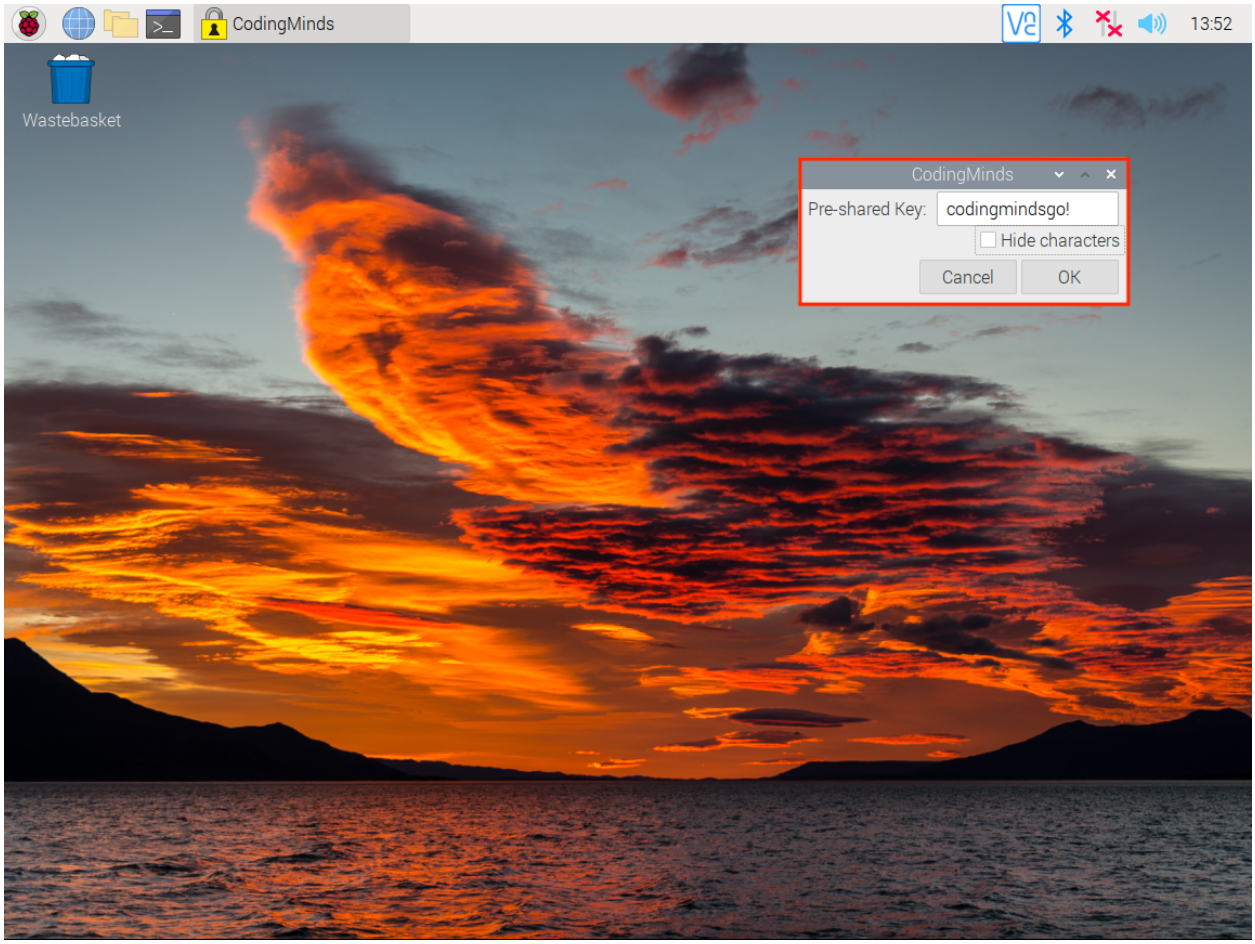


First Time Setup

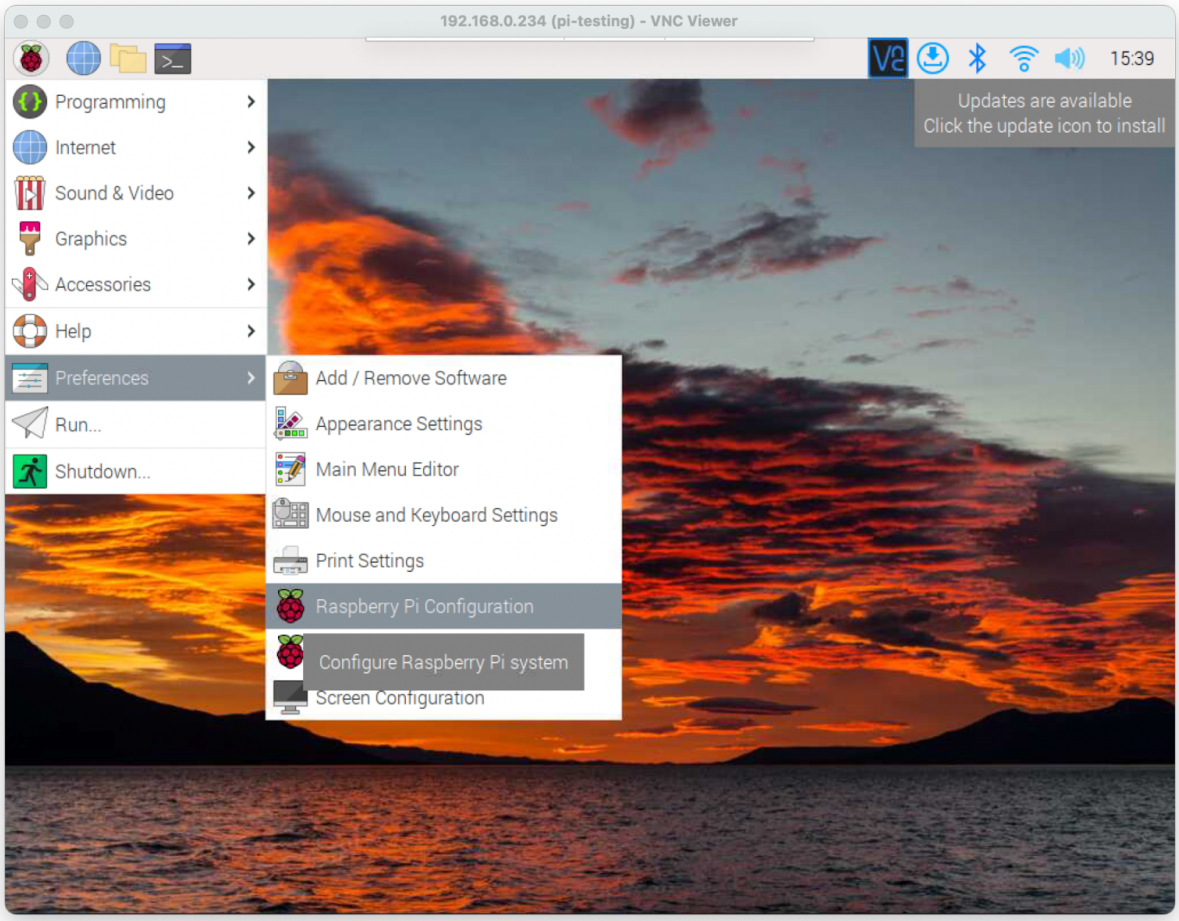
Once the Raspberry Pi OS is installed on the SD card, mount the SD card to the Raspberry Pi.
Next to the Raspberry Pi connect a monitor, a mouse, a keyboard and the power supply. When you connect the power supply, the Raspberry Pi is powered on.
If you followed the steps correctly on Installing the Raspberry Pi OS section, the Raspberry Pi should auto connect to the office network. If not, follow these stops to set up the wifi configurations:
On the top right corner of the Raspberry Pi; select the wifi icon then select the network you want to join.



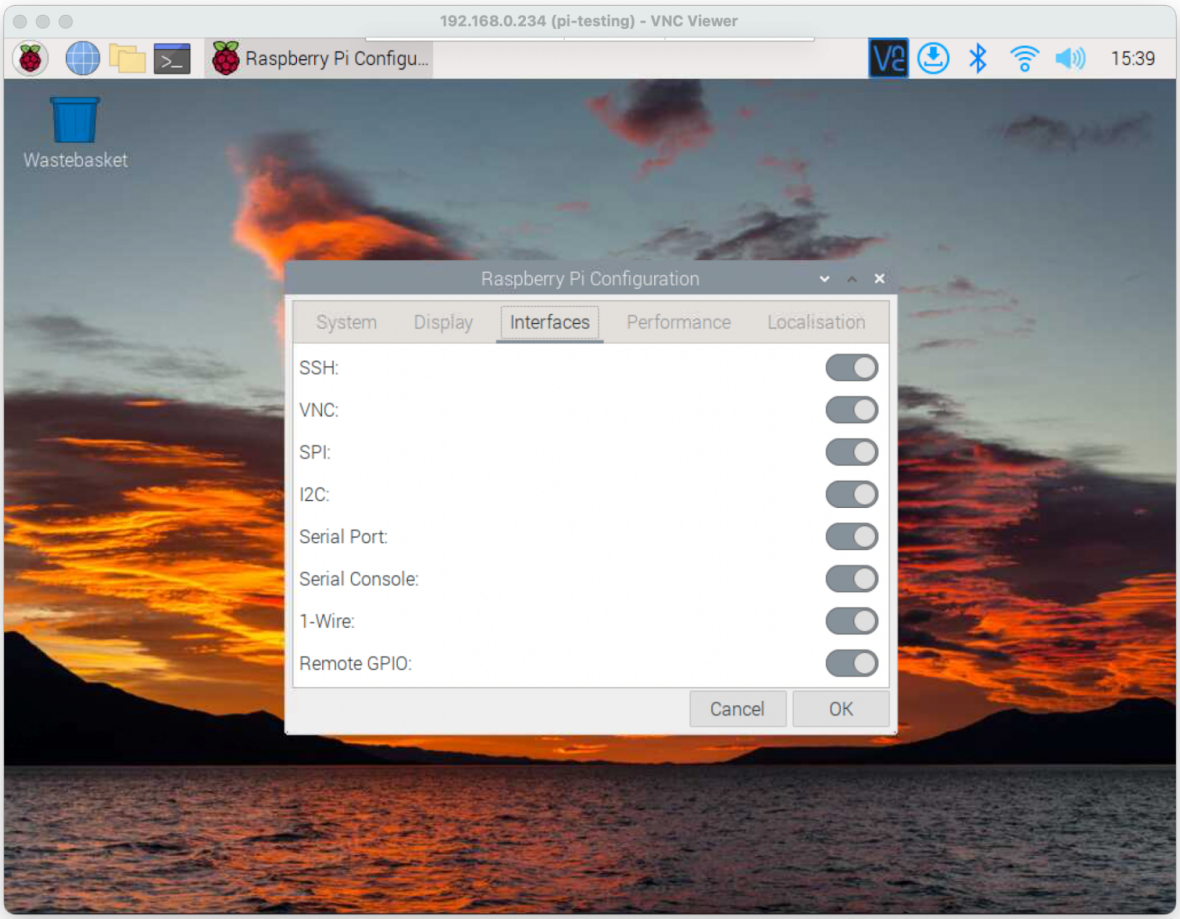
Enter the password to the wifi network to join

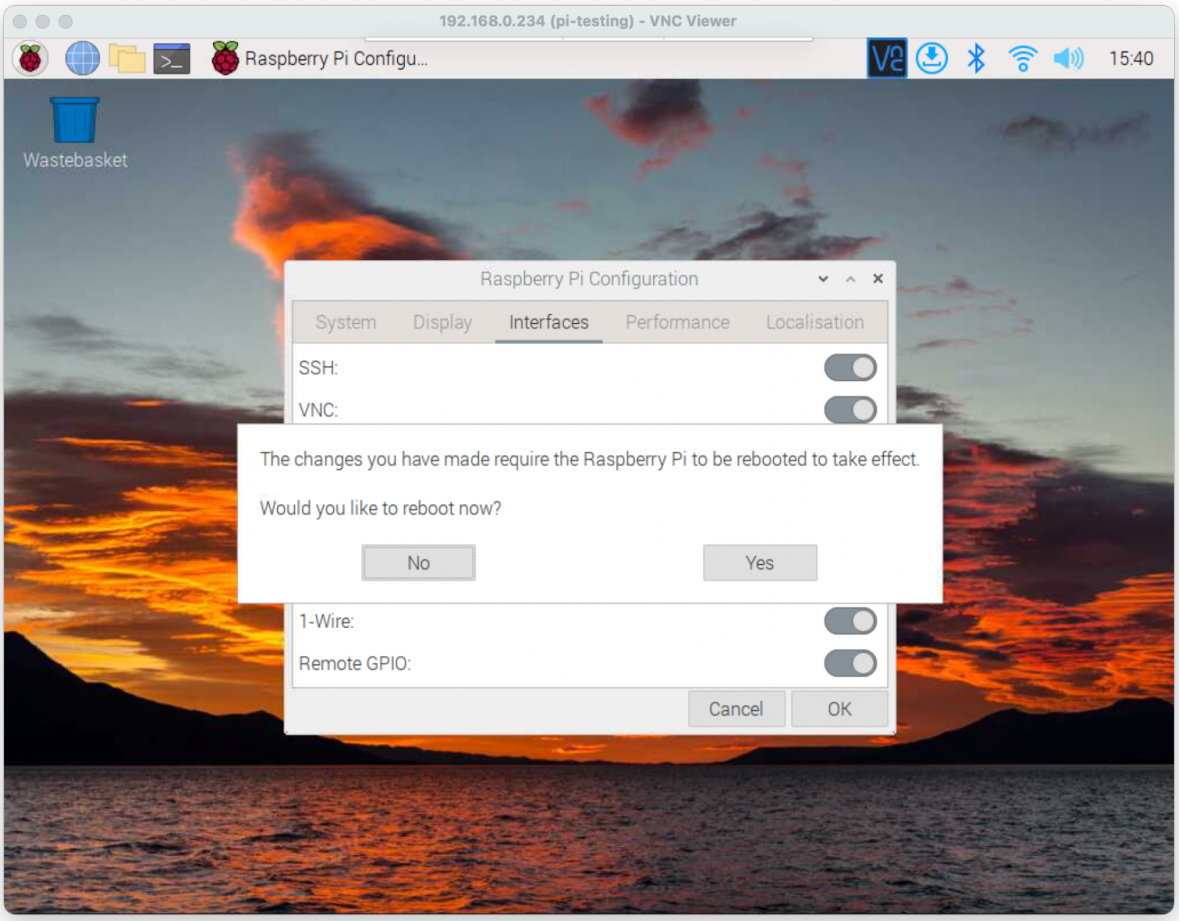


When the Raspberry Pi is connected to your network, click on the **Raspberry Pi Start** button at the top left, it is the Raspberry Pi logo. Go to **Preferences,** then select the **Raspberry Pi** configurations option opening the **Raspberry Pi Configurations** window.

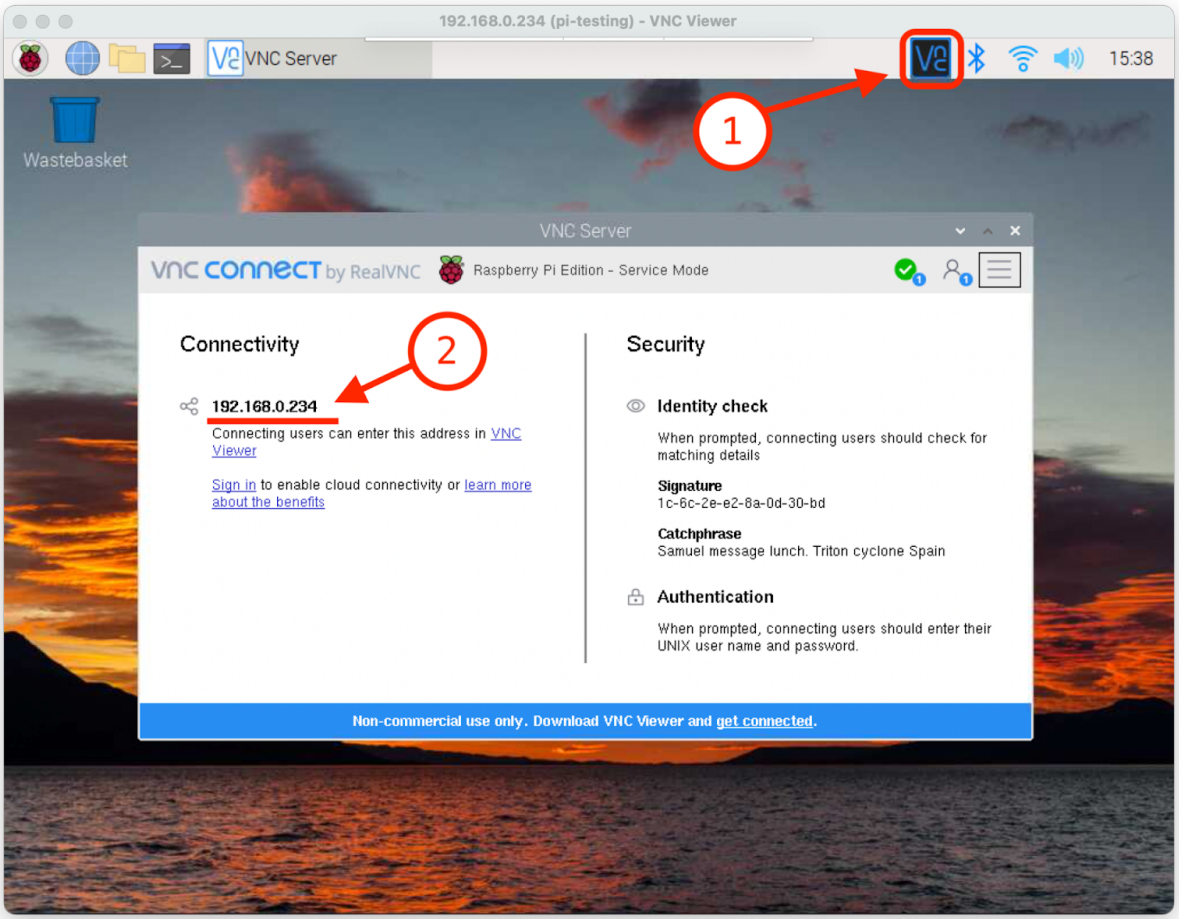


In the **Raspberry Pi Configurations** window, select the **Interfaces** tab. Enable all the options, then select the **OK** button. A window will prompt you to restart the Raspberry Pi, select **Yes**.


192.168.0.234 (pi-testing) - VNC Viewer
Raspberry Pi Configu...
15:39
Wastebasket
Raspberry Pi Configuration
System
Display
Interfaces
Performance
Localisation
SSH:
VNC:
SPI:
I2C:
Serial Port:
Serial Console:
1-Wire:
Remote GPIO:
Cancel
OK



After the Raspberry Pi has rebooted, The VNC server will be running and an VNC icon will be visible on the top right. Click on it to open a **VNC Server** window. Note the IP Address.

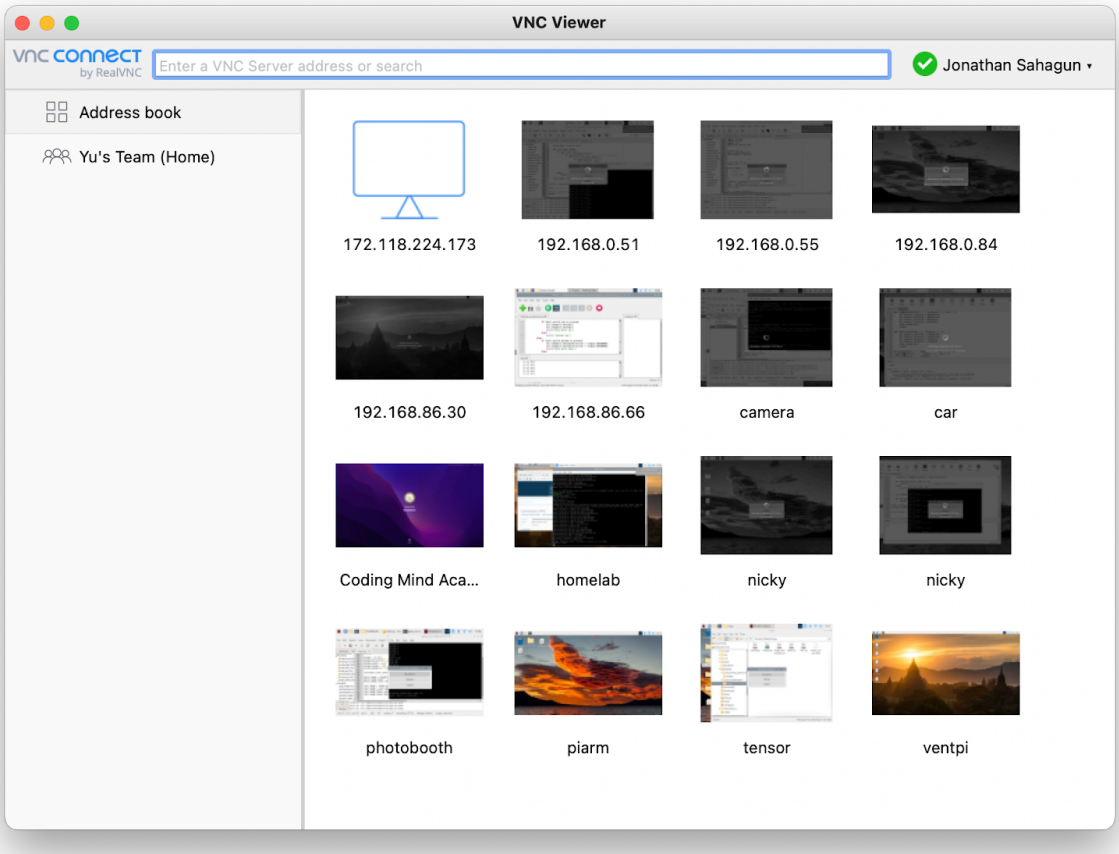


Connecting using the VNC Viewer

VNC Viewer is a program available for Linux, MacOS, and Windows, that will allows you to connect, view and control the raspberry pi remotely, so long as your computer and the Raspberry Pi are on the same network, without have to connect a mouse, keyboard and monitor the the Raspberry Pi.

To install, visit the RealVNC website to download the VNC Viewer, https://www.realvnc.com/en/connect/download/viewer/.

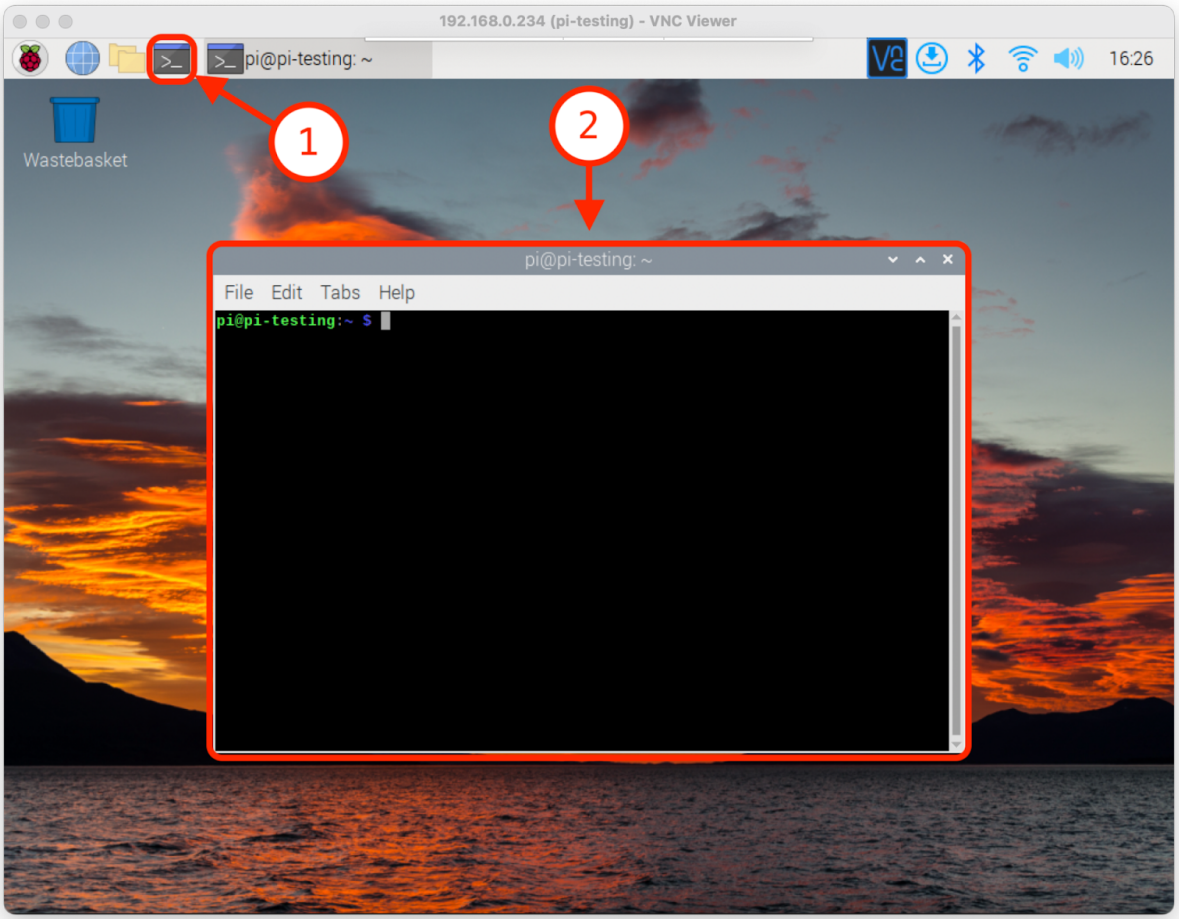
Open the program and follow the instructions. You do not need to create an account.
Once it's installed you see a window like this:



In the text box, enter the hostname of the Raspberry Pi, it should look like “pi-projectname”; it's the hostname that you set up in the “Installing the Raspberry Pi OS” section. If the hostname is not working then enter the IP Address you noted in the “First Time Setup” section.
You’ll be prompted to enter the username and password for the Raspberry Pi, use the username and password from the “Installing the Raspberry Pi OS” section.
On your computer a new window opens showing you the desktop of the Raspberry Pi and allowing you to control it.

Updating the Raspberry Pi

To have the latest version the Raspberry Pi OS update Raspberry PI. Open the terminal, the black rectangle icon in the top left of the screen on the TaskBar.



In the Terminal, enter the following commands:

```
sudo apt update
sudo apt upgrade
```

The Raspberry Pi will now be up to date.

Setting Up the Camera

For this bootstrap we will be using the Raspberry Pi Camera Module 3.

Camera Spec's:

- 11.9 Megapixels
- Max Image Resolution 4608 x 2592 pixels
- Video Modes 2304 × 1296p56, 2304 × 1296p30 HDR, 1536 × 864p120
- Autofocus

For more details visit the following websites:

- https://www.raspberrypi.com/products/camera-module-3
- https://www.raspberrypi.com/documentation/accessories/camera.html

Follow this video tutorial to install the Raspberry Pi Camera on the Raspberry Pi.

Checking the Raspberry Pi Camera

The software to run the Raspberry Pi is already installed on the latest versions of the operating system. To test if everything is working, run the following command in that terminal:

```
libcamera-hello
```

It starts the camera, displays a preview window, and does nothing else for 5 seconds.

Note: you may get the error message:

```
ERROR: *** no cameras available ***
```

If you do double check the camera was installed correctly and try rebooting the Raspberry Pi.

Installing OpenCV

Create a python file with the following code:

```
# Imports
import cv2
from picamera2 import Picamera2
import time


# Initialize the pi camera
pi_camera = Picamera2()
# Convert the color mode to RGB
config = pi_camera.create_preview_configuration(main={"format": "RGB888"})
pi_camera.configure(config)


# Start the pi camera and give it a second to set up
pi_camera.start()
time.sleep(1)


while True:
      # Get a image frame as a numpy array
      image = pi_camera.capture_array()


      # display the image
      cv2.imshow("Video", image)
```

```
        # This waits for 1 ms and if the 'q' key is pressed it breaks the loop
        if cv2.waitKey(1) == ord('q'):
        break

# Close all the windows
cv2.destroyAllWindows()


print('Done!')
```

This code will create a video feed from the camera. It uses the Picamera2 library to take the image and cv2 to display it. The Picamera2 library is preinstalled with the Raspberry Pi OS.

Installing MediaPipe

On the Raspberry Pi install the libraries by running the following commands:

```
python -m pip install mediapipe
```

Then, downgrade protobuf to run on the Raspberry Pi.

```
pip install 'protobuf<=3.20.1' --force-reinstall
```

Installing Base Code

Install the base code from this GitHub repo.

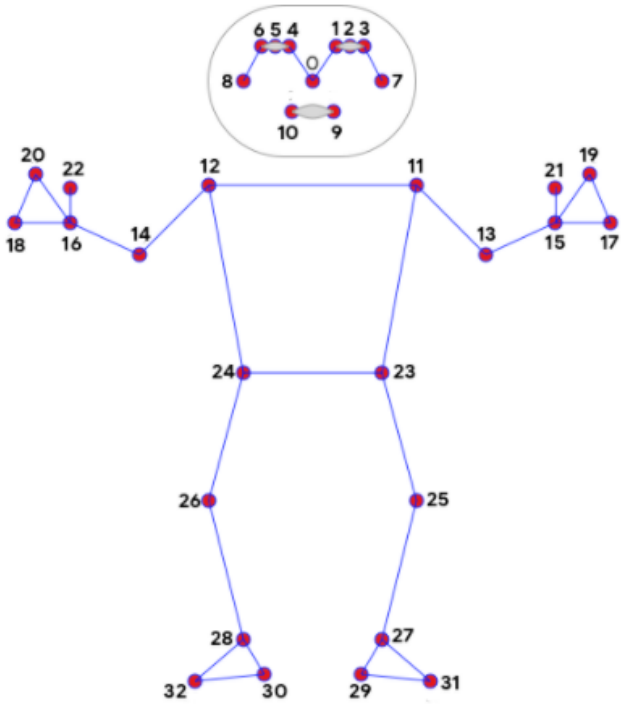
Bootstrap

For this Bootstrap use the code provided in the section above. In this bootstrap we will be using MediaPipe and its pose landmarking task. This machine learning model outputs body pose landmarks in image coordinates and in 3-dimensional world coordinates relative to the image. Here is an example of an landmark:

```
landmark {
      x: 0.4372488260269165
      y: 0.9652527570724487
      z: 0.09486006200313568
      visibility: 0.9793179035186768
}
```

Note the x and y coordinates are percentages relative to the size of the image or video, while z is relative to the distance from where the camera took the image. Visibility is also a percentage value of how visible the landmark is. For example a hand is obscured from the image, MediaPipe will estimate its location but returns a low visibility percent.

The pose landmarking task uses 33 landmarks. Use the image below as a reference for the landmarks:



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Task 1

Modify the draw_pose function to draw on the image. The repository will include a person.png file to test with. When you run the python file it should generate a file named output.py and should look like the image below.

Task 2

Modify the code over to use the Raspberry Pi camera to take an image of a person instead of loading a file. For this use the Picamera2 library to take the image.

Task 3

Modify the code after completing Task 1 and Task 2 show a video feed using the Raspberry Pi camera instead of taking one image and saving it.

BootStrap Assignment Submission

To submit your project, submit the following in the submission field for this bootstrap in your “My Bootstraps” tab on ShareMyWorks.

1) A link to the Github repository containing the code you used in this project
2) A short video demonstrating your device in action; specifically, it must demonstrate the camera being used to record footage of someone moving and the video must show recognition of the body’s key points. A Youtube link of the video would be preferred (you don’t have to make the video public; you can make it unlisted or simply upload it to Google Drive).