**People Counting with the Raspberry Pi**

We’ll experiment using a Raspberry Pi and OpenCV to count objects moving across a detection zone using a webcam or Raspberry Pi camera.

First we need to get the Pi set up. Ideally begin from a fresh installation of the latest version of Raspbian Buster – this can be downloaded from <https://www.raspberrypi.org/downloads/raspbian/> - it’s usually best to select the “full” version, you’ll need at least an 8gb Micro SD card.

Once downloaded you need to “flash” the Raspbian image onto the SD card. The easiest way to do this is to download Balena Etcher from <https://www.balena.io/etcher/>, which once installed will walk you through the process.

After flashing the Raspbian image to the card, pop it in your Pi and connect it up. You’ll need at minimum an HDMI monitor, mouse & keyboard connected before you power up the Pi with a micro-usb power source (Ideally 1.5 amp or above).

The Pi desktop should load up, and the wizard will walk you through changing the default password (important, make a note of it), setting up WiFi and downloading/installing updates. Once the updates have finished it’s a good idea to restart. Once restarted hover your mouse over the WiFi icon and make a note of the Ethernet/WiFi IP Addresses.

In my experience it’s a good idea to set some config options now. Go to Start > Preferences > Raspberry Pi configuration, and change the Hostname from raspberrypi to something more memorable. It’s also good to set a default resolution, usually one of the 720p options will be best.

Changing the hostname helps identify the Pi individually on your network if your IP address should change. Now on the Interfaces tab, enable Camera, SSH and VNC. SSH will allow you to connect and program the Pi’s terminal from another computer using an SSH client like Putty. VNC is an excellent remote desktop utility, allowing you to program the Pi using a full desktop view from another computer (you need to install the PC/MAC client from [https://www.realvnc.com/en/connect/download/viewer/https://www.realvnc.com/en/connect/download/viewer/](https://www.realvnc.com/en/connect/download/viewer/) - it’s available for Android etc too). This can make things MUCH easier as VNC lets you copy & paste text between the host computer & Pi, as well as easily transfer files back & forth.

With the config options saved it’ll prompt to restart, go for it. If you prefer the taskbar at the bottom of the page you can right-click on it and change the location in Panel Settings. Now we need to install some software, mostly from the terminal, which has a handy icon in the taskbar.

**Installing OpenCV**

There are quite a few guides to this online – some say you need a specific version, some say you have to compile it from source, I tried about 5 different ways but the one below worked best.

In the terminal window start by installing dependencies…

sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5-103

sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5

sudo apt-get install libatlas-base-dev

sudo apt-get install libjasper-dev

Then OpenCV itself

pip3 install opencv-python==3.4.6.27

pip3 install imutils

And if you need it the adafruit IO module (for uploading to dashboards)

pip3 install adafruit-io

**Install the script**

The script I used is on GitHub at

<https://github.com/MisterEmm/TT_Count>

Download it to the Pi, it doesn’t really matter which folder. You can then double-click it, which will open it in the Thonny editor, then hit run from there, or assuming you saved it into the /pi/ folder you can type…

Python3 /home/pi/count.py

..in a terminal window.

If you don’t want to send the count data to an Adafruit dashboard you can just comment out lines 25-27 and 162-163.

The script itself is sourced from an excellent Hackster write-up…

<https://www.hackster.io/phfbertoleti/counting-objects-in-movement-using-raspberry-pi-opencv-015ba5>

…which gives much more detailed information about how the code works and what it does, it’s a couple of years old which is why I’ve suggested a different setup method (OpenCV is now easier to install and the code in my version should work with Python 3)

Hopefully I’ve remembered everything!