

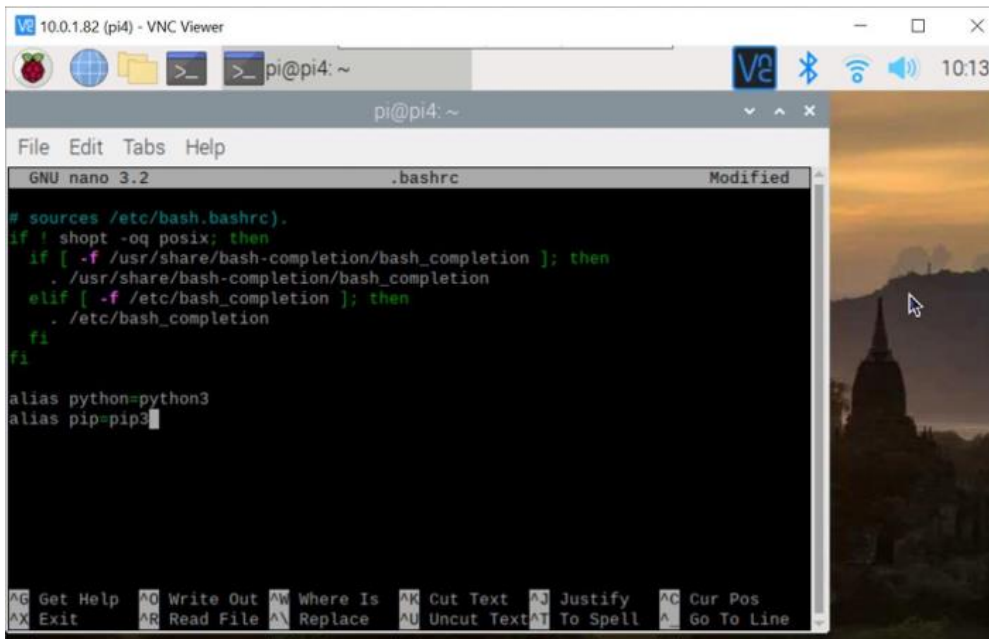
Για την ανίχνευση αντικειμένων (Object Detection) χρησιμοποιείται το πλαίσιο (framework) TensorFlow Lite και ένα προεκπαιδευμένο (pre-trained) μοντέλο Single Shot MultiBox Detector (SSD). Για το σκοπό αυτό, μπορεί να χρησιμοποιηθεί η Raspberry Pi Camera ή μια κοινή USB κάμερα. Ενεργοποιούμε το Raspberry Pi 4 και στο τερματικό του τρέχουμε τις εντολές που ακολουθούν.

Για να χρησιμοποιήσουμε τις εντολές python και pip με την έκδοση Python 3 και όχι την Python 2 πληκτρολογούμε:

```
cd ~  
nano .bashrc
```

Όπως φαίνεται στη εικόνα που ακολουθεί, στο τέλος του κειμένου προσθέτουμε:

```
alias python=python3  
alias pip=pip3
```



Κάνουμε save και exit (ctrl-x). Για να μην χρειαστεί να ανοίξουμε ξανά το τερματικό κάνουμε reload με την εντολή:

```
source .bashrc
```

Μπορούμε να επιβεβαιώσουμε την έκδοση της Python με την εντολή:

```
python --version
```

Θα δημιουργήσουμε ένα εικονικό περιβάλλον (virtual environment) για να διατηρήσουμε τις εγκατεστημένες βιβλιοθήκες και τα πακέτα απομονωμένα. Το TensorFlow Lite μπορεί να απαιτεί πολύ συγκεκριμένες εκδόσεις βιβλιοθηκών, οπότε ένα εικονικό περιβάλλον βοηθά να τις κρατήσουμε ξεχωριστά από το υπόλοιπο σύστημα. Αυτό θα το κάνουμε εγκαθιστώντας το virtualenv και δημιουργώντας ένα περιβάλλον tfllite στο directory του έργου μας.

Για να δημιουργήσουμε το μονοπάτι πληκτρολογούμε:

```
mkdir -p Projects/Python/tflite
```

Για να πάμε μέσα στον φάκελο που δημιουργήσαμε:

```
cd Projects/Python/tflite
```

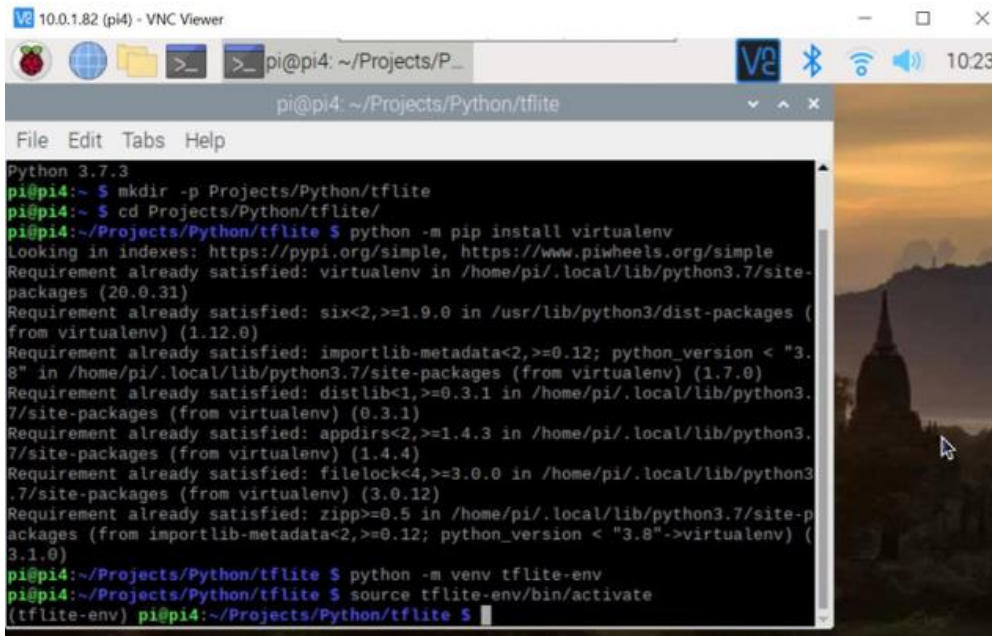
Για την εγκατάσταση του virtualenv και το περιβάλλον tfllite:

```
python -m pip install virtualenv  
python -m venv tflite-env
```

Για να ενεργοποιήσουμε το εικονικό περιβάλλον:

```
source tflite-env/bin/activate
```

Αναμένουμε να δούμε την παρακάτω εικόνα στο τερματικό.



```
pi@pi4: ~/Projects/P...
pi@pi4: ~/Projects/Python/tflite
Python 3.7.3
pi@pi4:~$ mkdir -p Projects/Python/tflite
pi@pi4:~$ cd Projects/Python/tflite/
pi@pi4:~/Projects/Python/tflite$ python -m pip install virtualenv
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: virtualenv in /home/pi/.local/lib/python3.7/site-packages (20.0.31)
Requirement already satisfied: six<2,=>1.9.0 in /usr/lib/python3/dist-packages (from virtualenv) (1.12.0)
Requirement already satisfied: importlib-metadata<2,=>0.12; python_version < "3.8" in /home/pi/.local/lib/python3.7/site-packages (from virtualenv) (1.7.0)
Requirement already satisfied: distlib<1,=>0.3.1 in /home/pi/.local/lib/python3.7/site-packages (from virtualenv) (0.3.1)
Requirement already satisfied: appdirs<2,=>1.4.3 in /home/pi/.local/lib/python3.7/site-packages (from virtualenv) (1.4.4)
Requirement already satisfied: filelock<4,=>3.0.0 in /home/pi/.local/lib/python3.7/site-packages (from virtualenv) (3.0.12)
Requirement already satisfied: zipp>=0.5 in /home/pi/.local/lib/python3.7/site-packages (from importlib-metadata<2,=>0.12; python_version < "3.8"->virtualenv) (3.1.0)
pi@pi4:~/Projects/Python/tflite$ python -m venv tflite-env
pi@pi4:~/Projects/Python/tflite$ source tflite-env/bin/activate
(tflite-env) pi@pi4:~/Projects/Python/tflite$
```

Προετοιμάζουμε το σύστημα για την εγκατάσταση των βιβλιοθηκών με μία ενημέρωση:

```
sudo apt-get update
```

Μέσα στο εικονικό περιβάλλον εγκαθιστούμε τις παρακάτω βιβλιοθήκες:

```
sudo apt -y install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev
libx264-dev qt4-dev-tools libatlas-base-dev libhdf5-103
```

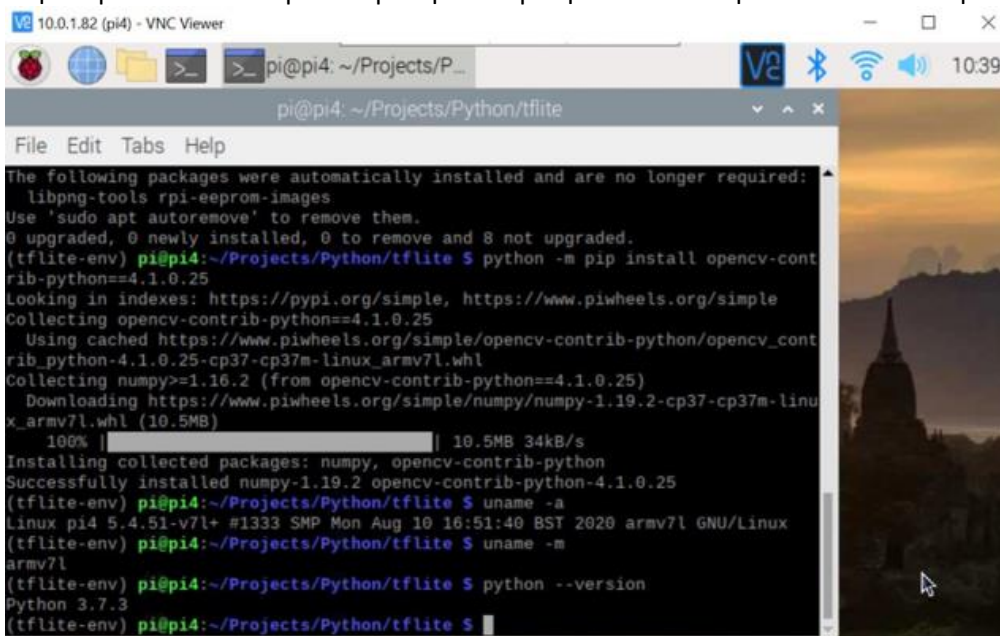
Όταν ολοκληρωθεί η εγκατάσταση των παραπάνω θα χρησιμοποιήσουμε την εντολή pip για να εγκαταστήσουμε συγκεκριμένη έκδοση του OpenCV:

```
python -m pip install opencv-contrib-python==4.1.0.25
```

Αν και το Raspberry Pi 4 διαθέτει επεξεργαστή 64-bit, το προτεινόμενο λειτουργικό σύστημα Raspberry OS είναι 32-bit (τουλάχιστον μέχρι και τα τέλη του 2021 που γράφεται αυτός ο οδηγός). Μπορούμε να το επιβεβαιώσουμε αλλά και να δούμε την έκδοση της Python με τις εντολές:

```
uname -m
python --version
```

Στην παρακάτω εικόνα βλέπουμε την απόκριση armv7l που δηλώνει 32-bit και την έκδοση 3.7.3 για την Python.



```
pi@pi4: ~/Projects/P...
pi@pi4: ~/Projects/Python/tflite
The following packages were automatically installed and are no longer required:
  libpng-tools rpi-eeprom-images
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
(tflite-env) pi@pi4:~/Projects/Python/tflite$ python -m pip install opencv-contrib-python==4.1.0.25
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-contrib-python==4.1.0.25
Using cached https://www.piwheels.org/simple/opencv-contrib-python/opencv_contrib_python-4.1.0.25-cp37-cp37m-linux_armv7l.whl
Collecting numpy>=1.16.2 (from opencv-contrib-python==4.1.0.25)
Downloading https://www.piwheels.org/simple/numpy/numpy-1.19.2-cp37-cp37m-linux_armv7l.whl (10.5MB)
100% |#####| 10.5MB 34kB/s
Installing collected packages: numpy, opencv-contrib-python
Successfully installed numpy-1.19.2 opencv-contrib-python-4.1.0.25
(tflite-env) pi@pi4:~/Projects/Python/tflite$ uname -a
Linux pi4 5.4.51-v7l+ #1333 SMP Mon Aug 10 16:51:40 BST 2020 armv7l GNU/Linux
(tflite-env) pi@pi4:~/Projects/Python/tflite$ uname -m
armv7l
(tflite-env) pi@pi4:~/Projects/Python/tflite$ python --version
Python 3.7.3
(tflite-env) pi@pi4:~/Projects/Python/tflite$
```

Εγκαθιστούμε το TensorFlow Lite κατεβάζοντας το κατάλληλο αρχείο που στη συγκεκριμένη περίπτωση αντιστοιχεί στην έκδοση 32-bit:

```
pip3 install https://dl.google.com/coral/python/tflite_runtime-2.1.0.post1-cp37-cp37m-linux_armv7l.whl
```

Απενεργοποιούμε το εικονικό περιβάλλον με την εντολή:

```
deactivate
```

Θα χρησιμοποιήσουμε ένα pre-trained μοντέλο αξιοποιώντας το Common Objects in Context (COCO) dataset. Περιλαμβάνει μια μεγάλη συλλογή εικόνων που έχουν επισημανθεί (tagged και labeled) από ερευνητές της Microsoft, και διάφορων πανεπιστημίων. Περιέχει πάνω από 200.000 εικόνες και περίπου 90 κατηγορίες αντικειμένων. Τα μοντέλα ανίχνευσης αντικειμένων ή ταξινόμησης αντικειμένων μπορούν να εκπαιδευτούν στο σύνολο δεδομένων COCO για να μας δώσουν μια αφετηρία για την αναγνώριση αντικειμένων καθημερινής χρήσης, όπως άτομα, αυτοκίνητα, ποδήλατα, σκύλοι, γάτες κ.ά. Θα χρησιμοποιήσουμε το μοντέλο MobileNet V1 που έχει εκπαιδευτεί στο σύνολο δεδομένων COCO ως μοντέλο ανίχνευσης αντικειμένων.

Δημιουργούμε το μονοπάτι:

```
mkdir -p ~/Projects/Python/tflite/object_detection/coco_ssd_mobilenet_v1
```

Πάμε στον φάκελο:

```
cd ~/Projects/Python/tflite/object_detection
```

Κατεβάζουμε το αρχείο coco\_ssd\_mobilenet\_v1\_1.0\_quant\_2018\_06\_29.zip που βρίσκεται στη διεύθυνση <https://github.com/robotakia-os/ellak2021/tree/main/pre-trained%20model> και επιβεβαιώνουμε ότι βρίσκεται στον φάκελο των λήψεων.

Μετακινούμε το αρχείο και το αποσυμπιέζουμε με τις εντολές:

```
mv ~/Downloads/coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip .
unzip coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip -d coco_ssd_mobilenet_v1
```

Πληκτρολογούμε την επόμενη εντολή για να πάμε έναν φάκελο παραπάνω:

```
cd -
```

Βρισκόμαστε και πάλι στη θέση /Projects/Python/tflite και ενεργοποιούμε και πάλι το εικονικό περιβάλλον:

```
source tflite-env/bin/activate
```

Θα εγκαταστήσουμε βιβλιοθήκες που σχετίζονται με τα pins του Raspberry:

```
pip install Rpi.GPIO
pip install gpiozero
```

Επιπλέον θα εγκαταστήσουμε και τις βιβλιοθήκες που σχετίζονται με την αναπαραγωγή της μουσικής:

```
pip install pygame
sudo apt-get install git curl libsdl2-mixer-2.0-0
```

Πληκτρολογούμε την παρακάτω εντολή και κλείνουμε το τερματικό:

```
deactivate
```

Ο κώδικας που θα τρέχει στο Raspberry Pi 4 είναι το αρχείο pi4\_code.py. Αφού κατεβάσουμε το συγκεκριμένο αρχείο με τον κώδικα από το GitHub, θα πρέπει να το βάλουμε στον φάκελο /Projects/Python/tflite/object\_detection . Τελικά για να ενεργοποιήσουμε την αναγνώριση αντικειμένων από το Raspberry Pi 4 πληκτρολογούμε στο τερματικό 4 εντολές.

Αρχικά πάμε στον φάκελο:

```
cd ~/Projects/Python/tflite
```

Ενεργοποιούμε το εικονικό περιβάλλον:

```
source tflite-env/bin/activate
```

Πάμε στον φάκελο:

```
cd object_detection
```

Τρέχουμε το πρόγραμμα με την εντολή:

```
python pi4_code.py --modeldir=coco_ssd_mobilenet_v1
```