Step-by-step: Object detection on a Fresh/New Raspberry pi 4 using TensorFlow Lite

I recently bought a Raspberry pi 4(4GB) which is an amazing board with diverse set of capabilities and features available. It's kind of a mini-computer. I wanted to leverage Raspberry pi to detect objects in a live cam. Tensorflow Lite has made is very simple to perform this.

This is a step-by-step tutorial right from opening a new Raspberry pi 4 till a working prototype.

Tutorial is divided into 4 stages

1. Hardware and Software
2. Setup new Raspberry pi 4
3. Setup Camera
4. Deploy Image Classification Model

Note:

- The Instructions are preformed on Windows 10 Machine
- Commands are in *italics* and needs to be executed on terminal

Hardware and Software

Hardware

- Raspberry pi 4 (1GB, 2GB or 4GB)

- Micro SD Card >= 8GB
- Micros SD card Reader OR Adapter
  - Reader

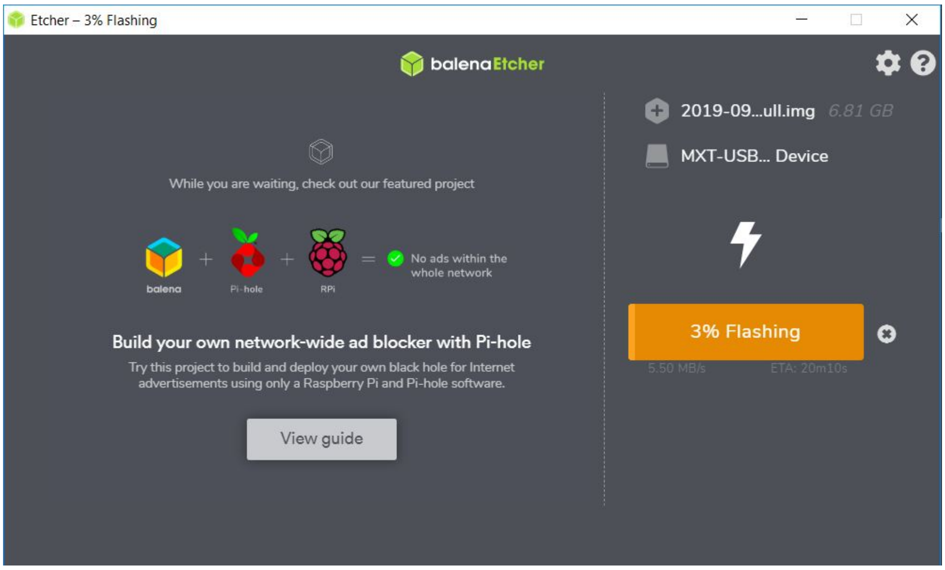
    - Adapter

Software

- Raspian OS image
- SD Association's Formatting Tool
- OS Image burning software - Balena
- Remote access to Raspberry Pi - MobaXterm

Setup new Raspberry pi 4

- **Non-Headless setup:** Follow Official setup page in case Monitor, Keyboard and mouse is present Setup Raspberry pi - Raspberry website
- **Headless Setup:** This is followed when external Monitor, Keyboard and Mouse is not present. A PC/Laptop is present.
    - **References**:
        - Headless Raspberry pi ssh wifi setup
        - Raspberry Pi Headless Setup With WiFi and SSH Enabled
    - **PC/Laptop**
        - Raspian OS: Download from link
            - Raspbian Buster with desktop and recommended software
                - Size(img file) ~ 6 GB
                - Full featured, supports Remote desktop, VNC
                - Used in this tutorial
            - Raspbian Buster Lite
                - Size(img file) ~ 400 MB
                - Minimal features, no Remote desktop or VNC
        - Micro SD Card
            - Attach Micro SD card to Its reader/adapter and plug it to PC/Laptop
            - For old Micro SD card, take backup and format it using SD Association's Formatting Tool
            - Micro SD card will appear as a USB drive in the PC.
            - Note down the letter of drive(E:). Double click to open the drive
        - Burn Raspian image to Micro SD card
            - Download Balena(Image burning software)
            - Launch application:
                - Select raspian os image (2019-09-26-raspbian-buster-full.img)
                - Select MicroSD drive(E:)
                - Click Flash
            - In-Progress

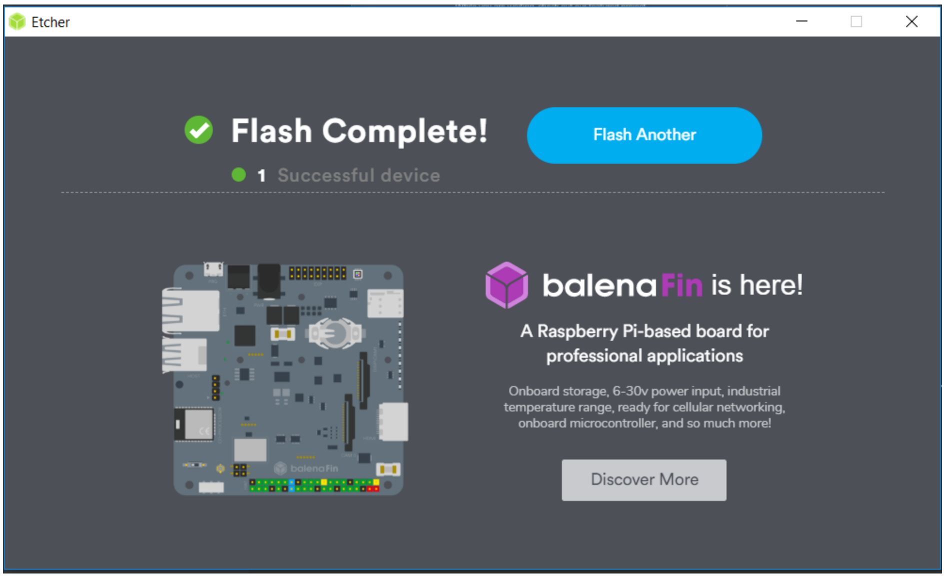


While you are waiting, check out our featured project

Build your own network-wide ad blocker with Pi-hole

Try this project to build and deploy your own black hole for Internet advertisements using only a Raspberry Pi and Pi-hole software.

    - Completed



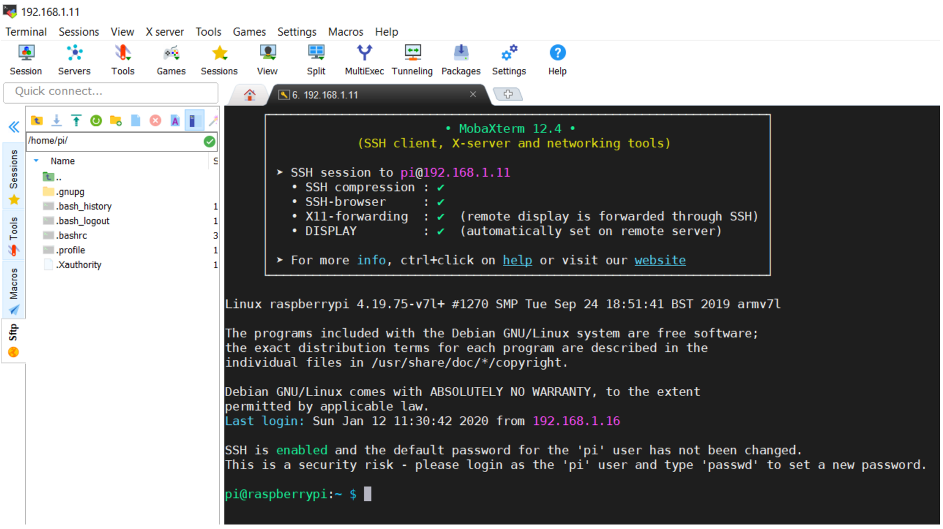
A Raspberry Pi-based board for professional applications

Onboard storage, 6-30v power input, industrial temperature range, ready for cellular networking, onboard microcontroller, and so much more!

        - **Failure**: In case there is a failure due to checksum error, please follow steps in below link
        - [How to prevent creation of “System Volume Information” folder in Windows 10 for USB flash drives?]()
            - [Checksums do not match]()
    - Re-insert the MicroSD card
        - A drive might have been created with name as 'boot'
    - Double click to navigate to this drive
- Enable SSH for remote login
    - Create an empty file by name 'ssh', and put it in the root folder of SD card. No extension to file
    - Enable Wifi
        - Create a file with name 'wpa_supplicant.conf' and place it in the boot root folder
        - Give proper country, ssid and password

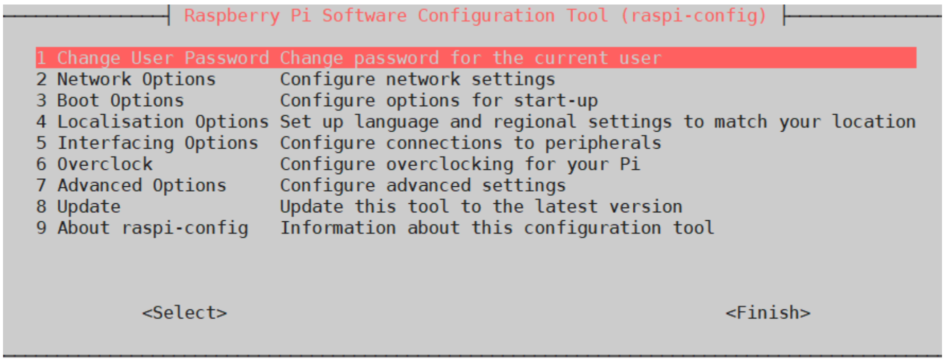
```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
ap_scan=1
fast_reauth=1
country=IN

network={
    ssid="NETWORK-NAME"
    psk="NETWORK-PASSWORD"
    id_str="0"
    priority=100
}
```

- **Raspberry Pi**
  - Boot Raspbery Pi with SD card
    - Eject MicroSD card from PC
    - Connect Raspberry Pi to Router through LAN cable
    - Plug it into Raspbery Pi and wait for 2 minutes
  - Router
    - Login to router such as 192.168.1.1
    - In the device list, a device containing 'raspberry pi' should appear
    - Get the IP of the this device such as 192.168.1.11
  - Login Remotely over Wi-Fi
    - Download and install MobaXterm
    - Click on session and type pi IP(192.168.1.11) in Remote Host under Basic Settings. Click ok
    - Enter username: pi and password: raspberry



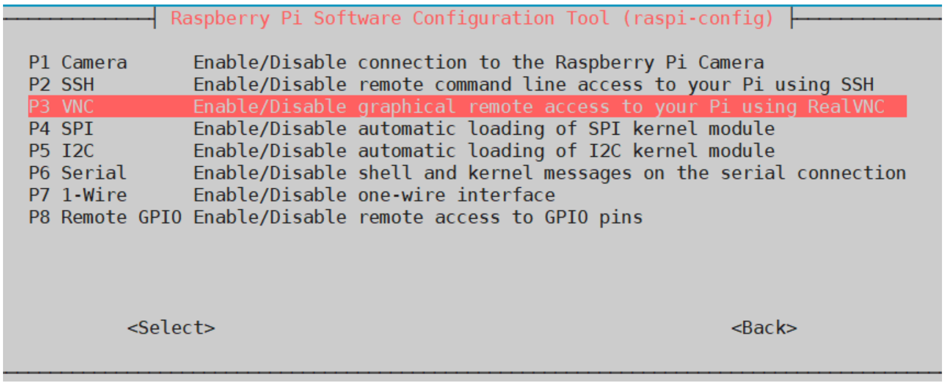
```
                    • MobaXterm 12.4 •
        (SSH client, X-server and networking tools)

 ➤ SSH session to pi@192.168.1.11
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Linux raspberrypi 4.19.75-v7l+ #1270 SMP Tue Sep 24 18:51:41 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jan 12 11:30:42 2020 from 192.168.1.16

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $
```

  - Remote desktop vnc
    - Enter *sudo raspi-config*



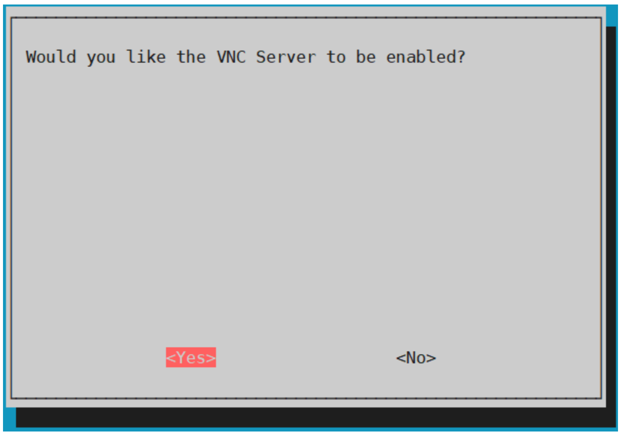
```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the current user
2 Network Options      Configure network settings
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool

           <Select>                                        <Finish>
```

- Select Option 5: Interfacing Options



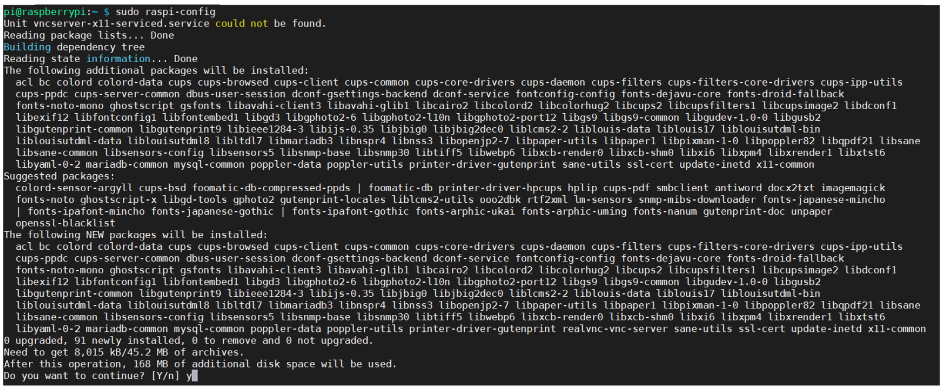
```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

          <Select>                                        <Back>
```

- Select P3-VNC and Press Yes



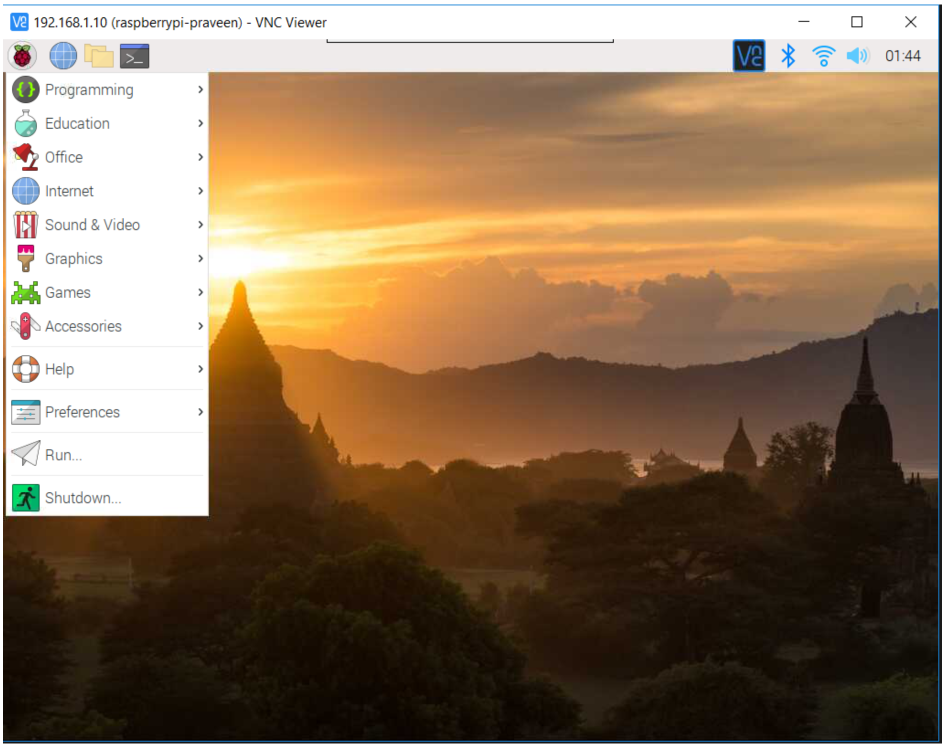
```
Would you like the VNC Server to be enabled?

              <Yes>                 <No>
```

- Pres Y to continue

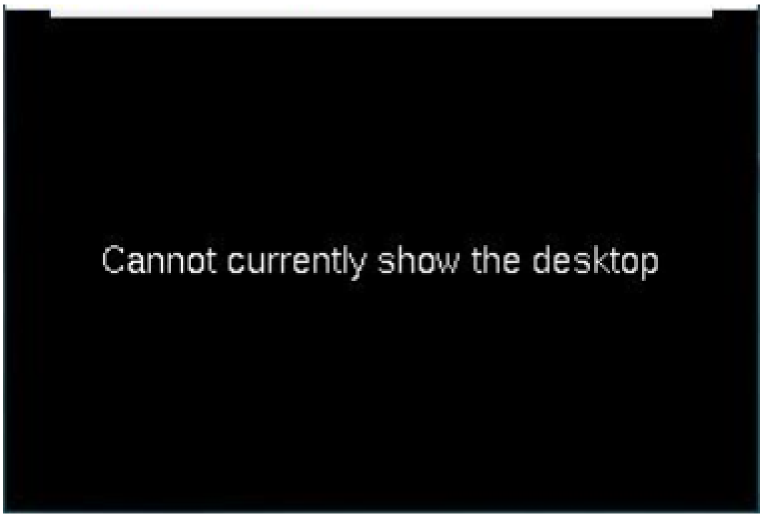


```
pi@raspberrypi:~ $ sudo raspi-config
Unit vncserver-x11-serviced.service could not be found.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  acl bc colord colord-data cups cups-browsed cups-client cups-common cups-core-drivers cups-daemon cups-filters cups-filters-core-drivers cups-ipp-utils
  cups-ppdc cups-server-common dbus-user-session dconf-gsettings-backend dconf-service fontconfig-config fonts-dejavu-core fonts-droid-fallback
  fonts-noto-mono ghostscript gsfonts libavahi-client3 libavahi-glib1 libcairo2 libcolord2 libcolorhug2 libcups2 libcupsfilters1 libcupsimage2 libdconf1
  libexif12 libfontconfig1 libfontembed1 libgd3 libgphoto2-6 libgphoto2-l10n libgphoto2-port12 libgs9 libgs9-common libgudev-1.0-0 libgusb2
  libgutenprint-common libgutenprint9 libieee1284-3 libijs-0.35 libjbig0 libjbig2dec0 liblcms2-2 liblouis-data liblouis17 liblouisutdml-bin
  liblouisutdml-data liblouisutdml8 libltdl7 libmariadb3 libnspr4 libnss3 libopenjp2-7 libpaper-utils libpaper1 libpixman-1-0 libpoppler82 libqpdf21 libsane
  libsane-common libsensors-config libsensors5 libsnmp-base libsnmp30 libtiff5 libwebp6 libxcb-render0 libxcb-shm0 libxi6 libxpm4 libxrender1 libxtst6
  libyaml-0-2 mariadb-common mysql-common poppler-data poppler-utils printer-driver-gutenprint sane-utils ssl-cert update-inetd x11-common
Suggested packages:
  colord-sensor-argyll cups-bsd foomatic-db-compressed-ppds | foomatic-db printer-driver-hpcups hplip cups-pdf smbclient antiword docx2txt imagemagick
  fonts-noto ghostscript-x libgd-tools gphoto2 gutenprint-locales liblcms2-utils ooo2dbk rtf2xml lm-sensors snmp-mibs-downloader fonts-japanese-mincho
  | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum gutenprint-doc unpaper
  openssl-blacklist
The following NEW packages will be installed:
  acl bc colord colord-data cups cups-browsed cups-client cups-common cups-core-drivers cups-daemon cups-filters cups-filters-core-drivers cups-ipp-utils
  cups-ppdc cups-server-common dbus-user-session dconf-gsettings-backend dconf-service fontconfig-config fonts-dejavu-core fonts-droid-fallback
  fonts-noto-mono ghostscript gsfonts libavahi-client3 libavahi-glib1 libcairo2 libcolord2 libcolorhug2 libcups2 libcupsfilters1 libcupsimage2 libdconf1
  libexif12 libfontconfig1 libfontembed1 libgd3 libgphoto2-6 libgphoto2-l10n libgphoto2-port12 libgs9 libgs9-common libgudev-1.0-0 libgusb2
  libgutenprint-common libgutenprint9 libieee1284-3 libijs-0.35 libjbig0 libjbig2dec0 liblcms2-2 liblouis-data liblouis17 liblouisutdml-bin
  liblouisutdml-data liblouisutdml8 libltdl7 libmariadb3 libnspr4 libnss3 libopenjp2-7 libpaper-utils libpaper1 libpixman-1-0 libpoppler82 libqpdf21 libsane
  libsane-common libsensors-config libsensors5 libsnmp-base libsnmp30 libtiff5 libwebp6 libxcb-render0 libxcb-shm0 libxi6 libxpm4 libxrender1 libxtst6
  libyaml-0-2 mariadb-common mysql-common poppler-data poppler-utils printer-driver-gutenprint realvnc-vnc-server sane-utils ssl-cert update-inetd x11-common
0 upgraded, 91 newly installed, 0 to remove and 0 not upgraded.
Need to get 8,015 kB/45.2 MB of archives.
After this operation, 168 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

- Enter *sudo reboot now* to reboot in order the settings to take into effect
- Now start VNC and enter the IP of raspberry pi. It should display the Raspberry pi desktop.



  - In case black screen with message 'Cannot currently show the desktop' is displayed, follow below steps

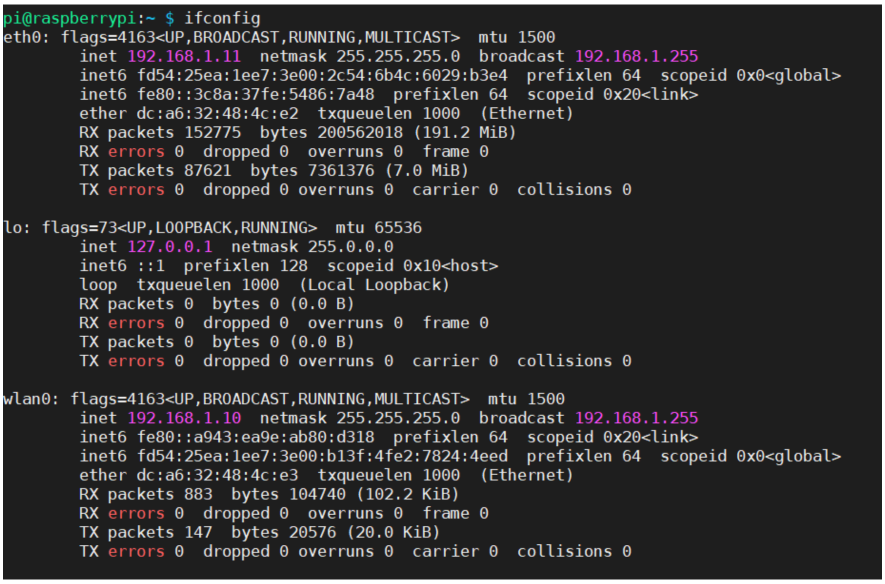


Cannot currently show the desktop

    - Connect raspberry pi through SSH
    - Enter *sudo raspi-config*
    - Select 7 - Advanced Operations
    - Select 5 - Resolution
    - Select last option and press OK
    - Restart Raspberry pi
    - Start VNC and desktop will be displayed
- **[Optional] Installing On-screen keyboard**
  - [Matchbox Keyboard - Raspberry Pi Touchscreen Keyboard]()
    - *sudo apt-get install matchbox-keyboard*
    - Enter *sudo matchbox-keyboard* to open keyboard
    - Launch keyboard through double click
      - Create a file on the Desktop 'keyboard.sh'
      - Write below statement in the file

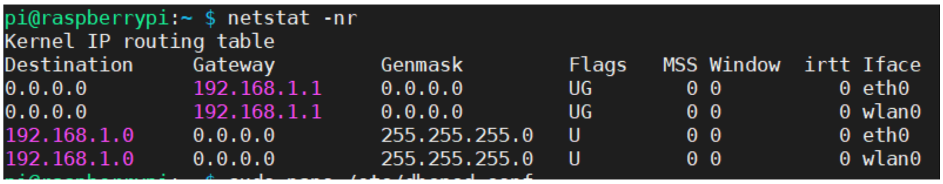
```
#!/bin/bash
matchbox-keyboard
```

    - Open terminal and enter below commands
        - *cd Desktop*
            - *chmod +x keyboard.sh*
            - Double click to launch keyboard
- **[Optional] Setting Static IP**
    - Reference
        - [How to Set Static IP for Raspberry Pi in Raspbian Jessie]
        - [2016: Assign a Static IP Address to Raspberry Pi]
        - [Raspberry Pi - Tutorial 12 - Networking - How to Configure a Static IP Address & Setup Wifi]
    - Execute *ifconfig* and note down the IP address (inet)



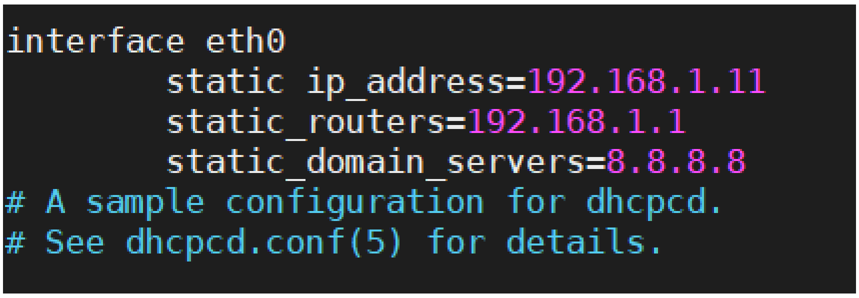
```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.11  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fd54:25ea:1ee7:3e00:2c54:6b4c:6029:b3e4  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::3c8a:37fe:5486:7a48  prefixlen 64  scopeid 0x20<link>
        ether dc:a6:32:48:4c:e2  txqueuelen 1000  (Ethernet)
        RX packets 152775  bytes 200562018 (191.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 87621  bytes 7361376 (7.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a943:ea9e:ab80:d318  prefixlen 64  scopeid 0x20<link>
        inet6 fd54:25ea:1ee7:3e00:b13f:4fe2:7824:4eed  prefixlen 64  scopeid 0x0<global>
        ether dc:a6:32:48:4c:e3  txqueuelen 1000  (Ethernet)
        RX packets 883  bytes 104740 (102.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 147  bytes 20576 (20.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

    - Execute *netstat -nr* . Note down Gateway IP : 192.168.1.1, it will be used to set static_routers



```
pi@raspberrypi:~ $ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 eth0
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 wlan0
```

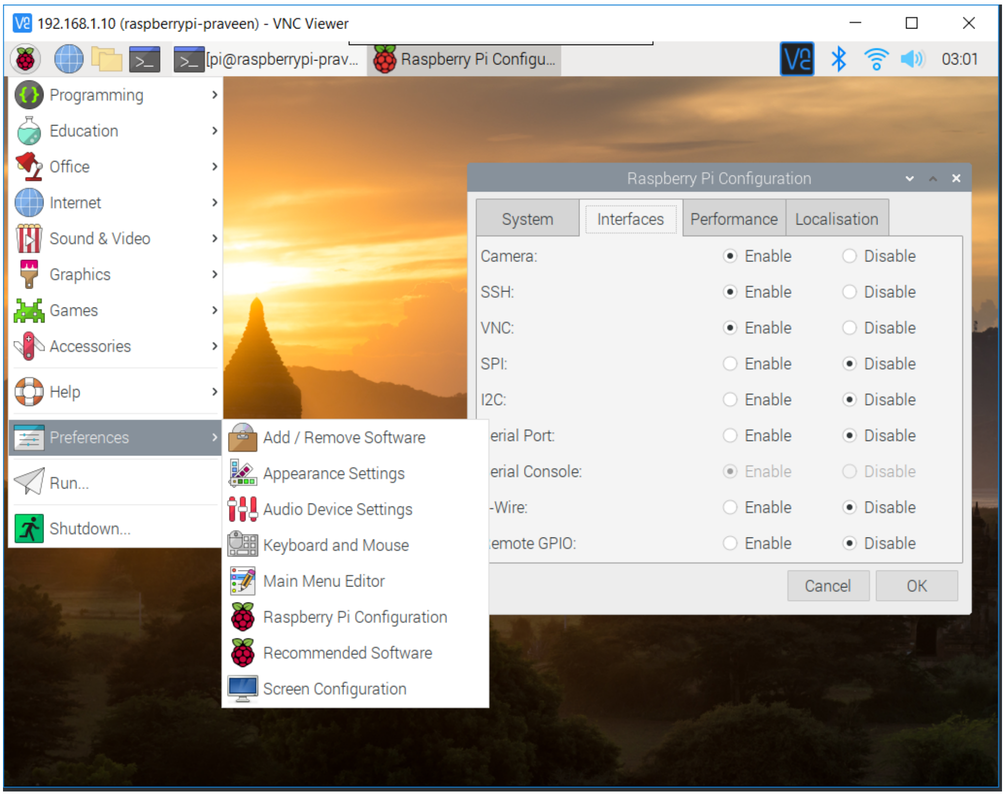
    - Execute *sudo nano /etc/dhcpcd.conf*. File will be opened in edit mode. Add top four lines at the beginning of file. Ctrl+X and Y to save the file.



```
interface eth0
        static ip_address=192.168.1.11
        static_routers=192.168.1.1
        static_domain_servers=8.8.8.8
# A sample configuration for dhcpcd.
# See dhcpcd.conf(5) for details.
```

    - Execute *sudo reboot* to reboot Raspberry Pi
    - Execute *ifconfig* and check the inet address, it must be the one given above.

    - References
        - https://magpi.raspberrypi.org/articles/set-up-raspberry-pi-4
        - https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up
        - https://www.instructables.com/id/How-to-Setup-a-Raspberry-Pi/
- **Setup Camera**
    - Reference: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera
    - Enable Camera
        - Open Raspberry Pi Configuration
        - Go to 'Interfaces' tab and select Camera 'Enabled'



192.168.1.10 (raspberrypi-praveen) - VNC Viewer

Raspberry Pi Configuration

System | Interfaces | Performance | Localisation

Camera: ● Enable ○ Disable
SSH: ● Enable ○ Disable
VNC: ● Enable ○ Disable
SPI: ○ Enable ● Disable
I2C: ○ Enable ● Disable
erial Port: ○ Enable ● Disable
erial Console: ● Enable ○ Disable
-Wire: ○ Enable ● Disable
emote GPIO: ○ Enable ● Disable

Cancel | OK

        - Reboot *sudo reboot now*
        - Attach camera to Raspberry pi
        - Enter *raspistill -o Desktop/image.jpg* to take an image from raspberry pi. It gets saved on Desktop
        - Check image
- **Deploy image classification model**
    - Reference: Part 2 - How to Run TensorFlow Lite Object Detection Models on the Raspberry Pi (with Optional Coral USB Accelerator)
    - TensorFlow Lite
        - Followed step by step instructions in the tutorial and it worked without any issue. Summarized steps/commands.
            1. *sudo apt-get update*
            2. *sudo apt-get dist-upgrade* (Time consuming)
            3. Download code from github
                1. *git clone https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi.git*

   2. Move code to tflite1 directory *mv TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi tflite1*
   3. Change directory *cd tflite1*
4. Create Virtual Environment for isolation
   1. Install virtual environment *sudo pip3 install virtualenv*
   2. Create virtual environment *source tflite1-env/bin/activate*
5. Install dependencies *bash get_pi_requirements.sh*
6. Google's sample TFLite model
   1. Download *wget https://storage.googleapis.com/download.tensorflow.org/models/tflite/coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip*
   2. Unzip *unzip coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip -d Sample_TFLite_model*
7. Run model to start webcam and start making predictions
   1. *python3 TFLite_detection_webcam.py --modeldir=Sample_TFLite_model*
8. Awesome!!!

References

- How to Set Static IP for Raspberry Pi in Raspbian Jessie
- 2016: Assign a Static IP Address to Raspberry Pi
- Raspberry Pi - Tutorial 12 - Networking - How to Configure a Static IP Address & Setup Wifi
- https://magpi.raspberrypi.org/articles/set-up-raspberry-pi-4
- https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up
- https://www.instructables.com/id/How-to-Setup-a-Raspberry-Pi/
- https://projects.raspberrypi.org/en/projects/getting-started-with-picamera
- https://www.youtube.com/watch?v=aimSGOAUI8Y
- https://www.techrepublic.com/article/raspberry-pi-and-machine-learning-how-to-get-started/
- https://www.youtube.com/watch?v=npZ-8Nj1YwY
- https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/Raspberry_Pi_Guide.md#step-1e-run-the-tensorflow-lite-model
- https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/Raspberry_Pi_Guide.md
- https://www.pyimagesearch.com/2017/10/02/deep-learning-on-the-raspberry-pi-with-opencv/
- https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/
- Matchbox Keyboard - Raspberry Pi Touchscreen Keyboard