TensorFlow Guide

Learning Outcomes

By the end of this ride, you'll be able to:

- Spin up a dedicated TensorFlow workspace like a boss—no stray files messing with your vibe.
- Compile and install Python 3.11 from source, flexing those optimization muscles.
- Create and toggle virtual environments, isolating your experiment like a mad scientist.
- Fetch and unpack a pretrained SSD-MobileNet model + COCO label map, so you're not reinventing the wheel.
- Wire up Flask, OpenCV, NumPy, and TensorFlow into a live webcam-powered object detector.

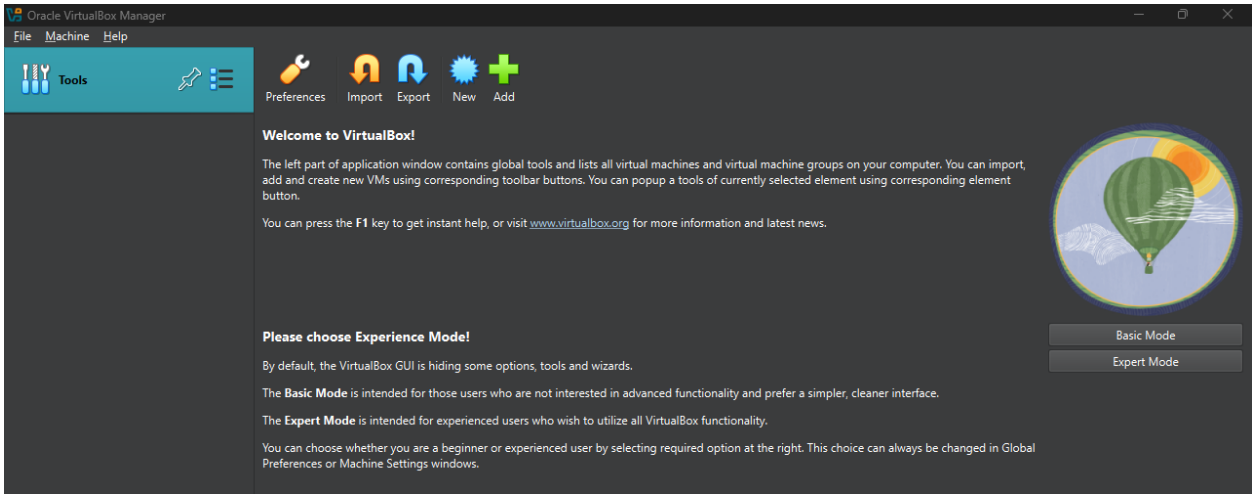
Prerequisites

- An Ubuntu VM goto https://lchs.mm.fcix.net/ubuntu-releases/25.04/ubuntu-25.04-desktop-amd64.iso
- A vm with access to your webcam (ideally a usb webcam or built in)
- At least 200mb of free space
- VM specs (ideally)
    - **CPU (x86)**
        - **Min:** 4 vCPUs (2.0 GHz+)
        - **Sweet spot:** 8 vCPUs or more
    - **RAM**
        - **Min:** 8 GB
        - **Reco:** 16 GB+
    - **Other**
        - Git and Curl Installed
    - **Display**
        - 320x320 minimum

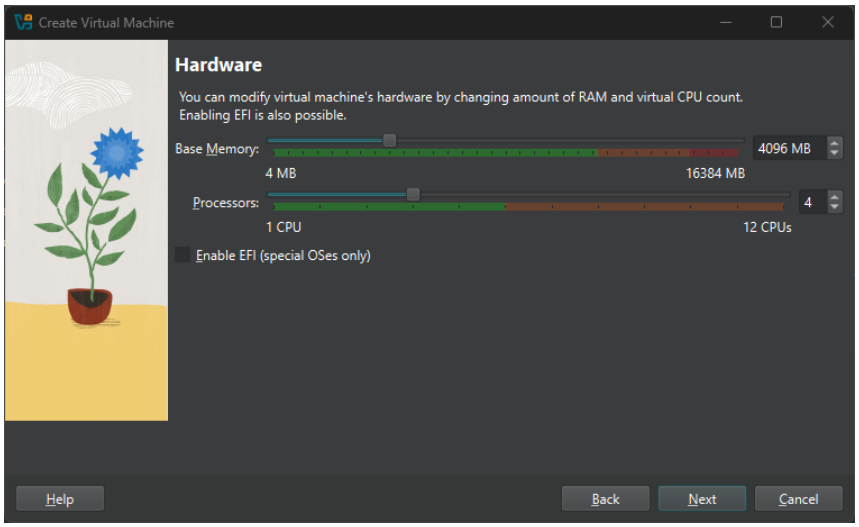
Setting up the Ubuntu VM

1. Go to https://lchs.mm.fcix.net/ubuntu-releases/25.04/ubuntu-25.04-desktop-amd64.iso to download the iso file
2. Go to https://www.virtualbox.org/wiki/Downloads, Install and setup Virtualbox.

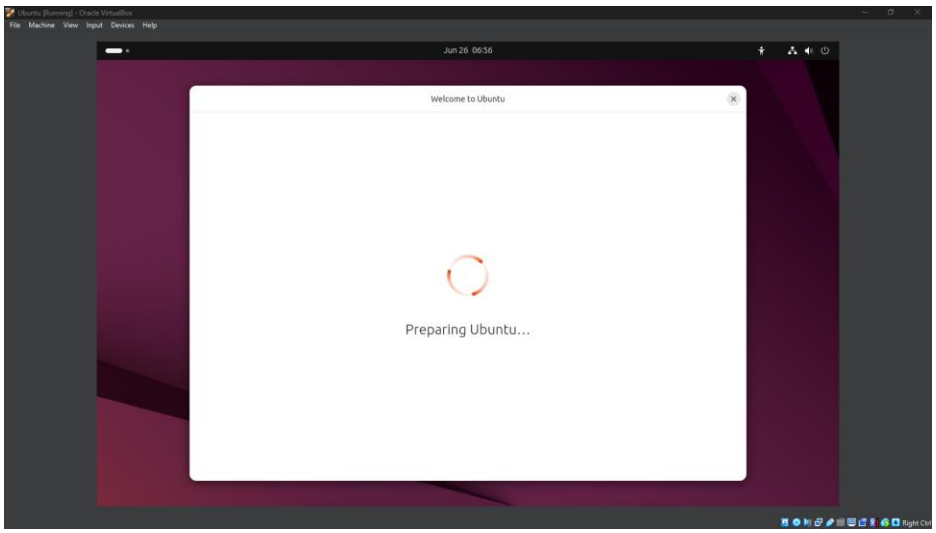
You should get a result like this:



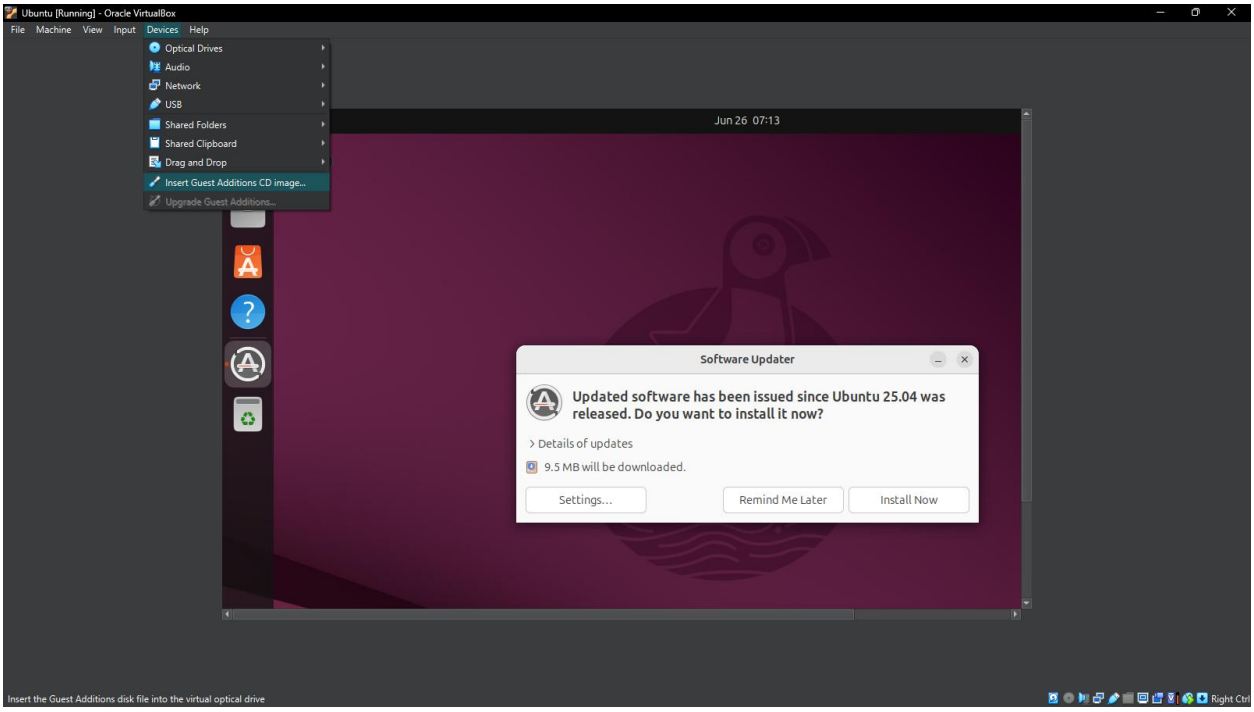
3. Following along, press the new button and follow instructions to setup the vm
   Note: allow about 4 GB of RAM and 4 CPUs to run



4. VM should start running. Press Try and install Ubuntu in grub and follow along the unintended GUI installation

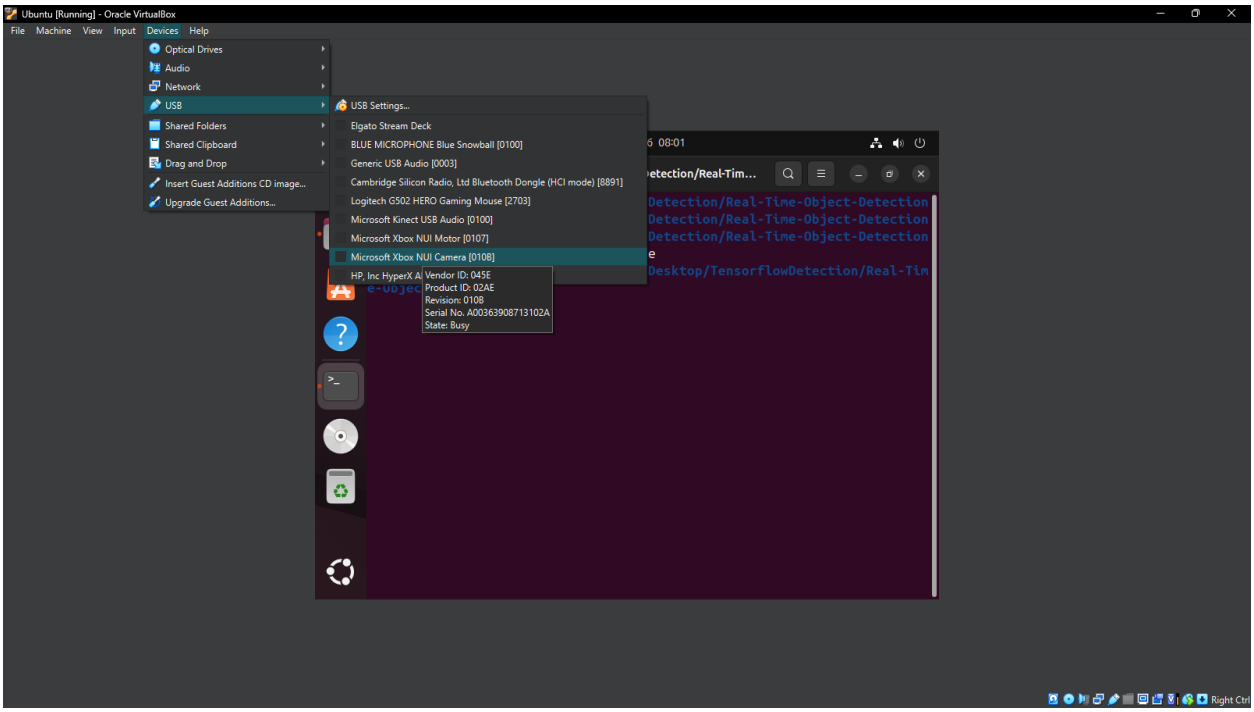


5. Optional: Once the vm reboots, feel free to install the Guest Additions for a smoother experience (& shared clipboard)



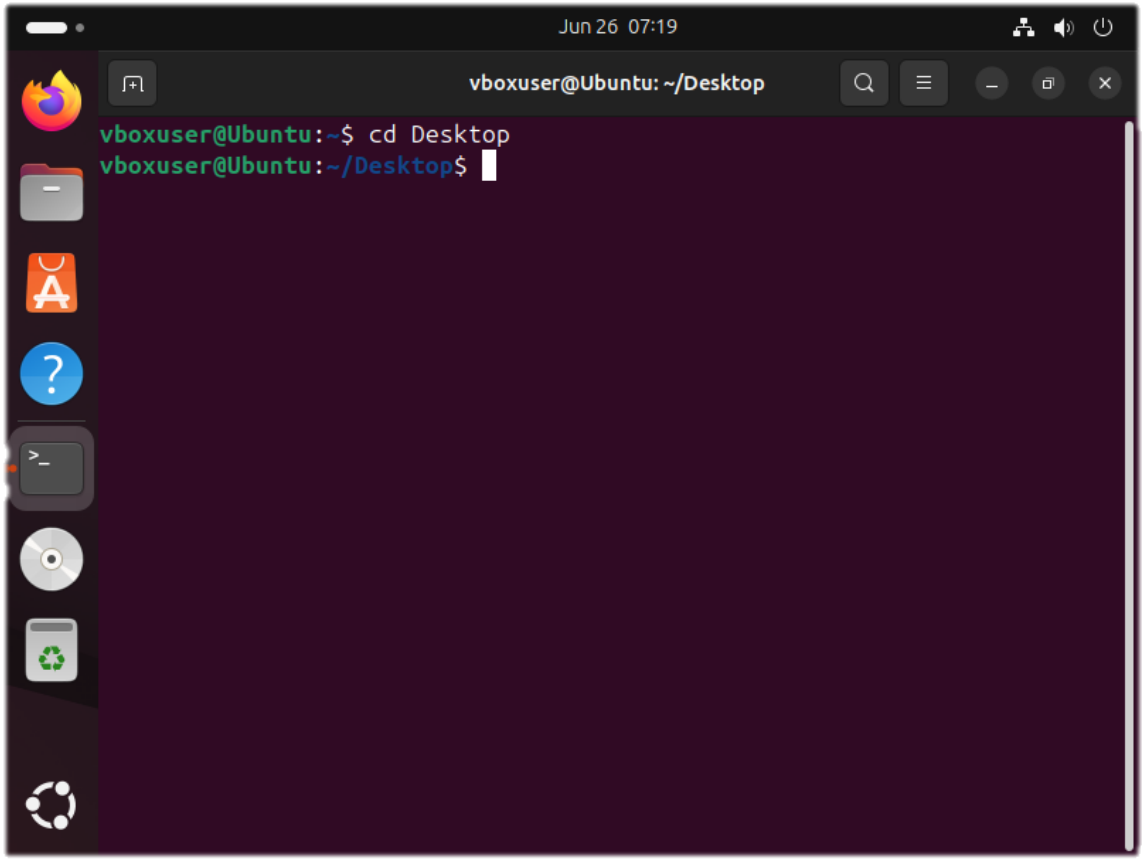
Enable shared clipboard at Devices > Shared Clipboard > Bidirectional

6. Now enable usb webcam passthrough.



Setting up Folders

Now on the desktop, open up terminal (ctrl+alt+t), and enter `cd Desktop` to the terminal to get to the desktop.



Now run the following to setup git and curl, for later.

```
sudo apt install git

sudo apt install curl
```

Now, to setup the file system for this project.

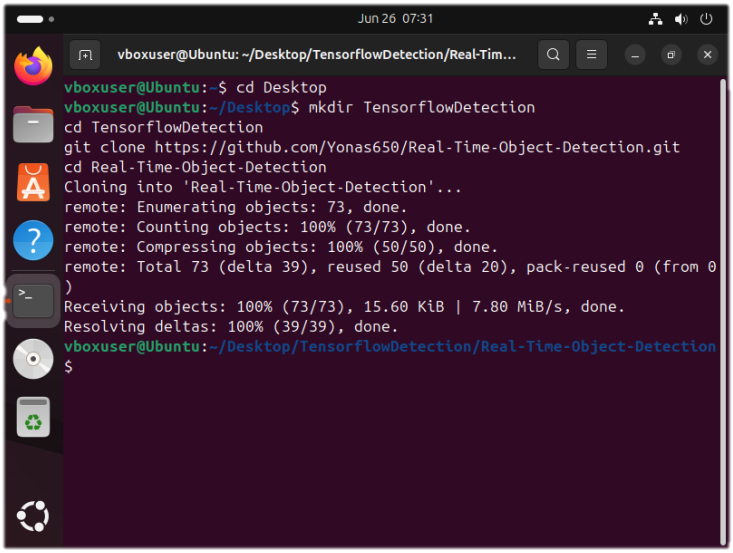
```
mkdir TensorflowDetection

cd TensorflowDetection

git clone https://github.com/Yonas650/Real-Time-Object-Detection.git

cd Real-Time-Object-Detection
```

Why we do this:

- **mkdir TensorflowDetection**: sets up your own lil' sandbox—keeps your files from minglin' with your desktop.
- **cd TensorflowDetection**: jump into that sandbox.

- **git clone ...**: grabs the entire Real-Time-Object-Detection codebase so you can hack on it.
- **cd Real-Time-Object-Detection**: enters the project folder, where all the magic lives.

Final Result:



Setup Update & Dependencies

Now its time to install more prerequisites.

```
sudo apt update

sudo apt install -y wget build-essential zlib1g-dev libncurses5-dev \

libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev
libsqlite3-dev \

libbz2-dev liblzma-dev
```

What's poppin' here:

- **sudo apt update**: syncs your package list with the interwebs—keeps you fresh.
- **sudo apt install -y ...**: grabs core tools and libraries you need to compile Python from scratch.
    - *build-essential*: GCC, make, all that compile jazz.

- *zlib, ncurses, openssl, sqlite…*: these let Python support compression, terminal UIs, encryption, and that sweet built-in database.

Download & Build Python 3.11.9

TensorFlow requires an older version of Python over the new one, because of this , we ill be using an older version.

```
cd /tmp
wget https://www.python.org/ftp/python/3.11.9/Python-3.11.9.tgz
tar -xf Python-3.11.9.tgz
cd Python-3.11.9
./configure --enable-optimizations
make -j$(nproc)
sudo make altinstall
cd ~/Desktop/TensorflowDetection/Real-Time-Object-Detection
```

Deets:

- **cd /tmp**: temp folder so your home dir stays clean.
- **wget …Python-3.11.9.tgz**: download the source tarball of the latest Python you wanna rock.
- **tar -xf …**: unpacks that tarball.
- **./configure --enable-optimizations**: sets flags to squeeze out performance.
- **make -j$(nproc)**: build using all your CPU cores—parallel hustle.
- **sudo make altinstall**: installs as python3.11 so you don't stomp your system Python.

Now check your python version with `python3.11 -version`

```
vboxuser@Ubuntu:~/Desktop/TensorflowDetection/Real-Time-Object-Detection
$ python3.11 --version
Python 3.11.9
```

Setup & Activate Virtual Environment

```
python3.11 -m venv  tensorflow-venv

source tensorflow-venv/bin/activate

pip install --upgrade pip
```

Why VENV?

- **Isolates dependencies**: no clashes with global packages.
- **pip install —upgrade pip**: keeps pip on fleek.

Make sure you have (tensorflow-venv) in your terminal, this means ur in the virtual environment.

```
(tensorflow-venv) vboxuser@Ubuntu:~/Desktop/TensorflowDetection/Real-Tim
e-Object-Detection$
```

Install Python Libraries

```
pip install tensorflow opencv-python numpy

pip install flask

cd ~/Desktop/TensorflowDetection/Real-Time-Object-Detection
```

What you get:

- **TensorFlow**: the neural-net engine.
- **OpenCV**: vision ops—reading frames & drawing boxes.
- **NumPy**: tensor math under the hood.
- **Flask**: lightweight web server to serve your camera stream.

Note: This might take a while….

```
Downloading tensorflow-2.19.0-cp311-cp311-manylinux_2_17_x86_64.manylinu
x2014_x86_64.whl (644.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 240.1/644.9 MB 5.6 MB/s eta 0:01:13
```

Installing Model and Label Map

```
wget
http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd
_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz -O
ssd_mobilenet.tar.gz

tar -xzvf ssd_mobilenet.tar.gz

curl -o mscoco_complete_label_map.pbtxt
https://raw.githubusercontent.com/tensorflow/models/master/research/obj
ect_detection/data/mscoco_complete_label_map.pbtxtcd

cd ~/Desktop/TensorflowDetection/Real-Time-Object-Detection
```

Note: do each command one by one

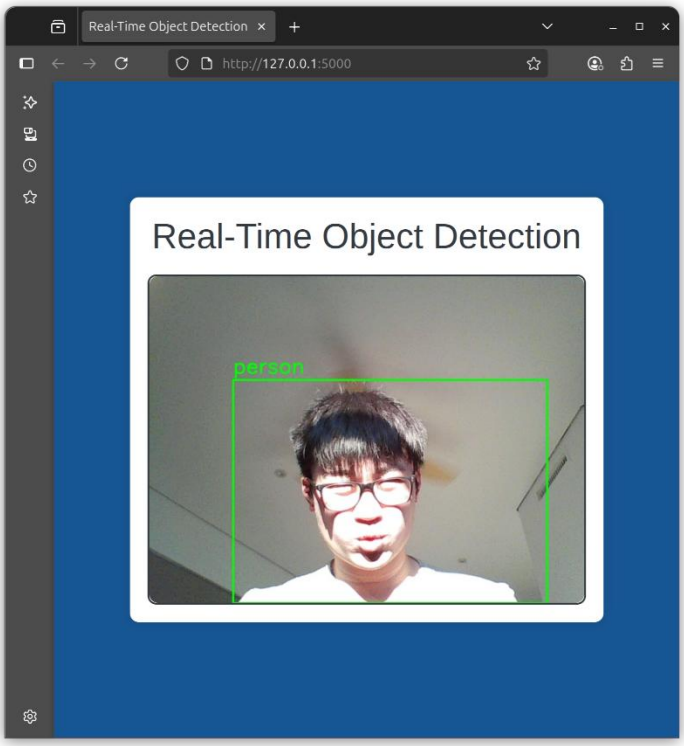
Breakdown:

- **wget ...ssd_mobilenet...**: pulls the SSD-MobileNet model pre-trained on COCO so you don't train from scratch.
- **tar -xzvf**: unpacks the model files (checkpoints, pipeline config).
- **curl -o ...label_map.pbtxt**: downloads the label map—maps integer IDs to human labels (e.g., "person", "dog").

Run and pray 🙏

```
python3.11 model.py

firefox http://127.0.0.1:5000 & # in a new terminal tab
```

Note: if you see a window like this:



Congrats! Youve Reached the end!