

Raspberry Pi Zero (this is only for TFT Monitor 1.8 Inch on Raspberry Pi Zero)

1. Connecting to the eduroam Network or any institutional Network

Universities and enterprises use a different authentication protocol to access internet than the one used at home. This is the PEAP Protocol.

In the case of Germany, to connect to the University network called eduroam, the following steps must be followed:

Now open the data for Network configurations in the root folder:

```
sudo nano /etc/network/interfaces
```

Now we verify that the file has the following information:

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Save and exit by typing **CTRL+O** and **CTRL+X**

Now we edit the wpa_supplicant.conf file in /etc/wpa_supplicant/wpa_supplicant.conf

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

And verify the following content:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
```

```
country=DE
```

In addition the next information must be added:

```
network={
ssid="eduroam"
    proto=RSN
key_mgmt=WPA-EAP
eap=PEAP
    identity="user@uni-bremen.de"
    password="XXXXX"
    phase1="peaplabel=0"
    phase2="auth=MSCHAPV2"
}
```

Save and exit by typing **CTRL+O** and **CTRL+X**

To test the wpa_supplicant can be started typing:

```
sudo wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf
```

Finally reboot and the raspberry should be connected to the internet

```
sudo reboot
```

2. Mplayer

The mplayer is used to play audio and video

```
apt-get install -y mplayer
```

3. Install Pico text to speech

First step. We want the raspberry pi to be able to speech. Install pico2wave

Google Android TTS engine. Very good quality speech.

```
sudo apt-get install libttspico-utils
```

Now test the program using the following command

Verwendbare Sprachkürzel sind de-DE, en-US, en-GB, es-ES fr-FR, it-IT. Eine direkte Wiedergabe kann z.B. mit einer Eingabe wie

```
pico2wave -w test.wav "Hello Christian. How are you?" | mplayertest.wav
```

```
pico2wave --lang de-DE --wave /tmp/Test.wav "Hierfolgt der Test-Text" ; play /tmp/Test.wav; rm /tmp/Test.wav
```

Useful Link

[https://elinux.org/RPi_Text_to_Speech_\(Speech_Synthesis\)](https://elinux.org/RPi_Text_to_Speech_(Speech_Synthesis))

<https://github.com/samjbrahams/tensorflow-on-raspberry-pi>

4. Installing web camera on Raspberry pi

```
sudo apt-get install fswebcam
```

specify resolution

```
fswebcam -r 1280x720 image2.jpg
```

specify no banner

```
fswebcam -r 1280x720 --no-banner image3.jpg
```

Useful Link

<http://downloads.raspberrypi.org/raspbian/images/raspbian-2017-03-03/>

5. Building Tensorflow

Installing tensor flow 1.1 in rasbian 8.0 02/03/2017

<https://github.com/samjabrahams/tensorflow-on-raspberry-pi>

Download Raspbian for the above installation

<http://downloads.raspberrypi.org/raspbian/images/raspbian-2017-03-03/>

<https://github.com/samjabrahams/tensorflow-on-raspberry-pi/blob/master/GUIDE.md>

Troubleshooting:

Installing latest Bazelrelease

<https://github.com/bazelbuild/bazel>

<https://github.com/bazelbuild/bazel/releases>

bazel-0.15.0-dist.zip

INSTALLING TENSORFLOW ORIGINAL

https://www.tensorflow.org/install/install_raspbian

The tensorflow team has uploaded the TensorFlow binaries to piwheels.org. Therefore, you can install TensorFlow through pip.

Just make sure you have the prerequisites programs

- python 3.4+
- Atlas
- Raspbian jessie v9.0 +

```
$ sudo apt-get install python3-pip # for Python 3.n  
sudo apt-get install python-pip # for Python 2.7
```

Install Atlas

```
$ sudo apt install libatlas-base-dev
```

The command originally in the web page is:

```
<pre> $ <b>pip3 install tensorflow</b> # Python 3.n
```

You may only need to run this:

```
sudo pip3 install tensorflow # Python 3.n
```

To find out tensorflow version

```
Pip3 list | grep tensorflow
```

Update: Installing tensorflow on python 2.7

MNIST models seem not to work properly using python 3.+ while evaluating the models (incompatibility of numpy arrays between Python 2 and 3) so it is advisable to work using python 2.7

You may only need to run this:

```
sudo pip install --no-cache-dir tensorflow # Python 2.n
```

use the `--no-cache-dir` command in case of **memory error**

MNIST in TensorFlow

MNIST is a handwritten digit database with a bunch of images, so it can be used to detect handwritten digits

<https://github.com/tensorflow/models/tree/master/official/mnist>

Installing Tensorflow Official models

<https://github.com/tensorflow/models/tree/master/official#running-the-models>

Installing official Models

First navigate to the tensorflow directory

```
cd /usr/local/lib/python3.5/dist-packages/tensorflow
```

Now clone the official models repo from github

```
git clone https://github.com/tensorflow/models.git
```

Now you can follow the instructions from the link above.

Installing Requirements for official models

<https://github.com/tensorflow/models/tree/master/official#running-the-models>

Please follow the below steps before running the official models:

Add the top-level /models folder to the Python path with the command:

```
export PYTHONPATH="$PYTHONPATH:/path/to/models"
```

Install Dependencies

```
pip3 install --user -r official/requirements.txt
```

```
pip install --user -r official/requirements.txt
```

Nota: replace `:/path/to/models` with the path to the folder models you have

```
export PYTHONPATH="$PYTHONPATH:/usr/local/lib/python3.5/dist-packages/tensorflow  
/models"
```

```
pip3 install --user -r /usr/local/lib/python3.5/dist-packages/tensorflow  
/models/official/requirements.txt
```

```
pi@raspberrypi:~$ pip list | grep tensorflow
DEPRECATION: The default format will switch to columns in the future. You can use
--format=(legacy|columns) (or define a format=(legacy|columns) in your pip.conf
under the [list] section) to disable this warning.
tensorflow (1.9.0)
pi@raspberrypi:~$ pip3 list | grep tensorflow
DEPRECATION: The default format will switch to columns in the future. You can use
--format=(legacy|columns) (or define a format=(legacy|columns) in your pip.conf
under the [list] section) to disable this warning.
tensorflow (1.9.0)
pi@raspberrypi:~$ export PYTHONPATH="/usr/local/lib/python2.7/dist-
packages/tensorflow/models"
pi@raspberrypi:~$ pip install --user -r /usr/local/lib/python2.7/dist-packages/
tensorflow/models/official/requirements.txt
Collecting google-api-python-client>=1.6.7 (from -r /usr/local/lib/python2.7/dis
t-packages/tensorflow/models/official/requirements.txt (line 1))
Collecting google-cloud-bigquery>=0.31.0 (from -r /usr/local/lib/python2.7/dist-
packages/tensorflow/models/official/requirements.txt (line 2))
Using cached https://files.pythonhosted.org/packages/0f/9f/45a7e4d1731d6b2cc0f
6011d763fa4eec85956515306f47ecc50b38bdf6d/google_cloud_bigquery-1.4.0-py2.py3-no
ne-any.whl
```

Abbildung1Exporting pythonpath and installing official requirements

Training the Convolutional Neural Network

The MNIST data set has its own training tutorial on the link.

<https://github.com/tensorflow/models/tree/master/official/mnist>

Run the following command to train the CNN using the MNIST data set

```
python /usr/local/lib/python2.7/dist-packages/tensorflow/models/official/mnist/mni
st.py --export_dir /tmp/mnist_saved_model
```

with python 3.5 it would be

```
python3.5 /usr/local/lib/python3.5/dist-packages/tensorflow/models/official/mnist/
mnist.py --export_dir /tmp/mnist_saved_model
```

Note: just make sure the path is the official models path downloaded before.


```
pi@raspberrypi:~/TensorFlow/mnist_model $ ls -l
total 200172
-rw-r--r-- 1 pi pi      277 Jul 21 01:39 checkpoint
drwxr-xr-x 2 pi pi     4096 Jul 20 15:30 eval
-rw-r--r-- 1 pi pi  7256736 Jul 21 01:39 events.out.tfevents.1532098946.raspberrypi
-rw-r--r-- 1 pi pi  3806654 Jul 21 00:44 graph.pbtxt
-rw-r--r-- 1 pi pi 39295624 Jul 21 00:55 model.ckpt-13395.data-00000-of-00001
-rw-r--r-- 1 pi pi    976 Jul 21 00:55 model.ckpt-13395.index
-rw-r--r-- 1 pi pi  159951 Jul 21 00:55 model.ckpt-13395.meta
-rw-r--r-- 1 pi pi 39295624 Jul 21 01:05 model.ckpt-13563.data-00000-of-00001
-rw-r--r-- 1 pi pi    976 Jul 21 01:05 model.ckpt-13563.index
-rw-r--r-- 1 pi pi  159951 Jul 21 01:05 model.ckpt-13563.meta
-rw-r--r-- 1 pi pi 39295624 Jul 21 01:15 model.ckpt-13728.data-00000-of-00001
-rw-r--r-- 1 pi pi    976 Jul 21 01:15 model.ckpt-13728.index
-rw-r--r-- 1 pi pi  159951 Jul 21 01:15 model.ckpt-13728.meta
-rw-r--r-- 1 pi pi 39295624 Jul 21 01:28 model.ckpt-13735.data-00000-of-00001
-rw-r--r-- 1 pi pi    976 Jul 21 01:28 model.ckpt-13735.index
-rw-r--r-- 1 pi pi  159951 Jul 21 01:28 model.ckpt-13735.meta
-rw-r--r-- 1 pi pi 39295624 Jul 21 01:39 model.ckpt-13736.data-00000-of-00001
-rw-r--r-- 1 pi pi    976 Jul 21 01:39 model.ckpt-13736.index
-rw-r--r-- 1 pi pi  159951 Jul 21 01:39 model.ckpt-13736.meta
```

In the official tutorial you can get new predictions with some test data from the tutorial y the means of the following command. If the saved_model_cli function is not available, you may need to download and run the function using python. (see troubleshooting below)

```
saved_model_cli run --dir /tmp/mnist_saved_model/TIMESTAMP --tag_set serve --signature_def classify --inputs image=examples.npy
```

This command was useful with previous versions of tensorflow, however now you may need to create your own python script and use the tf.train.import_meta_graph class since the format of the modles has changed (see tutorial on the link)

About the class:

https://www.tensorflow.org/api_docs/python/tf/train/import_meta_graph

The tutorial

<http://cv-tricks.com/tensorflow-tutorial/save-restore-tensorflow-models-quick-complete-tutorial/>

b) Load the parameters:

We can restore the parameters of the network by calling restore on this saver which is an instance of `tf.train.Saver()` class.

```
1
2 with tf.Session() as sess:
3     new_saver = tf.train.import_meta_graph('my_test_model-1000.meta')
4     new_saver.restore(sess, tf.train.latest_checkpoint('./'))
```

After this, the value of tensors like `w1` and `w2` has been restored and can be accessed:

```
1
2 with tf.Session() as sess:
3     saver = tf.train.import_meta_graph('my-model-1000.meta')
4     saver.restore(sess,tf.train.latest_checkpoint('./'))
5     print(sess.run('w1:0'))
6 ##Model has been restored. Above statement will print the saved value of w1.
```

So, now you have understood how saving and importing works for a Tensorflow model. In the next section, I have described a practical usage of above to load any pre-trained model.

Troubleshooting

Saved_model_cli:

https://www.tensorflow.org/versions/r1.2/programmers_guide/saved_model_cli

SavedModel is a universal serialization format for Tensorflow. It provides a language-neutral format to save machine-learned models and enables higher-level systems and tools to produce, consume and transform TensorFlow models.

It can happen that after the installation Raspbian does not recognize this function. In that case we can download the original `.py` function from the official repo

https://github.com/tensorflow/tensorflow/blob/f202958ee2d5177a474e3d107fdbf0c83174d099/tensorflow/python/tools/saved_model_cli.py

```
wgethttps://raw.githubusercontent.com/tensorflow/tensorflow/f202958ee2d5177a474e3d107fdbf0c83174d099/tensorflow/python/tools/saved_model_cli.py
```

Run with the following command

```
Python saved_model_cli.py 'run' --dir ~/TensorFlow/route/to/model --tag_set serve
--signature_def classify --inputs image=examples.npy
```

6. Visualize png images using console

Use xdg-open command

```
Xdg-open image.png
```

7. List of useful commands in Linux

Find command

Find a file called testfile.txt in current and sub-directories.

```
find . -name testfile.txt
```

Find all .jpg files in the /home and sub-directories.

```
find /home -name '*.jpg'
```

git command

to clone repositories from github

```
git clone https://github.com/tensorflow/models.git
```

grep command

The grep command is used to search text.

You can search recursively i.e. read all files under each directory for a string "word"

```
grep -r "word" /etc/
```

Other examples

```
grep 'word' filename  
grep 'word' file1 file2 file3  
grep 'string1 string2' filename  
cat otherfile | grep 'something'  
command | grep 'something'  
command option1 | grep 'data'  
grep --color 'data' fileName
```