Ambient AI Bootcamp

Practice 5

SNU Graduate School of Data Science

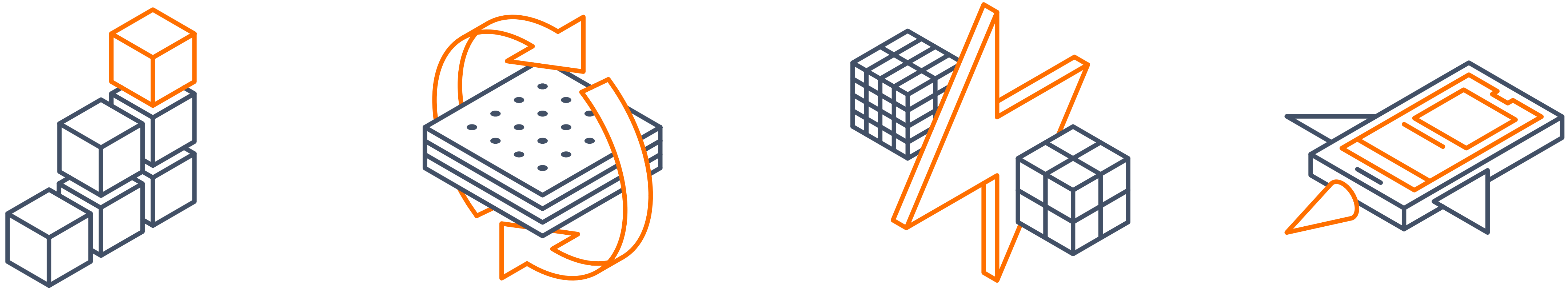
Table of Contents

- Introduction to TensorFlow Lite
- Quantization
  - Post-Training Quantization
  - Quantization-Aware Training
- Pruning
- Coral Dev Board

5-1. Introduction to TensorFlow Lite

TensorFlow Lite

Library for deploying models on mobile, microcontrollers, and other edge devices



1. Build a model

2. Convert

3. Optimize

4. Deploy

TensorFlow Lite

Optimized for five core constraints

1. **Latency**
   - No round-trip to a server
2. **Privacy**
   - No personal data leaves the device
3. **Connectivity**
   - Internet connection not required
4. **Size**
   - Reduced model size, smaller download size
5. **Power consumption**
   - Efficient inference

Tensorflow Lite: Getting Started

First, download and normalize the fashion MNIST dataset

```
import tensorflow as tf
import numpy as np

# Load MNIST dataset
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) =
fashion_mnist.load_data()

# Normalize the input image
train_images = train_images.astype(np.float32) / 255.0
test_images = test_images.astype(np.float32) / 255.0
```

1. Build a model

Tensorflow Lite: Getting Started

Next, define the model architecture

```
model = Sequential([
    InputLayer(input_shape=(28, 28)),
    Reshape(target_shape=(28, 28, 1)),
    Conv2D(filters=16, kernel_size=3, padding='same', activation='relu'),
    MaxPool2D(pool_size=(2,2), strides=(2,2)),
    Conv2D(filters=32, kernel_size=3, padding='same', activation='relu'),
    MaxPool2D(pool_size=(2,2), strides=(2,2)),
    Flatten(),
    Dense(10, activation='softmax')
])
```

1. Build a model

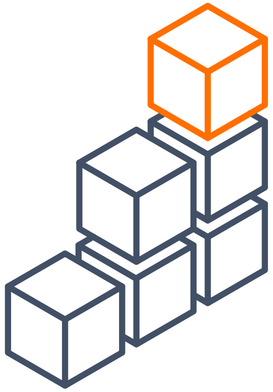
Tensorflow Lite: Getting Started

Next, train/optimize the model

- We can also add ***quantization aware training*** in this step

```
model.compile(optimizer='adam',
              loss=SparseCategoricalCrossentropy(
                  from_logits=False),
              metrics=['accuracy'])
model.fit(
  train_images,train_labels, epochs=10,
  validation_data=(test_images, test_labels)
)

metrics = model.evaluate(test_images, test_labels)
```



1. Build a model

```
Model validation loss: 0.254 | validation accuracy: 90.99%
```

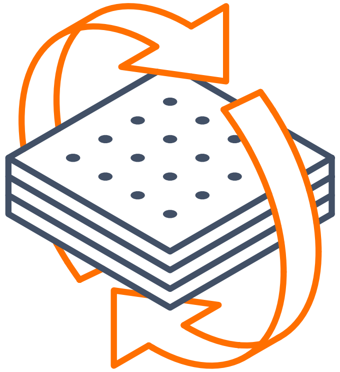
Tensorflow Lite: Getting Started

To convert the model to TFLite, initialize a ***converter***

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
```

We can now save the tflite model and deploy it on mobile!

```
with open('model.tflite', 'wb') as f:
    f.write(tflite_model)
```



2. Convert

Tensorflow Lite: Getting Started

Next, we have several options available to *optimize* the model

- Typically, we use *Tensorflow Model Optimization Toolkit*

Two methods:

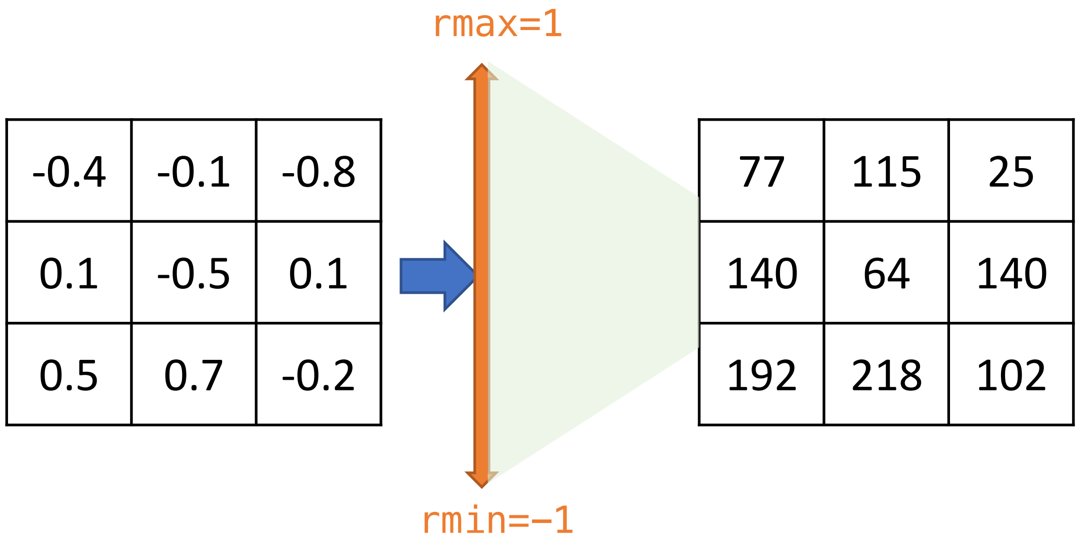
- Quantization
  - Post-Training Quantization (PTQ)
  - Quantization-Aware Training (QAT)
- Pruning

3. Optimize

Post-Training Quantization

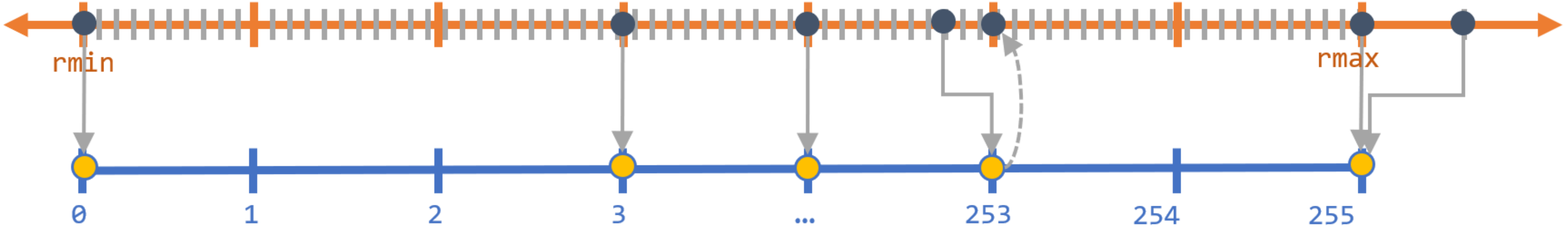
PTQ는 학습을 완료한 다음에 Quantization 하는 방법

- Edge TPU가 있는 Coral Board를 사용할 때, 8-bit Integer 연산만 가능함
- Quantization을 통해 모든 weight와 activation은 0~255 또는 2's complement -128~127 의 8-bit 정수로 변환됨
- 32bit→8bit로, 모델의 크기는 75% 정도 작아짐



-0.4	-0.1	-0.8
0.1	-0.5	0.1
0.5	0.7	-0.2

77	115	25
140	64	140
192	218	102



Min/max real values map to min/max quantized value

Some values are exactly representable, and 0.f is always one of them

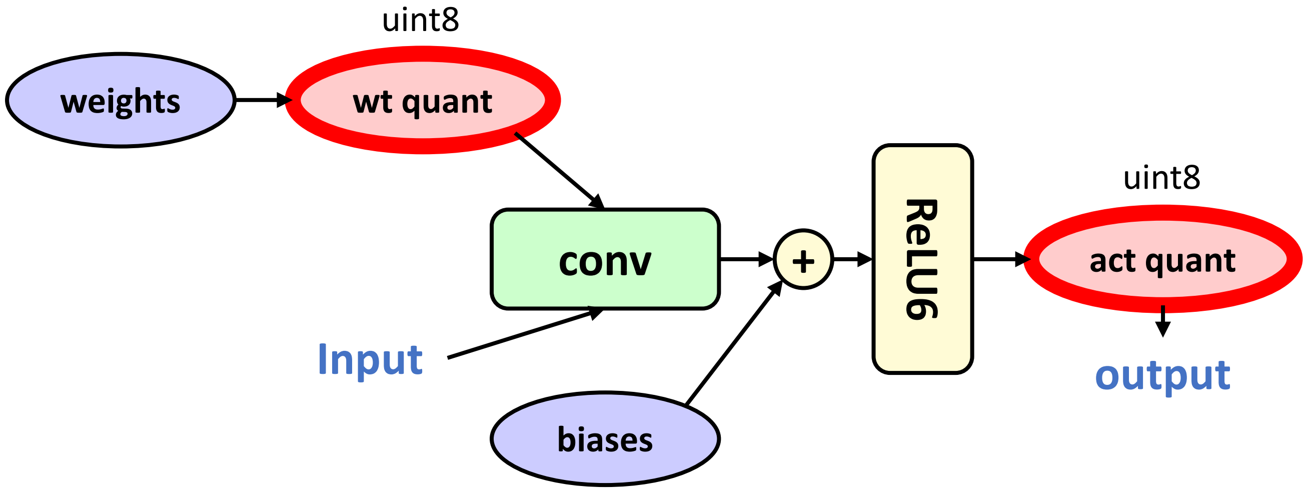
Others end up getting rounded. Dequantizing them back gives the nearest fixed-point number

Values outside the original min/max range get clamped

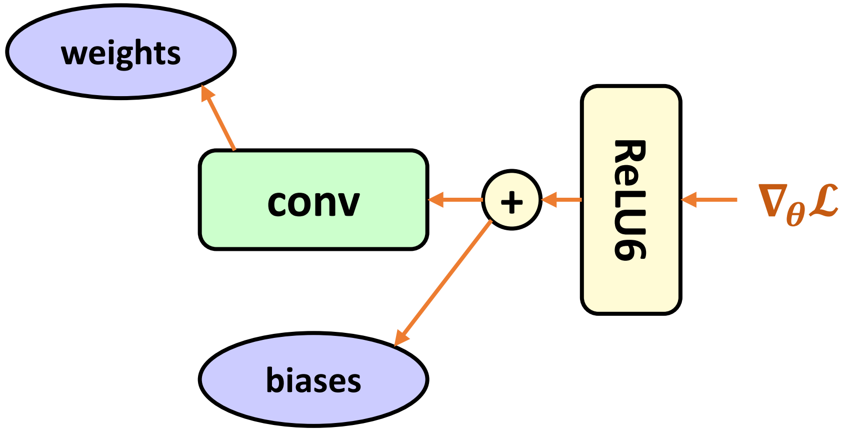
Quantization-Aware Training

QAT는 **학습 도중**에 이루어지고, inference할 때는 integer연산, backpropagation할 때에는 full-precision으로 모델을 학습함

- QAT 방식으로 학습하면, 최종 quantized 성능이 PTQ보다 좋다고 함



Inference



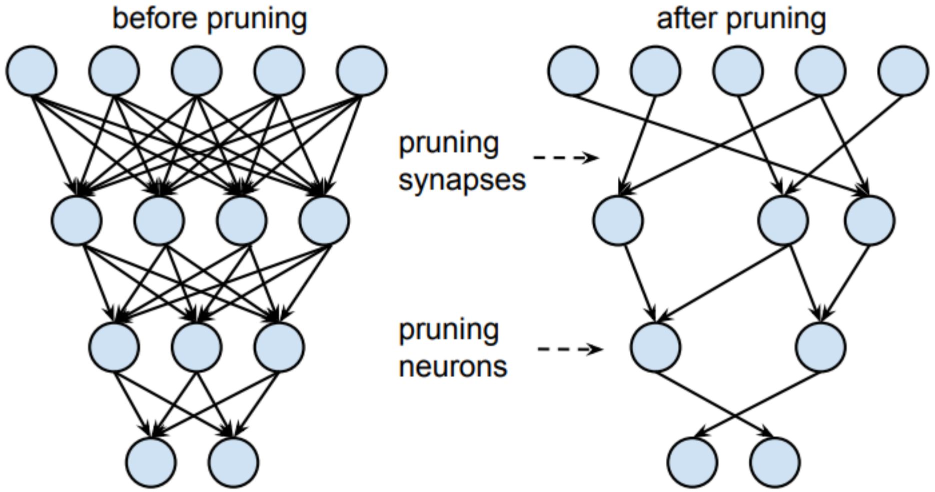
Backpropagation

Pruning

불필요한 (0에 가까운) weight들을 0으로 만들고 없애면서 모델 경량화

TFLite에서는 Gradual Pruning 방법론을 사용함

- `initial_sparsity`: pruning을 시작할 때의 sparsity를 몇으로 할지
- `final_sparsity`: pruning을 끝낼 때 sparsity를 몇으로 할지
- `begin_step`: pruning을 언제부터 진행할 지(batch 단위의 step)
- `end_step`: pruning을 언제 끝낼 지



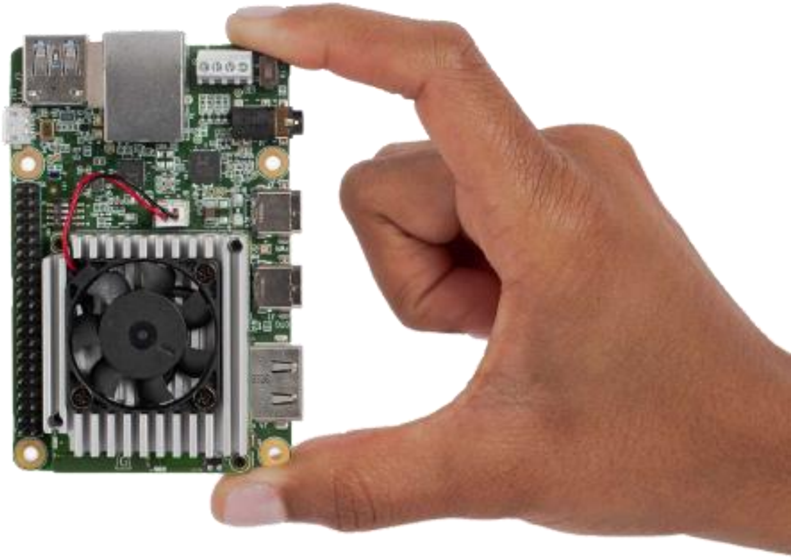
To prune, or not to prune: exploring the efficacy of pruning for model compression [arXiv '17]

5-2. Introduction to Coral Dev Board

Coral Board

- Coral Dev Board Introduction
  - Single-board computer that performs high-speed ML in a small form factor
  - On-board Edge TPU (Tensor Processing Unit) performs 4 trillion operations per second(TOPS), using only 0.5 watts for each TOPS

- Device Specifications
  - CPU : NXP i.MX 8M SoC(Quad-core Arm Cortex-A53, plus Cortex-M4F)
  - GPU : Integrated GC7000 Lite Graphics
  - ML accelerator : Google Edge TPU
  - RAM : 1GB LPDDR4(or 4GB)
  - eMMC(Storage) : 8GB + MicroSD



Coral Board Requirements

- ☑ A host computer running Linux (recommended), Mac, or Windows ≥ 10
  - ☑ **(Important)** Python3 installed
- ☑ One USB-C power supply (e.g. phone charger)
- ☑ One USB-C to USB-A cable (to connect to your computer)
- ☑ An available Wi-Fi Connection

If starting from scratch, visit the official website for more information!

Coral Board Access (Windows)

1. Install Git Bash terminal on Windows, and open the Git Bash terminal (it should look like below)

```
user@AIOT-Desktop MINGW64 ~
$
```

2. Add the Python3 executable file to PATH
   - Replace <PATH> with the path to the executable file (e.g., /C/Users/user/Executables/Python3.10/python.exe)

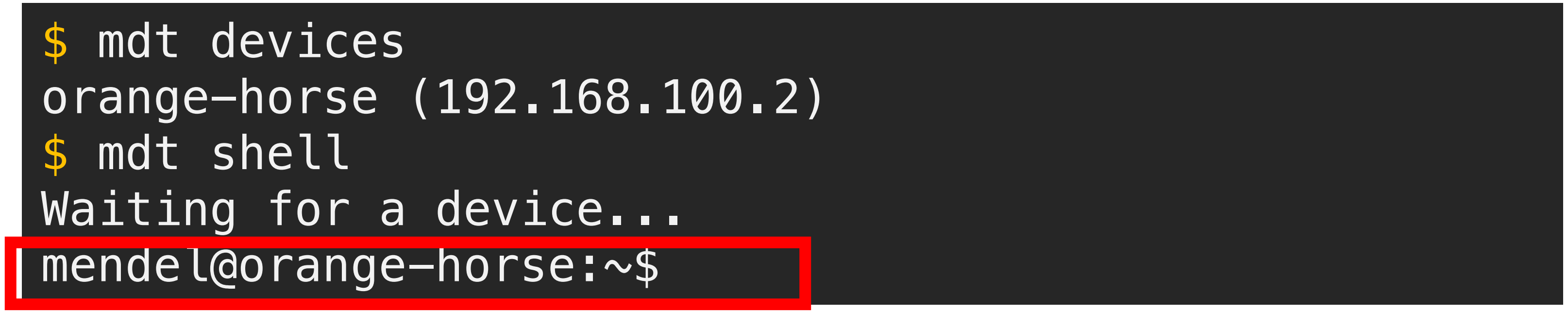
```
$ echo “alias python=‘winpty <PATH>’” >> ~/.bash_profile
$ source ~/.bash_profile
```

3. Install MDT and add mendel to PATH
   - Replace <PATH> with the path containing Python3 (e.g., Executables/Python3.10)

```
$ python –m pip install mendel-development-tool
$ echo ‘export PATH=“$PATH:$HOME/.local/bin”’ >> ~/.bash_profile
$ echo ‘export PATH=“$PATH:$HOME/<PATH>/Scripts”’ >> ~/.bash_profile
$ echo “alias mdt=‘winpty mdt’” >> ~/.bash_profile
$ source ~/.bash_profile
```

Coral Board Access (Windows)

- Connect to the board’s shell via MDT



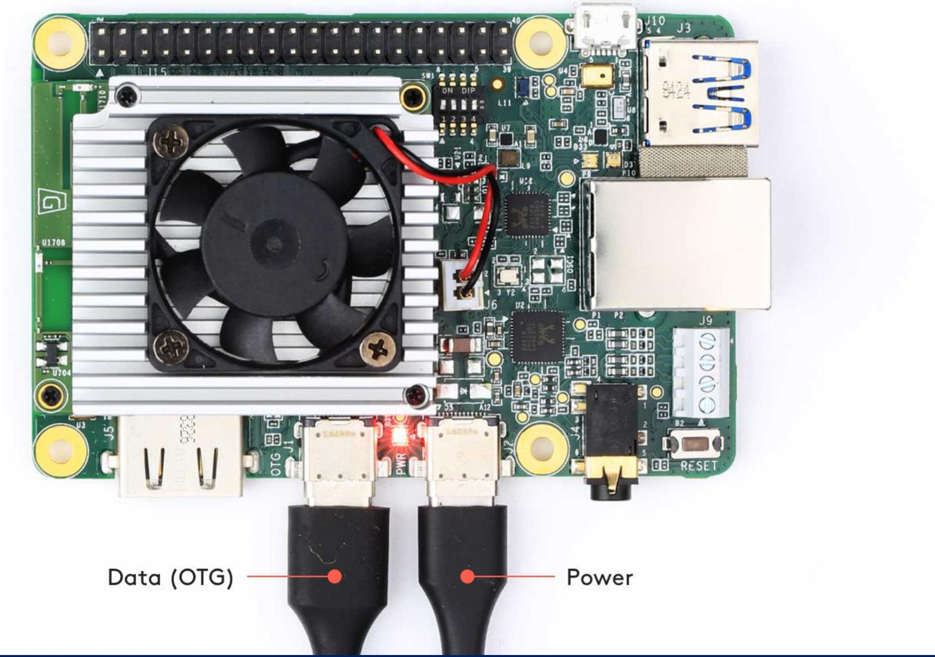
```
$ mdt devices
orange-horse (192.168.100.2)
$ mdt shell
Waiting for a device...
mendel@orange-horse:~$
```

- Connect to Wi-Fi

```
mendel@orange-horse:~$ nmtui
```

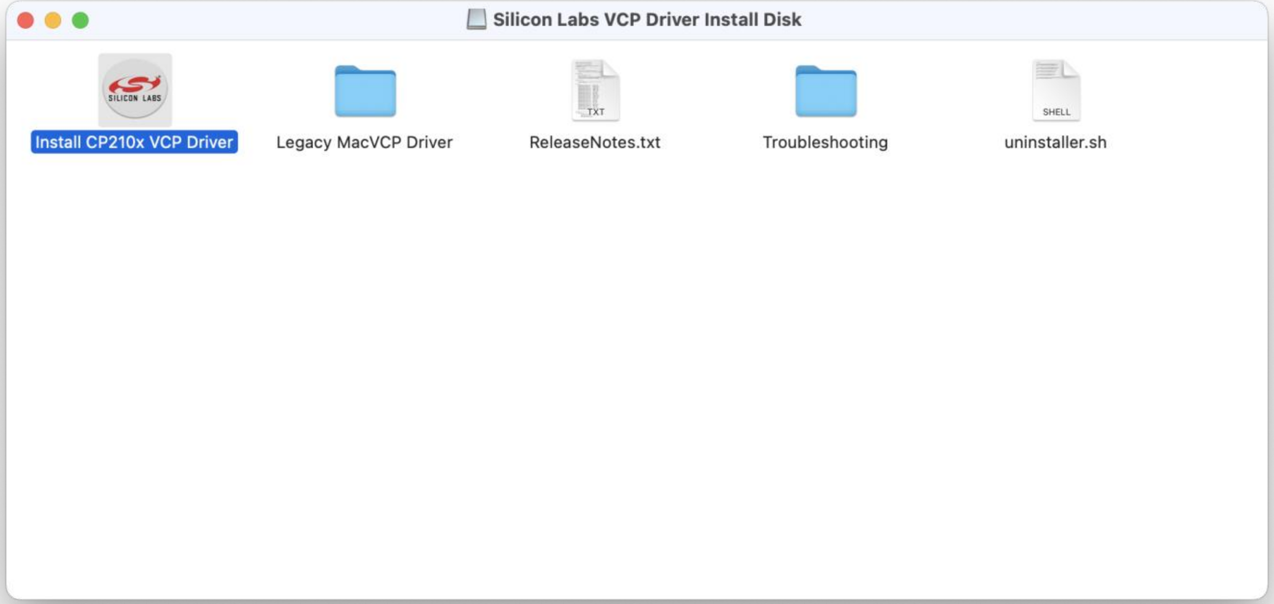
- Shut down the coral board using:

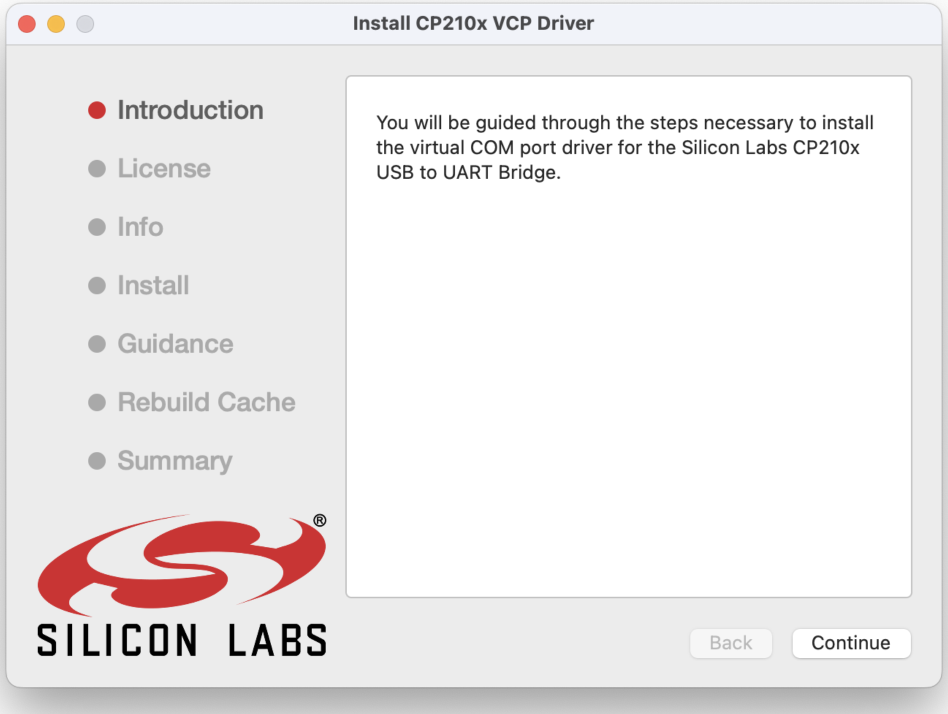
```
mendel@orange-horse:~$ sudo shutdown sh
```



Coral Board Access (Mac)

- Install the CP210x USB to UART Bridge VCP Driver
  - Download the driver from this [link]
  - Unzip the package and install the driver

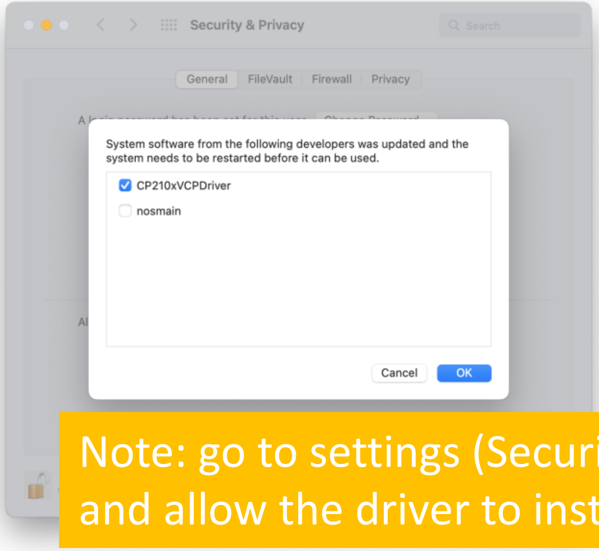




You will be guided through the steps necessary to install the virtual COM port driver for the Silicon Labs CP210x USB to UART Bridge.

Coral Board Access (Mac)

- Install the CP210x USB to UART Bridge VCP Driver
  - Download the driver from this link
  - Unzip the package and install the driver

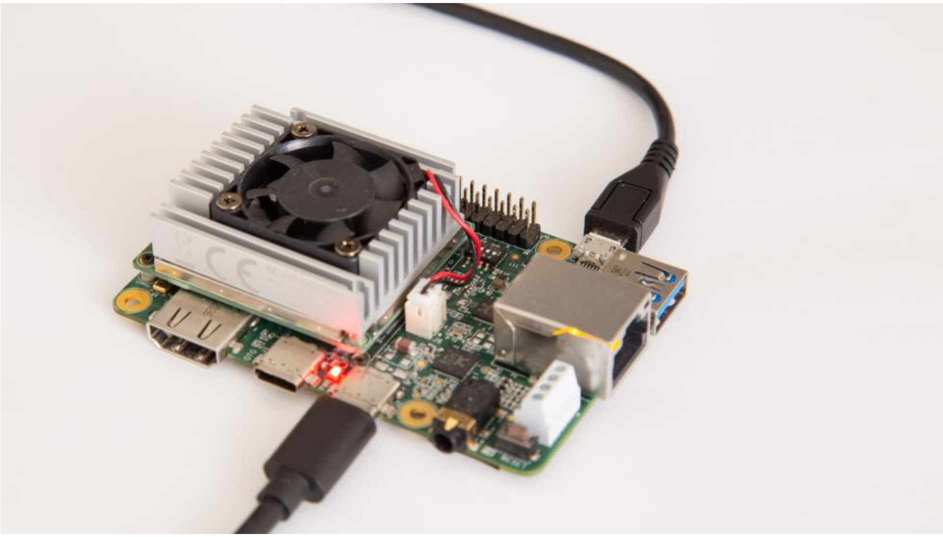


System software from the following developers was updated and the system needs to be restarted before it can be used.

☑ CP210xVCPDriver
□ nosmain

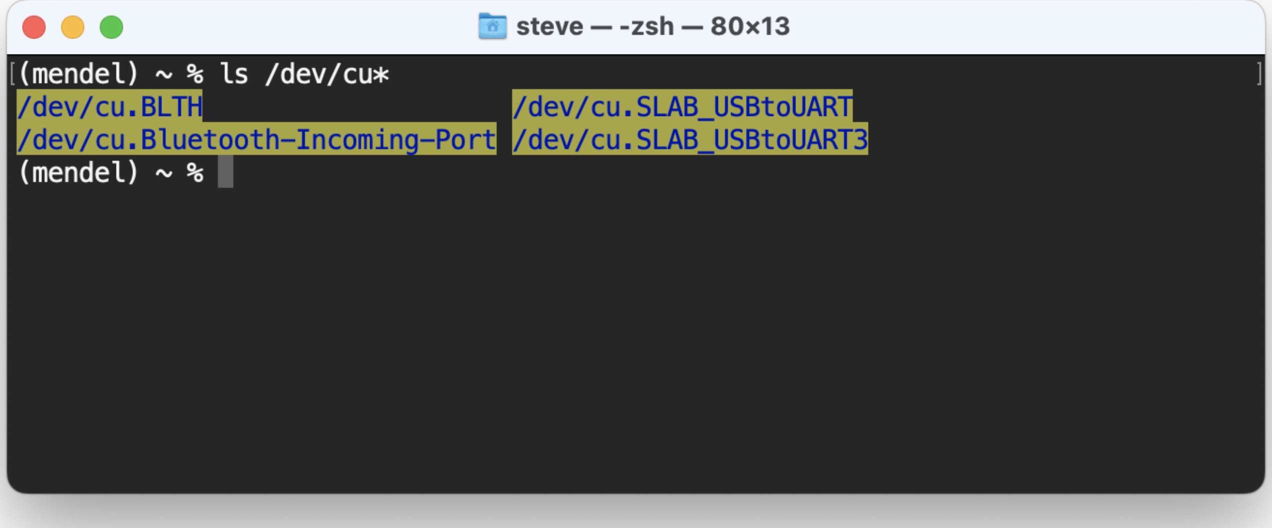
Note: go to settings (Security and Privacy) and allow the driver to install

- Connect your computer to the board with the micro-B USB cable and connect the board to power



Coral Board Access (Mac)

- Verify the CP210x driver is working by running this command:



```
(mendel) ~ % ls /dev/cu*
/dev/cu.BLTH                        /dev/cu.SLAB_USBtoUART
/dev/cu.Bluetooth-Incoming-Port /dev/cu.SLAB_USBtoUART3
(mendel) ~ %
```

- You should see the /dev/cu.SLAB_USBtoUART listed
- If not, check this link for more details

Coral Board Access (Mac)

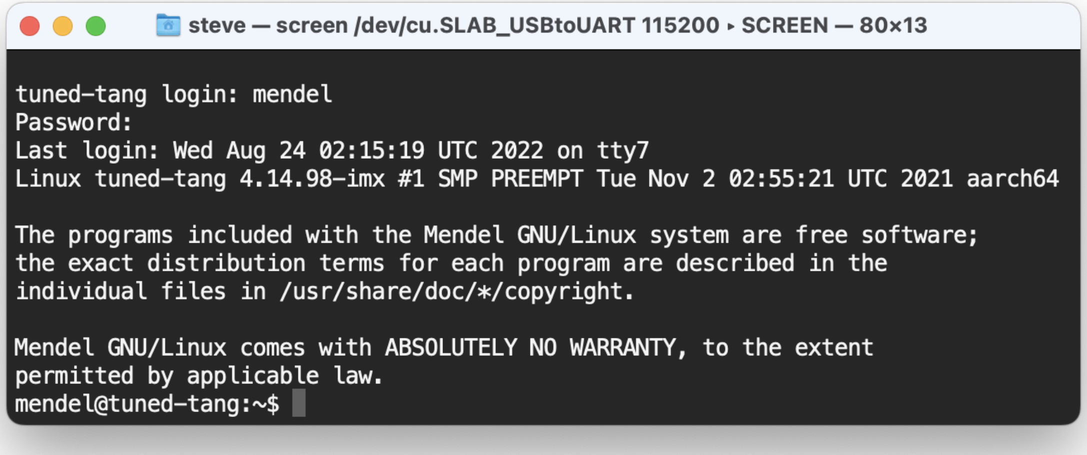
- Connect to the board with this command

```
(mendel) ~ % screen /dev/cu.SLAB_USBtoUART 115200
```

- You will probably see a blank screen.
  - Press enter and you will see a screen as follows:

```
tuned-tang login:
```

  - The username and password are both “mendel” (without the apostrophes)


steve — screen /dev/cu.SLAB_USBtoUART 115200 ▸ SCREEN — 80×13

```
tuned-tang login: mendel
Password:
Last login: Wed Aug 24 02:15:19 UTC 2022 on tty7
Linux tuned-tang 4.14.98-imx #1 SMP PREEMPT Tue Nov 2 02:55:21 UTC 2021 aarch64

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

Mendel GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
mendel@tuned-tang:~$
```

Coral Board Access (Mac)

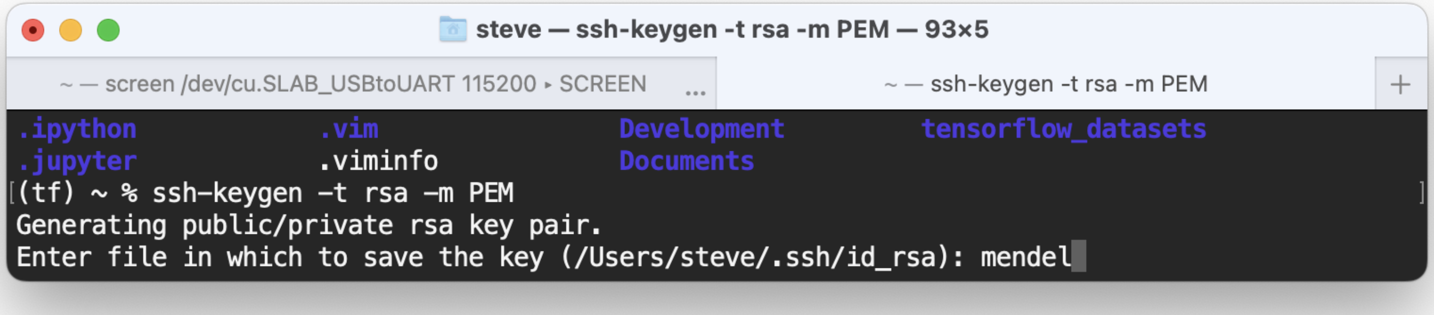
- In the serial console, create a new file for the public SSH key

```
mkdir /home/mendel/.ssh && vi /home/mendel/.ssh/authorized_keys
```

- On your Mac, open another terminal and create a PEM-formatted SSH key pair

```
ssh-keygen -t rsa -m PEM
```

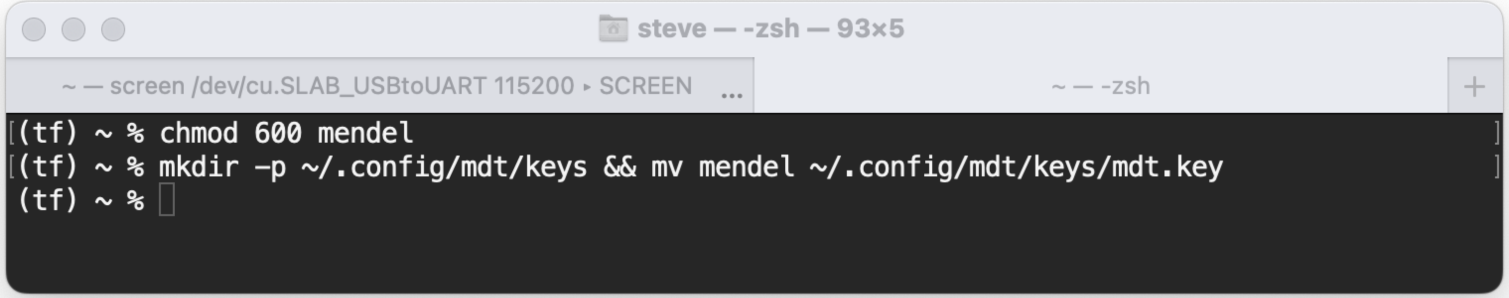
  - When prompted to enter a file name, type “mendel” and leave the passphrase empty


steve — ssh-keygen -t rsa -m PEM — 93×5
~ — screen /dev/cu.SLAB_USBtoUART 115200 ▸ SCREEN
~ — ssh-keygen -t rsa -m PEM

```
.ipython              .vim                  Development           tensorflow_datasets
.jupyter              .viminfo              Documents
(tf) ~ % ssh-keygen -t rsa -m PEM
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/steve/.ssh/id_rsa): mendel
```

Coral Board Access (Mac)

- Set the file permissions and relocate the private key on your Mac as shown here:



```
[(tf) ~ % chmod 600 mendel
[(tf) ~ % mkdir -p ~/.config/mdt/keys && mv mendel ~/.config/mdt/keys/mdt.key
(tf) ~ %
```

- Now put the public key on the Coral board:
  - In your Mac terminal, view the `mendel.pub` file (type `cat mendel.pub`) and copy the file contents
  - Go to the serial console and paste the key into the authorized_keys file you created
  - Save and close the file (ESC -> :wq -> ENTER)

- Make sure your Coral board is **on the same local network** as your Mac (same Wi-Fi)
- Finally, open a new terminal on your Mac and connect to the board

```
mdt shell
```

Setting up the Coral Dev Board

1. Connect to WiFi using `nmtui`
    - If `nmtui` doesn’t work, use the following command:

```
nmcli dev wifi connect <NETWORK_NAME> password <PASSWORD> ifname wlan0
```

2. Update the Coral Board

```
sudo apt-get update
sudo apt-get dist-upgrade
```

Run a Model Using the PyCoral API

Let’s perform an inference on the EdgeTPU using the TFLite API

1. Download the example code from GitHub

```
mkdir coral && cd coral
git clone https://github.com/google-coral/pycoral.git
cd pycoral
```

2. Download the model, labels, and a bird photo

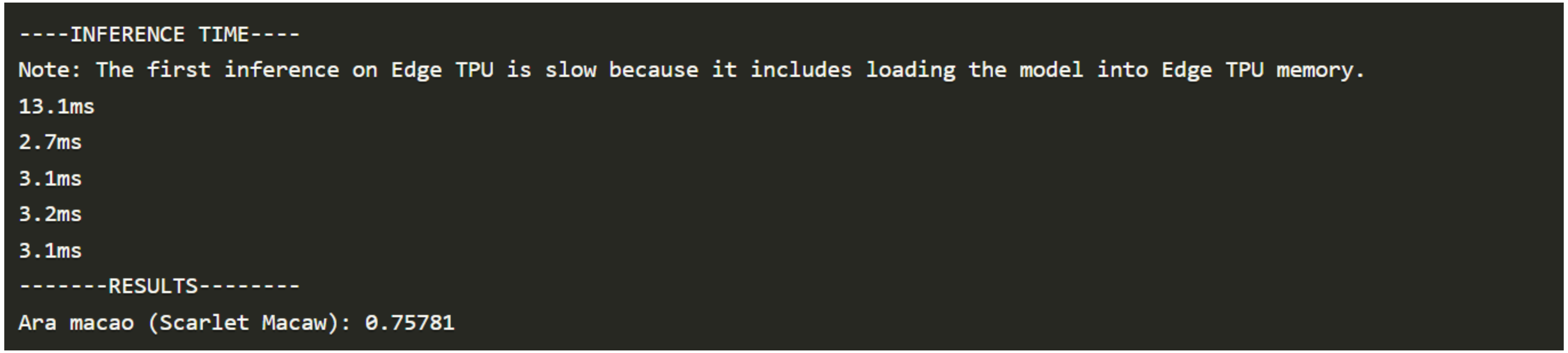
```
bash examples/install_requirements.sh classify_image.py
```

3. Run the image classifier with the bird photo

```
python3 examples/classify_image.py \
--model test_data/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite \
--labels test_data/inat_bird_labels.txt \
--input test_data/parrot.jpg
```

Run a Model Using the PyCoral API

You should see results as follows:

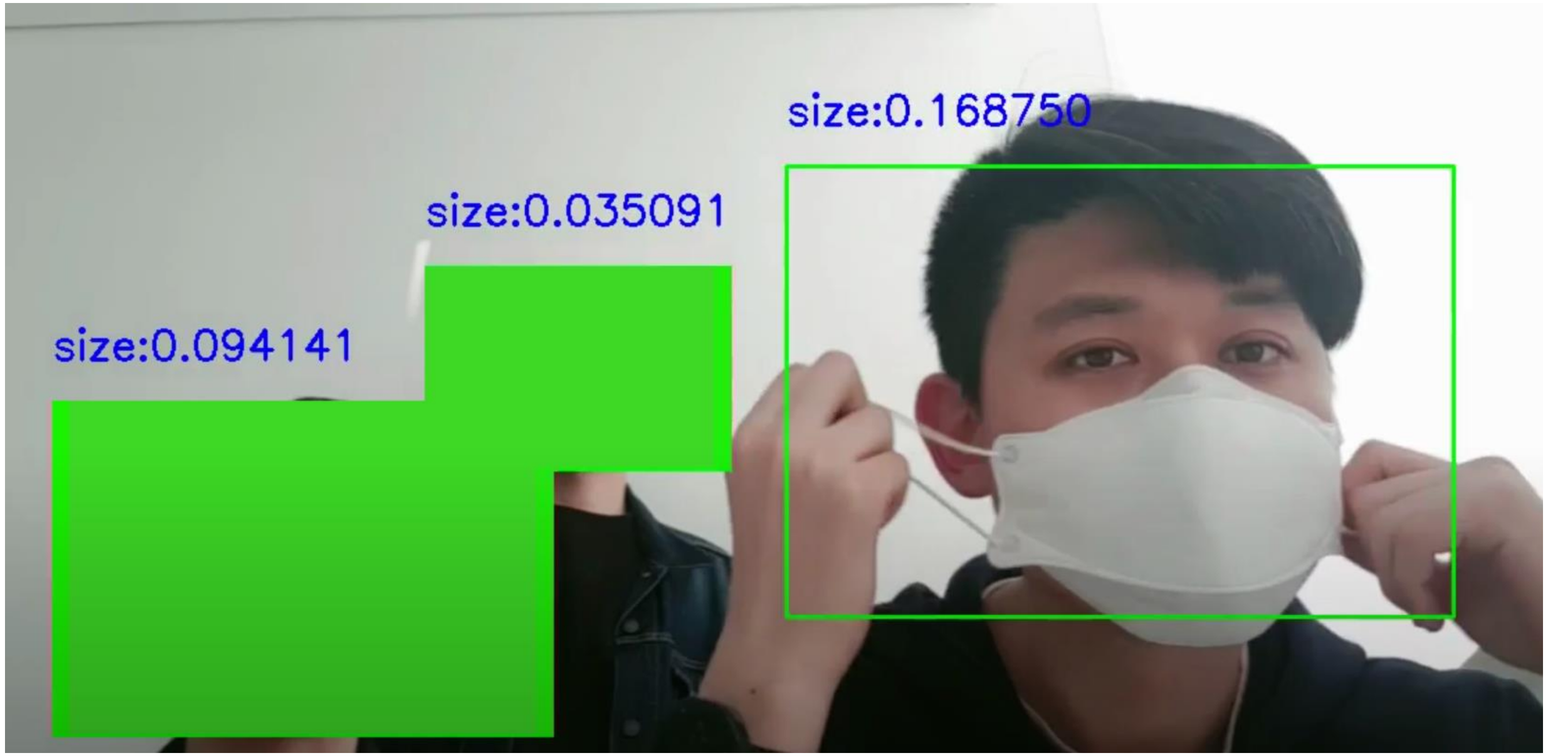


```
----INFERENCE TIME----
Note: The first inference on Edge TPU is slow because it includes loading the model into Edge TPU memory.
13.1ms
2.7ms
3.1ms
3.2ms
3.1ms
-------RESULTS--------
Ara macao (Scarlet Macaw): 0.75781
```

Check the link for more information

Next Class…

- Face Detection with Coral Dev Board
  - Largest face detection
  - Mask all other faces



Thank You!

If you need a coral dev board, contact me at
steve2972@snu.ac.kr

Supplementary Slides

Appendix1. Setting up the coral dev board

Setting Up Coral Board for the first time (Linux)

- SD카드를 이용하지 않고 최초 세팅하는 방법이다. 코랄 공식 홈페이지에는 SD카드를 이용하는 방법이 메인으로 소개되어 있으나 본 강의에서는 SD카드를 사용하지 않는다.
- Screen, fastboot 설치

```
sudo apt-get install screen
sudo apt-get install fastboot
```

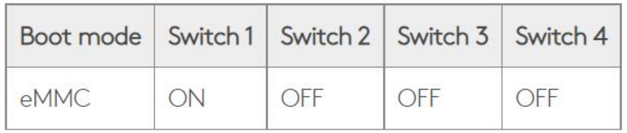
- Fastboot 위한 설정

```
sudo sh -c "echo 'SUBSYSTEM==\"usb\", ATTR{idVendor}==\"0525\", MODE=\"0664\", \
GROUP=\"plugdev\", TAG+=\"uaccess\"' >> /etc/udev/rules.d/65-edgetpu-board.rules"
```

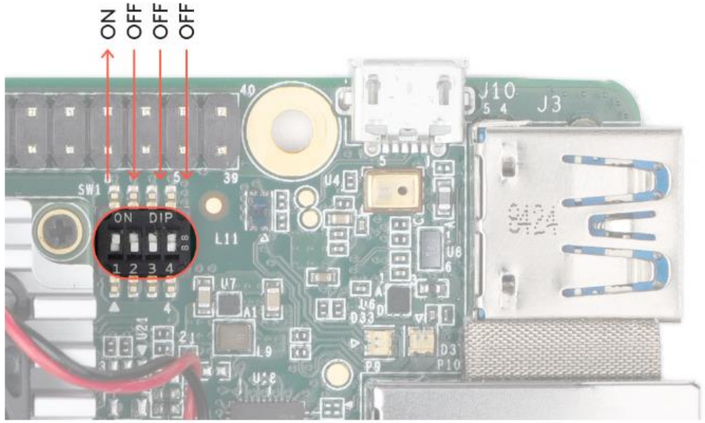
```
sudo udevadm control --reload-rules && sudo udevadm trigger
```

```
sudo usermod -aG plugdev,dialout <username>
```

- 코랄에 boot mode가 잘 설정되어 있는지 확인

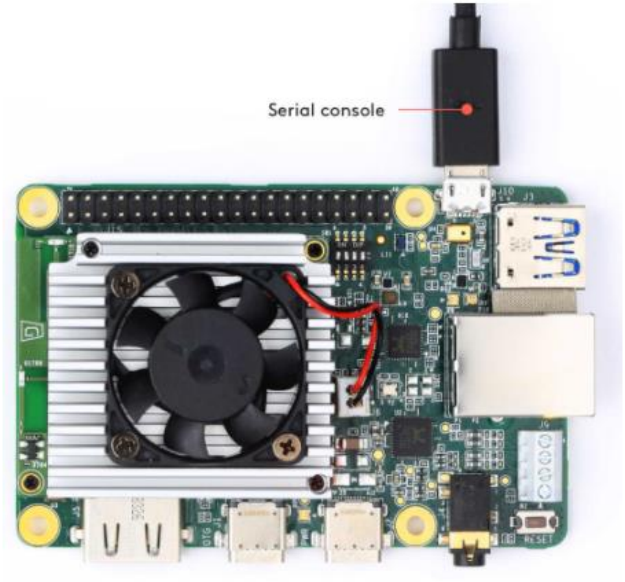


Boot mode	Switch 1	Switch 2	Switch 3	Switch 4
eMMC	ON	OFF	OFF	OFF



Setting Up Coral Board for the first time (Linux)

- 5pin짜리 케이블 통해 컴퓨터와 코랄 연결. 전원은 연결하지 않아도 된다.



- 연결이 잘 되었는지 확인. 아래 command 입력했을 때 메시지가 나와야 한다.

```
dmesg | grep ttyUSB
[ 6437.706335] usb 2-13.1: cp210x converter now attached to ttyUSB0
[ 6437.708049] usb 2-13.1: cp210x converter now attached to ttyUSB1
```

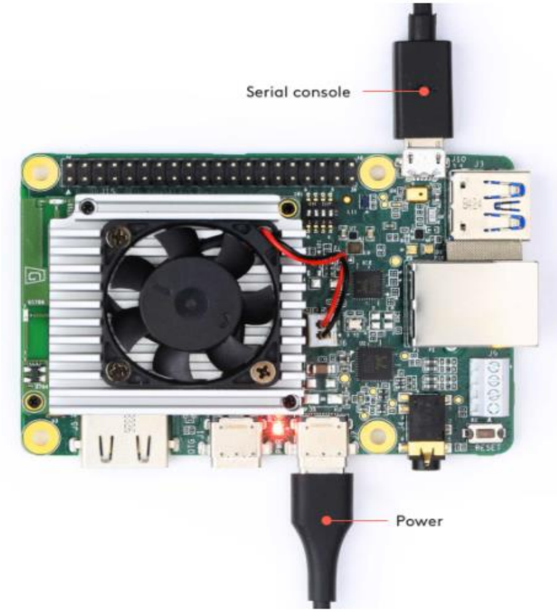
- Screen을 통해 coral에 접속한다. Terminal이 빈 화면으로 바뀔 것이다.

```
sudo screen /dev/ttyUSB0 115200
```

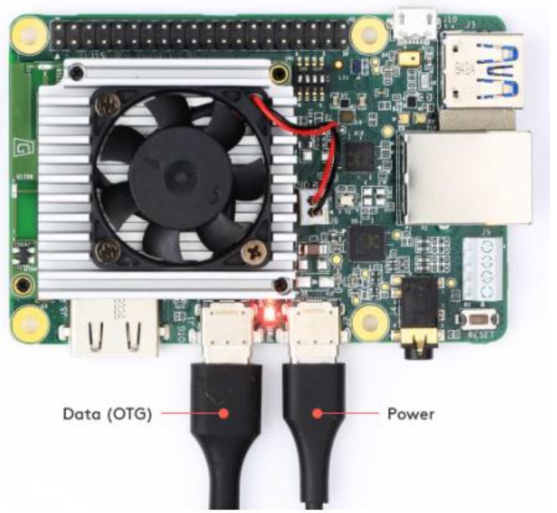
Setting Up Coral Board for the first time (Linux)

- 전원을 연결한다. **부팅 메시지가 주르륵 나온 뒤 Fastboot mode로 설정되었다고 나올 것이다.**
  - Fastboot mode 설정이 안된다면 전원 연결 후 부팅 메시지가 나오기 전에 재빨리 아무키나 입력하여 U-boot mode로 들어간 후 에서 아래와 같이 입력해준다.

```
fastboot 0
```



- 5-pin 케이블을 제거하고 USB-C 케이블을 통해 코랄과 컴퓨터를 연결한다. USB-C를 꽂는 곳이 두군데가 있으니 위치를 잘 확인하고 꽂도록 하자.3



Setting Up Coral Board for the first time (Linux)

- Fastboot가 코랄을 보고 있는지 확인한다. 아래 커맨드를 입력했을 때 뭔가 아웃풋이 있어야 한다. 아무것도 나오지 않는다면 케이블을 뺐다가 다시 꽂아보도록 하자

```
sudo fastboot devices
0b2249d6ef944da7 fastboot
```

- 아래 커맨드를 입력하여 flash 스크립트를 실행시키면 포맷 또는 최초 세팅을 시작한다. 마지막줄의 –H 옵션을 제거하면 /home 아래의 파일만 삭제한다. 약 5분 가량 소요된다.

```
cd ~/Downloads
curl -O https://mendel-linux.org/images/enterprise/eagle/enterprise-eagle-20210204152958.zip
unzip enterprise-eagle-20210204152958.zip \ && cd enterprise-eagle-20210204152958
sudo bash flash.sh -H
```

Setting Up Coral Board for the first time (Linux)

- 시작전에 mendel development tool이 설치되어 있어야 한다. 아래 커맨드로 설치 가능하다.

  ```
  pip3 install --user mendel-development-tool
  ```

- 이제 mdt를 통해 코랄에 접속할 수 있다. Mdt devices를 입력하여 기기가 뜨는지 확인해보자. 꽂고 나서 조금 기다려야 한다.

  ```
  Mdt devices
  Zippy-valet (192.168.100.2)
  ```

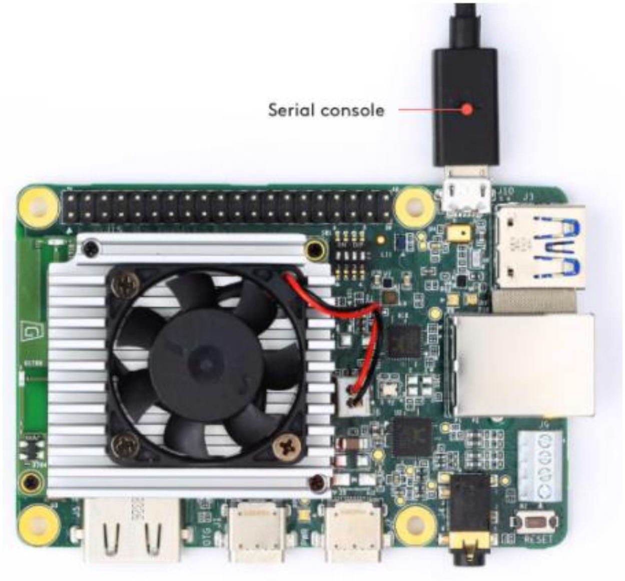
- Mendel key를 아래 파일들로 통일하도록 하자.
  - Host 컴퓨터용(~/.config/mdt/keys/mdt.key): https://drive.google.com/file/d/1KZUr9JG7XNGX4qtLWYS35eqi6fmvM6Tp/view?usp=sharing
  - 코랄용: https://drive.google.com/file/d/1TfmM2BPNJO4xxHMO9eq6eUsgkIbne9TX/view?usp=sharing

    ```
    mdt push [다운로드받은 authorized_keys 파일] /home/mendel/.ssh
    ```

Appendix2. Coral Board Re-Setting

Coral Board Re-setting (on Linux)

- 5pin짜리 케이블 통해 컴퓨터와 코랄 연결. 전원은 연결하지 않아도 된다.



- 연결이 잘 되었는지 확인. 아래 command 입력했을 때 메시지가 나와야 한다.

```
dmesg | grep ttyUSB
[ 6437.706335] usb 2-13.1: cp210x converter now attached to ttyUSB0
[ 6437.708049] usb 2-13.1: cp210x converter now attached to ttyUSB1
```

- Screen을 통해 coral에 접속한다. Terminal이 빈 화면으로 바뀔 것이다.

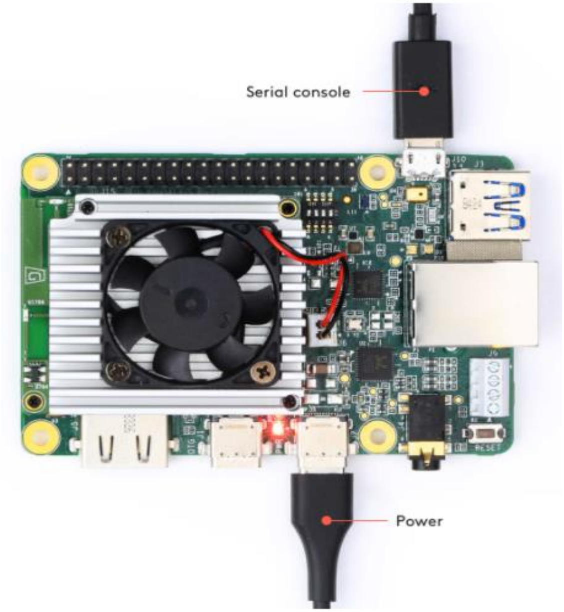
```
sudo screen /dev/ttyUSB0 115200
```

Coral Board Re-setting (on Linux)

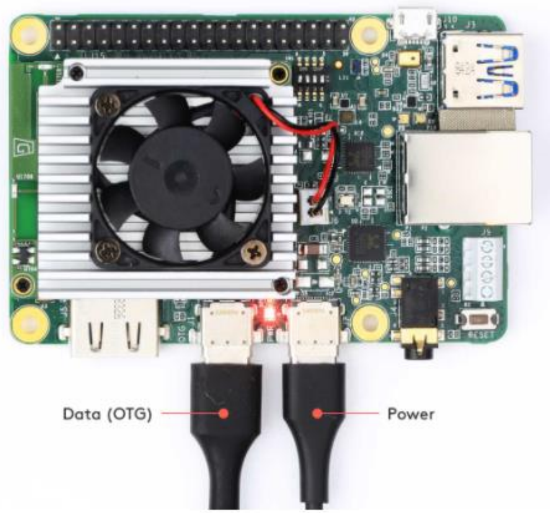
- 전원을 연결한다. 부팅 메시지가 주르륵 나올 것이다.
- Login id: mendel, pw: mendel 을 입력하여 접속 후 Key를 삭제한다.

```
rm /home/mendel/.ssh/authorized_keys
```

- ctrl A K 로 종료



- 5-pin 케이블을 제거하고 USB-C 케이블을 통해 코랄과 컴퓨터를 연결한다. USB-C를 꽂는 곳이 두군데가 있으니 위치를 잘 확인하고 꽂도록 하자.



Coral Board Re-setting (on Linux)

- Reboot bootloader를 실행한다.

```
mdt reboot-bootloader
```

- Reboot bootloader가 성공적으로 실행되었다면 fastboot mode가 활성화되었을 것이다. 아래 커맨드를 입력했을 때 뭔가 아웃풋이 있어야 한다.

```
sudo fastboot devices
0b2249d6ef944da7 fastboot
```

- 아래 커맨드를 입력하여 flash 스크립트를 실행시키면 포맷 및 재설정을 시작한다. 마지막줄의 –H 옵션을 제거하면 /home 아래의 파일만 삭제한다. 약 5분 가량 소요된다.

```
cd ~/Downloads
curl -O https://mendel-linux.org/images/enterprise/eagle/enterprise-eagle-20210204152958.zip
unzip enterprise-eagle-20210204152958.zip \ && cd enterprise-eagle-20210204152958
sudo bash flash.sh -H
```

Coral Board Re-setting (on Linux)

- 시작전에 mendel development tool이 설치되어 있어야 한다. 아래 커맨드로 설치 가능하다.

```
pip3 install --user mendel-development-tool
```

- 이제 mdt를 통해 코랄에 접속할 수 있다. Mdt devices를 입력하여 기기가 뜨는지 확인해보자. 꽂고 나서 조금 기다려야 한다. (안뜨면 계속 기다렸다가 다시 시도)

```
Mdt devices
Zippy-valet (192.168.100.2)
```

- Mendel key를 아래 파일들로 통일하도록 하자.
  - Host 컴퓨터용(~/.config/mdt/keys/mdt.key): https://drive.google.com/file/d/1KZUr9JG7XNGX4qtLWYS35eqi6fmvM6Tp/view?usp=sharing
  - 코랄용: https://drive.google.com/file/d/1TfmM2BPNJO4xxHMO9eq6eUsgkIbne9TX/view?usp=sharing

```
mdt push [다운로드받은 authorized_keys 파일] /home/mendel/.ssh
```

Appendix3. Miscellaneous

Install TensorFlow on Coral Dev Board

- swap memory 파일을 생성하여 메모리를 확보
  - sudo fallocate -l 1G /swapfile
  - sudo chmod 600 /swapfile
  - sudo mkswap /swapfile
  - sudo swapon /swapfile

- Prerequisites and Dependencies
  - sudo apt-get install -y python3 python-dev python3-dev \ build-essential libssl-dev libffi-dev \
  - libxml2-dev libxslt1-dev zlib1g-dev \ python-pip libhdf5-dev python3-h5py
  - python -m install --upgrade setuptools

- Install Tensorflow
  - wget https://github.com/lhelontra/tensorflow-on-arm/releases/download/v2.4.0/tensorflow-2.4.0-cp37-none-linux_aarch64.whl
  - (Tensorflow 버전에 따라서 url 주소를 입력한다.)
  - sudo pip3 install tensorflow-2.4.0-cp37-none-linux_aarch64.whl
  - tf.__version__를 이용하여 확인한다.

Useful Sites

- Coral 공식 사이트: https://coral.ai/
- Coral에서 사용가능한 모델들 모음: https://coral.ai/models/