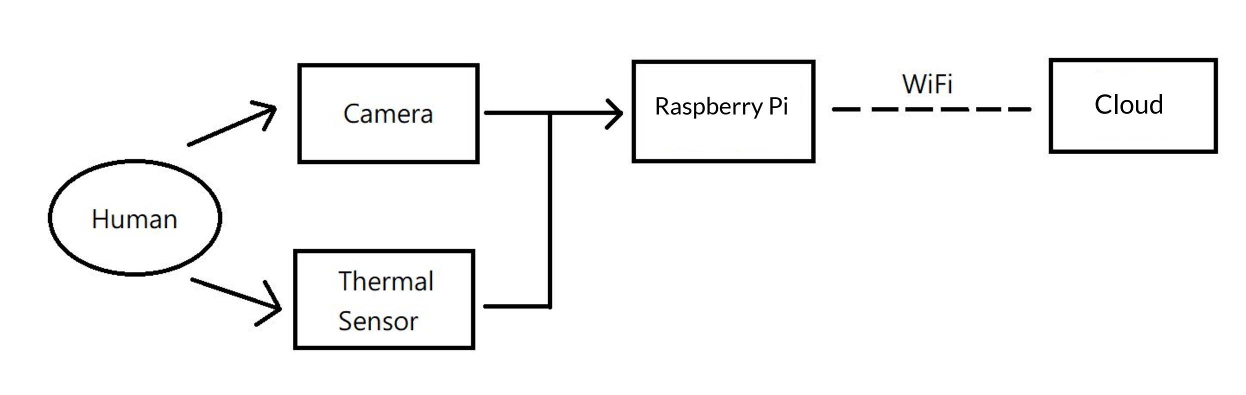
Mask Detection and Body Temperature Sensing

Cheng-Tse Lu

Overall Project Goals and Specific Aims

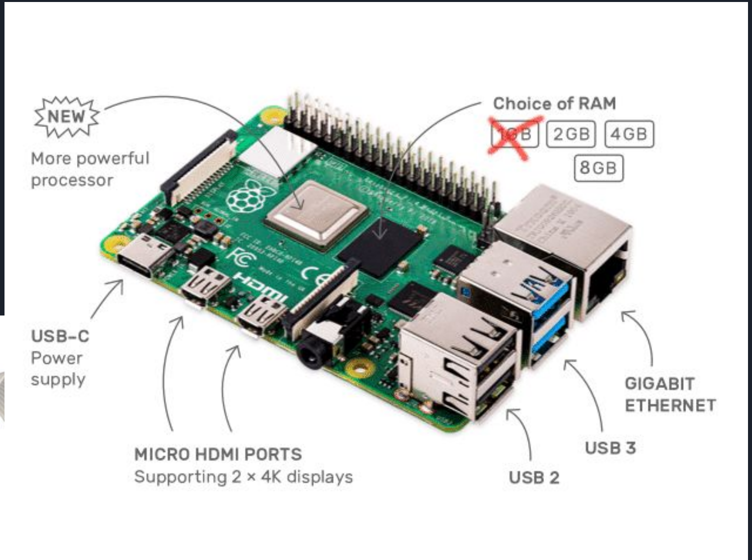
- Using deep learning algorithm for mask detection
- Incoporate a thermal sensor for non-contact temperature measurement
- Store the mask detection and the body temperature data in cloud



Technical Approach

Hardware

- Raspberry Pi 4 Model B (4GB RAM)
- Raspberry Pi camera v2.1
- GY-MLX90614-DCI
- Google Coral (edgetpu)



Mask Detection (1 / 2)

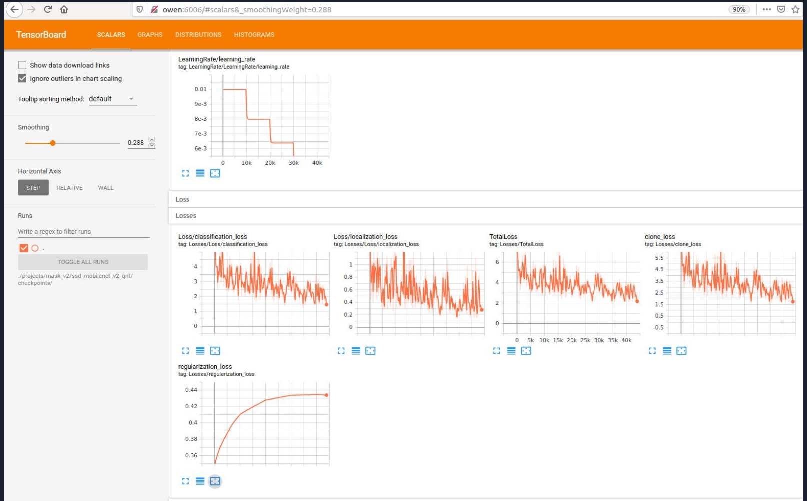
Deep Learning Algorithm

- Platform: Tensorflow 1.14 and Tensorflow-lite 1.14
- Algorithm
  - Fast: need to be real-time detection so that the processor can have time handle other tasks
  - Small size: able to run on raspberry pi since we only have 4GB RAM
  - Tensorflow Lite Compatible: a lightweight library designs for edge devices to deploy models
- Dataset: open sources dataset with 7959 images (mask and nomask)
- Training on RTX 2060
- Using Google Coral to accelerate object detection on raspberry pi

Mask Detection (2 / 2)

Training Steps (Google Object Detection API)

- Step 1: Generating TFRecords from dataset
- Step 2: Configuring training
- Step 3: Training model
- Step 4: Freeze model to .tflite
- Step 5: Convert .tflite to Edge TPU compatible model



owen:6006/#scalars&_smoothingWeight=0.288

☐ Show data download links

☑ Ignore outliers in chart scaling

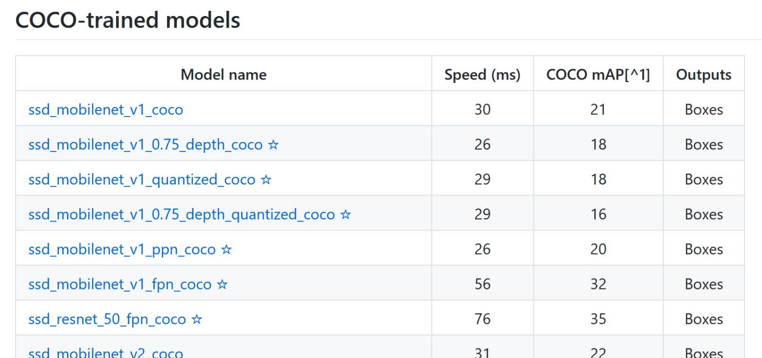
Tooltip sorting method: default

Smoothing: 0.288

Horizontal Axis: STEP

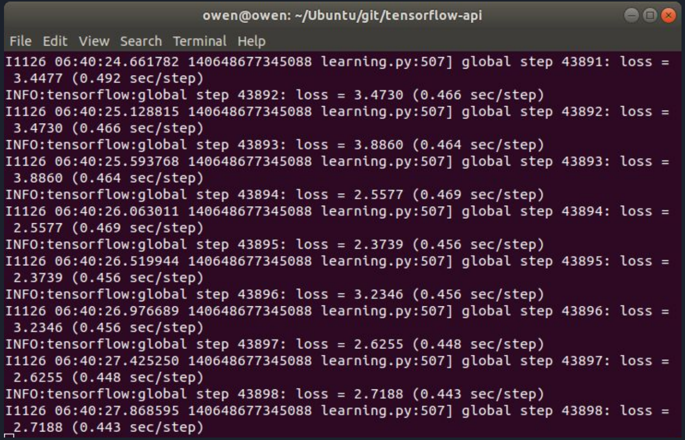
Runs

./projects/mask_v2/ssd_mobilenet_v2_qnt/checkpoints/



COCO-trained models

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes



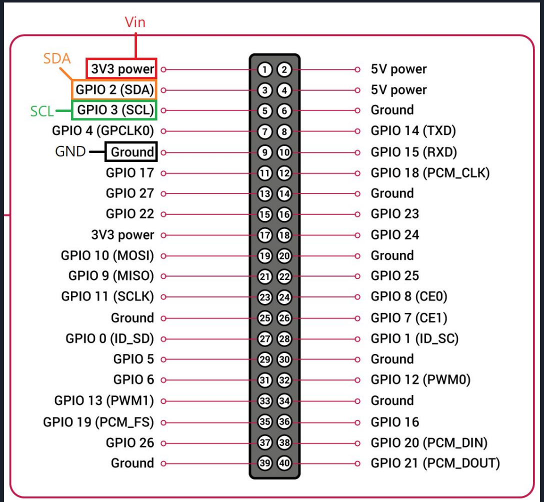
owen@owen: ~/Ubuntu/git/tensorflow-api

```
I1126 06:40:24.661782 140648677345088 learning.py:507] global step 43891: loss =
 3.4477 (0.492 sec/step)
INFO:tensorflow:global step 43892: loss = 3.4730 (0.466 sec/step)
I1126 06:40:25.128815 140648677345088 learning.py:507] global step 43892: loss =
 3.4730 (0.466 sec/step)
INFO:tensorflow:global step 43893: loss = 3.8860 (0.464 sec/step)
I1126 06:40:25.593768 140648677345088 learning.py:507] global step 43893: loss =
 3.8860 (0.464 sec/step)
INFO:tensorflow:global step 43894: loss = 2.5577 (0.469 sec/step)
I1126 06:40:26.063011 140648677345088 learning.py:507] global step 43894: loss =
 2.5577 (0.469 sec/step)
INFO:tensorflow:global step 43895: loss = 2.3739 (0.456 sec/step)
I1126 06:40:26.519944 140648677345088 learning.py:507] global step 43895: loss =
 2.3739 (0.456 sec/step)
INFO:tensorflow:global step 43896: loss = 3.2346 (0.456 sec/step)
I1126 06:40:26.976689 140648677345088 learning.py:507] global step 43896: loss =
 3.2346 (0.456 sec/step)
INFO:tensorflow:global step 43897: loss = 2.6255 (0.448 sec/step)
I1126 06:40:27.425250 140648677345088 learning.py:507] global step 43897: loss =
 2.6255 (0.448 sec/step)
INFO:tensorflow:global step 43898: loss = 2.7188 (0.443 sec/step)
I1126 06:40:27.868595 140648677345088 learning.py:507] global step 43898: loss =
 2.7188 (0.443 sec/step)
```

Body Temperature Sensing

GY-MLX90614-DCI

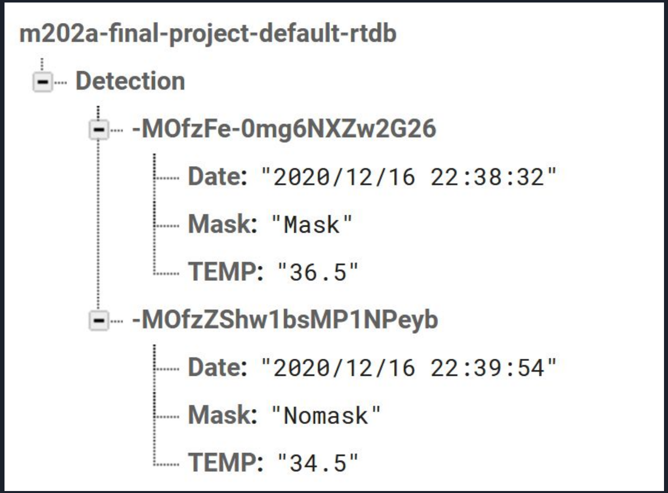
- High precision contactless temperature measurement
- Detect object temperature from -70 to +380 C
- Minimum of 0.02 C resolution
- The maximum distance measured can go up to 50 cm
- Ideal distance to measure body temperature is within 15 cm



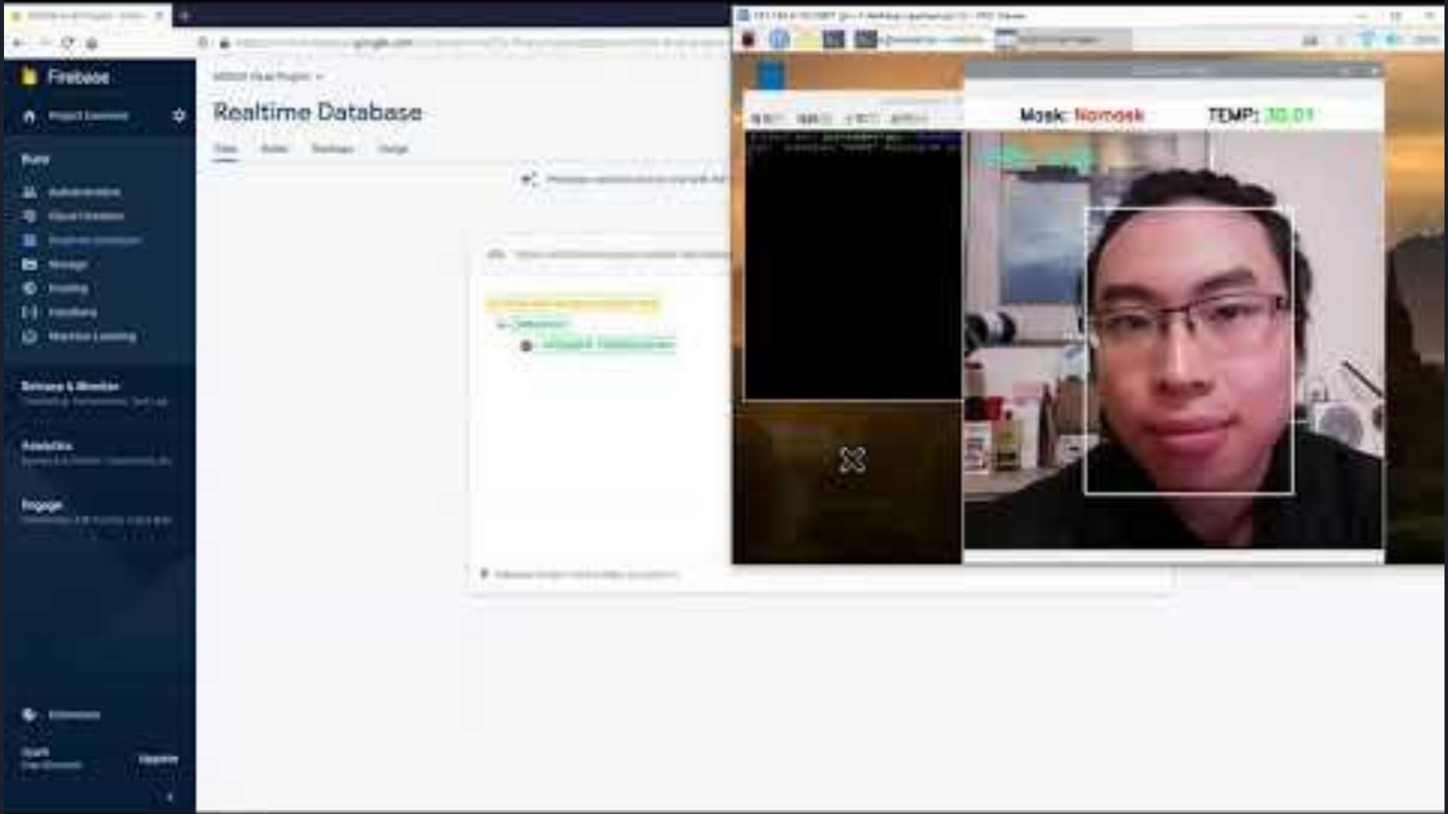
3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)

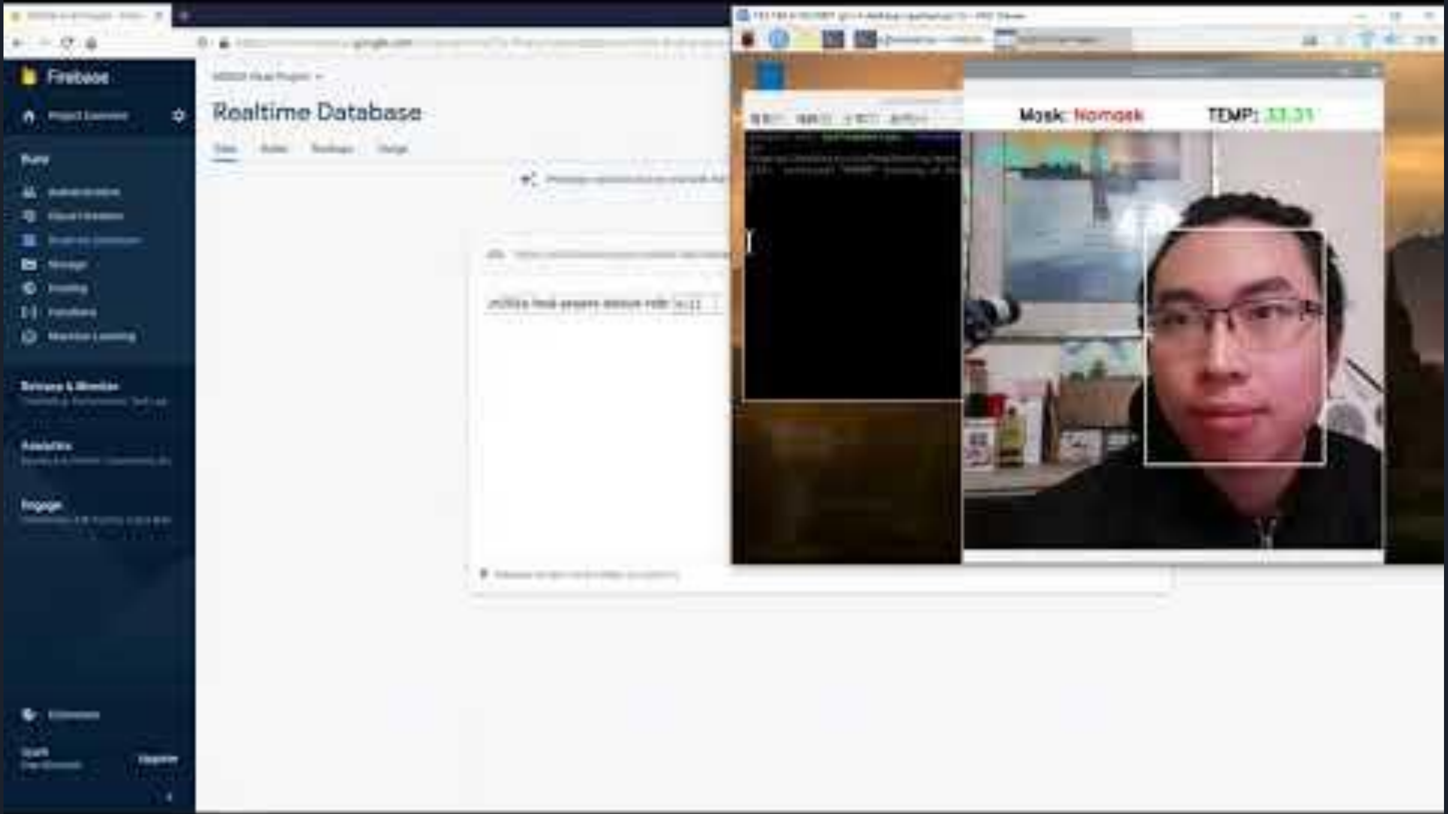
Cloud Storage

- Platform: Google Firebase database
- Able to store and sync data between users in realtime
- Data in Google Firebase is stored as JSON files
- Able to define own data structure



```
m202a-final-project-default-rtdb
 ⊟ Detection
     ⊟ -MOfzFe-0mg6NXZw2G26
         Date: "2020/12/16 22:38:32"
         Mask: "Mask"
         TEMP: "36.5"
     ⊟ -MOfzZShw1bsMP1NPeyb
         Date: "2020/12/16 22:39:54"
         Mask: "Nomask"
         TEMP: "34.5"
```


Firebase
Project Overview
Build
Authentication
Cloud Firestore
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release & Monitor
Analytics
Engage
Extensions
Spark
Upgrade
Realtime Database
Mask: Nomask
TEMP: 30.01



Firebase
Realtime Database
Release & Monitor
Analytics
Engage
Upgrade
Mask: Nomask
TEMP: 33.31

Conclusion and Future Directions

To sum up, I successfully implement a edge computing system that can detect masks, sense body temperature without contacting, and store detection data to cloud service.

Strenghts

- Fast and Accurate deep learning algorithm
- Contactless temperature measurement
- Realtime data storage to cloud

Weakness

- Short temperature measurement range
- The temperature measurement may be inaccurate if the sensor is pointing to hair instead of pointing to forehead

Links

- Github Repo: https://github.com/ChengTseLu/MaskDetectionTempSensing
- Github Page: https://chengtselu.github.io/MaskDetectionTempSensing/

Thanks!