AIY - Vision Kit test

Benjamin IOLLER

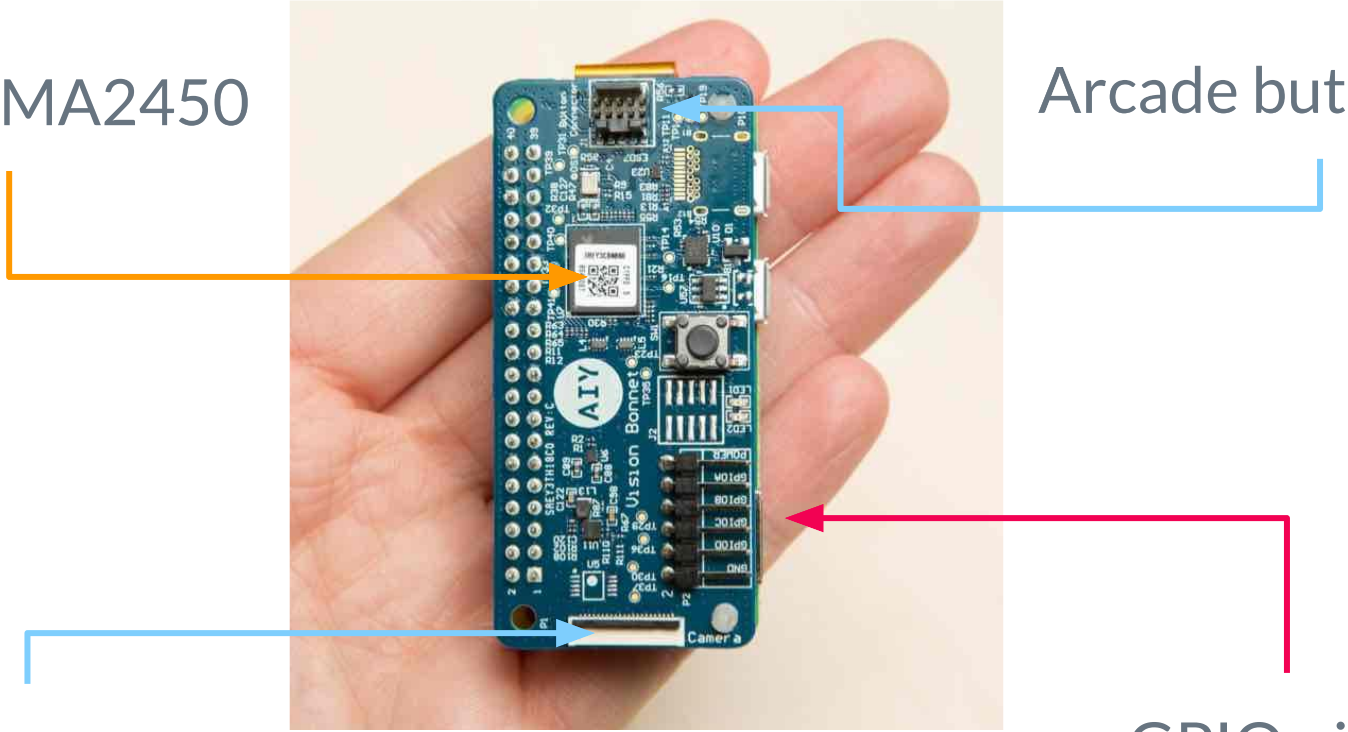
2019-02-01

EAMS Lab Co. Ltd

Table of Content

1. AIY vision presentation
2. Potential use case
3. Crack Classification demo
4. Sum-up
5. Ressources and tips

I] AIY - Vision Bonnet: RaspberryZero Shield



Movidius MA2450

Arcade button port

PiCam port

GPIO pins

I] AIY - Advantages

AI on Edge	Less latency | Privacy | No inference cost
Size	65 x 30 x 20
Weight	22.4 (without PiCam)
Low price	Kit around 90$

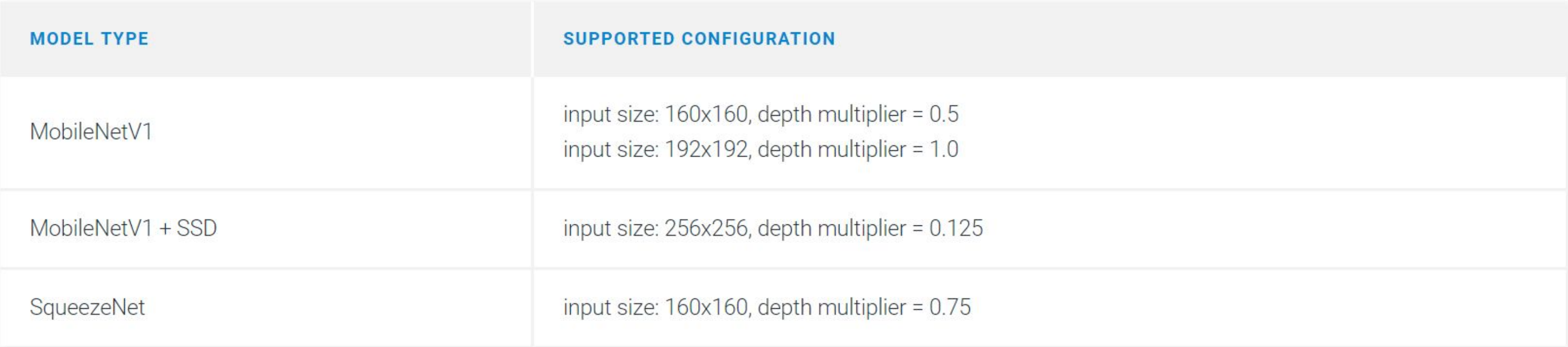
Small size and weight = easy integration:

I] AIY - Performances

1] Movibius*: SoC for image processing

- 2 Trillion 16-bit operation | 6 Mpixel second | 4K-HD cameras | Low power
- Experienced in lab: Classification (avg 20 fps) | Detection (avg 6 fps)

2] Supported Network**:



MODEL TYPE	SUPPORTED CONFIGURATION
MobileNetV1	input size: 160x160, depth multiplier = 0.5 input size: 192x192, depth multiplier = 1.0
MobileNetV1 + SSD	input size: 256x256, depth multiplier = 0.125
SqueezeNet	input size: 160x160, depth multiplier = 0.75

*http://uploads.movidius.com/1441734401-Myriad-2-product-brief.pdf **https://aiyprojects.withgoogle.com/vision/

III] Possible Use case

1] Detection-Classification

- ▷ Easy to integrate into small UVA-UGV
- ▷ Non-real time application (eg: Autonomous control based on it)

My ideas: **detection of stable**, slow changing phenomena like

Wall crack:

Fired :

2] Cloud preprocessing:

- ▷ Frame selection
- ▷ Feature extraction

Come speak to me if you have any ideas :)

III] Crack image classification - demonstration

1] Objectif

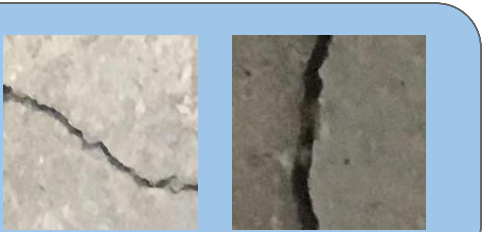
- Being able to detect crack on the wall, integrating the board on a small drone
- First quick demonstration is done with image classification, holding by hand and turning led red if crack is detected

2] Used dataset

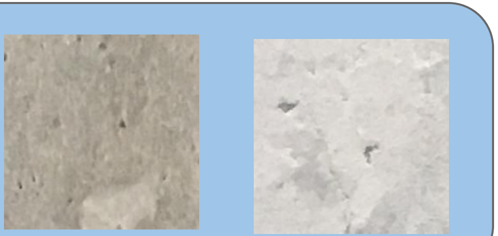
Concrete Crack Images for Classification* :

- 40000 images 227x277 RGB, half for each classes

positive:



negative:



*https://data.mendeley.com/datasets/5y9wdsg2zt/1

III] Crack image classification - demonstration

3] Network description:

I used the MobileNet V1 pretrained network with:

- INPUT: 160x160
- Depth multiplier (original network capacity): 0.5

4] Training:

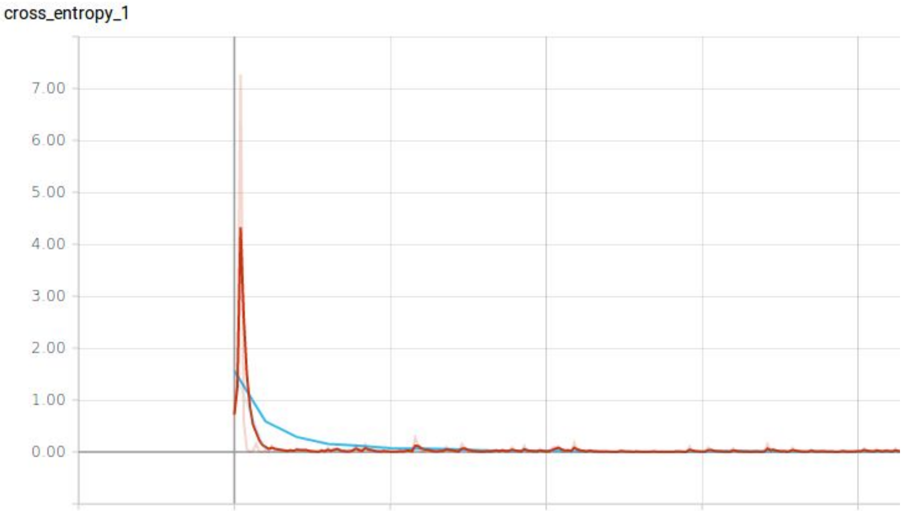
The board API is full python-tensorflow:

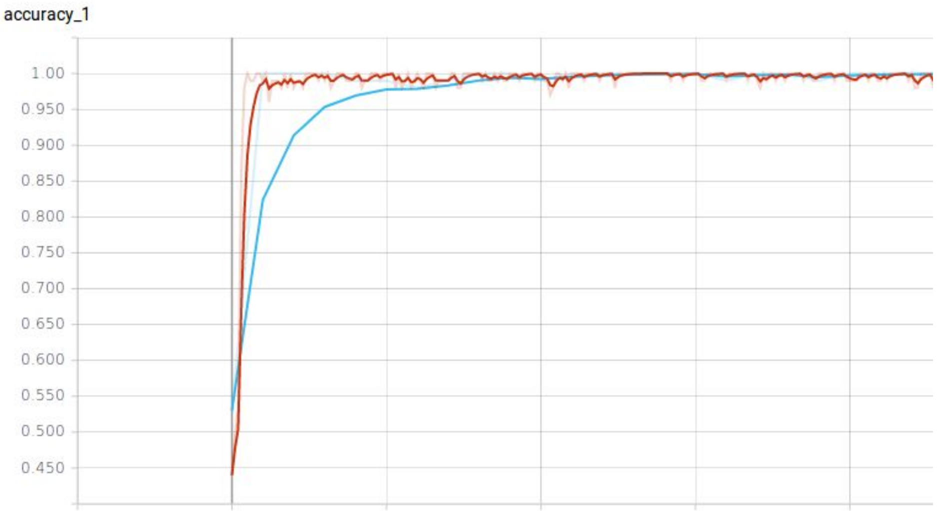
- Train your network on local machine
- Compile the graph (on Ubuntu 14.4 / 16.4)
- Upload the graph and label on the Raspberry
- Call inference and load model with aiy API

III] Crack image classification - demonstration

5] Results:

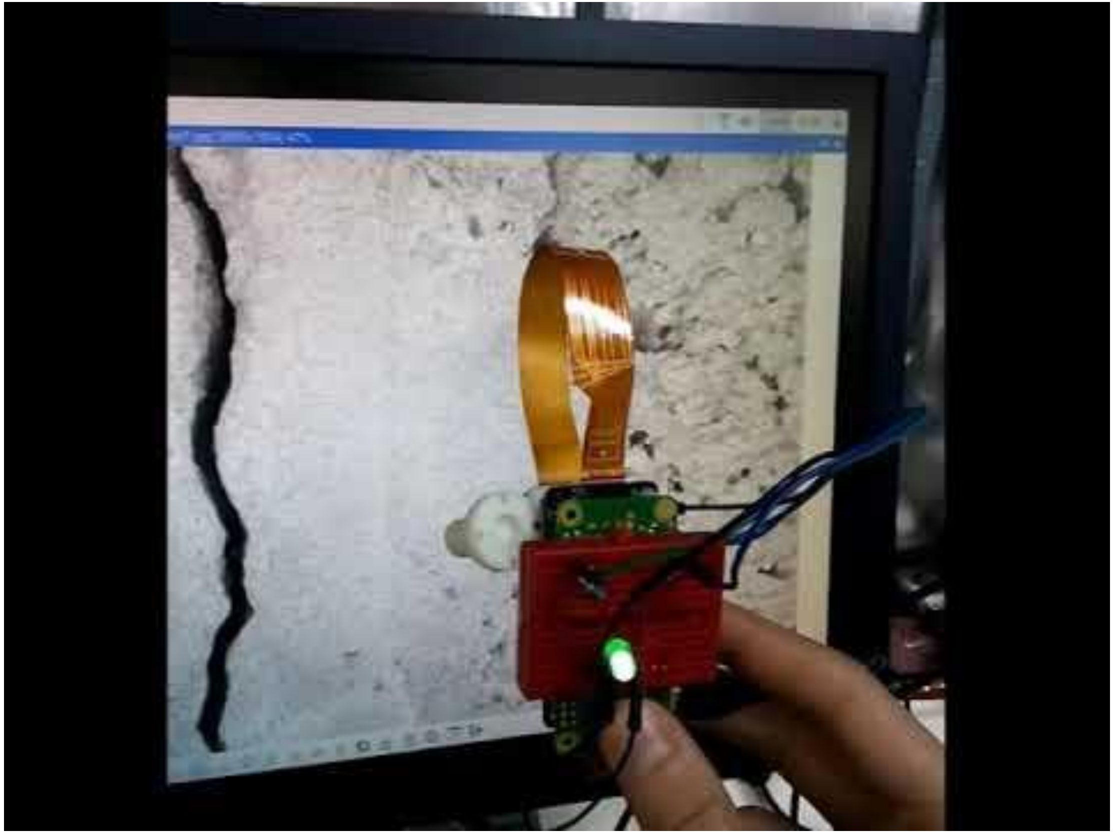
The board API is full python-tensorflow.





Classification running at average 20 fps

III] Crack image classification - demonstration



In case of video issue: https://www.youtube.com/watch?v=e4FoHp6COhM

IV] Sum-up

- Can provide a cheap Ai-on-Edge solution with easy integration on drone, small rover

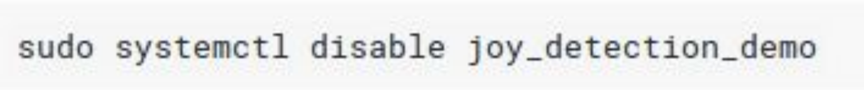
- Seems not able to tackle difficulties task like autonomous control due to hardware limitation

- Active support from Google but product is not stable

V] Ressources and Tips

- Don’t upgrade your flashed image, it seems to mess up with the spicomm module

- When flashed first reflex:



```
sudo systemctl disable joy_detection_demo
```

- Don’t try to preview the camera, it crash the pi.

V] Ressources and Tips

- https://aiyprojects.withgoogle.com/vision/#try-more-demos--stop-the-joy-detector
- https://github.com/google/aiyprojects-raspbian/tree/aiyprojects/src/examples
- https://cogint.ai/custom-vision-training-on-the-aiy-vision-kit/
- https://aiyprojects.readthedocs.io/en/latest/
- https://blog.hackster.io/teething-troubles-for-the-new-aiy-projects-vision-kit-625ed4e9287a
- https://github.com/google/aiyprojects-raspbian/releases
- https://aiyprojects.withgoogle.com/models/
- https://aiyprojects.readthedocs.io/en/latest/