Intel Neural Compute Stick 2 with Transfer Learning

Using the AWS SageMaker Transfer Learning Example
and a Raspberry Pi robot with a camera

Intel Neural Compute Stick

- Formally the Movidius Neural Compute Stick
- A Neural Net Vision Processing Unit (VPU)
- Neural Compute Engine
- - a dedicated hardware accelerator for deep neural network inferences.
- Inference, not learning
- The NCS2 uses the Intel Movidius Myriad X VPU

Why a Neural Net Computer Stick?

- Development aid for the Intel Movidius Myriad VPU's.
- The VPU's are for low power edge devices, not your desktop.
- Edge devices such as camera, drones, robots and other edge sensors.
- Specialised Inference at the edge devices allows for:
  - Quicker, low latency, decisions
  - Low bandwidth or no-bandwidth inference
  - Reduce the processing power requirements of smart edge devices.
  - No reliance on centralised servers to keep them running.
  - Sensitive information doesn't have to leave the device. Good for sensitive solutions such as children's toys and security cameras.

SageMaker

- Amazon SageMaker is a fully managed machine learning service.
- Quickly and easily build and train machine learning models, and then directly deploy them into a production-ready hosted environment.
- Provides an integrated Jupyter authoring notebook instance.
- Common machine learning algorithms
- Support for bring-your-own-algorithms and frameworks.

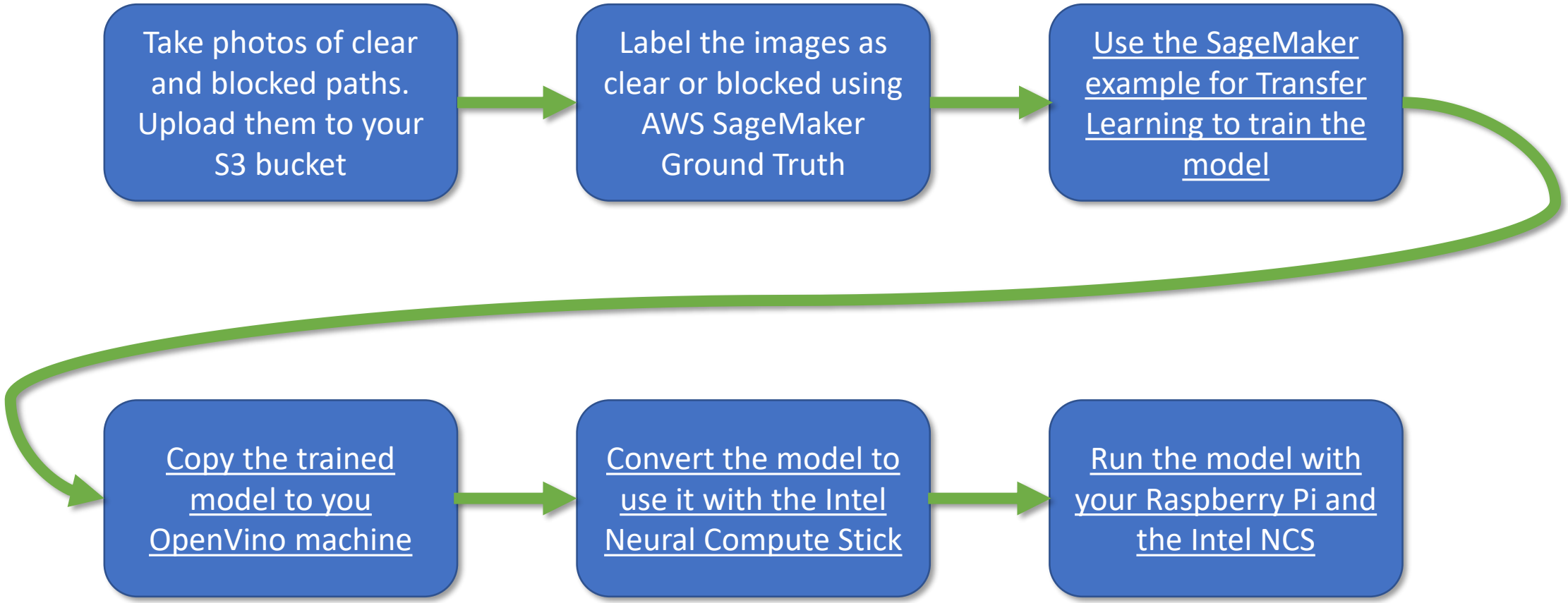
Transfer Learning

- **Transfer learning** takes a model trained for one purpose (e.g. detecting cars in a picture) and applying it to another related problem (detecting trucks).
- In this example, we will use the Caltech-256 object category dataset model
- We will use the AWS SageMaker transfer learning example
- Uses the Apache MXnet deep learning library to perform transfer learning to determine if the path in front of the robot is blocked or not.
- This is not a recommendation for this task. This is for learning the process rather than a best approach.

Getting Started

- You'll need the following to perform this process:
  - Raspberry Pi
  - Camera for the Raspberry Pi (Raspberry Pi Camera or USB Camera)
  - AWS Account for SageMaker and an S3 bucket
  - Machine with OpenVino installed. The Raspberry Pi version doesn't include the tools to convert the models. (https://software.intel.com/en-us/openvino-toolkit)
  - A Robot for the Raspberry Pi to control (if you want to test it on a robot)

Process Overview

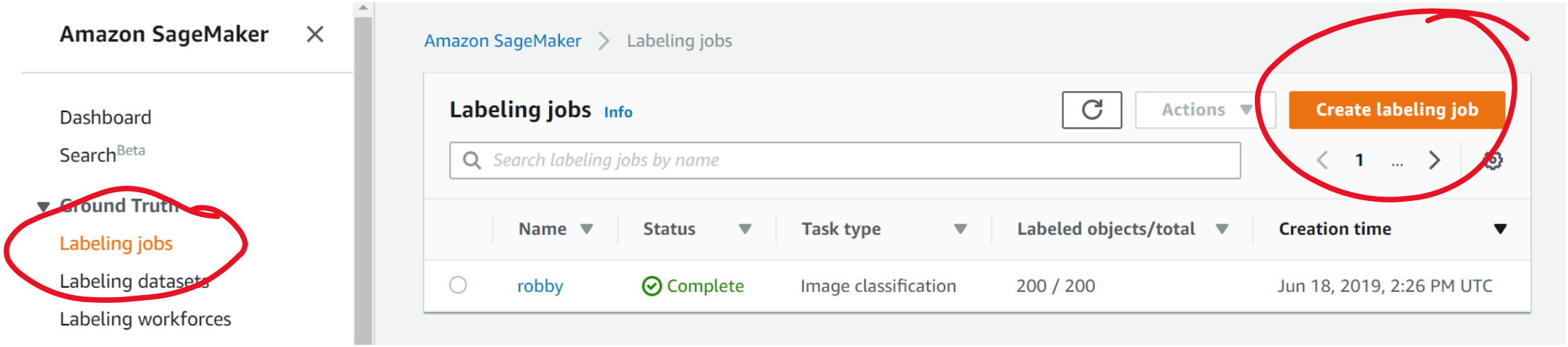


Taking lots of photos

- You'll need at least 200 photos (more is better) of blocked and free paths for the robot. You can use you mobile phone.
- The images will need to be cropped square and reduced to 224 x 224 pixels.
- Upload them to the S3 bucket you will use for training and put them in a folder called images.

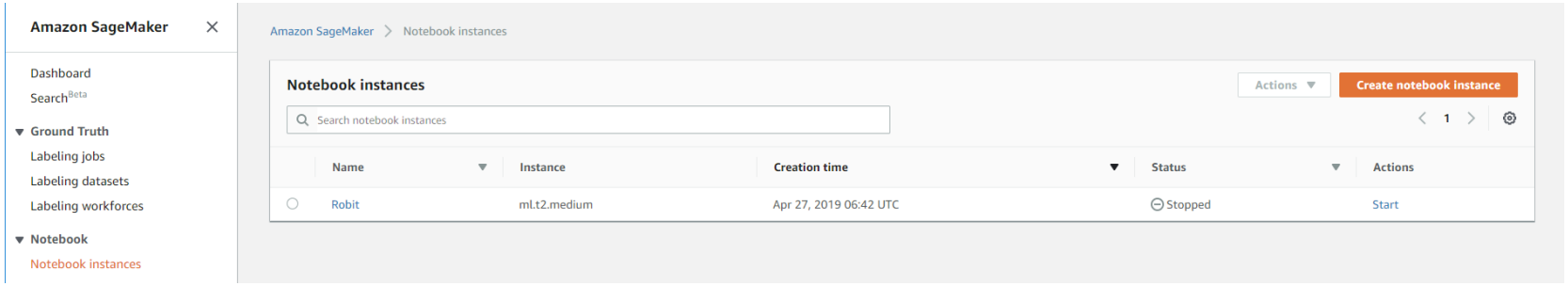
Label the images using SageMaker Ground Truth

- Open up AWS SageMaker and create a Labelling Job in Ground Truth.
- Follow the instructions to label the job.



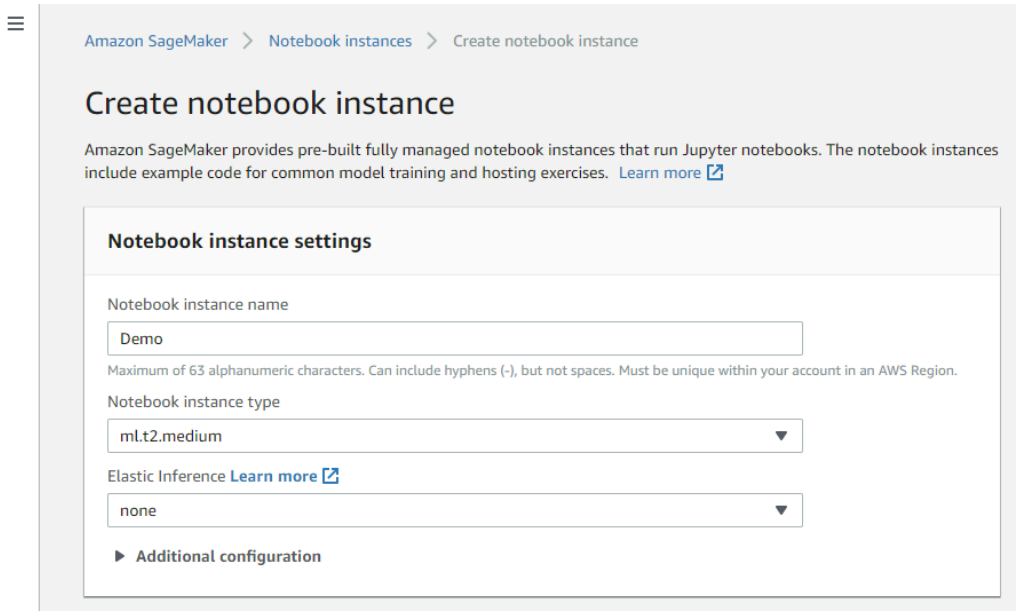
Open Notebook Instances

Click on Notebook Instances



Create a Notebook

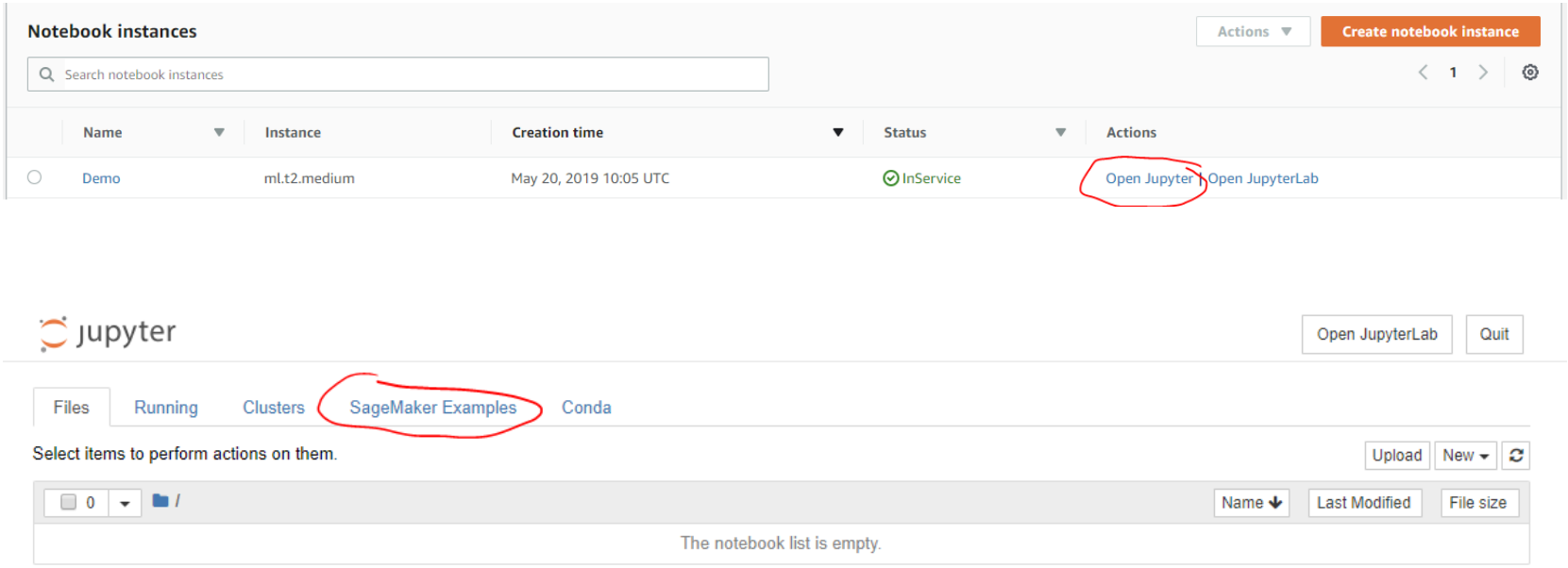
Create a new Notebook



Click **Create notebook instance** at the bottom of the page

Open a Jupyter Notebook

Open the Jupyter Notebook

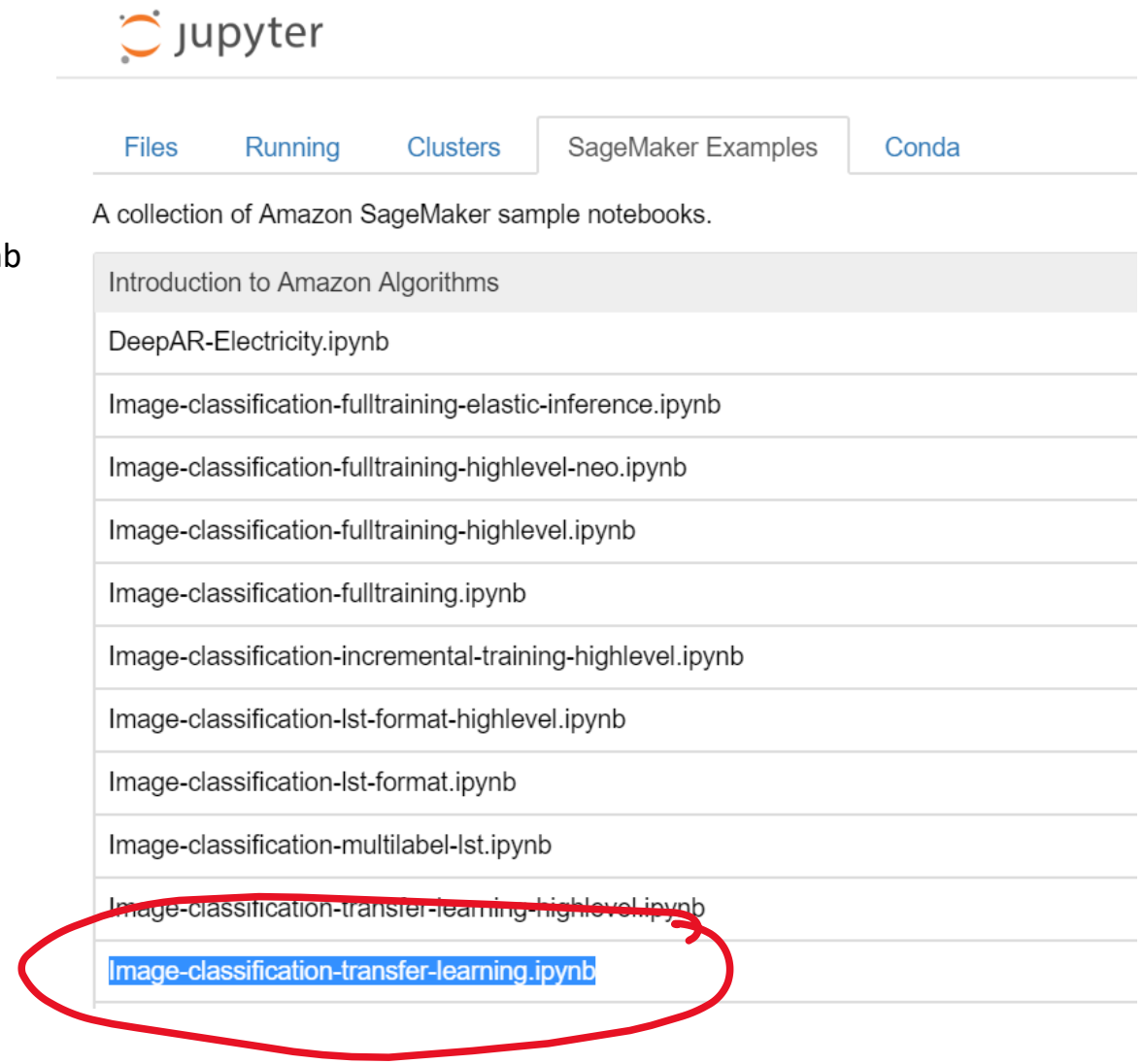


Open SageMaker Examples

Opening the Example

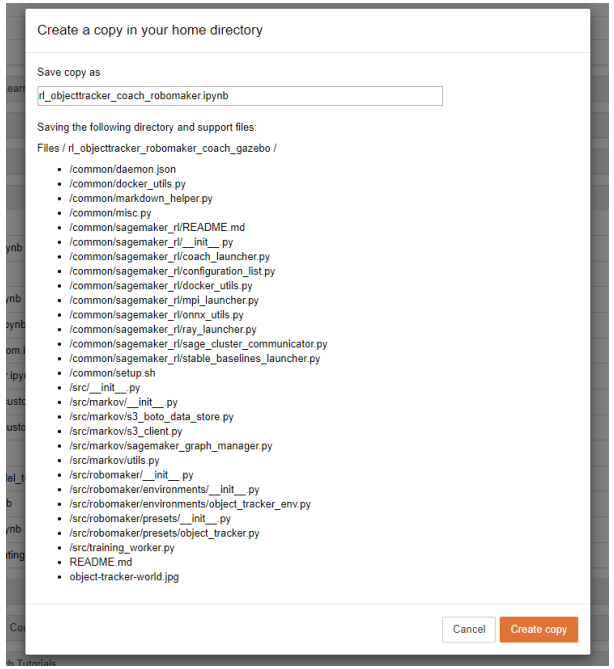
Use the example

Image-classification-transfer-learning.ipynb



Create a copy of the example

Click [Create Copy]



Run the Jupyter Notebook

- Make the following changes to the Jupyter notebook:

1. Set the bucket name to the name of your S3 bucket
   bucket='robby.robot’
2. Set the training parameters to match your sample size and number of classes (clear,blocked = 2):
   num_training_samples = 200
   num_classes = 2
   mini_batch_size = 100

- Follow the Instructions in the Jupyter Notebook:

Converting to run on the NCS

- Your model will be in a TAR compressed file in you s3 bucket at the following location except the timestamp highlighted will be different:
  DEMO-imageclassification/output/DEMO-imageclassification-2019-06-18-15-44-02/output/model.tar.gz
- Copy the model to the OpenVino machine and decompress the files. You should have three files but we only need the follow two:
  image-classification-0001.params << It may be a different number.
  image-classification-symbol.json
- Run the following command (assuming Linux install):
  python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo_mxnet.py --input_model image-classification-0001.params --input_symbol image-classification-symbol.json --input_shape [1,3,224,224] --data_type FP16
- This will give you the following two files that you need to transfer to the Raspberry Pi:
  image-classification-0001.bin
  image-classification-0001.xml

Running on the Raspberry Pi

- You need to install OpenVino for Raspberry Pi and create a python script from one of the examples to test the model inference.

https://docs.openvinotoolkit.org/latest/_docs_install_guides_installing_openvino_raspbian.html

- Run the python script with the two files you copied over as the trained model as input.

References

Suppliers of the Intel NCS2:

- RS Components: https://au.rs-online.com/web/p/processor-microcontroller-development-kits/1811851/
- Mouser: https://au.mouser.com/search/refine.aspx?Ntk=P_MarCom&Ntt=153249968

Intel OpenVino toolkit:
https://software.intel.com/en-us/openvino-toolkit

AWS SageMaker:

https://aws.amazon.com/sagemaker/

Robot used in the demonstration:

https://www.waveshare.com/wiki/AlphaBot2-Pi

Videos

Transfer Learning by Saraj Raval:

https://www.youtube.com/watch?v=Ui1KbmutX0k

NVIDIA Jetbot Neural Network Based Collision Avoidance Demo - Uses transfer learning

https://www.youtube.com/watch?v=U3VJCSDqdG4

Articles

Comparison of Edge Compute Devices

https://medium.com/@aallan/benchmarking-edge-computing-ce3f13942245