DIY SECURITY CAMERA USING INTEL® MOVIDIUS™ NEURAL COMPUTE STICK

Ashwin Vijayakumar
Lead Applications Architect, Embedded Machine Intelligence
Intel® Artificial Intelligence Products Group (AIPG)

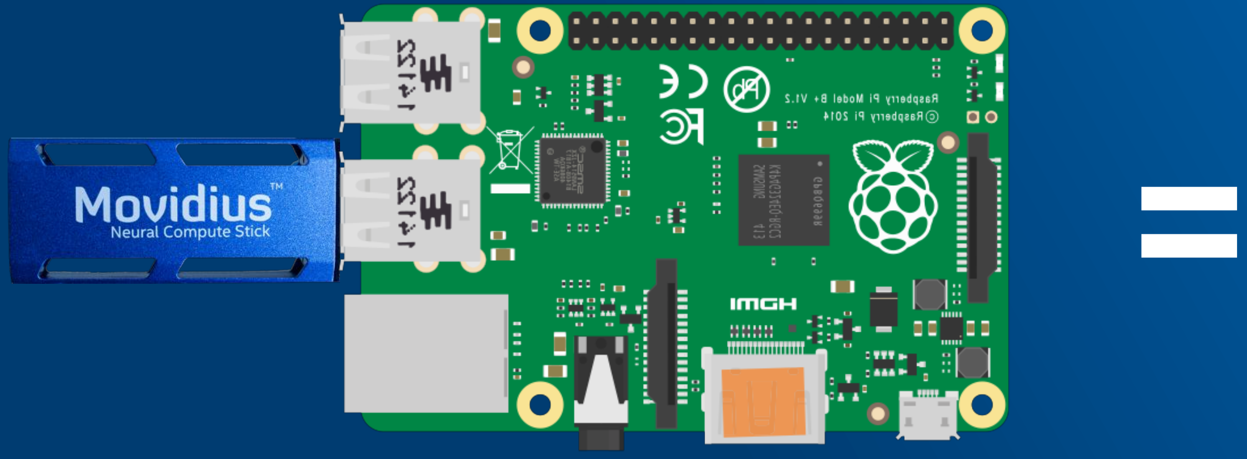
What happened to my package?

“Porch piracy” has been on the rise due to the growing trends in online purchases

- 11 million homeowners have had a package stolen in the last year
- 53% of homeowners are worried about packages left outside their home being stolen
- 74% of packages are stolen during the day when homeowners are at work
- The average value of stolen packages is $50 - $100
- Victims spend close to $200 to replace each stolen package
- 70% of all homeowners expect to receive a package over the holiday season
- FedEx and UPS delivered more than 30 million packages a day between Black Friday and Christmas Day 2016

Data source: https://www.thepackageguard.com/wp-content/uploads/2017/05/DataSheat_PackageTheftDeliveryindustrystats-1.pdf
*Other names and brands may be claimed as the property of others

DIY Porch Pirate Buster!



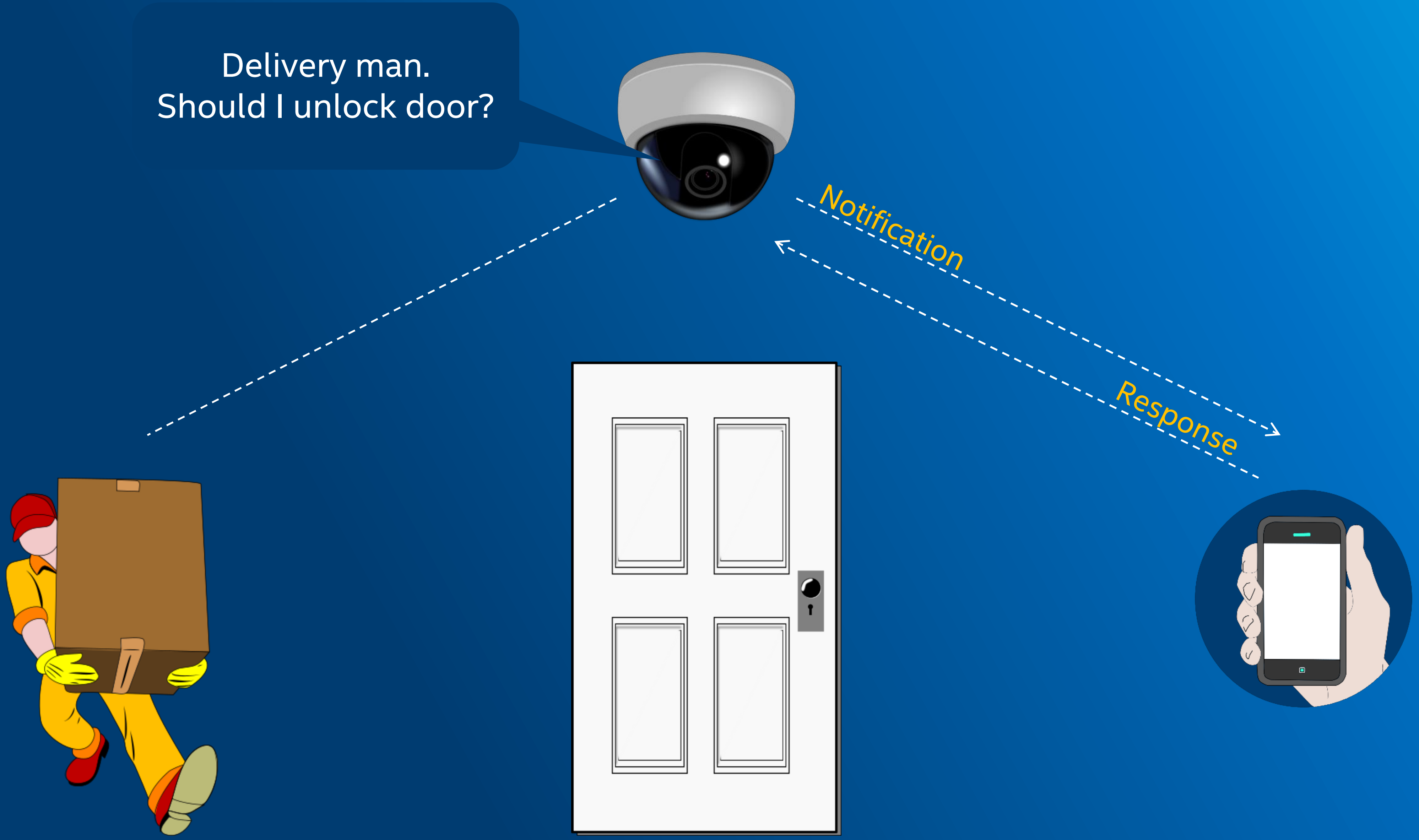
Raspberry Pi image source: https://upload.wikimedia.org/wikipedia/commons/2/26/Raspberry_Pi_B%2B_illustration.svg
Pirate image source: https://commons.wikimedia.org/wiki/File%3APiratey%2C_vector_version.svg

Agenda

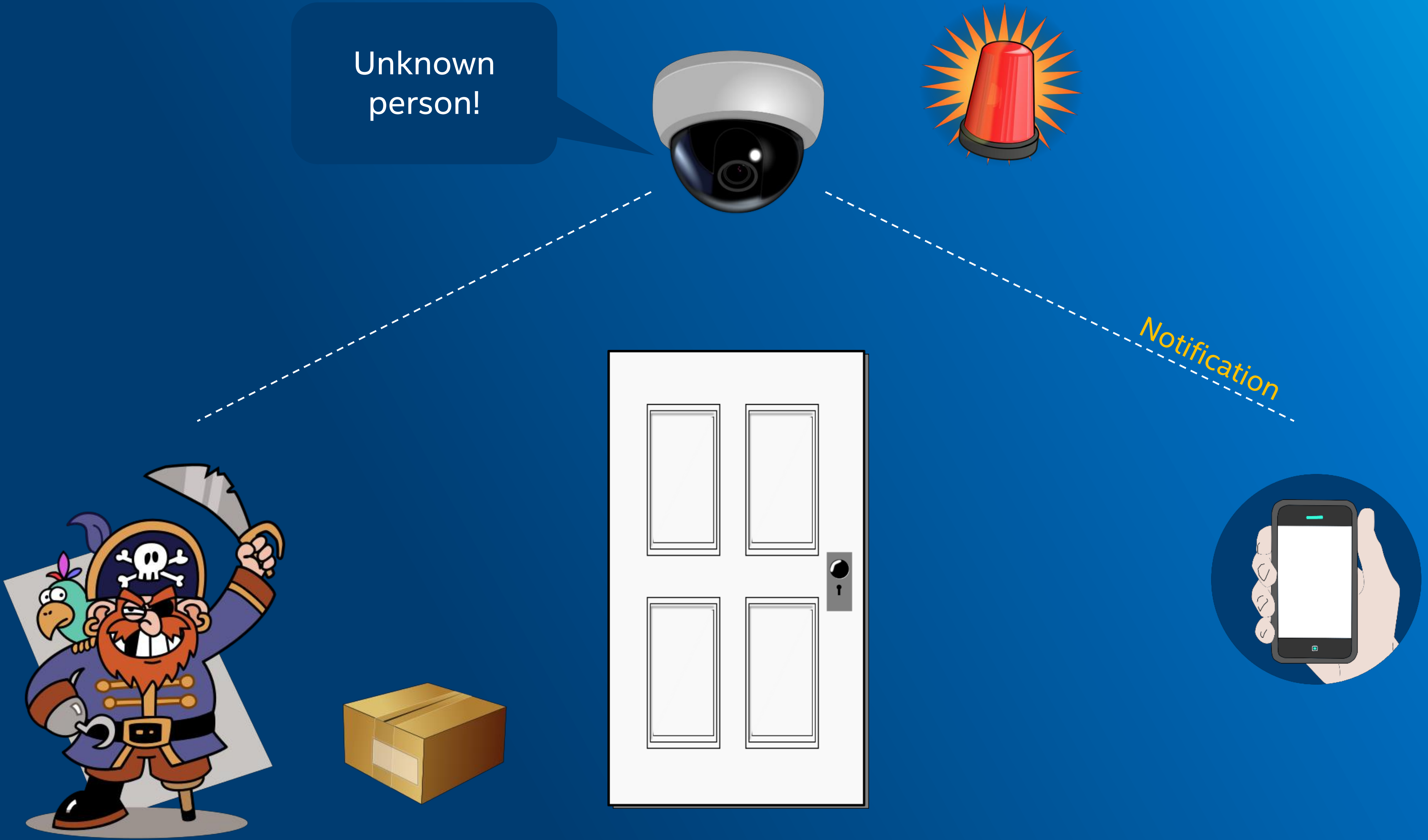
- Final outcome: Build a DIY security camera
- Section 1 [0:15]: Requirements gathering
- Section 2 [0:30]: Getting started with the Intel® Movidius™ Neural Compute Stick
- Section 3 [1:30]: Prototype the security cam application
- Section 4 [2:30]: Deploy the application onto an embedded platform
- Section 5 [3:20]: Wrap-up, Q&A

Section 1: Requirements gathering

What do you think a “porch pirate buster” should be able to do?



Image(s) source: https://publicdomainvectors.org/



Unknown person!

Notification

Image(s) source: https://publicdomainvectors.org/

Group exercise

Take 5 mins to build a list of requirements

Great list of requirements, but let’s be realistic!

Filter the requirements to the most basic features, so it can fit into a 3.5 hour hands-on session

Person detected.
Notify user.

Simplified requirements

1. Small enough to fit on/near a door
2. Low-power consumption
3. Low-cost hardware requirements
4. Detects a ‘person’ in real-time

Section 2: Getting started

With the Intel® Movidius™ Neural Compute Stick

Deploy Caffe Model on Intel® Movidius™ NCS

Reference: https://github.com/movidius/ncappzoo/tree/master/caffe/GoogLeNet

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/caffe/GoogLeNet
5. make run

Deploy TensorFlow* Model on Intel® Movidius™ NCS

Reference: https://github.com/movidius/ncappzoo/tree/master/tensorflow/inception

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/tensorflow/inception
5. make run

*Other names and brands may be claimed as the property of others

Deploy MobileNets Models on Intel® Movidius™ NCS

Reference: https://github.com/movidius/ncappzoo/tree/master/tensorflow/mobilenets

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/tensorflow/mobilenets
5. make run
   - Repeat the exercise with different depth multipliers and input image sizes
   - make run DEPTH=0.75 IMGSIZE=192
   - make run DEPTH=0.50 IMGSIZE=160
   - make run DEPTH=0.25 IMGSIZE=128

What did you observe from this exercise?

HINT: Think about the “real-time” aspect of our requirements

Pick a model that best suits our "real-time" requirement

MobileNets are mobile-friendly neural networks

Network	Inference time	Frames per second
GoogLeNet	91.33099 ms	10.9 fps
Inception-v3	325.1207 ms	3.07 fps
MobileNet (1.0 | 224)	39.26307 ms	25.4 fps

Configurations:
Hardware: Intel Movidius Neural Compute Stick and a laptop based on Intel Core i5-6600 CPU @ 3.3GHz, 32GB RAM
Software: Ubuntu 16.04 + NCSDK 1.12
Test code: FPS numbers were generated using sample codes in https://github.com/movidius/ncappzoo, which is released to the public under MIT license.
[1] https://github.com/movidius/ncappzoo/tree/master/caffe/GoogLeNet
[2] https://github.com/movidius/ncappzoo/tree/master/tensorflow/inception
[3] https://github.com/movidius/ncappzoo/tree/master/tensorflow/mobilenets

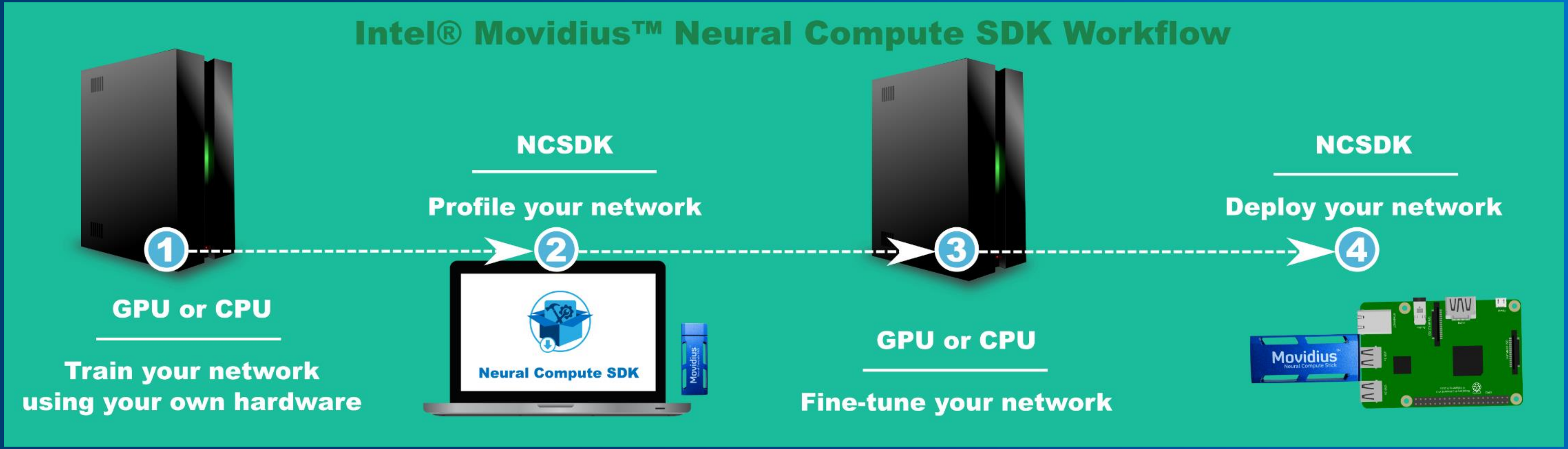
Let’s detect a ‘person’

HINT: Use image-classifier.py --image workspace/ncappzoo/apps/security-cam/captures/screen_shot.jpg

Key learnings from this section

Take a moment to look into ncappzoo/tensorflow/mobilenets/Makefile

- Frames per second is a key requirement to build a real-time computer vision product
- Image classifier models do not meet our “person detection” requirement
- Basic workflow of deploying neural networks on Intel Movidius NCS



Run object detection on your Intel® Movidius™ NCS

Embedded application development using NC SDK API framework

- Step 1: Open the enumerated device and get a handle to it
- Step 2: Load a graph file onto the NCS device
- Step 3: Pre-process the images
- Step 4: Offload an image to the NCS to run inference
- Step 5: Close the NCS device

HINT: See ~/workspace/ncappzoo/apps/ssd-object-detector

Key learnings from this code walk-through

Take a moment to look into ssd-object-detector/utils/deserialize_output.py

`ssd-object-detector` formats NCS output into an easy-to-use python dictionary

- output_dict['num_detections'] = Total number of valid detections
- output_dict['detection_classes_<X>'] = Class ID of the detected object
- output_dict['detection_scores_<X>'] = Percentage of the confidance
- output_dict['detection_boxes_<X>'] = A list of 2 tuples [(x1, y1) (x2, y2)]

Where <X> is a zero-index count of num_detections

Section 3: Prototype security cam app

Modify the code to only detect person

HINT: Retraining is out-of-scope, so apply `if statements` on output_dict['detection_classes_<X>’]

Run inference on a live camera feed

HINT: See ncappzoo/apps/live-object-detector

Run inference on a live camera feed

Real-time object detection using Intel Movidius NCS

- Step 1: Open the enumerated device and get a handle to it
- Step 2: Load a graph file onto the NCS device
- Step 3: Pre-process the images (includes grabbing frames from a camera)
- Step 4: Offload an image to the NCS to run inference
- Step 5: Close the NCS device

HINT: See ~/workspace/ncappzoo/apps/live-object-detector

Develop my-security-cam

Modify live-object-detector to only detect and capture person(s)

HINT: See ~/workspace/ncappzoo/apps/security-cam

Section 4: Deploy on embedded board

Assemble the Raspberry Pi system

Deploy my-security-cam onto an embedded board

Ensure the dev machine and embedded board are on the same WiFi network

- Step 1: Clone https://github.com/movidius/ncsdk on your embedded board
- Step 2: Clone https://github.com/movidius/ncappzoo on your embedded board
- Step 3: Copy the MobileNet SSD graph and your app from the dev machine to the embedded board
  - scp ~/workspace/ncappzoo/caffe/SSD_MobileNet/graph ncs@raspberrypi-2:~/workshop/ncappzoo/caffe/SSD_MobileNet/
  - scp -r ~/workspace/ncappzoo/apps/my-security-cam ncs@raspberrypi-2:~/workshop/ncappzoo/apps/my-security-cam
- Step 4: Run the app on your embedded board
  - ssh ncs@192.168.1.[YOUR BOARD NUMBER]
  - cd ~/workspace/ncappzoo/apps/my-security-cam
  - python3 security-picam.py
- View images captured by your embedded security camera
  - scp ncs@raspberrypi-2:~/workshop/ncappzoo/apps/my-security-cam/captures ~/Pictures

Feedback

Scan this QR code to fill out our feedback form

The Intel® Movidius™ NCS is a product designed specifically for developers and data scientists. Our goal with these workshops and competitions is to listen to developer pain points first hand, and incorporate your feedback into upcoming software and hardware revisions.

https://goo.gl/SyopsX

Questions?

0x51 0x75 0x65 0x73 0x74 0x69 0x6F 0x6E 0x73 0x3F

Legal Disclaimer

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at https://developer.movidius.com.

Intel, the Intel logo, and Movidius are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Configurations and Workloads

Hardware: Intel Movidius Neural Compute Stick and a laptop based on Intel Core i5-6600 CPU @ 3.3GHz, 32GB RAM

Software: Ubuntu 16.04 + NCSDK 1.12

Test code: FPS numbers were generated using sample codes in https://github.com/movidius/ncappzoo, which is released to the public under MIT license.

experience
what's inside™

Backups

Feedback

Scan this QR code to fill out our feedback form

The Intel® Movidius™ NCS is a product designed specifically for developers and data scientists. Our goal with these workshops and competitions is to listen to developer pain points first hand, and incorporate your feedback into upcoming software and hardware revisions.

https://goo.gl/WMTnjJ

Feedback [China]

Scan this QR code to fill out our feedback form

The Intel® Movidius™ NCS is a product designed specifically for developers and data scientists. Our goal with these workshops and competitions is to listen to developer pain points first hand, and incorporate your feedback into upcoming software and hardware revisions.