



# Intel® Movidius™ Neural Compute Developer Tools

Theme: I am network architect / data scientist

Justin Shenk  
Intel Software Innovator

# Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

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

Copyright © 2017, Intel Corporation. All rights reserved.

# Agenda

9:00 AM: Welcome and setup

9:15 AM: Introduction to Intel® Movidius™ Neural Compute Stick (NCS)

9:30 AM: How to run pre-trained neural networks on NCS [Hands-on session]

9:45 AM: Deep dive into NCS developer tools [Hands-on session]

12:00 PM: Lunch

1:00 PM: Open hands-on session

3:00 PM: Break

5:00 PM: Wrap-up

# Intel® Movidius™ technology

# Game-changing intelligent devices

Powered by Movidius VPU



Hikvision  
Intelligent Camera



Hikvision  
Industrial Camera



DJI Inspire 2



DJI  
Phantom 4 Pro



DJI Mavic Pro



Uniview  
IP Camera



Dahua  
Industrial Camera



Moto 360°  
Camera

# Movidius Neural Compute Stick

Redefining the AI developer kit



- Neural Network Accelerator in USB Stick Form Factor
- No additional heat-sink, no fan, no cables, no additional power supply
- Prototype, tune, validate and deploy deep neural networks at the edge
- Features the same Movidius vision processing unit (VPU) used in drones, surveillance cameras, VR headsets, and other low-power intelligent and autonomous products

# Movidius Neural Compute Stick

Redefining the AI developer kit



NC SDK

Free download @ [developer.movidius.com](http://developer.movidius.com)

NC Toolkit

Profiler

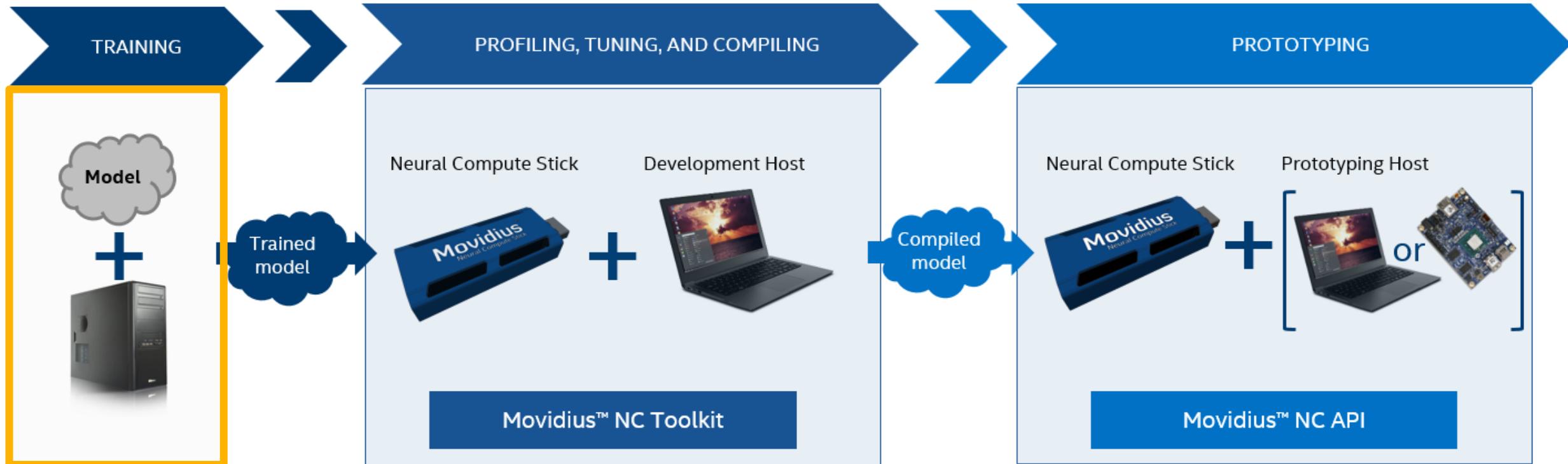
Checker

Compiler

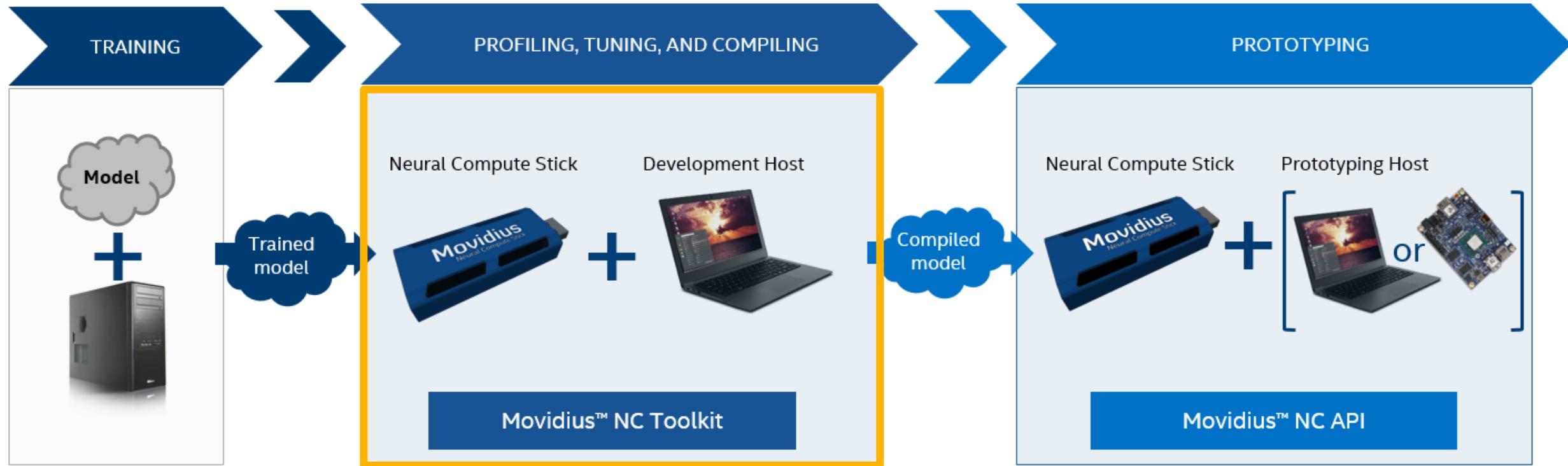
NC API

API

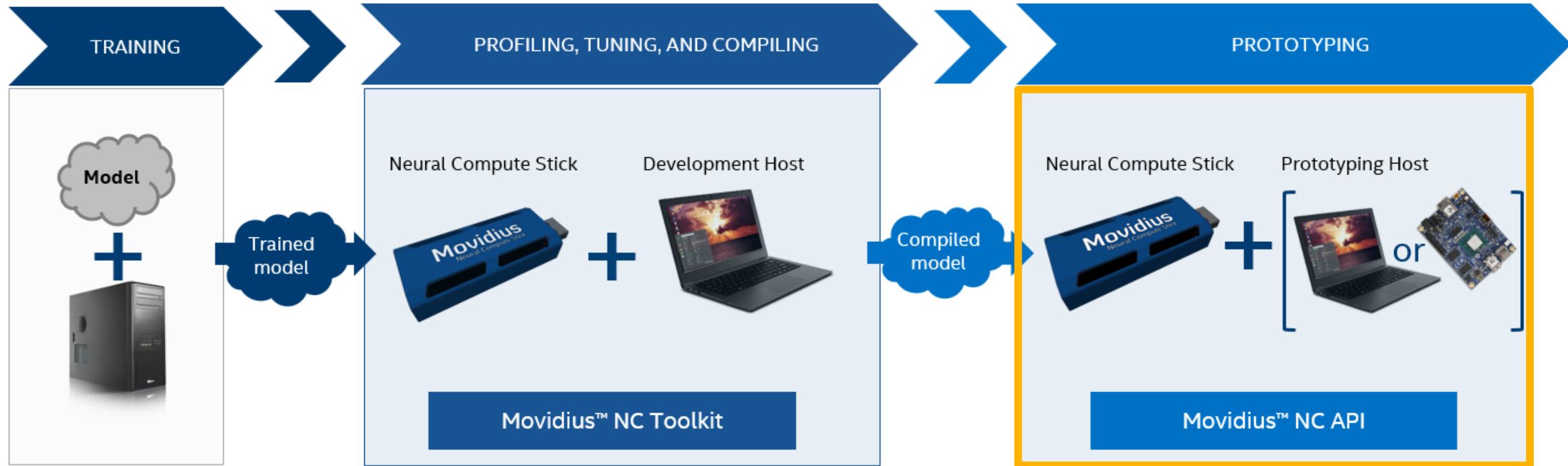
# NC SDK workflow



# NC SDK workflow



# NC SDK workflow



# Hands-on exercises

# Explore developer.movidius.com

A developer-friendly website

Try out the following pages:

- Start
- Github (ncsdk, ncappzoo, blogs)
- **Support > Docs**
- Support > Forums
- Where to buy

The screenshot shows the developer.movidius.com website. At the top, there's a navigation bar with links for HOME (which is highlighted in orange), GET STARTED, FORUM, DOWNLOADS, and WHERE TO BUY. The main title is "Movidius™ Neural Compute Stick" with the subtitle "Accelerate deep learning development at the edge". Below this, there's a "WHERE TO BUY" button. The central content area has two main sections: "What is the Neural Compute Stick?" which describes it as a tiny fanless deep learning device powered by a low-power high-performance Movidius™ Vision Processing Unit (VPU) found in various smart security cameras, drones, and industrial equipment; and "What can you do with the NCS?", which discusses its use for rapid prototyping, validation, and deployment of Deep Neural Network (DNN) inference applications at the edge. Both sections include images: the first shows the physical blue stick, and the second shows a bowl of fruit with colored bounding boxes around them.

# Running a pre-trained neural network on NCS

# Run a simple image classifier

Reference: <https://movidius.github.io/blog/ncs-image-classifier/>

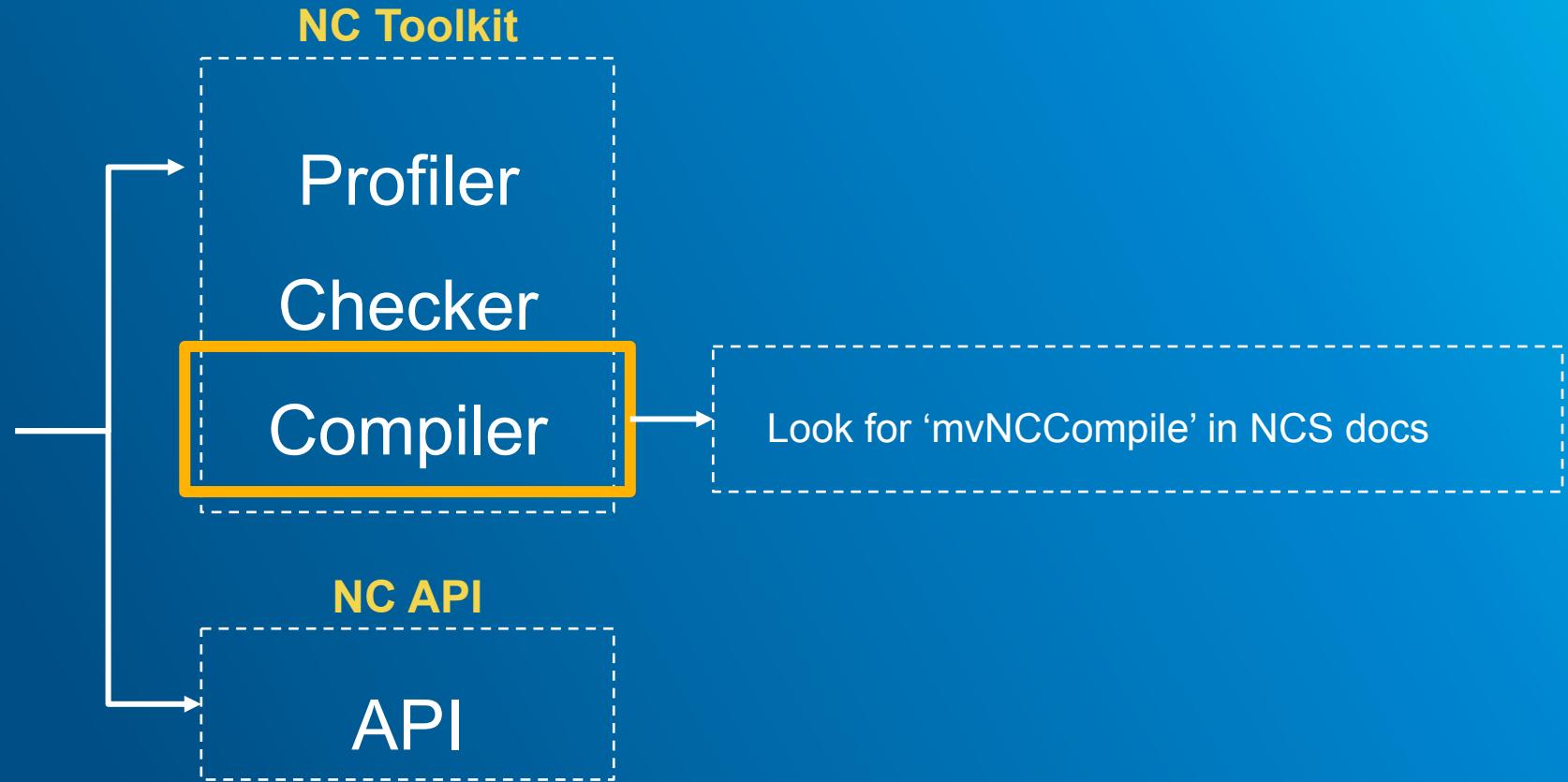
1. `mkdir -p ~/workspace`
2. `cd ~/workspace`
3. `git clone https://github.com/movidius/ncappzoo`
4. `cd ~/workspace/ncappzoo/apps/image-classifier`
5. `make run`

# NC SDK Compile

Convert your network into a binary graph file that can be loaded onto the NCS



**NC SDK**



# Compile GoogLeNet caffe model to NCS graph file

Reference: developer.movidius.com > Support > Docs > search for “mvNCCCompile”

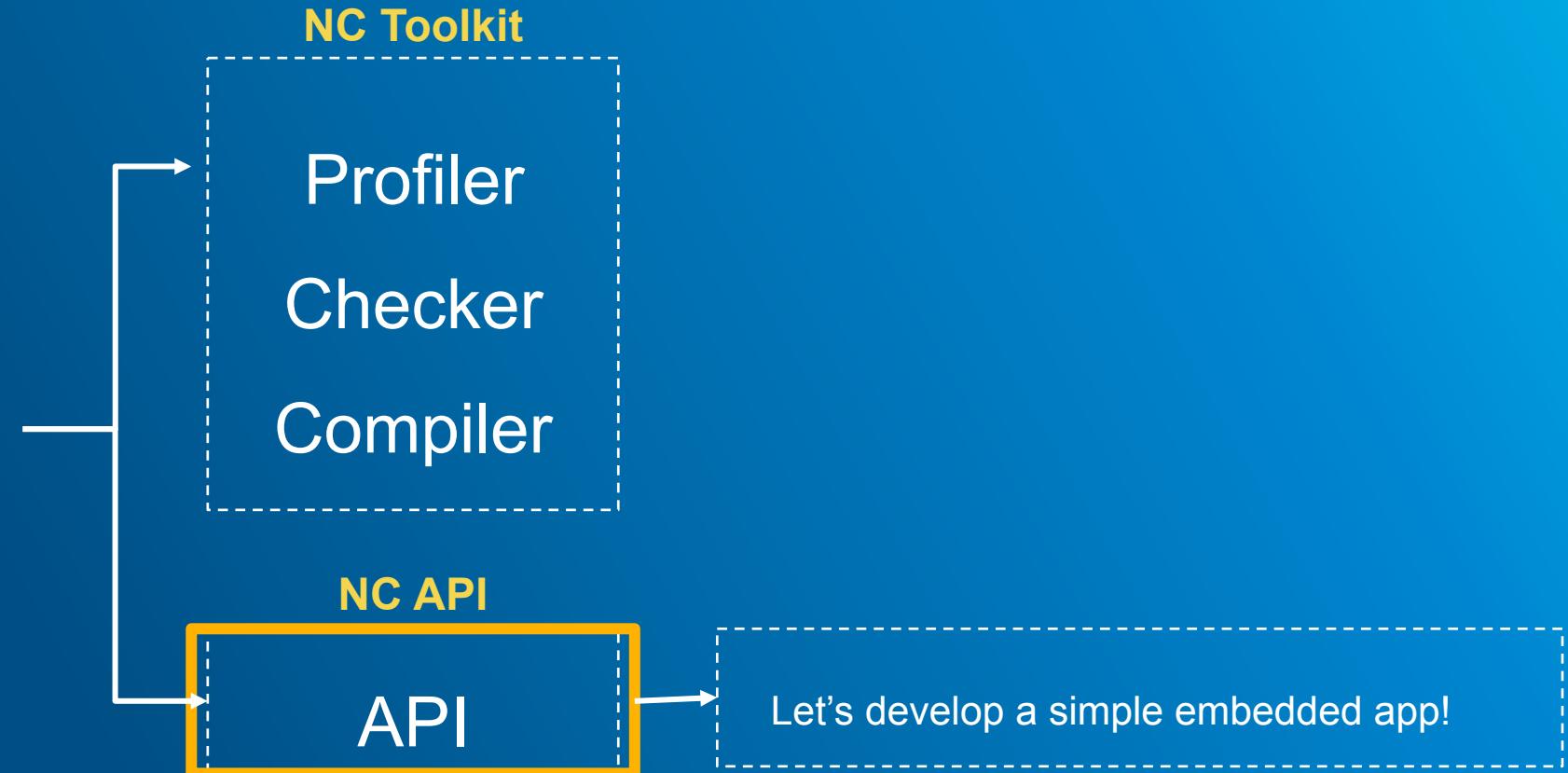
1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/caffe/GoogLeNet
5. rm graph
6. mvNCCCompile -s 12 deploy.prototxt -w bvlc\_googlenet.caffemodel
7. Look for ~/workspace/ncappzoo/caffe/GoogLeNet/graph

# NC SDK API framework

Develop your own embedded application with deep-learning accelerated image processing



NC SDK



# Code walkthrough: image-classifier.py

Reference: <https://movidius.github.io/blog/ncs-image-classifier/>

# Deep dive into NCS developer tools

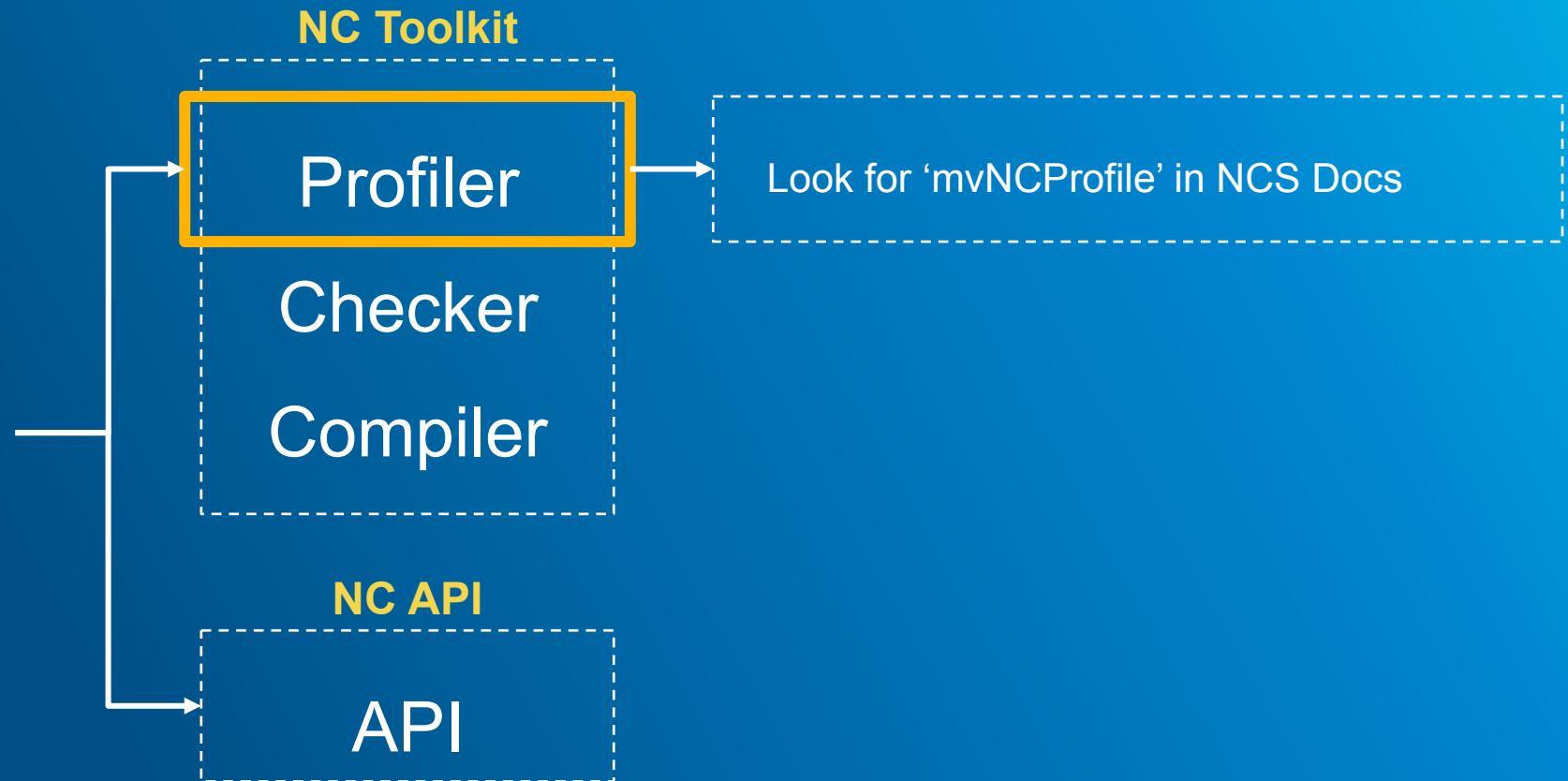
**mvNCProfile: Tool to help optimize your neural network for edge applications**

# NC SDK Profiler

Get a better insight into your network's complexity, bandwidth & execution time



**NC SDK**



# Run GoogLeNet through NCS profiler

Reference: developer.movidius.com > Support > Docs > search for “mvNCProfile”

1. mkdir -p ~/workspace
2. cd ~/workspace
3. git clone https://github.com/movidius/ncappzoo
4. cd ~/workspace/ncappzoo/caffe/GoogLeNet
5. mvNCProfile -s 12 deploy.prototxt -w bvlc\_googlenet.caffemodel
6. Look for ~/workspace/ncappzoo/caffe/GoogLeNet/output\_report.html

# Win \$20,000 by building a embedded image classifier

Scan this QR code to register for the challenge



In this challenge you will be pushing your network training skills to its limits by fine-tuning CNNs that are targeted for embedded applications. Contestants are expected to leverage the Neural Compute SDK's mvNCProfile tool to analyze the bandwidth, execution time and complexity of their network at each layer, and tune it to get the best accuracy, execution time

[developer.movidius.com/competition](http://developer.movidius.com/competition)

# Modifying GoogLeNet to run faster on NCS

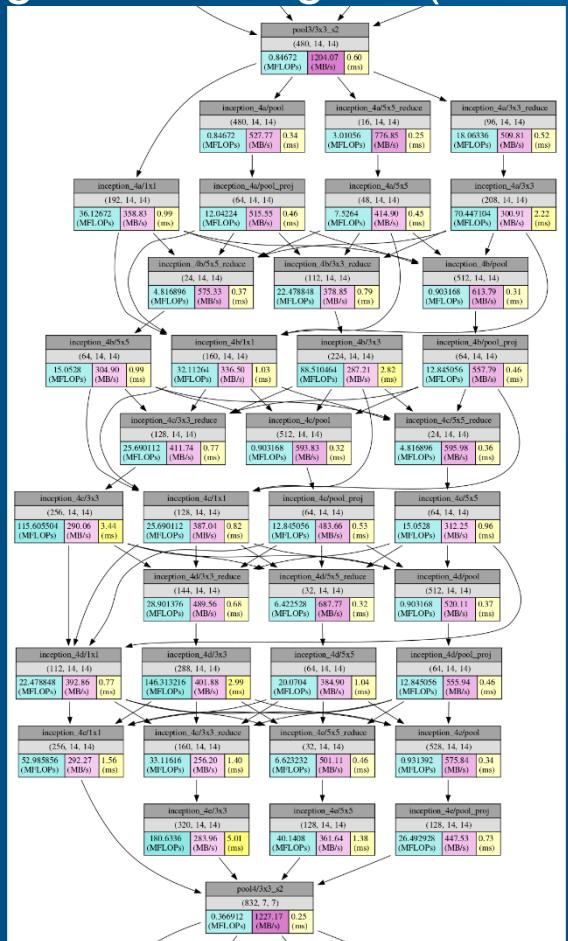
An example project that explores the entire NCS workflow of training-profiling-deploying

# Modifying GoogLeNet to run faster on NCS

What we modified, why we modified it, how did it impact execution time?

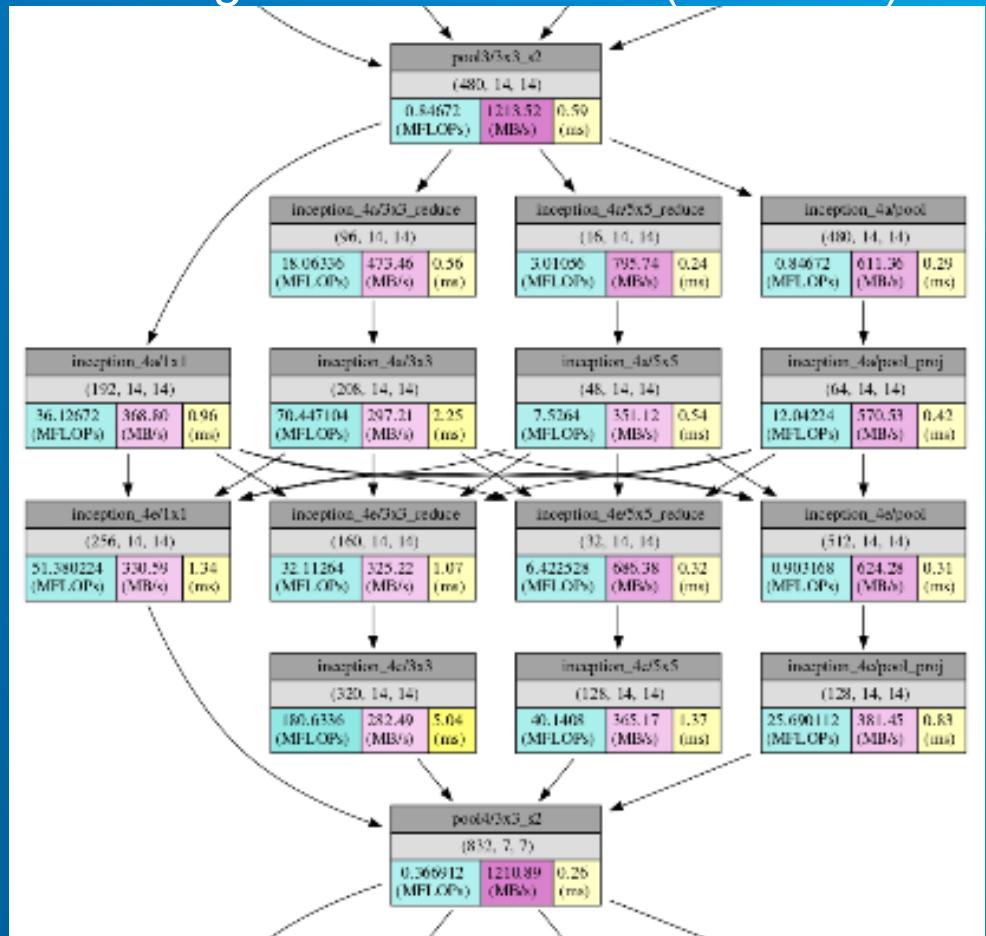
GoogLeNet – original (94.28ms)

4a →  
4b →  
4c →  
4d →  
4e →



GoogLeNet – modified (73.03ms)

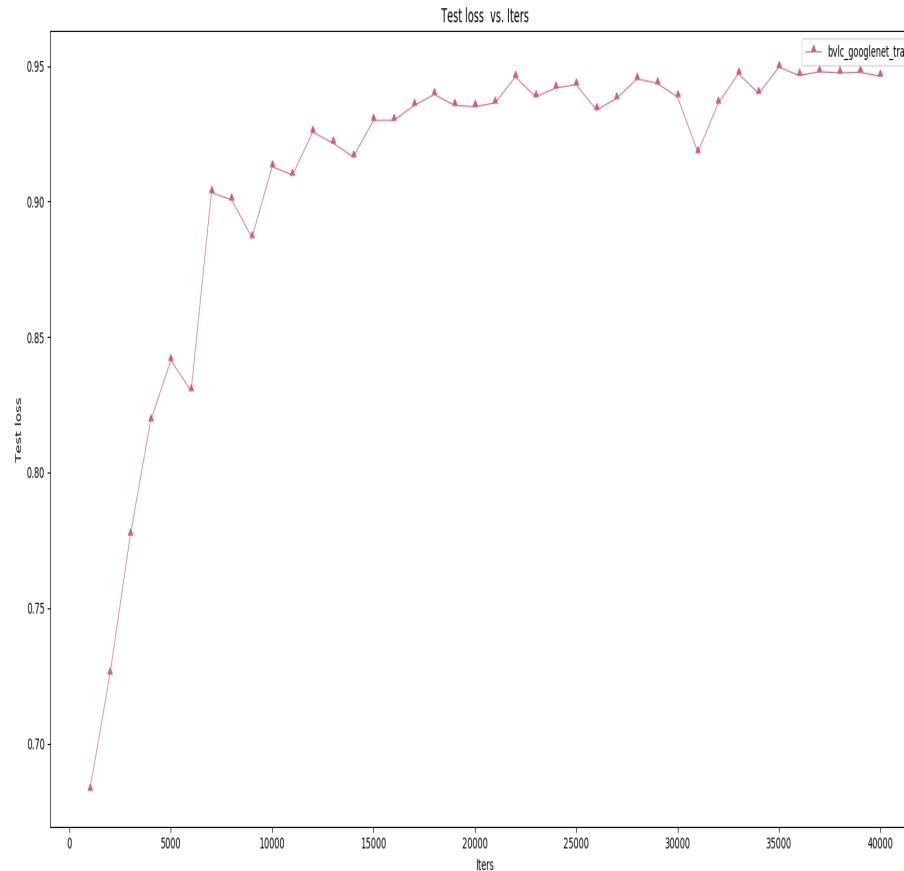
4a →  
4e →



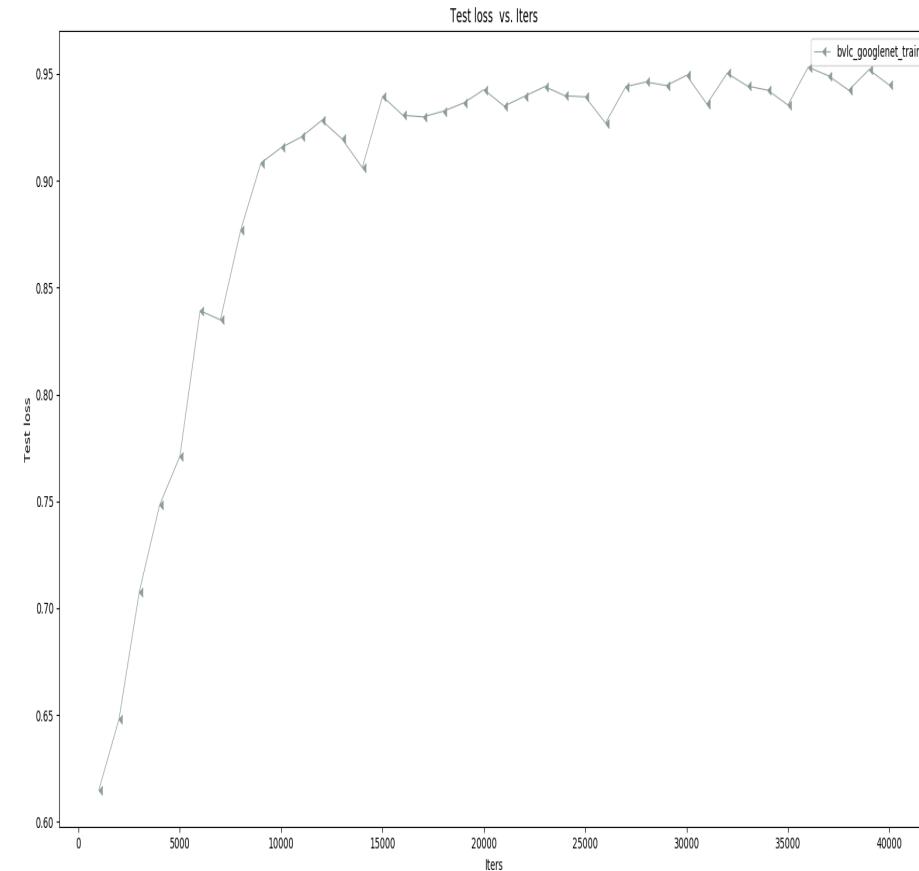
# Modifying GoogLeNet to run faster on NCS

Was there an impact on accuracy?

GoogLeNet - original



GoogLeNet - modified

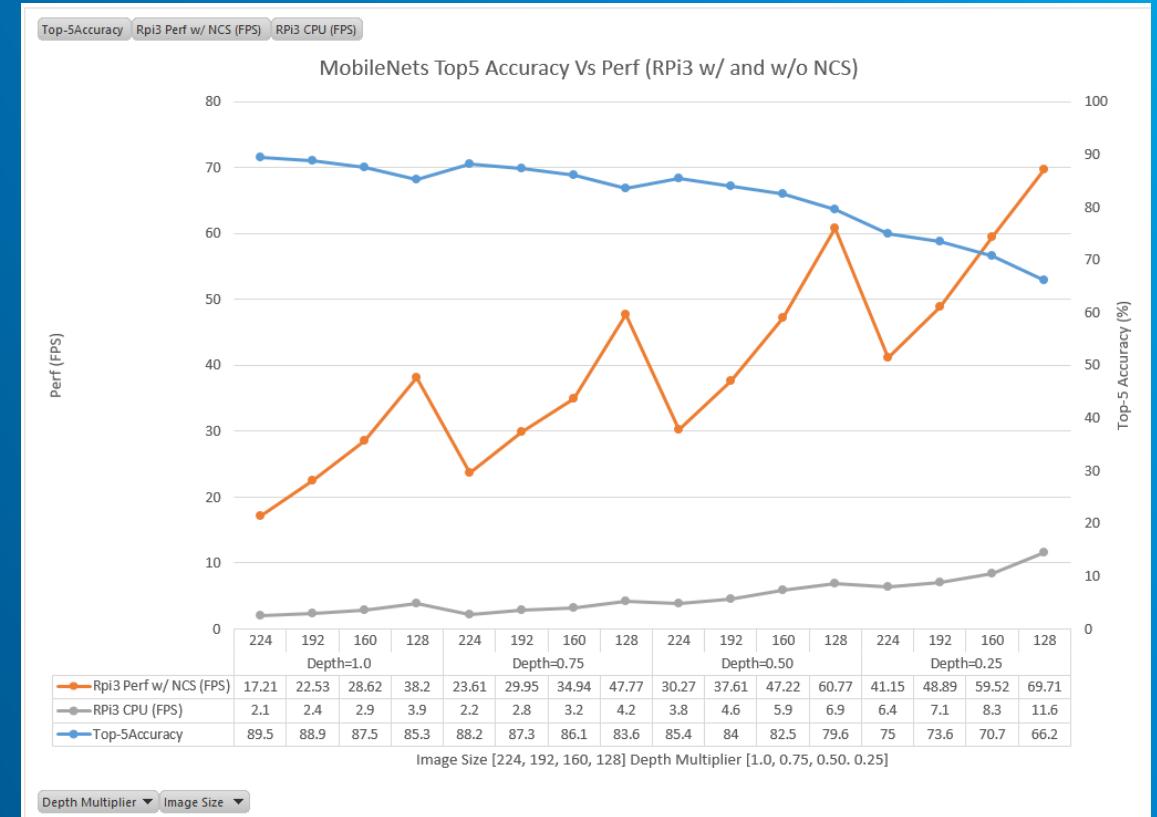
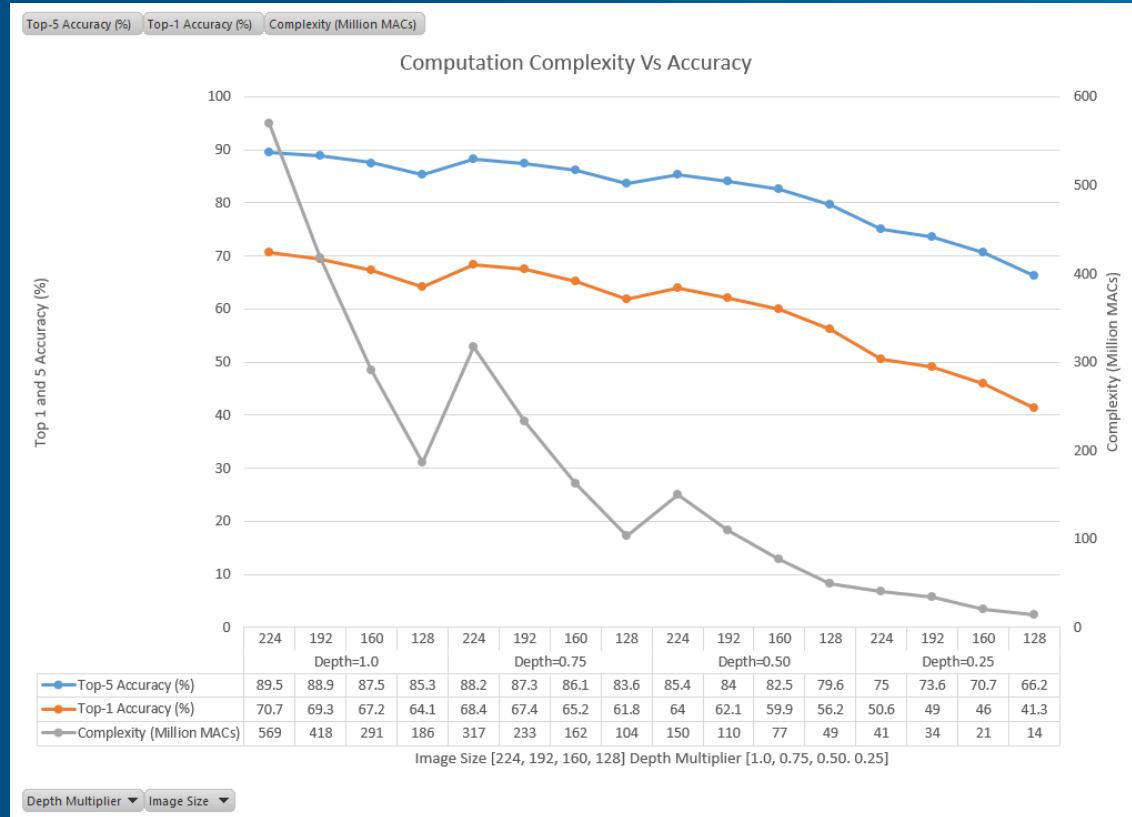


# Profiling MobileNets on NCS

**Use NCS developer tools to profile, compile and deploy mobilenets on NCS**

# Comparing execution times of mobilenets on NCS

Reference: <https://movidius.github.io/blog/ncs-rpi3-mobilenets/>



# Run MobileNet(s) on NCS

Reference: developer.movidius.com > Support > Docs > search for “mvNCProfile”

1. `mkdir -p ~/workspace`
2. `cd ~/workspace`
3. `git clone https://github.com/movidius/ncappzoo`
4. `cd ~/workspace/ncappzoo/tensorflow/mobilenets`
5. `mvNCProfile -s 12 output/mobilenet_v1_1.0_224.meta -in=input -on=MobilenetV1/Predictions/Softmax`
  - Look for `~/workspace/ncappzoo/tensorflow/mobilenets/output_report.html`
6. Modify image-classifier to run inferences using MobileNet(s)
  - Repeat exercise for mobilenets with different depth multipliers and image sizes

# Open hands-on session

**Use NCS developer tools to profile, compile and deploy mobilenets on NCS**

# Feedback

Scan this QR code to fill out our feedback form

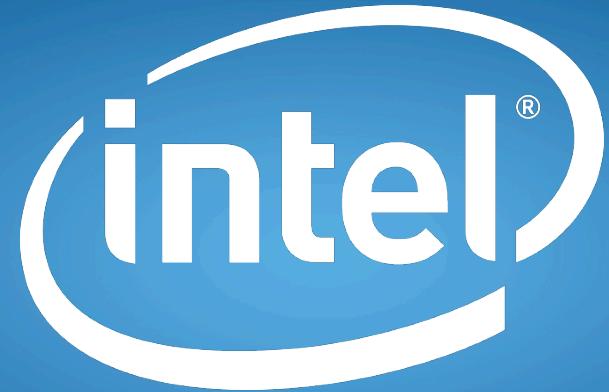


The Movidius NCS is a product designed specifically for developers and data scientists. Our goal with these workshops and competitions is to listen to developer pain points first hand, and incorporate your feedback into upcoming software and hardware revisions.

<https://goo.gl/WMTnjJ>

# Questions?

[shenk.justin@gmail.com](mailto:shenk.justin@gmail.com)



experience  
what's inside™