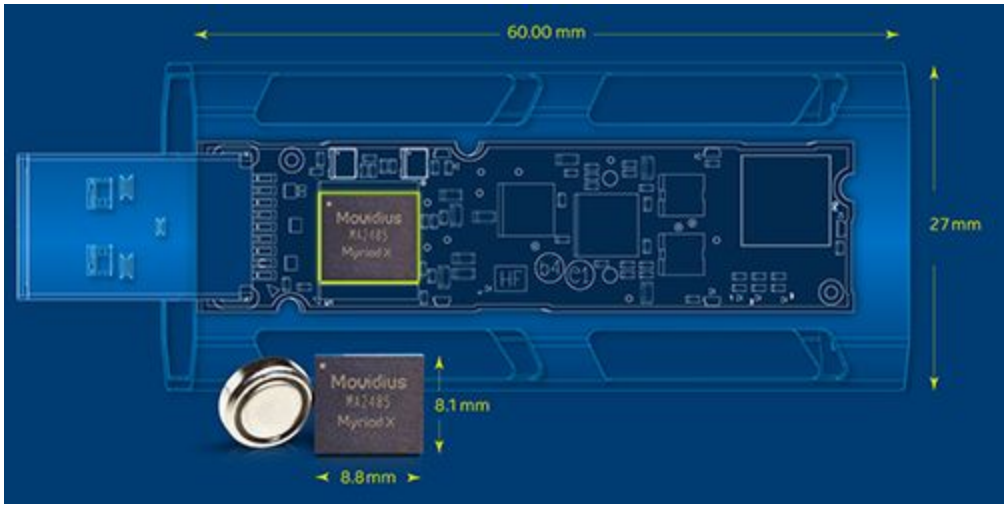
Trying out the Intel Neural Compute Stick 2 - Movidius NCS2

Jonas Werner - February 27th, 2019

Disclaimer: The opinions in this article (and on this website in general) are entirely mine and not those of my employer Dell EMC. Testing has been done in a short period of time and may not accurately reflect real-world performance.

What is it?

This week I've tested the Intel Neural Compute Stick 2 (NCS2). A USB stick for visual computing which originally comes from Movidius - a company acquired by Intel in September 2016. Intel's landing page for the NCS2 can be found here.
The NCS2 is equipped with 16 VPU's or Visual Processing Units which are designed specifically for image and video processing. With this it's possible to run Machine Learning frameworks like Caffe and Tensorflow and to leverage CNN (Convolutional Neural Networks) to do inferencing on data.



So what?

What makes this really interesting is that since it comes in USB format it can simply be plugged into devices at the edge that normally lack the processing power to run machine learning frameworks. With that it becomes possible to process IoT data where it's generated. The NCS2 has the potential to make Edge Computing a reality instead of just a buzzword.
The compound impact could be significant if the network backhaul is taken into consideration. Imagine replacing a constant stream of video data across the network with just the metadata

resulting from running inferencing on the very same video stream. Massive amounts of video vs. some text data. Network savings alone could pay for this pretty quickly.
Note that the very same type of chip is sold embedded in video cameras and drones. They're pretty expensive though. A webcam + a Raspberry Pi and one of the NCS2 sticks could be a very low-cost way to get a security camera with real-time item recognition for very little money in comparison.

Why not use a GPU?

Many edge devices lack the capabilities to host a GPU, either due to space, cost or thermal limitations (GPU's generate a lot of heat and many edge devices are passively cooled).

Hang on - isn't that just a devkit?

Yes and no. The USB stick can be used as a devkit to develop code for embedded versions of the Movidius chip of the kind that may be destined for cameras, drones, robots, etc. However, it can also be used as a very flexible drop-in solution for any edge devices or IoT gateways with low processing power but where the capability to do inferencing is desired.

Is it actually useful?

Yes, it looks like it might actually be able to do the job. The job being processing video directly on the edge devices. In particular it shines when plugged into platforms with low-end CPUs which would never be able to run inferencing on their own.

Functionality testing

To find out if it's powerful enough to process video real-time I used a webcam and fed the video stream to a sample application from the OpenVINO toolkit (link). This particular demo app actually does several things at the same time: Facial recognition, Age detection, Pose detection and Mood detection. All these are run stacked in one command (all details will be included hands-on post shortly). It actually performs very well, although note that the video is not in full HD. Accuracy on the age detection isn't the best, but of course that reflects more on the algorithm / training data than the NCS2 (it thinks I'm in my 20's which is flattering).

Facial, Age and Mood detection with Intel Neural Compute Stick 2

Performance testing

Inferencing can be done on a CPU as well. So, for the NCS2 to be useful it would have to outperform whatever CPU is already on the platform it's plugged into. Therefore I ran a benchmark test (this one) on both the NCS2 and a number of CPUs. The CPUs it was compared to were:

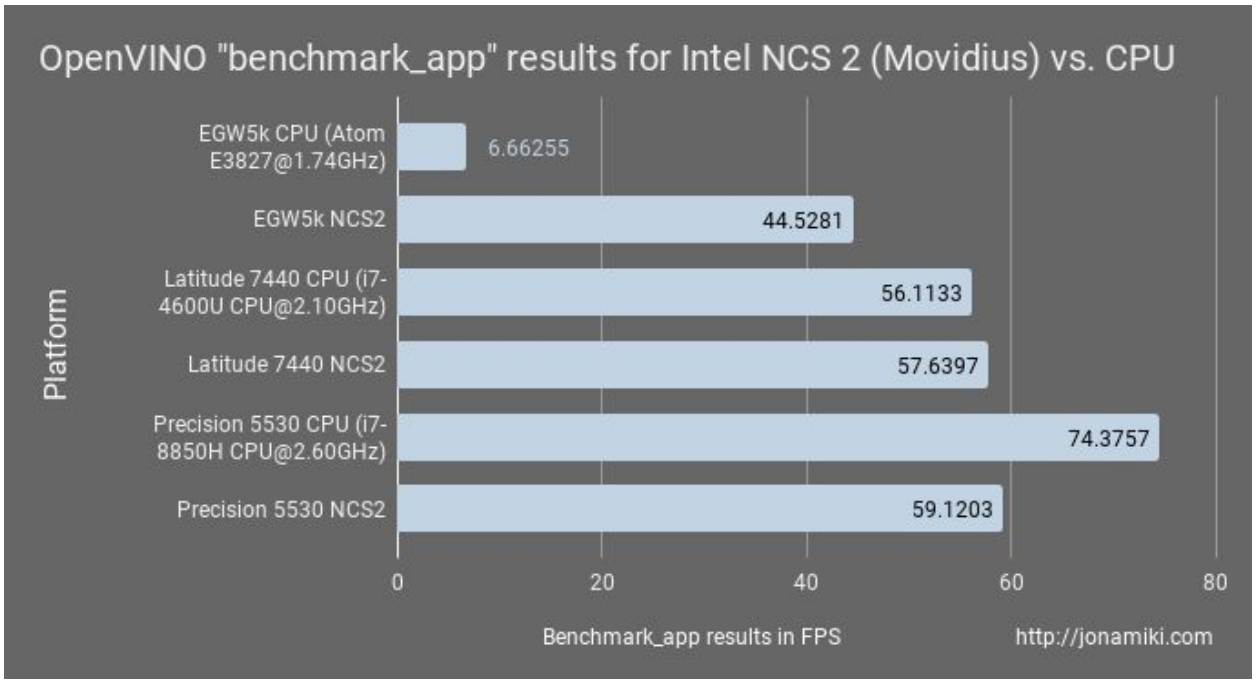
- Atom E3827@1.74GHz

- i7-4600U CPU@2.10GHz
- i7-8850H CPU@2.60GHz

When the NCS2 was being used for the benchmark I was also curious to see if the platform / computer the NCS2 was plugged into affected the benchmark results. Maybe the NCS2 performance would be affected by the host CPU, memory and storage?
The platforms where the NCS2 was tested:

- Dell Edge Gateway 5000
- Dell Latitude 7440
- Dell Precision 5530

Note that the floating point precision differs when running the benchmark on CPU vs the NCS2, so it's not completely apples to apples. This is because the NCS2 only support half precision (FP16) whereas the CPU only support full, or normal, precision (FP32). This probably doesn't make a huge difference when doing inferencing, which is the only thing the NCS2 is likely to be doing in a real-world application. For learning however, the floating point precision may cause the algorithm to learn either garbage or nothing at all. This article is summarizes the topic nicely for those interested: [FP16 and FP32 difference for deep learning]
Each platform was tested with CPU and with the NCS2. Three platforms x 2 tests = 6 results.



NOTE: I only ran these tests a few times, so please don't consider it exhaustive. It would need to be run dozens of times for each and have the results balanced out to get more accurate readings. However, this is all I had time for and it's at least an indicator of performance.
The NCS2 completely outperform the Atom CPU on the Edge Gateway 5000. This is where it was tested initially. Further testing shows that it's more or less equal to a Gen4 Intel i7 but falls behind when compared to a Gen8 Intel i7 CPU.

http://jonamiki.com/2019/02/27/trying-out-the-intel-neural-compute-stick-2-movidius-ncs2/

This is expected of course. The NCS2 isn't a very expensive device at $87.99 USD (Amazon.com at the time of writing). This is a pretty cheap way to add Machine Learning power on devices which have lower-end CPUs, like IoT edge gateways.
There are slight differences in results when the NCS2 is running on less powerful platforms vs. newer machines. This indicates that there are more factors that play into the results than the NCS2 itself, like the type of CPU, memory and storage on the platform the NCS2 is plugged into.
From the results it's clear that the Movidius NCS2 can't compete with a modern i7 CPU, but of course it's a lot cheaper and supposedly draw a lot less power. That would make it ideal for connecting to edge devices where limiting power consumption may be desired.

Practical considerations

For those who may be interested in getting one these I'd like to point out a few things.

1. USB speeds

The NCS2 changes speeds when an app uses it for execution of a neural network. More importantly, when this happens the OS believes that the original USB 2.0 device has been removed and is being replaced with a new USB 3.0 device. This is reversed when code execution finishes.

Movidius stick plugged in:

```
Feb 22 06:29:33 localhost kernel: [  396.100651] usb 1-1: new high-speed USB device number 11
using xhci_hcd
Feb 22 06:29:33 localhost kernel: [  396.230055] usb 1-1: New USB device found, idVendor=03e7,
idProduct=2485
Feb 22 06:29:33 localhost kernel: [  396.230068] usb 1-1: New USB device strings: Mfr=1,
Product=2, SerialNumber=3
Feb 22 06:29:33 localhost kernel: [  396.230077] usb 1-1: Product: Movidius MyriadX
Feb 22 06:29:33 localhost kernel: [  396.230084] usb 1-1: Manufacturer: Movidius Ltd.
Feb 22 06:29:33 localhost kernel: [  396.230091] usb 1-1: SerialNumber: 03e72485
```

Benchmark_app run starting

```
Feb 22 06:30:19 localhost kernel: [  442.564640] usb 1-1: USB disconnect, device number 11
Feb 22 06:30:20 localhost kernel: [  442.993334] usb 1-1: new high-speed USB device number 12
using xhci_hcd
Feb 22 06:30:20 localhost kernel: [  443.122975] usb 1-1: New USB device found, idVendor=03e7,
idProduct=f63b
Feb 22 06:30:20 localhost kernel: [  443.122989] usb 1-1: New USB device strings: Mfr=1,
Product=2, SerialNumber=3
Feb 22 06:30:20 localhost kernel: [  443.122997] usb 1-1: Product: VSC Loopback Device
Feb 22 06:30:20 localhost kernel: [  443.123005] usb 1-1: Manufacturer: Intel Corporation
Feb 22 06:30:20 localhost kernel: [  443.123012] usb 1-1: SerialNumber: 00000000000000000
```

Benchmark_app run finished

```
Feb 22 06:31:28 localhost kernel: [  511.851893] usb 1-1: USB disconnect, device number 12
Feb 22 06:31:29 localhost kernel: [  512.126213] usb 1-1: new high-speed USB device number 13
using xhci_hcd
```

```
Feb 22 06:31:29 localhost kernel: [  512.255008] usb 1-1: New USB device found, idVendor=03e7,
idProduct=2485
Feb 22 06:31:29 localhost kernel: [  512.255020] usb 1-1: New USB device strings: Mfr=1,
Product=2, SerialNumber=3
Feb 22 06:31:29 localhost kernel: [  512.255027] usb 1-1: Product: Movidius MyriadX
Feb 22 06:31:29 localhost kernel: [  512.255034] usb 1-1: Manufacturer: Movidius Ltd.
Feb 22 06:31:29 localhost kernel: [  512.255040] usb 1-1: SerialNumber: 03e72485
```

2. The NCSDK and NCSDK2 can't be used

There are two versions of the NCSDK available. Both of these are for the original NCS and won't work with the NCS2. This is clearly stated on the Intel webpage but if you're like me you may miss it and make the assumption that the NCSDK2 is for the NCS2 stick. I wasted a fair bit of time on this before realizing it wasn't working by design.
Instead use the Intel distribution of OpenVINO which is available here.

3. Can it be run in a container?

Yes, but there are no pre-written Dockerfiles for the NCS2 as there were for the original NCS. The NCSDK2 contains a Docker file but it won't work since it's for a different version of the neural compute stick (see the NCSDK note above).
However, it's not hard to build a container with OpenVINO and run that. I have verified that this works. In fact, there's an excellent Dockerfile by Mateo Guzman available here and I've forked it here since I wanted to make some modifications to it. Feel free to use either of them.
Additionally, if you're impatient or short on time, I've made a pre-built docker image on Docker Hub which can be accessed here.
NOTE: *The container has to be run in privileged mode.* This is due to the NCS2 device ID changing and the USB device being re-enumerated the moment code is loaded onto it (see the USB speed changes note above). The NCSDK2 has a way to change the mode of the original NCS so it can be run in non-privileged mode but this won't work on the NCS2. I don't know of a workaround so far.
I'll write a post on usage shortly which will contain more detail on how to run the demo apps, download and optimize the models for the sample apps as well as how to runt the sample apps themselves.

4. Can it be used on a Raspberry Pi?

Yes, it appears so although I haven't tested it on the Pi yet. Intel has instructions for installing OpenVINO on the Pi here. I may post a Dockerfile or Docker image for the Pi when I've had a chance to test it.

5. What about heat generation and cooling?

The NCS2 is passively cooled and actually is its own heatsink. It's made of metal and the "fins" on both sides of it allow enough airflow to keep it cool. It does get warm during testing but so far not extremely so.

Intel Movidius NCS2 heatsink

6. Size

It's a bit broad, which can make it difficult to plug in at times. It also risks covering other USB ports or as in my case - the power inlet for my laptop. A USB extension cable or powered USB hub could easily mitigate this of course.

Conclusion

The Intel Movidius Neural Compute Stick 2 does seem like a valid option for running inferencing at the edge. While it's not as powerful as a full-on GPU nor a modern CPU, it has the potential to excel in the niche of low-power edge devices like IoT gateways where the onboard CPU isn't powerful enough to do inferencing on its own.