



Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.

www.ayyucekizrak.com/

Mar 4 · 3 min read

A Brief Guide to Intel Movidius Neural Compute Stick with Raspberry Pi 3

Today, low-power consumption is indispensable for autonomous/unmanned vehicles and IoT (Internet of Things) devices. In order to develop deep learning inference application at the edge, we can use Intel's both energy efficient and low cost Movidius USB stick!

Movidius Neural Compute Stick (NCS) is produced by **Intel** and it can be run without any need of Internet. This software development kit enables rapid prototyping, validation, and deployment of deep neural networks. Profiling, tuning, and compiling a DNN on a development computer with the tools are provided in the Intel Movidius Neural Compute SDK. The Movidius NCS' compute capability comes from Myriad 2 VPU (Vision Processing Unit).



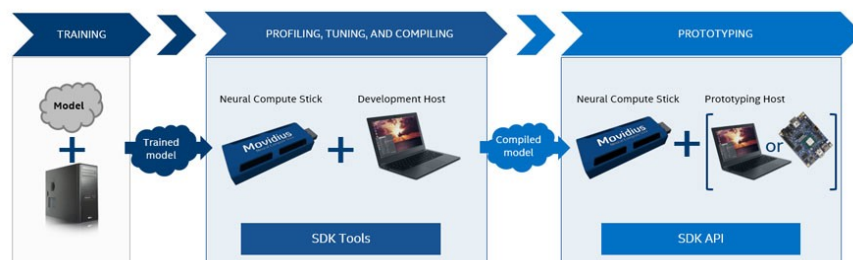
Movidius and DJI Bring Vision

Running Deep Learning models efficiently on low capacity graph processors is very painful. Movidius allows us to optimize the operation of large models such as GoogLeNet (thanks to Yann LeCun) with multi-use support. It is an easy-to-use kit that allows you to design and

implement applications such as classification and object recognition as physical products.

We can *simply* think of Movidius NCS as a GPU running on USB (Graphics Processing Unit). However, training of the model is not performed on this unit, the trained model works optimally on the unit and is intended to be used in physical environments for testing purposes.

- We can use with Ubuntu 16.04 or Raspberry Pi 3 Raspbian Stretch.
- It supports two DNN frameworks (TensorFlow and Caffe).
- Movidius Myriad 2 VPU works with Caffe based convolutional neural networks.
- We can run complex deep learning models like SqueezeNet, GoogLeNet and AlexNet on your computer with low processing capability.



Movidius Neural Compute Stick Workflow

APPLICATION

It is so simple to run a image classification demo. Now we can use NC App Zoo repo for classifying an image. We need to take **graph** file to activate application of Movidius NCS. It has compiled GoogLeNet model for ready to run. This application needs some files.

`make` command is used for creating the files that Movidius needs as graph file. Graph file is a demo of image-classifier.

View the steps below for a quick applicaiton:

- For using property of NCSDK API *add (import) mvnc library*

```
1 import mvnc.mvncapi as mvnc
```

- You can access to Movidius NCS using API like any other USB Devices. Also you can use parallel Movidius devices at ones if you need more capacity to compute your model. For now, one kit is enough for this application. Select and open process:

```
1 # Look for enumerated Intel Movidius NCS device(s); quit pr
2 devices = mvnc.EnumerateDevices()
3 if len(devices) == 0:
4     print('No devices found')
5     quit()
6 # Get a handle to the first enumerated device and open it
```

- Pretrained GoogLeNet* model for using compiled graph file:

```
1 # Read the graph file into a buffer
2 with open(GRAPH_PATH, mode='rb') as f:
3     blob = f.read()
4 # Load the graph buffer into the NCS
```

- Need to do some pre-processing before loading the image in Movidius NCS:

```
1 # Read & resize image [Image size is defined during trainin
2 img = print_img = skimage.io.imread( IMAGES_PATH )
3 img = skimage.transform.resize( img, IMAGE_DIM, preserve_ra
4 # Convert RGB to BGR [skimage reads image in RGB, but Caff
5 img = img[:, :, ::-1]
6 # Mean subtraction & scaling [A common technique used to ce
```

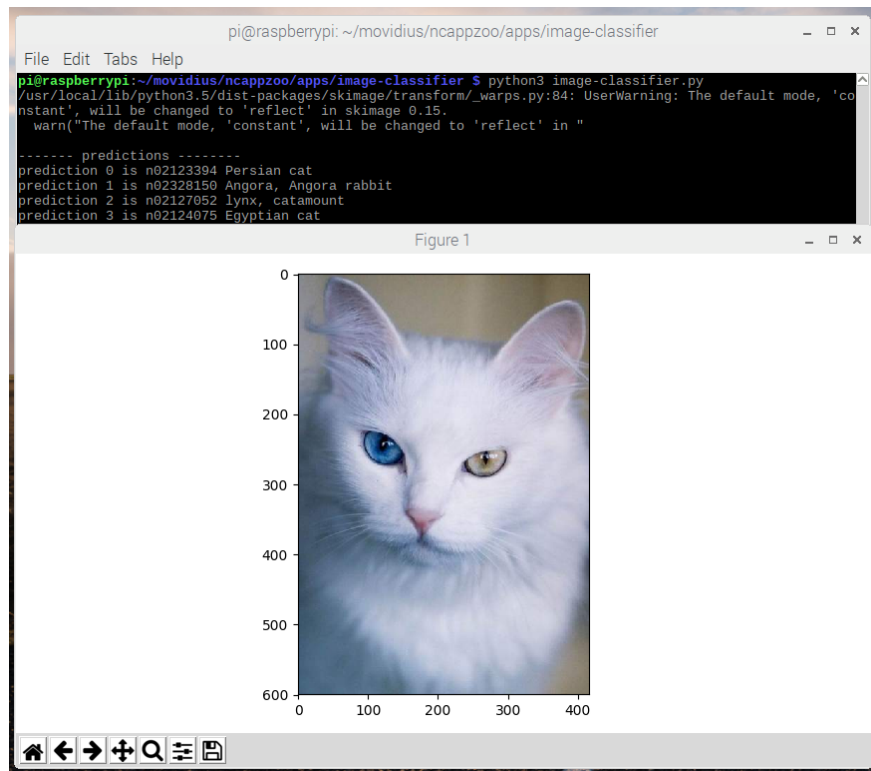
- Use `LoadTensor()` for loading the image to Movidius NCS:

```
1 # Load the image as a half-precision floating point array
2 graph.LoadTensor( img.astype( numpy.float16 ), 'user object
```

- Give the input image to pretrained model than getting output using by `GetResult()` :

```
1 # Get the results from NCS
2 output, userobj = graph.GetResult()
```

- Printing prediction of model output and corresponding labels.
Also displaying the input image at the same time.



Test Result of Image Classification Problem (Fail)



Test Result of Image Classification Problem (Successful)

- At the last step, *clear and shutdown* to **Movidius NCS** device for use it again:

THANKS

I'd like to thank **Intel Türkiye** and **Mustafa Aldemir** for kindly and immediate donate a **Movidius Neural Compute Stick** to support my Deep Learning Research.



Movidius Neural Compute Stick

References:

1. <https://developer.movidius.com/>
2. <https://www.movidius.com/news/movidius-and-dji-bring-vision-based-autonomy-to-dji-phantom-4>
3. <https://github.com/movidius/ncappzoo>
4. <https://software.intel.com/en-us/articles/build-an-image-classifier-in-5-steps-on-the-intel-movidius-neural-compute-stick>
5. <https://towardsdatascience.com/getting-started-with-intel-movidius-d8ba13e7d3ae>

