*Tensorflow Object Detection*

*on Raspberry Pi and Intel Movidius*

Melentev Nikita

Skoltech

Moscow, Russia, 2019

ndmelentev@skoltech.ru

*Abstract* — This paper describes parameters such as time of execution, overheating and CPU load using Raspberry Pi alone and with Intel Movidius to execute pre-trained neural networks for object detection. Several sets of experiments were conducted on neural networks with different depth and input size. Limitations of using Movidius are described and guide as well as code to launch experiments is provided.

Keywords—machine learning, object detection, neural networks, raspberry pi, embedded systems, movidius, tensorflow.

# Introduction

With recent progress in machine learning there as still some cases in which we cannot use the full potential of strong algorithms such as neural networks due to lack of computational power of some devices. One way to improve this situation is to use portable extensions to execute these algorithms. In our work we will use Intel Movidius [0] with Raspberry Pi.

To test the difference between raw Raspberry Pi and Movidius performances Mobilenet\_v1 [1] neural networks with different depth and image input size are used. They are pre-trained on ILSVRC-2012-CLS [2] dataset and can predict up to 1000 classes of objects. The following measurements are compared: FPS that camera can obtain, time to process one image, CPU load, CPU temperature, RAM load.

# Requirements and Specification

## Harware requirements

* Raspberry Pi 3 model B;
* SD card running Raspbian stretch;
* Movidius USB stick;
* Power cable, USB extension cable;
* Изображение выглядит как карта

  Описание создано автоматическиKeyboard, mouse;
* HDMI screen, USB camera;
* Internet access (Ethernet or Wi-fi).

## Software requirements

* Raspbian stretch OS;
* Python3.5;
* Tensorflow;
* Movidius neural compute SDK.

# Preparing Raspberry Pi for Experiments

Movidius can only work with graphs that can be converted from Tensorflow or Caffee models. Because of this, we will use pure Tensorflow neural networks. So, the first step to launch experiments on Raspberry Pi is to install Tensorflow and all its dependencies [3].

After successful installation we can download and execute pre-trained Mobilenet v1 tensorflow models.

The next step is to convert previously downloaded models into Movidius format [4][5]. Another limitation that we must consider is that only models with strict input size can be converted.

Finally, we can now launch the same models but using power of Movidius and check how much performance did we gain.

Step by step instructions are available in my GitHub repository [6].

# Experiments

Mobilenet v1 set of neural networks are different in depth and image input size. Depth is encoded in four classes: 0.25, 0.5, 0.75 and 1, each class corresponds to accuracy that can be achieved and number of iterations in such network (figure 1). As we can see, greater depth parameter results in greater accuracy, but we will need to do additional computations.

Figure 1

Изображение выглядит как карта

Описание создано автоматически Similar situation can be observed with input size. It is encoded in four classes: 128, 160, 192 and 224 pixels. All input sizes are square. With increase in input size, accuracy is increased but so is number of iterations (figure 2).

Figure 2

From now on in figures we will use combined notation of model depth and input size, so that ‘1\_128’ is model with 1 depth and 128 input size and ‘0.25\_224’ is model with 0.25 depth and 224 input size.

# Conclusion

# References

[0] Intel Movidius. Neural Comput SDK. <https://movidius.github.io/ncsdk/index.html>

[1] Mobilenet pre-trained neural network models. <https://github.com/tensorflow/models/blob/master/research/slim/nets/mobilenet_v1.md>

[2] Large Scale Visual Recognition Challenge 2012.

<http://www.image-net.org/challenges/LSVRC/2012/>

[3] EdgeElectronics. Setting up TensorFlow on Raspberry Pi. <https://github.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi>

[4] Humphrey Sheil. Movidius with Raspberry Pi. <https://medium.com/@hsheil/movidius-neural-compute-stick-and-raspberry-3-quick-start-guide-a89ff5e1d7ca>

[5] Intel Movidius. Guide to compile tensorflow models. <https://movidius.github.io/ncsdk/tf_compile_guidance.html>

[6] Melentev Nikita. Complete guide to launch object detection of Raspberry Pi with Movidius. <https://github.com/ndmel/tensorflow_object_detection_on_raspberry_pi>