

## Problem:

When working with video analysis and processing in python, high processing power is required to get the best results. One such application is real time object detection. Commonly used algorithm for this purpose is MobileNetSSD.

If such algorithm is executed on a standard desktop computer, it gives a good output with frame rate of about 15 frames per second. However, the same algorithm shows a huge drop in performance when executed on a micro controller such as Raspberry Pi. The performance drops to about 4 frames per second.

## Solution:

The MobileNetSSD model is a Single-Shot multi-box Detection (SSD) network that is used to detect objects. The Caffe\* framework is used to implement this model.

Caffe is a deep learning framework that prioritizes flexibility, speed, and modularity.

**Microcontroller – Example: Raspberry Pi V4 with a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor. Is well sufficient for running small image classification network.**

**But when comparing it with basic desktop computer processor with Intel® Core™ i7-1160G7 Processor with 12M Cache, and up to 4.40 GHz or i9, Microcontrollers are far behind.**

Also running huge image learning classification models designed using the Caffe framework requires heavy resources to provide good efficiency in frame rates.

**Still, there are some key points by which we can enhance the frame rate accuracy of the Raspberry Pi V4 model.**

### 1. Looking for Another Deep Learning Model for Classifying images that require fewer resources and provide more accurate results.

I am currently doing my research thesis which is somewhere 80% similar to this scenario, I am classifying animal images from wildlife using **Deep Learning Model Resnet V2 for wildlife monitoring** and the IoT tool which is required for capturing images is “Camera Traps”. A camera trap is an Internet of Things (IoT) system that is activated by a shift in behavior in its immediate vicinity, such as the presence of an animal. It normally has a motion sensor, which is either a passive infrared (PIR) or an active infrared (AIR) sensor that uses an infrared light beam.

Using TensorFlow and Resnet V2 model instead of using Caffe framework.

Resnet V2 is the most advanced reinforcement learning algorithm in deep learning for classifying images.

Tensorflow is far better and lightweight in terms of running on any microcontroller device.

TensorFlow	Caffe
TensorFlow provides high-level APIs for model building, allowing us to experiment with TensorFlow APIs quickly. It has a suitable gui for machine learning jobs in Python (which is the preferred language for data scientists).	Caffe lacks higher-level APIs, making it difficult to play with Caffe's configuration in non-standard ways using low-level APIs. Caffe's middle-to-low-level API approach offers minimal high-level support and deep configurability. Caffe's interface is more C++-based, which means users must perform more tasks manually, such as creating configuration files.
We can use CPUs by calling <code>tf.device()</code> , which allows us to make all required modifications without the need for additional documentation or API changes. We can run two copies of a model on two CPUs and a single model on two CPUs in TensorFlow.	Python does not have any tool support. As a result, all instruction must be done using a C++ command line interface. It only supports one kind of multi-GPU configuration, while TensorFlow allows for a variety of different multi-GPU configurations.

## 2. Using Xailient over MobileNetSSD as it shows goof fps than MobileNetSSD.

Example: Test Conducted by Sabina Pokhrel on Real-Time Vehicle detection using Xailient and MobileNetSSD.

Experiment: Both Test conducted on Lenovo Yoga 920 with Ubuntu18.04 operating system and Raspberry Pi 3B+ with Raspbian Buster operating system.

Results: Xailient shows a far better FPS rate than MobileNetSSD

Citation: <https://www.xailient.com/post/real-time-vehicle-detection-with-mobilenet-ssd-and-xailient>

## 3. Change is some configurational setting.

- A. Connect a Reliable Power Supply
- B. Using a Lightweight Operating System
- C. Using High-Performance micro-SD Card
- D. Overclocking CPU can also increase performance.
- E. Using External USB for booting will certainly speed up booting and make performance more reliable while expanding the storage capacity of the Pi.
- F. Increase RAM size by adding external hardware.

## 4. Looking for other microcontrollers

Example:

- A. Coral Dev Board from Google
- B. Jetson NanoEdge TPU
- C. Working with Intel Hardware with OpenVINO framework
- D. Working on NVIDIA Jetson Nano. Built around a 64-bit quad-core Arm Cortex-A57 CPU running at 1.43GHz alongside an NVIDIA Maxwell GPU with 128 CUDA cores.

For Comparison, please check below image.

Board	MobileNet v1 (ms)	MobileNet v2 (ms)	Idle Current (mA)	Peak Current (mA)	Price (US\$)
Coral Dev Board	15.7	20.9	600	960	\$149.00
Coral USB Accelerator	49.3	58.1	470	880	\$74.99+\$35.00
NVIDIA Jetson Nano (TF)	276.0	309.3	450	1220	\$99.00
NVIDIA Jetson Nano (TF-TRT)	61.6	72.3			
Movidius NCS	115.7	204.5	500	860	\$79.00+\$35.00
Intel NCS2	87.2	118.6	480	910	\$79.00+\$35.00
MacBook Pro <sup>1</sup>	33.0	71.0	1570	1950	>\$3,000
Raspberry Pi	480.3	654.0	410	1050	\$35.00

<sup>1</sup> The MacBook Pro takes a +20V supply, all other platforms take a +5V supply.

Citation: [Measuring Machine Learning. From desktop, to single board computer... | by Alasdair Allan | Towards Data Science](#)