Exploring Video Classification on Raspberry Pi

Matthew Joseph Guidry

October 13, 2017

1 Abstract

The Raspberry Pi lacks the performance necessary to do streaming data analysis with Tensorflow. Even processing a single frame at a time does not result in performance that could be called ”realtime”. However, this project could be a good demo to show basic aspects of deep learning. A faster language or a more optimized framework could give a result more significant than a technical demo.

2 Approach and Results

Tensorflow was chosen for the device because it is a common deep learning framework that supports the python programming language. Tensorflow must be built for the Pi either directly on the device or by cross compiling because it is not directly supported. Building on the device gives some challenges as the Pi does not have enough memory to build Tensorflow. A storage must be attached to use as a temporary swap space. The building takes several hours and may give several errors. Instead of spending an entire week trying to build it, a prebuilt package was installed. A version of tensorflow exists for mobile devices such as iOS, Android, and Raspberry Pi, but it does not have any language bindings.

The first thing to do was to test the performance of the device by classifying a single image. The InceptionV3 example from the Tensorflow repository was chosen to test this. It was modified to take an image from the Raspberry Pi camera using the PiCamera python library. The PiCamera has two ports that it can use to capture images. Both ports have supported standard resolutions. The ”still port” can only directly capture fullsize images without GPU resizing. The ”video port” has a wider range of direct capture resolutions.

Performance for Varying Resolutions		
Image Size	Video Port	Still Port
3280x2464(Full)	1.965s	2.016s
640x480	1.815s	1.857s
320x240(GPU Resize)	1.795s	1.832s

These times include the time for the image to be captured and the time for classifying the image. The time differences were small, but were consistantly different over multiple runs.

The example was then heavily modified to capture "video" at two frames per second. The capture resolution is set to 320x240. This means that an image is taken at a higher resolution and then downscaled on the GPU. Single images are continuously taken from the stream and classified, returning only the tag with the highest confidence value. Since the video port is being used with resolution of 320x240, the individual classification times are in line with the times from the single image run, being 1.786 seconds on average.

3 Conclusion

This resulting performance is not what would be called "realtime" video classification. What is being processed is not even what could be called "video". This project could be a good technical demo to show people unfamiliar with deep learning what it is capable of. There are several methods that could improve this result and make it more useful. A deep learning framework better optimized for small scale devices could substitute Tensorflow. Using Tensorflow for the C programming language could remove the python overhead. In general, the Raspberry Pi does not have the raw processing power necessary to do the large computations necessary in deep learning applications.