Miguel Millan

CSI4999 - Senior Capstone

1/18/2019

Hardware Research

**Objective of Research**

* To determine the hardware requirements for deploying an effective event detection system within the limitations of a bus camera system.
* To identify potential camera and computing systems for use in the local subsystem of our event detection system.

**Findings of Research**

<https://blog.algorithmia.com/hardware-for-machine-learning/>

* Essentially, when it comes to machine learning it’s all about how efficient the chip performing calculations is. CPUs are great and all, but only perform one calculation at a time. GPUs can be used to great effect due to their parallel computing capabilities. However, it would seem that Application Specific Integrated Circuits (ASICs) designed specifically with machine learning in mind would be the best types of chips to use for machine learning.
* Google has created such an ASIC called the Tensor Processing Unit (TPU) which can train models in the cloud for roughly $4.50 USD per hour of use (<https://cloud.google.com/tpu/docs/pricing>). The issue here is that it would only allow for training in the cloud, which may be good for training the algorithm, but does not provide localized hardware we could put on a bus.
* GPU computing tends to be the fastest alternative, but is usually about 4 times more expensive than CPU computing. As we are not necessarily concerned with speed at the moment, I think it would be best to stick with CPU computing for training the algorithm.
* If we were to host the cloud application in AWS we’d have the option of using CPU or enabling GPU computing for training, but again there is the cost benefit we’d need to consider.

<https://dzone.com/articles/how-to-train-tensorflow-models-using-gpus>

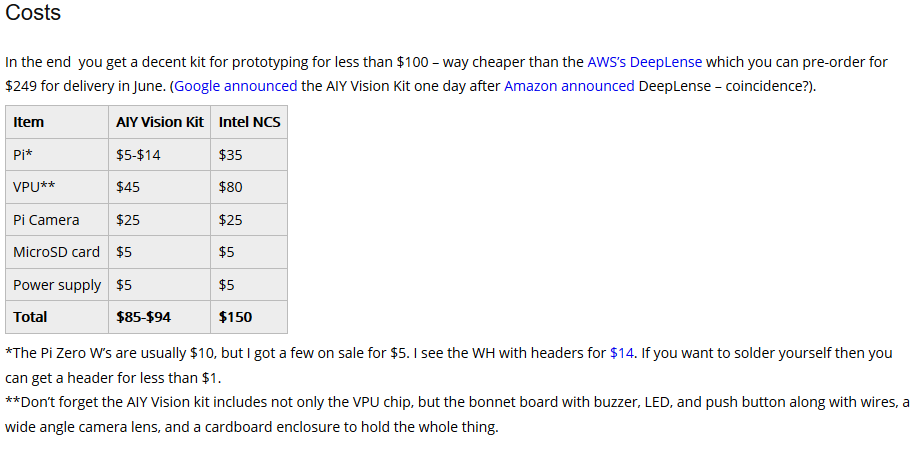
* Guide/example of using a GPU enabled AWS Ubuntu instance for training data using TensorFlow.
* Also provides some cost comparisons. Of note, it takes a 16 CPU instance ($0.796 per hour) to match the power of a 1 GPU instance ($0.650 per hour). Both of these are much cheaper than the specialized TPU google offers.

<https://www.royyak.com/build-image-classifier-robot-using-raspberry-pi-deep-learning/>

* Very interesting, this one is. In this blog, they run through the process of creating an Image classifier robot on a Raspberry Pi.
* Although the Pi is slow and can lag behind, it may be possible to use deep learning on the Pi to classify events on the bus, at least as a proof of concept, and use more powerful hardware to deploy an actual system.
* Interestingly, the pi used a camera that was only $9 to capture images for classification.
* The main concern I have is that, this project focused on single image processing, while we may need to take a large number of images (over the video stream), in which case the Pi might not be up to snuff, even for proof of concept.

<https://webrtchacks.com/aiy-vision-kit-tensorflow-uv4l-webrtc/>

* This one might be perfect for our use case.
* Essentially, we could use an AIY Vision Kit to deploy a simple camera attached to a raspberry pi equipped with an [Intel® Movidius™ Myriad 2 MA2450](https://www.movidius.com/myriad2) vision processing unit (VPU) chip.
* It would cost around $100, but that’s honestly not too bad considering it includes the camera, a VPU, the PI, SD card and power supply.



* To enable streaming to the cloud system, we could use WebRTC with UV4L
* Here’s the kit for $90 at Target: <https://www.target.com/p/-/A-53417081>

<https://blog.coast.ai/continuous-online-video-classification-with-tensorflow-inception-and-a-raspberry-pi-785c8b1e13e1>

* Here is yet another project that utilizes a raspberry pi for classification.
* What I really like about this article is that it shows an example of what we kind of want to do, but in the simpler case of detectiving football or ad in a live stream.
* Essentially, they gather data and performed offline training on a more powerful computer. Then they tested the trained model a new set of data (different from the training set). Then, they transferred the model to a raspberry pi 3 equipped with a camera. The raspberry pi would send the video feed to the tensorflow model instead of saving it. The only downside really was that it took about four seconds to classify a frame.

<https://www.techrepublic.com/article/raspberry-pi-not-powerful-enough-check-out-these-20-beefy-boards/>

* Here are a list of a number of raspberry pi alternatives.
* On the more expensive side there is the UDOO X86 Ultra at $267 which would be about 10x faster, have much more memory, and is more power efficient.
* There is the less expensive Banana Pi M3 which has 8-cores clocked at a faster speed than the raspberry pi and double the memory, but cost $85 and is regards as more difficult to set up.
* Another less expensive option that is also faster than the Pi is the Odroid-XU4 for $59 which boasts double the memory of the pi3 and a faster 8-core cpu.
* The NanoPi K2 is slightly faster than a Pi3 with a cost of $39.99, but probably wouldn’t be much better than a Pi in all honesty.
* There are a number of other alternatives listed, but they are either less powerful, don’t support some OSs, or are comparable to the above.

<https://www.amazon.com/Tinker-board-RK3288-1-8GHz-Mali-T764/dp/B06VSBVQWS>

* Also found this Tinker board on amazon for $57.45. It is faster than the pi3 with a 1.8GHz cpu.

**Research Conclusion**

* To train our algorithm, it would be best to utilize a GPU-enabled AWS instance from amazon. This would cost about $0.650 per hour of use.
  + Alternatively, we could use a much slower instance (about 3.3x slower) which cost $0.199 per hour, but training time would make the cost roughly equivalent.
  + Another alternative would be to just use a cheap instance to host the web page, and the more expensive GPU-enable instance would only be used to initially train the model/algorithm (this way we aren’t running the expensive instances all the time).
* If we only use the GPU-enabled instance for 30hrs of training it would cost $19.50. If we assume we’ll have a total training time of 100hrs, then it would cost $65.
* For the local system, we don’t necessarily need a very powerful system to utilize a trained model. Bear in mind, however, that the more cpu power the computer has the faster it will be able to classify events. For this reason, ideally we would use a UDOO X86 Ultra which costs $267.
  + While there are cheaper alternatives, the performance goes down with the price. For instance, a raspberry pi3 at $35 would be able to classify an image about every 4 seconds.
* The most cost efficient option (in my opinion) would be an AIY Vision Kit which cost about $89 and is specifically designed for vision processing on a low spec device. A nice bonus being that it comes with the camera included.