Comparisons and Optimisations of CNN Accelerators.

James Garland

September 24, 2021

Abstract

Machine learning (ML) is a subset of artificial intelligence (AI) that are used for many applications such as hand writing recognition, face and voice recognition. Convolutional neural networks (CNNs) (as subset of ML) are used widely for object detection and recognition in images and videos. CNNs are trained for recognition on high-performance machines but can be implemented in embedded devices for inference (the detection and recognition of objects in new images). Devices such as Raspberry Pi, Google Coral, Nvidia Jetson Nano, Intel neural compute stick (NCS) accelerators are widely used for inference. Benchmark comparisons of known CNN models must be measured and optimisations made to increase the performance of the models.

1 Proposal

This project aims to use various accelerator development boards to prototype different forms of CNN models. Each model shall be bench marked against each other. Once bench marking has been established, optimisations of the hyperparameters of the models shall be made to increase performance on each development board. The optimisations shall then be bench marked to investigate which board, model and hyperparameter set performs the best.

The coding of the models shall generally be written in Python but may also include some C/C++. We shall investigate using known training, test and validation datasets to train the models. The trained model shall then be implemented on each of the accelerator boards and executed in inference mode to obtain the performance metrics.

The models inference hyperparameters shall be adjusted to attempt to achieve better performance metrics of each accelerator board. If time allows, the training hyperparameters may also be adjusted for greater performance.

2 Specifications

The outline specifications are:

- A Google Coral, Nvidia Jetson Nano, Raspberry Pi 4, Intel NCS development board shall be used.
- The CNN model such as YOLO and Mask R-CNN shall be coded in Python and/or C/C++.
- The trained model and inference data sets shall be implemented on each board.
- Performance metrics shall be obtained. Optimisations shall be made to increase performance characteristics such as speed, classification accuracy and energy consumption.
- If time allows, the training hyperparameters may be adjusted for increased performance metrics.