Object Detection Using Low Cost Embedded Systems

California State University SAN MARCOS

Thomas Thuan Le Nguyen
lengu002@cougars.csusm.edu

Dr. Ali Ahmadinia
aahmadinia@csusm.edu

Department of Computer Science and Information Systems, CSUSM

1. preface:

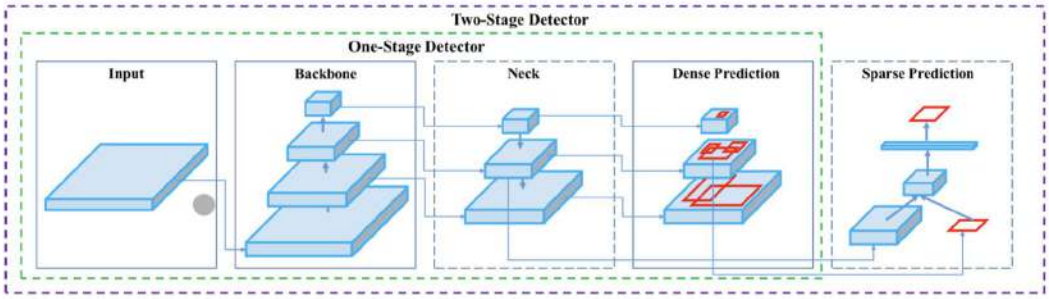
Object detection with Coral Dev Board, which is a low cost embedded system made by Google that contains an Edge TPU coprocessor. The Coral Dev Board can be used to run machine learning models for object detection.

Main focuses included:
* benchmark its runtime and FPS using MS COCO dataset
* find ways to improve the board's FPS when it is running the model

2. introduction:

Computer vision (CV) is a field of artificial intelligence (AI) that allows computers to have a strong understanding from images and videos. CV is useful because it enables a computer to do something that the human visual system can do.There are different forms of computer vision such as image classification, object detection and semantic segmentation. Object detection is a computer vision technique for detecting and locating objects in images and videos. It can be used to count objects in a scene, locate and track their specific locations, and label them. It is helpful in fields that include autonomous machines, security, and tracking objects.

* YOLOv4 (you only look once) is an object detection algorithm
* 4th version provides a backbone and a better neck, easier to use a singular GPU
* The backbone is a deep neural network which is composed mainly of convolution layers that are an extraction of essential features
    * Bag of freebies - increases the cost of training or changes the training strategy and leaves the cost of inference low
    * Bag of specials - increases inference cost but can improve the accuracy of object detection
    * CSPDarknet53 - concatenates (2) the previous input with the current input before going to one of the many dense layers
* The neck collects feature maps from different stages.
* The head contains dense prediction and sparse prediction
* YOLOv4 is a one-stage so it uses dense prediction
    * Dense prediction (one-stage) - vector that hold the coordinates of the predicted bound box and the confidence score of the object
* Sparse prediction (two-stage)



3. test:

hardware:
training:
* NVIDIA Tesla T4 GPU
* Intel(R) Xeon(R) CPU @ 2.20GHz
* 12.5GB of RAM

testing:
* Integrated GC7000 Lite Graphics
* NXP i.MX 8M SoC (quad Cortex-A53, Cortex-M4F)
* 1 GB LPDDR4

software:
training:
* Google Colab
* Tensorflow-GPU version 2.3.0-rc0
* Python version 3.7.11
* opencv version 4.1.1.26

testing:
* Tensorflow-GPU version 2.3.0-rc0
* Python version 3.7.3
* opencv version 4.1.1.26

4. challenges:

* Colab would crash because the runtime would be too long
* some installations that are potentially needed to run on the Coral Dev Board would not work/did not work at all because there are not any installations that are compatible with it, such as CUDA Toolkit or NVIDIA cuDNN
* runtime took too long, so if there were any errors we would have to see it after it was done processing
* converting tflite model so it is compatible with Edge TPU was difficult and unfinished because some of its operations were not transferable and compatible with the board's operations
* using a mouse or keyboard is useless with the board because there is no browser or functioning Desktop, had to run everything through Terminal on another computer

5. results:

Test	Image Size:	Tiny	Tflite	Average FPS
Colab:	320	FALSE	FALSE	37.767
	320	FALSE	TRUE	0.749
	320	TRUE	FALSE	142.065
	320	TRUE	TRUE	6.681
	416	FALSE	FALSE	29.162
	416	FALSE	TRUE	0.461
	416	TRUE	FALSE	120.267
	416	TRUE	TRUE	4.209
	608	FALSE	FALSE	17.094
	608	FALSE	TRUE	0.197
	608	TRUE	FALSE	96.106
	608	TRUE	TRUE	1.978
Coral Dev Board:	320	FALSE	FALSE	0.225
	320	FALSE	TRUE	0.103
	320	TRUE	FALSE	2.541
	320	TRUE	TRUE	1.102

I picked the network size of 320 because it maintained a consistently high FPS resulting in less time and less resources used.

Test	Image Size:	Tiny	Tflite	FPS Differences
Colab:	320		FALSE	3.762
	320		TRUE	8.920
	320	FALSE		50.423
	320	TRUE		21.264
Coral Dev Board:	320		FALSE	11.293
	320		TRUE	10.699
	320	FALSE		2.184
	320	TRUE		2.306

Table above has blank spots in the tiny and tflite columns because for the FPS differences calculations, they had both models (tiny & non-tiny or tflite & non-tflite).

6. conclusion:

* on Colab:
    * both tiny models have greater FPS (~4 times and ~9 times) than both non-tiny models
    * both tflite models have less than 20 times slower than both non-tflite models
* on Coral Dev Board:
    * both tiny models have 11 times greater FPS than both non-tiny models
    * both tflite models are more than 2 times slower than both non-tflite models
* Tflite models have significantly low FPS
* significant FPS differences between tiny and non-tiny, having tiny models resulted in faster runtimes
* find a way to increase FPS on the Coral Dev Board using eGPUs

6. references:

[1] theAIGuysCode. (n.d.). theAIGuysCode/tensorflow-yolov4-tflite: YOLOv4, YOLOv4-tiny, YOLOv3, YOLOv3-TINY implemented in TENSORFLOW 2.0, ANDROID. Convert YOLO v4 .weights Tensorflow, TENSORRT AND TFLITE. GitHub. https://github.com/theAIGuysCode/tensorflow-yolov4-tflite.
[2] Bochkovskiy, A., Wang, C.Y., & Liao, H.Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
[3] RUGERY, P. (2020, September 15). Explaining Yolcli4 a one stage detector. Medium. https://becominghuman.ai/explaining-yolov4-a-one-stage-detector-cdac0826cbd7.
[4] Wang, C.Y., Mark Liao, H.Y., Wu, Y.H., Chen, P.Y., Hsieh, J.W., & Yeh, I.H. (2020). CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 390–391).
[5] Wang, C.Y., Bochkovskiy, A., & Liao, H.Y. (2021). Scaled-YOLOv4: Scaling Cross Stage Partial Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 13029-13038).
[6] Zicong Jiang, Liquan Zhao, Shuaiyang Li, & Yanfei Jia. (2020). Real-time object detection method based on improved YOLOv4-tiny. arXiv preprint arXiv:2011.04244.

7. acknowledgements:

This research project could not have been done without the directors Dr. Jameson and Dr. Read, as well as my professor Dr. Ahmadinia. Special thanks to Genentech for funding this project.

Genentech
A Member of the Roche Group