FPGA PYNQ Hardware Implementation for UAV tracking system

Ting-Shen Kuo, ***Institute of Data Science and Engineering , National Yang Ming Chiao Tung University, Taiwan*** **(tsk81634@gmail.com)**
Advisor: Prof. Chin-Tien Wu(吳金典)

Abstract

Until now, object detection and tracking system are widely utilized in many fields. Quite impressive attention has been paid for real-time object detection and tracking. However, it's always a challenge to perform the task with fast inference while maintain the accuracy. Although Graphics Processing Units(GPUs) are more efficient and stable, they require large power, energy consumption, and have large computational load problems. Recently, Xilinx Field Programmable Gate Array(FPGA) and PYNQ DPU module [1-3] provide us an efficient way to reduce such cost by hard-ware acceleration of the model and overcome this problem. In this work, we propose a Unmanned Aerial Vehicle(UAV) tracking system on ultra96 v2 PYNQ board based on the combination of object detection deep learning model and mathematical algorithm such as Kalman filter. The Kalman filter provides us a fast but preliminary estimation of UAV position while the object detection models further localize the UAV. Our implementation meets 20 fps on the Ultra96 v2 board and nearly achieve the real-time detection.

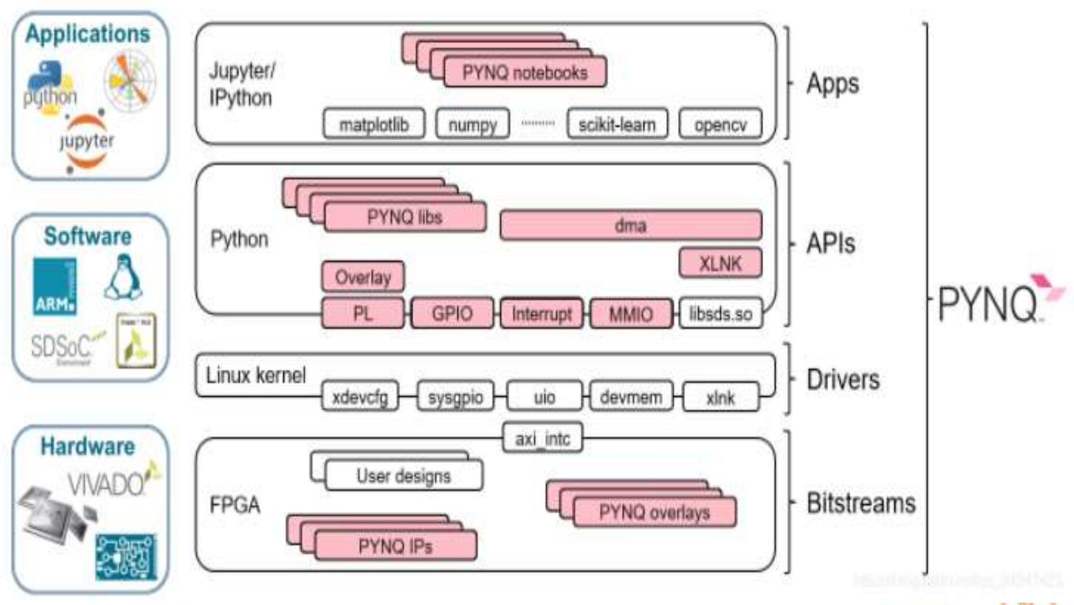
Problem description

Motivation

The real-time UAV detection becomes more and more ubiquitous in many application domains. Although the real-time detection can be achieved by GPU device and efficient network architecture, it becomes a challenge on the edge device. Therefore, the software and hardware acceleration should be considered in such scenario. In this work, we utilize Field Programmable Gate Array(FPGAs) that offers hardware acceleration and the Kalman filter as a preliminary detection to aid the detection in the the software aspect. We will implement our real time UAV tracking system based on Xilinx Zynq UltraScale+ MPSoC development board.

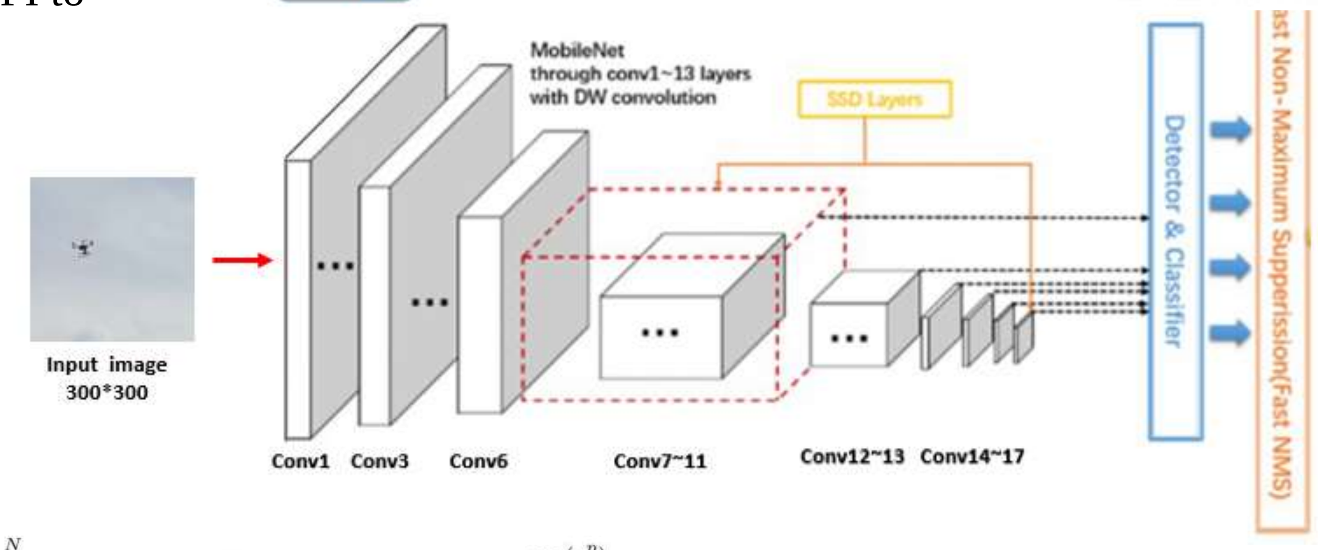
PYNQ board

The advantage of using FPGAs is the inherent parallelism. Comparing to the conventional parallelism on the multiple CPU devie, the parallelism of FPGA is automatically utilized without any software adaption. PYNQ is developed based on FPGA Zynq devices to design programmable logic circuit. One can simply control FPGA accelerated library by python API to achieve AI model acceleration.



Method

The progress of our tracking system can be divided into two parts: Training SSD Mobilenet V1 on the GPU device and the deployment onto PYNQ. Our SSD Mobilenet V1 is trained based on CenekAlbl drone dataset [9]. The architecture of SSD Mobilenet V1 is composed of Mobilenet as feature extractor and SSD head, consists of convolution layers, followed by the feature extractor. Mobilenet first extracts features of specific object, then SSD head will generate the possible bounding boxes and their corresponding confidence scores. The loss function of SSD Mobilenet can be divided into two parts, confidence loss and localization loss.



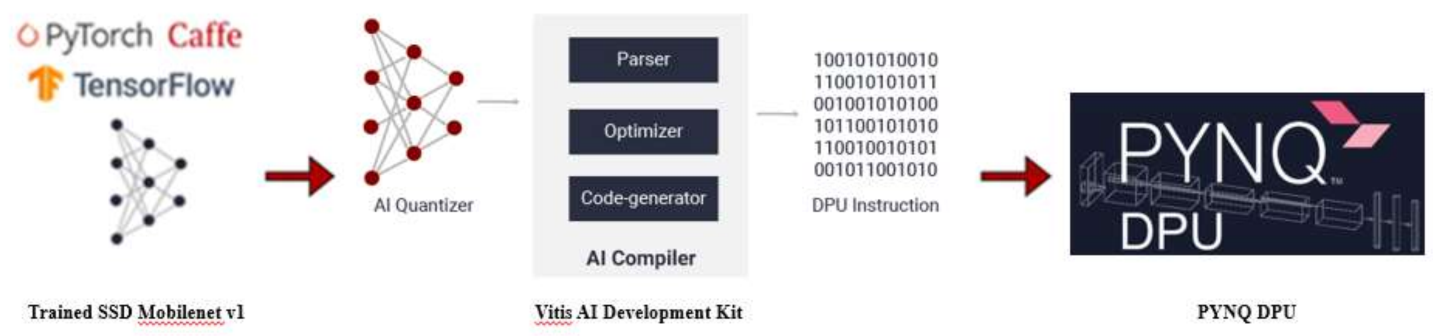
$$L(x,c,l,g)=\frac{1}{N}(L_{conf}(x,c)+\alpha L_{loc}(x,l,g))$$,where

$$L_{loc}(x,l,g)=\sum_{i\in Pos}^{N}\sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k}\mathrm{smooth_{L1}}(l_i^m-\hat{g}_j^m)$$
$$\hat{g}_j^{cx}=(g_j^{cx}-d_i^{cx})/d_i^{w}\qquad \hat{g}_j^{cy}=(g_j^{cy}-d_i^{cy})/d_i^{h}$$
$$\hat{g}_j^{w}=\log\left(\frac{g_j^w}{d_i^w}\right)\qquad \hat{g}_j^{h}=\log\left(\frac{g_j^h}{d_i^h}\right)$$

and

$$L_{conf}(x,c)=-\sum_{i\in Pos}^{N}x_{ij}^{p}log(\hat{c}_i^p)-\sum_{i\in Neg}log(\hat{c}_i^0)\quad \text{where}\quad \hat{c}_i^p=\frac{\exp(c_i^p)}{\sum_p\exp(c_i^p)}$$

The next part is to deploy our SSD Mobilenet V1 model on DPU PYNQ v1.2. The deploying part can be divided into two steps, quantization and compilation. We compile the quantized model to create its corresponding .elf file ready for execution on DPU accelerator IP. After getting the execution elf file, we are able to implement accelerated SSD Mobilenet v1 on PYNQ DPU.



Kalman filter

Kalman filter[10], also known as linear quadratic estimation, is an mathematical algorithm that uses a series of measurements observed overtime, containing statistical noise and other inaccuracies, and generates predictions of unknown variables that tend to be more accurate than those based on only one measurement alone. The equations of motion and observation are as follows:

$$x(k+1)=F(k)x(k)+G(k)w(k)$$
$$z(k)=H(k)x(k)+v(k)$$

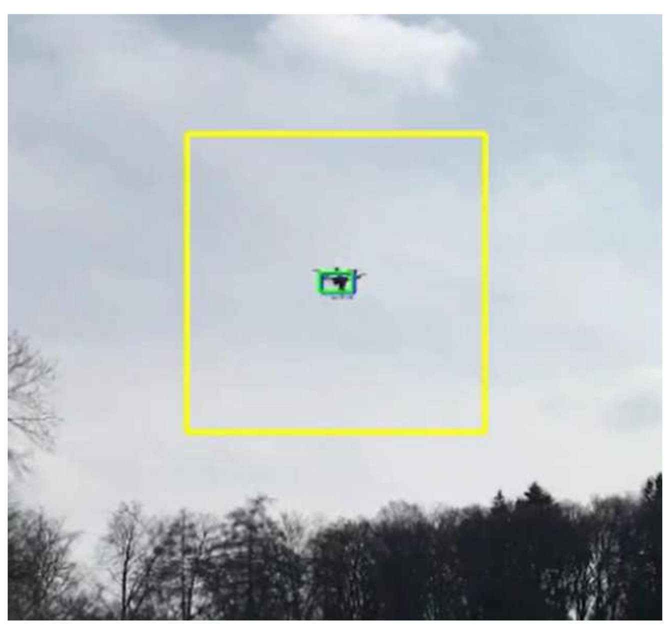
,where F(k) is the state transition matrix
B(k) is the control-input model
w(k) is the process noise
H(k) is the observation model
v(k) is the observation noise

The update equation are

$$y=F\hat{x}$$
$$S=FPF^T+GQG^T$$
$$\Gamma=S\,H^T\,[H\,S\,H^T+R]^{-1}$$
$$\hat{x}=y+\Gamma[z-Hy]$$
$$P=[I-\Gamma H]S$$

,where Γ is the filter gain

Results and discussion



Results

The SSD Mobilenet V1 model is trained on CenekAlbl drone dataset with 20000 images. The performance of SSD Mobilenet V1 was evaluated on the testing dataset which is about 25% of the collected data. It reaches a good detection accuracy of 90% with the speed of 24.5ms per image. After deploying our model on PYNQ DPU, the speed of whole tracking including data preprocessing, inference, post processing and displaying can even achieve about 20fps.

Discussions

In practical, the video resolution such as 720p,1080p,2K or even 4K is often higher than 300*300. However, there will be a problem on PYNQ DPU while inferencing the testing image with size that is not equal to 300*300 in our research. Although we are able to crop the image into many small images with size 300*300, the computation cost will be much more expensive. As a result, we try to combine SSD Mobilenet V1 with Kalman filter in order to reduce computation cost and achieve higher fps.

Conclusions

In this work, a unmanned aerial vehicle tracking system is proposed based on PYNQ DPU for real-time detection. SSD Mobilenet V1 is applied to detecting UAV which can capture the small UAV precisely. The speed of detection can achieve 24.5ms per image and the speed of whole tracking system including showing the object prediction box in real time can achieve 20fps. Meanwhile, we combine the SSD Mobilenet V1 with Kalman filter in order to achieve faster performance and improve the flexibility of tracking system. The experiment results demonstrate that SSD Mobilenet V1 has good performance, speed and flexibility on PYNQ DPU.

References

1. http://www.pynq.io/
2. https://ultra96-pynq.readthedocs.io/en/latest/getting_started.html
3. https://pynq.readthedocs.io/en/v2.6.1/
4. https://github.com/Xilinx/Vitis-AI/tree/master/models/AI-Model-Zoo
5. Https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/ultra96-v2/
6. https://github.com/Xilinx/PYNQ-ComputerVision
7. https://github.com/Xilinx/DPU-PYNQ
8. https://github.com/tensorflow/models/tree/master/research/object_detection
9. https://github.com/CenekAlbl/drone-tracking-datasets
10. https://en.wikipedia.org/wiki/Kalman_filter
11. Implementation of a tracking Kalman filter on a digital signal processor
J Tan, N Kyriakopoulos - IEEE transactions on industrial electronics, 1988
12. FPGA-Based Implementation of Kalman Filter for Real-Time Estimation of Tire Velocity and Acceleration
F Sandhu, H Selamat, SE Alavi, VBS Mahalleh - IEEE Sensors Journal, 2017