Detection and Implementation of Driver’s Seatbelt Based on FPGA

Wu Tianshu¹², Zhang Zhijia¹*, Liu Zhanning¹, Liu Yunpeng² and Wang Shixian¹
¹ School of Information Science and Engineering, Shenyang University of Technology
² Shenyang Institute of Automation, Chinese Academy of Sciences
*Email: zzjsut@126.com

Abstract. In this paper, the driver’s seatbelt detection algorithm is transplanted to the PYNQ embedded platform of XILINX to meet the technical requirements of automatic identification and detection of whether the driver wears a seatbelt or not. In this paper, aiming at the characteristics of the hardware of the FPGA, the driver’s seatbelt detection algorithm is realized by IP core from XILINX open source. Vehicle detection is realized by using YOLO object detection algorithm with 3 bit weight and 1 bit activation, and driver’s seatbelt classification is realized by using binary model with 1 bit weight and 1 bit activation. On the PYNQ embedded hardware of XILINX, the acceleration of the algorithm is realized by calling the hardware of FPGA on the ARM side.

1. Introduction
In recent years, with the increasing number of motor vehicles in China, the problem of traffic safety has become increasingly serious. When a traffic accident occurs, the driver should fasten his seatbelt to protect the driver’s life as much as possible. However, some drivers have a weak sense of safety and do not wear seatbelts during driving. At present, in order to urge drivers to wear seatbelts, the public security departments mainly use manual methods to analyze traffic monitoring images to determine whether drivers wear seatbelts or not, but the manual detection efficiency is low and the cost is high. With the development of computer vision, automatic seatbelt detection based on image recognition has become an inevitable trend.

Documents on driver’s seatbelt detection have been published since 2013. At present, the literature mainly realizes seatbelt detection through image classification [1] [2]. Although the method of image classification can detect whether the driver wears seatbelt and can balance speed and accuracy, there are limitations in using this method to complete seatbelt detection. Whether a driver wears a seatbelt or not is essentially a classification problem, but the driving scene is complex. Whether a driver wears a seatbelt or not can not be the main difference between positive and negative samples.

With the use of deep learning for image recognition, more and more applications have been made in the field of practical engineering. In the aspect of traffic safety monitoring and detection, using convolution neural network to detect has become a new mainstream development direction. In the field of object detection, Ross Girshick et al. proposed RCNN [3] in 2014, achieving an average precision of 53.7% on PASCAL VOC data set, while DPM [4] algorithm using manual features such as HOG has an average precision of 35.1%. However, RCNN has the problem of slow detection speed because it is not end-to-end detection. The detection speed is only 47 seconds per frame. This is because about 2,000 suggestion boxes derived from selective search algorithm [5] in each frame need to be computed once
by forward propagation of convolutional neural network, which covers the repeated computation of the same region in a frame.

In 2015, Ross Girshick et al. proposed an improved Fast-RCNN algorithm based on RCNN. With the same average accuracy, the detection speed is increased to 0.32 seconds per frame [6]. In the same year, Chinese scholar He Kaiming proposed Faster-RCNN. Faster-RCNN uses simple network structure to extract features. The target detection speed reaches 17 FPS and the complex network can reach 5 FPS [7]. But Faster R-CNN still has the problem of slow detection speed. In 2018, Wu T et al. proposed Deconv-SSD vehicle detection algorithm and Squeeze-YOLO driver region location algorithm [8]. Deconv-SSD object detection algorithm can effectively improve the accuracy of vehicle detection, Squeeze-YOLO algorithm can achieve faster driver area positioning speed with high accuracy.

In the field of semantic segmentation, Jonathan L et al. proposed FCN [9] at the CVPR conference in 2015. Full-convolution convolution neural network was used to realize semantic segmentation, which won the CVPR Best Paper Award. However, there are many problems in current semantic segmentation algorithms, such as large amount of computation, long detection time and so on. Taking Cityscapes test set as an example, the semantics segmentation speed of 8 times sampling FCN is 0.5 seconds per frame, which cannot satisfy the application of semantics segmentation algorithm in practical engineering. Paszke A et al. proposed ENet [10] in 2016, which uses a Resnet-like structure and a continuous asymmetric convolution structure to reduce computational complexity. In 2019, Wu T et al. proposed to judge whether the driver wears a seat belt by FCN semantic segmentation algorithm, and to detect the driver’s seatbelt by judging the threshold of the maximum connected area [11].

In this paper, the driver’s seatbelt detection algorithm is transplanted to the PYNQ embedded hardware platform of XILINX, and the hardware acceleration of the algorithm is implemented on the FPGA using XILINX open source IP core. The driver’s seatbelt detection algorithm is implemented on the embedded platform of low power consumption and low cost, which is of great practical significance to enhance driver’s safety driving awareness and enhance traffic safety.

2. Algorithms design
Because of the large size of the nearby vehicle target, the object detection algorithm does not need multi-scale feature integration, and does not need to consider the phenomenon of low recall of small targets. Multi-scale feature integration and multi-resolution feature map detection will only bring redundant calculation. Therefore, the basic target detection loss function of YOLO V1 is selected for vehicle detection and front windshield detection.

In vehicle detection, this paper uses XILINX to develop QNN IP core for PYNQ platform. Because FPGA is not suitable for calculating floating-point numbers, QNN uses integer quantization to quantify the weight to 1 bit weight, so it can carry out bit operation according to the characteristics of FPGA hardware.

QNN implements convolution calculation with 1 bit weight and 3 bit activation. The loss function of target detection adopts the design idea of Yolo V 1. The front-end feature structure is improved according to tiny yolo. Among them, 1 bit weight quantization is based on BNN binary weight algorithm, and 3 bit activation maps relu activation function directly.

BNN is Matthieu Courbariaux equivalent to the binary network proposed in 2016 [12]. BNN optimizes the convolution neural network in both vertical and horizontal directions. Horizontally, the convolution weight is quantized to -1 and 1 values by means of symbolic function, and then the scale factor is designed to make the quantized weight and the original floating-point level weight the closest to the fitting result of the feature graph, so as to achieve the same fitting ability, as shown in equation (1).

\[ I * W \approx (I \oplus B)\alpha \] (1)

Among them, the input feature graph and floating-point level weights are represented respectively, which are binary weights, and the symbolic calculation of the feature graph and the binary weights is represented.
According to equation (2), in order to minimize the reconstruction error of feature fitting ability, the optimal objective function can be obtained, as shown in equation (2).

\[ J(B, \alpha) = \|W - \alpha B\|^2 \]  

(2)

The minimum sum of time is obtained by optimizing the objective. Equation (2) is expanded to obtain equation (3).

\[ J(B, \alpha) = \alpha^2 B^T B - 2\alpha W^T B + W^T W \]  

(3)

The sum is a constant. Since there are only 1 and -1 values, equation (3) can be converted into equation (4), and the optimal binary convolution layer can be obtained. It represents the optimal proportion coefficient of convolution layer weight, and N represents the dimension of convolution core.

\[ \alpha^* = \frac{W^T \text{sign}(W)}{n} = \frac{1}{n} \|W\|_1 \]  

(4)

In the direction of vertical gradient propagation, in order to ensure the accuracy of chain bias back-propagation, the floating-point gradient can still be propagated, which acts on the floating-point weight of convolution layer.

Because the input image needs to retain the original features, the last layer of feature map needs to output the location of the target box, so there is no Binarization in the first layer and the last layer of convolution layer, using 8-bit weight and 8-bit activation value to shape the offline quantization.

After getting the vehicle test results, because the windshield occupies a large proportion of the whole picture and has obvious features, this paper uses XNOR feature extraction structure, and uses YOLO V1 activated by 1 bit weight to detect the windshield. Because XILINX does not have an open source YOLO model with 1 bit weight and 1 bit activation deployed on the FPGA side, but only an open source XNOR classification algorithm, this paper simulates the speed of YOLO with 1 bit weight and 1 bit activation on the FPGA side, and runs on the ARM side when it runs.

When the weight and activation are both 1 bit, XOR bit operation can be used to calculate the convolution layer. FINN determines the activation value by judging the structure of the batch normalization layer and saves computing resources.

Since the activation value is 1 bit and belongs to the symbolic function, the activation output value can be determined directly by judging whether the output value of the batch normalization layer is greater than 0 after calculation. The threshold can be derived from the batch normalization formula, as shown in equation (5).

\[ \tau_k = \mu_k - (B_k / (\gamma_k \delta_k)) \]  

(5)

Among them are characteristic mean, characteristic variance and batch normalization layer coefficient. By judging the activation value by threshold, the hardware resources of the FPGA are saved.

3. Hardware environment construction and result analysis

This section introduces the basic environment of embedded hardware of FPGA, designs the seatbelt detection algorithm based on XILINX open source IP core, realizes low power consumption and low cost operation on embedded Soc, and introduces the accuracy and running speed of the seatbelt detection algorithm on PYNQ.

3.1. Embedded hardware environment of FPGA

This paper uses PYNQ Z2 embedded SOC developed by XILINX company. PYNQ Z2 is improved on the basis of ZYNQ 7020. It can use Python to call IP core. And XILINX designed the jupyter notebook development environment for PYNQ Z2. Jupyter notebook is a browser-based development environment that supports more than 40 languages. PYNQ series embedded devices support converting IP cores developed by HDL or HLS into overlay, and PYNQ provides Python library to call overlay to achieve the purpose of Python calling IP cores [13].

In PYNQ, the lower part of each frame input image is cut out, and only the cut area input algorithm is used for seatbelt detection. Wait until the first half of the vehicle travels forward to the area before
being detected by the algorithm, there is no phenomenon of missed detection. Because the second half of the clipping is close to the traffic safety surveillance camera, the vehicle target size is large, which reduces the difficulty of the target detection algorithm and the seatbelt segmentation algorithm. At the same time, it reduces the resolution of the input image and meets the need of running on the embedded platform.

3.2. Experimental result
In ZYNQ and PYNQ Soc series, the hard core of DSP48E type is included. The hard core of DSP48E is different from the chip of DSP. The hard core of DSP48E is a programmable circuit. Different calculation circuits can be made according to the need. In BNN and QNN, multiple DSP48Es are used for parallel computation to realize the MAC calculation of SIMD. The resource requirements for hardware design of BNN and QNN are shown in Table 1.

| Hardware | LUT    | BRAM | DSP48E | FLIP FLOP |
|----------|--------|------|--------|-----------|
| BNN      | 26072  | 124  | 24     | 41312     |
| QNN      | 46888  | 140  | 57     | 43139     |

Table 1. Comparison of Optimization effects of GPU

Using QNN IP core, vehicle detection is trained and tested on PASCAL VOC data set up to 50.1% mAP. Although it is lower than 79.6% of the 32-bit weight training of float in GPU, there will be no missing detection because only vehicles close to traffic cameras are detected. Compared with GPU, the mAP decreases by 29% mainly because of the lack of multi-scale feature fusion and multi-resolution feature map detection for small object detection. Under the environment of jupyter notebook, the effect of vehicle detection using PYNQ is shown in Figure 1.

Figure 1. PYNQ Z2 vehicle detection

After getting the vehicle test results, the windshield is detected. Because the windshield features are obvious, xornet with 1 bit weight and 1 bit activation is used to detect the windshield. The detection effect is shown in Figure 2.

Under the Darknet deep learning framework supporting XNOR operation, YOLO, whose front-end structure is XNOR feature extraction, can train and test 99% mAP by using self-made windshield detection data set, which is similar to the precision of 32-bit float detection in GPU. The main reason is that the windshield features are obvious, so the detection difficulty is low. Because of XOR bit operation and low resolution of input image, the speed of vehicle detection based on QNN IP core is obviously improved.

Because XILINX does not have the conversion model of YOLO deployed on the FPGA with XNOR as the open source feature extraction structure, this paper runs the same input resolution XNOR classification algorithm on the FPGA to simulate the YOLO speed with XNOR as the front end, and in
practice, the windshield detection is completed on the ARM side. It takes about 0.0016s to run XNOR feature extraction structure detection on FPGA.

Figure 2. Detection of PYNQ Z2 windshield

The right half of the windshield is taken as the driver’s area, and the driver’s area is classified. Because of the two classifications of 1 bit weight and 1 bit activation, the seatbelt detection on the FPGA is less than that of the float 32 bit weight on the GPU. The accuracy of the seatbelt detection on the FPGA is about 81%, which is lower than that of the semantic segmentation algorithm about 94%. However, the accuracy of seatbelt detection using VGG-16 and other classification algorithms in other literatures is similar.

Seatbelt detection algorithm on FPGA consumes 0.00158s at the low cost PYNQ end with a power consumption of only 2.5W, which is about 633 FPS. The speed of semantics segmentation algorithm after pruning in high power and high cost GPU is only 305 FPS, which reduces the accuracy, but achieves fast operation with low power and low cost. The effect of seatbelt detection is shown in Figure 3.

Figure 3. PYNQ Z2 seat belt test
4. Conclusion
In this paper, on the PYNQ embedded hardware of XILINX, the acceleration of the algorithm is realized by calling the hardware of FPGA on the ARM side. The driver’s seat belt detection algorithm is designed and transplanted to the PYNQ embedded hardware platform of XILINX. The driver’s seat belt detection algorithm is implemented on the low power and low cost embedded platform by using XILINX open source IP core to accelerate the algorithm on the FPGA.

Acknowledgement
This work was financially supported by Key Projects of the Joint Fund of the Chinese Academy of Sciences(Y8K4160401), Shenyang Artificial Intelligence Key Laboratory Fund and National Natural Science Foundation of China(61540069).

References
[1] Fu C 2015 Research on seatbelt detection method based on deep learning (Wu Han: Huazhong University of Science and Technology)
[2] Yang K, Zhang D and Yang L 2017 Safety belt detection based on deep learning Journal of China University of Metrology 28 326-33
[3] Girshick R, Donahue J, Darrell T, et al 2014 Proc.of the IEEE Conf. on computer vision and pattern recognition (Washington) vol 1 (DC:IEEE Computer Society Washington) p 580
[4] Zeng J and Cheng X 2016 Pedestrian detection combined with single and couple pedestrian DPM models in traffic scene Acta electronic sinica 44 3268-75
[5] Uijlings J, Sande K, Gevers T and Smeulders A 2013 Selective search for object recognition International journal of computer vision 104 154-71
[6] Girshick R 2015 Proc. of the IEEE Int. Conf. On computer vision (Santiago) vol 1 (Santiago: IEEE) p 1440
[7] Ren S, He K, Girshick R and Sun J 2015 Faster r-cnn: Towards real-time object detection with region proposal networks Advances in neural information processing systems(Montreal) vol 39(MA: Massachusetts Institute of Technology Press) p91
[8] Wu T, Zhang Z, Liu Y, Pei W and Cheng H 2018 A lightweight small object detection algorithm based on improved SSD Infrared and Laser Engineering 47 703005
[9] Long J, Shellehammer E and Darrell T 2015 Proc. of the IEEE Conf. on computer networks for semantic segmentation (Santiago) vol 1 (Santiago: IEEE) p 3431
[10] Paszke A, Chaurasia A, Kim S and Culurciello E 2016 Enet: A deep neural network architecture for real- time semantic segmentation preprint arXiv:1606.02147
[11] Wu T, Zhang Z, Liu Y, Guo W and Wang Z 2018 Driver seatbelt detection based on YOLO detection and semantic segmentation Journal of Computer-Aided Design & Computer Graphics 31 126-31
[12] Courbariaux M, Hubara I, Soudry D El-Yaniv R and Bengio Y 2016 Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1 Preprint arXiv:1602.02830
[13] Pan X 2003 Moving target detection(Hang Zhou: Zhejiang University)