DOI:10.19651/j.cnki.emt.2315145

# 基于 FPGA 的道路标识检测系统设计*

王新伟[1,2] 丁红昌[1,2] 曹国华[1,2]
(1.长春理工大学机电工程学院 长春 130022; 2.长春理工大学重庆研究院 重庆 401135)

**摘 要**: 为达到道路标识实时检测的要求,针对目前主流的目标检测算法在图像处理器上存在模型参数量大、实时性差、功耗大和成本高的问题,提出一种基于 FPGA 的道路标识实时检测方案。为减少参数量、提高检测速度,采用 YOLOv3-tiny 作为特征提取网络,进行权重参数的训练与优化;将模型浮点数参数量化为 8 位定点数,并将量化后的网络模型在 FPGA 上完成部署实验。实验结果表明,在 Yolov3-tiny 网络检测速率上,本系统对实验数据集的测试帧率可达到 153 fps,功耗为 4.92 W,峰值 GOP/s 为 115GOP/s。该系统可以满足实时目标检测的要求,并且能够在低功耗的状态下实现系统的部署。

**关键词**: 目标检测;YOLO;硬件加速;FPGA
**中图分类号**: TP391 **文献标识码**: A **国家标准学科分类代码**: 520.6040

# Design of road marking detection system based on FPGA

Wang Xinwei[1,2] Ding Hongchang[1,2] Cao Guohua[1,2]
(1. School of Mechanical Engineering, Changchun University of Science and Technology, Changchun 130022, China;
2. Changchun University of Science and Technology Chongqing Research Institute, Chongqing 401135, China)

**Abstract**: In order to meet the requirements of real-time detection of road signs, for the current mainstream target detection algorithms on the image processor there are a large number of model parameters, poor real-time performance, high power consumption and high cost, a real-time detection of road signs based on FPGA is proposed. In order to reduce the number of parameters and improve the detection speed, YOLOv3-tiny is used as the feature extraction network for the training and optimization of the weight parameters; the model floating-point parameters are quantized into 8-bit fixed-point numbers, and the quantized network model is used to complete the deployment experiments on the FPGA. The experimental results show that at the Yolov3-tiny network detection rate, the test frame rate of this system for the experimental dataset can reach 153 fps, the power consumption is 4.92 W, and the peak GOP/s is 115GOP/s. This system can satisfy the requirement of real-time target detection, and it can realize the deployment of the system under low power consumption.

**Keywords**: object detection;YOLO;hardware acceleration;FPGA

## 0 引 言

随着新能源汽车的快速发展,中国的汽车产业发生了大的结构调整,上下游的产业链迸发了许多生机与活力,包括电池技术、电机技术、充电技术等都有了长足的进步。自动驾驶技术也迎来了高速发展,2021 年 8 月 6 日由交通运输部、工业和信息化部、公安部联合发布了《智能网联汽车道路测试与示范应用管理规范(试行)》。截至目前全国共建设有 16 个智能网联汽车测试示范区,开放了 3 500 多千米的测试道路,共发放 700 余张测试牌照。

自动驾驶汽车(AVs)的快速发展推动了现有交通模式的新视角和潜在的挑战。目前,第 3 级及以下级的驾驶辅助系统已被广泛应用[1],第 4 级系统对特定情况的几种应用也逐渐发展。通过提高自动化水平和车辆智能,这些系统可以进一步推进到完全自动驾驶。然而,第 5 级自动驾驶的开发仍处于起步阶段,可以肯定的一点是,其对系统的要求会更为复杂,更为严格。这也对系统的整体功耗提出了相当高的要求。

为应对低功耗的实时监测,目前有云计算[2]、检测算法优化[3]、硬件搭建等解决方案。首先云计算方面,相对其他方案具有更便捷、更高效、适配性强等特点[4],但其对网络

收稿日期:2023-12-11
* 基金项目:173 计划技术领域基金(B 类)(2022-JCJQ-JJ-0257)、重庆市自然科学基金(2022NSCQ-MSX0340)项目资助

延迟及带宽有严格要求，在特殊状况下如：长隧道，偏远地区，受灾地区等状态下。无法实现系统的功能。其次检测算法方面，包括轻量化网络结构设计和模型压缩两种方法[5]，其中轻量化网络结构设计主要是对模型进行轻量化处理，改变现有的网络结构，以减少检测过程中的运算量，模型压缩包括知识蒸馏、剪枝、量化、低秩分解四类[6]，通过以上两种方法，可以达到降低系统功耗的目的，但由于其本质上仍需使用 CPU 或 GPU，在功耗方面偏高，最后硬件搭建方面，目前主流的方式有 STM32[7]，Jetson[8]，FPGA 等，相对于其他两种部署方案，FPGA 具有可扩展性强、运算速度快、可操作性强等优点[9]。

早在 2015 年，Han 等[10]设计了定值的权重参数系统，其设计的 FPGA 加速器系统采用定点计算代替浮点计算，优化了卷积计算过程，有很好的借鉴意义。目前主流的 FPGA 加速系统设计分为两个方向：分别是采用高层次综合(HLS)编译和寄存器传输级(RTL)代码编译。张丽丽等[11]针对 YOLOv3-SPP 网络模型、李伟琦[12]针对 YOLOv3-tiny 网络模型、戴伟杰等[13]针对 YOLOv2 网络模型采用高层次综合的方法进行了加速器设计。采用高层次综合设计具有开发速度快，开发流程简单的优点。但相比于胡永阳等[14]以及梅志伟等[15]提出的，针对 YOLOv3-tiny 网络模型采用寄存器传输级(RTL)代码编译的加速器还是存在一定的不足，前者系统设计思路与本文相似，但未完成系统设计，后者代码会占用大量冗余资源，降低系统的运行速度，提升系统功耗，没有将 FPGA 的工作效率差最大化。

综上所属，本文提出以 YOLOv3-tiny 网络结构为基础，以道路标示符检测为应用目的的目标检测网络，并且在 FPGA 采用寄存器传输级(RTL)代码编译部署，且使系统较以上文献所述系统检测速率有较大提升。

## 1　卷积算法优化

### 1.1　YOLOv3-tiny 框架搭建

2018 年，作者 Redmon 在经典的 YOLOv2 的基础上做了一些改进。特征提取部分采用 darknet-53 网络结构，利用特征金字塔网络结构实现了多尺度检测，分类方法使用逻辑回归代替了 softmax，在兼顾实时性的同时保证了目标检测的准确性[16]。

本次方案中选用的 YOLOv3-tiny 是目前工业生产项目中的首要选择，这个网络的原理与 YOLOv3 相似，就是在 YOLOv3 的基础上去掉了一些特征层，只保留了 2 个独立预测分支，具体的结构如图 1 所示。

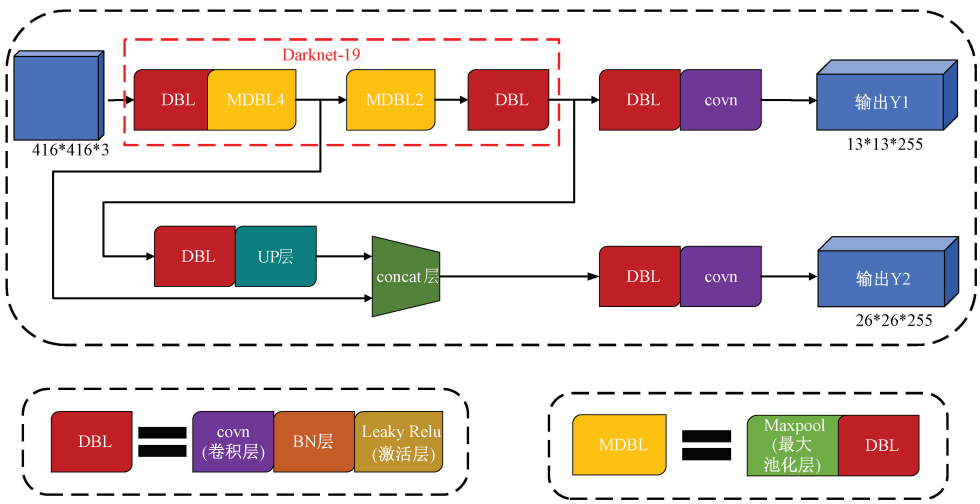


图 1　YOLOv3-tiny 网络结构

### 1.2　数据集介绍

根据本文所研究方向，目前暂无较为适配的开源数据集，所以为了满足实验要求，特建立了一个包含多特征、多场景、多环境的数据集，其中图片来源包括网络开源数据集图片、互联网全景地图、实景拍摄的部分图片等，具体来源如图 2。

由于数据集中大部分素材来源于国外，故加入百度全景地图中的国内街景图片，以提升模型对于国内环境的适应性。实景图片主要由非城镇街道和高速公路实景图片组成，也包含一些少量的城市街道补充图片，两者在开源数据集中很少呈现，所以此次实验特别加入两者以提升模型兼容度。

数据集中图像数据存放在 images 文件夹，使用 Labelimg 软件标注生成的 TXT 文件(含类别和 anchor 的坐标)存放在 labels 文件夹。数据中包含，不同时间段(早晨、上午、中午、下午和黄昏)及不同环境下(晴天、雨天、和雾天)状态图像，部分图像如图 3 所示。

数据集中除开源图像外，实景图片采用相机在车辆行驶状态下拍摄，图像格式为标准的 JPG 格式。然后对互联网开源图片及拍摄的实景图像进行整理，剔除不符合要求的图像，如抖动导致图像模糊不清等，最终得到 9 000 张图像。使用 Labelimg 开源软件对图像进行标注，标注存储格

式为 YOLOv3-tiny 支持的 TXT 文档。数据集共包含 9 000 张图片,其中训练集占 70%,测试集占 20%,验证集占 10%。

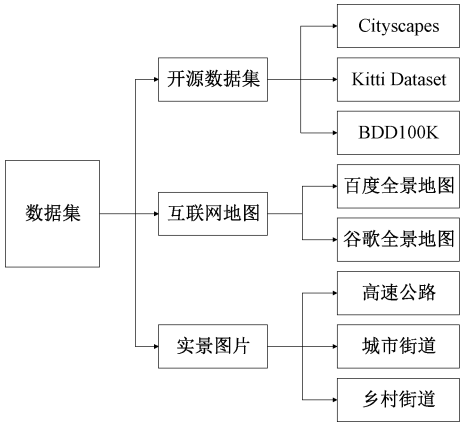


图 2　数据集结构

图 3　数据集样例

该数据集为此次实验实时目标检测研究项目提供了数据支撑,用以实现项目中深度学习训练过程,构建适用于此次实验的卷积神经网络模型。

### 1.3　定点数处理

卷积神经网络在训练阶段为确保网络的收敛性和准确性通常采用 32 位浮点数,而在使用 FPGA 进行卷积神经网络的硬件加速实现时,受限于 FPGA 片上资源,往往需要将训练过程中的浮点数转换为定点数,从而减少资源的消耗,进而降低整个系统的功耗,所以此次实验中采用 8 位动态定点数量化[17],定点数 $x_{fixed}$ 可由式(1)表示:

$$x_{fixed} = \sum_{i=0}^{b_w-1} B_i 2^{-p} 2^i \tag{1}$$

式中:$b_i$ 的取值范围为 0 到 1;$b_w$ 是定点数 $x_{fixed}$ 的位宽;$p$ 是阶码。在定点数 $x_{fixed}$ 中最高位是符号位,通常使用补码来表示。浮点数 $x_{float}$ 与定点数 $x_{fixed}$ 的相互转化公式如式(2)~(3)所示。

$$x_{fixed} = (\text{int})(x_{float} 2^{b_w}) \tag{2}$$

$$x_{float} = (\text{float})(x_{fixed} 2^{b_w}) \tag{3}$$

在定点数中阶码保持不变,且位宽也是有限制的,在训练阶段,通过将浮点数转换为定点数可以在保持计算精度的情况下节省一定资源。

## 2　模型编译与部署

### 2.1　整体架构设计

YOLOv3tiny 网络结构如图 4 所示,主要由 4 个部分组成,即输入端、Darknet-19 主干网络、FPN 网络和输出

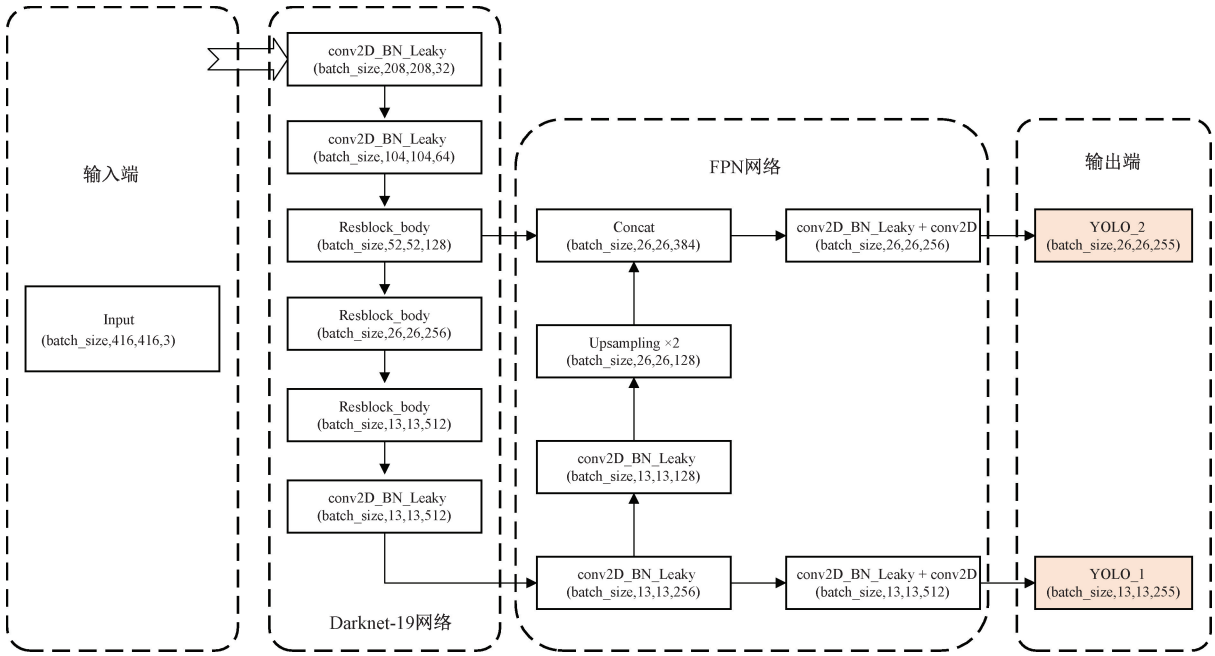


图 4　YOLOv3-tiny 卷积架构

端[18]。网络结构中，运算部分集中在 Darknet-19 主干网络和 FPN 网络中，Darknet-19 主干网络进行输入特征图数据与权重参数的卷积运算，FPN 网络负责对卷积运算之后的结果进行分类，进而在两条预测分支上输出两种预测结果。可以看出网络模型中卷积运算占据了绝大部分的运算量，所以本文设计系统主要针对卷积运算过程优化，以加速 YOLOv3-tiny 网络的前向推理速度，具体的加速器架构如图 5 所示。

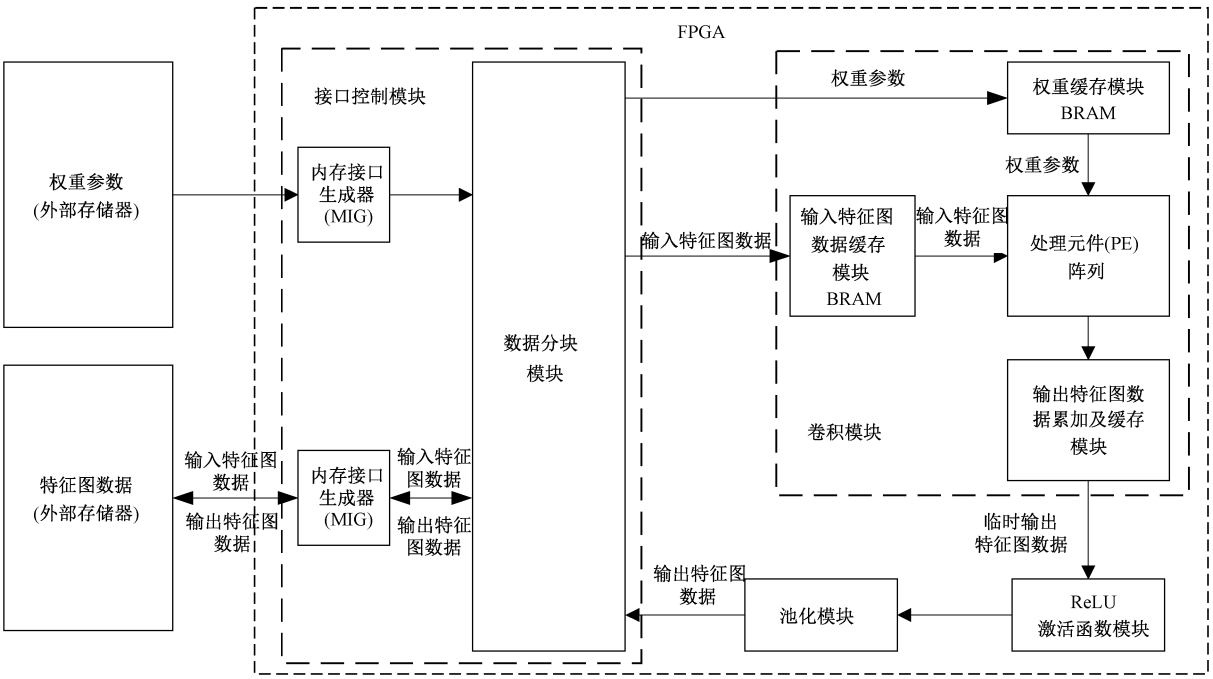


图 5　加速器架构

在图 5 所示的加速器架构中，由于实时检测系统中，权重参数及特征图数据的数据量较大，本文所选型号 FPGA 无法将全部权重参数及特征图数据存储在 FPGA 内部，所以使用外部存储器将其储存，再通过接口控制模块中的控制内存接口生成器(MIG)核调度外部存储器与片上缓存块随机存取存储器(BRAM)之间数据的存储与访问；BRAM 片上缓存资源，由于 FPGA 内部构造原因，系统所需数据需要在 BRAM 上先进行缓存，然后经由 BRAM 与计算单元进行数据交换。处理元件(PE)阵列，通过大量组合的处理原件阵列来完成特征图数据和权重参数的卷积运算。

此次设计的加速器的工作流程如下：首先将一小部分特征图数据及权重参数经由接口控制模块导入 FPGA 内部，将其分别缓存在输入特征图数据缓存和输入权重参数缓存上；然后将特征图数据缓存和输入权重参数缓存上的数据导入处理原件阵列进行计算，再将计算后的数据存储在输出特征图数据累加及缓存模块中。在输出特征图数据累加及缓存模块中的数据满足数据输出条件时，将数据传输到 ReLU 激活函数模块及池化模块中进行处理，再通过接口控制模块将 FPGA 内部的输出特征图数据传输到外部存储器中，完成目标检测的前向推理。

## 2.2　卷积模块设计

在深度学习网络模型中，整个网络模型运算量中最大的就是卷积运算，会占用大量的存储空间以及计算单元，所以此次设计，主要对针对卷积过程进行了计算优化。主要从循环优化、数据分块、数据复用 3 个方面对卷积过程进行加速优化，进而实现系统的快速检测。

1)循环并行展开设计

卷积计算过程中优化主要包含输入通道循环、输入特征图循环、卷积核循环和输出通道循环这 4 种循环。这 4 种循环共同影响着卷积运算的过程，其优化过程直接影响卷积运算的流水线架构，进而影响系统设计中的存储访问模式及数据复用操作。

在实际的设计过程中，受限于本文设计所选用的 FPGA 型号，无法全部优化上述四种循环过程，根据 YOLOv3tiny 网络结构特点，在实际的优化过程中，采用输入通道循环展开和输出通道循环展开，将两者的并行度设为 16，每次运行将会占用 256 个 DSP 单元。

2)卷积计算架构

根据上文所述，卷积模块采用输入通道循环展开和输出通道循环展开，在此架构中，采用了 16×16 的计算单元布局，共需 256 个 DSP 单元。根据卷积模块架构，为高效处理权重数据，本次设计中共设计了 16 个权重缓存队列，每个队列内配备 16 个 BRAM 权重缓存单元，总计 256 个单元。每个队列独立更新对应计算矩阵列的权重，从而实现整个计算矩阵的高效权重更新。

在输入特征图缓存的设计方面,采用了一种逐点递进的更新机制,保证每个输入通道能够有效地更新其特征点,并将更新后的数据复制给相应计算单元。由于计算矩阵包括 16 个输入通道,这使得系统可以一次更新 16 个特征点。完成权重和特征点的更新后,计算单元内的乘法器将这些数据相乘,并存储在中间结果缓存队列中。

3)数据分块与数据复用

由于 FPGA 内部存储资源限制,所以此次设计系统数据读取过程采用数据分块方法,即将输入特征图数据划分为更小的数据块,再将划分好输入特征图数据逐块加载到 FPGA 内部,进行逐块运算。数据分块过程是按照输入数据原始结构,以先进先出的原则进行逐块输出,以保证数据的完整性以及后续计算的稳定性,进而满足 FPGA 内存要求以及实验结果要求。具体的分块方式如图 6 所示,其中彩色数据块是数据分块示意。

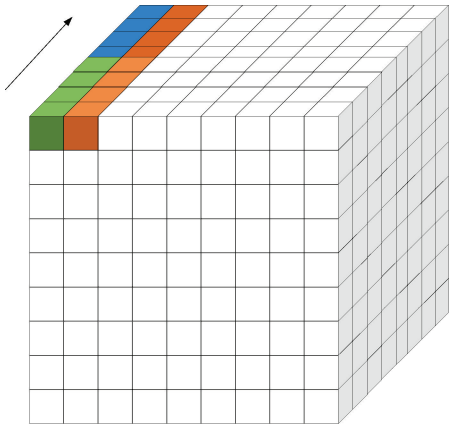

图 6 特征图数据分块

其次,针对卷积过程中的卷积计算,系统采用了卷积窗口复用的机制。在卷积过程中,往往采用卷积核滑动的方式进行运算,将卷积后的结果存储在新的卷积中。每次卷积的卷积都需要重复从权重参数缓存模块中调用,为实现系统的进一步优化,采用了如图 7 所示的卷积核复用机制。如图所示,使用一个卷积核与多个特征图数据块进行卷积运算,减少了大量数据调用的操作,加快进了卷积运算过程,在时序逻辑上也更加稳定。

对于上述步骤要求的权重参数,此次设计也采用数据分块的方法,将卷积过程中所需的权重参数依次加载到 FPGA 的片上缓存内,在进行每次卷积运算前,将同一卷积层中所需的全部卷积核加载到 FPGA 内部,当一层卷积完成后,通过接口控制模块进行下一层卷积权重参数的调用。如图 8 所示的是权重参数调用方式,从 00 数据块开始调用,依次调用至单次特征图数据运算完毕。权重参数的分块调用,不仅加快了数据调用速度、减少了后续运算过程的代码复杂度,同时也使输入数据更加稳定,更具可视性。

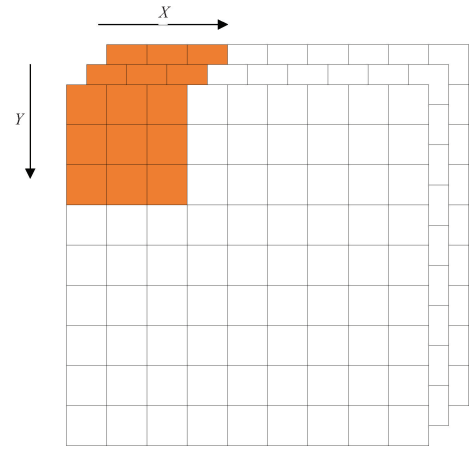


图 7 卷积核窗口复用

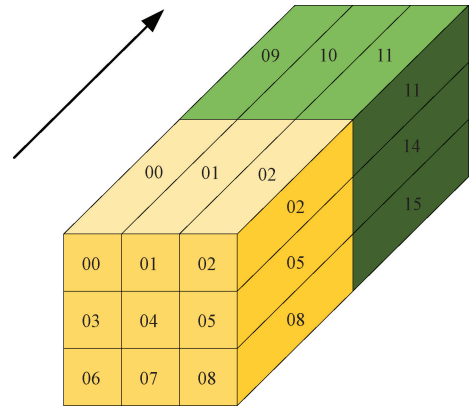


图 8 权重数据分块

## 3 实验结果

### 3.1 实验环境设置

本文使用的实验设备是搭载 Ubuntu18.04 操作系统的计算机,其配备了 12 代英特尔 i5-12400FCPU、16 GB 内存和英伟达 GeForce RTX 2070Ti 显卡。使用 Python 编程语言,在 VS Code 开发环境下使用 PyTorch 深度学习框架进行实验。

本文设计系统在 FPGA 搭建阶段使用 Xilinx 公司 Vivado2021.2 软件进行,硬件编程语言为 Verilog,使用 ModelSim 2019 进行功能仿真,以 Xilinx 公司的一颗型号为 xczu4ev 的 FPGA 作为硬件平台,该芯片详细参数如表 1。

### 3.2 实验结果与分析

实验使用的 YOLOv3-tiny 模型,在自建数据集下进行计算,其主要评价参数是损失函数,是模型在训练期间需要最小化的目标函数,它定义了模型预测值和真实值之间的差异。损失函数通常用于衡量模型对训练数据的拟程度,并根据误差指导模型参数的更新。YOLO 算法的损失函数可以分为 3 个部分:置信度损失、分类损失和目标框定位损失。

**表 1　FPGA 详细参数**

| 参数 | 详情 |
|---|---|
| 型号 | xczu4ev |
| DSP | 728 |
| 查找表(LUT) | 88 K |
| 逻辑单元(Login Cells) | 192 K |
| 外部接口(IO) | 252 |
| APU | 四核 Arm Cortex-A53 MPCore |
| RPU | 双核 Arm Cortex-R5F MPCore |

本文算法模型训练结果如图 9 所示，从整体上来看，置信度损失(obj_loss)、分类损失(cls_loss)和目标框定位损失(box_loss)随着模型的训练，均达到了收敛状态，置信度损失(obj_loss)在前 80 轮训练中迅速下降至 0.04 以下，随着训练轮次的增加，损失函数逐渐在 300 次训练中达到稳定，最后达到至 0.03 左右，模型收敛。

本文算法训练过程的 mAP_0.5 曲线，在第 50 练后，模型的 mAP_0.5 已经超过 85%，随后逐渐趋于稳定，最大的 mAP 为 95.5%。基于上述分析，所构建的模型及数据

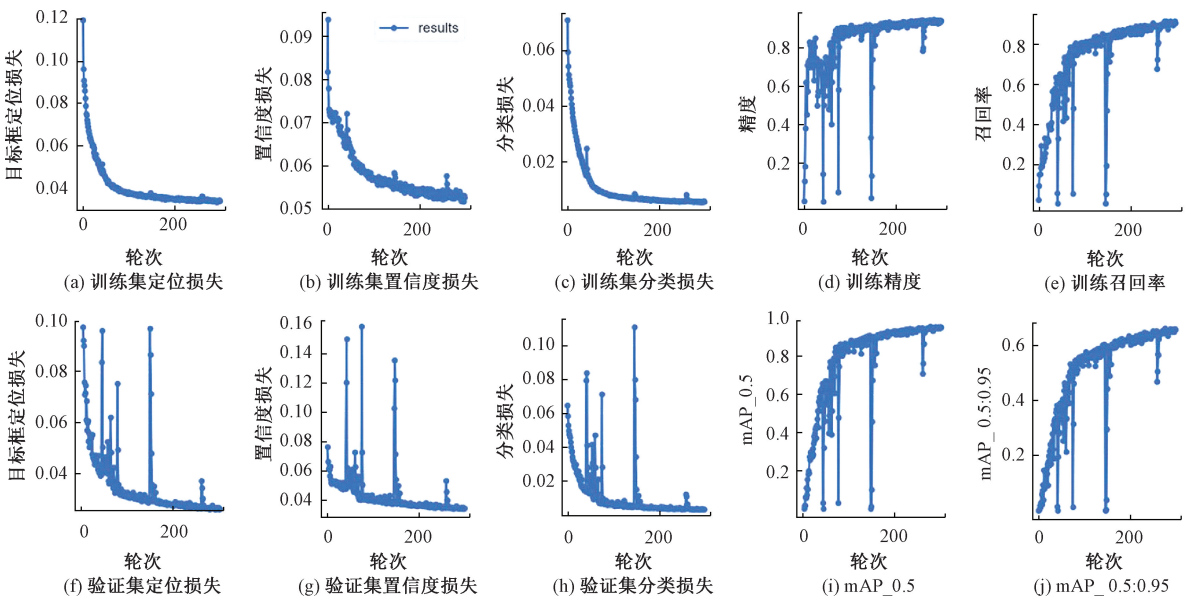


图 9　模型训练结果

集在此次实验上表现出了较好的效果，具体证明了本文数据集的合理性和硬件优化的可行性。

在本实验 FPGA 板载阶段，项目采用的数据精度为 8 bit 定点数，FPGA 运行的时钟频率为 200 MHz，对实验数据集的测试中 FPS 可达到 153，功耗为 4.92 W，峰值 GOP/s 可以达到 115 GOP/s。实际系统检测效果图如图 10 所示。该方案可以满足实时目标检测的要求，并且能够在低功耗的状态下部署在自动驾驶汽车，无人机等设备上。

表 2 中给出的是将本文实验结果与其他文献进行对比的情况，从中可以看出，本文与文献[11]和[12]相比，两者均采用高层次综合(HLS)，前者虽然帧率较高，但其功耗对比于本文及其他文献过高，后者虽功耗较低，但其检测速度过低，在实际应用中适用度不高。相较于文献[13]，虽然其使用更简单的 YOLOv2 网络模型，但本实验在帧率方面对其有碾压性的优势，本文与文献[14]在吞吐量方面基本持平，但文献[14]并没有完成整体实验设计，本文系统的完成了整个实验，相较于文献[15]，本实验虽然功耗有一定增加，但检测速率有较大提升，总的来说较其有一定优势。

图 10　系统检测效果图

整体来说，此次实验选用的 YOLOv3-tiny 可以完成此次实验要求，可以在低功耗状态下完成复杂背景下目标的实时检测。速度上，相比于传统的 CPU 和 GPU 检测在功耗比方面有很大提升，但距离满足自动驾驶状态下的实时检测的实际要求，还有一定的优化空间。

表 2 实验结果横向对比

| 性能 | 文献[11] | 文献[12] | 文献[13] | 文献[14] | 文献[15] | 本文 |
|---|---|---|---|---|---|---|
| 实验平台 | Zynq UltraScale+ | AX7350 | Zynq 7020 | Zynq UltraScale+ | VC707 | Zynq UltraScale+ |
| 加速方法 | HLS | HLS | HLS | RTL | RTL | RTL |
| 网络模型 | YOLOv3-tiny | YOLOv3-tiny | YOLOv2 | YOLOv3-tiny | YOLOv3-tiny | YOLOv3-tiny |
| 平均帧率/FPS | 177 | 2.38 | 0.36 | — | 117 | 153 |
| 功耗/W | 23.02 | 3.304 | 2.494 | — | 2.952 | 4.92 |

## 4 结 论

本文针对自动驾驶解决方案中的一个关键问题进行了探讨,拟解决自动驾驶过程中目标实时检测中的问题,搭建一套高效率,高精度,高稳定性,低功耗,低负载的检测系统。从自动驾驶及目标检测算法的研究现状出发,参考了多种实时目标检测系统方案,选择并实验了基于 FGPA 的目标实时检测系统方案,取得了较好的实验效果。

本文所述系统对实验数据集的测试帧率可达到 153 fps,功耗为 4.92 W,峰值 GOP/s 为 115 GOP/s 比 CPU 提高 315.8%的检测速度,比 GPU 降低 92.4%的功率。该方案可以满足实时目标检测的要求,并且能够在低功耗的状态下部署在自动驾驶汽车,无人机等设备上。在本文的设计中虽然完成了基于 ZYNQ 的目标检测硬件加速工作,并完成了的卷积加速设计,但由于 YOLOv3-tiny 网络模型的特点,已无法满足现阶段自动驾驶的需求,本文虽完成了实验室状态下的 YOLOv3-tiny 网络模型前向推理加速,但距离应用到实际的自动驾驶汽车上仍有很大的改进空间,在安全性、稳定型还需要多做考量。但进一步优化移动端小体积的卷积神经网络加速装置设计,是一个很好的研究方向,对于推动更高级别的自动驾驶有着深远的意义。

## 参考文献


[1] 张艳,彭宏伟,刘然,等. 基于 L3 级自动驾驶的 HMI 设计[J]. 专用汽车,2023(10):15-18.

[2] 李升波,刘畅,殷玉明,等. 汽车端到端自动驾驶系统的关键技术与发展趋势[J]. 人工智能,2023(5):1-16.

[3] 张新钰,高洪波,赵建辉,等. 基于深度学习的自动驾驶技术综述[J]. 清华大学学报(自然科学版),2018,58(4):438-444.

[4] CHEN S Z, HU J L, SHI Y, et al. A vision of C-V2X: Technologies, field testing, and challenges with Chinese development[J]. IEEE Internet of Things Journal, 2020, 7(5): 3872-3881.

[5] 李升波,关阳,侯廉,等. 深度神经网络的关键技术及其在自动驾驶领域的应用[J]. 汽车安全与节能学报,2019,10(2):119-145.

[6] 王军,冯孙铖,程勇. 深度学习的轻量化神经网络结构研究综述[J]. 计算机工程,2021,47(8):1-13.

[7] 李聪,毛剑琳,李大焱,等. 一种面向轻量型卷积神经网络的嵌入式图像识别系统[J]. 自动化与仪器仪表,2021(1):152-155.

[8] 刘军,后士浩,张凯,等. 基于增强 Tiny YOLOV3 算法的车辆实时检测与跟踪[J]. 农业工程学报,2019,35(8):118-125.

[9] 马进,王超. 基于改进 YOLOv4-tiny 的印刷电路板缺陷检测研究[J]. 电子测量技术,2022,45(23):99-106.

[10] HAN S, MAO H, DALLY J W. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding[J]. CoRR, 2015, DOI: 10.48550/arXiv.1510.00149.

[11] 张丽丽,陈真,刘雨轩,等. 基于 ZYNQ 的 Yolo v3-SPP 实时目标检测系统[J]. 光学精密工程,2023,31(4):543-551.

[12] 李伟琦. 基于高层次综合的 YOLOv3-tiny 硬件加速设计与研究[D]. 西安:西安电子科技大学,2022.

[13] 戴伟杰,王衍学,李昕鸣,等. 面向 FPGA 部署的改进 YOLO 铝片表面缺陷检测系统[J]. 电子测量与仪器学报, 2023(9):160-167.

[14] 胡永阳,李淼,孟凡开,等. 基于 Tiny-YOLOv3 的网络结构化压缩与加速[J]. 电子科技,2023,36(8):43-48,55.

[15] 梅志伟,丁兴军,刘金鹏. 基于 FPGA 的 YOLOv3-tiny 卷积神经网络加速设计[J]. 舰船电子对抗,2022,45(2):81-88,108.

[16] ZOU Z X, CHEN K Y, SHI Z W, et al. Object detection in 20 Years: A survey[J]. Proceedings of the IEEE, 2023, 111(3): 257-276.

[17] 陈畅,黄均才,刘鉴栋,等. 卷积神经网络定点化设计及 FPGA 实现[J]. 单片机与嵌入式系统应用,2022,22(2):41-45.

[18] PINGSHU G, LIE G, DANNI H, et al. Light-weighted vehicle detection network based on improved YOLOv3-tiny[J]. International Journal of Distributed Sensor Networks, 2022, 18(3).


## 作者简介


**王新伟**,硕士研究生,主要研究方向为光机电系统集成与智能检测。

E-mail:wangxiinwei@163.com

**丁红昌**,博士,教授,博士生导师,主要研究方向为光学在线检测等技术。

E-mail:dinghc@cust.edu.cn

**曹国华**,博士,教授,博士生导师,主要研究方向为光机电一体化等。