

Hardware Design of General-Purpose Convolutional Neural Network Based on ZYNQ^{*}

LIU Xian¹, WU Ruiqi¹, GAO Shangshang¹, LIU Zehao¹, LIU Haibo¹,
KONG Xiangye¹, WANG Qing¹, GUO Naihong², ZHOU Feng¹, WANG Rugang^{1*}

(1. School of Information Technology, Yancheng Institute of Technology, Yancheng Jiangsu 224051, China;
2. Yancheng XiongYing Precision Machinery Company Limited, Yancheng Jiangsu 224006, China)

Abstract: A general-purpose convolutional neural network model based on ZYNQ is designed to address the problems of long development cycles and troublesome adjustment of network models based on FPGA platform. Firstly, the neural network is built and trained on Tensorflow platform to get the weight of each layer of the model. Secondly, the IP cores of the convolutional and pooling layers are designed by using the high-level synthesis tool. Then the model is deployed on ZYNQ platform. Finally, the LeNet-5 and AlexNet-8 neural networks are run respectively for verification. The experimental results show that the convolutional and pooling speed are 3.65 and 2.31 times faster respectively than the ARM platform with minimal loss of accuracy, and that the model is generalizable.

Key words: convolutional neural network; ZYNQ; high-level synthesis tool

EEACC: 1265A doi: 10.3969/j.issn.1005-9490.2023.01.021

基于 ZYNQ 的通用型卷积神经网络设计与实现^{*}

刘 颀¹, 吴瑞琦¹, 高尚尚¹, 刘泽浩¹, 刘海波¹, 孔祥晔¹, 王 庆¹, 郭乃宏², 周 锋¹, 王如刚^{1*}

(1. 盐城工学院信息工程学院, 江苏 盐城 224051; 2. 盐城雄鹰精密机械有限公司, 江苏 盐城 224006)

摘 要: 针对基于 FPGA 平台的神经网络开发周期过长、调节网络模型麻烦等问题, 设计了一种基于 ZYNQ 的通用型卷积神经网络模型。首先通过 Tensorflow 平台搭建神经网络并训练得到模型各层权重; 其次利用高层次综合工具进行卷积层和池化层的 IP 核设计; 然后在 ZYNQ 平台上部署模型; 最后分别运行了 LeNet-5 和 AlexNet-8 神经网络进行验证。实验结果表明模型在只损失极少准确度的情况下, 卷积与池化运算速度相比于 ARM 平台分别提高了 3.65 倍和 2.31 倍, 并具备通用性。

关键词: 卷积神经网络; ZYNQ; 高层次综合工具

中图分类号: TP183

文献标识码: A

文章编号: 1005-9490(2023)01-0121-05

卷积神经网络(Convolutional Neural Network, CNN)是由人工神经网络发展起来的一种深度学习网络, 由于能够实现待测目标的精确检测和识别等任务, 已经被广泛应用于计算机视觉、图像识别与处理等多个领域中^[1-2]。虽然 CNN 具备大规模图像处理的高精度优势, 但这种优势是以高计算量为代价来实现的, 为了解决高计算量的问题, 研究人员在计算加速器方面进行了大量的研究, 从研究成果来看, 目前的主要集中在利用特殊应用集成电路(Application Specific Integrated Circuit, ASIC)、图形处理器(Graphics Processing Unit, GPU)和现场可编程门阵列(Field Programmable Gate Array, FPGA)等

加速器来实现^[3-5]。其中 ASIC 开发成本比较高, 且灵活性较差, 不适用于复杂多变的卷积神经网络^[6]; 基于 GPU 的加速系统具有较大的体积和较大的功耗^[7]。而 FPGA 具有体积小、功耗低、能源效率高、并行运算等特点更适用于卷积神经网络的硬件加速器开发^[8-13]。不同于传统的 FPGA 产品, ZYNQ 具备 ARM + FPGA 的体系结构, 在总线协议(Advanced Extensible Interface, AXI)框架下设计了自定义数据存储传输的 IP 核, 实现了处理速度与带宽最大化^[14], 既可以在 ARM 上部署操作系统及运行软件程序, 也能够利用 FPGA 的并行性特点对软件中并行加速的部分进行加速, 提高了整个平台的

项目来源: 江苏省研究生实践创新计划项目(SJCX21_1517); 江苏省高等学校自然科学研究重大项目(19KJA110002); 国家自然科学基金项目(61673108); 江苏省产学研合作项目(BY2020335, BY2020358); 江苏省高校自然科学研究面上项目(20KJB140025); 江苏省自然科学基金项目(BK20181050)

收稿日期: 2021-10-06 修改日期: 2021-12-09

运行效率,因此,嵌入式神经网络单元非常适合搭建在 ZYNQ 平台上。2020 年,张新伟等^[15] 研究人员进行了基于 ZYNQ 集群的神经形态计算加速研究与实现,实验结果表明通过软硬件协同的分布式方式,可以更方便地探索体系结构。2021 年,尹震宇等^[16] 研究人员设计并实现了一种基于 ZYNQ 平台的卷积神经网络单元,通过充分利用 ZYNQ 平台上 FPGA 端并行计算的特点,对卷积神经网络中卷积层进行加速,实现在嵌入式平台上对手写数字的高效识别。2021 年,刘杰等^[17] 研究人员设计了一种基于 ZYNQ 的具有高吞吐率和低功耗的可重构神经网络加速系统,以 VGG16 网络为例,利用 ZYNQ 对系统进行加速,在计算性能上达到 62.00 GPOS 的有效算力。通过这些研究,基于 ZYNQ 的卷积神经网络获得较大的进步,获得了较好的研究成果。

现有研究成果虽然实现了对卷积神经网络的加速但是开发周期长、难度高、模型更换十分麻烦,针对这一问题,设计了一种基于 ZYNQ 的可配置的通用卷积神经网络模型,实验利用 ZYNQ 平台拥有处理器系统 (Processor System, PS) 端与可编程逻辑 (Programmable Logic, PL) 端的特性以及高层次综合工具 HLS 实现了可配置的卷积模块与池化模块的设计。

1 卷积神经网络基本知识

传统的神经网络随着识别对象复杂程度的增加计算量也随之增加,卷积神经网络相比于传统的神经网络加入了卷积操作和池化操作,以便更好地对识别对象的特征进行有效提取,且减少了计算量,由于其中权重的共享性使网络具有更大的适应性,不会使训练过拟合从而导致在识别训练集以外的对象时无法保持良好的识别效果。卷积神经网络一般是由卷积层、池化层、全连接层组成,其中卷积层和池化层是对目标的特征提取,后者还能降低计算量和降低优化难度,全连接层相当于再次转化为传统的神经网络,一般在最后一层全连接层中权重为该网络所需识别的标签数,采用 Softmax 函数将其中最大的一个值筛选出来即为卷积神经网络的识别结果。

LeNet-5 是一种经典的卷积神经网络结构,由两个卷积层、两个池化层和三个全连接层构成。LeNet-5 模型的第 1 层为 6 个 28×28 大小的卷积核,第 2 层为 14×14 的池化,第 3 层包含 16 个大小为 10×10 的卷积核,第 4 层为 5×5 的池化。其中两层卷积层和两层池化层交替出现,再经由三层全连接层后输出^[18]。网络结构图如图 1。

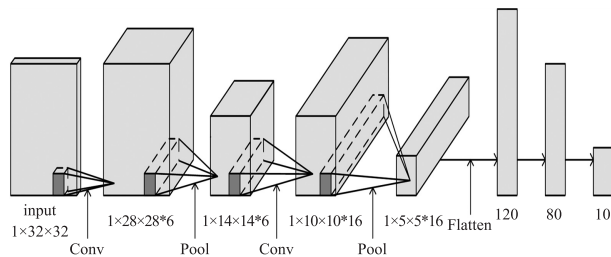


图 1 LeNet-5 网络结构图

而 AlexNet-8 网络是一种深度卷积神经网络,由五个卷积层、三个池化层和三个全连接层构成^[19]。AlexNet-8 模型的网络结构如图 2 所示。AlexNet-8 模型的第 1 个卷积层包含 96 个 11×11 大小的卷积核,第 2 个卷积层包含 256 个大小为 5×5 的卷积核。第 3、4 个卷积层分别包括 384 个 3×3 大小的卷积核,第 5 个卷积层具有 256 个 3×3 大小的卷积核,网络的识别结果通过最后一个全连接层输出。后三层卷积层通过在外层加入填充来保持卷积后输出的尺寸不变。AlexNet-8 相比于 LeNet-5,在增加了三层卷积层后,引入线性整流函数 (Rectified Linear Unit, ReLU) 激活函数来加快训练的速度,缓解了梯度消失的问题,并造成了网络的稀疏性,一定程度上缓解了过拟合,使得网络能更精准。

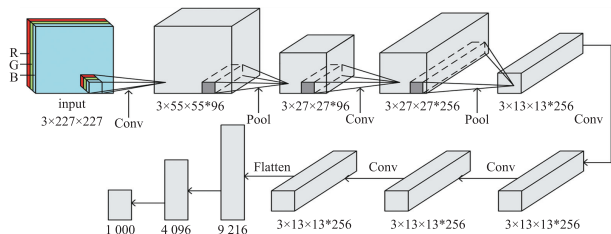


图 2 AlexNet-8 网络结构图

2 系统设计

2.1 系统的总体设计

系统采用 Xilinx 公司推出的 Zybo-Z7 开发板作为实验平台,Zybo-Z7 是基于 Xilinx ZYNQ-7000 全可编程片上系统架构的实验平台,该平台还搭载了 XC7Z010-1CLG400C FPGA 芯片,其可编程逻辑等效于 Artix-7 FPGA。另外平台还配套了 SD 卡口,来进行大量数据的存储。相比于传统的 FPGA 平台,ZYNQ 搭载了一颗双核 ARM Cortex-A9 处理器,整体分为 PS 模块与 PL 模块,其中 PS 模块为处理系统也就是与 FPGA 无关的 ARM 片上系统 (System on Chip, SoC) 部分,PL 模块则是 FPGA 部分。通过双片 BRAM 与 AXI 总线控制能实现 PS 与 PL 之间的通信,从而高效地实现嵌入式开发。编程方面,对 ARM 指令的支持是由 Xilinx 软件开发包 (Software Development Kit, SDK) 来实现的。这使得 ZYNQ 平台既能

方便地在 ARM 上部署系统来对 FPGA 的参数进行调整或者是发送相应的参数选择性地改变 FPGA 的逻辑工作模式,也可利用 FPGA 高效的运算能力来对运算进行加速,提高了整个系统运行的速度,同时保留了嵌入式平台体积小、功耗小、便于部署的优点。

本设计分为软件部分和硬件部分,其中软件部分由 ZYNQ 的 PS 部分完成,其主要的功能有读取 SD 卡中的网络参数、对卷积与池化 IP 核的配置以及串口的通信,该部分由 Xilinx 软件开发包 SDK 用 C 语言进行系统设计。硬件部分由 ZYNQ 的 PL 部分完成,其主要功能是利用 FPGA 的逻辑资源对卷积运算与池化运算进行硬件加速。其中卷积模块与池化模块由 HLS 高级综合工具进行 C 语言设计后进行 IP 核的封装,并对其进行可配置的设计,使其

不仅被电路重复利用并可以根据所选的模型进行模式的调整来满足当前的网络,然后在 Vivado2016 进行电路的整合。硬件整体电路图如图 3 所示,首先加入系统处理器 ARM processing_system7_0 以及模块 Conv_0 与 Pool_0 及前面设计的卷积模块与池化模块,然后通过 Vivado 的 Connection Automation 功能来对模块进行连接,由于在设计卷积模块与池化模块时设计的是 AXI 通讯方式,所以会生成 3 个 AXI Interconnect 的 IP 核,它们起到的作用是完成 IP 核的 AXI 总线通讯。processing_system7_0 可以通过 ps7_0_axi_periph 来同时访问 Conv_0 与 Pool_0,同样,Conv_0 与 Pool_0 分别能通过 axi_intercom 以及 axi_intercon_1 来访问 processing_system7_0,而 rst_ps7_0_100M 为时钟复位模块。

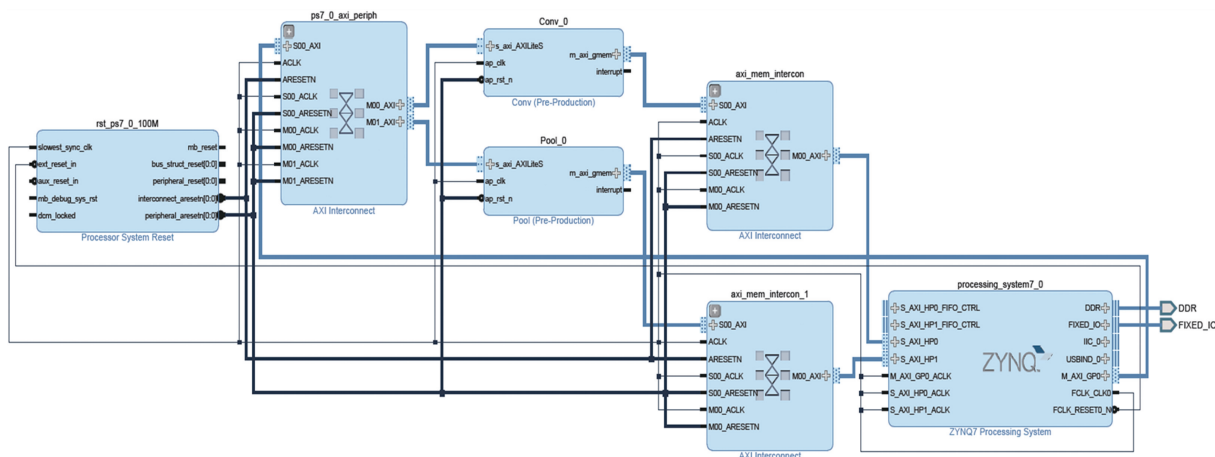


图3 实验系统硬件电路

系统由 PS 端对 PL 端进行调用来完成整个数据的传输与卷积神经网络的运算,由 PS 端完成权重文件读取、接收输入、调用 PL 端的卷积与池化 IP 核进行计算以及与上位机进行通讯完成实验测试,其流程如图 4 所示。

2.2 卷积模块和池化模块 IP 核设计

Xilinx 推出了 HLS 高级综合工具结合 FPGA 的设计与开发,用户可以选择多种不同的高级语言(如 C、C++、system C)来进行 FPGA 的设计,根据代码生成相应的硬件电路,并快速优化 FPGA 硬件结构,提高执行效率,降低开发的难度。

为了实现两个加速电路,本文利用 HLS 生成相应的 IP 核。其中卷积操作是多次的乘加运算,利用 AXI 总线控制的方式来实现模块与 ARM 之间的通讯,通过参数的传递来给模块提供输入并配置卷积模块中卷积的模式与大小,同样计算的结果也通过 AXI 总线由模块传递给 PS 端,其参数有 CHin 输入通道数、Hin 输入通道的高度、Win 输入通道的宽度、CHout 输出通道数、kx 卷积核宽度、feature_in[] 输入、W[] 权重、feature_out 输出、relu_en 是否经过 relu 运算、mode 模式调整以及 i 与 j 为循环变量。其中 mode 模式是用于填充调节,当模式为 0 采用 VAILD 模式,当模式为 1 采用 SAME 模式。卷积计

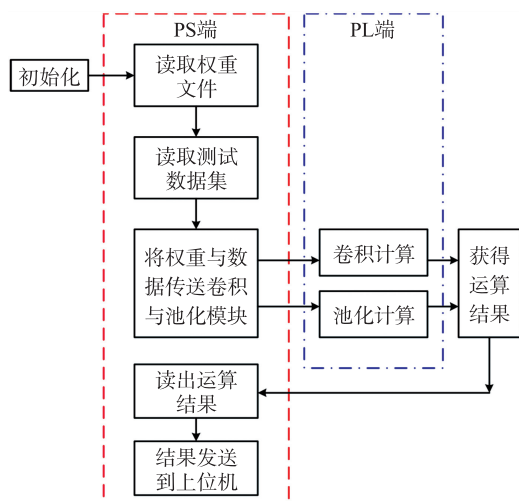


图 4 系统流程图

算部分本设计采用了 6 层 for 循环来完成乘加操作,每一次的卷积操作可以表示为:

$$\text{feature_in}[h \times \text{CHin} \times \text{Win} + w \times \text{CHin} + \text{cin}] \times W[ii \times Kx \times \text{CHin} \times \text{CHout} + jj \times \text{CHin} \times \text{CHout} + \text{cin} \times \text{CHout} + \text{cout}]$$

池化模块同样也具备以上大部分的参数,取消了 relu_en 变量,mode 模式分为 3 种模式,分别为平均池化、最大池化与最小池化,可以分别表示为:

平均: $\text{sum} += \text{feature_in}[h \times \text{CHin} \times \text{Win} + w \times \text{CHin} + c]$

最小: $\text{sum} = \min(\text{sum}, \text{feature_in}[h \times \text{CHin} \times \text{Win} + w \times \text{CHin} + c])$

最大: $\text{sum} = \max(\text{sum}, \text{feature_in}[h \times \text{CHin} \times \text{Win} + w \times \text{CHin} + c])$

这样的设计使得该模块能对大多数的卷积神经网络模型通过 IP 核的调用来实现前向传播,从而对结果进行识别预测。

2.3 选取网络模型训练以及权重的提取与二值化

为了验证该设计的可配置性以及加速能力,选取了 LeNet-5 和简化的 AlexNet-8 等两个模型来进行验证,先通过 TensorFlow 进行网络搭建,喂入数据集进行迭代训练,为了更加还原不同场景的应用,其中 LeNet-5 选用的是手写字数据集, AlexNet-8 选用的是 fashion 数据集。经过 10 次训练集的喂入, LeNet-5 对手写字的识别度达到了 99.31%, AlexNet-8 对 fashion 测试集的识别度达到了 91.49%。由于所选的数据集图像为 28×28,为了更适应小尺寸的识别,对 AlexNet-8 进行了简化,对卷积核的数量与层数进行了一定量的减少,训练结束后对各个网络的各层权重进行保存,此时权重为 10 进制的浮点数,对于 FPGA 来说是无法进行识别的,本设计编写了一段 C 脚本将权重转化为二进制数据存于 bin 类型文件中,在实验阶段,这些文件将存入 Zybo-Z7 开发板的 SD 卡中,通过 PS 端来对 SD 卡中的权重进行读取。

3 实验与结果

本实验采用 Zybo-Z7 开发板,其主芯片是 XC7Z010,主要由 PS 和 PL 两部份组成,PS 端是 650 MHz 双核 Cortex-A9 处理器,PL 端的时钟频率为 100 MHz。开发环境为 Vivado2016,首先在 HLS 工具中对通用可配置卷积模块与池化模块进行 IP 核设计,然后在 Vivado2016 中用对总体硬件电路进行设计,卷积模块及池化模块与 ARM 的通信采用 AXI 总线的方式进行,生成比特流后导出设计运行 SDK,先通过软件计算与调用 IP 运算的结果进行对

比,对卷积模块与池化模块的功能进行验证,然后写入 SD 卡通信程序,最后对整个系统进行整合,对于不同的神经网络只需要根据网络层数来多次调用卷积与池化 IP 核,并对参数进行相应的修改即可。将测试集与权重文件放入 SD 卡中,然后在 SDK 中通过调用卷积模块与池化模块来对 LeNet-5 与 AlexNet-8 进行硬件模型搭建,再通过 SD 卡读取出权重与相应的测试数据集来计算,最后通过 ZYNQ 与上位机得到最后的验证结果,其中数据集与 TensorFlow 平台所用测试集相同。网络加速后的结果如表 1 所示。从表 1 可以看出,在相同测试数据集的测试下,两个网络经过加速后识别精度只有轻微的下降,这主要是由于权重在转化为二进制以及数据的传输中出现了些许的丢失,但是依旧保留了十分良好的识别度。

表 1 网络加速后精度对比 单位: %

网络类型	原本准确度	加速后准确度
Lenet5	99.31	99.18
AlexNet8	91.49	91.26

通过 SDK 编写系统对卷积模块与池化模块进行调用,并在 ARM 平台运行相同的计算进行对比,得到的结果如表 2 所示,从表 2 可以看出,在同样的运算参数下 FPGA 在卷积计算速度上相比于 ARM 平台速度提高了将近 3.65 倍,在池化计算上也有着 2.31 倍的速度提升。

表 2 单次运算不同平台运算速度对比 单位: μs

平台	卷积运算时间	池化运算时间
ARM	127 031	886
FPGA	34 846	382

4 结语

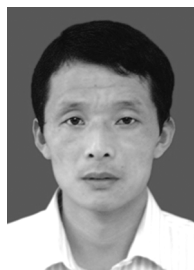
基于 FPGA 平台的神经网络具备体积小、功耗低、算力高的优点,但是其开发周期很长,调节网络模型需要大量的时间。针对这些问题,设计了一种基于 ZYNQ 的通用型卷积神经网络模型,实现了对卷积运算与池化运算的硬件加速,利用高层次综合工具进行卷积层和池化层的 IP 核设计,实现了模块的通用化。在设计平台上,分别运行了 LeNet-5 和 AlexNet-8 神经网络。从实验结果可以看出,该平台可以实现便捷的网络模型调节,且卷积运算速度与池化运算速度相比于 ARM 平台分别提高了 3.65 倍和 2.31 倍。与其他神经网络实现方式相比,该模型实现在 ZYNQ 平台上,功耗低、轻巧、方便且具备通用性,易在各种环境下进行部署,有着极强的实际应用价值。

参考文献:

- [1] Simayi W, Ibrayim M, Hamdulla A. Study the Preprocessing Effect on RNN Based Online Uyghur Handwritten Word Recognition[J]. Wireless Networks, 2021, <https://doi.org/10.1007/S11276-021-02621-w>.
- [2] Musikhin A G, Burenin S Y. Face Recognition Using Multitasking Cascading Convolutional Networks[J]. IOP Conference Series: Materials Science and Engineering, 2021, 1155(1): 012057.
- [3] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211–252.
- [4] Ammendola R, Biagioni A, Frezza O, et al. APENet+: A 3D Torus Network Optimized for GPU-Based HPC Systems[J]. Journal of Physics: Conference Series, 2012, 396: 042059.
- [5] Mittal S. A Survey of FPGA-Based Accelerators for Convolutional Neural Networks[J]. Neural Computing and Applications, 2020, 32(4): 1109–1139.
- [6] 李永博, 王琴, 蒋剑飞. 稀疏卷积神经网络加速器设计[J]. 微电子学与计算机, 2020, 37(6): 30–34, 39.
- [7] 仇越, 马文涛, 柴志雷. 一种基于 FPGA 的卷积神经网络加速器设计与实现[J]. 微电子学与计算机, 2018, 35(8): 68–72, 77.
- [8] 魏浚峰, 王东, 山丹. 基于 FPGA 的卷积神经网络加速器设计与实现[J]. 中国集成电路, 2019, 28(7): 18–22, 67.
- [9] 王红亮, 程佳风. 基于嵌入式设备应用的 CNN 加速器的设计研究[J]. 电子器件, 2021, 44(4): 797–801.
- [10] 秦华标, 曹钦平. 基于 FPGA 的卷积神经网络硬件加速器设计[J]. 电子与信息学报, 2019, 41(11): 2599–2605.
- [11] 郭谦, 贺光辉. 基于 FPGA 的卷积神经网络硬件加速器设计空间探索研究[J]. 微电子学与计算机, 2020, 37(8): 66–71.
- [12] 王晓峰, 蒋彭龙, 周辉, 等. 面向卷积神经网络的高并行度 FPGA 加速器设计[J]. 计算机应用, 2021, 41(3): 812–819.
- [13] 王婷, 陈斌岳, 张福海. 基于 FPGA 的卷积神经网络并行加速器设计[J]. 电子技术应用, 2021, 47(2): 81–84.
- [14] 熊国强, 徐渊, 朱明程, 等. 一种基于 ZYNQ 的嵌入式高尔夫球体花纹识别系统设计与实现[J]. 电子器件, 2019, 42(2): 484–490.
- [15] 张新伟, 李康, 郁龚健, 等. 基于 ZYNQ 集群的神经形态计算加速研究与实现[J]. 计算机工程与应用, 2020, 56(21): 65–71.
- [16] 尹震宇, 徐光远, 张飞青, 等. 面向 ZYNQ 平台的卷积神经网络单元设计与实现[J]. 小型微型计算机系统, 2022, 43(3): 231–235.
- [17] 刘杰, 葛一凡, 田明, 等. 基于 ZYNQ 的可重构卷积神经网络加速器[J]. 电子学报, 2021, 49(4): 729–735.
- [18] 姜凡旭, 傅洪亮, 陶华伟, 等. 一种基于卷积神经网络特征表征的语音情感识别方法[J]. 电子器件, 2019, 42(4): 998–1001.
- [19] 刘世泽, 秦艳君, 王晨星, 等. 基于多尺度特征提取的交通模式识别算法[J]. 计算机应用, 2021, 41(6): 1573–1580.



刘 晔(1998—),男,湖南益阳人,硕士生,主要从事智能控制系统及其信号检测方面的研究;



王如刚(1975—),男,江苏睢宁人,博士,教授,硕士生导师,主要研究方向为通信技术、计算机视觉及图像处理等, wrg3506@ycit.edu.cn。