硕士学位论文

（学术学位论文）

基于 FPGA 和 Lenet-5 的字符识别系统设计与实现

DESIGN AND IMPLEMENTATION OF CHARACTER RECOGNITION SYSTEM BASED ON FPGA AND LENET-5

程永龙

哈尔滨工业大学

2022 年 6 月

国内图书分类号：TP391.4　　学校代码：10213
国际图书分类号：621.3　　密级：公开

硕士学位论文

基于 FPGA 和 Lenet-5 的字符识别系统设计与实现

硕士研究生：程永龙
导　　师：林玉荣 副教授
申请学位：专业学位
学　　科：电子信息
所在单位：航天学院
答辩日期：2022 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index: TP391.4
U.D.C: 621.3

Dissertation for the Master's Degree

DESIGN AND IMPLEMENTATION OF CHARACTER RECOGNITION SYSTEM BASED ON FPGA AND LENET-5

Candidate: Cheng YongLong
Supervisor: Assoc. Prof.Lin YuRong
Professional Degree Applied for: Master of Engineering
Speciality: Electronic Information
Affiliation: School of Astronautics
Date of Defence: June, 2022
Degree-Conferring-Institution: Harbin Institute of Technology

摘 要

FPGA（Field Programmable Gate Array）由于具有多通道同时运算、可单独运作以及可反复修改功能等特点，在航空航天领域、战略军事武器、芯片设计及验证、5G 通信以及医疗领域有着广泛的应用。本文的主要研究内容如下：

考虑到不同光照条件对检测精度的影响，本文分析了不同光照条件下不同的图像预处理算法对字符识别系统检测精度的影响，对于后续目标识别的项目提供了一定的实验基础，对于以深度学习网络为手段实现的目标识别算法在 FPGA 平台的部署提供了很好的实验资料。

分析了国内外 FPGA 平台实现卷积神经网络的难点之后，以提高字符识别系统的实时性和卷积神经网络基本计算模块的复用性为目的，对 Lenet-5 卷积神经网络在 FPGA 平台的网络推演过程进行了深入的理论研究并提出了卷积神经网络在 FPGA 平台的实现方案。针对卷积计算数据读取问题和 XC7k325tffg-676-2 型号的 FPGA 平台的特性提出了卷积计算地址读取方案——首地址锚定法；针对特征图中间数据存储问题提出了适用于提出了卷积神经网络在 FPGA 平台上层间通信和流水线设计的方案；针对 FPGA 平台无法进行小数计算的问题提出了 Lenet-5 计算权值在 FPGA 平台的量化方案并保证了可允许范围内的精度损失。

对卷积神经网络在 FPGA 平台的实现方案进行理论研究之后，在 FPGA 平台搭建了 Lenet-5 卷积神经网络的硬件电路，并对其进行仿真调试，实现了字符识别功能。通过在不同计算层间加入使能信号，实现了 Lenet-5 卷积神经网络不同层之间的层间通信和层级递进的计算协同；通过研究补码计算规律，提出了数值计算的防溢出方案；通过在相邻计算层间加入 Block RAM 实现了网络中间数据的存储和网络流水线设计，提高了整体运行速度；针对后续模块复用问题规范化了卷积神经网络基本计算模块并完成 IP 测试与封装，如 Convolution IP 核、Pooling IP 核以及 Full Connection IP 核等。

经过实验验证了在保证字符识别精度的情况下 Lenet-5 在 FPGA 平台上的运行速度的提高。为之后实现复杂的深度学习网络在 FPGA 平台的推演工程项目设计打下了坚实的基础。

关键词：FPGA；Lenet-5；字符识别；流水线设计；首地址锚定法

Abstract

Field Programmable Gate Array (FPGA) has a wide range of applications in aerospace, strategic military weapons, chip design and verification, 5G communications and medical fields due to its characteristics of multi-channel simultaneous computing, independent operation and repeated modification. The main research contents of this paper are as follows:

Considering the influence of different illumination conditions on the detection accuracy, this paper analyzes the different lighting condition of image preprocessing algorithm for character recognition system, the influence of accuracy for later target recognition project some experimental basis will be provided for by means of deep learning network to achieve the target recognition algorithm in FPGA platform deployment provides a good experimental data.

After analyzing the difficulties of realizing convolutional neural network on FPGA platforms at home and abroad, in order to improve the real-time performance of character recognition system and the reusability of basic computing modules of convolutional neural network, The network deducting process of Lenet-5 convolutional neural network on FPGA platform is studied theoretically and the implementation scheme of convolutional neural network on FPGA platform is proposed. Aiming at the data reading problem of convolution computation and the characteristics of XC7K325TFFG-676-2 FPGA platform, the first address anchoring method of convolution computation is proposed. Aiming at the problem of feature graph data storage in the middle, a scheme of intercommunication and pipeline design of convolutional neural network on FPGA platform is proposed. Aiming at the problem that THE FPGA platform can not calculate decimal, the quantization scheme of lenet-5 calculation weight on the FPGA platform is proposed and the accuracy loss within the allowable range is guaranteed.

After the theoretical research on the implementation scheme of convolutional neural network in FPGA platform, the hardware circuit of Lenet-5 convolutional neural network is built on FPGA platform, and the simulation debugging is carried out to realize the character recognition function. By adding enabling signals between different computing layers, the inter-layer communication and

hierarchical computing collaboration between different layers of LENet-5 convolutional neural network are realized. By studying the rule of complement code calculation, a numerical overflow prevention scheme is proposed. By adding Block RAM between adjacent computing layers, the intermediate data storage and network pipeline design are realized, and the overall running speed is improved. For subsequent module reuse problems, the basic computing module of convolutional neural network is standardized and IP testing and encapsulation is completed, such as Convolution IP core, Pooling IP core and Full Connection IP core. On this basis, by adding image preprocessing module, the working effect of character recognition system in bad environment is improved.

Experiments verify that lenet-5 can improve the running speed on FPGA platform under the condition of ensuring the accuracy of character recognition. It lays a solid foundation for the design of complex deep learning network in FPGA platform.

Keywords：FPGA, Lenet-5, Character recognition, Pipeline design, Head address anchoring method

目 录

第1章 绪 论

1.1 课题研究背景及现实意义

近年来，深度学习（Deep Learning, DL）成为了各大研究机构研究的热点。很多企业迎风而起，如人工智能（Artificial Intelligence, AI）四小龙；很多科研院所，如中科院自动化所，都在做深度学习芯片相关的研究；还有一些院所控股的企业也在做深度学习算法到 FPGA 的移植以构建综合系统。深度学习算法被应用到很多地方，大到国防侦查，小到微信的语音翻译，深度学习在我们生活中无处不在。

深度学习算法自出现之后就表现出了其高度实用性。与之前的目标检测算法不同的是，它凭借多层的特征提取网络获取图像特征以及监督机制大大提高了目标检测的准确性。在此基础之上，深度学习算法得到了大众的青睐，走入寻常百姓家，例如，门禁上的人脸识别系统[1]、微信等语音输入功能、钞票数额检测[2]、交通标志识别系统[3-6,29-30]、智慧交通数据中心用于抓捕套牌车的车牌文字识别系统以及医疗领域的影像品质提升和基于视觉的医学器械应用等。

本文所研究的基于 FPGA 和 Lenet-5 的字符识别系统可用于智能交通中交通标志的识别以及套牌车车牌识别。交通标志的识别可用于自动驾驶中的驾驶辅助。套牌车车牌识别主要用于智慧交通数据中心的套牌车检测。针对智慧交通中目标检测算法的实时性问题和算法调试问题，选择了具备并行计算优势和可以在工程地编程特点的 FPGA 平台，并对 Lenet-5 卷积神经网络在 FPGA 平台的网络推演加速设计进行了深入的研究。

为了获得更好的目标检测效果，深度学习算法的网络越来越庞大，精度得到了大幅度的提升，检测目标类型越来越多，这也造成了深度学习算法训练和检测时运行速度慢、实时性差的问题。深度学习算法亟需并行处理来提高运行速度。

目前，深度学习算法针对 FPGA 平台的硬件特性进行网络推演的应用主要有以下三种方法。

第一种是利用 HLS（High Level Synthesis）一类的高层次综合软件[1,7-8]，通过 C#语言编写和库文件调用实现其功能。然后由 HLS 封装成 IP，由

Vivado 调用 IP 实现其功能。简而言之，就是把 C#语言翻译成 HDL。这种方法的优点是可以大大缩短开发周期，开发人员不用非要会用 HDL，只需熟练 C#语言即可。但其缺点也很明显，它会浪费很多开发板上的资源，降低资源利用率，而且不好维护。

第二种是使用 PS（处理器端, 如 DSP、ARM）+PL（逻辑门端， FPGA）的架构实现深度学习算法[9-20]。PL 作为相机的驱动，对工业相机采集到的图像数据进行数据高速传输，负责数据采集和加速，PS 作为数据处理端，负责数据处理和结果显示。这种方法的优点是可以利用 FPGA 并行性高的特点实现数据的高速采集，同时 DSP（Digital Signal Processing）多核心同时运作，提高数据处理的速度。DSP 和 FPGA 放在同一块开发板上，两芯片之间通过 Rapid IO 通信。它的缺点是当运行复杂的深度学习算法时，无法取得很好的实时性，而且此架构价格昂贵，成本高。

第三种是利用 HDL 编写深度学习算法。它基于 FPGA 平台实现，按照时钟信号、层间通信信号以及其他使能信号实现层级递进的网络推演过程，而且每一层所有通道的数据都会同时出现，这就实现了数据的高速处理，大大提高了算法的实时性。但由于深度学习算法需要存储大量的中间数据，且 FPGA 片上 Block RAM 存储资源有限，故而仅可用 Block RAM 实现小的神经网络。当中间数据过多时，就要动用板上的 DDR 存储资源，那样算法实时性将会大打折扣。

1.2 深度学习及图像预处理相关理论的发展概况

Lenet-5 于 1998 年提出，在当时很多机器学习领域使用人工设计的网络进行目标识别的背景下，作者提出使用样本训练卷积神经网络的方法获得网络权重。Lenet-5 卷积神经网络成功用于实际应用中，负责检测钞票字符，这也是卷积神经网络的开端。

1.2.1 深度学习的发展概况

1998 年，Lenet-5 卷积神经网络[21-37]问世之后，实现了成功的应用。提出了一种实际应用效果优良的网络搭建方法，首先确定一个网络架构，这个很大部分来自于实践分析，然后通过反向推导的方法确定每一层的权重信息，最后利用这些权重信息推演得到一些高维度特征图，最后实现目标分类。同时他提出的利用卷积计算提取图片二维特征的方法奠定了卷积神经网络的基础。

1.2.2 图像预处理算法发展概况

一般执行字符识别任务，基本遵循图像滤波、图像增强以及字符分割这几个步骤，它决定了字符识别的效果。下面分别介绍各个步骤。

（1）**图像滤波**　图像滤波算法[38-42]主要是去除图片中不相关的细节。图像滤波算法的选用要根据具体场景中可能出现的噪声类型选择，如适合过滤椒盐噪声的中值滤波、适合过滤通过图像扫描产生的颗粒噪声的均值滤波以及适合过滤高斯噪声的高斯滤波。也有把图像滤波和图像增强结合起来实现高清晰度边缘检测的算法。文献在本文第 2 章中，将简单介绍各种图像滤波算法的基本原理和应用场景。根据图像滤波算法的特点以及实际应用场景，选择合适的图像滤波算法。

（2）**图像增强**　图像增强算法[43-48]就是通过一些数学分析工具，使得图像中我们所要识别的目标变得更加明显，使得图像中无用的信息或者噪声信息变得可以忽略。图像增强的目的是使图像变得更易于人眼识别或者更易于机器学习。图像增强算法广泛应用于医疗影像领域和机器学习领域。在医疗领域，图像增强算法可以使 X 光等影像片变得更易于人眼识别。而在机器学习领域，图像增强算法可以使图像更易于机器学习，作为目标识别、目标跟踪的图像预处理操作。

（3）**字符分割**　字符分割算法[2-6,29-30]主要有间隙法和投影法等两种方法。在进行字符识别任务之前，由于本课题训练的 Lenet-5 卷积神经网络是针对单字符识别的，所以首先要进行字符分割。拿车牌识别来说，车牌号由一串由汉字、数字和字母组成，相机传感器拍摄到的图像包含一整张车牌，要想获得一个车牌的车牌号，先要确定每个字符的位置，然后挨个检测字符并输出检测结果。如果这一步出现一些错误，就会影响后面的结果，比如，如果分割任务处理数字“8”得到的是一个“3”，那么该结果进入 Lenet-5 卷积神经网络之后识别的结果是“3”而不是“8”。所以在 Lenet-5 卷积神经网络前，需要加上一个字符分割模块，否则就会出现错误识别等情况。这就是没有字符分割的系统会产生的问题，而且字符分割之前还需要校正原始图片，从而使得字符识别更加地精确无误。

1.3 FPGA 应用概况

FPGA（Field programmable gate array）是一种可重复使用的用于代替专门芯片进行性能调试的芯片，这种芯片可以通过 Vivado 软件进行硬件设计，

从而改变芯片功能。由于它的这种特性，FPGA 芯片广泛应用于芯片未确定的项目前期，比如 5G 基站大量使用 FPGA 芯片作为过渡。这是由芯片设计生产的流程所决定的，如果单纯生产一颗芯片用来测试，其代价过于昂贵，前期的设计验证费用加上芯片生产费用。而且如果生产出的芯片不能达到预期效果，还要重新来过。所以芯片生产要经过严密的后期实地测试才可以。

人们把目光投向了半定制芯片，FPGA 就是非常成功的例子。FPGA 既是芯片设计和芯片验证的模板，也是项目前期调试时芯片的替代品，很好的解决了芯片生产成本高的问题。而且由于 FPGA 具备高并行性、可现场编程以及可移植性等特点，FPGA 俨然成为了研究的热点。

以往的算法研究，从最初的深度学习，到最近的 YOLO（You Only Look Once）v4 网络，算法的深度和广度不断增加，随之而来的是精度的提高、占用空间增加以及运行速度的减慢。在很多应用场景下，需要牺牲一些精度来提高运行速度，比如交通智慧数据中心使用路况相机抓拍识别套牌车的车牌等。这些场景对系统的实时性有着很高的要求。所以像最近的深度学习网络，庞大的识别网络虽然有着很高的精度，但系统的运行速度难以满足实际要求，运行时间只能在 s 级别，所以只能简化网络。而 FPGA 由于具备高并行性的特点，执行算法的速度可以达到 us 级别。这是由于深度学习网络是由很多层组成的，FPGA 按照时钟的时序执行每一层的计算，而每一层的所有通道的计算都是同时进行的，这大大提高了运行速度。而且如果适当地提高时钟频率，FPGA 的运行速度将会进一步提高。

1.4 卷积神经网络在 FPGA 平台实现的发展概况

目前，卷积神经网络（Convolutional Neural Network, CNN）在 FPGA 平台的实现主要有以下三种方法。

第一种是利用 Vivado HLS 一类的高层次综合软件，通过 C#语言编写和库文件调用实现其功能。然后由 Vivado HLS 封装成 IP，由 Vivado 调用 IP 实现其功能。简而言之，就是把 C#语言翻译成 HDL。这种方法的优点是可以大大缩短开发周期，开发人员不用非要会用 HDL，只需熟练 C#语言即可。但其缺点也很明显，它会浪费大量开发板上的资源，而且不好维护。

第二种是使用 PS（DSP 或 ARM）+PL（FPGA）的架构实现深度学习算法。一般 FPGA 作为数据采集端，负责数据采集和加速，DSP 或者 ARM（Advanced RISC Machine）作为数据处理端，负责数据处理和结果显示。这种方法的优点是可以利用 FPGA 并行性高的特点实现数据的高速采集，同

时 DSP 多核心同时运作，提高数据处理的速度。这种架构有很多不同的搭配。第一种是把 DSP 或 ARM 和 FPGA 放在同一块开发板上，两芯片之间通过 Rapid IO 通信；第二种是 ZYNQ 架构，它分为 PS 端和 PL 端，PS 端负责指令，PL 端负责数据处理；第三种是在 FPGA 芯片上集成一些软核，如 Microblaze，它类似于 DSP，但性能要远低于 DSP。这种架构可以更方便地进行模块化设计，对软件开发能力的要求比较高，也需要硬件开发的理解。它的缺点是当运行复杂的深度学习算法时，无法取得很好的实时性，而且此架构价格昂贵，功耗高。

第三种是利用 HDL 编写深度学习算法。它基于 FPGA 平台实现，按照时钟信号、层间通信信号以及其他使能信号实现层级递进的网络推演过程，而且每一层所有通道的数据都会同时出现，这就实现了数据的高速处理，大大提高了算法的实时性。但由于深度学习算法需要存储大量的中间数据，且 FPGA 片上 Block RAM 存储资源有限，故而仅可用 Block RAM 实现小的神经网络。当中间数据过多时，就要动用板上的 DDR（Double Data Rate）存储资源，那样算法实时性将会大打折扣。

在今天，许多 FPGA 开发工程师在做深度学习算法到 FPGA 平台的移植工作。但受限于片上存储资源，在移植大型深度学习算法时，工程师们选择使用 DDR 作为中间数据存储空间，但效果并不是很理想。原因就是 DDR 的读写很耗费时间。例如 YOLOv2 的主干网络用到的 Darknet-19，输入为 448*448*3 的彩色图像，它有 19 层中间图像，每经过一层就要读取并存储图像数据到 DDR 中。而在移植小型深度学习算法时，片上存储资源可以符合要求。本文所使用的 FPGA 为 K 系列的 325tffg676-2，片上存储资源为 445 个 Block RAM，每个 Block RAM 大小为 36Kb，所以满足小型深度学习算法的存储空间要求。

1.5 本文的主要研究内容

本课题的研究内容主要是针对 Lenet-5 在 FPGA 平台的部署进行展开，分别研究了适用于卷积计算的 Block RAM 读写地址生成、流水线设计、卷积池化计算模块规范化、有符号数计算防溢出设计、Lenet-5 网络训练以及网络检测等内容。系统架构如图 1-1 所示。下面分别简单介绍一下各项内容。

在字符识别算法的选择部分，本文将对不同环境下不同的图像预处理算法对字符识别系统工作特性的影响进行实验分析，并对 Lenet-5 卷积神经网络进行网络训练和权重量化，对于后续目标识别的项目提供了一定的实验基

础，对于以深度学习网络为手段实现的目标识别算法在 FPGA 平台的部署提供了很好的实验资料。

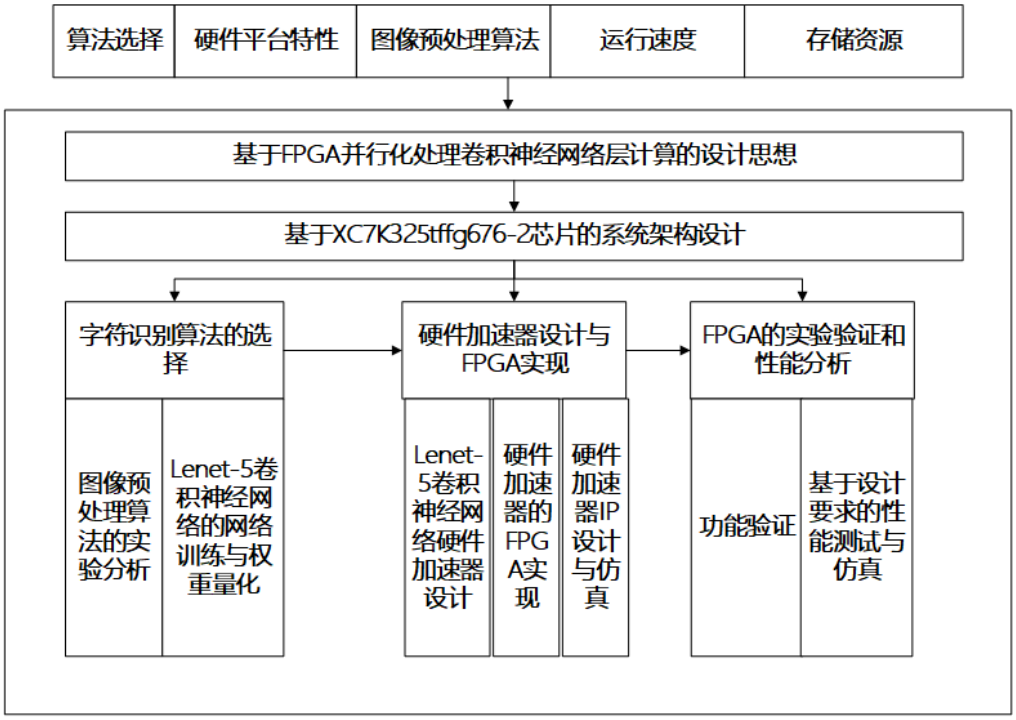


图 1-1 系统任务划分

Fig.1-1 System task division

在硬件加速器设计和 FPGA 实现部分，本文将在分析国内外 FPGA 平台部署卷积神经网络案例的基础上，以提高字符识别系统的实时性和卷积神经网络基本计算模块的复用性为目的，通过理论分析、仿真计算和上板验证对 Lenet-5 卷积神经网络在 FPGA 平台的推演过程进行深入研究。本文将会提出卷积神经网络在 FPGA 平台的实现方案，针对 XC7k325tffg-676-2FPGA 平台的特性提出了卷积计算地址读取方案——首地址锚定法；提出卷积神经网络在 FPGA 平台上层间通信和流水线设计的方案并设计实现；提出 Lenet-5 计算权值在 FPGA 平台的量化方案；规范化卷积神经网络基本计算模块并完成 IP 测试与封装，如 Convolution IP 核、Pooling IP 核以及 Full Connection IP 核等。这些工作将会为之后实现复杂的深度学习网络在 FPGA 平台的实现打下坚实的基础。

在 FPGA 的实验验证和性能分析部分，本文将在对卷积神经网络运算过程进行分析的基础上，在 FPGA 平台搭建 Lenet-5 卷积神经网络的硬件电路，并对其进行仿真调试，实现字符识别功能。针对卷积和池化计算的特点，提出图像数据读取地址生成方案——首地址锚定法；通过在不同计算层间加入

使能信号，实现 Lenet-5 卷积神经网络的层间通信和计算协同；通过研究补码计算规律，解决数值计算的溢出问题；通过在相邻计算层间加入 Block RAM 实现了网络中间数据的存储和网络流水线设计，实现系统同时图像连续处理，并以此提高系统的运行速度。

第 2 章 图像处理算法及 FPGA 简介

2.1 引言

视觉是人类最大的感官，它可以帮助我们更好的感受世界。同样，对于计算机来说，图像数据可以更快、更好地传递大量信息。但是，在工业相机拍摄图像时，由于强光照干扰、光照不足或者传感器过热等因素，产生的图像会带有一些噪声，这会严重影响到最终该系统的检测结果。为了滤除噪声，考虑在图像进入 Lenet-5 卷积神经网络之前对图像进行噪声过滤处理，如以图像二值化为滤波器的背景滤波算法、以均值构建滤波器实现尖锐噪声滤波的均值滤波、以中值为滤波器实现中低频噪声滤波的中值滤波以及通过局部分析实现噪声抑制的小波变换等。如果图片未经处理就输入网络，误差会通过网络传递，对检测结果产生很严重的影响。所以一般图片要先经过滤波处理，去除图片中的背景噪声，再进入下一步的处理。

在图像滤波之后一般会采用图像增强算法来提升图像品质。这样处理图像可以提高目标字符与背景噪声之间的差异性，更好的突出字符细节，以便于后续 FPGA 平台上 Lenet-5 卷积神经网络的识别。从符合机器视觉特性、噪声抑制以及信息熵最大化的角度出发，实现图像增强算法。

本课题主要研究 Lenet-5 字符识别，所以会涉及到字符分割相关内容。字符分割是字符识别的基础，字符分割的效果将会大大影响字符识别的精度。字符分割主要有间隙法和投影法两种方法。

在图像产生时，因为光照、传感器等因素会使得生成的图像会带有噪声信号。如果使用原始图像进行字符识别任务，Lenet-5 字符识别的精度将会大大降低。因此在机器学习领域，一般会在图像生成之后进行图像预处理操作。图像预处理主要包括图像滤波、图像增强以及字符分割。

2.2 图像滤波算法

图像滤波算法是通过一些数据分析方法，利用图像中目标物体与背景噪声之间的差异，实现滤波去噪的算法。可以根据实际实验中产生的不同类型的噪声选用最优的图像滤波算法。图像滤波可以滤除原始图像中的噪声信号，起到提高字符识别精度和防止噪声放大的作用。

2.2.1 均值滤波算法

均值滤波算法主要用来滤除频率比较低的噪声信号，而对于频率比较高的噪声信号的效果并不明显，允许通过低频信号，滤除高频信号。均值滤波器可以滤除图像尖锐噪声，使图像变得更加平滑和模糊。

均值滤波的基本原理是先计算输入图像单元的固定大小的邻域内的所有单元的平均值，然后把结果作为在输出图像的相应位置的单元的值。均值滤波窗口就是输入图像中一个 3*3 的邻域，要计算邻域中所有像素点的平均值，然后将计算得到的数值放到 p5 在输出图像上的对应单元。计算公式如(2-1)所示。计算结束之后的图像会变得模糊，同时高频尖锐噪声得到滤除。

$$f'(x，y)=\frac{1}{9}\sum_{(s，t)\in S}f(x+s,y+t) \tag{2-1}$$

式中 S——滤波滑动窗口；

$(s，t)$——滤波滑动窗口 S 上的点；

$f'(x，y)$——输出图像各个像素点的像素值。

2.2.2 中值滤波算法

在处理频率比较低的噪声信号时，中值滤波是一个比较好的选择。而当对高频噪声进行处理时，高斯滤波的噪声滤除效果要优于中值滤波。高斯滤波的缺点在于会使图像变得模糊，使原图像的边缘信息损失。

中值滤波，简单来说就是取固定大小的邻域内的像素数据的中值点代替该邻域的像素值，从而实现滤波。设$\{f'(x，y)\}$为数字图像各点的像素点，$\{f'(x，y)\}$为输出数字图像各点的像素值。滤波滑动窗口 S 内的中值滤波公式如(2-2)所示。

$$f'(x，y)=Med\{f(x+s,y+t)，(s，t)\in S\} \tag{2-2}$$

式中 S——滤波滑动窗口；

（$s，t$）——滤波滑动窗口 S 上的点；

$f'(x，y)$——输出图像各个像素点的像素值。

2.2.3 高斯滤波算法

高斯滤波是针对高斯噪声的滤波算法，如公式(2-3)所示为二维正态分布函数。

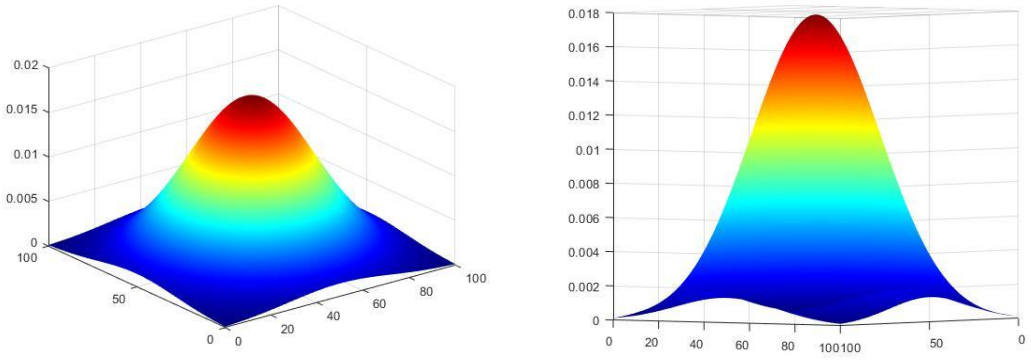
$$G(x，y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2-3}$$

式中 G（x，y）——高斯函数；

σ——高斯函数的标准差；

(x，y)——高斯函数二维点的坐标。

二维高斯函数的图像如图 2-1 所示，高斯噪声就是一些符合正态分布的图像噪声信号。而在处理高斯噪声时，一般会使用高斯滤波窗口对数字图像进行处理。高斯滤波窗口的幅度满足二维高斯分布，如图 2-1（b）所示。



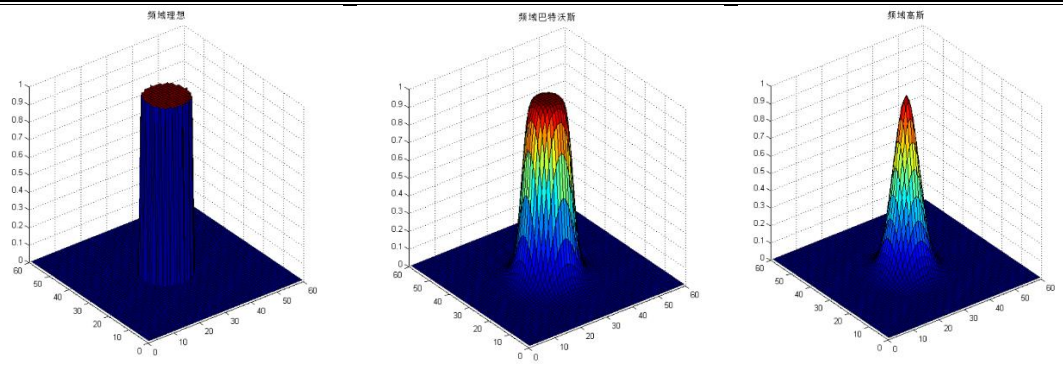
(a) 三维高斯分布斜视图　　(b) 三维高斯分布侧视图

图 2-1 三维高斯分布斜视图及侧视图

Fig.2-1 3d Gaussian oblique view and side view

高斯滤波的原理就是用一个滑窗对整张图片进行扫描，每当滑窗中心扫描到图片的某一个像素点时，用该滑窗邻域内的计算结果代替该像素。高斯滤波可以避免振铃现象，优于理想滤波器。

如果一个滤波器在频域内有突然的变化，这会使得产生的结果图在边缘处变得模糊。图 2-2 为三种滤波器的频域波形对比。理想滤波器在频域中的图像像一个圆柱体，在圆柱体的上边缘部分产生了 90 度转变，而巴特沃斯滤波器在频域中的图像像一个小“山丘”，在“山丘”顶端也产生了相应的转变，如图所示，所以经过它们处理之后产生的图片都会使图像的边缘信息变得模糊。而高斯滤波器在时域和频域均是高斯函数，所以它处理之后的图片不会产生上述现象。



(a) Ideal filter (b) Butterworth filter (c) Gaussian filter

图 2-2 不同滤波器频域波形对比

Fig.2-2 Comparison of waveforms in frequency domain of different filters

2.2.4 小波变换滤波算法

小波变换是一种通过数字图像信号和噪声信号在不同部分的分布规律，调整门限阈值，修改小波系数，实现图像噪声滤波的图像滤波算法。由于图像噪声大部分集中于图像的高频部分，经过小波分析后，图像噪声大部分集中在次低频、次高频以及高频部分，而高频部分几乎全部是噪声信号，所以可以将高频部分全部设为 0，然后对次低频和次高频部分进行噪声抑制，就可以实现噪声滤除的操作。

2.3 图像增强算法

图像增强算法就是通过一些数学分析工具，在后期可以滤除背景噪声的前提下，在一定尺度上提高目标的清晰度。图像增强的目的是使图像变得更易于人眼识别或者更易于机器学习。图像增强算法广泛应用于医疗影像领域和机器学习领域。在医疗领域，图像增强算法可以使 X 光等影像片变得更易于人眼识别。而在机器学习领域，图像增强算法可以使图像更易于机器学习，作为目标识别、目标跟踪的图像预处理操作。

图像增强算法通过对图像的像素进行数学分析操作，可以提高识别目标的清晰度。图像增强算法之前或之后应该有图像滤波去噪的步骤，防止背景噪声传递错误信息。图像增强算法对于普通光照条件和强光照条件下拍摄的图片可以起到增强目标信息的效果，但同时也会放大噪声。

2.3.1 直方图均衡（HE）算法

直方图均衡算法就是根据当一张图像的灰度分布均匀时图像最清晰这条理论，对目标图像的灰度分布进行调节，使得其结果图的灰度分布尽可能地均匀。

灰度直方图的概率计算公式如公式（2-4）所示。

$$P(k) = \frac{n_k}{n}, k = 0,1,\dots,L-1 \tag{2-4}$$

式中 $P(k)$——第 k 个灰度级灰度分布的概率；

n——图像像素点总个数；

n_k——第 k 个灰度级中像素点个数。

所有灰度级的概率和等于 1，如公式（2-5）所示。

$$\sum_{k=0}^{L-1} P(k) = 1 \tag{2-5}$$

式中 $P(k)$——第 k 个灰度级灰度分布的概率；

L——灰度级个数。

灰度直方图反映了图像灰度分布情况，当灰度直方图均匀分布时，图像最清晰。根据这个原理，只需使图像的直方图分布更加均匀就能使图像变得更加清晰。直方图均衡算法就是通过直方图变换实现灰度分布均匀从而达到图像增强目的的算法。它的基本原理是对目标图像的灰度分布进行调节，使得其结果图的灰度分布尽可能地均匀。这样就可以提高识别目标的清晰度，便于后续 Lenet-5 卷积神经网络的检测。

直方图均衡化算法的灰度级计算公式如（2-6）所示：

$$S(k) = \int_0^k P(r)dr \tag{2-6}$$

式中 $S(k)$——累积分布函数；

k——图像前 k 个灰度级；

$P(r)$——第 r 个灰度级的概率。

通过灰度级均衡化算法处理之后，原始图像的灰度分布会变得更加均匀，从而使图像更加清晰，灰度级均衡化的映射公式如（2-7）所示：

$$f(k) = (L-1) * S(k) \tag{2-7}$$

式中 $S(k)$——累积分布函数；

$f(k)$——灰度级映射函数；

L——灰度级个数。

虽然直方图均衡化算法的运行速度快、实现原理简单，但也存在图像灰度级均衡化之后整体图像亮度不均匀，细节信息丢失等缺点。

2.3.2 偏微分方程图像增强算法

图像增强算法一般分为两种，在时域增强或在频域增强。而偏微分方程算法是在梯度域进行图像增强，将图像的边缘、亮度、对比度等信息量化成梯度值进行处理，从而达到图像增强的效果。在实际应用中，由于光照、传感器等因素，获得的图像整体偏暗，目标字符和背景的差别不大。这时对图像进行直方图均衡化图像增强算法，虽然可以起到一定的效果，但这会使得原本图像中暗的地方更暗，甚至会大大损失目标字符信息。在这种情况下，可以选用偏微分方程算法进行图像增强。

偏微分方程图像增强算法就是从改善图像的细节纹理出发，将二维灰度图像转换到梯度域进行图像增强。为了不改变梯度的方向，使算法满足不变性的要求，变换后的梯度函数方程如（2-11）所示：

$$S=\left(1-\cos\left(\frac{||\nabla u||-\min_{||\nabla u||}}{\max_{||\nabla u||}-\min_{||\nabla u||}}*\pi\right)\right)*\frac{\max_{||\nabla u||}}{2}*\frac{\nabla u}{||\nabla u||} \tag{2-11}$$

式中 S——梯度值；

$||\nabla u||$——原图像的梯度函数的模；

$\min_{||\nabla u||}$——梯度模的最小值；

$\max_{||\nabla u||}$——梯度模的最大值。

经过梯度变换之后梯度场从$[\min_{||\nabla u||},\ \max_{||\nabla u||}]$到$[0,\ \max_{||\nabla u||}]$，这使得图像的梯度分布区间更广，更加均匀。这样就可以突出原来不太明显的梯度，使得图像对比度和边缘信息得到了加强，从而实现图像增强的效果。

2.4 字符校正与分割算法

字符分割主要有利用字符间隙分割字符的间隙法和利用水平像素投影和垂直像素投影实现字符分割的投影法两种方法。经过字符分割的图像将直接进入Lenet-5卷积神经网络，因此字符识别的效果将直接影响检测结果。

字符分割是字符识别中很重要的一步。如果要使用单字符识别的Lenet-5卷积神经网络，就必须在前端加上一个字符分割器，否则就会出现错误识别等情况。比如，数字8的一半可能会被识别生成3。这就是没有字符分割的系统会产生的问题，而且字符分割之前还需要字符的校正，从而保证字符的识别精确无误。

2.4.1 字符校正算法

字符校正算法是一种通过校正字符空间位置提高字符分割精度的算法。字符校正算法的作用主要是校正倾斜字符，未在图像中心的字符等。字符校正算法主要有倾斜角度探测、旋转校正以及切向校正等几个步骤。

（1）倾斜角度检测算法 倾斜角度检测算法主要是检测图像的倾斜角度，为后续的字符校正算法提供参考。倾斜角度检测算法主要有边缘法和直方图法两种。边缘法主要是通过大目标的边缘计算倾斜角度。直方图法主要是把图像转换为一张灰度直方图，计算直方图灰度分布的方差。同时不断调整图像的倾斜角度，当直方图灰度分布方差最大时，图像处于水平状态。

（2）旋转校正 计算得到原始图像的倾斜角度之后，需要通过以下公式对图像进行旋转校正。以逆时针为正方向，根据之前检测到的图像的倾斜角度旋转图像。旋转校正公式如公式（2-13）所示：

$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta - y\cos\theta \end{cases} \tag{2-13}$$

式中 $(x'，y')$——校正之后图像像素坐标；

$(x，y)$——校正之前图像像素坐标；

θ——图像倾斜角度。

（3）切向校正 在笛卡尔坐标系下，当一个坐标轴不变而另一坐标轴角度发生改变时，这种叫做切向畸变。所以在采集到原始图像之后，字符校正模块中除了要校正图像的旋转倾斜外，还有图像扫描产生的畸变。切向校正公式如下：

$$\begin{cases} x' = x + y\cot\theta \\ y' = y \end{cases} \tag{2-14}$$

式中 $(x'，y')$——校正之后图像像素坐标；

$(x，y)$——校正之前图像像素坐标；

θ——图像倾斜角度。

2.4.2 投影法字符分割算法

投影法字符分割的实现原理就是当一张图片的垂直投影的黑色像素点直方图的某一个邻域内的黑色像素点个数为 0 时，说明这个位置是两个字符之间的空隙处；当一张照片的水平投影的黑色像素点直方图的一个区间都有黑色像素点时，说明这个位置是一行字符而不是噪声。

首先，对要检测的图像进行二值化，即低于某一阈值的像素点置为 0，高于某一阈值的像素点置为 1。之后，对二值化之后的图像进行垂直方向和水平方向的黑色像素点直方图绘制。最后，根据黑色像素点直方图进行字符分割。

2.5 Lenet-5 卷积神经网络

1998 年，Lenet-5 卷积神经网络问世之后，实现了成功的应用。经过手写字符数据集训练网络获得合适的权重，这种卷积神经网络可以非常高效的识别手写体字符。因为 Lenet-5 的成功应用于支票数字识别并获得商业价值，深度学习逐渐普及。Lenet-5 卷积神经网络的提出和成功应用，推动了深度学习算法的发展。

卷积（Convolution）计算因为其优异的二维图像信息提取功能在深度学习领域广泛使用。可以说，深度学习的发展离不开卷积计算的应用。在最初深度学习产生时，就选用卷积运算进行特征提取。这是经过不断的试验得出的结论：卷积计算对与提取二维图片特征来说是最好的选择。在 Lenet-5 之后，深度学习网络基本都是使用卷积计算作为特征提取器。与高等数学中的卷积不同，图像处理中的卷积操作实际上是一个滑窗内像素的内积，也就是“互相关”。它的过滤器不经过反转，而高数中的卷积计算的过滤器需要反转之后再进行运算。这就是它们之间细微的差异。

卷积计算的加入是为了滤波或者提取图片特征，不同的滤波权重可以滤除不同噪声，一张图像经过不同的卷积权重计算得到的结果图会保留目标的特征信息。网络训练的意义就是找到一组最优卷积核对要提取的图片特征敏感。

2.5.1 单字符卷积神经网络识别

单字符卷积神经网络识别通过使用一个经过训练的分类器实现字符识别。在本文中的 Lenet-5 算法中，全连接层是一个分类器。在网络训练过程中，依靠特征提取器实现图像特征提取。但是这种图像识别方法存在一定的缺陷。

第一个问题是在将图片送入卷积神经网络之前，图片需要被裁剪成固定的大小，如本文的图片大小为 32*32。在原作者在文章中说没有找到完美的方法去解决这个问题。

第二个问题是输入图像的拓扑结构被忽略了，该网络的信息提取并不会保留图片的拓扑结构。这是因为在卷积计算提取图像信息时，卷积计算的滑窗邻域大小为 5*5，而不是整张图片。

Lenet-5 是一种用于平面二维目标识别的卷积神经网络，它的基本结构如图 2-3 所示。要对平面二维目标进行识别，这要求输入图像的大小必须固定为

32*32，而且识别目标尽可能地在图像中间位置。每一层的特征图都由上一层的对应特征图经过相应计算得到。且在一张图像上进行特征提取的权重是相同的。固定大小的滑窗可以提取图像的目标信息，通过不断的提取整合，得到与所提取的特征最相似的结果。如图 2-3 所示，Lenet-5 卷积神经网络的第一层卷积计算产生了 6 个特征图。特征图中的每个数据都是通过上一层图像中对应数据所在的大小为 5*5 的滑窗邻域内的所有数据的加权平均值和偏差系数的加和计算结果。而每次滑窗邻域移动一步，所以相邻的滑窗邻域的有些数据是相同的。比如，在卷积神经网络的第一层卷积计算时，相邻的滑窗邻域的 20 个数据是相同的。而且每一张图片进行卷积计算使用同一个权重和偏差系数。而同一层中的不同特征图所使用的权重和偏差系数是不同的，这是为了在不同的特征图中保留不同部分的图像特征。以 Lenet-5 卷积神经网络的网络组成结构为例，第一层卷积计算通过 6 个卷积权重和偏差系数生成 6 个特征提取通道，可以得到 6 个特征图。使用一个滑窗邻域对一张图片进行扫描，并将滑窗邻域内的加权平均值和偏差系数的加和作为对应位置的数据，这个计算过程类似于高等数学中的卷积计算，所以这个特征提取层就叫做卷积层。如果对输入的图像数据进行平移，那么这种变化只会使卷积计算的结果发生相应的平移，而不会影响图像数据，这也是卷积计算作为特征提取的很重要的一个原因。

、

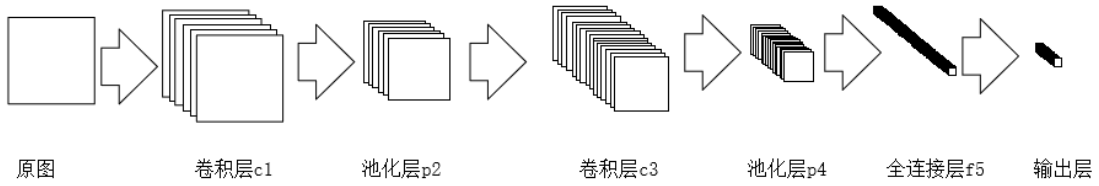


图 2-3 Lenet-5 卷积神经网络结构

Fig.2-3 Lenet-5 convolutional neural network structure

利用卷积神经网络进行目标识别时，重点在于目标的识别，而不是目标的位置。卷积神经网络所关注的重点是目标的不同特征之间的相对位置，比如，如果卷积神经网络检测到图像左边有个拐角，中间有一条竖线和横线，那么就可以基本确定这个字符是“4”。当通过第一层卷积层生成的 6 个特征提取通道获得保留图像的不同特征的 6 个特征图之后，为了降低计算量，在第一层卷积层之后加入降采样层，也就是池化层。本课题选择的是最大值池化。第二层池化层生成 6 个池化通道分别对上一层产生的特征图进行降采样处理。池化计算的滑窗邻域大小为 2*2，对滑窗邻域内的数据进行最大值池化计算得到的结果和偏差系数的加和作为结果图对应位置的数据。当然，在数据存入之前，要通

过一个激活函数 Relu 计算。在池化层，相邻的滑窗邻域互相不重叠。所以经过池化计算之后的结果图的行数和列数是前一层的一半。

卷积神经网络的所有权值都是通过样本不断进行学习获得的。一张图片使用同样的权重虽然降低了计算量，减少了权重训练的个数，但同时也降低了网络学习的能力。

2.5.2 Lenet-5 卷积神经网络架构

本节将主要介绍字符识别网络的架构。在进行网络训练时，选择使用 minst 数据集进行训练和测试。该数据集中图像的分辨率都比网络的输入图像的分辨率小。网络的输入图像的分辨率为 32*32，而数据集中的图像的分辨率都小于它，大概在 20*20 左右。这可以使网络学习更多的字符特征，从而获得更有效的权重系数和偏差系数。

Lenet-5 卷积神经网络的第一层卷积层包含由 6 个二维卷积通道生成的 6 个大小为 28*28 特征图。每个特征图中的每个数据都与原始图像中对应位置的数据所在的滑窗邻域相连接。

Lenet-5 卷积神经网络的第二层池化层包含由 6 个特征图降采样通道生成的 6 个大小为 14*14 的特征图。每个特征图中的每个数据都与第一层卷积层中对应位置数据所在的大小为 2*2 的滑窗邻域相连接。池化的过程就是选择滑窗邻域内的 4 个数据的最大值，经过偏差系数加和之后，再经过 relu 激活函数激活就可以得到结果图。这一步中的所有的滑窗邻域是互不重叠的。

Lenet-5 卷积神经网络的第三层卷积层包含由 16 个特征提取通道生成的 16 个大小为 10*10 的特征图。

Lenet-5 卷积神经网络的第四层池化层包含由 16 个特征图降采样通道生成的 16 个大小为 5*5 的特征图。每个特征图中的每个数据都与第三层卷积层对应特征图的对应位置的数据所在的大小为 2*2 的滑窗邻域相连接。

Lenet-5 卷积神经网络的第五层卷积全连接层包含由 120 个全连接通道生成的 120 个特征图。每个特征图的每个数据都连接到所有第四层池化层的特征图的对应位置的数据所在的大小为 5*5 的滑窗邻域。因为第四层池化层得到的特征图的大小为 5*5，所以第五层全连接层的特征图的大小为 1*1。

如果跟原作者一样使用 Sigmoid 压缩函数，这样处理会使图像产生梯度消失的现象，造成图像特征缺失。现在普遍使用 Relu 激活函数。

2.6 FPGA 平台简介

本课题使用的 FPGA 平台为创龙公司发布的 C6678F-EVM 开发板。开发板

上有一颗型号为 C6678 的 DSP 芯片和型号为 XC7K325tffg676-2 的 FPGA 芯片，还有一些如 CPLD、ROM、RAM、晶振、电源、LED 等硬件资源，并通过工业级高速 B2B 连接器引出 IO 管脚。实验所使用的平台如图 2-4 所示。

图 2-4 系统组成实物图

Fig.2-4 Physical diagram of system composition

本课题主要通过工业相机对图像信息进行采集，然后图像信号以 LVDS 的形式传递到 TL288AP 转接卡上，通过芯片处理将 LVDS 转换为 28 位的并行电平信号。然后转接卡通过欧式母座将并行电平信号传入 FPGA 芯片。由于这 28 位的并行电平信号是根据高速图像传输协议 Cameralink 传输的，所以在信号调用之前要对信号进行解码。经过图像协议解码之后，就可以得到位宽为 8 的图像像素信号和图像使能信号，根据这些信号的时序关系对图像信息进行编辑。可以对图像进行降噪、增强、字符分割等处理，只是算法的运行要根据硬件平台的特性进行计算。

2.7 本章小结

本章节介绍了图像处理的几个重要的步骤，每一步有不同、相应的分工。图像滤波主要是起到抑制噪声的作用。图像增强可以使图像整体对比度提高，凸显图像细节。字符分割可以实现单个字符的定位和分割，实现字符的中心化，提高 Lenet-5 卷积神经网络的识别精度，这是加入字符分割的一个很重要的原因。

当然，系统中也可以不加入字符分割模块。字符分割模块的有无取决于 Lenet-5 卷积神经网络的训练方法。如果 Lenet-5 卷积神经网络的训练样本是字

符串，那么该卷积神经网络是训练用来识别字符串，因此不需要字符分割。如果 Lenet-5 卷积神经网络的训练样本是中心化的单字符，那么该系统就是被训练用来识别单字符，因此必须要字符分割模块对图像进行预处理。

第 3 章 字符识别系统方案设计

3.1 引言

本课题的字符识别流程图如图 3-1 所示。第一层是字符识别系统方案设计的设计要求：算法选择、硬件平台特性、图像预处理算法、运行速度以及存储资源。首先根据 FPGA 芯片的硬件特性提出一个加速卷积神经网络层计算的设计思想。然后根据 XC7K325tffg676-2 的 FPGA 平台的资源及算力情况进行字符识别系统的架构设计。最后，对系统功能模块进行划分，分为字符识别算法的选择、硬件加速器设计与 FPGA 实现以及 FPGA 的实验验证和性能分析。字符识别算法的选择分为图像预处理算法的实验分析和 Lenet-5 卷积神经网络的网络训练和权重量化；硬件加速器设计与 FPGA 实现分为 Lenet-5 卷积神经网络硬件加速器设计、硬件加速器的 FPGA 设计以及硬件加速器 IP 设计与仿真；FPGA 的实验验证和性能分析分为功能验证和基于设计要求的性能测试与仿真。

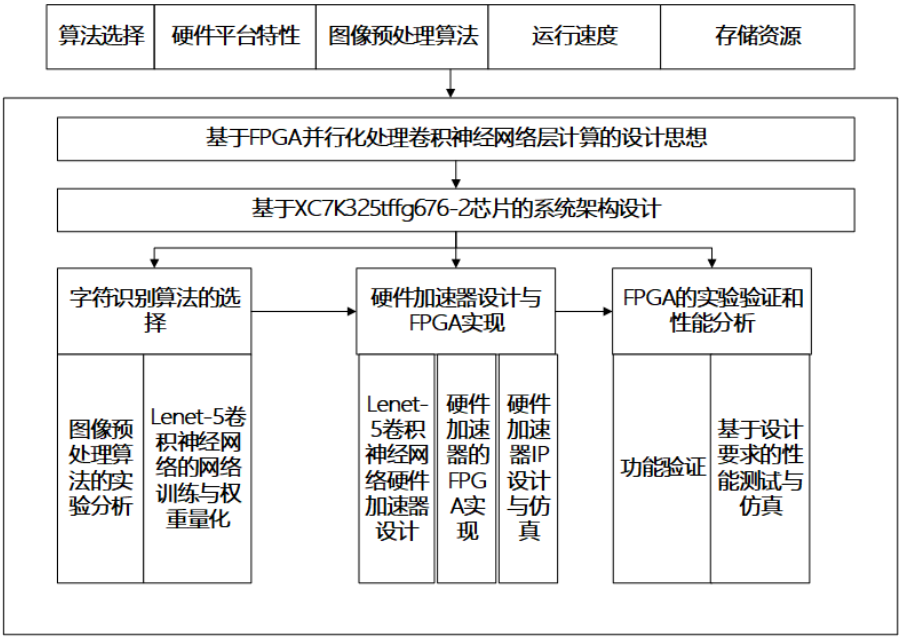


图 3-1 字符识别系统研究流程图

Fig.3-1 Research flow chart of character recognition system

本课题旨在实现一个抗干扰能力强、运行速度快而又识别精度高的字符识别系统。该系统主要包括负责采集原始图像的图像生成模块、负责对图像进行初步处理的图像预处理模块和负责对图像中目标字符进行识别的图像识别模块三大部分，图像生成模块主要是获取目标的图像信息，图像预处理模块主要通

过一些数学分析算法实现图像质量的优化，图像识别模块主要是对经过初步处理的目标特征比较明显的图像进行特征学习和图像识别推演。字符识别系统的总体设计框图如图 3-2 所示。

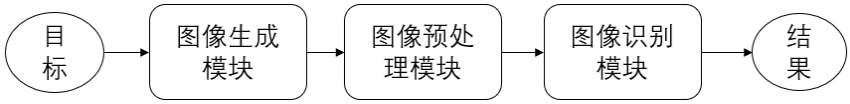


图 3-2 字符识别系统总体设计
Fig.3-2 Character recognition system overall design

3.2 图像生成模块

图像生成模块由相机和图像协议解码模块组成，如图3-3所示。相机选用微视图像的 MVC1381SAM-POCL60 工业相机，该相机是一款支持 Camera link 协议的热噪声抑制工业相机。它的拍摄帧率为每秒 60 帧，大小为 1280*1024。要想获得相机拍摄到的图像信息，就要对 Camera link 协议进行解码，把原始图像信号转换为 TTL 并行的电平信号。

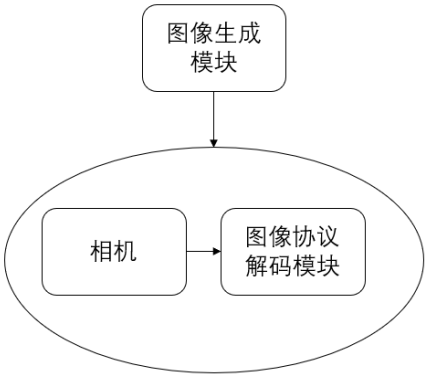


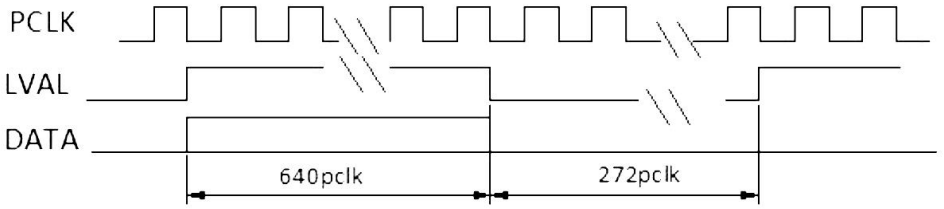
图 3-3 图像生成模块组成
Fig.3-3 Image generation module composition

3.2.1 Camera link 协议解码

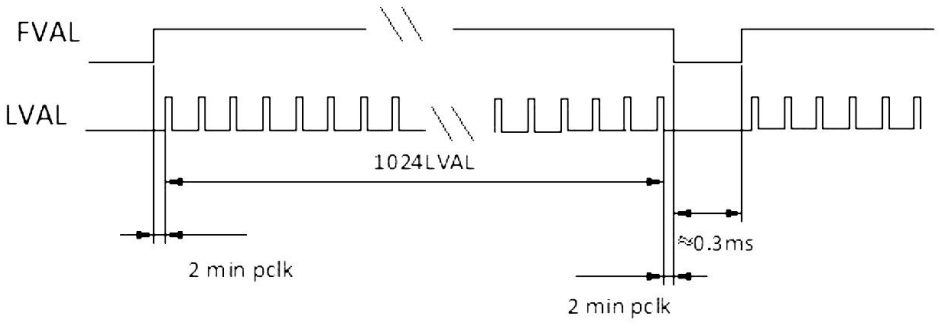
Camera Link 协议是由美国自动化工业学会（American Institute of Automation， AIA）发布的一种专门用于快速地传输视频数据的协议。Camera Link 协议是以 Channel Link 协议为基础定制的协议。类似于 Channel Link 协议，Camera Link 协议也以低压差分信号(Low Voltage Differential Signal, LVDS)为数据传输载体实现数据的快速传输，这使得图像信息可以在设备间实现高速传输。相机的 Camera link 接口如图 3-4 所示。

图 3-4 相机的 Camera link 接口
Fig.3-4 Camera's Camera link interface

Channel link 技术是 Camera link 的一部分。图像数据和图像使能信号以低压差分信号的形式通过 Channel link 芯片进行传输。本课题中主要用到协议中的三个信号：FVAL - frame Valid、LVAL - line Valid 和 DVAL - data Valid。所有的使能信号都由相机的 Channel link 芯片产生。相机产生的 LVAL 信号和 FVAL 信号的时序关系如图 3-5 所示。



(a) 行有效信号与数据时序



(b) 行有效信号与帧有效信号时序

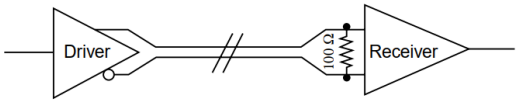
图 3-5 图像数据及图像使能信号时序
Fig.3-5 Image data and image enabling signal timing

根据 Camera link 协议的比特分配，可以在 FPGA 上得到图像像素信号。其

中包含了 FPGA 对应的引脚、输入信号的名字、LVAL，FVAL 以及 DVAL。由于本课题使用的相机只支持 Base 模式，所以这里用到了 portA、portB、portC 三个端口，每个端口传输 8 比特的像素数据，即 8 位二进制数据表示 0-255 之间的像素值。而由于本项目使用的相机采用双 Tap 传输，故仅有 portA、portB 有效。

3.2.2 低压差分信号（LVDS）解析

该部分把低压差分信号转换为并行的视频数据。低压差分信号（LVDS）依靠一对低电压的信号进行高速数据传输。而且由于 LVDS 的低压振幅，LVDS 的功耗相比之下更低，而且具有良好的抗噪性，能够保持信号不被干扰。



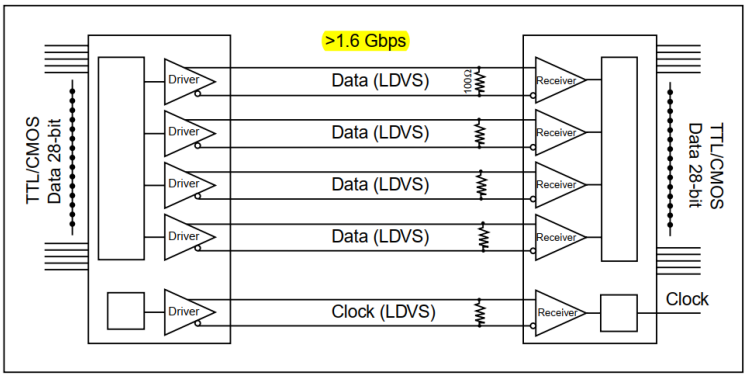


图 3-6 Camera link 信号传输
Fig.3-6 Camera Link signal transmission

本课题采用的是当前在图像数据传输中比较流行的 7:1LVDS 接口。如图 3-6 所示。

下面介绍 FPGA 中 1:7 串行解码和数据接收的内容。

数据流是传入时钟速率的倍数(x7)，时钟信号是接收数据时分帧的依据。在一个时钟周期内，数据的行状态改变了 7 次。这种 7:1 的接口广泛应用于相机，平板电视和显示器。在这些应用场合中，接收的时钟在 PLL 或 MMCM 中进行 7 倍倍频，由此产生的高速时钟通过一个全局的缓冲器(BUFG)、I/O 缓冲器(BUFIO)或水平时钟缓冲器(BUFH)发送给 ISERDES 逻辑。时钟缓冲器在 SDR 和 DDR 模式下都可以使用。

通过一个由状态机训练得到的 IODELAYE2 元件将输入数据提供给 ISERDES 逻辑，使得数据相对于样本时钟的延迟尽可能小于一半单位间隔(UI)时钟。这是通过使用输入时钟作为训练标准实现的。从输入像素时钟获得的帧

数据也与 ISERDES 的位滑移特性一起用来生成与原始数据具有相同的字节关系的并行数据传输数据。最后，将这些串行解码的并行数据按原始输入时钟的速度输出。

这里有两个设计上的问题需要注意。当使用 SDR 采样时钟时，ISERDES 是以 1:7 模式直接使用。ISERDES 支持的唯一时钟元件组合是两个 BUFG，或者两个 BUFH 或者一个 BUFIO 加一个 BUFR。使用 BUFG 的优点是可以在全域时钟上可以使用并行数据。使用 BUFR 或 BUFH，仅可在当前时钟域内使用并行数据。当多个通道运行在相同的像素时钟频率下，它们可以被放置在相同的 I/O Bank 并且可以使用所有时钟选项，或者它们可以放置在不同的 I/O Bank 并且只可以使用 BUFG 时钟。

当使用 DDR 采样时钟时，ISERDES 以 1:4 模式使用，1:7 模式需要使用基于分布式 RAM 的变速箱进行模拟。这种方法需要三个时钟域。所述采样时钟，将采样时钟除以 4，并将采样时钟除以 7，也即初始输入的像素时钟。与 SDR 时钟采样的情况一样，当多个通道运行在相同的像素时钟频率下，它们可以被放置在相同的 I/O Bank 并且可以使用所有时钟选项，或者它们可以放置在不同的 I/O Bank 并且只可以使用 BUFG 时钟。

在这两种情况下，当锁相环用于时钟倍频时，只有 BUFH 或 BUFG 时钟是可以使用的，因为这些缓冲区是只有锁相环可以访问的类型。

3.3 字符识别模块

字符识别模块主要分为图像数据读取地址生成方案、存储方案选择、流水线设计、层间通信及计算协同以及各个计算模块的设计等几个问题。首先根据 FPGA 平台 XC7K325tffg676-2 的硬件特性对实现卷积神经网络进行方案分析，其次，分别对图像数据读取地址生成方案、存储方案选择、流水线设计、层间通信及计算协同等问题进行解决。最后，完成计算模块的设计实现。

3.3.1 FPGA 平台实现 Lenet-5 卷积神经网络方案分析

FPGA 平台有很多的组合逻辑资源、寄存器资源以及存储资源。每进行一层卷积计算时，需要同时对多张图片进行卷积计算，可以通过增加寄存器使用个数增加卷积计算的并行性，这可以大大提高网络推演的速度。FPGA 片上的存储资源为 Block RAM，Block RAM 的存储空间比较小，每个存储空间大小为 36Kbit。本课题所使用的 FPGA 型号为 XC7K325tffg676-2，其片上 Block RAM 的个数为 445 个。所有的 Block RAM 的存储空间仅可存储一张分辨率为 720p 的灰色图片。

Block RAM 的使用需要调用 Block RAM 的 ip 核 Block RAM Memory Generator。Block RAM 可以先读后写。当给出读写地址信号、读写使能信号和读写状态信号之后，数据的读写就开始了。除此之外，在 FPGA 平台上还有 DDR 存储器。DDR 存储器需要通过 MIG ip 核调用。本课题所使用的 FPGA 平台的片上存储资源有 438 个 Block RAM 和两个 DDR。

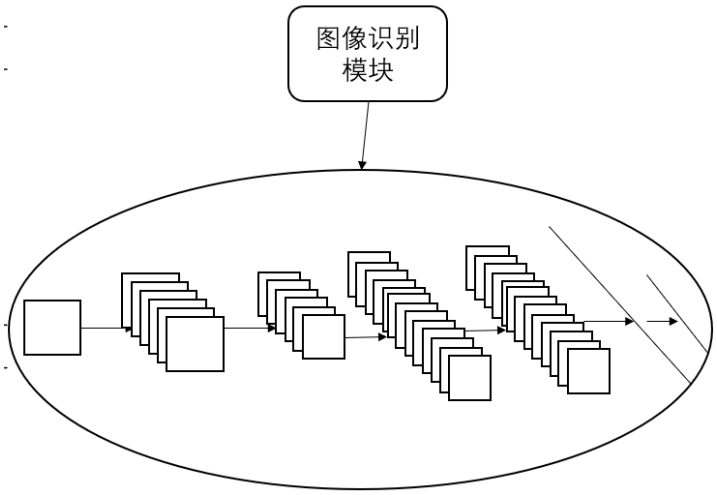


图 3-7 图像识别模块结构

Fig.3-7 Image recognition module structure

FPGA 平台有很多的组合逻辑和寄存器资源，这非常有利于 Lenet-5 卷积神经网络的搭建。由于 Lenet-5 卷积神经网络的中间数据比较多，这会非常占用存储空间。FPGA 平台上的 Block RAM 的存储空间比较小，但调用方便且读写速度快；DDR 的存储空间大，但调用麻烦且读写速度慢。所以本课题采用片上 Block RAM 完成 Lenet-5 卷积神经网络的中间数据存储。为了不超出 Block RAM 的存储空间，Lenet-5 卷积神经网络选择输入图像大小为 32*32。Lenet-5 卷积神经网络的结构如图 3-7 所示。

根据 Lenet-5 卷积神经网络的运行模式来看，中间数据必须要经过一次 Block RAM 寄存。所以本课题拟在每两个层之间加入一个 Block RAM 进行中间数据存储。一方面，这样可以加强数据读取的稳定性，使系统不会出现数据误读的情况；另一方面，这使 Lenet-5 卷积神经网络完成了一个流水线设计，类似于芯片内部加入寄存器一样，这样做可以大大提高系统的运行速度。

3.3.2 读写地址生成规律

卷积计算是一个“滑窗”操作，就是用一个扫描窗口对图像二维矩阵进行扫描，扫描窗口与图像二维矩阵重合的位置要进行加权平均值计算，这样得到的结果组成一个矩阵，就是图像卷积计算的结果。卷积计算的过程如图 3-8 所

示。

图片数据存入 Block RAM 时是按照数据顺序存入的，即第一个数据的地址为 0，第二个数据的地址为 1 等以此类推。而在卷积计算时，每次计算所需要的数据是扫描窗口内的所有的图像像素数据。而这些数据的 Block RAM 内的存储地址是不连续的。如图 3-8（a）所示，扫描窗口第一行的第一个数据的地址为 0，第一行第二个数据的地址为 1，第一行第三个数据的地址为 2，而第二行的第一个数据的地址为 6，第二行第二个数据的地址为 7，第二行第三个数据的地址为 8，第三行的第一个数据的地址为 12，第三行第二个数据的地址为 13，第三行第三个数据的地址为 14，很明显可以看出来，扫描窗口内的图像像素数据的存储地址虽然是不连续的，但遵循一定的规律。在一个大小为 m*m 的图像上进行“滑窗”操作，第 n 行的图像像素的首地址是列数 m 乘上（n - 1），如公式（3-1）所示。所以在数据读取这部分要设计一个符合卷积计算规律的数据读取地址生成器。

$$ADDR = m * (n - 1) \tag{3-1}$$

式中 *ADDR*——图像像素读取地址；

m——图像列数；

n——卷积计算所在行数。

依据以上数据读取的规律，本课题提出了一种适用于卷积计算“滑窗”操作的图像数据地址生成方案。即按照“读满一列再读一行”的顺序进行数据读取地址生成。本课题所用到的窗口大小为 5*5，每次卷积计算需要读取 25 个图像像素数据。在第一行的五个数据的数据读取地址生成结束之后，需要生成第二行的五个数据的数据读取地址，如图3-8所示（图中每一个格子里上面是像素数据，下面是像素存储地址），而由前述可知，第一行五个数据的地址为 0、1、2、3、4、5，第二行的五个数据地址为 6、7、8、9、10、11，第二行数据的地址与之前的第一行的五个数据的地址并不连续。但从图3-9中可以看出，虽然行与行的数据读取地址不连续。但每一行的数据地址都是连续的，而且不同行首地址遵循一定的规律，如公式（3-1）所示。所以根据这个原理，本课题提出了一种适用于卷积计算的图像数据地址读取方法——首地址锚定法。详细讲解将会在第 5 章展开。

0	1	0	9	0	B
2	4	0	0	a	a
1	3	0	1	0	0
0	1	0	0	0	0
0	2	0	0	0	0
0	0	0	0	0	0

(a) 卷积计算第一步

0	1	0	9	0	B
2	4	0	0	a	a
1	3	0	1	0	0
0	1	0	0	0	0
0	2	0	0	0	0
0	0	0	0	0	0

(b) 卷积计算第二步

0	1	0	9	0	B
2	4	0	0	a	a
1	3	0	1	0	0
0	1	0	0	0	0
0	2	0	0	0	0
0	0	0	0	0	0

(c) 卷积计算第三步

0	1	0	9	0	B
2	4	0	0	a	a
1	3	0	1	0	0
0	1	0	0	0	0
0	2	0	0	0	0
0	0	0	0	0	0

(d) 卷积计算第四步

图 3-8 卷积计算原理

Fig.3-8 Principle of convolution computation

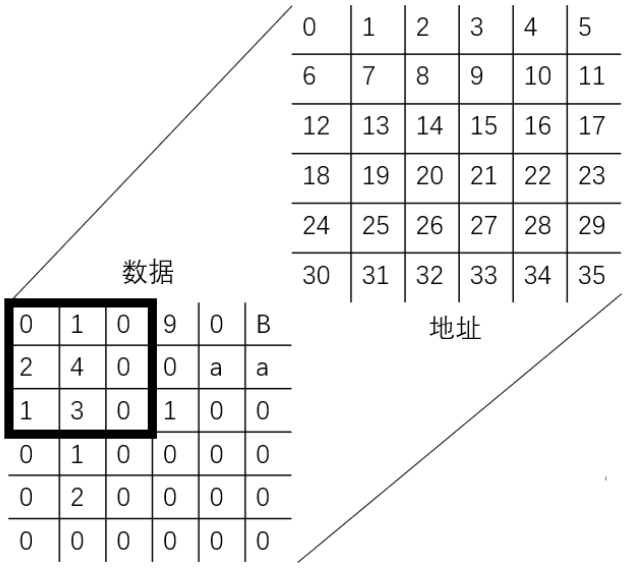


图 3-9 数据读取地址原理

Fig.3-9 Data read address principle

3.3.3 存储方案选择及流水线（Pipeline）设计

数据读取的地址生成方案基本确定之后，之后的问题就是考虑 Lenet-5 卷积神经网络每一层的中间数据的存储问题。因为中间数据的特征图数据量比较大，不可能选择寄存器变量进行存储，而 DDR 存储器的读写过于繁琐且读写所需时间长，Block RAM 具有读写速度快、调用方便等优势，而且存储空间满足本课题的要求，所以本课题选择 FPGA 芯片内部的 Block RAM（块存储器）作为中间数据的存储器。

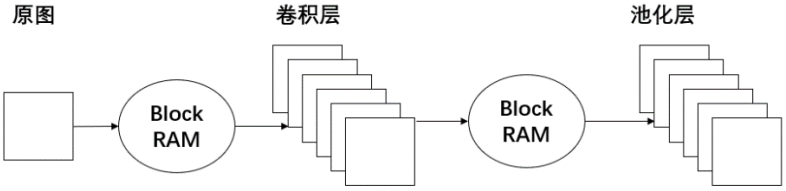


图 3-10 Lenet-5 卷积神经网络流水线设计
Fig.3-10 Pipeline design of Lenet-5 convolutional neural Network

如图 3-10 所示，方形图案代表特征图，椭圆形图案代表 Block RAM。本课题拟在每两个计算层之间都加入 Block RAM，用于存储图像中间数据。图像中间数据的读取依照上述规律进行读取，数据的写入按照正常顺序即可，每进行一次卷积或池化计算就把得到的计算结果存储到 Block ram 中。

从结构上来看，这种不同计算层之间加入寄存器的设计类似于芯片设计中的流水线设计。与芯片设计中的流水线设计一样，这也是“面积换性能”的情况，即用存储空间换取处理速度。简单来说就是在某两个功能单元之间加入了一个寄存器，如图 3-10 所示是本课题的卷积神经网络流水线设计图。当第三层的卷积模块在计算第一张原始图像产生的 6 张特征图时，此时池化模块 p2 并非停滞状态，而是在处理第二张原始图像产生的 6 张特征图。同样，此时卷积模块 c1 在处理第三张原始图像产生的特征图，就像生产线的“流水线”一般运作。大概的流程类似手机装机的时候，流水线上的工人对未成形的器件进行加工。所以在系统运行的每一时刻，在 Lenet-5 卷积神经网络的每一层都有一个原始图像处理的“半成品”，这些“半成品”对应不同时刻产生的原始图像。这样能大大提高数据处理的速度。

层间 Block RAM 的加入既使得图像中间数据得到了保存，又使得整个系统实现了流水线设计，大大提高了图片连续处理的运行速度。在芯片设计的时候，也经常会采取在功能模块间加入寄存器来提高运行速度的方法。这样虽然会用到很多的寄存器资源，但能使芯片得到很大的性能提升，即“面积换性能”。这

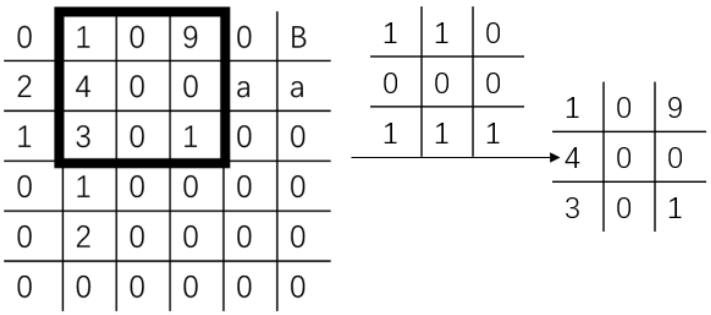
大大提高了系统处理的速度。

3.3.4 层间通信及计算协同

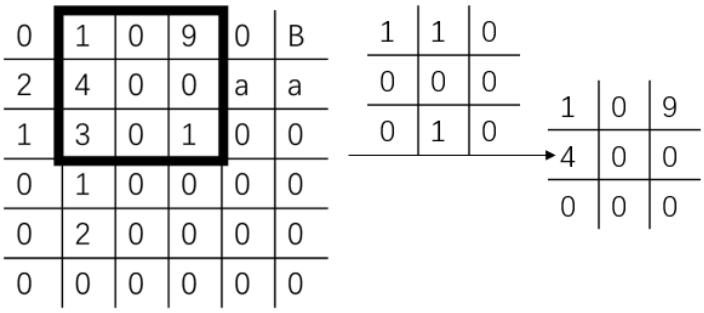
Lenet-5 卷积神经网络是一个由五层计算模块组成的卷积神经网络，在本文的第二章相关小节以及做了介绍。

FPGA 的特点是 FPGA 数据处理跟时钟的时序息息相关，这一点与 CPU 处理器大不相同。FPGA 会在一个时钟上升沿到来之后同时执行所有的代码，而 CPU 是代码顺序执行代码。如果在一个时间点，计算所需要的信号未得到相应的状态，也就是处于高阻态或者复位状态，这个时候就是代码的时序出现问题了，需要修改代码里有关信号间时序关系的部分。所以如果卷积模块在数据还未到来或者一个卷积核对应的数据未读取完的时候进行计算，将使整个网络计算结果出现错误，如图 3-11 所示，图 3-11（a）的卷积计算的扫描窗口内的图像数据读取完全，计算得到正确结果“5”；而如图 3-11（b）所示，卷积计算的扫描窗口内的图像数据未完全读取，计算得到错误结果“1”。

基于以上分析，Lenet-5 卷积神经网络要想实现正确推演并得到正确结果，必须要实现计算层之间的协同运作。为了实现 Lenet-5 卷积神经网络的不同层之间协同运作，本课题在第一、三、五层卷积计算模块、第二、四层池化模块以及第六层全连接模块中加入了层使能端口 en。数据读取地址生成、数据读取、数据存储、卷积计算、存储地址生成等操作必须在使能端口 en 的使能下才可以开始。使能信号的加入使得整个网络的运作更加协调，在进行第一层卷积计算之后，才能开始 Block RAM 数据存入操作并生成相应地址。并且地址的生成要在合适的时间点，否则就可能出现数据的误存储或者数据的重复存储从而覆盖掉之前的数据等情况。下面我将会详细阐述 Lenet-5 卷积神经网络的不同层之间协同运作的过程。



(a) 卷积计算正确的情况



(b) 卷积计算错误的情况

图 3-11 卷积计算错误情况与正确情况对比

Fig.3-11 The convolution calculation error case is compared with the correct case

当第一层卷积计算结束之后，此时所有卷积计算得到的数据都存入了 Block RAM 中。这个时候需要提醒下一层的池化计算开始，由于池化计算产生数据的同时需要产生数据存储地址并将数据存入 Block RAM，所以需要同时产生 Block RAM 数据写入端的使能信号 ena 和数据读写状态信号 wea。而卷积模块下层的 Block RAM 存储模块以及再下一层的池化模块都受到卷积模块产生的存储使能 ena 和读取使能 enb 的控制。通过仿真实验发现，Block RAM 的读取会延迟两个时钟，也就是在给到 Block RAM 读取地址的两个时钟之后才可以获得图像数据。所以 Block RAM 的读取以及读取计数要在给出地址两个时钟之后才能开始。这样就完成了一次层间通信，由卷积模块向存储模块 Block RAM 和池化模块发送使能信号。同样，当池化模块计算结束之后，此时池化计算的结果已经全部存入 Block RAM 中，这时需要提醒下层的卷积计算模块进行卷积计算并将数据存入 Block RAM 中。所以池化模块会产生卷积使能信号 conve 和 Block RAM 数据写入端的使能信号 ena 和数据读写状态信号 wea。卷积神经网络的层间通信实现方案的框图如图 3-12 所示。

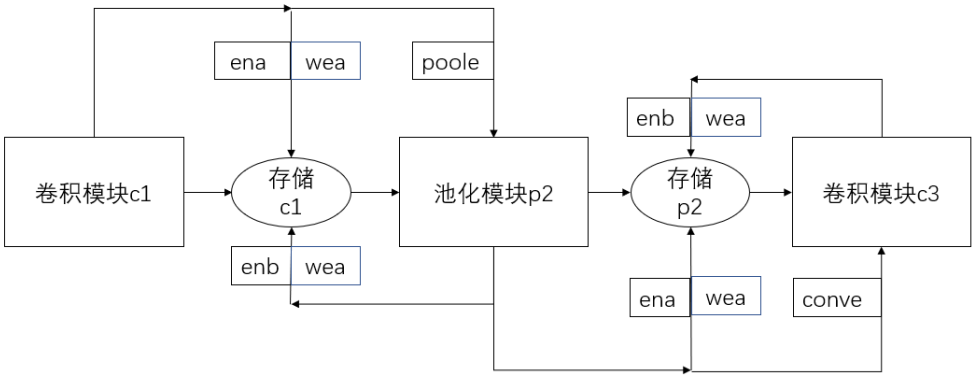


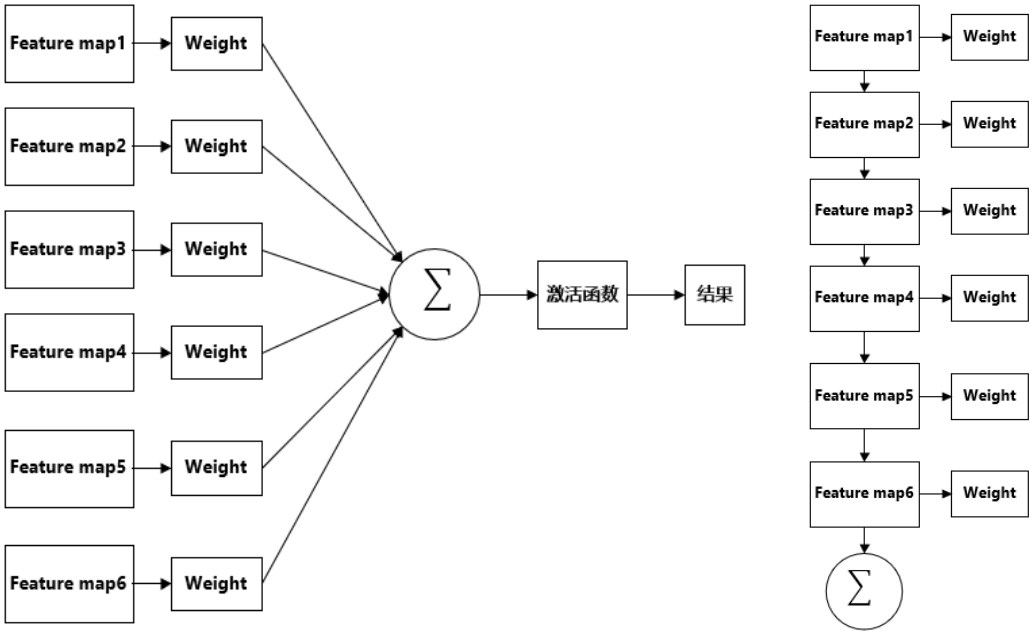
图 3-12 卷积神经网络的层间通信

Fig.3-12 Interlayer communication of convolutional neural networks

通过上述设计就可以实现 Lenet-5 卷积神经网络网络不同层之间的通信，从而使得 Lenet-5 卷积神经网络的字符识别推演过程变得井然有序。使系统在保证最高实时性的同时，避免出现数据误存储、误读取等情况。

3.3.5 计算模块实现

在介绍计算模块的实现之前，先介绍以下 FPGA 进行多线程计算的原理。如图 3-13 所示为 FPGA 进行多线程计算的原理图。从图中可以明显看出，FPGA 在进行卷积计算时，会在同一时间得到所有的卷积计算结果；而在处理器端进行卷积计算时，要依次对各个图片进行卷积计算，这个计算时间会成倍地多于 FPGA 端。当要处理的卷积计算过程比较多时，FPGA 并行计算的优势就会大大突显出来。



(a) 卷积计算加速器设计　　(b) CPU 计算顺序

图 3-13 卷积计算模式对比

Fig.3-13 Convolution computation parallelizes computation

图 3-13（a）为 FPGA 平台上卷积计算加速器设计，从图中可以明显看出，所有卷积计算通道都是同时进行计算的，经过一定的时间之后，可以在输出端口同时得到所有的卷积计算的结果；图 3-13（b）为 CPU 的计算顺序，卷积计算是顺序进行的，先计算完一张图像的卷积计算，然后进行下一张图像的卷积计算，以此类推。不考虑算力问题，图 3-13（a）的加速器架构的计算速度要远远快于图 3-13（b）。这个计算速度与网络计算的卷积计算个数有关，卷积计算

个数越多，并行计算的优势也就越明显，速度提升幅度也就越大。而且由于FPGA 平台的计算是在时钟信号的时序下进行的，理论上当时钟周期越短时，FPGA 平台的算法运行速度越快。

（1）卷积计算（Convolution）模块

卷积计算的第一步是读取数据。由卷积模块给出数据读取端使能信号 enb 以及读写状态信号 wea，在这之后生成数据读取地址，读取地址的生成必须要在 enb 信号使能和 wea 信号变为读状态之后。地址给到 Block RAM 后，要经过两个时钟的延迟后才能在数据输出端得到相应存储地址的数据。因为每一层计算之前都要先读取数据，所以在每两个层之间都有一个 Block RAM 进行数据存储和数据读取。数据的对应地址如图 3-14 所示。

卷积计算的第二步是“滑窗”确定和计算。上述的数据读取的过程要经过25次才可以确定一个“滑窗”，在数据读取地址生成时，就要遵循“读满一列再读一行”的顺序进行数据读取，即先生成地址 0，之后依次生成地址 10、20、30、40。这里需要一个计数器去进行计数，从 1 数到 25。当数到 25 时，计数器 cnt 跳回 1，同时进行卷积计算，即求加权平均值。而由第一步所述，地址给到 Block RAM 后，要经过两个时钟的延迟后才能在数据输出端得到相应存储地址的数据。由于这两个时钟的延迟，卷积计算的时间也要延迟两个时钟。这个时候计数器 cnt 计数到 2，所以每当 cnt 计数到 2 时进行加权平均计算。数值计算会在一个时钟周期内结束，这时把得到的数据给到数据输出端，同时产生数据写入地址。也就是说，数据写入地址计数 25 次改变一次。同时，保持下层的 Block RAM 的数据写入端使能信号 ena 拉高，wea 信号为写入状态。这样，每隔 25 个时钟周期存入加权平均后的数据。

卷积计算的第三步就是“滑窗”移动的操作。卷积计算的“滑窗”移动步长为 1，当计算完第一个“滑窗”内加权平均的数值之后，cnt 从 2 开始计数，数据读取的首地址为 1，依照上述第二步的过程进行加权平均值计算。由于第一个卷积计算结束之后并不是从第六个数开始卷积，而是从第二个数进行卷积，同理第二行卷积计算时也需要计数。所以在模块设计时采用多线计数的方式进行卷积计算。cnt 负责卷积邻域计数，是否已经读取完 25 个像素数据；row 负责卷积计算列计数，标示是否已经卷积完一行；line 负责卷积计算行计数，标示是否已经卷积完一列。这些计数器的状态标示着当前计算的进程，拿第一层卷积层举例，cnt 是卷积邻域计数器，它计数 25 次，当计数到 25 时，标志着一个卷积计算的邻域读取完成，可以开始卷积计算；row 是特征图的列计数器，它计数 28 次（因为第一层的图片大小为 32*32），当 row 计数到 28 时，标志着某一行的卷积计算已经结束，需要换到下一行进行卷积计算。

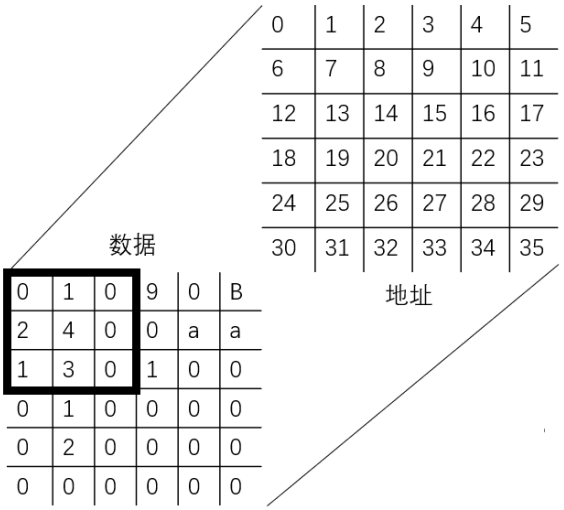


图 3-14 卷积计算地址生成规律

Fig.3-14 Rule of address generation in convolution computation

（2）池化计算（Pooling）模块

池化，也就是降采样，伴随卷积神经网络产生。池化层的加入是为了降低计算量。因为加入一个池化层之后，原本需要处理的数据变成了原来的四分之一，这样大大降低了计算量。池化主要有 Average Pooling 和 Max Pooling 两种。一开始平均池化的使用更常见，然后通过不断的实验，逐渐地淘汰了平均池化，选择效果更好地最大值池化。池化算法的实现原理也很简单。池化算法就是通过一个大小为 2*2 的滑窗对图像进行扫描，滑窗内四个数据选择最大值输出，这就是最大值池化。即滑窗选择四个像素点，求出一个像素点，实现四比一的映射，池化计算的过程图如图 3-15 所示。

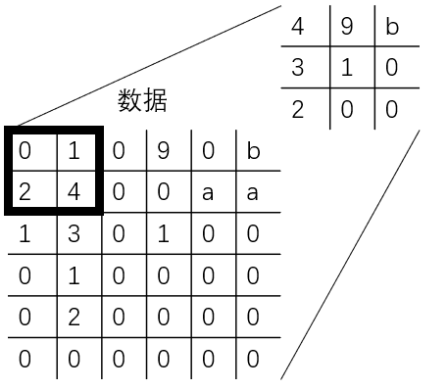


图 3-15 池化计算

Fig.3-15 pooling

池化计算与卷积计算不同。池化计算在计算完第一个池化之后会从第三个数开始读入数据，即步长为 2。所以在数据读取时要同样计算 cnt、row 和 line，row 和 line 计数时每次加二。

池化模块接收 enb 使能信号之后开始数据读取。数据读取完之后进行有符号数大小比较，选择最大的数作为结果，存储到下一层的 Block RAM 中。

3.4 本章小结

本章主要针对字符识别系统的实现进行了 FPGA 平台上的方案设计。

首先，根据工业相机的视频协议对图像数据进行解码。解码部分主要分为两部分——低压差分信号解串和 Camera link 协议解码。

之后，根据 XC7K325tffg676-2FPGA 平台的硬件特性，对 Lenet-5 卷积神经网络在 FPGA 平台的部署进行了总体的方案分析。在方案选择上，分别对 DSP+FPGA 架构的硬件加速设计、Vivado HLS 把 C#语言翻译成 HDL 语言简化开发过程以及 HDL 编码实现 Lenet-5 卷积神经网络进行了优缺点分析，最终确定了 HDL 编码实现 Lenet-5 卷积神经网络的课题方向。之后分别对卷积计算的图像数据读取地址生成规律、FPGA 开发板存储方案选择、流水线设计、层间通信以及计算模块实现进行了可行性分析，大致上确定了字符识别模块的实现方案。

第 4 章 字符识别的图像预处理算法实现

4.1 引言

本章节主要研究字符识别系统的图像预处理算法部分的实现。在这一部分作者将图像预处理模块分成几个简单的子模块。即图像预处理要经过图像增强、图像滤波、图像二值化以及字符分割几个子模块。如图4-1所示为图像预处理阶段地简图。

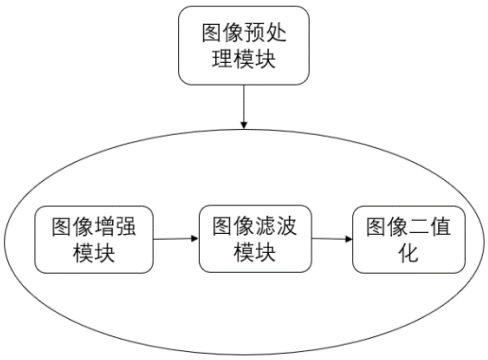


图 4-1 图像预处理模块结构
Fig.4-1 Image preprocessing module structure

作者分别选取了正常光照下、强光照下以及弱光照下的字符例图作为图像预处理对照。对不同光照条件下的图像预处理流程进行了深入的研究和实验验证。

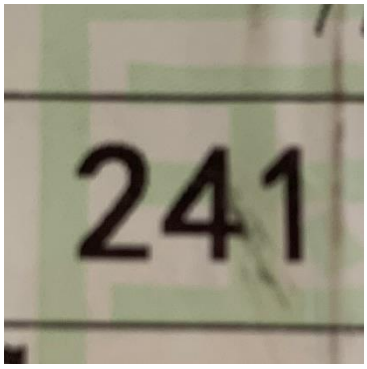
4.2 正常光照下图像预处理实验

作者在正常光照条件下拍摄了如图 4-2（a）所示的原始图像。此时图片中的字符为“程永龙”三个字，也就是本课题的检测目标，背景是校园卡上图书馆背景图。从原始图像来看，目标字符和图像背景的差距很大，这种情况下的字符分割会比较简单。

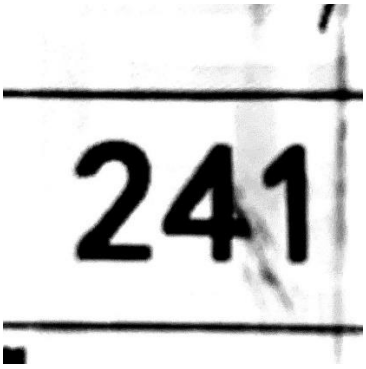
然后以原始图像为输入，分别对图像进行图像增强、图像 sobel 边缘检测以及图像二值化等操作，结果如图 4-2（b）、（c）、（d）所示。

图 4-2（b）是对原始图像进行图像增强的结果。从图中可以看出，原本标签上的污渍和背景变得更加地清晰了。所以如果对原始图像进行图像增强操作，会使背景噪声放大，不利于字符识别。图 4-2（c）是对原始图像进行 sobel 边缘检测的结果图。从图中可以看出，目标字符“241”仅仅保留了一些上边缘和左

侧边缘，这使得字符信息过分丢失。所以对原始图像进行 sobel 边缘检测操作会损失大量信息。图 4-2（d）为对原始图像进行图像二值化后的结果图。从图中可以看出，图像附近并无明显噪点，且边缘清晰，信息完整，是字符识别系统所需的理想结果。图 4-2（e）为对原始图像进行 gaussian 滤波后的结果图，并无明显效果。图 4-2（f）为对原始图像进行腐蚀膨胀之后的结果图，可以看出字符边缘出现残影。

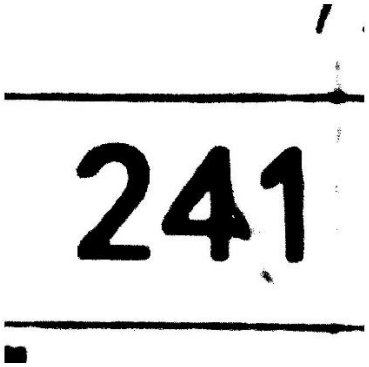


(a) 原始图像

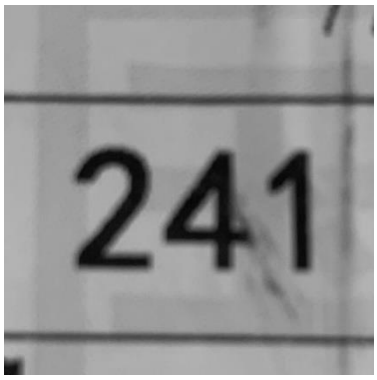


(b) 图像增强后的图像

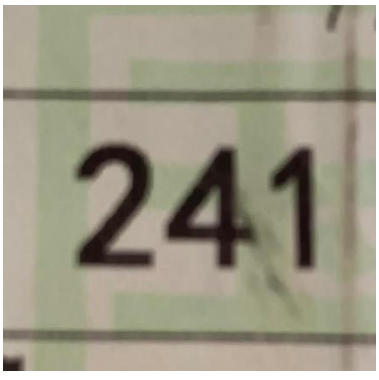
(c) 边缘检测后的图像



(d) 二值化后的图像



(e) gaussian 滤波后的图像



(f) 腐蚀膨胀后的图像

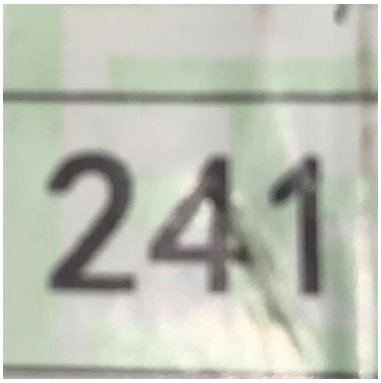
图 4-2 正常光照下图像预处理结果

Fig.4-2 Image preprocessing results under normal illumination

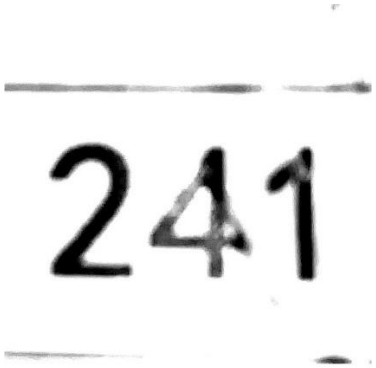
综上所述，在正常光照条件下，由于字符与图像背景区别比较明显，适合对图像进行二值化处理，这样就可以得到比较理想的用于字符特征提取的图像结果。

4.3 强光照下图像预处理实验

在强光照条件下，作者拍摄了如图 4-3（a）所示的图像预处理例图。该图像与正常光照下的图片一样，只是增加了光源影响。

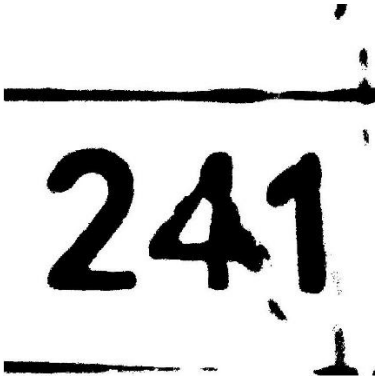


(a) 原始图像

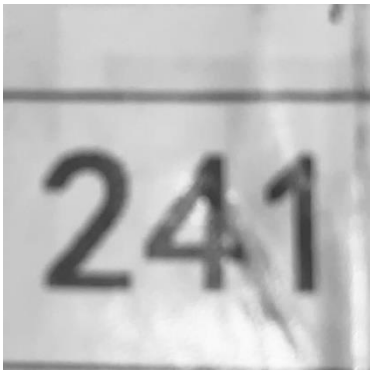


(b) 图像增强后的图像

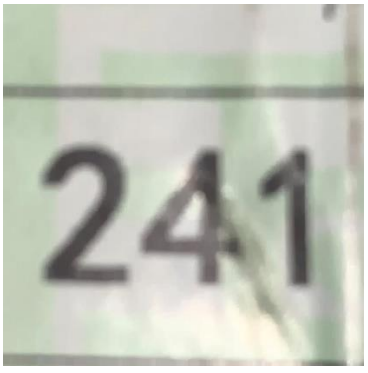
(c) 边缘检测后的图像



(d) 二值化后的图像



(e) 高斯滤波后的图像



(f) 腐蚀膨胀后的图像

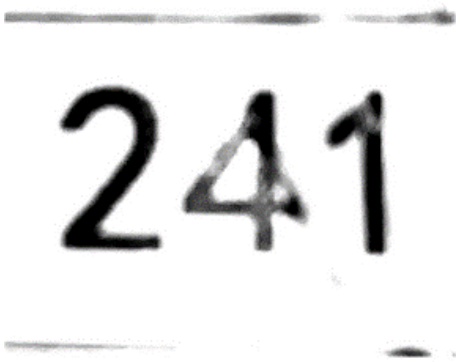
图 4-3 强光照下图像预处理结果

Fig.4-3 Results of image preprocessing under strong light

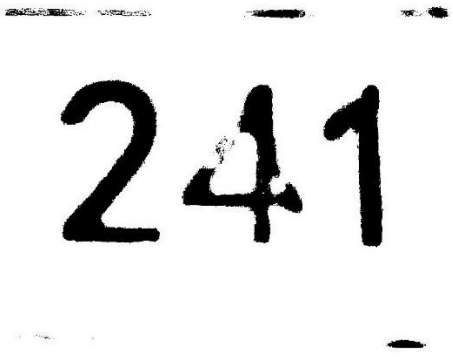
可以看出在原始图像的下方有一块高亮区域，这是光源使得标签的反射光增强的结果。而且同 4-2（a）对比，可以发现在强光照条件下，相机拍摄的字符变得模糊。

图 4-3（b）为对原始图像进行图像增强的结果图。从图中可以看出，图像左上方有一些背景噪声，而且由于原始图像下方的强光照的影响，字符“4”的边界信息变得模糊，而其他字符清晰可见，考虑在图 4-3（b）的基础之上进行图像增强处理，然后根据实际实验结果确定图像处理步骤。图 4-3（c）为对原始图像进行 Sobel 边缘检测后的结果图。从图中可以看出，图像边缘由于强光照影响损失更加严重。图 4-3（d）为对原始图像进行二值化后的图像。从图中可以明显看出，图像下半部分显示清晰，但图像上半部分被黑色阴影遮住了。这是因为光源在图片下半部分，这显得图像上半部分比较暗。因此图像二值化的阈值降低，把图像上半部分大部分像素数据判定为黑色。图 4-3（e）和（f）分别为对图像进行高斯滤波和腐蚀膨胀操作后的结果图。从图中可以明显看出，两种算法并没有起到明显的图像滤波的效果。

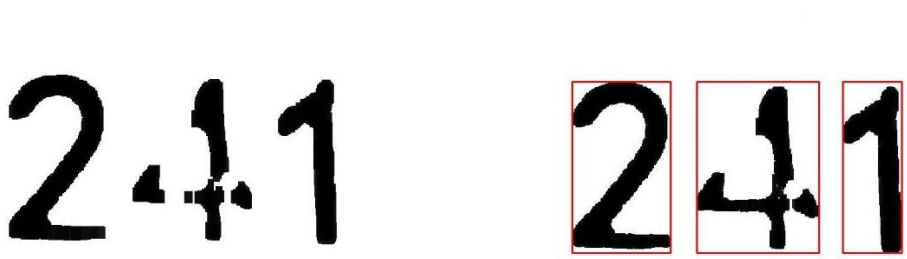
在上述实验的基础上，进行下一步的图像处理。在强光照影响下，图像增强算法的结果图中的某些字符的笔画变细。因为图像增强算法效果最为显著，字符的边缘信息得到提取，所以下一步的实验在图像增强后的结果图的基础上展开。



(a) 图像增强后的图像



(b) 二值化后的图像



(c) 膨胀腐蚀后的图像　　(d) 字符分割后的图像

图 4-4 图像增强基础上的图像预处理结果

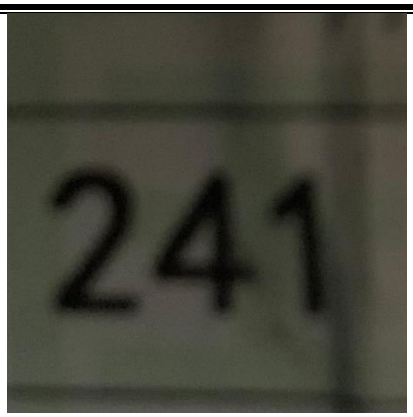
Fig.4-4 Image preprocessing results based on image enhancement

图 4-4 中的（b）、（c）、（d）都是在前一张图片的基础上进行相应操作得到的结果图。因为在该图 4-4（a）的左上方有背景噪声，而且字符“4”的笔画变细，考虑使用图像二值化滤除噪声，加深字符笔画。以图像增强后的结果图为输入，对图像增强后的图像二值化，得到图像增强之后的结果图如图 4-4（b）所示。从图中可以明显看出，原本图像的背景噪声得到了滤除，而字符的信息变得更加明显。也有部分字符信息不完整，“4”字的部分笔画出现明显断裂。在此基础之上，考虑通过先膨胀后腐蚀的闭操作完成笔画闭合，得到的膨胀腐蚀操作之后的结果如图 4-4（c）所示。从图中可以明显看出，“4”字的笔画基本补全，而“4”字的左上方的笔画的断裂因间隔距离过大，难以补全。如果要达到补全“4”字笔画的效果，膨胀算法的效果会使得“2”和“4”字完全变成一片黑色，这属于膨胀算法参数过大导致的过膨胀。最后，对腐蚀膨胀后的结果图进行字符分割操作，得到的经过算法处理的结果图如图 4-4（d）所示。这里用到的字符分割算法是投影法。从图中可以明显看出，字符分割的效果非常显著。图中没有因为图像噪点产生的一些小的字符框，这说明前期的图像预处理算法达到了字符分割所需的要求。

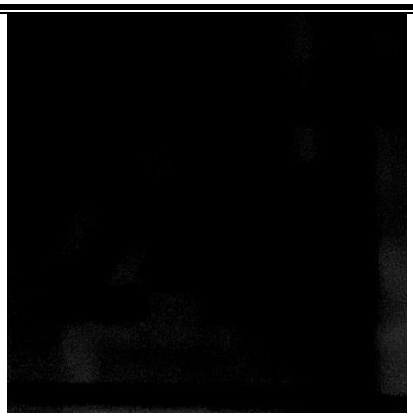
综上所述，在处理强光照下的图像时，可以按照图像增强、图像二值化、膨胀腐蚀、字符分割的顺序进行图像预处理操作。

4.4 弱光照下图像预处理实验

在弱光照条件下，相机拍摄到的图像整体偏暗，目标字符与背景噪声相差不大，所以不好区分。实验结果如图 4-5 所示。



(a) 原始图像



(b) 图像增强后的图像

(c) 边缘检测后的图像

(d) 图像二值化后的图像

图 4-5 弱光照下图像预处理结果

Fig.4-5 Results of image preprocessing under low light

从图4-5中可以看出，通过图像二值化算法可以很好的提取目标特征。所以在该系统实际应用时，可以通过图像二值化之后进行滤波的算法提取字符特征，或者配备光源。

4.5 本章小结

在正常光照条件下，目标字符和图像背景的差距很大，这种情况下的字符分割会比较简单。由于字符与图像背景区别比较明显，适合对图像进行二值化处理，这样就可以得到比较符合字符特征提取的图像。

在强光照条件下，对原始图像进行整体灰度增强之后，图像会有一些背景噪声，字符的边界信息也会变得模糊，但处理之后的图像大幅度保证了字符信息的完整性，在此基础之上进行图像预处理操作。可以按照图像增强、图像二值化、膨胀腐蚀、字符分割的顺序进行图像预处理操作。而图像二值化可以滤除背景噪声，加深字符笔画，适合在图像增强之后执行。由于经过以上操作处理的字符的部分笔画有细微的断裂，所以考虑选用先膨胀后腐蚀的闭操作完成笔画补全。最后用投影法对图像进行字符分割，取得了显著的效果。

第 5 章 FPGA 平台实现 Lenet-5 卷积神经网络

5.1 引言

FPGA 芯片就像是一个“积木”，它可以通过 Verilog HDL 编程设计调用不同的组合逻辑资源和寄存器资源，生成不同的布局布线方案，从而产生不同的功能设计。Verilog 语言的表述方式与 C 语言类似，但所描述的内容大不相同。Verilog 语言描述的是一个由组合逻辑电路和寄存器电路组成的硬件电路图，所有语句都是跟随时钟并行执行的。而 C 语言使用的是高层次的语言，语句按顺序执行。FPGA 项目开发的流程主要包括项目设计、仿真、综合布线以及板级验证等步骤。

本章节主要研究 Lenet-5 算法在 FPGA 上的实现，按技术难点划分，主要包括卷积计算地址读取、卷积有符号计算、流水线设计、最大池化以及全连接防溢出。

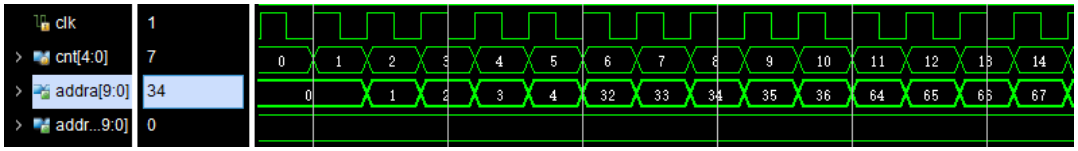
FPGA 项目开发的流程主要包括项目设计、仿真、综合布线以及板级验证。首先，要确定项目的实现功能以及所用的方案。比如本课题要实现字符识别任务，选择使用图像预处理算法对原始图像进行初步处理，然后通过训练过的卷积神经网络实现字符识别。第二，要对项目实现功能进行细分，并思考用哪种方法实现这些功能。比如本课题要在 FPGA 平台实现 Lenet-5 卷积神经网络的硬件加速器设计[27]-[48]，就要考虑在模块计算时如果提高运行速度，在模块间组合运算时如何提高计算速度，然后在顶层进行模块例化实现网络搭建。第三，对每个模块进行设计和仿真。分析一个模块需要实现的功能和给出的接口，然后进行代码实现，根据仿真情况不断调试代码。比如要实现卷积运算模块，就要实现中间层特征图的存储，读取特征图时的地址生成问题，输出特征图的地址生成问题，并要考虑 Block RAM 的读写延迟问题，避免数据写错地址。

本文在该方向的研究内容主要包括卷积模块读取地址生成、层间通信及流水线设计、最大池化以及全连接防溢出等。卷积模块读取地址生成要根据卷积计算的原理进行读取，卷积计算相当于一个滑窗操作，在一个大小为 5*5 的滑窗内求加权平均值。而滑窗内数值对应的地址是不连续的，所以在生成读取地址时，要按照“读满一列再读一行”的顺序生成地址。而写入地址的生成要与数据产生的时序相匹配，保证生成一个地址的同时，有一个数据写入。最重要的是，要保证在第一个数据产生时，要把地址 0 给到第一个数据，这样存入的

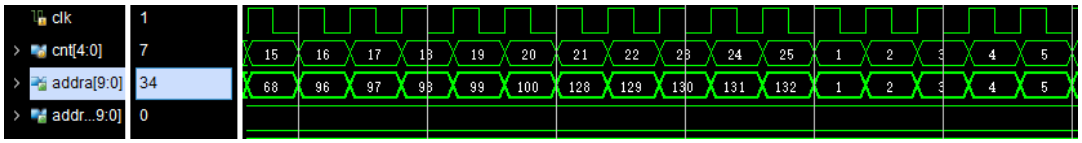
数据在读取时才不会发生错误。这个过程通过波形仿真对代码进行不断地修改，不断地调试，使得地址生成顺序符合项目要求。

5.2 读写地址生成方案——首地址锚定法

首地址锚定法，从字面意思来看，就是依靠首地址确定图像数据生成读取地址。实际上，该方法就是依靠确定首地址来生成后续的图像数据读取地址，首地址就像是“滑窗”每一行数据读取地址生成时的起始点，只要确定了起始点，后面同一行图像数据的读取地址也就确定了。



(a) 仿真波形前段



(b) 仿真波形后段

图 5-1 卷积计算首地址锚定法仿真波形

Fig.5-1 Convolution computation head address anchoring simulation waveform

在进行这一步实验时，作者首先对卷积模块的数据读取地址进行了调试，调试的最终结果如图 5-1 所示。从图中可以看出，代表数据读取地址的 addra 信号按照 0、1、2、3、4、32、33、34、35、36、64、65、66、67、68……128、129、130、131、132 的顺序生成，而当 addra 信号生成第一个卷积邻域内的最后一个数据的读取地址 132 后，卷积滑窗前移一步，这时代表图像数据读取地址的 addra 信号从地址 1 开始，按照 1、2、3、4、5、33、34、35、36、37、65、66、67、68、69……129、130、131、132、133 的顺序生成，这与我们之前方案设计时的预想是一样的。第一行的图像数据的首地址为 0，而由于图像的行数和列数都是 32，所以第二行的图像数据的首地址为 32。

首地址锚定法的提出是为了解决卷积或者池化计算时图像数据读取地址不连续的问题。它可以使卷积或池化模块按照图 5-1 所示的 addra 信号的规律生成图像数据读取地址。而 addra 信号的首地址锚定是根据计数器信号 cnt 来确定的。cnt 是用来进行卷积邻域内数据计数的。因为卷积邻域大小为 5*5，所以 cnt 从 1 计数到 25。从图 5-1 中可以看出，当 cnt 计数到 1 时，确定第一行卷积邻域内数据的首地址，之后的四个数据读取地址在每次 cnt 加一时加一；当 cnt 计数到 6 时，确定第二行卷积邻域内数据的首地址，之后的四个数据读取地址在每次 cnt

加一时加一；当 cnt 计数到 11 时，确定第三行卷积邻域内数据的首地址，之后的四个数据读取地址在每次 cnt 加一时加一；当 cnt 计数到 16 时，确定第四行卷积邻域内数据的首地址，之后的四个数据读取地址在每次 cnt 加一时加一；当 cnt 计数到 21 时，确定第五行卷积邻域内数据的首地址，之后的四个数据读取地址在每次 cnt 加一时加一。图 5-1 所示的图像数据读取地址的生成规律通过本文提出的首地址锚定法实现。

5.3 流水线（Pipeline）设计

通过上述设计就可以实现 Lenet-5 卷积神经网络网络的卷积和池化模块的图像数据读取地址的生成，读取地址生成之后，此时数据读取端使能信号 ena 和读写状态信号 wea 都被拉高，表示此时 Block RAM 进入读数据状态。数据进入到卷积或池化模块后，会进行相应的计算，得到相应的特征图。在第三章的 3.4 节分析了网络相邻层之间加入 Block RAM 存储器的必要性和可行性。

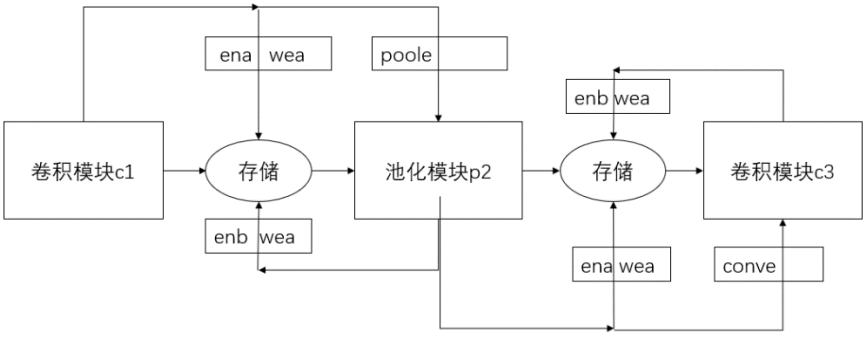


图 5-2 卷积神经网络的流水线设计
Fig.5-2 Pipeline design of convolutional neural network

从结构上来看，相邻层间加入 Block RAM 的 Lenet-5 卷积神经网络网络的网络架构是一个流水线设计。跟芯片设计中的流水线设计类似，这也是“面积换性能”，即用存储空间换取处理速度。简单来说就是在某两个功能单元之间加入了一个寄存器。如图 5-2 所示是本课题的流水线的简图。当卷积模块 c3 在计算第一张原始图像产生的特征图时，此时池化模块 p2 并非停滞状态，而是在处理第二张原始图像产生的特征图。同样，此时卷积模块 c1 在处理第三张原始图像产生的特征图，就像生产线的 “流水线”一般运作。大概的流程类似手机装机的时候，流水线上的工人对未成形的器件进行加工。这样能大大提高数据处理的速度。当卷积计算模块在运行时，下层的池化模块在计算上一张图片，这可以使整个网络的运行速度提高 2.5 倍。而这主要得力于层间 Block RAM 的加入。在芯片设计的时候，也经常会采取在功能模块间加入寄存器来提高运行速度的方法。这样会用到很多的寄存器，但会得到很大的性能提升，即“面积换

性能”。这大大提高了系统处理的速度。

图 5-3 流水线设计仿真结果

Fig.5-3 Simulation results of pipeline design

图 5-3 所示为流水线设计在 FPGA 平台上的仿真波形。dina_c1_x 为第一层 Block RAM 的数据输入端口，dout_c1_x 为第一层 Block RAM 的数据输入端口，可以明显看出数据的读出相比数据的写入要快得多。这是因为在第一层卷积计算的数据产生时，每得到一个结果就要读取 25 个卷积邻域数据，所以理论上数据的写入时间是数据读取时间的 25 倍以上。从图中可以看出，从 0us 开始，卷积模块 c3 进行读取数据、卷积计算，可以明显看出网络一直源源不断地向 Block RAM 中存入数据，到 390us，数据存入完毕，下一层的数据读取开始。从 390us 到 410us，池化模块 p2 进行读取数据、池化计算以及数据存入。从 410us 到 460us，卷积模块 c3 进行读取数据、卷积计算以及数据存入。每个模块进行工作时，其他模块都是处于停滞状态。而当此时系统连续输入两张图片时，网络中的不同模块就会同时处理两张图片，这大大提高了系统运行速度。

5.4 层间通信及计算协同

Lenet-5 卷积神经网络是一个由五层计算模块组成的卷积神经网络，详细描述在第四章。FPGA 的特点是，FPGA 数据处理跟时钟的时序息息相关，这一点

与 CPU 处理器大不相同。FPGA 会在一个时钟上升沿到来之后同时执行所有的代码，而 CPU 是代码顺序执行。如果在一个时间点，计算所需要的信号未得到相应的状态，也就是处于高阻态或者复位状态，这个时候就是代码的时序出现问题了，需要修改代码里有关信号间时序关系的部分。所以如果卷积模块在数据还未到来或者一个卷积核对应的数据未读取完的时候进行计算，将使整个网络计算结果出现错误。

为了实现 Lenet-5 卷积神经网络的不同计算层之间的协同运作，本课题在第一、三、五层卷积计算模块、第二、四层池化模块以及第六层全连接模块中加入了层使能端口 en。数据读取地址生成、数据读取、数据存储、卷积计算、存储地址生成等操作必须在使能端口 en 的使能下才可以开始。使能信号的加入使得整个网络的运作更加协调，在进行第一层卷积计算之后，才能开始 Block RAM 数据存入操作并生成相应地址。并且地址的生成要在合适的时间点，否则就可能出现数据的误存储或者数据的重复存储从而覆盖掉之前的数据等情况。下面我将会详细阐述 Lenet-5 卷积神经网络的不同层之间协同运作的过程。

当第一层卷积计算结束之后，此时所有卷积计算得到的数据都存入了 Block RAM 中。这个时候需要提醒下一层的池化计算开始，由于池化计算产生数据的同时需要产生数据存储地址并将数据存入 Block RAM，所以需要同时产生 Block RAM 数据写入端的使能信号 ena 和数据读写状态信号 wea。这样就完成了一次层间通信，由卷积模块向存储模块 Block RAM 和池化模块发送使能信号。同样，当池化模块计算结束之后，此时池化计算的结果已经全部存入 Block RAM 中，这时需要提醒下层的卷积计算模块进行卷积计算并将数据存入 Block RAM 中。所以池化模块会产生卷积使能信号 conve 和 Block RAM 数据写入端的使能信号 ena 和数据读写状态信号 wea。

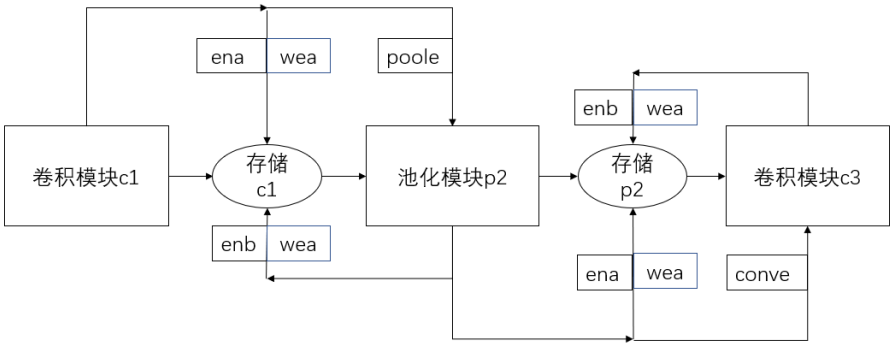


图 5-4 卷积神经网络的层间通信

Fig.5-4 Interlayer communication of convolutional neural networks

而卷积模块下层的 Block RAM 存储模块以及再下一层的池化模块都受到卷

积模块产生的存储使能 ena 和读取使能 enb 的控制。通过仿真实验发现，Block RAM 的读取会延迟两个时钟，也就是在给到 Block RAM 读取地址的两个时钟之后才可以获得图像数据。所以 Block RAM 的读取以及读取计数要在给出地址两个时钟之后才能开始。

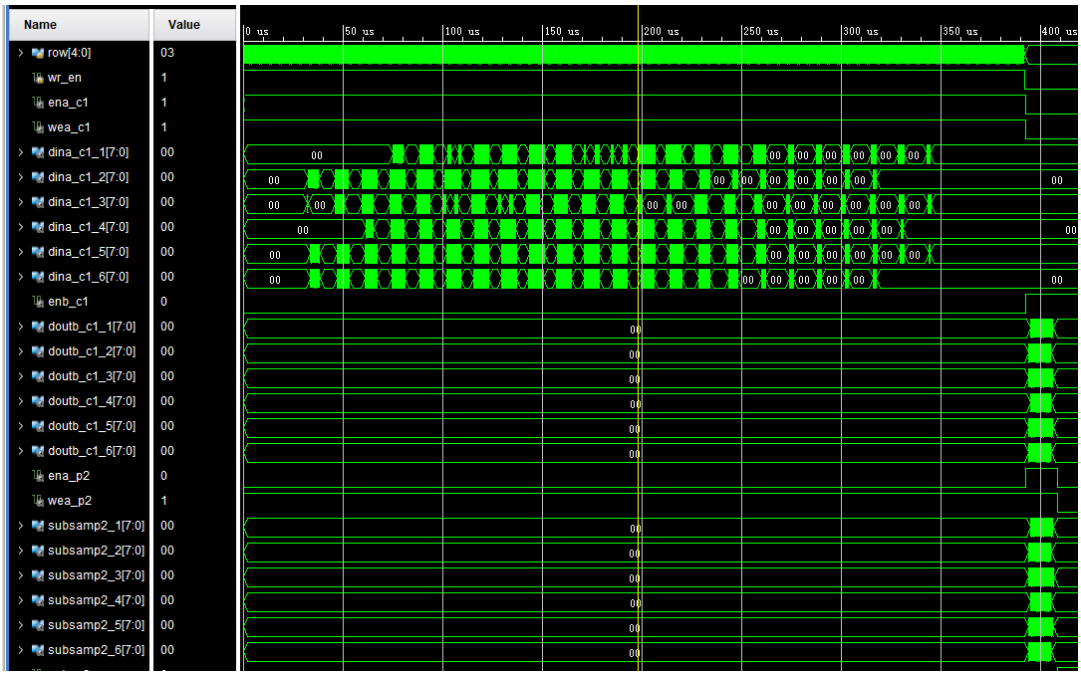


图 5-5 卷积神经网络的层间通信仿真结果

Fig.5-5 Simulation results of interlayer communication in convolutional neural network

如图 5-5 所示为 Lenet-5 卷积神经网络的层间通信的仿真结果，wea_c1 为图 5-4 中存储 c1 原始图像存储的 Block RAM 的读写状态信号，ena_c1 为图 5-4 中存储 c1 的数据写入端的使能信号，enb_c1 为图 5-4 中存储 c1 的数据读取端的使能信号，ena_p2 为图 5-4 中存储 p2 的数据写入端的使能信号，wea_p1 为图 5-4 中存储 p2 原始图像存储的 Block RAM 的读写状态信号。从图中可以明显看出，当 ena_c1 信号被拉高时，存储 c1 的数据写入端处于工作状态，当 wea_c1 信号被拉高时，存储 c1 处于数据写入状态，只有当 ena_c1 信号和 wea_c1 信号同时被拉高时，存储 c1 的数据写入过程开始。从图中可以看出，在 ena_c1 信号和 wea_c1 信号同时被拉高时，数据 dina_c1_1 到 dina_c1_6 都出现波形，说明数据源源不断地存入存储 c1 中。

5.5 计算模块实现

5.5.1 卷积计算（Convolution）模块

卷积计算的第一步是读取数据。由卷积模块给出数据读取端使能信号 enb

以及读写状态信号 wea，在这之后生成数据读取地址，读取地址的生成必须要在 enb 信号使能和 wea 信号变为读状态之后。当卷积计算模块生成的图像数据读取地址出现在 Block RAM 的读地址端口之后，要经过两个时钟的延迟后才能在数据输出端得到相应存储地址的数据。因为每一层计算之前都要先读取数据，所以在每两个层之间都有一个 Block RAM 进行数据存储和数据读取。

卷积计算的第二步是“滑窗”确定和计算。上述的数据读取的过程要经过 25 次才可以确定一个“滑窗”，在数据读取地址生成时，就要遵循“读满一列再读一行”的顺序进行数据读取，即先生成地址 0，之后依次生成地址 10、20、30、40。这里需要一个计数器去进行计数，从 1 数到 25。当数到 25 时，计数器 cnt 跳回 1，同时进行卷积计算，即求加权平均值。而由第一步所述，地址给到 Block RAM 后，要经过两个时钟的延迟后才能在数据输出端得到相应存储地址的数据。由于这两个时钟的延迟，卷积计算的时间也要延迟两个时钟。这个时候计数器 cnt 计数到 2，所以每当 cnt 计数到 2 时进行加权平均计算。数值计算会在一个时钟周期内结束，这时把得到的数据给到数据输出端，同时产生数据写入地址。也就是说，数据写入地址计数 25 次改变一次。同时，保持下层的 Block RAM 的数据写入端使能信号 ena 拉高，wea 信号为写入状态。这样，每隔 25 个时钟周期存入加权平均后的数据。

卷积计算的第三步就是“滑窗”移动的操作。卷积计算的“滑窗”每经过一次计算之后挪动 1 步，当计算完第一个“滑窗”内加权平均的数值之后，cnt 从 2 开始计数，数据读取的首地址为 1，依照上述第二步的过程进行加权平均值计算。由于第一个卷积计算结束之后并不是从第六个数开始卷积，而是从第二个数进行卷积，同理第二行卷积计算时也需要计数。所以在模块设计时采用多线计数的方式进行卷积计算。cnt 负责卷积邻域计数，是否已经读取完 25 个像素数据；row 负责卷积计算列计数，标示是否已经卷积完一行；line 负责卷积计算行计数，标示是否已经卷积完一列。这些计数器的状态标示着当前计算的进程，拿第一层卷积层举例，cnt 是卷积邻域计数器，它计数 25 次，当计数到 25 时，标志着一个卷积计算的邻域读取完成，可以开始卷积计算；row 是特征图的列计数器，它计数 28 次（因为第一层的图片大小为 32*32），当 row 计数到 28 时，标志着某一行的卷积计算已经结束，需要换到下一行进行卷积计算。

如图 5-6 所示为卷积计算在 Xilinx Vivado 开发工具内的仿真波形，temp 是一个图像数据存储矩阵，conv0 到 conv4 是 dina_c1_1 的每一行的计算结果，dina_c1_1 等于 conv0 到 conv4 的加和。从图中可以看出，conv0 到 conv4 的加和为 11，而这个数据是经过 8 倍量化的，所以需要除以 8，得到结果为 1，所以 dina_c1_1 等于 1。

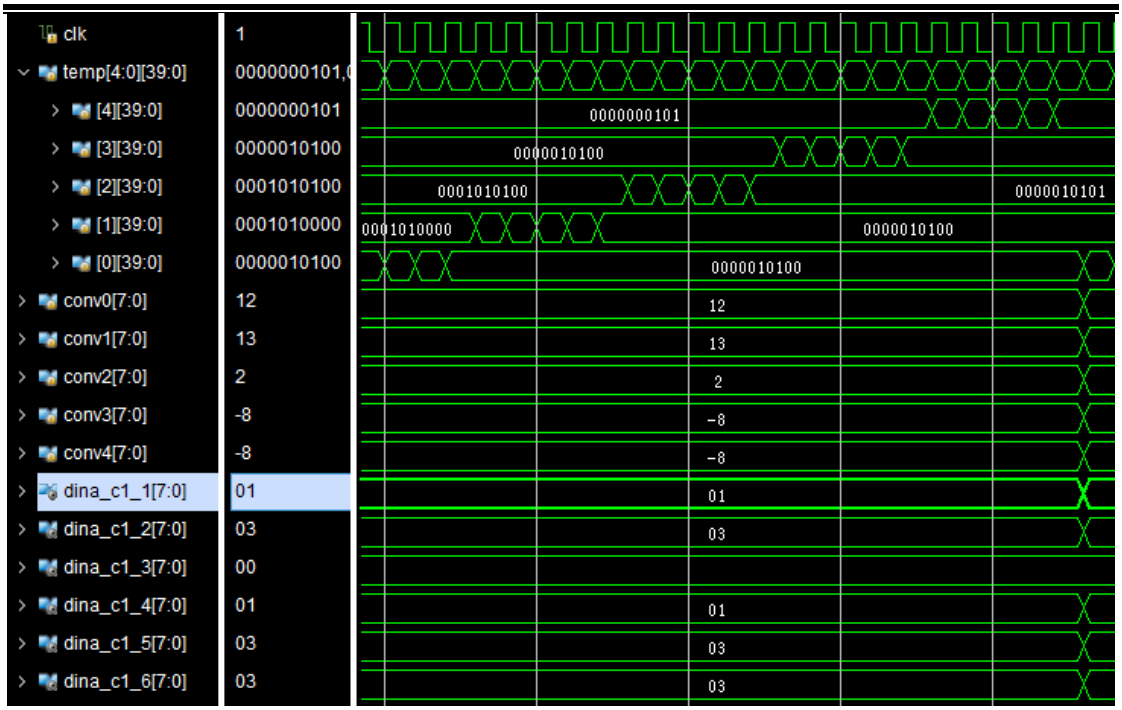


图 5-6 卷积计算仿真波形

Fig.5-6 Convolution computation simulation waveform

还有一个需要注意的点是有符号数计算的问题。由于权重量化都是 8 比特的有符号数，所以计算结果也是 8 比特有符号数，有时就会出现数值溢出的现象，导致计算错误，这里采取补码 8 比特有符号数转换为 9 比特有符号数的方式防止数值溢出。

5.5.2 池化计算（Pooling）模块

池化，也就是降采样，伴随卷积神经网络产生。在第一层卷积计算提取完原始图像的特征信息之后，这时会得到大量的特征图数据，为了降低计算量，保持原始的特征图中的目标特征的完整性，对特征图进行降采样处理。池化主要有邻域平均值降采样和最大值降采样两种。一开始邻域平均值降采样的使用更常见，后来在实践中邻域平均值降采样算法就被逐渐淘汰了，人们开始选用效果更加优异的最大值降采样算法进行数据降采样操作。池化算法的基本原理也很简单。池化算法就是通过一个大小为 2*2 的最大值滑窗对一张图片进行扫描，并将最大值滑窗内的所有数据的最大值作为输出结果。简单来说就是最大值滑窗选择四个数据，求出一个数据，实现四比一的映射，如图 5-7 所示。池化的作用是减小数据空间的大小，从而降低计算量，提高计算速度。

池化计算与卷积计算不同。池化计算在计算完第一个池化之后会从第三个数开始读入数据，即步长为 2。所以在数据读取时要同样计算 cnt、row 和 line，

row 和 line 计数时每次加二。

池化模块接收 enb 使能信号之后开始数据读取，数据读取完之后进行有符号数大小比较，选择最大的数作为结果，存储到下一层的 Block RAM 中。池化计算的仿真结果图如图 5-7 所示，doutb_c1_1 到 doutb_c1_6 就是读取得到的数据流，经过池化计算之后得到 subsamp2_1 到 subsamp2_6 的池化结果。从图中可以看出，doutb_c1_2 的前四个数据为 0、0、2、1，经过最大值池化计算之后得到的结果为 2，之后 doutb_c1_2 的四个数据为 1、3、3、1，经过最大值池化计算之后得到的结果为 3，后续的结果以此类推。

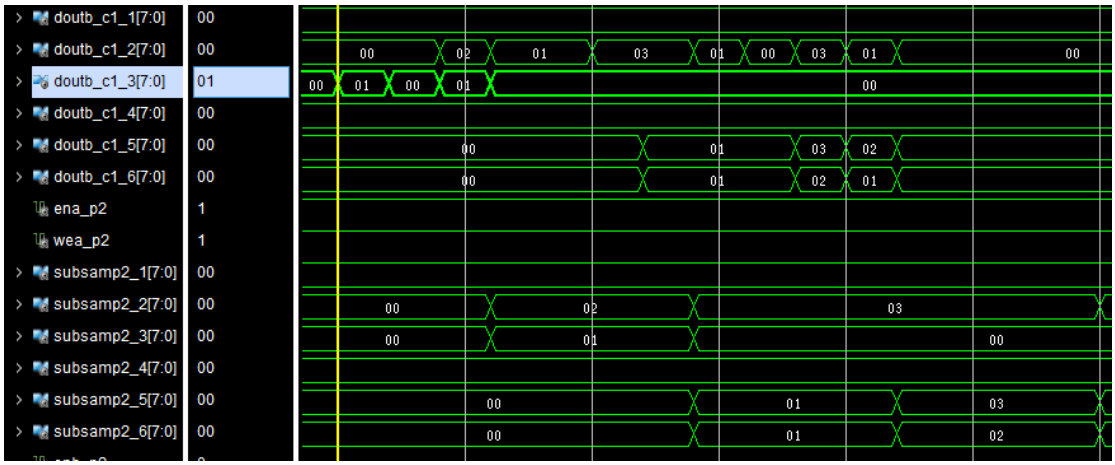


图 5-7 池化计算仿真波形

Fig.5-7 Pooling computation simulation waveform

5.6 结果展示

在对 Lenet-5 卷积神经网络进行网络训练之后，可以得到一组最合适的权重，使用 10000 个测试样本对这些权重组成的卷积神经网络的识别准确度进行测试。基本过程就是使用 10000 张带有标签的字符的不同字体的样本对 Lenet-5 卷积神经网络进行测试，然后把测试结果和样本标签进行比对，如果检测结果和样本标签一致，则通过测试，如果检测结果和样本标签不一致，则不通过测试。通过以上的过程可以大概测试网络的识别精度，样本越多，误差越小。

如图 5-8 所示为使用命令行对 Lenet-5 卷积神经网络进行精度测试的过程图，经过 10000 个样本的测试之后，可以在命令行看到在 10000 个样本中有 9717 个样本通过了测试。可以计算得知本课题所使用的 Lenet-5 卷积神经网络的识别精度为 97.17%。当然这是用 C 语言编程得到的测试结果，经过权重量化后的算法精度会有所降低。在 FPGA 平台上，本文对权重数据进行了 8 倍量化，每一层卷积计算结束之后，对计算结果进行量化还原，因为所有数据都经过 8 倍量化，数据相乘会使数据变成真实数据的 64 倍。这样提高了数据计算的准确性，弥补

了 FPGA 平台难以进行小数计算的缺陷，充分发挥其并行化计算以提高算法运行速度的优势。

图 5-8 Lenet-5 卷积神经网络精确度测试

Fig.5-8 Accuracy test of Lenet-5 convolutional Neural Network

图 5-9 为 Lenet-5 卷积神经网络在 FPGA 平台计算的中间数据可视化后的架构图。从图中可以明显看出，Lenet-5 卷积神经网络共有 5 层中间数据，1 层输入图像数据，1 层输出结果数据。

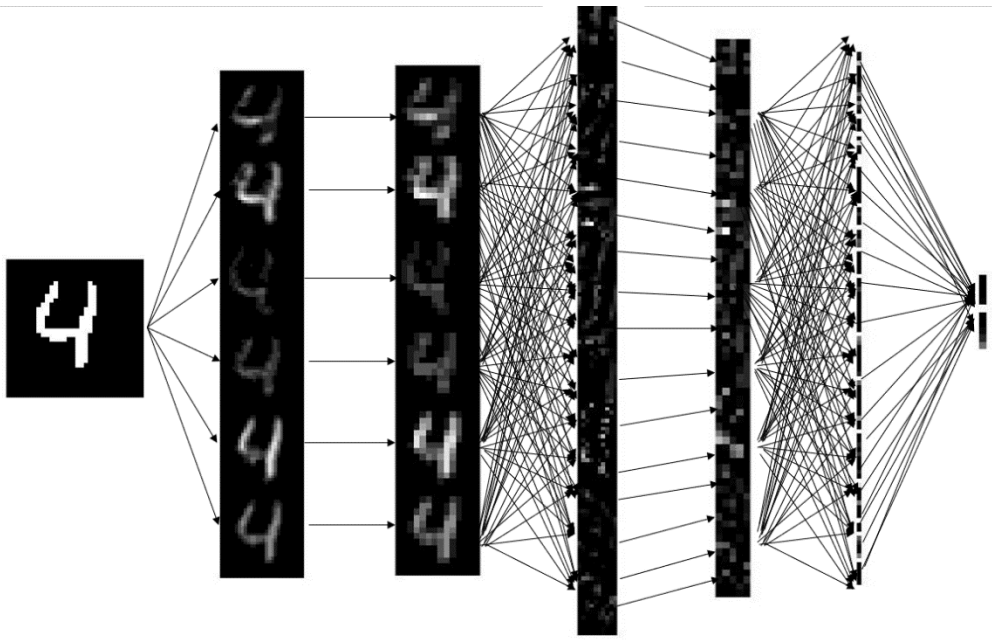


图 5-9 Lenet-5 卷积神经网络中间数据可视化

Fig.5-9 Lenet-5 Convolutional neural Network intermediate data visualization

其中第一层为卷积层，每个特征图对应一个权重和偏差（bias），从图中可

以看出经过第一次卷积之后，6 张特征图变得比原始图像模糊了一点，但保留了数字‘4’的基本框架，每张特征图所关注的字符的特征也是不一样的，这样可以提高识别的准确性，这就是特征提取的过程；第二层为池化层，从图中可以看出，经过第二层池化之后图片变得更模糊了，这就是最大值池化之后的效果，最大值池化可以大大减小数据量，并保留图像特征的关键信息；第三层为卷积层，从图中可以看出，经过第三层卷积之后，特征图已经到达比较高的层次，字符的形体变得不那么明显了，这时特征图所保留的特征已经是比较高的层次了；第四层为池化层，特征图通过最大值池化之后进一步缩小，同时也减小了后续计算量；第五层为卷积全连接层，经过第五层计算之后，特征图被进一步提取成大小为 1*120 的特征矩阵，特征矩阵最后经过一次全连接计算就可以得到最后的结果。

从图中可以明显看出，输出层向量的第五个输出单元是白色的，因为是从 0 开始数的，所以数字‘4’在第五行，这说明网络识别结果为‘4’，证明数字识别成功。而且从字符识别的结果来看，数字 4 的置信度比其他字符的置信度要高得多，这保证了网络识别的稳定性和健壮性。

5.7 本章小结

在分析国内外 FPGA 平台部署卷积神经网络案例的基础上，以提高字符识别系统的实时性和卷积神经网络基本计算模块的复用性为目的，通过理论分析、仿真计算和上板验证对 Lenet-5 卷积神经网络在 FPGA 平台的推演过程进行了研究。

本节提出了卷积神经网络在 FPGA 平台的部署方案，针对 XC7k325tffg-676-2FPGA 平台的特性提出了卷积计算地址读取方案——首地址锚定法；提出了适用于提出了卷积神经网络在 FPGA 平台上层间通信和流水线设计的方案并设计实现；提出了 Lenet-5 计算权值在 FPGA 平台的量化方案并保证了可允许范围内的精度损失；规范化了卷积神经网络基本计算模块并完成 IP 测试与封装，如 Convolution IP 核、Pooling IP 核以及 Full Connection IP 核等。为之后实现复杂的深度学习网络在 FPGA 平台的实现打下了坚实的基础。

本节在第 3 章对卷积神经网络运算过程进行分析的基础上，在 FPGA 平台搭建了 Lenet-5 卷积神经网络的硬件电路，并对其进行仿真调试，实现了字符识别功能。针对卷积和池化计算的特点，提出了图像数据读取地址生成方案——首地址锚定法；通过在不同计算层间加入使能信号，实现了 Lenet-5 卷积神经网络的不同计算层之间的层间通信和层级递进的计算协同；通过研究补码计算规律，提出了数值计算的防溢出方案；通过在相邻计算层间加入 Block RAM 实现了网络中间数据的存储和网络流水线设计，实现系统同时图像连续处理，大大

提高了系统的运行速度。并在此基础上，通过加入图像预处理模块，提高了字符识别系统在恶劣环境下的工作效果。

结 论

本文对 Lenet-5 在 FPGA 平台的移植进行了理论和实践研究，利用手写体数字的数据集对 Lenet-5 网络权重进行了训练，使用数据量化把权重转换为八位数据，使用数学软件对 Lenet-5 模型的计算进行了模拟，并证明了数据量化之前的算法精度在 97.17%，随后成功在 FPGA 平台部署了 Lenet-5 算法，测得单张图片识别速度为 0.439ms。同时，本文对 Lenet-5 在 FPGA 的算法移植进行了流水线优化，并且在识别多张图片时，单张图片识别速度可以达到 0.2ms。本论文的主要创造性工作归纳如下：

1. 本文分析了不同环境下不同的图像预处理算法对字符识别系统工作特性的影响，对于后续目标识别的项目提供了一定的实验基础，对于以深度学习网络为手段实现的目标识别算法在 FPGA 平台的部署提供了很好的实验资料。

2. 在分析国内外 FPGA 平台部署卷积神经网络案例的基础上，以提高字符识别系统的实时性和卷积神经网络基本计算模块的复用性为目的，通过理论分析、仿真计算和上板验证对 Lenet-5 卷积神经网络在 FPGA 平台的推演过程进行了研究。

3. 本文提出了卷积神经网络在 FPGA 平台的部署方案，针对 XC7k325tffg-676-2FPGA 平台的特性提出了卷积计算地址读取方案——首地址锚定法；提出了适用于提出了卷积神经网络在 FPGA 平台上层间通信和流水线设计的方案并设计实现；提出了 Lenet-5 计算权值在 FPGA 平台的量化方案并保证了可允许范围内的精度损失；规范化了卷积神经网络基本计算模块并完成 IP 测试与封装，如 Convolution IP 核、Pooling IP 核以及 Full Connection IP 核等。为之后实现复杂的深度学习网络在 FPGA 平台的推演项目设计打下了坚实的基础。

4. 本文在对卷积神经网络运算过程进行分析的基础上，在 FPGA 平台搭建了 Lenet-5 卷积神经网络的硬件电路，并对其进行仿真调试，实现了字符识别功能。针对卷积和池化计算的特点，提出了图像数据读取地址生成方案——首地址锚定法；通过在不同计算层间加入使能信号，实现了 Lenet-5 卷积神经网络不同层之间的层间通信和层级递进的计算协同；通过研究补码计算规律，提出了数值计算的防溢出方案；通过在相邻计算层间加入 Block RAM 实现了网络中间数据的存储和网络流水线设计，提高了整体运行速度。并在此基础上，通过加入图像预处理模块，提高了字符识别系统在恶劣环境下的工作效果。

今后还应在以下几个方面继续深入研究：

1. 本文仅是使用卷积模块、池化模块以及全连接模块实现了 Lenet-5 卷积神经网络，以后将会尝试 YOLOv3 等大型网络在 FPGA 平台上的移植。由于已经有了各个功能模块，后续的开发将会变得更简单。对于移植 YOLOv3，我会将重点放在计算优化和存储分配方案上，如果上述问题能得到解决，YOLOv3 的高速运行将成为可能，智能交通也不再遥远。

2. 本文在权重计算时使用的是数值量化，每次计算都会损失一定的精度，后续将会在 FPGA 的小数计算上继续研究，寻找更适合 FPGA 的小数计算方法。

3. 本文的研究还未经过实践的检验，或许还存在一些小问题还有待改进，前期芯片设计的项目都要经过 FPGA 芯片的不断调试，直到芯片设计成熟之后投入量产，这也是 FPGA 过渡到专用芯片的必然路径。

参考文献

[1] 范世朝，郑国强，孙国庆，韩旭. 基于 FPGA 和卷积神经网络的人脸识别系统[J]. 电子元器件与信息技术, 2021,5(05)：98-99.

[2] 沈成龙，王笑梅，王晨. 民国纸币冠字号码的提取与分割[J]. 计算机仿真，2022，39(01)：437-440+455.

[3] 孙鹏，李赛，寇鹏，朱佳俊，韩喜瑞，张天奕. 基于手持设备图像的车牌定位与车牌识别系统设计[J]. 软件工程，2022，25(01)：29-32.

[4] 陈黎，黄心汉，王敏，李炜. 基于聚类分析的车牌字符分割方法[J]. 计算机工程与应用，2002(06)：221-222+256.

[5] 迟晓君，孟庆春. 基于投影特征值的车牌字符分割算法[J]. 计算机应用研究，2006(07)：256-257.

[6] 李文举，梁德群，王新年，于东. 质量退化的车牌字符分割方法[J]. 计算机辅助设计与图形学学报，2004(05)：697-700.

[7] Rongshi D, Yongming T. Accelerator Implementation of Lenet-5 Convolution Neural Network Based on FPGA with HLS[C]// International Conference on Circuits, System and Simulation, 2019, 64-67.

[8] 张丽丽. 基于 HLS 的 Tiny-yolo 卷积神经网络加速研究[D]. 重庆大学, 2017.

[9] Madadum H, Becerikli Y. FPGA-Based Optimized Convolutional Neural Network Framework for Handwritten Digit Recognition[C]// International Informatics and Software Engineering Conference: Innovative Technologies for Digital Transformation, 2019.

[10] Chen Y.-H, Fan C.-P, Chang R.C. Prototype of Low Complexity CNN Hardware Accelerator with FPGA-based PYNQ Platform for Dual-Mode Biometrics Recognition[C]// International SoC Design Conference, 2020, 189-190.

[11] Shi Y, Gan T, Jiang S. Design of Parallel Acceleration Method of Convolutional Neural Network Based on FPGA[C]// International Conference on Cloud Computing and Big Data Analytics, 2020, 133-137.

[12] 王春林. 基于 ZYNQ 的卷积神经网络软硬件协同设计研究与实现[D]. 大连海事大学， 2020.

[13] 邹丹音. 基于深度学习的目标检测算法 FPGA 实现[D]. 哈尔滨工业大学，2019.

[14] 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(02)：81-88+108.

[15] 洪波，文鹏程，李亚辉. 基于 FPGA 的卷积神经网络加速器设计[J]. 信息技术与信息化， 2022(04)：117-120.

[16] 肖帅，杨秀芝. 基于深度学习的 HEVC 帧内预测算法研究及 FPGA 硬件加速[J]. 广播电视网络， 2022，29(03)：107-110.

[17] 曹远杰，高瑜翔，杜鑫昌，涂雅培，吴美霖. 基于改进 YOLOv4-Tiny 的 FPGA 加速方法[J]. 无线电工程， 2022，52(04)：604-611.

[18] Maraoui A, Messaoud S, Bouaafia S, Ammari A.C, Khriji L, Machhout M. PYNQ FPGA Hardware implementation of LeNet-5-Based Traffic Sign Recognition Application[C]// IEEE International Multi-Conference on Systems, Signals and Devices, 2021, 1004-1009.

[19] 张玉婷. 基于卷积神经网络的手势识别算法优化及嵌入式实现[D]. 西安邮电大学， 2018.

[20] 陈浩敏，姚森敬，席禹，张凡，辛文成，王龙海，任超. YOLOv3-tiny 的硬件加速设计及 FPGA 实现[J]. 计算机工程与科学， 2021，43(12)：2139-2149.

[21] Zhang C, Yue X, Wang R, Li N, Ding Y. Study on Traffic Sign Recognition by Optimized Lenet-5 Algorithm[C]// International Journal of Pattern Recognition and Artificial Intelligence, 2020.

[22] Zhang C-W, Yang M-Y, Zeng H-J, Wen J-P. Pedestrian detection based on improved LeNet-5 convolutional neural network[J], Journal of Algorithms and Computational Technology, 2019, 13.

[23] 赵志宏，杨绍普，马增强. 基于卷积神经网络 LeNet-5 的车牌字符识别研究[J]. 系统仿真学报， 2010，22(03)：638-641.

[24] 李丹，沈夏炯，张海香，朱永强. 基于 Lenet-5 的卷积神经网络改进算法[J]. 计算机时代， 2016(08)：4-6+12.

[25] 王秀席，王茂宁，张建伟，程鹏. 基于改进的卷积神经网络 LeNet-5 的车型识别方法[J]. 计算机应用研究， 2018，35(07)：2215-2218.

[26] 邓长银，张杰. 基于改进 LeNet-5 模型的手写数字识别[J]. 信息通信，2018(01)：109-112.

[27] 张力超，马蓉，张垚鑫. 改进的 LeNet-5 模型在苹果图像识别中的应用[J]. 计算机工程与设计， 2018,39(11):3570-3575.

[28] 刘金利，张培玲. 改进 LeNet-5 网络在图像分类中的应用[J]. 计算机工程与

应用， 2019，55(15)：32-37+95.
[29] 汪贵平，盛广峰，黄鹤，王会峰，王萍. 基于改进 LeNet-5 网络的交通标志识别方法[J]. 科学技术与工程， 2018，18(34)：78-84.
[30] 党倩，马苗，陈昱莅. 基于二级改进 LeNet-5 的交通标志识别算法[J]. 陕西师范大学学报(自然科学版)， 2017，45(02)：24-28.
[31] 马苗，陈芳，郭敏，陈昱莅. 基于改进 LeNet-5 的街景门牌号码识别方法[J]. 云南大学学报(自然科学版)， 2016，38(02)：197-203.
[32] Sun L, Wang Y, Dai L. Convolutional neural network protection method of lenet-5-like structure[C]// ACM International Conference Proceeding Series, 2018, 77-80.
[33] Li T, Jin D, Du C, Cao X, Chen H, Yan J, Chen N, Liu S. The image-based analysis and classification of urine sediments using a LeNet-5 neural network [C]// Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization, 2020, 109-114.
[34] Islam M.R, Matin A. Detection of COVID 19 from CT Image by the Novel LeNet-5 CNN Architecture [C]// International Conference on Computer and Information Technology, Proceedings,2020.
[35] Kayed M, Anter A, Mohamed H. Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture[C]// Proceedings of 2020 International Conference on Innovative Trends in Communication and Computer Engineering, 2020, 238-243.
[36] Tan S, Tan Z. Improved LeNet-5 Model Based On Handwritten Numeral Recognition[C]// Proceedings of the 31st Chinese Control and Decision Conference, 2019, 6396-6399.
[37] Zhang Z.-H, Yang Z, Sun Y, Wu Y.-F, Xing Y.-D. Lenet-5 Convolution Neural Network with Mish Activation Function and Fixed Memory Step Gradient Descent Method[C]// International Computer Conference on Wavelet Active Media Technology and Information Processing, 2019, 196-199.
[38] 葛婷. 几种数字图像滤波算法[D]. 南京信息工程大学, 2006.
[39] 关新平，赵立兴，唐英干. 图像去噪混合滤波方法[J]. 中国图象图形学报, 2005(03)：332-337.
[40] 方莉，张萍. 经典图像去噪算法研究综述[J]. 工业控制计算机, 2010,23(11)：73-74.
[41] 宁媛，李皖. 图像去噪的几种方法分析比较[J]. 贵州工业大学学报(自然科学版), 2005(04)：63-66.
[42] 刘祝华. 图像去噪方法的研究[D]. 江西师范大学, 2005.

[43] 吴迪，朱青松. 图像去雾的最新研究进展[J]. 自动化学报， 2015，41(02)：221-239.

[44] 王浩，张叶，沈宏海，张景忠. 图像增强算法综述[J]. 中国光学, 2017，10(04)：438-448.

[45] 陈永亮. 灰度图像的直方图均衡化处理研究[D]. 安徽大学, 2014.

[46] 刘雪峰，刘秋月. 图像阴影检测与增强算法研究[J]. 现代电子技术, 2022，45(10)：105-110.

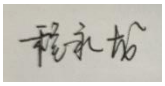
[47] 李晨曦，李健. 基于 GAN 和 U-Net 的低光照图像增强算法[J]. 计算机系统应用， 2022,31(05)：174-183.

[48] 马龙，马腾宇，刘日升. 低光照图像增强算法综述[J]. 中国图象图形学报，2022,27(05)：1392-1409.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于 FPGA 和 Lenet-5 的字符识别系统设计与实现》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：  日期： 2022 年 6 月 17 日

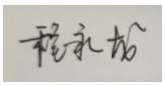
学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：  日期： 2022 年 6 月 17 日

导师签名：  日期： 2022 年 6 月 17 日

致 谢

经过几个月的刻苦攻关，我的硕士毕业论文即将完成，回顾论文写作的每一个过程，都是那么的美好而又充实。

在论文选题初期，周荻教授和林玉荣副教授就给了我很大的支持，这让我有勇气选择从未做过的题目。也是在他们两位的支持和鼓励下，我的毕业论文才会顺利地完成。

首先感谢我的授业恩师周荻教授。两年前有幸求学于恩师门下，恩师不以吾愚钝，耐心蒙之，亲人待之，值以信赖，委以重任。两年的教诲此刻犹如一幅幅画面浮现在眼前，恩师渊博的专业知识，严谨的治学态度，精益求精的工作作风，朴实乐观的人格魅力对我影响深远。两年间我不仅仅收获了学位，更学会了以审慎和积极态度面对人生。

然后感谢我的恩师林玉荣老师，感谢您对我的帮助指点和无私教诲，对我一直以来的帮助和支持。虽然我即将毕业，但是我将把恩师的教诲永远铭记于心，您永远都是指引我前进方向的灯塔，是我最坚强的后盾。让我感激涕零，常怀滴水涌泉之心。衷心感谢恩师的教诲！老师，您辛苦了。

在论文中期答辩时，我沉迷于技术攻关的迷宫里不可自拔，忽视了论文的学术性，孙光辉老师在听了我的答辩之后，当即给我拉回现实，让我意识到自己论文存在的不足，在这里由衷地感谢孙光辉老师！如果没有孙光辉老师的批评，我的论文可能不会顺利完成。

在论文写作时期，我参加了博士的综合考核，在实验室同门王琳玮师兄、李玉堂同学、马悦萌等同门的鼎力帮助下，在父母的全力支持下，顺利完成了博士的综合考核，并取得了不错的结果。在这里衷心地感谢我的师兄和同学的帮助，感谢父母的培养。

能力越大，责任也就越大。国家当今处于技术被动的境地，身为当代大学生和中国共产党党员，必须以身作则，投入到相关的技术研究领域，以报效国家。这样，我作为一名党员和知识分子的社会价值才得以体现。中国梦的实现，需要我们共同努力！我们都在路上。

感谢实验室全体老师和同窗们的热情帮助和支持！