文章编号:1671-4598(2024)10-0180-07 DOI:10.16526/j.cnki.11-4762/tp.2024.10.026 中图分类号:TH274 文献标识码:A

基于 FPGA 的手写蒙文字体转换系统设计及实现

李永辉[1]，颜世威[2]，施 展[1]，王立国[1]，冯 冲[1]
(1. 大连民族大学 信息与通信工程学院，辽宁 大连 116600；
2. 国核自仪系统工程有限公司，上海 200241)

摘要：蒙文字体转换在促进蒙文应用和推广、丰富中国文化多样性以及促进蒙古族地区经济繁荣方面具有关键作用；针对蒙文字体转换的效率和准确率低的问题，提出一种基于轻量化卷积神经网络（CNN）和 FPGA 硬件加速器的方法；即使用 CNN 进行手写蒙文识别，并结合字体转换库，通过识别结果和字体的映射关系实现了简单高效的蒙文字体转换；相比于其他方法，该方法结合了高效的 CNN 和 FPGA 硬件加速器的优势，既提高了转换效率，又满足了设备成本、功耗和便携性的需求；使用 Xilinx 公司的 XC7Z020CLG400-2 完成网络模型电路的设计和优化工作，在此基础上实现了手写蒙文字体转换系统，测试结果表明，手写体蒙文转换为目标字体的准确率为 95.62%，转换时间为 1.43 ms，功耗为 0.341 W，加速器峰值吞吐量为 6.64 Gops；研究成果对于促进蒙古族文化传承和经济发展具有重要意义。

关键词：蒙文字体转换；FPGA 加速器；低功耗系统

Design and Implementation of Handwritten Mongolian Script Conversion System Based on FPGA

LI Yonghui[1], YAN Shiwei[2], SHI Zhan[1], WANG Liguo[1], FENG Chong[1]
(1. College of Information and Communication Engineering, Dalian Minzu University, Dalian 116600, China;
2. State Nuclear Power Automation System Engineering Company, Shanghai 200241, China)

Abstract: Mongolian script conversion plays a pivotal role in promoting the application and dissemination of the Mongolian script, enriching the diversity of Chinese culture, and fostering economic prosperity in Mongolian regions. Aimed at the poor efficiency and accuracy of Mongolian script conversion, this paper proposes a method based on lightweight convolutional neural network (CNN) and FPGA hardware accelerator The CNN is used to carry out the handwritten Mongolian script recognition, and combined with the font conversion library, a simple and efficient Mongolian font conversion is achieved through the mapping relationship between the recognition results and the fonts. This approach has the advantages over other methods in the efficiency of the CNN and FPGA hardware accelerator, improving the conversion efficiency while meeting the requirements of device cost, power consumption, and portability. The circuit of network model is designed and optimized by using Xilinx's XC7Z020CLG400-2 chip. On this basis, a handwritten Mongolian font conversion system was implemented. Test results show an accuracy of 95.62% for converting the handwritten Mongolian script to the target font, with a conversion time of 1.43 ms, a power consumption of 0.341 W, and a peak throughput of 6.64 Gops for the accelerator. It is of significance for the research results to promote the preservation of Mongolian culture and economic development.

Keywords: Mongolian script conversion; FPGA accelerator; low-power system

0 引言

文字作为人类社会的符号系统和沟通工具，在传递信息、记录知识、促进文化传承和推动思维发展等方面发挥着重要作用。蒙文作为蒙古族的主要文字系统，具有保护和传承蒙古族语言、历史和文化的重要意义[1]。蒙文字体转换在促进蒙文应用和推广、丰富中国整体文化多样性以及促进蒙古族地区经济繁荣方面发挥着关键作用。通过字体转换，蒙文可以在数字化媒体、印刷品和网页设计等领域更广泛地应用，扩大其影响力和可视性，为蒙古族文化传承提供支持并促进经济发展[2]。

传统的蒙文字体转换流程包括确定风格、制作基本字形规范，设计模板字并扩充字符集[3]。随着计算机技术的发展，自动化的字体转换算法和工具逐渐流行，提高了效

收稿日期:2023-09-03； **修回日期**:2023-10-13。
基金项目:国家自然科学基金(62071084)；校研究生项目(YJS2023JG03,YJS2023JG11)。
作者简介:李永辉(1997-),男,硕士研究生。
通讯作者:冯 冲(1977-),男,博士,副教授,硕士生导师。
引用格式:李永辉,颜世威,施 展,等. 基于 FPGA 的手写蒙文字体转换系统设计及实现[J]. 计算机测量与控制,2024,32(10):180-186,200.

率。例如，文献［4］在一小组字母上成功地使用深度学习模型来区分字体，并生成相同风格的字符。文献［5］提出了一个端到端的堆栈条件的生成式对抗网络模型，有效地实现了 10 000 种不同英文字母字体间的风格迁移。文献［6］提出了一种使用递归神经网络（RNN，recurrent neural network）分别作为识别汉字的判别式模型和生成汉字的生成式模型的框架。文献［7］提出了一个可生成中国书法风格字体图像的生成式对抗网络模型。这些算法都部署在较高版本的计算机设备中，需要较大的计算资源[8]。

由于字体生成算法需要大量计算资源，无法在资源有限的移动设备上搭载。而在实际应用中，移动设备通过手写输入法和笔画输入法虽然可以识别和预测[9]，但处理复杂的字体和少数民族文字时准确率低，导致转换结果质量下降，使得在移动设备上实现高效、准确的蒙文字体转换成为挑战。本文提出使用轻量化卷积神经网络（CNN，convolutional neural network）进行手写蒙文识别[10]，并通过识别结果构建蒙文字体转换库。轻量化 CNN 模型在移动设备上运行效率高，实现实时的手写蒙文识别。建立字符与蒙文字体的对应关系，实现简单高效的蒙文字体转换，并在移动设备上进行。方法提高转换效率、置信度和准确率，为蒙古族文化传承和经济发展提供支持。具有较好解释性，方便用户理解和调整转换结果，确保蒙文字体转换效果符合预期。

轻量化卷积神经网络搭载在移动设备上是重要技术进步，使移动设备可以本地高效蒙文字体转换。然而，复杂的转换任务或需更高性能场景中，移动设备处理能力仍面临挑战。为进一步优化性能，引入 FPGA（Field Programmable Gate Array）加速器应用是解决方案。搭载在 FPGA 上的轻量化卷积神经网络发挥并行计算和低功耗优势。FPGA 作为硬件加速器，专门优化卷积神经网络计算过程，加快运算速度[11]。相较于 CPU 和 GPU，FPGA 具有更高并行计算能力，适用于密集计算任务[12]。蒙文字体转换在 FPGA 加速器上高效处理，进一步提升移动设备上转换效率。除了计算能力，相较于 GPU 等高功耗设备，FPGA 具有更低的功耗特性。

因此本文设计一种基于 FPGA 的蒙文字体转换系统，以实现高准确率和转换效率的蒙文字体转换。这将有助于将大量的蒙古文纸质文档转化为电子文档，提高蒙古文文档资源的利用效率。这不仅有利于促进蒙古族地区的社会发展和科技进步，还有助于更好地传承和弘扬少数民族的优秀传统文化，促进民族团结，并深入饯行民族共同体的发展理念。因此，这一领域的研究和发展具有重要的战略意义。

由于受数据集大小、手写体文字风格多样等因素限制，目前手写蒙文字体转换技术尚达不到实际应用水平，因此本文以手稿文件的数字化存储为背景，针对文稿在纸张、字迹等对数字化保存过程中的影响，在字体转换过程中引入神经网络模块对图像特征进行提取来解决字迹模糊和残缺等问题，提高字体识别准确度，使用 FPGA 对神经网络进行加速来解决在实时性场景下应用的问题并降低设备的使用成本。

1　系统方案设计及实现

本系统设计选定 Xilinx ZYNQ-7020 作为系统实现的目标芯片，其芯片资源如表 1 所示：其中，包括可编程逻辑单元 53 200 个、140 个 BRAM（4.9 M），220 个可编程 DSP 单元，同时提供时钟管理和通用输入输出接口。

表 1　ZYNQ7020 开发板主要参数

名称	具体参数
Part Number	XC7Z020
Maximum Frequency	666 MHz
Look-UP Tables(LUTs)	53 200
Block RAM(＃36 kB Blocks)	140
DSP slices	220
Flip-flops	106 400
IO	125

系统总体设计方案如图 1 所示，字体转换系统包括字符识别系统设计和字体库两个部分。其中字符识别系统可以分为 PC 机和 FPGA 两个部分：其中 PC 机部分主要是设计轻量化网络模型、数据集的处理与制作以及网络权值数据的量化 3 部分，FPGA 部分包括数据传输系统的设计与实现以及 CNN 硬件加速器的部署。

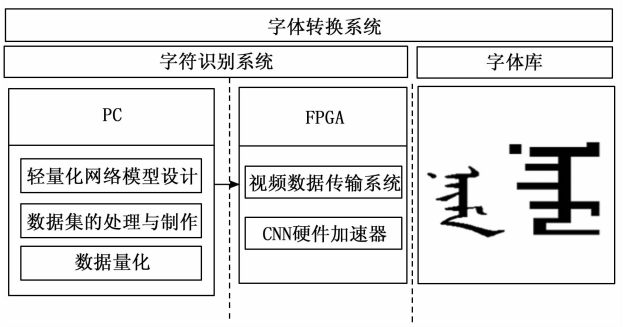

图 1　系统设计方案

整个系统的工作流程如下：首先，在 PC 机上进行网络模型的训练和数据集的处理与制作；然后，对网络权值数据进行量化，并将其传输到 FPGA 中，在 FPGA 上，利用部署的 CNN 硬件加速器对量化后的权值数据进行加速计算，得到字符识别的预测结果；最后，通过字体库与预测结果的对应关系，获取所匹配的字体数据。蒙文字体转换系统如图 2 所示，包括手写体图像采集单元、手写体识别单元、字体库模块和图像及结果显示单元。

该系统集成了摄像头、LCD 等模块，摄像头传感器实时采集字体的结构图形，并通过在数据通路关联手写体识别单元，实现基于视频数据的手写体字体转换系统，同时转换结果在 LCD 模块上反馈给用户。

1.1　字符识别系统的设计与实现

1.1.1　数据集图像处理与制作

首先，蒙文图像预处理。蒙文序数手写体是由人工书

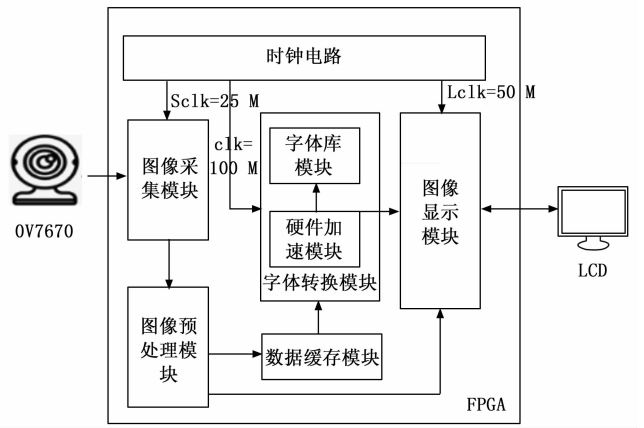

图 2 手写蒙文字体转换系统框图

写，由于手写和图像获取过程中存在墨迹干扰、图片扫描阴影干扰、图片分割尺寸不统一等瑕疵，对原始数据进行了图像滤波、去噪、图形的膨胀与腐蚀以及图形的伸缩变换等操作，以减小数据集瑕疵对网络识别结果准确性的影响，图像处理前后的对比如图 3 所示。

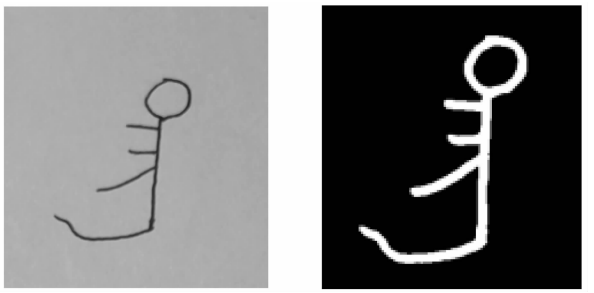

图 3 图像处理前后对比图

其次，蒙文数据集建立：1）蒙文手写体序数图库分类。按照比例 6∶2∶2 将数据图库划分为训练图像数据集（Mengwen. train）、验证图像数据集和测试图像数据集（Mengwen. test）；2）按照分类后的图片数据所在的文件夹名称依次生成所对应的标签数据集和图像数据集，并通过张量的形式进行网络训练。张量构建过程示例如下：在训练图像数据集中，标签集（Mengwen. train. label）是由［2 250，10］的张量表示，“2 250”用来表示对应位置的图片，“10”用来索引每张图片的标签。标签是分别由 0～9 来命名，以［0，0，0，0，0，0，0，0，0，0］二进制数形式表示，如标签 0、4、2，对应的数据形式分别为［1，0，0，0，0，0，0，0，0，0］、［0，0，0，0，1，0，0，0，0，0］、［0，0，1，0，0，0，0，0，0，0］，如图 4 所示。

对于图像数据集（Mengwen. train. image），每张图片大小为 56 * 56，在以图像像素数据构成数组表示后，其长度为 3 136，因此，训练集中图像数据集是由［2 250，3 136］的张量表示（如图 5 所示），“2 250”用来表示对应位置的图片，“3 136”用来索引每张图片的像素。

1.1.2 AlexNet 模型结构改进与训练及数据处理

本文选取 AlexNet 网络作为测试模型，AlexNet 模型相

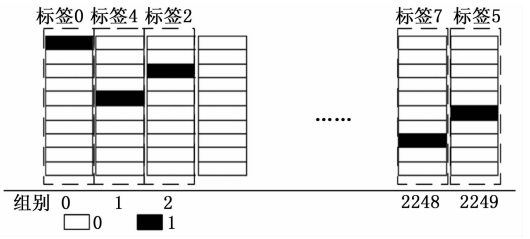

图 4 标签数据表示形式

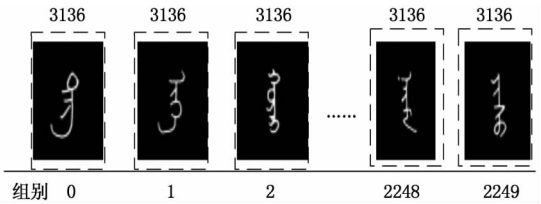

图 5 图像数据表示形式

比传统的 LeNet5 和其他传统模型具有更好的识别精度。AlexNet 网络是由多伦多大学的 Krizhevsky 等人[13]在 2012 年提出。然而 AlexNet 模型存在数据参数巨大、训练时间长、识别速度慢等缺点。因此，针对以上缺点以及数据集图片分辨率特点，本文使用一种改进的模型算法：对两层隐藏全连接层进行卷积、池化替换[14]，改进后的 AlexNet 模型如表 2 所示。

表 2 改进型 AlexNet 模型结构

层次	输入	步长	填充	卷积核尺寸	输出
Conv-1	56 * 56	1	same	3 * 3	56 * 56 * 4
MaxPool-1	56 * 56 * 4	2	—	—	28 * 28 * 4
Conv-2	28 * 28 * 4	1	same	3 * 3	28 * 28 * 4
MaxPool-2	28 * 28 * 4	2	—	—	14 * 14 * 4
Conv-3	14 * 14 * 4	1	same	3 * 3	14 * 14 * 4
Conv-4	14 * 14 * 4	1	same	3 * 3	14 * 14 * 8
Conv-5	14 * 14 * 8	1	same	3 * 3	14 * 14 * 16
MaxPool-3	14 * 14 * 16	2	—	—	7 * 7 * 16
Conv-6	7 * 7 * 16	1	same	3 * 3	7 * 7 * 32
Conv-7	7 * 7 * 32	1	same	3 * 3	7 * 7 * 32
GMP	7 * 7 * 32	—	—	—	32
FC	32	—	—	—	11

在完成网络训练后得到完整的模型权重参数，但数据类型均采用浮点数表示，而在硬件部署时，浮点数尾数和指数操作较为困难，通常采用定点数计算[15]。为了能在 FPGA 上有效实现 AlexNet 网络模型，需要将前述网络参数由浮点型转换为定点型。由于权重系数通过训练得到，很容易得到输入的最大值（m_a）和最小值（m_b），设 $M=max(|m_a|,|m_b|)$，将权重系数除以 M，得到对任意输入数据在该层的数值不超过 1，并且在网络的每一层使用相同的方法，使每一层输入的值始终在［－1，1］之间，之后通过式（1）完成浮点数到定点数转换。

$$x_b = x \cdot 2^N \tag{1}$$

其中：x 为浮点数，x_b 为转换后的定点数。经过转换后的数值必须除以 2^N，才可以得到实际值，在硬件实现时，N 就是系统的位宽。

在 FPGA 中，同一个定点数可以有不同位宽的实现形式，不同网络权值位宽与网络识别结果准确性的关系需要通过实验测试，测试数据如图 6 所示。图中显示，位宽为 12、13 和 14 时，网络识别准确性已经达到 100%，即位宽增加不再影响识别准确性。所以，相同识别准确性情况下，考虑节约 FPGA 资源的因素，选定本系统数据位宽为 12。

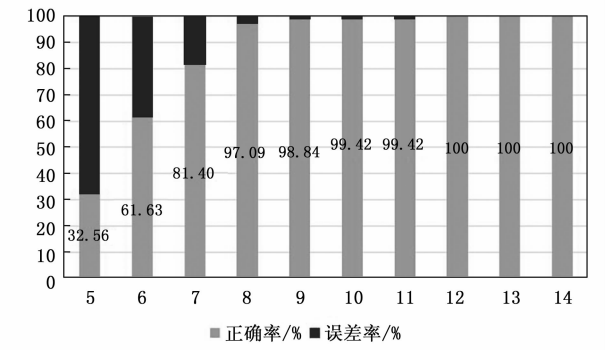

图 6　数据位宽与预测准确度关系

1.1.3　基于 FPGA 卷积神经网络硬件加速器设计

FPGA 上的卷积神经网络硬件加速器通常采用单计算单元架构或多计算单元架构来处理卷积计算。计算单元（CE，computing element）作为独立的硬件子系统，执行矩阵乘法、加法和激活函数等基本操作[16]。与单计算单元架构逐层计算的方式不同，多计算单元架构能够同时处理不同的卷积层，增加了并行性。由于每个卷积层核心的计算是相互独立的，设计上更加灵活。因此，本文中的卷积处理模块采用了多计算单元架构来实现。通常 CNN 模型包含多个全连接层，这些全连接层的计算量相对较小，因此对整个系统的吞吐量效率影响相对较小。因此，为了方便起见，采用单计算单元架构来实现全连接层，这样可以简化设计和提高整体性能。本文所提出的神经网络并行加速设计如图 7 所示。

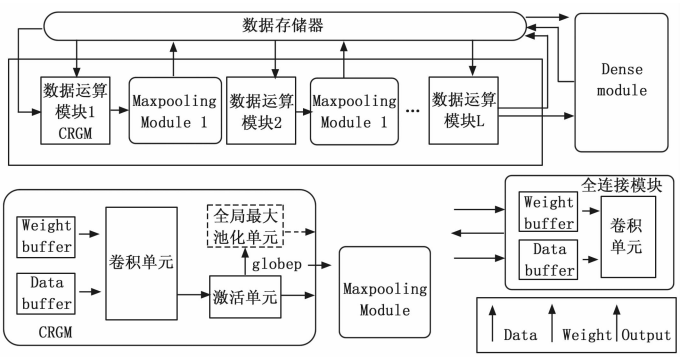

图 7　神经网络并行加速设计结构图

在卷积核心单元设计时，本文使用了 3 个位宽为 15 的寄存器来缓存输入特征图的图像数据，第 0 号寄存器 reg0［14：0］相较于输入特征图的图像数据延迟一个时钟周期进行数据调用，剩余寄存器相较于前一个寄存器延迟一个时钟周期进行数据调用。这样到第 4 个时钟周期时 3 个寄存器中可同时调用 3×3 大小的输入特征图，并且以后每个时钟周期都可调用同样大小的输入特征图图像数据，这样可以实现每个时钟周期进行一次卷积运算所需的数据调用。同时，采用权值共享技术，将同一个卷积核权值用于所有的卷积计算，避免了每次计算都要重新加载权值的时间，并将卷积核权值存储在片上存储器中，每次将 3 * 3 的数据矩阵与同一份权值矩阵进行卷积运算，从而实现卷积计算，如图 8 所示。

本设计采用模块化设计方法，通过卷积复用的方式将上一次卷积的输出作为下一次卷积的输入，实现了对卷积操作的高效复用。它减少了计算量，节省了存储空间，加快了推理速度，同时提高了硬件资源的利用率。通过充分利用硬件资源，包括使用多个计算单元并行执行计算，加速了网络推理过程，并提高了计算速度和效率。整体设计使得网络能够在各种环境下高效运行，并充分发挥硬件资

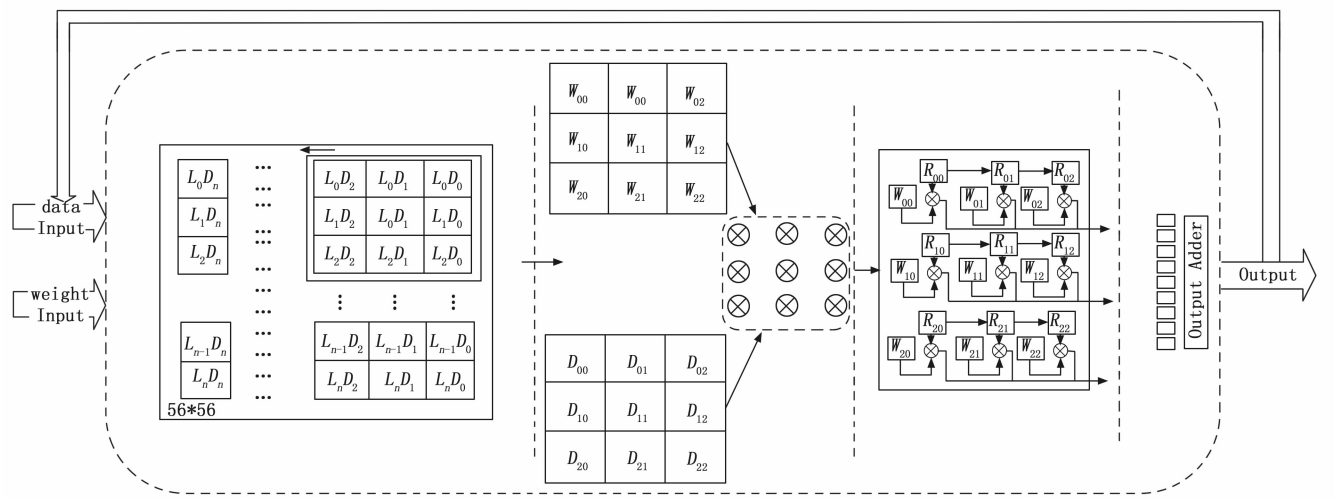

图 8　卷积运算结构图

源的优势，从而实现了快速而准确的图像分类和识别。

1.1.4 网络模型的 FPGA 部署

卷积层在卷积神经网络中扮演着至关重要的角色，其卷积运算贡献了整个 CNN 硬件加速器中 90%以上的总运算量，因此卷积层电路的设计成为整个系统的核心。进一步地，卷积运算电路是卷积层电路的核心，下面介绍在 FPGA 芯片上部署网络模型中的卷积运算电路设计过程。

1.1.4.1 卷积计算流程

在卷积模块中，卷积运算的功能是卷积核在特征图矩阵中以固定的滑动步长与特征图子矩阵进行卷积，并且每次滑动都需要计算卷积核与特征图中对应子矩阵的点积，也即卷积核对应元素与子矩阵对应元素分别乘，然后累加求和，如图 9 所示。

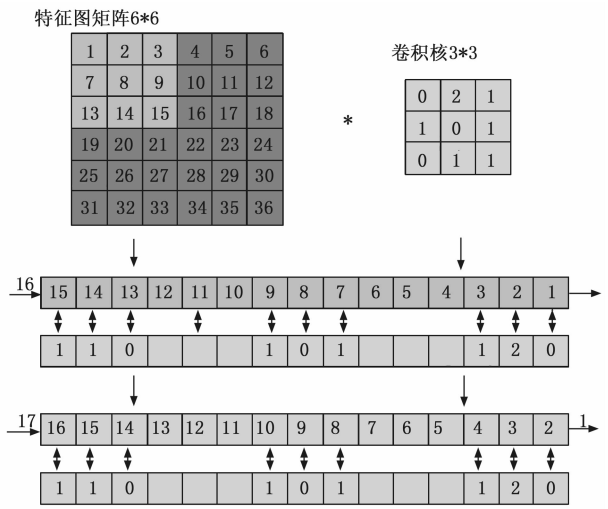

图 9 特征图矩阵和卷积核展开过程

1.1.4.2 运算结构选择

卷积计算电路设计本质是一种乘累加电路设计，其中两个数据的乘法通过乘法运算符可直接实现，而良好的累加电路设计将提升系统运算速度，是工作重点。为了能让卷积计算电路拥有较好的并行性和时序收敛，确定选用二叉树型累加方式，如图 10 所示。

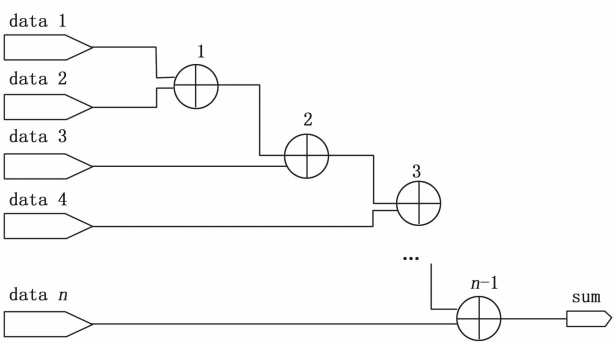

图 10 链式加法

特征图矩阵与卷积核展开之后具体的卷积运算过程如图 11 所示，其中 3 * 3 大小的卷积核与 3 * 3 的特征图子矩阵进行对应元素的乘累加（MAC）运算，因为数据位宽为 12，在 Verilog 代码中通过 $Y=A*B$ 的形式进行数据乘积。

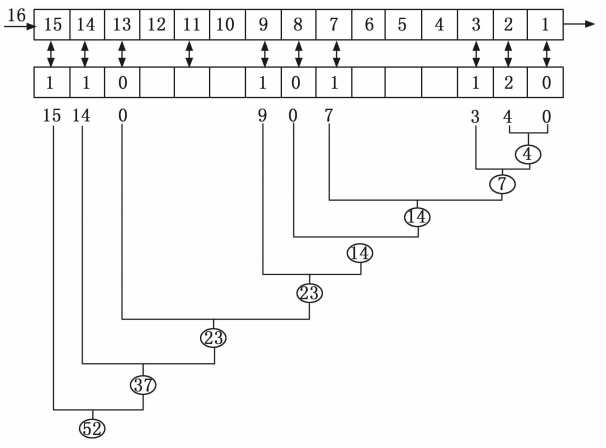

图 11 展开后的具体卷积运算过程

在本文所使用网络的卷积运算中，输入特征图数据和权重数据之间不存在依赖性，可以利用 FPGA 自身并行的特点一次进行 9 个元素的计算。另外可以复用卷积核，尽最大程度地发挥 FPGA 自身并行优势。

1.1.4.3 电路仿真验证

本文在设计位宽长度为 15 的寄存器数组用来存放特征图第一行、第二行以及第三行中与卷积核对应的元素，然后把卷积核里的元素固定在寄存器数组中的对应位置，最后进行乘法运算。卷积计算电路系统窗口滑动以及卷积计算过程在 Vivado 软件中进行仿真，如图 12 所示。

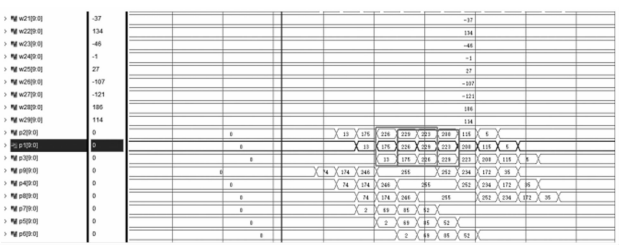

图 12 卷积计算过程仿真图

1.1.4.4 网络模型 FPGA 电路仿真验证

硬件加速电路在 FPGA 平台上部署完成后，首先进行卷积推断电路的测试，蒙文手写体序数 2 经过 python 处理为二进制文件 .txt 文件，参数个数为 3 136，经过 dp _ database 端口传入卷积神经网络电路进行运行推断，其结果会在 RESULT 端口显示，其仿真结果如图 13 所示。

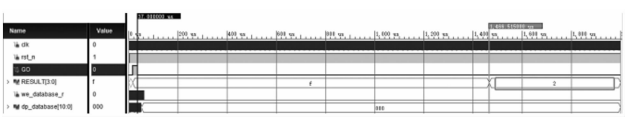

图 13 神经网络推断仿真结果图

1.2 蒙文字体库建立

Unicode 是一种字符编码标准，用于表示世界上几乎所有的字符。Unicode 对不同的字符分配了唯一的标识符。本文借鉴它的编码方式制作字体库，字体库是一个存储了不

同字体、字号下的字符图像的字典。在制作字体转换表时，首先需要收集一定数量的不同字体、字号下的字符图像，并进行预处理，如图像大小统一、字符去噪等。然后将处理好的字符图像与其对应的字体存储到字体转换表中，形成一个一对一映射表。在进行字符字体转换时，先使用字符识别技术确定输入字符的类型，然后根据输入字符的类型在预先构建的字体转换表中进行查找，得到对应的字符图像，最终输出经过风格转换后的字符图像，蒙文字体样式如图 14 所示。

图 14　蒙文字体样式示例

相比于生成式对抗神经网络模型，使用查找表进行字符字体转换的方法简化了模型的训练时间和对计算设备的资源占用，通过构建好字体转换表，使用查找表进行字符字体转换适合于需要处理大量字符图像的应用场景，如 OCR（Optical Character Recognition）系统等[17]。

2　结果与讨论

本文在 Python 的集成开发环境 Pycharm2017.2 下进行开发，模型搭建和模型训练使用基于 TensorFlow 的 Keras 编程框架。在训练过程中调用 metrics＝［keras.f1_score，keras.precision，keras.recall］，既准确率（Accuracy）、精度（Precision）、召回率（Recall）、F_1 值等指标使用网络训练的测试集来评价改进型 AlexNet 网络对蒙文序数识别的表现。其中，准确度表示预测正确的蒙文序数占总数的比例；精度表示预测正确的正类样本数占据总正类样本比例；召回率表示预测正确的正类样本数占据真实为正类的样本数的比例；F_1 值为精度和召回率的调和平均值。*Precision* 体现了模型对负样本的区分能力，*Precision* 越高，模型对负样本的区分能力越强；*Recall* 体现了模型对正样本的识别能力，*Recall* 越高，模型对正样本的识别能力越强。F_1 值是两者的综合，F_1 值越高，说明模型越稳健。所使用的 4 个指标定义如下所示：

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = \frac{precision \times recall \times 2}{precision + recall} \tag{4}$$

TP 为被模型预测为正类样本的正样本数量；*TN* 为被预测为负类样本的负样本的数量；*FP* 为预测为正类的负样本的数量；*FN* 为被预测为负类的正样本的数量。

表 3 给出上述指标在蒙文序数专用测试集上的提取数据定量评价结果，可以看出，本文所制作的数据集和改进型 AlexNet 在精度、召回率、F_1 值都达到了较好的精度，分别为 96.99、95.64、96.20％。

表 3　模型性能测试表

Type	*Precision*	*Recall*	F_1
0	1.000	0.875	0.933
1	0.941	1.000	0.970
2	1.000	1.000	1.000
3	0.938	0.938	0.938
4	0.941	1.000	0.970
5	0.938	0.938	0.938
6	0.941	1.000	0.970
7	1.000	1.000	1.000
8	1.000	0.875	0.933
9	1.000	0.938	0.968

由上述测试结果可以得知，改进后的 AlexNet 模型和专用数据集在训练准确度和测试准确度都达到了实用型水平，为该网络模型进行 FPGA 移植提供了理论依据。

表 4 显示了本文模型使用的片上资源占总资源的比例。占用较多的分别有 LUT、FF 及 BRAM。其中 LUT 用来实现逻辑运算，卷积运算中存在大量的乘加运算，所以 LUT 资源消耗较多。LUTRAM 是将 LUT 用作分布式 RAM，用来存储数据，在模型的数据存储过程中会使用。BRAM 主要用来存放模型的权重数据，所以 BRAM 资源消耗也较多，DSP 共用了 27 个，用于全连接层和池化层的乘累加运算。

表 4　部署神经网络模型资源占比表

逻辑资源	使用数	总数	百分比/％
LUT	11 133	53 200	20.93
FF	2 490	106 400	2.34
BRAM	90	140	64.29
DSP	27	220	12.27
IO	33	125	26.40

模型搭建后，通过行为仿真可以得到模型的总预测时间，将总预测时间和 PC 机端通过 CPU 的结果相对比，如表 5 所示，其中 CPU 运算时间是在 python 环境下执行模型预测函数，得到预测一张图片的时间，FPGA 平台上的运算时间是采用 CNN IP 运算得到的预测时间。可以看到经过 FPGA 计算加速，预测时间提高了 11.04 倍。

表 5　模型预测时间结果对比

平台	COREi3-6400(CPU)	XC7Z020CLG400-2
模型	改进型 AlexNet	改进型 AlexNet
数据集	专用数据集	专用数据集
预测时间/ms	99.293 9	1.429
功耗/W	8.25	0.341

在完成模型测试后，将该神经网络电路关联视频传输电路，完成神经网络的部署，编译工程，生成对应的比特流文件并下载到板载系统中，进行功能测试，测试结果如

图 15 所示。

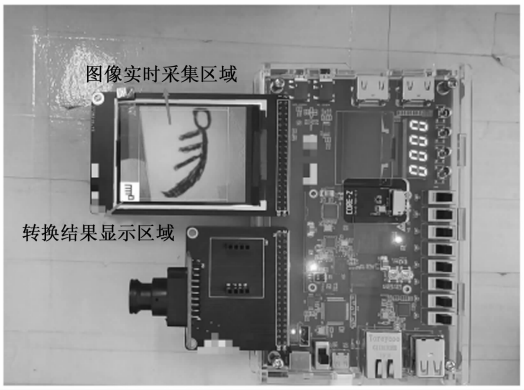

图 15 测试结果图

在系统搭建测试完毕之后，本文对系统准确性进行测试，通过组织志愿者对蒙文手写体数字 0～9 的进行收集，共得到 160 张蒙文手写体图片进行系统测试得到准确率达到 95.62%，测试结果如表 6 所示。

表 6 蒙文识别结果

蒙文数字	样本	识别准确数	正确率/%
0	16	16	100.00
1	16	14	87.50
2	16	16	100.00
3	16	15	93.75
4	16	16	100.00
5	16	16	100.00
6	16	15	93.75
7	16	16	100.00
8	16	14	87.50
9	16	15	93.75

表 7 是本文与其他使用 FPGA 作为硬件平台进行卷积神经网络的部署对比，可以看出本文在功耗指标上有较大优势。尽管在数据吞吐率方面本文低于其他的设备平台，但考虑到实际应用方面，该设备的能量效率是文献［18］的 1.28 倍，是文献［19］的 3.16 倍，是文献［20］的 2.97 倍。表明本文在满足手写体实时字体转换的同时，把设备的能耗尽可能地降低，保证设备在特定场景下的应用成本。

表 7 FPGA 设备平台和其他设备的对比

	Work[18]	Work[19]	Work[20]	Ours
Device	Zynq-XC7Z045	Stratix-VGSD8	Virtex-7VX690	Zynq-XC7Z020
f/MHz	150	120	150	100
Power	9	19.1	126	0.341
Precision	Fixed(16b)	Fixed(8－16b)	Fixed(16b)	Fixed(12b)
Through/GOPs	137	117.8	825.6	6.64
Power efficiency /(GOPS/J)	15.2	6.17	6.55	19.47

3 结束语

本文基于轻量化卷积神经网络和 FPGA 硬件加速器的方法提出了一种蒙文字体转换的方式，提高了字体转换的效率和准确率。通过模型压缩的方式使得 AlexNet 网络模型权重数量减少到 9 460，并将其部署到 FPGA 平台，在视频通路关联该神经网络模块最后结合字体转换库，实现了简单高效的蒙文字体转换。相比于手写和笔画输入方法，该方法结合了高效的 CNN 和 FPGA 硬件加速器的优势，既提高了转换效率又满足了设备成本、功耗和便携性的需求。试验结果表明本系统具有实时识别、准确度高、可移植性强以及低功耗等特点，为其他字体的转换比如汉字等，提供了一定的依据与经验，为系统产品化奠定了基础。同时也注意到本系统仍存在 FPGA 硬件开发难度大，数据集私有程度大等特点，为系统功能进一步提升指明了方向。

参考文献：

[1] 魏延梅. 在自觉、互动和对话中传承民族文化［D］. 北京：中央民族大学，2010.

[2] 苏日娜. 蒙古文字体设计及应用［D］. 内蒙古师范大学，2013.

[3] SUVEERANONT R，IGARASHI T. Example-based automatic font generation［C］//Proc. of International Conference on Smart Graphics. Berlin：Springer-Verlag，2010：127－138.

[4] BALUJA S. BALUJA S. Learning typographic style：from discrimination to synthesis［J］. Machine Vision and Applications，2017，28（5/6）：551－568.

[5] AZADI S，FISHER M，KIM V G，et al. Multi-content GAN for few-shot font style transfer［C］//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2018：7564－7573.

[6] ZHANG X Y，YIN F，ZHANG Y M，et al. Drawing and recognizing Chinese characters with recurrent neural network［J］. IEEE Transactions on Pattern Analysis and Machine Intelligence，2017，40（4）：849－862.

[7] LYU P，BAI X，YAO C，et al. Auto-encoder guided GAN for Chinese calligraphy synthesis［C］//14th IAPR International Conference on Document Analysis and Recognition（2017 ICDAR）. IEEE，2017：1095－1100.

[8] JING Y，YANG Y，FENG Z，et al. Neural style transfer：a review［J］. IEEE Transactions on Visualization and Computer Graphics，2019，26（11）：3365－3385.

[9] ROISYDA S S，PURBOYO T W. A review of various handwriting recognition methods［J］. International Journal of Applied Engineering Research，2018，13（2）：1155－1164.

[10] XIA M，HUANG Z，TIAN L，et al. SparkNoC：an energy-efficiency FPGA-based accelerator using optimized lightweight CNN for edge computing［J］. Journal of Systems Architecture，2021，115：101991.

（下转第 200 页）