引文格式:高凡,路威,甘麟露.基于卷积神经网络的地理空间域计算强度预测与分解方法[J].武汉大学学报(信息科学版),2026,51(5):1048-1058.DOI:10.13203/j.whugis20240119

Citation:GAO Fan,LU Wei,GAN Linlu.A CNN-Based Method for Computational Intensity Prediction and Geospatial Domain Decomposition[J]. Geomatics and Information Science of Wuhan University, 2026, 51(5): 1048-1058. DOI: 10.13203/j. whugis20240119

基于卷积神经网络的地理空间域计算强度预测与分解方法

高 凡[1] 路 威[1] 甘麟露[1]

1 陆军工程大学通信工程学院,江苏 南京,210007

摘 要:地理空间域分解是并行地理计算中实现负载均衡的核心问题,而如何准确提取地理空间域上特征并预测域上计算强度是地理空间域均衡分解的关键。然而,在地理空间域特征提取和计算强度模型建模方面,现有方法过于依赖专家知识,存在特征适用性差、建模过程复杂、模型精度低的问题。为摆脱专家知识依赖,实现高精度、自动化地理空间域计算强度预测,提出一种基于人工智能深度学习的地理空间域计算强度预测和分解方法。所提方法从数据驱动角度出发,将地理空间域对齐卷积神经网络输入结构,利用卷积神经网络自动化捕捉地理空间域特征,预测域上计算强度,并将计算强度预测模型嵌入地理空间域分解,实现任务的均衡划分。以矢量空间相交为例,证明了所提方法在计算性能和自动化程度两个方面的优势。

关键词:并行地理计算;地理空间域;负载均衡;计算强度;深度学习

中图分类号:P237 **文献标识码**:A **收稿日期**:2025-02-25

DOI:10.13203/j.whugis20240119 **文章编号**:1671-8860(2026)05-1048-11

A CNN-Based Method for Computational Intensity Prediction and Geospatial Domain Decomposition

GAO Fan[1] *LU Wei*[1] *GAN Linlu*[1]

1 College of Communication Engineering, Army Engineering University of PLA, Nanjing 210007, China

Abstract: **Objectives**: The evaluation of heterogeneity is essential for domain decomposition in parallel geocomputation. According to the theory of geospatial domain, the assessment of heterogeneity can be transformed to the computational intensity (CI) modelling, thus the key of domain decomposition is feature extraction and computational intensity prediction. However, the existing methods rely on expert knowledge for feature extraction and CI modelling of geospatial domain, suffering from poor feature applicability, complex modelling process and low model accuracy. In order to relieve the dependence on expert knowledge and achieve accurate computational intensity prediction, a computational intensity prediction and decomposition method for geospatial domain based on artificial intelligence (AI) deep learning is proposed. **Methods**: We use convolutional neural network (CNN) to capture the features of geospatial domain automatically, and a fully connected layer is used to predict the computational intensity. An added component is developed to match the geospatial domain and the input of CNN. **Results**: Spatial intersection on vector data is implemented to compare the proposed method and traditional methods. The results demonstrate the advantages of the proposed method in terms of the usability and parallel performance. **Conclusions**: The proposed method optimizes computational intensity prediction and domain decomposition from data science perspective, which provides a reference on how AI deep learning can be used in high-performance geocomputation.

基金项目:国家自然科学基金(42301489);江苏省自然科学基金(BK20231030)。

第一作者:高凡,博士,讲师,主要从事高性能地理计算、地理信息服务方面的研究。fgao_lgd@163.com

通信作者:路威,博士,副教授。luwei_lgd@163.com

Key words: parallel geocomputation; geospatial domain; load balance; computational intensity; deep learning

近年来,智能社会的发展催生了智能计算新兴概念,智能计算是支撑万物互联的数字文明时代新的计算理论方法、架构体系和技术能力的总称。根据具体的实际需求,智能计算以最小的代价完成计算任务,匹配足够的计算能力,调用最好的算法,获得最优的结果[1-2]。并行计算作为智能计算的一项关键技术,已被广泛应用于信息技术领域。按照智能计算范式要求,并行计算需要以最高效的资源利用率、在最短的时间内完成计算任务[3-4]。为了达到这一目标,需要考虑若干问题:(1)算法自身复杂度:如何以最低的时间和空间复杂度来完成串行算法任务;(2)问题的分解:按照问题的分解方法,并行计算策略大体可分为任务并行和数据并行,但无论采用何种分解方法,分解后子问题的数量越大,意味着分解代价、合并代价、通信代价越高;(3)负载均衡:并行计算的性能通常受制于负载失衡现象,在异质性极强的地理空间数据上表现尤为突出。针对该问题,目前研究通过静态调度或动态调度的方式来缓解,但两种方式的效果仍依赖于对分解任务计算强度的评估[5-6]。如在动态调度中,如果最后一项分解任务的计算强度过大,那么整体计算时间将受到该任务影响。本文旨在从负载均衡角度切入,通过引入人工智能深度学习技术,自动化预测地理计算任务的计算强度,保证并行计算过程中的负载均衡,提高并行地理计算性能。

在并行地理计算领域,地理空间域被定义为一组具有不同计算强度的空间计算任务,其中计算强度被用来描述计算单元上承受的负载大小[7]。通过提取地理空间特征,评估地理空间域上并行任务的计算强度并映射到计算单元,可解决由于地理空间异质性所导致的负载失衡问题[8]。文献[9]根据地理空间域的异质性和分布规律,将地理空间域归纳为了规则同质域、不规则同质域、规则异质域和不规则异质域。基于地理空间域理论,并行地理任务的异质性评估被转换为地理空间域上的特征提取和计算强度评估,然而地理空间域上单元计算强度受到诸多因素影响:(1)数据类型:不同数据类型如矢量和栅格会直接影响计算强度的大小;(2)计算依赖:包括本地依赖、邻域依赖、区域依赖和全局依赖,不同依赖代表域上单元计算时,所需要参与计算的周围单元;(3)分解单元:根据分解方法不同可分为序列对象分解单元、网格分解单元、索引块分解单元、聚类块分解单元,不同分解单元的结构也会影响计算强度的大小[10]。除了上述因素外,地理空间域上计算强度的大小还受到计算环境的影响,同一个域上单元在不同算力的计算节点上执行,具有不同的计算强度[11]。

针对地理空间域的特征提取,目前研究可分为基于经验知识和基于机器学习的特征提取方法。前者完全依赖于对特定算法的专家知识,缺少对特征提取的可靠性或重要性评估过程,影响后续计算强度预测模型精度。文献[12]在点云内插数字高程模型算法中,选取了分解单元内点云数量作为唯一特征,并直接用于评估分解单元计算强度;文献[13]在多边形栅格化算法中,选取了多边形顶点数量、形态特征、覆盖像素数量作为特征空间,并同样直接用于计算强度预测模型建模;文献[7]在反距离加权内插算法中,选取了分解单元内已知点数量、待内插点数量、邻域已知点数量作为特征空间,缺少对特征空间可靠性或重要性的评估;文献[14]提出了基于机器学习的特征提取方法,相较于完全依赖经验知识的提取方法,该方法通过机器学习特征选择算法对特征空间进行重要性评估,可剔除无关特征或同类特征,保证特征空间的有效性;文献[15]对比了不同机器学习算法在计算强度评估上的精确性。

基于经验知识或基于机器学习的特征提取方法虽然在特定的地学算法上应用效果较好,但都无法摆脱对专家知识的依赖,需要根据对算法的理解提供一个初始的特征空间,特征空间缺乏通用性,无法应用于其他地学算法。为解决上述问题,本文提出一种基于卷积神经网络(convolutional neural network,CNN)的地理空间域计算强度预测与分解方法。CNN是一种主要用于图片目标识别与分类的深度学习模型,由卷积层、池化层、全连接层构成,其中卷积层负责特征的学习抽象,池化层负责降低参数数量,全连接层用于任务拟合[16]。本文方法主要特点包括:(1)初始的地理空间域特征空间提取不再依赖专家知识,而是通过CNN自行学习;(2)摆脱了特征空间的重要性评估过程,基于多层CNN的特征学习内嵌特征重要性评估;(3)遵循智能计算范式,为如何在并行地理计算领域使用人工智能深度学习技术提供了一个参考框架。

1 基于CNN的地理空间域计算强度预测和分解

1.1 地理空间域计算强度预测

本文提出一种基于CNN自动学习地理空间域的特征并构造计算强度预测模型,方法原理如图1所示。在样本生成方面,分别面向不同地理空间域分解单元类型和地理空间域类型设计了样本构造策略;在计算强度预测模型建模方面,分别比较了多种常见的CNN模型,并从预测精度和预测时间代价方面对模型进行了综合评估。

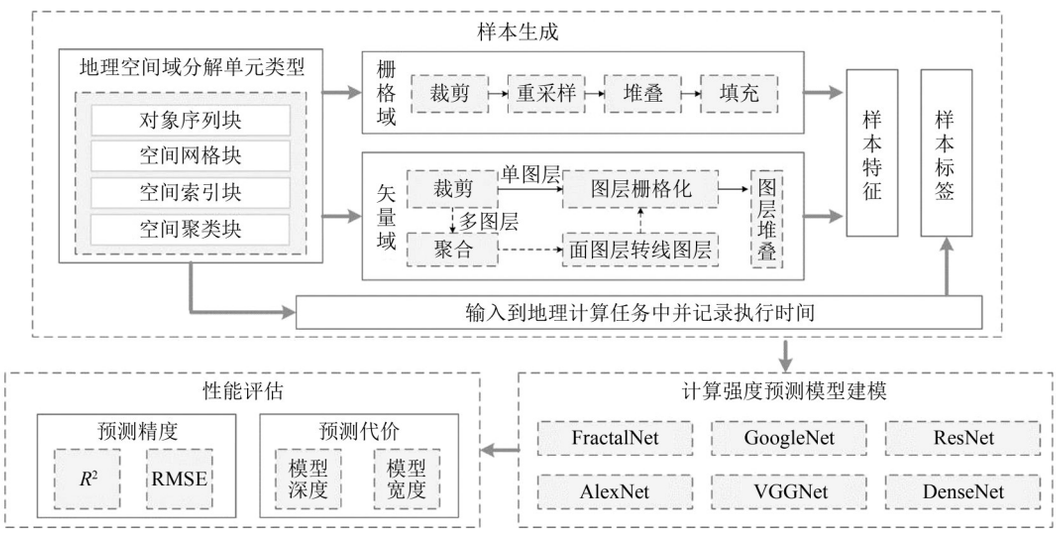


图1 基于卷积神经网络的地理空间域计算强度预测模型建模

Fig. 1 Computational Intensity Prediction of Spatial Domain Based on CNN

1.1.1 面向CNN的地理空间域样本生成

应用CNN提取地理空间域特征以及预测域上分解单元计算强度,需要将地理空间域转换为CNN输入形式,但存在两个问题:(1)域上数据类型。CNN的输入为图片数据,因此可将栅格域分解单元通过简单转换构造为CNN输入样本,但针对矢量地理空间域,则难以构造为CNN输入样本。(2)模型输入大小。CNN的输入大小通常为固定大小,如227×227×3,但地理空间域分解单元的大小和地学算法输入(如二路矢量空间相交的输入为两个图层)以及分解策略(如基于R树划分为R树索引块)有关。

针对上述问题,本文在样本生成模块分别针对不同地理空间域分解单元类型和地理空间域类型设计了样本构造策略。如图1所示,其中地理空间域分解单元类型包括不考虑空间关系直接划分生成的对象序列块、基于规则网格划分生成的空间网格块、基于空间索引划分生成的空间索引块、基于聚类特征划分生成的空间聚类块,地理空间域类型以地学领域基本的栅格和矢量数据类型为主。

针对栅格地理空间域,域上分解单元通常为网格块、索引块和聚类块,具体策略如下:

1)基于规则网格划分时,可直接将网格大小设定为CNN输入的长×宽,执行域分解后直接裁剪,保留原始栅格域上信息,然后将栅格分解单元的图层数量复制并堆叠为CNN输入的深度。

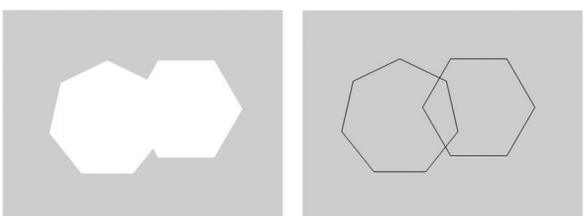
2)基于空间索引或聚类特征划分时,分解单元的形状一般不规则,可通过重采样和填充操作将分解单元转换为CNN输入的长×宽,然后再执行裁剪,并同样通过复制和堆叠操作来匹配CNN输入的深度。

针对矢量地理空间域,域上分解单元可为对象序列块、网格块、索引块和聚类块,在执行裁剪前需要将矢量域分解单元进行栅格化操作,栅格化时可指定分辨率参数为CNN输入的长×宽,同时栅格化操作会自行填充空白区域,然后同样进行堆叠操作即可。然而,矢量域的地学分析通常包含多图层数据,如二路或多路矢量空间相交,如何使得样本同时包含多图层数据信息是面向CNN的矢量域样本构造的关键,具体策略如下:

1)单图层矢量域分析,即以一个矢量图层作为输入(如多边形栅格化),针对该类分析,可按照图1中流程即裁剪、栅格化、图层堆叠操作,得到CNN输入样本。

2)多层矢量域分析,即以多个矢量图层作为

输入(如矢量空间相交),针对该类分析,需要使得样本中包含每个图层的信息。如图1所示,首先将多图层矢量数据执行聚合操作,如果聚合后的图层中只包含点、线数据,则继续按单图层样本构造方式生成样本即可。如果聚合后的图层中包含面数据,需要将面图层转换为线图层后,再执行栅格化操作。图2展示了对面图层和线图层执行栅格化后的结果,其中面图层栅格化后的结果未保留多边形的边缘信息,而线图层栅格化后的结果则保留了多边形的边缘信息,这对后续模型学习不同图层间的空间关系具有重要意义。



(a) 面图层栅格化　　(b) 线图层栅格化

图2　面图层和线图层的栅格化

Fig. 2　Rasterization on Polygon and Polyline

按照上述方法可灵活生成面向不同CNN模型输入的样本,在生成样本的同时,将尚未执行裁剪操作的地理空间域分解单元输入到算法中执行,统计每个单元的执行时间作为样本标签。需要注意的是,本文的实验都是在同构计算环境下执行,异构环境的算力特征尚不在本文研究范围。

1.1.2　地理空间域计算强度预测模型建模

目前已有较多关于CNN模型的研究,可从中选取若干模型进行对比,如AlexNet[17]、VGG-Net[18]、GoogleNet[19]、ResNet[20]、DenseNet[21],但利用神经网络预测计算强度属于并行计算过程中的一个阶段,除了要考虑模型预测精度,还需顾及模型预测代价。模型预测精度决定了后续任务分配是否均衡,而模型预测代价则会影响并行效率。在并行计算过程中,地理空间域划分会产生大量的分解单元,提取每个分解单元的特征并预测其计算强度会带来不可忽略的时间代价,对并行性能整体产生影响。因此在选用CNN架构时,不宜选取层数过深或宽度过大的网络,这里的宽度为特征图的数量,特征图越多,通常意味着计算量越大。AlexNet和ResNet18相较于其他模型,结构较为简单、层数较浅,因此本文实验选用了这两个模型进行实验评估,两个模型的结构如图3所示。由于计算强度预测是回归问题,因此对最终的输出层做了修改,将输出向量调整为只包含1个数值,损失函数采用的均方差(mean squared error,MSE)。为了获得更好的拟合结果,本文将AlexNet的全连接层增至7层,ResNet的全连接层增至4层。

1.2　地理空间域分解

1.2.1　案例分析

矢量空间相交是地学领域最常用的分析之一,兼具数据密集和计算密集特征,并行地理计算领域针对该分析提出了诸多并行策略[22-24]。其中基于分区的空间合并和关联(partition based spatial-merge join,PBSM)是目前应用较为广泛的方法[22],该方法将矢量空间相交分析拆分为了过滤和精解两个阶段,如图4所示。过滤阶段将矢量地理空间域分解成若干子域,常用的分解策略包括规则网格划分、四叉树划分、R树索引划分。以网格划分为例,过滤阶段首先判断每个图层内多边形和网格的空间关系,如果和某个网格相交,则将多边形分配到该网格,一个多边形可能会被分配到多个网格内。精解阶段以网格为单位,对每个网格内的多边形对进行空间求交。精解阶段避免了每个多边形都需要和另一图层的所有多边形求交,只需要和网格内的另一图层的多边形求交,大大降低了空间求交的代价。

过滤阶段和精解阶段为矢量空间相交的并行化提供了基础。在过滤阶段,可将网格信息发送到各个子节点,各个子节点负责判读一部分多边形和网格的空间相交关系,最后由主节点进行信息汇聚。在精解阶段,为每个子节点分配一定数量的网格,各子节点只需对分配的网格内的多边形求交,最后将求交结果发送给主节点进行汇聚输出。PBSM方法给矢量空间相交并行化提供了基本思路,但在过滤阶段和精解阶段存在两个主要问题:

1)过滤阶段,计算密集部分集中在各计算节点判断多边形和网格的空间相交关系,那么应该给每个节点分配多少多边形才能保证各节点同时判断完毕?

2)精解阶段,计算密集部分集中在各计算节点对网格内的多边形进行空间求交,那么应该给各个节点分配多少网格才能保证各节点同时执行完毕?

针对问题1),需要考虑判断多边形和网格空间相交关系的计算强度受到哪些特征影响。针对问题2),需要考虑一个网格内多边形求交的计

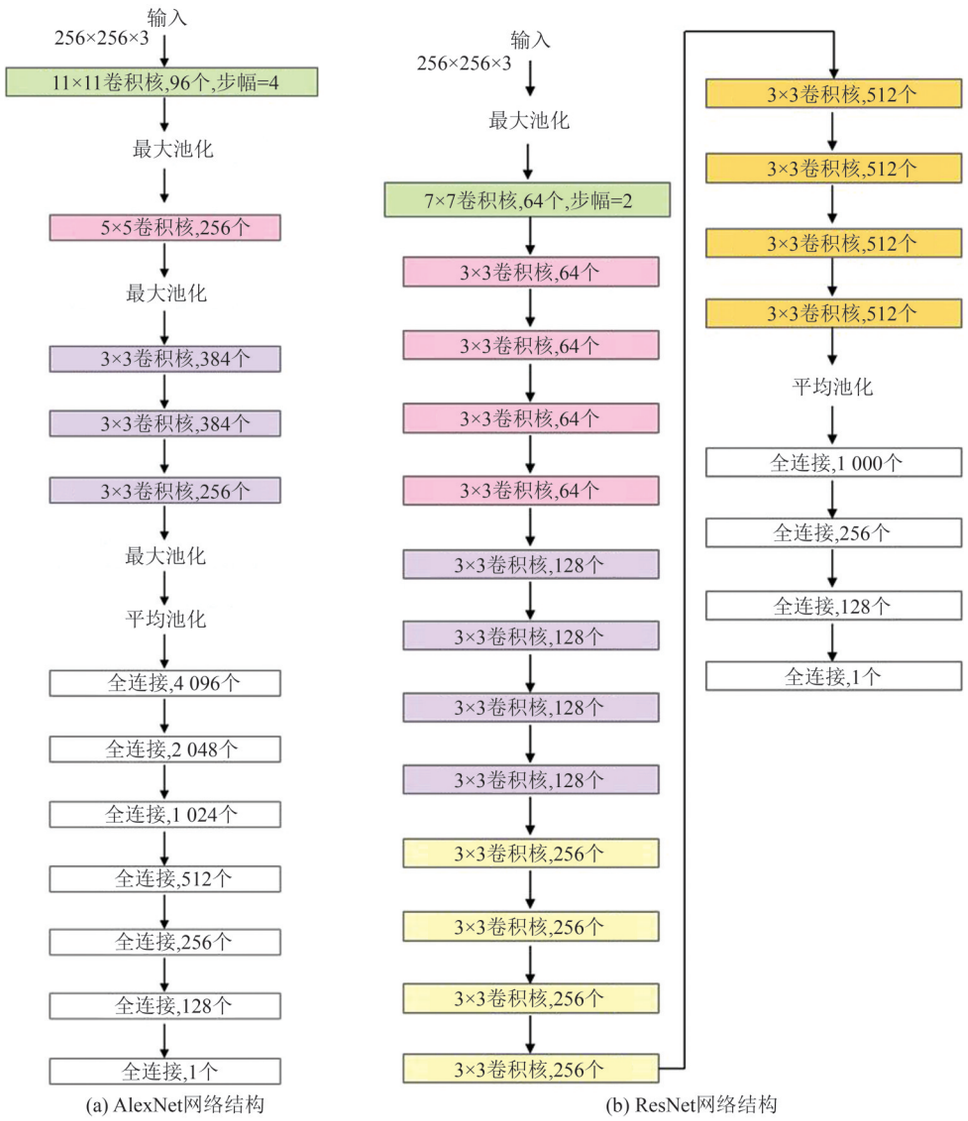


图3 计算强度预测建模所用网络结构

Fig. 3 CNN Structure for Computational Intensity Prediction Modelling

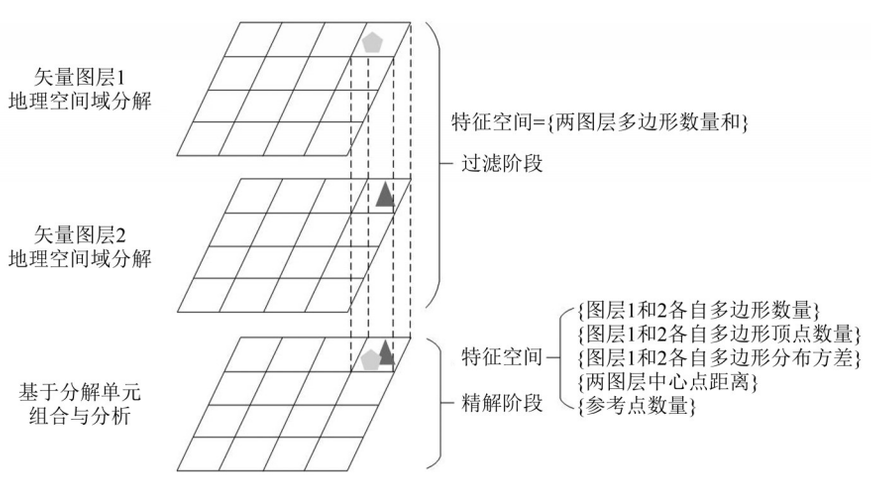


图4 PBSM方法以及地理空间域特征分析

Fig. 4 PBSM Method and Feature Analysis of Geospatial Domain

算强度受到哪些特征影响。目前已有相关研究基于专家知识针对各阶段推荐了诸多候选特征,笔者在前期研究[10]中,已通过机器学习特征选择技术对各候选特征进行了评估。结果证明,过滤阶段是一个弱计算密集阶段,可以用多边形的数量作为计算强度评估指标,即过滤阶段中每个分

解单元的特征空间仅由多边形数量特征组成。精解阶段是一个强计算密集阶段,该阶段每个分解单元的特征空间包括网格内多边形数量、多边形顶点数量、多边形分布方差、两个图层中心点距离、参考点数量。其中,多边形分布方差是根据多边形中心点坐标计算的二维分布方差,两个图层中心点距离是两个图层多边形中心点坐标均值的差值,参考点机制是用以避免一个多边形可能被分配到多个网格而导致重复求交的一种策略[25],网格内参考点数量是一个网格内需要实际求交的多边形对数。为了精确评估精解阶段的计算强度,传统基于经验知识的策略需要根据严密的专家知识来提取特征空间,而本文提出的CNN优化方法则降低了对专家知识的依赖程度,后续将对基于传统机器学习的计算强度评估策略和基于深度学习的优化策略进行对比分析。

1.2.2　CNN优化的地理空间域分解

以矢量空间相交为例,采用空间网格分解策略,基于CNN优化PBSM方法流程如图5所示。原生PBSM策略只包含过滤和精解两个阶段,基于CNN优化后的PBSM在两个阶段之间额外增加一个计算强度预测阶段,用于预测精解阶段每个网格的计算强度。3个阶段中,过滤阶段是一个弱计算密集的阶段,该阶段的计算强度主要受到多边形的数量影响,可基于多边形数量分配任务。计算强度预测阶段是一个相对计算密集的阶段,该阶段涉及栅格化等步骤,因此也需要进行并行化。但相较于精解阶段,该阶段可直接以多边形顶点数量作为计算强度并进行任务分配,即可基本实现负载均衡,无需使用CNN进行计算强度预测。精解阶段是PBSM策略中计算最为密集的阶段,需要根据上一阶段预测的计算强度进行任务分配,否则会出现严重的负载失衡问题。因此,优化后的PBSM策略是基于经验知识和CNN混合优化后的结果。

1)过滤阶段:主进程获取矢量数据的空间范围,并发送给各子进程。各子进程基于空间网格分解策略划分矢量空间域,主进程同步将矢量数据分解为等同于子进程数量的若干组,每组包含相同数量的一系列矢量ID,随后将分组分别发送到对应的子进程。各子进程接收到矢量ID后,读取矢量数据并判断和网格的空间关系,若相交,则将矢量ID绑定到该网格,同时统计该网格内多边形顶点数量。相交关系判断完毕后,各子进程将绑定一系列矢量ID的网格发送给主进程。

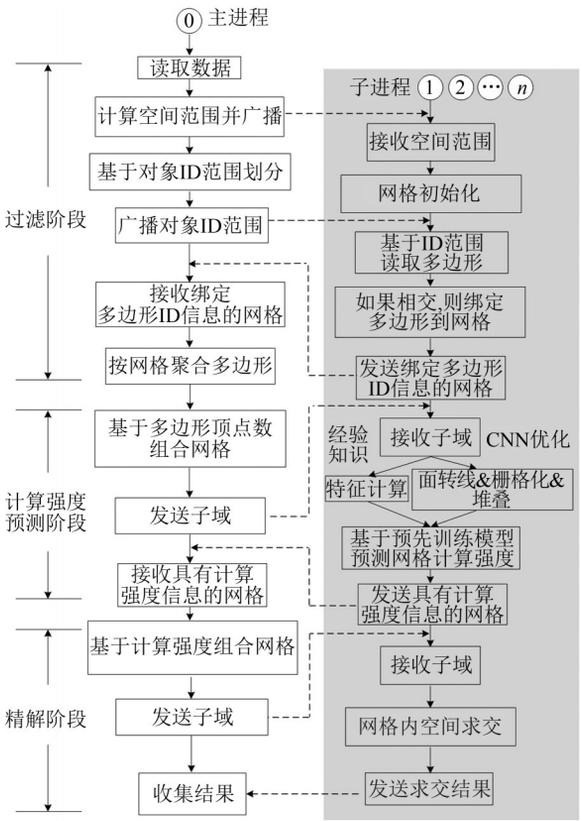


图5　基于卷积神经网络优化的PBSM方法

Fig. 5　Optimized PBSM Method Based on CNN

2)计算强度预测阶段:主进程接收各子进程发送的绑定了矢量ID的网格,并基于网格对矢量ID进行合并,每个网格包含了所有与其存在空间相交关系的矢量ID。首先,主进程获取每个网格内矢量的顶点数量,并基于顶点数量将网格均匀分配给各子进程。然后,各子进程并行地对网格内矢量数据进行合并、栅格化、堆叠操作,并将结果输入到提前训练完毕的CNN,预测精解阶段各网格的计算强度。图5还提供了基于经验知识预测计算强度的过程,即需要计算特征向量,然后输入到提前构造的计算强度预测模型,如多项式模型或机器学习模型。最后,各子进程将带有计算强度信息的网格发送给主进程。

3)精解阶段:主进程基于计算强度将网格均匀地分配给各子进程,各子进程读取网格内矢量数据,并行地对网格内矢量数据进行空间求交操作,并将结果发送给主进程。最后,主进程对求交结果进行合并和输出。

2　性能评估

2.1　实验数据和环境

本文以多边形空间相交为例,使用某省土地

利用图斑数据进行方法验证,实验使用的3组多边形数据集见表1,其中每个数据集的两个图层分别表示地理图斑和坡度图斑,应用需求为通过对两者求交,将与地理图斑相交面积最大的坡度图斑的坡度信息赋予地理图斑。实验环境由两台浪潮NF5280M4服务器组成,每台服务器配有28个CPU核心以及125 GB内存,操作系统为CentOS Linux Release 7.3.1611,服务器统一连接一组由84个16 TB的存储盘组成的存储阵列。软件采用了MPI(message passing interface)用于节点间通信,GEOS(geometry engine,open source)用于处理矢量数据,PyTorch用于CNN训练。

表1 实验使用的3组多边形数据集

Table 1 Three Datasets Used in Experiment

对比项	数据集1		数据集2		数据集3	
	图层1	图层2	图层1	图层2	图层1	图层2
描述	地理图斑	坡度图斑	地理图斑	坡度图斑	地理图斑	坡度图斑
数据量/GB	0.27	0.17	0.45	0.32	2.01	1.03
数量/个	188 449	184 038	312 989	296 986	1 138 641	926 294
经纬度范围	23.17°N~24.80°N,114.24°E~115.59°E		23.17°N~24.80°N,114.24°E~115.59°E		21.46°N~25.19°N,111.05°E~117.19°E	

2.2 模型精度评价

本文分别对比了AlexNet、ResNet和传统机器学习模型的精度,其中传统机器学习模型以§1.2.1中分析的7个候选特征为输入,包括网格内两个图层多边形数量、两个图层多边形顶点数量、多边形分布方差、两个图层中心点距离、参考点数量。样本方面,分别从3组数据集中生成约20 000个样本,样本图片是通过对3组数据进行网格划分、聚合、栅格化生成,其中网格划分是将与网格存在空间相交关系的多边形分配到该网格,并不对多边形自身进行切分,样本标签为精解阶段的执行时间。图6为AlexNet和ResNet模型训练精度,本文采用了拟合度R^2和MSE损失作为描述模型的性能指标。结果表明,AlexNet模型训练集和测试集的R^2分别收敛于0.61和0.60左右,训练集和测试集的MSE分别收敛于2.57 s^2和3.78 s^2左右。ResNet模型训练集和测试集的拟合度都收敛于0.98左右,训练集和测试集的MSE收敛于0.13 s^2和0.04 s^2左右。结果表明,ResNet模型的预测精度比AlexNet更好。

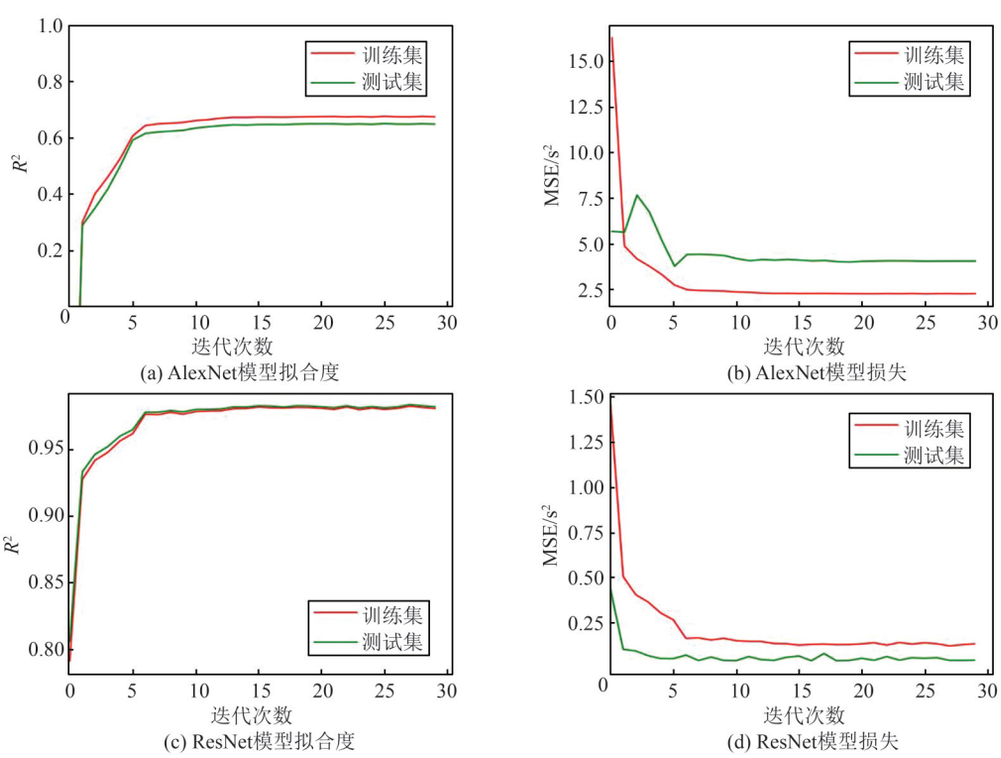


图6 AlexNet和ResNet模型精度对比

Fig. 6 Accuracy Comparison of AlexNet and ResNet

本文和前述工作即基于机器学习预测地理空间域计算强度的方法进行了对比[10],先利用机器学习特征选择技术对根据经验知识提取的7个候选特征进行精确评估,剔除无关特征后再利用机器学习回归算法进行模型建模。图7展示了各类机器学习特征选择算法和机器学习回归算法结合的结果,其中不同机器学习特征选择算法会剔除不同的特征,本文选取了6种特征选择算法和4种机器学习回归算法,特征选择算法包括全特征空间(all features,AF)、回归消除(regression relieff,RReliefF)、平均精度减少(mean decrease accuracy,MDA)、平均不纯度减少(mean decrease impurity,MDI)、递归特征消除(recursive feature elimination,RFE)、包裹式遗传算法(genetic wrapper algorithm,GWA),机器学习回归算法包括分类回归树(classification and regression tree,CART)、随机森林(random forest,RF)、梯度上升回归(gradient boost regression,GBR)、支持向量回归(support vector regression,SVR)。从结果来看,RF和RReliefF、MDI、MDA、RFE的组合获得了最高的精度,R^2达到0.84,平均绝对值误差(mean absolute error,MAE)达到190.00 ms,4个机器学习特征选择算法都从两个图层网格内各自多边形数量、各自多边形顶点数量、多边形分布方差、两个图层中心点距离、参考点数量中剔除了两个图层中心点距离这一特征。

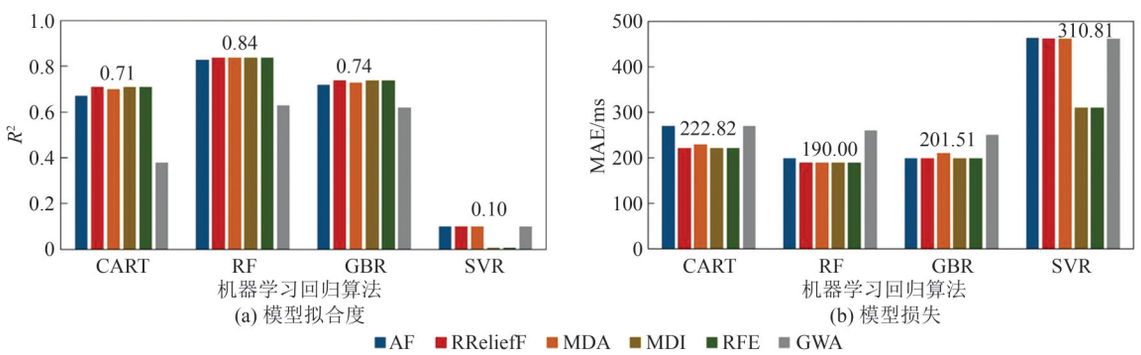


图7　不同机器学习模型精度

Fig. 7　Accuracy Comparison of Different Machine Learning Models

2.3　并行计算性能评价

本文对比了5种计算强度评估方法的并行计算性能,评估指标包括负载均衡性能、加速比和并行效率。第一种方法忽略计算强度(ignoring computational intensity,ICI),即在精解阶段的任务分配过程中,假设每个网格的计算强度一致,按照网格数量均匀分配;第二种方法直接将特征作为计算强度(feature as computational intensity,FCI),即将单个特征视为计算强度,如网格内多边形数量,然后根据该计算强度均匀分配网格;第三种方法是一种基于机器学习RF预测计算强度的方法(random forest to predict computational intensity,RFCI),该方法基于人工提取的特征空间,结合机器学习特征选择和RF评估计算强度,然后基于计算强度均匀分配网格;第四种和第五种方法则分别是本文提出的基于AlexNet预测计算强度(AlexNet to predict computational intensity,ACI)和基于ResNet预测计算强度(ResNet to predict computational intensity,RCI)的方法,即通过CNN自动学习特征后预测计算强度。

实验对比了5种方法在16进程和24进程下的负载均衡性能,如图8所示。结果表明,RCI和RFCI中各进程上的负载最为均衡,而ICI则负载失衡现象较为严重。值得注意的是,ACI的负载均衡性能较FCI差,究其原因在于AlexNet模型对于地理空间域的特征学习能力较差,计算强度预测精度较低。

为进一步评估5种方法的负载均衡性能,实验对比了在16进程和24进程下各进程执行时间的标准差,如图9所示。结果表明,RFCI的进程执行时间标准差最低,在16进程和24进程下分别为5.7 s和4.6 s,RCI的进程执行时间标准差次之,分别为6.8 s和7.2 s。整体来讲,RFCI与RCI方法的性能相当,RFCI得益于经验知识下特征提取的准确性,而RCI中地理空间域特征完全由CNN自动学习,自动化程度较高。

实验对比了5种方法的串行执行时间以及在4、8、12、16、24进程下的并行执行时间、加速比和并行效率,结果分别如图10、图11所示。从上述结果可以看出,在执行时间方面,RCI和RFCI的并行执行时间在任意进程数下都最短;在加速比方面,RCI和RFCI在4、8、12、16、24进程数下均

全面优于其他方法,ICI在任意进程数下加速比都最低;在并行效率方面,RCI和RFCI虽然随着进程数量的增多,并行效率有所下降,但都保持在0.7以上,且全面优于其他方法。值得注意的是,RFCI在加速比和并行效率上均略优于RCI,一方面得益于RF较高的计算强度预测精度,另一方面则由于采用CNN预测计算强度,增设了额外的栅格化步骤,虽然本文对栅格化步骤采用了并行化方式进行优化,但仍然会影响RCI方法的整体性能。

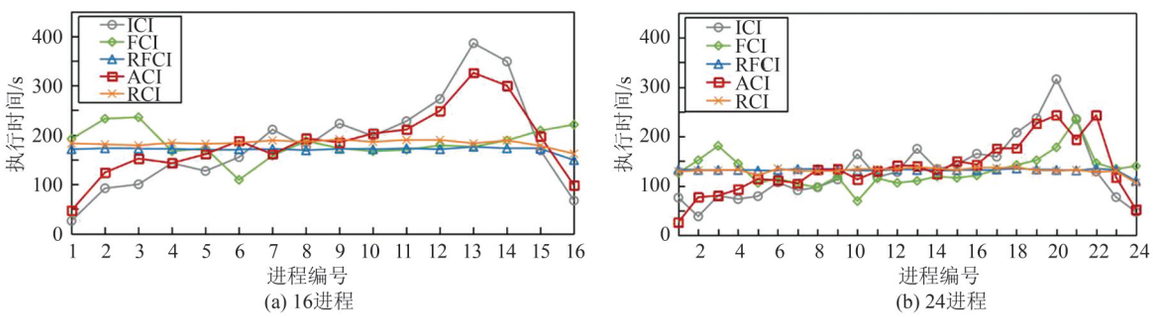


图8 5种方法在不同进程下的负载均衡性能

Fig. 8 Load Balance Performance of Different Methods Under 16 and 24 Processes

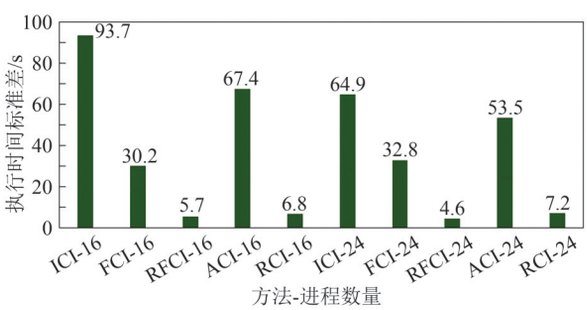


图9 5种方法在不同进程下的进程执行时间标准差

Fig. 9 Standard Deviations of Execution Times of Different Methods Under 16 and 24 Processes

为优化并行矢量空间相交,本文在过滤阶段和精解阶段的基础上,增加了计算强度评估阶段,为评估计算强度预测对整体性能的影响程度,本文分析了不同方法每个主要阶段的计算时间,结果如图12所示。FCI的计算强度预测只涉及到单个特征计算,在16进程和24进程中分别占比0.9%和0.4%;RFCI的计算强度涉及到若干特征计算和模型预测,在16进程和24进程中分别占比9.1%和10.5%;ACI的计算强度预测之前需要进行栅格化操作,加上特征提取和模型预测时间,在16进程和24进程中分别占比7.3%和6.0%;RCI的计算强度预测之前同样需要进行栅格化操作,加上特征提取和模型预测时间,在16进程和24进程中分别占比12.4%和10.2%。综上分析可知,相较于RFCI,RCI中计算强度预测导致的额外时间占比在16进程下相对较高,而在24进程下相对较低,这与图11中两种方法在16进程和24进程下的加速比和并行效率结果一致。

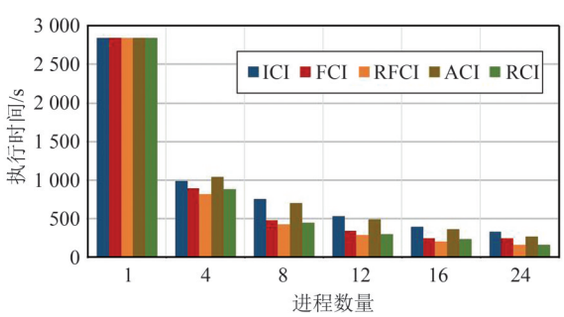


图10 5种方法的执行时间

Fig. 10 Execution Time of Different Methods

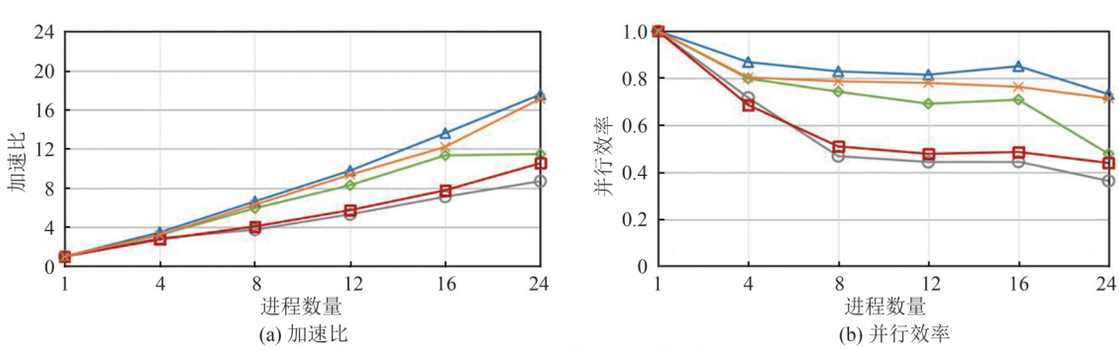


图11 5种方法加速比和并行效率

Fig. 11 Speedup Ratio and Parallel Efficiency of Different Methods

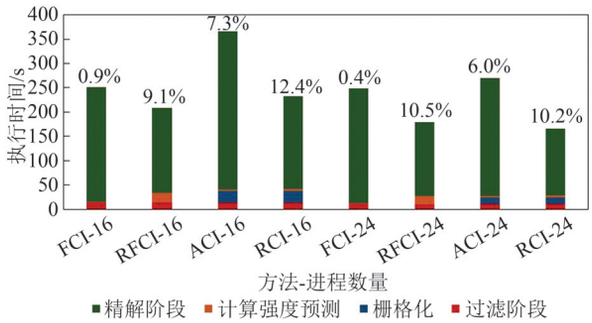


图 12　不同方法各阶段时间占比

Fig. 12　Time Cost of Each Stage in Different Methods

3　结　语

针对传统方法在地理空间域计算强度预测上依赖专家知识,存在特征适用性差、建模过程复杂、模型精度低的问题,本文提出了一种基于CNN的地理空间域计算强度预测方法。所提方法考虑了地学分析中的数据类型、分解粒度以及多层输入,将地理空间域转换为适应CNN结构的形式,并通过多层卷积网络自动学习地理空间域特征,使用全连接层预测计算强度。本文基于矢量空间相交案例,对比了5种方法的性能,包括在云计算平台中常见的忽略计算强度的方法、传统基于单特征评估计算强度的方法、基于机器学习RF评估计算强度的方法、基于深度学习AlexNet评估计算强度的方法和基于ResNet评估计算强度的方法。结果表明,基于ResNet评估计算强度的方法和基于RF评估计算强度的方法实现了相当的性能,但本文提出的基于ResNet评估计算强度的方法不依赖经验知识提取特征,明显提升了并行地理计算的自动化程度。本文研究是对前述基于机器学习优化并行地理计算研究[10,14]的自然深化,在智能地理计算范式下,进一步引入了深度学习到并行地理计算领域,提供了一种如何将深度学习技术应用到地理计算中的思路,有助于推动AI GIS的进一步发展。

参　考　文　献

[1] ZHU S Q, YU T, XU T, et al. Intelligent Computing: The Latest Advances, Challenges, and Future[J]. *Intelligent Computing*, 2023, 2: 6.

[2] 李德仁. 论时空大数据的智能处理与服务[J]. 地球信息科学学报, 2019, 21(12): 1825-1831.
LI Deren. The Intelligent Processing and Service of Spatiotemporal Big Data[J]. *Journal of Geo-Information Science*, 2019, 21(12): 1825-1831.

[3] ZHOU C, CHEN Z J, LI M C. A Parallel Method to Accelerate Spatial Operations Involving Polygon Intersections [J]. *International Journal of Geographical Information Science*, 2018, 32 (12): 2402-2426.

[4] ZHOU C, LI M C. A Systematic Parallel Strategy for Generating Contours from Large-Scale DEM Data Using Collaborative CPUs and GPUs[J]. *Cartography and Geographic Information Science*, 2021, 48(3): 187-209.

[5] WANG Y J, AI B B, QIN C Z, et al. A Load-Balancing Strategy for Data Domain Decomposition in Parallel Programming Libraries of Raster-Based Geocomputation[J]. *International Journal of Geographical Information Science*, 2022, 36 (5): 968-991.

[6] QIN C, WANG X B, ZHAO N. EMFEM: A Parallel 3D Modeling Code for Frequency-Domain Electromagnetic Method Using Goal-Oriented Adaptive Finite Element Method[J]. *Computers & Geosciences*, 2023, 178: 105403.

[7] WANG S W, ARMSTRONG M P. A Theoretical Approach to the Use of Cyberinfrastructure in Geographical Analysis[J]. *International Journal of Geographical Information Science*, 2009, 23 (2): 169-193.

[8] 亢晓琛, 刘纪平. 图划分支持下的大规模点要素并行缓冲分析方法[J]. 武汉大学学报(信息科学版), 2023, 48(6): 979-987.
KANG Xiaochen, LIU Jiping. Parallel Buffer Analysis of Large Scale Point Features Based on Graph Partitioning[J]. *Geomatics and Information Science of Wuhan University*, 2023, 48 (6): 979-987.

[9] ARMSTRONG M P, DENSHAM P J. Domain Decomposition for Parallel Processing of Spatial Problems[J]. *Computers, Environment and Urban Systems*, 1992, 16(6): 497-513.

[10] YUE P, GAO F, SHANGGUAN B Y, et al. A Machine Learning Approach for Predicting Computational Intensity and Domain Decomposition in Parallel Geoprocessing[J]. *International Journal of Geographical Information Science*, 2020, 34 (11): 2243-2274.

[11] 王鸿琰, 关雪峰, 吴华意. 一种面向CPU/GPU异构环境的协同并行空间插值算法[J]. 武汉大学学报(信息科学版), 2017, 42(12): 1688-1695.
WANG Hongyan, GUAN Xuefeng, WU Huayi. A Collaborative Parallel Spatial Interpolation Algorithmon Oriented Towards the Heterogeneous CPU/GPU System[J]. *Geomatics and Information*

Science of Wuhan University, 2017, 42(12): 1688-1695.

[12] REN Y B, CHEN Z J, CHEN G, et al. A Hybrid Process/Thread Parallel Algorithm for Generating DEM from LiDAR Points[J]. *ISPRS International Journal of Geo-Information*, 2017, 6(10): 300.

[13] ZHOU C, CHEN Z J, LIU Y X, et al. Data Decomposition Method for Parallel Polygon Rasterization Considering Load Balancing[J]. *Computers & Geosciences*, 2015, 85: 196-209.

[14] GAO F, YUE P, CAO Z P, et al. A Multi-source Spatiotemporal Data Cube for Large-Scale Geospatial Analysis[J]. *International Journal of Geographical Information Science*, 2022, 36(9): 1853-1884.

[15] HERNÁNDEZ Á B, PEREZ M S, GUPTA S, et al. Using Machine Learning to Optimize Parallelism in Big Data Applications[J]. *Future Generation Computer Systems*, 2018, 86: 1076-1092.

[16] ALZUBAIDI L, ZHANG J L, HUMAIDI A J, et al. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions[J]. *Journal of Big Data*, 2021, 8(1): 53.

[17] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks[J]. *Communications of the ACM*, 2017, 60(6): 84-90.

[18] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[EB/OL]. [2014-09-15]. https://arxiv.org/abs/1409.1556

[19] SZEGEDY C, LIU W, JIA Y Q, et al. Going Deeper with Convolutions[C]//IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, 2015.

[20] HE K M, ZHANG X Y, REN S Q, et al. Deep Residual Learning for Image Recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016.

[21] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely Connected Convolutional Networks[C]//IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 2017.

[22] PATEL J M, DEWITT D J. Partition Based Spatial-Merge Join[J]. *ACM SIGMOD Record*, 1996, 25(2): 259-270.

[23] AJI A, WANG F S, VO H, et al. Hadoop GIS: A High Performance Spatial Data Warehousing System over Mapreduce[J]. *Proceedings of the VLDB Endowment*, 2013, 6(11): 1009-1020.

[24] DU Z H, ZHAO X W, YE X Y, et al. An Effective High-Performance Multiway Spatial Join Algorithm with Spark[J]. *ISPRS International Journal of Geo-Information*, 2017, 6(4): 96.

[25] DITTRICH J P, SEEGER B. Data Redundancy and Duplicate Detection in Spatial Join Processing[C]//The 16th International Conference on Data Engineering, San Diego, USA, 2002.