

基于容器云技术的典型 遥感智能解译算法集成

赵智韬^{1,2}, 赵理君¹, 张正¹, 唐婷¹

1. 中国科学院空天信息创新研究院, 北京 100094;
2. 中国科学院大学资源与环境学院, 北京 100049

摘要

针对当前航天遥感信息处理向云计算发展的趋势, 将容器云技术应用于遥感数据的智能处理, 通过在计算集群中部署遥感解译算法镜像与分布式存储服务, 屏蔽复杂的环境依赖问题, 并通过配置文件进行流程管理, 形成了从开发到部署的整体技术路线, 为遥感智能解译技术的集成提供了高效可靠的新思路。以几种典型的遥感智能解译算法为例, 证明了该方案在智能解译模型的集成化开发部署上的高效性, 为遥感智能解译技术的新型云端模式探索了可行性方案。

关键词

人工智能; 深度学习; 遥感图像; 智能解译; 容器云技术

中图分类号: TP75

文献标志码: A

doi: 10.11959/j.issn.2096-0271.2022015

Integration of remote sensing intelligent processing algorithm using container cloud technology

ZHAO Zhitao^{1,2}, ZHAO Lijun¹, ZHANG Zheng¹, TANG Ping¹

1. Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
2. College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract

With the development of cloud computing technology in remote sensing data processing, container cloud technology was applied to remote sensing data processing, remote sensing intelligent processing algorithm image was deployed and storage services were distributed in the computing cluster, shielding complex environmental dependence problems. And management through configuration files was processed forming an overall technical route from development to deployment, which provides an efficient and reliable new scheme for the integration of remote sensing intelligent processing algorithms. Taking several typical remote sensing intelligent algorithms as examples, the efficiency of the scheme in the integrated development and deployment was proved, and the feasibility of the new cloud mode of remote sensing intelligent technology was explored.

Key words

artificial intelligence, deep learning, remote sensing image, intelligent processing, container cloud technology

0 引言

随着空天科学技术的发展和对地观测系统的持续运行,人们获得遥感图像数据变得越发简单,形成了多源多尺度遥感数据并存的局面,遥感数据的增长速度逐年提高^[1]。对于遥感智能解译算法的集成而言,海量、多源、异构的遥感数据不仅造成了数据集读取标准不统一、算法模型训练方法不一致等困难,而且导致现有的遥感数据产品生产与管理方式无法满足快速、高可靠性等服务需求。随着高分辨率对地观测卫星时代的到来,从各种途径获取的遥感图像数据的量正呈指数级增长。根据

开放地理空间信息联盟(Open Geospatial Consortium, OGC)预计,当前国家级遥感图像数据存档量将很快达到数PB之巨,而全球遥感图像数据量将达到EB级^[2]。在遥感大数据环境下,遥感产品生产涉及的算法种类和数据规模都是空前的^[3],部分产品的设计和生产还需要考虑多源多尺度数据处理的顺序层次关系,这对遥感算法的集成和协同都提出了更高的要求。然而现有的遥感智能解译算法系统生产的产品种类单一、生产流程固定,无法满足用户复杂多变、个性化、大规模的产品生产需求。遥感智能解译的基本流程如图1所示。

此外,以往遥感领域的深度学习算法往往是基于单一的深度学习环境开发的,不同环境之间兼容性低,算法集成

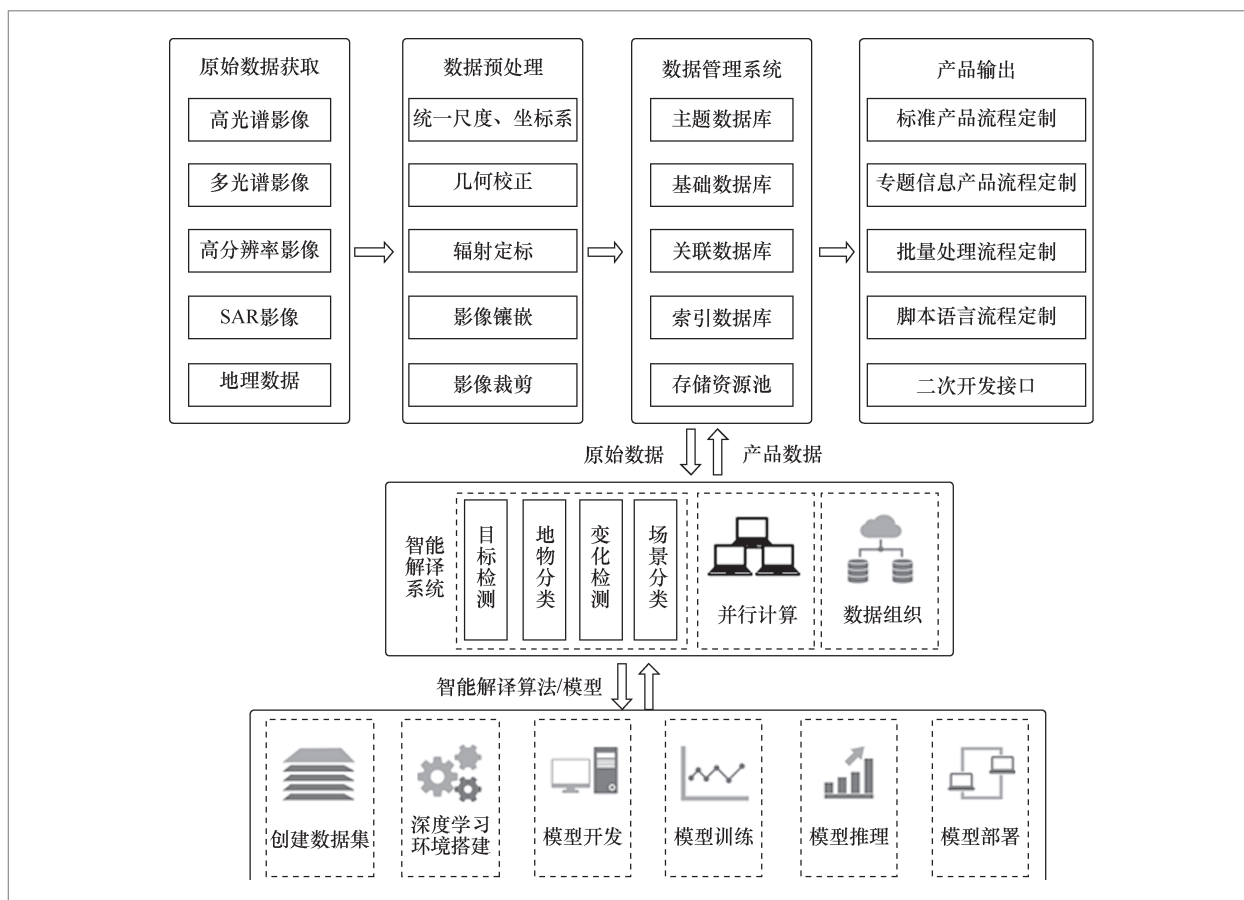


图1 遥感智能解译的基本流程

难度大。如何使深度学习代码无须重写,一次编写、到处运行(write once, run everywhere),具备良好的可移植性和复用性,成为一大挑战。传统的深度学习流程中算法在不同机器间进行环境迁移非常复杂,基于Conda包管理技术解决上述问题并不是一个高效的方法。传统的虚拟化技术使用KVM(kernel-based virtual machine)等虚拟化技术,理论上可以支持多种模型的部署和运行。然而遥感智能解译算法通常依赖于深度学习框架和NVIDIA推出的CUDA(compute unified device architecture)通用并行计算架构,需要使用物理GPU进行大规模的并行计算,而KVM虚拟机想要使用物理机的显卡,需要用到复杂的GPU透传(GPU passthrough)技术,其搭建流程烦琐且低效,无法满足遥感智能算法的开发与集成需求。而容器技术可以将不同深度学习算法集成到同一物理机上,与传统KVM等虚拟化技术相比,其具有小粒度、可拓展、灵活性和支持显卡计算等优点,为大量深度学习算法的环境隔离和资源隔离提供了可能,提高了算法开发训练和部署的灵活性。

因此,本文借助容器云技术的分布式数据存储、高性能计算、弹性扩展、按需服务等优点,对典型的遥感智能解译算法及产品生产中的关键技术展开研究,具体如下。

- 针对多源遥感数据集管理问题,引入GlusterFS分布式文件系统,该系统支持在线扩容,支持远程直接访问内存,支持跨集群云存储,方便遥感智能解译算法所需数据的索引查找、数据缓存与集成,从数据集的角度提升智能解译算法训练的速度可以大大减轻数据集管理和存储的负担,避免大型数据集的迁移工作。

- 针对遥感智能解译算法环境搭建的问题,基于Docker容器技术复现深度学习环境,缩短重复搭建环境的时间,且从集

成规范化的角度,提供具备统一开发功能模块的基础镜像,提供统一的模型参数接口约定和配置文件模板,满足多用户对算法环境的多样化需求。

- 针对遥感智能解译算法调优,首先通过建立模型统一可视化的框架,其次基于Kubernetes容器云技术,统一分配物理资源,实现显卡计算资源的管理自动化、利用率最大化,提高模型调优过程的效率。相较于自然图像数据,遥感数据的构成更加复杂,不同遥感数据源由于其卫星、传感器物理参数等的不一致,造成了在空间分辨率、时间分辨率、波谱范围、投影标准、分幅标准等方面的差异^[4]。因此,遥感智能解译算法在开发调试阶段需要更加频繁地调整算法和调试参数,并且需要将多源异构的数据处理的不同结果进行可视化研究。构建一个高可用、高可视化、高自动化的调优方案可以简化整个遥感智能解译流程,使开发人员能专注于算法本身。

- 针对遥感智能解译产品的生产问题,首先通过对遥感数据各级产品的层级关系和遥感智能解译算法的顺序关系进行梳理,构建遥感产品上下层级关系^[5],组织合理高效的遥感产品生产逻辑流程。之后引入Job引擎,动态生成多源遥感产品生产 workflow,实现多源、海量遥感数据的智能处理及产品生产。

- 针对遥感智能解译算法的集成部署问题,使用Docker容器,统一集成遥感智能解译算法环境。处理遥感数据的过程需要读写各种专门的遥感数据格式,并且涉及投影转换、多波段、空间信息、几何校正等特殊需求,因此解译算法依赖于多种遥感数据处理库和基于显卡加速的处理算法,不同算法难以集成。而Docker容器秒级启动和环境隔离的优点可以解决遥感智能解译算法在集成部署环节中部署慢、集成难的问题。

开展上述研究的意义在于,首先,从数

据集与模型交互的环节屏蔽了多源、异构遥感数据的元数据差异,统一了模型获取数据的方式,为多算法集成提供了基础;其次,通过调度与编排容器,按序执行容器内的算法,保证了复杂解译产品生产任务的顺利执行,提高了遥感智能解译算法之间的组织性;最后,本研究通过在几种典型的遥感智能解译算法上进行实验表明,使用容器云技术构建的GPU计算集群运行神经网络算法可有效地提高计算资源的利用率,提高训练效率,同时极大地提高部署的灵活性,为遥感智能解译技术的新型云端模式探索可行性方案。

1 技术背景

1.1 深度学习框架

深度学习框架是一种工具库,它使人们在无须深入了解底层算法细节的情况下,能够更容易、更快速地构建深度学习模型。深度学习框架利用预先构建和优化好的组件集合定义模型,为模型的实现提供了一种清晰而简洁的方法。

TensorFlow、PyTorch和Caffe等优秀的深度学习框架完全开源,并且有多个活跃社区可解答相关问题,同时提供用于训练、测试等环节的完整工具包,可以帮助使用者快速上手。这些优秀且对用户友好的深度学习框架从技术和成本上降低了人工智能的门槛,提升了国内中小企业以及个人开发者的效率。

1.2 遥感智能解译技术

遥感是指使用传感器通过远距离的、不与探测对象直接接触的探测方式,收集

探测对象的电磁辐射信息,并且分析探测到的数据,得到探测对象信息的技术。快速发展的高分遥感系统已成为各个国家的重要军事资源和必备设施。作为比较简单易携的航空遥感手段,无人机遥感技术在国内外国外都已得到普遍使用,其被广泛应用于国家军事水陆空的检测、天气预测、农业林业病虫害检测、环境破坏检测、城市土地资源监测等领域。遥感技术在地球信息收集、城市规划、定位导航和应急灾害等方面有重大的研究意义,遥感数据在军事和民用领域都有重要的应用价值。遥感图像的使用需要很精密的算法对图像上的物体进行识别、分割、分析等^[6]。

如图2所示,近年来关于遥感图像智能解译的众多研究都与深度学习技术具有较强的关联性,主要有目标检测、地物分类、场景分类和变化检测等实际应用。目标检测是计算机视觉领域的核心问题之一,它的任务是找出图像中所有感兴趣的目标(物体),确定它们的类别和位置。由于在遥感图像中各类地物有不同的外观、

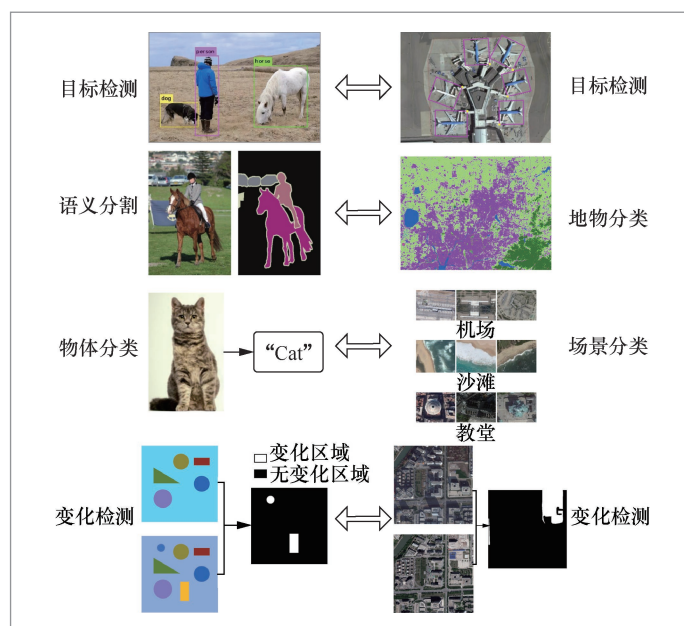


图2 深度学习与传统遥感任务的关联性

形状和姿态,加上成像时光照、遮挡等因素的干扰,传统的目标检测识别方法不适用于海量的遥感数据,其所依赖的特征表达式是人工设计的,很难从海量的数据中学习出一个有效的分类器来充分挖掘数据之间的关联。而深度学习强大的高级(更具抽象和语义意义)特征表示和学习能力可以为图像中的目标提取提供有效的框架^[7],YOLO、Faster-RCNN等深度学习算法已经在遥感图像目标检测中得到了广泛的应用。遥感地物分类就是利用计算机对遥感图像中各类地物的光谱信息和空间信息进行分析,选择特征,将图像中各个像元按照某种规则或算法划分成不同的类别,然后获得遥感图像中与实际地物的对应信息,从而实现图像的分类^[8],然而传统的地物分类方法在应用上受到很大限制。近年在各种出色的语义分割模型提出之后,基于语义分割方法对遥感图像地物分类的研究取得了非常大的进展,如UNet、SegNet。遥感图像场景分类致力于根据图像内容自动辨别土地利用或覆盖的类别,随着深度学习的出现与发展,其因强大的深层特征表示能力已经逐渐取代手工特征,成为场景分类的主流方法^[9]。遥感图像的变化检测可用于监测区域内不同时期的变化,通常通过收集某一区域在不同时刻的多幅遥感影像来识别该区域的土地覆盖变化,可以分为像素级别和对象级别,特征提取能力和端到端的方式使卷积神经网络对影像块具有很强的特征表达能力,相比于传统的特征工程方法有很大的优势,其能够直接从数据端得到变化检测的分类结果^[10]。

随着航天遥感信息技术的快速发展和全球对地观测体系的建立,多种遥感数据的提供能力越来越强,这些典型的遥感智能解译算法的使用场景越来越广泛,产品的精度也越来越高。为了进一步提高遥感

数据处理的精度和效率,需要更高效的遥感智能分析处理技术^[11]。

1.3 Docker容器技术

在容器技术出现之前,开发者大多使用虚拟机构建应用程序所需要的运行环境,如KVM、Xen等虚拟机技术^[12]。Docker作为一种新的开源虚拟化容器技术,一台主机能运行上百台容器,可实现秒级启动,并且在CPU计算效率、内存访问效率、GPU计算效率和网络请求响应效率等方面接近物理机,可以在各类云平台或本地机器上部署运行。Docker容器包含了三大基本概念:镜像、容器、仓库。镜像是创建容器的基础,通过联合文件系统技术,以分级存储方式进行创建,在一个镜像中的修改不会对其他镜像产生影响,每一个容器都运行在属于自己的独立环境中,从而很好地避免了依赖冲突。采用Docker容器技术封装深度学习算法和程序运行环境,保证了算法的高复用性,实现了资源隔离,解决了不同算法之间依赖冲突的问题,可以将不同开发人员在不同环境下开发的程序集成到一个标准环境中,在不同机器上无差别运行,节省配置环境和调试程序代码的时间。Docker架构如图3所示。

1.4 Kubernetes容器云平台

Kubernetes是基于容器技术的分布式云平台架构,是为容器集群打造的开源编排工具之一^[13]。作为开源社区非常流行的平台即服务(platform-as-a-service, PaaS)平台之一, Kubernetes支持Docker技术,具有资源管理自动化、容器资源相互隔离、资源调度灵活、兼容分布式存储系统等特点,已经成

为业界广泛使用的服务器资源共享的方式。Kubernetes容器集群编排系统将集群内的服务器按功能划分为主节点(master node)和计算节点(work node)^[14]。图4为Kubernetes系统的总体设计架构,主节点的重要组件包括:应用程序接口(application programming interface, API)服务器,该组件负责为客户端提供RESTful API服务接口,响应客户端发送的操作Pod请求,并将集群内Pod的配置信息存储到etcd元数据库中供主节点端的其他组件访问;控制管理器,该组件作为集群的控制器管理者,负责检测集群内计算节点、Pod、命名空间等出现的故障以及修复检测到的故障,以保障集群正常稳定运行^[15];调度器,该组件用于确保将Pod调度到一个合适的计算节点上运行。子节点端的重要组件包括:kubelet,该组件用于对运行在计算节点上的Pod对象实施具体的操作,包括创建、删除等;kube-proxy,该组件将用户提交的请求转发到后端的目标Pod实例上,同时支持将用户发送的请求负载均衡到同一个服务的多个后端Pod实例上。

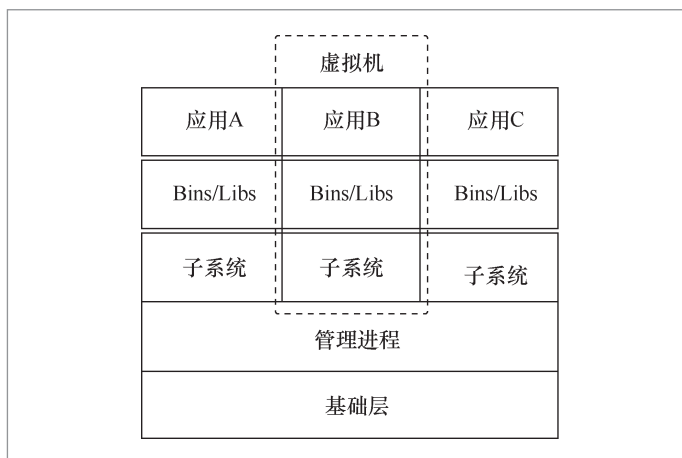


图3 Docker 架构

遥感智能解译算法模型众多,算法之间具有一定的层次和逻辑关系^[16],而Kubernetes容器云技术可以根据部署镜像之间的依赖关系,指定能够控制批任务的Job资源对象,按序执行容器内的算法,完成容器的合理调度与编排^[17],保证任务的顺利执行,失败时自动重启任务,并且可以将处理流程保存为YAML文件,保证处理流程随时可以完全重现。相对于传统的深度学习,基于Kubernetes容器云平台的深度学习具有很大优势,见表1。

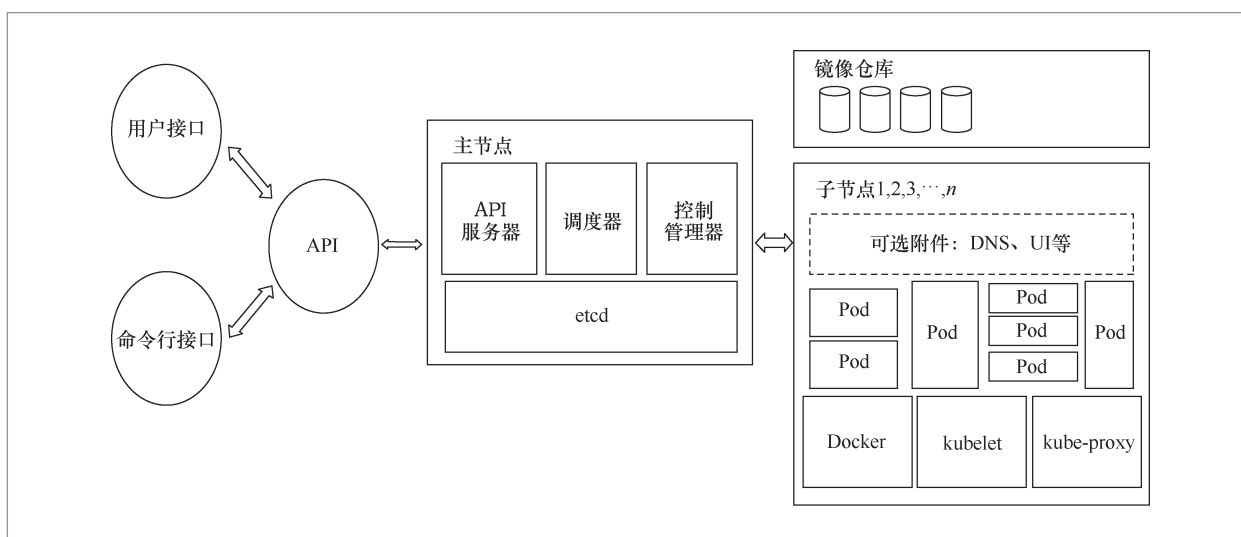


图4 Kubernetes 架构

表1 传统深度学习和基于 Kubernetes 容器云平台深度学习优缺点对比

对比项	传统深度学习	基于Kubernetes容器云平台的深度学习
GPU调度	手动调度	自动调度
数据存储	训练数据和模型文件难以保存	分布式存储GlusterFS
部署集成	困难,手动解决冲突	容器化集成部署

2 遥感智能解译算法集成方案设计

2.1 集成方案整体架构

随着计算机技术的发展,不同遥感解译处理算法、不同图像处理流程可以在统一的计算平台上集成与管理,不同遥感数据格式也可以相互集成与转换。为了应对多源、多算法的需求,集成方案的系统结构需要清晰且具有一定的开放性,用户可以用简单的方式添加新的功能处理模块,使集成方案具有弹性的纵向、横向扩展能力,为用户提供集遥感智能解译算法的数

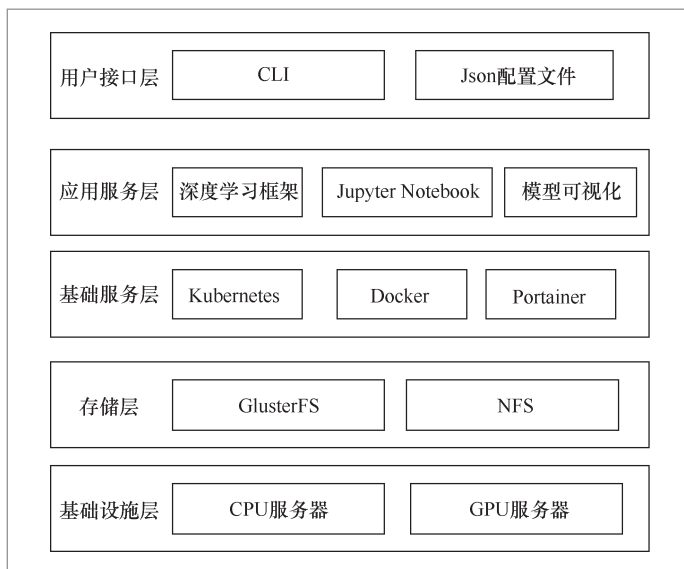


图5 方案分层架构

据读取,模型开发、训练、推理和部署,以及遥感产品生产于一体的“一站式”服务。为了满足遥感产品生产的需求,集成方案需要具备协同多计算节点工作完成遥感数据处理及产品生产的能力,构建以高性能计算集群为处理核心的集群架构。

因此,本文设计了基于容器云技术的遥感智能解译算法集成方案的分层架构,如图5所示。该框架可分为用户接口层、应用服务层、基础服务层、存储层、基础设施层。

- **用户接口层**: 即用户直接的交互界面,主要通过命令行界面(CLI)或者Json配置文件调用Kubernetes中的API服务,进行训练的配置和日志显示。基于该接口层,用户可以将训练参数、是否使用GPU、输出文件的存储路径等项目管理的配置传入配置文件。当用户进行多次实验时,仅需要修改配置文件,不需要关注底层设置,模型训练和推理流程全部自动化完成。训练节点在训练过程中自动将开始训练时间、当前训练时长、训练状态等数据写入日志文件,并同步到主节点,方便部署在主节点进行展示。

- **应用服务层**: 包括常见的深度学习框架PyTorch、TensorFlow,模型可视化和Jupyter Notebook服务。

- **基础服务层**: 包括Kubernetes服务、Docker服务和容器可视化管理服务Portainer,提供镜像制作、容器调度和编排等功能。

- **存储层**: 采用GlusterFS分布式存储系统和网络文件系统(network file system, NFS),负责保存深度学习模型、参数和日志文件,读取训练数据和测试数据。GlusterFS对Kubernetes兼容,具有良好的拓展性,可以将集群内闲置的存储资源统一起来,提高存储空间利用率,有效地提高深度学习模型和数据文件的读取速率。每个计算节点都部

署GlusterFS服务,从而使所有计算节点同时承担分布式存储任务。

- 基础设施层:包括CPU服务器和GPU服务器,是提供计算资源的基础。

总体而言,方案整体架构如图6所示,在主节点上部署Jupyter Notebook、控制面板(Dashboard)和深度学习可视化服务Visdom、Tensorboard,在计算节点上部署Docker服务和GlusterFS服务。用户还可以通过容器化部署的方式灵活添加自定义的功能服务。遥感智能解译算法模型开发和部署的总体技术路线如图7所示。

2.1 模型开发的设计

在集成方案的应用层,统一容器创建后的分配方式,为不同的框架补充提供统

一的服务功能,如容器对外链接服务,在预先设计好的基础开发镜像内部,封装基本的遥感特定数据格式文件读写软件包和常用的遥感预处理算法,免除用户从底层开始搭建环境的重复性工作,用户可以只根据自己的实际情况进行所需软件包的添加,轻松开启模型开发流程。模型开发根据预先设计的深度学习算法,基于深度学习框架,进行智能解译算法的代码编写和镜像构建。其基本流程如图8所示,具体如下。

(1)从私有仓库或开源仓库使用Docker pull操作获取所需的基础环境镜像。基于基础环境镜像,添加OpenCV、GDAL、SciPy等Python包。对基础镜像统一设置SSH登录端口,以便后期调试代码。

(2)在主节点,编写YAML文件,调用Kubernetes中的API服务器,分配镜像构

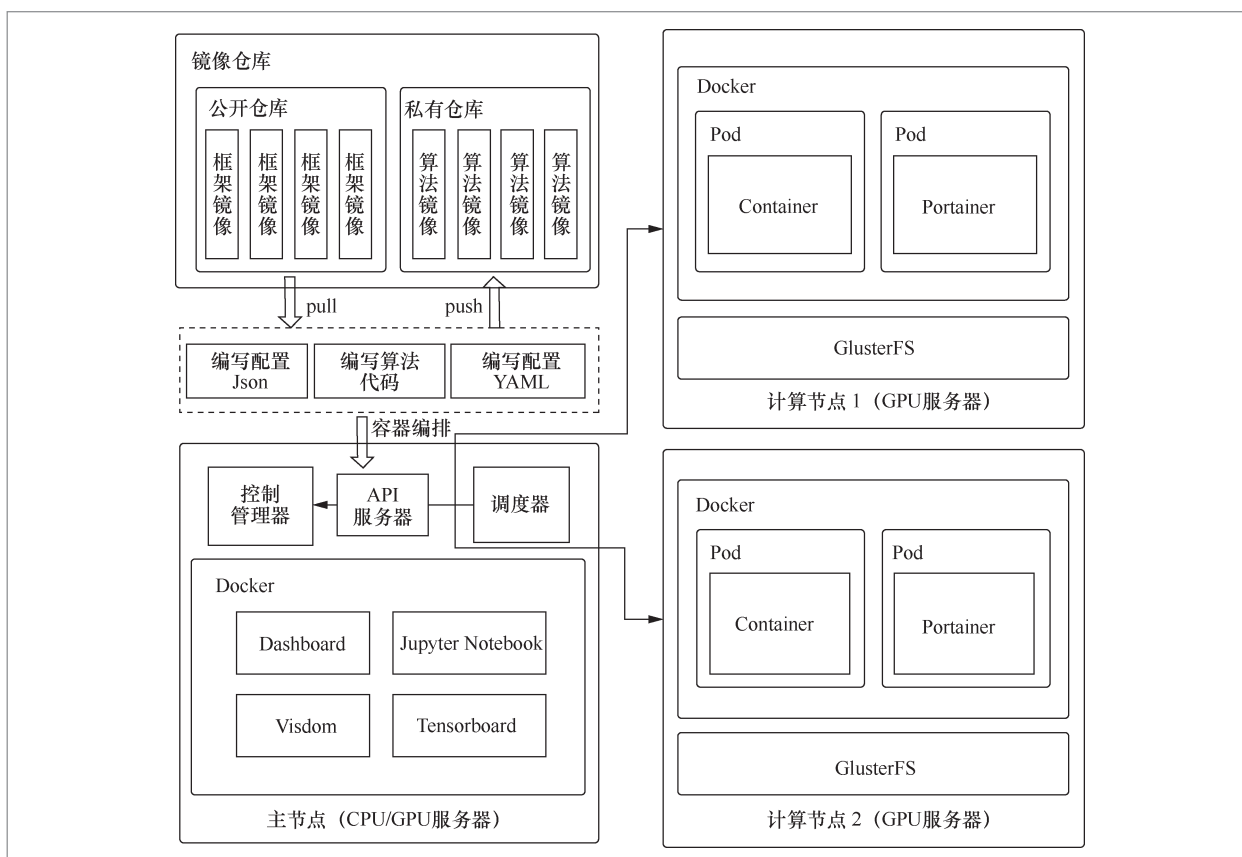


图6 方案整体架构

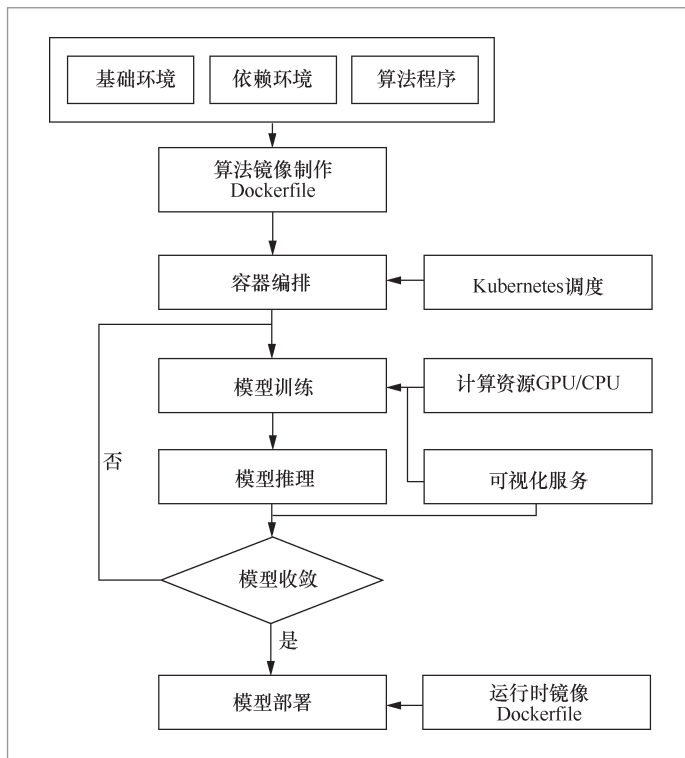


图7 总体技术路线

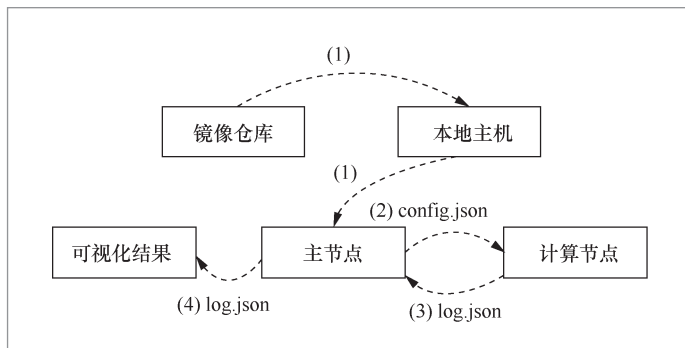


图8 模型开发流程

建的容器到Pod, 分配GPU给Pod, 并挂载数据集存储目录到容器, 编写算法代码, 对算法模型统一设置接口, 将模型开发所需的参数统一写入Json格式的文件config.json, 传输到计算节点。

(3) 计算节点对模型进行训练, 并输出模型输出日志为Json格式的文件log.json, 同步到主节点。

(4) 在主节点调用可视化服务, 将日志和模型调试过程中的影像结果可视化输出。

2.2 模型训练的设计

在模型训练的设计中, 为了将开发者从手动设置模型训练参数中解放出来, 将模型接口匹配到一个文件形式的配置数据上, 只需修改文件内容, 即可重复多次实验, 对比实验效果, 并实时同步训练日志和数据效果到主节点, 实现简单方便的可视化服务, 屏蔽了不同遥感智能解译算法训练结果的底层差异。模型训练通过设置配置文件进行实验, 使模型收敛, 可视化实验进程和对比实验效果, 基本流程如下。

(1) 将遥感数据采用分布式文件系统GlusterFS进行分布式存储, 统一数据格式。

(2) 制定实验计划, 修改config.json配置文件, 设定log.json日志文件的保存路径, 运行容器训练。设计的配置文件和日志文件的元数据见表2和表3, 保证不同的智能解译算法的对外接口一致。

(3) 利用Portainer服务可视化管理计算节点上容器的运行状态, 并通过IP: Port访问主节点上的Visom或Tensorboard服务, 实时监控训练过程。

(4) 保存计算节点日志文件到主节点, 对比实验结果。

2.3 模型推理的设计

在模型推理的设计中, 通过将不同节点归属到同一集群中, 省略将训练好的模型权重文件同步到主节点的过程, 减少了文件移动开销, 仅需通过模型权重文件的路径, 即可自动分配推理任务给计算节点, 并返回推理结果到统一路径保存。模型推理通过配置文件和预训练好的模

型权重进行测试,验证测试结果,并存储测试结果。其基本流程如图9所示,具体如下。

(1) 设置配置文件test.json和预训练权重的路径model.pth。

(2) 在主节点调用Kubernetes中的API服务器,分配模型推理的容器和配置文件config.json到计算节点,运行容器内算法,得到输出结果及日志文件log.json。

(3) 同步log.json到主节点,对推理结果进行显示和存储。

2.4 模型部署的设计

模型部署指高效地把训练好的深度学习的模型部署到生产环境,使用Kubernetes的容器编排服务,自动完成模型推理和结果输出。容器按照运行时间可分为两类:服务类容器和工作类容器。服务类容器通常持续提供服务,需要一直运行,如Jupyter Notebook和Portainer等。工作类容器则执行一次性任务,如批处理程序,完成后容器就退出。遥感信息产品的生产流程是固定的,变化部分仅是每个容器的运行参数,因此在模型部署环节,使用Kubernetes的Job机制,配置Job配置文件job.yaml,运行容器内的处理算法,在算法运行完成后,Job的状态会自动更新为Complete,即这个Job任务已经执行完成,Pod不再在计算节点中继续运行,输出并保存运行结果。

生产环境中往往需要在同一集群中部署大量的算法模型,因此如何高效地利用集群的存储空间也是模型部署环节需要考虑的问题。本方案通过搭建最小化的运行时环境镜像,可以有效缩小模型部署所需的存储空间,方便在物理机上集成大量的深度学习算法。参照Dockerhub上托管的官方镜像文件,TensorFlow和PyTorch的

表2 配置文件元数据

名称	数据类型	功能说明
Input_datadir	String	输入数据的挂载目录路径
Input_args	Array	输入参数列表
Output_dir	String	输出数据的存储路径
Output_weight	String	模型权重的保存路径
Phase_model	String	模型阶段(训练、推理)

表3 日志文件元数据

名称	数据类型	功能说明
jobstate	String	记录容器当前运行情况
loss	Array	模型损失函数
nowstep	String	当前运行步骤
totalstep	String	最大运行步骤
data	Object	输出数据

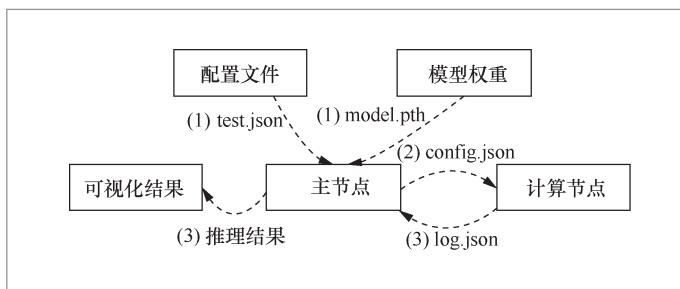


图9 模型推理流程

最小化运行时环境镜像和开发环境镜像所占空间大小对比如图10所示。

3 遥感深度学习算法集成实例

3.1 数据集与实验环境

为了验证该集成方案的有效性,本节基于Docker和Kubernetes的遥感深度学习算法集成方案,以DOTA、UC Merced Land-Use、DSLT、SpaceNet7这4个数据集为例,

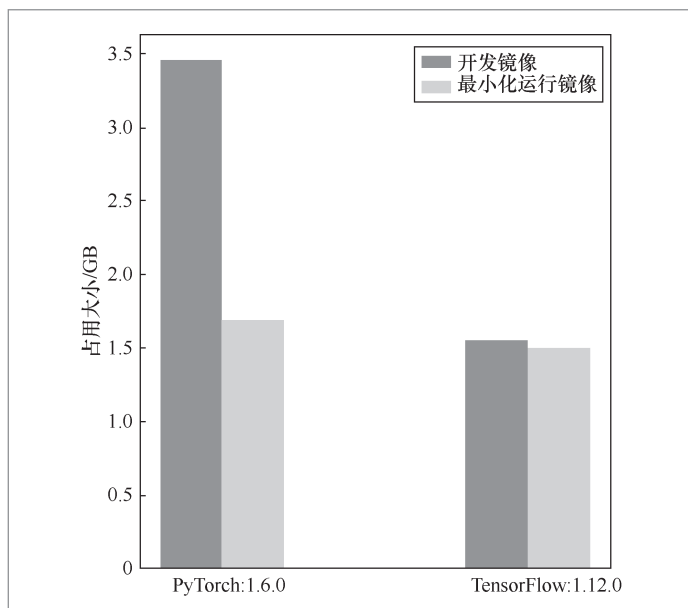


图 10 最小化运行时镜像与开发镜像

选取5种不同的基于深度学习的遥感智能解译算法,介绍4种遥感产品的生产。

DOTA是最大的面向对象的遥感卫星图像检测数据集,共发布了两个版本:DOTA-v1.0和DOTA-v1.5。DOTA-v1.0数据集中的图像来源于GoogleEarth以及我国的GF-1和JL-1卫星。该数据集共包含2 806张遥感图像(图像尺寸从800×800到4 000×4 000),一共有188 282个实例,分为15个类别:飞机、船只、储油罐、棒球场、网球场、篮球场、

田径场、港口、桥梁、大型车辆、小型车辆、直升飞机、足球场、环岛和游泳池。每个实例都由一个四边形边界框标注,顶点按顺时针顺序排列。UC Merced Land-Use数据集中的图像尺寸为256×256,像素分辨率为1英尺,包括棒球场、海滩、建筑物、森林、高速公路、港口、停车场、储油罐和网球场等21种类别,每一类有100张图片,共2 100张。DSLTL数据集提供了3波段和16波段格式的1 km×1 km卫星图像,标注了道路、建筑物、车辆、农场、树木、水道等10种不同的地物类型。SpaceNet7数据集包含约100个数据瓦片,这些数据瓦片是在两年的时间跨度内采集的4 m分辨率的多光谱图像,标注了相对应的建筑物标签,用来检测建筑物的变化情况。

实验中集群内的节点采用的配置见表4,采用Kubernetes 1.20,构建了一个GPU计算集群,在Kubernetes平台上运行容器化的深度学习模型,通过在主节点设置参数和配置文件,将训练数据集分别存放在不同节点上,进行训练和部署实验。

3.2 模型算法

目标检测算法采用两种算法,分别为旋转框目标检测算法S2ANet(single-

表 4 节点配置

IP地址	10.1.17.7	10.1.17.11	10.1.17.83
CPU	Inter Core i7-4790 CPU, 4 cores, 3.60 GHz	Intel Core Xeon E5-2620, 32 cores, 2.10 GHz	Intel Core Xeon E5-2620, 32 cores, 2.10 GHz
内存	24 GB RAM	126 GB RAM	126 GB RAM
硬盘空间	2 TB	16 TB	16 TB
GPU	1x NVIDIA GTX1080Ti	4x NVIDIA TITAN Xp	2x NVIDIA GTX2080Ti
操作系统	Ubuntu 18.04 LST	Ubuntu 16.04 LST	Ubuntu 20.04 LST
部署	Kubernetes 1.20, Docker 19.03, Dashboard	Kubernetes 1.20, Docker 19.03, Portainer 1.20	Kubernetes 1.20, Docker 19.03, Portainer 1.20

shot alignment network)^[18]和水平框目标检测算法YOLOv3^[19]。S2ANet通过修正锚定框,获取精确的旋转不变的卷积特征,以缓解分类和定位精度之间的冲突,用于大尺度遥感影像的目标检测任务,在DOTA等公开数据集上具有最先进的性能。YOLOv3是实时目标检测算法YOLO的第三个版本,其本身基于Darknet构建的神经网络算法引入了残差结构和多尺度融合策略,相对于YOLOv2,其对小物体的识别性能提高非常明显,更加适用于遥感图像的目标检测。

场景分类算法采用ResNet101网络,ResNet在2015年被提出,在ImageNet比赛分类任务上获得第一名。其引入了残差模块,缓解了随着网络加深模型准确率下降的情况^[20]。

地物分类算法采用UNet网络,此模型最开始用于医疗影像分割,采用编码器-解码器构造,解决的是二分类问题。将UNet的损失函数修改为多类别损失函数,使其适用于遥感图像的地物分类任务^[21-22]。

变化检测算法采用UNet-VGG16,即编码器部分使用VGG-16作为特征提取部分的UNet网络。

实验模型的设置见表5。

3.3 性能分析

在设置了参数和配置文件后,平台自动

创建了一个包含3个节点的集群,包含一个主节点和两个计算节点,在运行过程中,各节点算法正常训练。

在模型开发环节,采用公开仓库拉取环境镜像的方式搭建深度学习环境,因此只需在将镜像拉取到本地后,启动容器,即可进行模型开发。图11列出了启动容器的时间,每次启动可能会略有波动,启动容器时Kubernetes集群上没有运行其他容器,启动时间为容器从Stop状态转变为Running状态花费的时间。由图11可知,5个测试容器启动正常,启动效率达到了秒级,整体的启动时间都在5 s以内,实现了秒级部署深度学习环境。

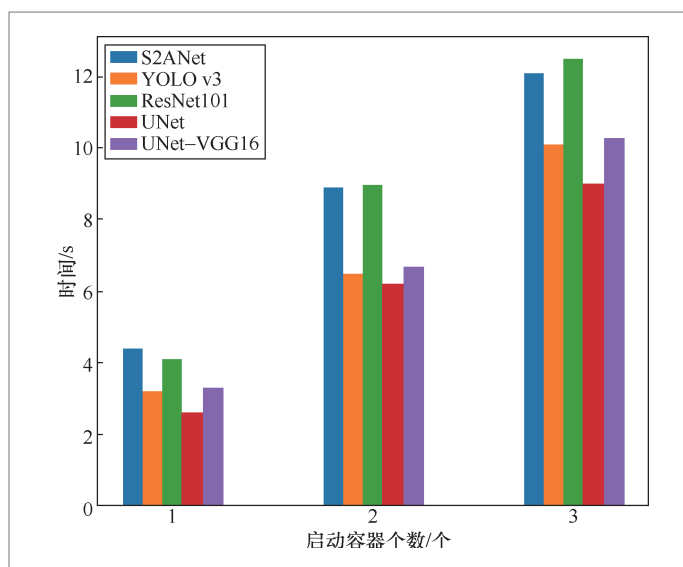


图 11 智能解译算法容器的启动时间测试结果

表 5 实验模型设置

算法	概述	数据集	基础镜像
S2ANet	旋转框目标检测算法 ^[1]	DOTA-v1.0	mmdetecitonv1:s2anet
YOLOv3	水平框目标检测算法 ^[2]	DOTA-v1.0	tensorflow/tensorflow:1.12.0-gpu-py3
ResNet101	场景分类算法 ^[3]	UC Merced	pytorch/pytorch:1.6.0-cuda10.1-cudnn7-devel
UNet	地物分类算法 ^[4]	DSTL	tensorflow/tensorflow:1.10.0-gpu-py3
UNet-VGG16	变化检测算法 ^[5]	SpaceNet7	sn7_baseline_image

在模型训练环节,在主节点可以得到统一可视化训练过程,如图12所示。在模型推理环节,采用Dashboard可视化管理监督模型的运行情况,如图13所示。

为了验证容器云动态部署方式进一步提升效率的效果,本文在物理机集群上测试了实验中4种算法部署任务的工作量,其中工作量以时间为衡量标准,如图14所示。搭建单个算法的环境或容器所需的平均工作量为1 min,3组实验分别进行部署,对比了使用3种不同部署方式的工作量^[17]。其中实验一直接在物理机上部署算法,通过复制上传数据;实验二使用Docker配置算

法容器,通过复制上传数据及镜像;实验三使用容器云动态配置算法容器及上传镜像,通过分布式存储访问数据。总体来看,在物理机上搭建算法环境与搭建算法容器工作量基本相同,镜像复制和数据复制、容器环境安装所需工作量较少,使用容器云部署算法容器之后,实验任务的工作量可减少到直接物理机部署的任务工作量的50%以下,大幅提升部署效率。

同时,对物理机部署及相同环境的容器化部署的4种智能解译算法进行了测试,对比了两种方式对相同数据进行推理所需的运行时间。运行时间是指在同一计算节

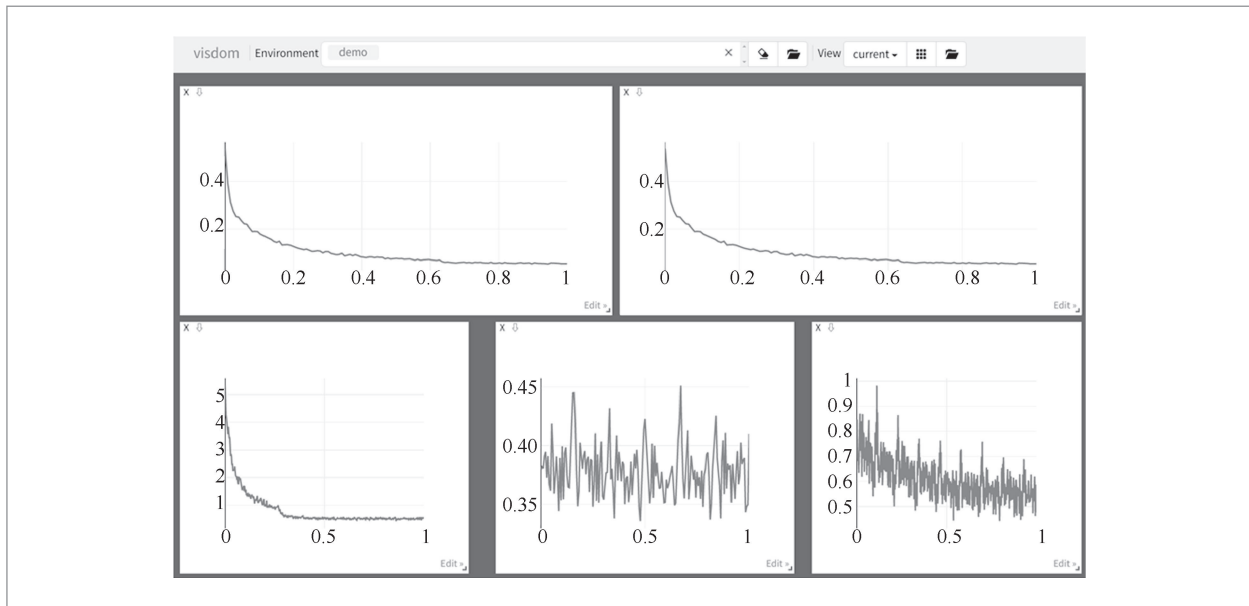


图 12 Visdom 可视化训练过程

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
unetvgg-pod	default	app: unetvgg version: v1	10.1.17.11	Running	0	0.00m	1.575k	2 minutes ago
unet-pod	default	app: unet version: v1	10.1.17.11	Running	0	0.00m	1.853k	2 minutes ago
resnet101-pod	default	app: pytorch version: v1	10.1.17.83	Running	0	0.00m	1.640k	2 minutes ago
yolov3-pod	default	app: yolov3 version: v1	10.1.17.83	Running	0	0.00m	1.775k	3 minutes ago
s2anet-pod	default	app: mmdetection version: v1	10.1.17.83	Running	0	0.00m	2.133k	3 minutes ago

图 13 Dashboard 可视化管理界面

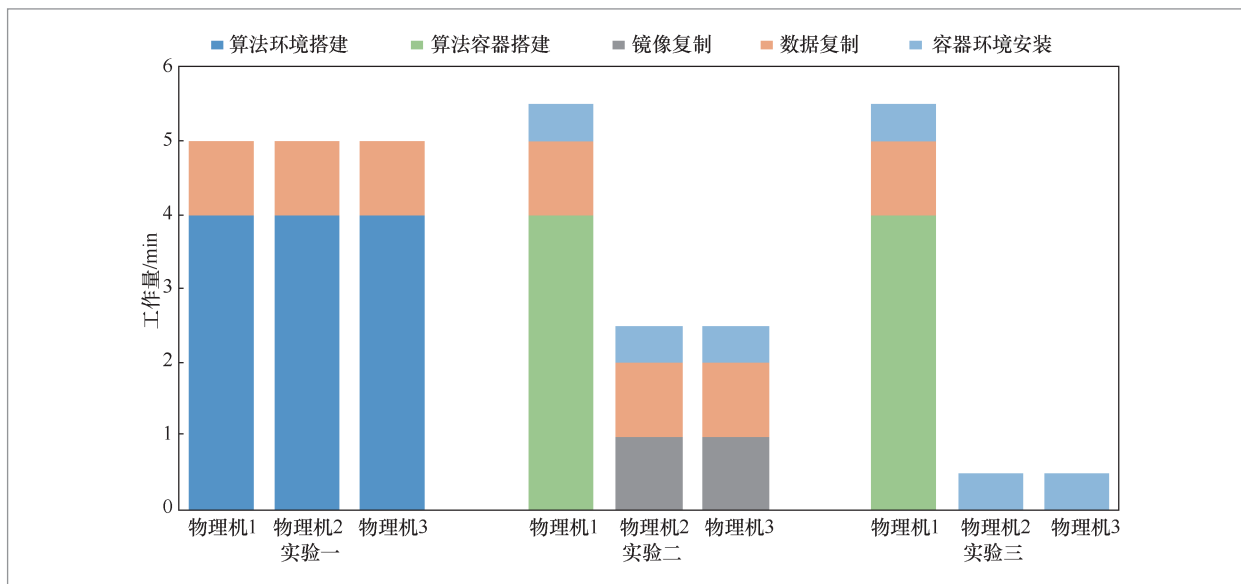


图 14 部署工作量

点,物理机上部署的算法和容器化部署的算法运行推理代码并输出检测结果所花费的时间。选取100张算法对应的标准图像进行测试,测试结果如图15所示,横坐标为测试图像数量,纵坐标为取对数后的运行时间,单位为秒(s)。Docker环境与物理机环境运行效率基本一致。但是在新的机器中安装部署生产环境时,相比物理机环境,Docker容器环境可以显著减少配置环节所需的工作,方便程序的迁移和复用,缩短算法部署和集成所需的时间。

4 结束语

高效精准地搭建深度学习环境,配置资源进行训练推理,是开发智能解译算法的核心步骤。然而,遥感算法程序的环境依赖众多,传统的开发环境管理技术导致资源占用大,容易造成资源浪费,集成部署的效率低,且模型开发流程复杂。

本文主要围绕新兴的Kubernetes容器云技术展开研究,针对目前基于深度学

习的遥感智能解译算法计算资源利用效率低、集成难、开发部署效率低的问题,提出了容器化的遥感信息高效智能解译架构,实现了规范化、流程化的遥感深度学习算法开发及部署,实现了多种典型的遥感智能解译算法的集成,实现了资源快速分配、开发环境秒级启动、模型训练高效化和自动化,缩短了遥感深度学习算法的开发周期。该框架可满足智能化遥感产品生产的需求,为我国未来航天遥感智能解译技术的统一集成提供参考。

参考文献:

- [1] 李振举,李学军,刘涛,等. 遥感云计算: 研究现状与展望[J]. 装备学院学报, 2015, 26(5): 95-100.
LI Z J, LI X J, LIU T, et al. Remote sensing cloud computing: current research and prospect[J]. Journal of Equipment Academy, 2015, 26(5): 95-100.
- [2] 李鹏飞. 遥感云计算中的基础设施关键技术研究[D]. 秦皇岛: 燕山大学, 2017.

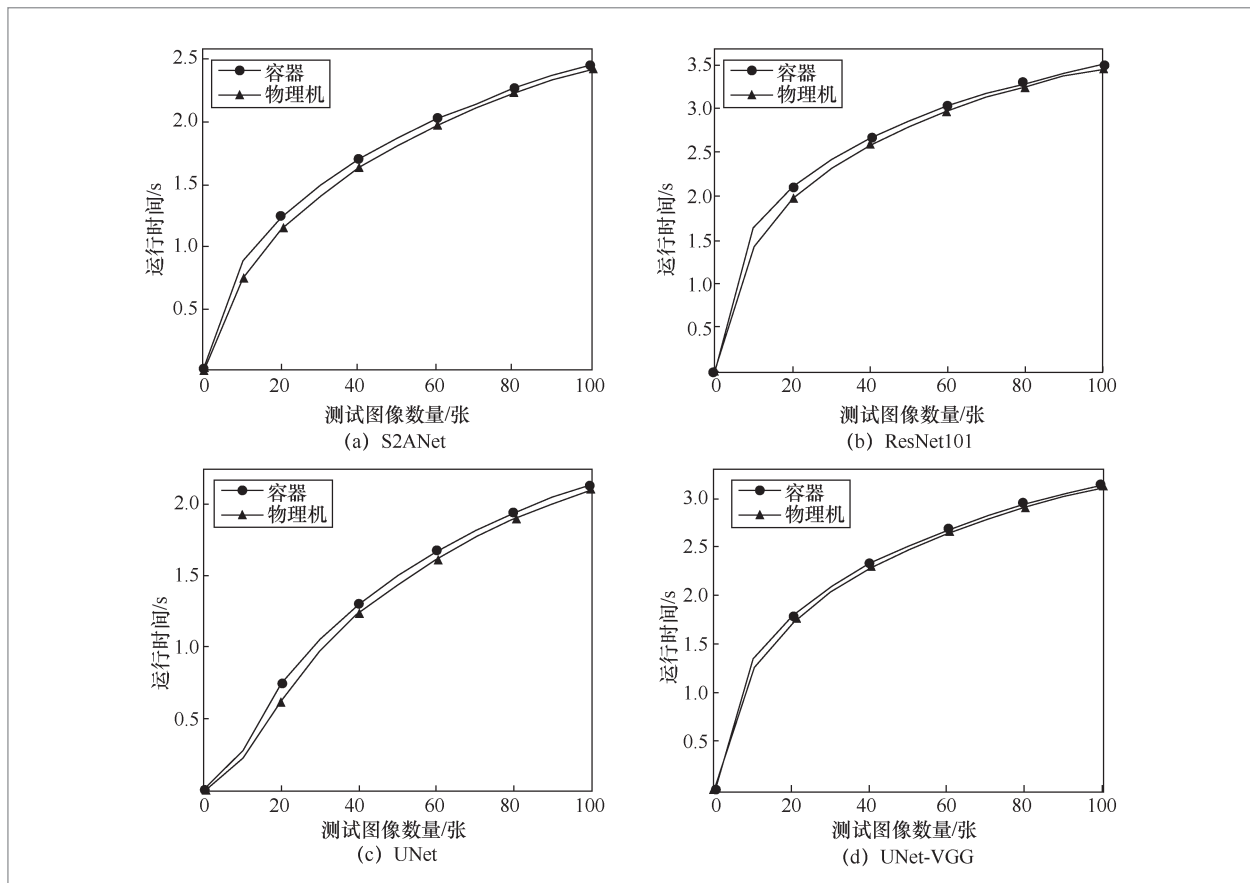


图 15 物理机上部署的算法和容器化部署的算法运行所耗时间

- LI P F. Research on key technologies of remote sensing cloud computing infrastructure[D]. Qinhuangdao: Yanshan University, 2017.
- [3] 张兵. 遥感大数据时代与智能信息提取[J]. 武汉大学学报·信息科学版, 2018, 43(12): 1861-1871.
- ZHANG B. Remotely sensed big data era and intelligent information extraction[J]. Geomatics and Information Science of Wuhan University, 2018, 43(12): 1861-1871.
- [4] 李德仁, 马军, 邵振峰. 论时空大数据及其应用[J]. 卫星应用, 2015(9): 7-11.
- LI D R, MA J, SHAO Z F. Spacetime big data and its applications[J]. Satellite Application, 2015(9): 7-11.
- [5] 官鹏. 对遥感科学应用的一点看法[J]. 遥感学报, 2019, 23(4): 567-569.
- GONG P. Towards more extensive and deeper application of remote sensing[J]. Journal of Remote Sensing, 2019, 23(4): 567-569.
- [6] 李德仁, 张良培, 夏桂松. 遥感大数据自动分析与数据挖掘[J]. 测绘学报, 2014, 43(12): 1211-1216.
- LI D R, ZHANG L P, XIA G S. Automatic analysis and mining of remote sensing big data[J]. Acta Geodaetica et Cartographica Sinica, 2014, 43(12): 1211-1216.
- [7] 高常鑫, 桑农. 基于深度学习的高分辨率遥感影像目标检测[J]. 测绘通报, 2014(S1): 108-111.
- GAO C X, SANG N. Deep learning for object detection in remote sensing image[J]. Bulletin of Surveying and Mapping, 2014(S1): 108-111.
- [8] 徐永斌, 王树文. 基于地学分析与遥感智能解译模型的土地覆盖/土地利用分类[J]. 测绘与空间地理信息, 2010, 33(4): 64-66, 69.
- XU Y B, WANG S W. Land cover and land

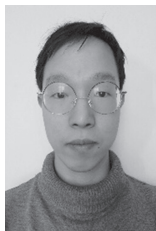
- use classification based on earth sciences analysis and automatic interpretation model of remote sensing[J]. *Geomatics & Spatial Information Technology*, 2010, 33(4): 64–66, 69.
- [9] 刘大伟, 韩玲, 韩晓勇. 基于深度学习的高分辨率遥感影像分类研究[J]. *光学学报*, 2016, 36(4): 0428001.
LIU D W, HAN L, HAN X Y. High spatial resolution remote sensing image classification based on deep learning[J]. *Acta Optica Sinica*, 2016, 36(4): 0428001.
- [10] 佟国峰, 李勇, 丁伟利, 等. 遥感影像变化检测算法综述[J]. *中国图象图形学报*, 2015, 20(12): 1561–1571.
TONG G F, LI Y, DING W L, et al. Review of remote sensing image change detection[J]. *Journal of Image and Graphics*, 2015, 20(12): 1561–1571.
- [11] 杨帆, 刘湘南. 遥感影像解译的研究现状和发展趋势[J]. *国土资源遥感*, 2004, 16(2): 7–10, 15.
YANG G, LIU X N. The present research condition and development trend of remotely sensed imagery interpretation[J]. *Remote Sensing for Land & Resources*, 2004, 16(2): 7–10, 15.
- [12] 肖熠, 鲁永泉, 谢思焯. 一种针对GPU资源的深度学习容器云研究[J]. *中国传媒大学学报(自然科学版)*, 2017, 24(6): 16–20.
XIAO Y, LU Y Q, XIE S Y. A deep learning container cloud study for GPU resources[J]. *Journal of Communication University of China (Science and Technology)*, 2017, 24(6): 16–20.
- [13] 董宁宁. 基于Kubernetes的软件工程教育云基础容器平台子系统的设计与实现[D]. 南京: 南京大学, 2019.
DONG N N. The design and implementation of container infrastructure subsystem in software engineering education cloud based on kubernetes[D]. Nanjing: Nanjing University, 2019.
- [14] 朱天放, 梁明, 王家乐, 等. 基于容器云的深度学习平台设计与实现[J]. *电子设计工程*, 2019, 27(9): 21–25.
ZHU T F, LIANG M, WANG J L, et al. Design and implementation of deep learning platform based on Kubernetes[J]. *Electronic Design Engineering*, 2019, 27(9): 21–25.
- [15] 余昌发, 程学林, 杨小虎. 基于Kubernetes的分布式TensorFlow平台的设计与实现[J]. *计算机科学*, 2018, 45(S2): 527–531.
YU C F, CHENG X L, YANG X H. Design and implementation of distributed TensorFlow platform based on kubernetes[J]. *Computer Science*, 2018, 45(S2): 527–531.
- [16] 李宏益, 唐婷, 单小军, 等. 多源定量遥感产品并行处理系统设计及实现[J]. *计算机应用研究*, 2016, 33(11): 3334–3339.
LI H Y, TANG P, SHAN X J, et al. Parallel processing system design and implementation for multi-sensor quantitative remote sensing production[J]. *Application Research of Computers*, 2016, 33(11): 3334–3339.
- [17] 张杰, 郑柯, 唐婷, 等. Docker容器化下的遥感算法程序集成方法[J]. *中国图象图形学报*, 2019, 24(10): 1813–1822.
ZHANG J, ZHENG K, TANG P, et al. Integration of remote sensing algorithm program using Docker container technology[J]. *Journal of Image and Graphics*, 2019, 24(10): 1813–1822.
- [18] HAN J M, DING J, LI J, et al. Align deep features for oriented object detection[J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2022, 60: 1–11.
- [19] 戴伟聪, 金龙旭, 李国宁, 等. 遥感图像中飞机的改进YOLOv3实时检测算法[J]. *光电工程*, 2018, 45(12): 84–92.
DAI W C, JIN L X, LI G N, et al. Real-time airplane detection algorithm in remote-sensing images based on improved YOLOv3[J]. *Opto-Electronic Engineering*, 2018, 45(12): 84–92.
- [20] WANG G L, FAN B, XIANG S M, et al. Aggregating rich hierarchical features for scene classification in remote sensing imagery[J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017, 10(9): 4104–4115.

- [21] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint, 2014, arXiv: 1409.1556.
- [22] 苏健民, 杨岚心, 景维鹏. 基于U-Net的高分辨率遥感图像语义分割方法[J]. 计算机工程

与应用, 2019, 55(7): 207-213.

SU J M, YANG L X, JING W P. U-Net based semantic segmentation method for high resolution remote sensing image[J]. Computer Engineering and Applications, 2019, 55(7): 207-213.

作者简介



赵智韬 (1998-), 男, 中国科学院大学资源与环境学院硕士生, 主要研究方向为遥感信息提取。



赵理君 (1986-), 男, 博士, 中国科学院空天信息创新研究院副研究员, 主要研究方向为高分辨率遥感图像信息解译算法研究及工程应用。



张正 (1989-), 男, 博士, 中国科学院空天信息创新研究院助理研究员, 主要研究方向为人工智能的遥感应用以及遥感算法集成运行系统的构建。



唐娉 (1968-), 女, 中国科学院空天信息创新研究院研究员、博士生导师, 主要研究方向为遥感图像处理和大数据技术研究。

收稿日期: 2021-09-29

基金项目: 国家自然科学基金资助项目 (No.41701397); 中国科学院空天信息创新研究院“未来之星”人才计划项目 (No.2020KTYWLZX03); 中国博士后科学基金资助项目 (No.2019M660852)

Foundation Items: The National Natural Science Foundation of China (No.41701397), “Future Star” Talent Plan of Aerospace Information Research Institute, Chinese Academy of Sciences (No.2020KTYWLZX03), China Postdoctoral Science Foundation (No.2019M660852)