

GLCC 2022 项目申请书

大图分割算法

项目摘要

在图神经网络训练场景中，为了针对大图，在单张GPU卡存不下的情况下，研究提供一种图分割算法，高效的切分到多卡下，且保证精度无损。

提案人：房森 wirth.fang@foxmail.com
导师：吴志华 wuzhihua02@baidu.com

项目摘要

项目概览

产出介绍

项目目标及技术实现梳理

完成大图分割算法分步工作

提供图分割代码

小图下，提供采用和不采用的采样结果、训练精度度量、性能数据

大图下，有代价函数评估分割的代价、最优性等

开发进度表

社区联络期 (June 25 - June 30)

编码阶段一 (July 1 - July 15)

编码阶段二 (July 15 - Aug 30)

编码阶段三 (Sep 1 - Sep 23)

缓冲时间 (7 days)

关于我

申请信息

Self-introduction

参考资料

项目概览

产出介绍

对基于飞桨开发大图分割算法的课题目标：

- ☐ 梳理课题目标结构，调研制定合理的实现方案
- ☐ 完成大图分割算法分步工作：
 - ☐ 提供图分割代码
 - ☐ 小图下，提供采用和不采用的采样结果、训练精度度量、性能数据
 - ☐ 大图下，有代价函数评估分割的代价、最优性等
- ☐ 实现相关算法并整合到飞桨Paddle的线上仓库（可选）
- ☐ 产出系统化报告
- ☐ 争取投递顶会paper

项目目标及技术实现梳理

首先，飞桨(PaddlePaddle)是一个开源DL平台，以百度多年的深度学习技术研究和业务应用为基础，是中国首个自主研发、功能完备、开源开放的产业级深度学习平台，集深度学习核心训练和推理框架、基础模型库、端到端开发套件和丰富的工具组件于一体。目前，飞桨累计开发者477万，服务企业18万家，基于飞桨开源深度学习平台产生了56万个模型。飞桨助力开发者快速实现AI想法，快速上线AI业务。帮助越来越多的行业完成AI赋能，实现产业智能化升级。

在我们实现目标时，基本上遵循 1. 模型设计 2. 准备数据 3. 训练设置 4. 应用部署 5. 模型评估 6. 循环重做等系列步骤。通过运用相关飞桨相关api方便快捷的实现模型的创新。

图像分割是计算机视觉研究中的一个经典难题，已经成为图像理解领域关注的一个热点，图像分割是图像分析的第一步，是计算机视觉的基础，是图像理解的重要组成部分，同时也是图像处理中最困难的问题之一。所谓图像分割是指根据灰度、彩色、空间纹理、几何形状等特征把图像划分成若干个互不相交的区域，使得这些特征在同一区域内表现出一致性或相似性，而在不同区域间表现出明显的不同。简单的说就是在一副图像中，把目标从背景中分离出来。对于灰度图像来说，区域内部的像素一般具有灰度相似性，而在区域的边界上一般具有灰度不连续性。关于图像分割技术，由于问题本身的重要性和困难性，从20世纪70年代起图像分割问题就吸引了很多研究人员为之付出了巨大的努力。虽然到目前为止，还不存在一个通用的完美的图像分割的方法，但是对于图像分割的一般性规律则基本上已经达成的共识，已经产生了相当多的研究成果和方法。

如果要完美的做好项目的开发实现，首先要做的是精通工具，我很久以前就在Windows环境中配置好了飞桨开发环境，查阅了官网文档，对使用其进行开发有一些心得：

在分析了任务课题之后，我初步制定了三个分步目标和相关产出：

- ☐ 完成大图分割算法分步工作：
 - ☐ 提供图分割代码
 - ☐ 小图下，提供采用或不采用的采样结果、训练精度度量、性能数据
 - ☐ 大图下，有代价函数评估分割的代价、最优性等
- ☐ 产出系统化报告

完成大图分割算法分步工作

可借鉴的分割方法

通过借鉴其他人的分割方法是有益的：

传统分割算法

1. 基于阈值的分割方法

阈值法的基本思想是基于图像的灰度特征来计算一个或多个灰度阈值，并将图像中每个像素的灰度值与阈值作比较，最后将像素根据比较结果分到合适的类别中。因此，该方法最为关键的一步就是按照某个准则函数来求解最佳灰度阈值。

阈值法特别适用于目标和背景占据不同灰度级范围的图。

图像若只有目标和背景两大类，那么只需要选取一个阈值进行分割，此方法成为单阈值分割；但是如果图像中有多个目标需要提取，单一阈值的分割就会出现作物，在这种情况下就需要选取多个阈值将每个目标分隔开，这种分割方法相应的成为多阈值分割。

阈值分割方法的最关键就在于阈值的选择。若将智能遗传算法应用在阈值筛选上，选取能最优分割图像的阈值，这可能是基于阈值分割的图像分割法的发展趋势。

2. 基于区域的图像分割方法

基于区域的分割方法是以直接寻找区域为基础的分割技术，基于区域提取方法有两种基本形式：一种是区域生长，从单个像素出发，逐步合并以形成所需要的分割区域；另一种是从全局出发，逐步切割至所需的分割区域。

基于边缘检测的分割方法

基于边缘检测的图像分割算法试图通过检测包含不同区域的边缘来解决分割问题。它可以说是人们最先想到也是研究最多的方法之一。通常不同区域的边界上像素的灰度值变化比较剧烈，如果将图片从空间域通过傅里叶变换到频率域，边缘就对应着高频部分，这是一种非常简单的边缘检测算法。

边缘检测技术通常可以按照处理的技术分为串行边缘检测和并行边缘检测。串行边缘检测是要想确定当前像素点是否属于检测边缘上的一点，取决于先前像素的验证结果。并行边缘检测是一个像素点是否属于检测边缘上的一点取决于当前正在检测的像素点以及与该像素点的一些临近像素点。

最简单的边缘检测方法是并行微分算子法，它利用相邻区域的像素值不连续的性质，采用一阶或者二阶导数来检测边缘点。近年来还提出了基于曲面拟合的方法、基于边界曲线拟合的方法、基于反应-扩散方程的方法、串行边界查找、基于变形模型的方法。



(a) 梯度算法处理的结果



(b) Roberts 算法



(c) Sobel 算法



(d) Prewitt 算法



(e) Kirsch 算法



(f) Laplacian 算法

基于小波分析和小波变换的图像分割方法

小波变换是近年来得到的广泛应用的数学工具，也是现在数字图像处理必学部分，它在时间域和频率域上都有量高的局部化性质，能将时域和频域统一于一体来研究信号。而且小波变换具有多尺度特性，能够在不同尺度上对信号进行分析，因此在图像分割方面的得到了应用，

二进小波变换具有检测二元函数的局部突变能力，因此可作为图像边缘检测工具。图像的边缘出现在图像局部灰度不连续处，对应于二进小波变换的模极大值点。通过检测小波变换模极大值点可以确定图像的边缘小波变换位于各个尺度上，而每个尺度上的小波变换都能提供一定的边缘信息，因此可进行多尺度边缘检测来得到比较理想的图像边缘。



上图左图是传统的阈值分割方法，右边的图像就是利用小波变换的图像分割。可以看出右图分割得到的边缘更加准确和清晰

另外，将小波和其他方法结合起来处理图像分割的问题也得到了广泛研究，比如一种局部自适应阈值法就是将Hilbert图像扫描和小波相结合，从而获得了连续光滑的阈值曲线。

小图下，提供采用和不采用的采样结果、训练精度度量、性能数据

我们通常把学习器的预测和真实值之间的差异成为误差，在训练集上的误差成为训练误差，而在新样本（测试集）上的误差成为泛化误差。显然我们希望得到的是泛化误差小的学习器。本次学习的重点，便是如何利用一组样本，来训练和测试学习器的性能，其中使用到样本的分组方法和性能的度量方法。

样本分组要尽可能满足训练集和测试集互斥，因为我们更希望测试学习器“举一反三”的能力。若测试集与训练集部分重合，相当于“泄题”。

留出法简单地说就是选取一部分样本作为训练集，剩余的样本为测试集

在使用留出法时，需要注意：

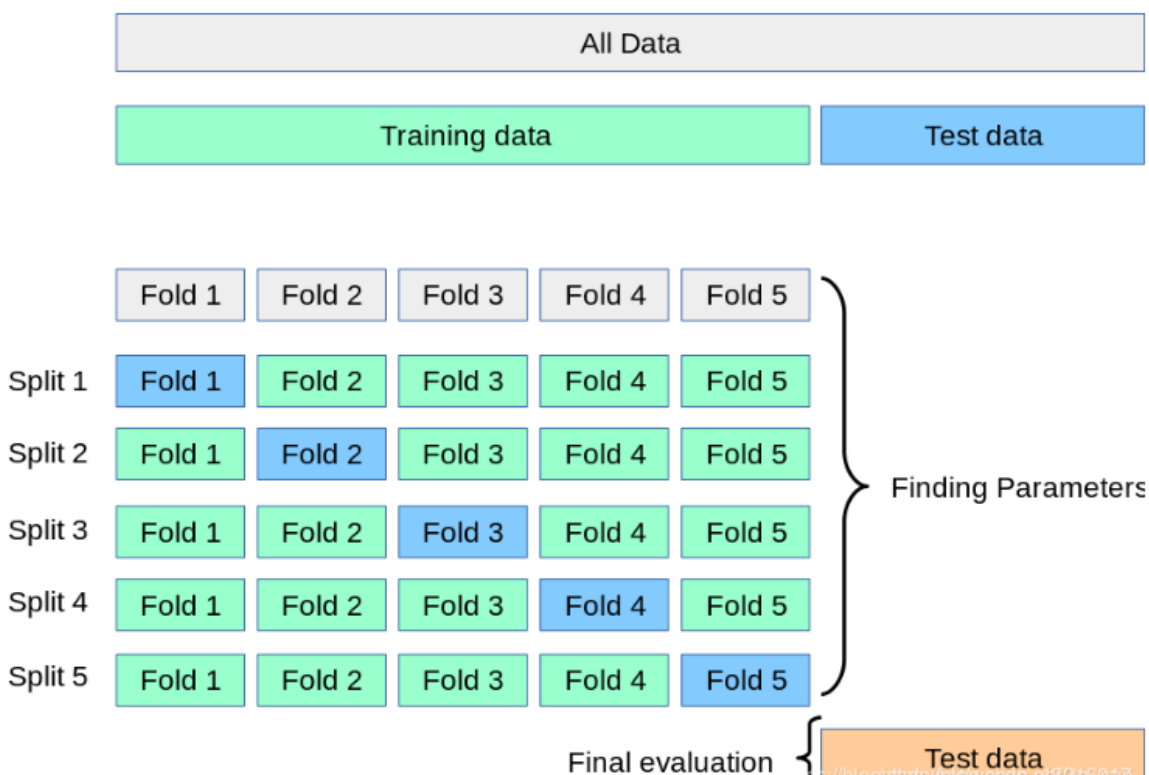
- 要有足够的样本量，以保证训练模型的效果
- 在划分时注意保证数据分布的一致性（如：500个样本中正例和反例的比为2:3，则在训练集和测试集中正例和反例的比也要求为2:3），只需要采用随机分层抽样即可
- 为了减弱随机划分的影响，重复划分训练集和测试集，对得到的多次结果取平均作为最后的结果

一般训练集和测试集的比例在8:2或者7:3

当然留出法的缺点也非常明显，即它会损失一定的样本信息；同时需要大样本。

交叉验证法(cross validation)可以很好地解决留出法的问题，它对数据量的要求不高，并且样本信息损失不多。

交叉验证法把样本分为k个大小相同的互斥子集，选取一个子集作为测试集而其他作为训练集，那么就得到了k组训练集/测试集。



性能度量 (performance measure) 是衡量模型泛化能力的评价标准。

性能度量反映了任务需求，对于一个模型，使用不同的性能度量进行评判往往会得到不同的结果。这意味着模型好坏的是对的，一个模型是好的，不仅取决于算法和数据，还取决于任务需求。

$$E = \frac{1}{m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

回归任务常用的性能度量是均方误差

大图下，有代价函数评估分割的代价、最优性等

对代价函数的理解：

一个好的代价函数需要满足两个最基本的要求：能够评价模型的准确性，对参数 θ 可微。

（1）概况来讲，任何能够衡量模型预测出来的值 $h(\theta)$ 与真实值 y 之间的差异的函数都可以叫做代价函数 $C(\theta)$ ，如果有多个样本，则可以将所有代价函数的取值求均值，记做 $J(\theta)$ 。因此很容易就可以得出以下关于代价函数的性质：

对于每种算法来说，代价函数不是唯一的；

代价函数是参数 θ 的函数；

总的代价函数 $J(\theta)$ 可以用来评价模型的好坏，代价函数越小说明模型和参数越符合训练样本 (x, y) ； $J(\theta)$ 是一个标量；

（2）当我们确定了模型 h ，后面做的所有事情就是训练模型的参数 θ 。那么什么时候模型的训练才能结束呢？这时候也涉及到代价函数，由于代价函数是用来衡量模型好坏的，我们的目标当然是得到最好的模型（也就是最符合训练样本 (x, y) 的模型）。因此训练参数的过程就是不断改变 θ ，从而得到更小的 $J(\theta)$ 的过程。

例如， $J(\theta) = 0$ ，表示我们的模型完美的拟合了观察的数据，没有任何误差。

（3）在优化参数 θ 的过程中，最常用的方法是梯度下降，这里的梯度就是代价函数 $J(\theta)$ 对 $\theta_1, \theta_2, \dots, \theta_n$ 的偏导数。由于需要求偏导，我们可以得到另一个关于代价函数的性质：

选择代价函数时，最好挑选对参数 θ 可微的函数（全微分存在，偏导数一定存在）

这三个步骤的目标是大图分割算法的主要工作内容。

开发进度表

社区联络期 (June 25 - June 30)

这个阶段的任务是深入了解项目，并可能在此过程中解决一些可能会干扰进度的因素与预案。对所需的技术做一些研究，与社区开发人员和导师交谈，并改变一些技术步骤或计划。

编码阶段一 (July 1 - July 15)

编码阶段一有2周的时间，并在阶段结束后进行评估计划质量与风险。第一阶段的目标是

- ☐ 梳理课题目标结构，调研制定合理的实现方案

编码阶段二 (July 15 - Aug 30)

编码第二阶段有6周时间，阶段结束后进行可能的中期评估，有两个初步目标，至于其具体执行细节有待商酌。

- ☐ 完成大图分割算法分步工作：
 - ☐ 提供图分割代码
 - ☐ 小图下，提供采用或不采用的采样结果、训练精度度量、性能数据
 - ☐ 大图下，有代价函数评估分割的代价、最优性等
- ☐ 实现相关算法并整合到飞浆Paddle的线上仓库（可选）

编码阶段三 (Sep 1 - Sep 23)

编码第二阶段有3周时间，如果任务进行顺利，我们将产出系统化报告，和考虑将相关材料发表为会议论文。我们注意到社区中的一些issue与，我们可以考虑完成它们。

- ☐ 产出系统化报告
- ☐ 争取投递顶会paper

一旦实现了本个阶段的目标，我们会检查代码并修复前面代码中的任何已知bug。编写代码应该被文档化。然后我们的工作基本完成了，然后是最后的评估。

缓冲时间 (7 days)

有至少7天的缓冲时间，以防前几周发生的事情没有按计划进行。我认为一些工作可能会在两周内完成，所以我们实际上会进展得更快，并拥有更多的容错能力。

关于我

申请信息


名称:	房森 (Wirth)
Email:	wirth.fang@foxmail.com
Github:	github.com/FangSen9000
时区:	UTC+08:00 (China)
地区:	郑州, 中国
Education:	河南大学, 澳大利亚Victoria University计算机科学与技术双学位
Telephone:	+86 18143465655
CSDN blog:	Wirth's blog (Chinese)

Self-introduction

至于我，我是一个探路者，我热爱开源，非常享受GSoC和OSPP或者其他开源活动的氛围。是Apache的顶级项目APISIX的中选人与执行人。我在河南大学虚拟现实实验室工作由阎朝坤教授指导，热爱技术的落地实现，身经百战，并且很高兴在飞桨Paddle社区与吴志华导师一起讨论协商并且完成这个项目。我很高兴能参与这个项目，我对它的前景充满信心，即使在GLCC结束后，我也会继续为这个项目做出贡献。

参考资料

- [1] [图分割算法的概述](#)
- [2] [深度学习框架飞桨 \(PaddlePaddle \) 概述](#)
- [3] [最全综述 | 图像分割算法](#)

- [4] [图像分割经典算法--图割 \(Graph Cut、Grab Cut-----python实现 \) 图割算法](#)
- [5] [飞桨PaddlePaddle-源于产业实践的开源深度学习平台](#)
- [6] [PaddlePaddle/Paddle: PArallel Distributed Deep LEarning: Machine Learning Framework from Industrial Practice \(github.com\)](#)
- [7] [C++性能优化 \(九 \) —— TCMalloc](#)
- [8] [数字图像处理中常用图像分割算法有哪些？](#)
- [9] [【机器学习】代价函数，损失函数，目标函数区别](#)
- [10]  附件: 飞桨提案人个人扩展简历
- [11] [附件:demo-paddle: 这是我为Paddle项目存放Demo的地方\(github.com\)](#)