

硕士学位论文

题目 基于深度学习的裂缝图像分割及其在手机端上的应用 英文题目 Crack Segmentation Based on Deep Learning and the Application on the Phone 姓名 伍宇明 学号 51609020 所在学院 工学院 导师姓名 范衠 专业 电子与通信工程 2016年9月 答辩日期 2019年6月

汕头大学学位论文原创性声明和学位论文使用授权声明

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别 加以标注和致谢的地方外,不包含其他人或其它机构已经发表或撰写过的研究成果。 对本文的研究做出贡献的个人和集体,均已在论文中以明确方式标明。本人完全意识 到本声明的法律责任由本人承担。

作者签名: 伍宁明 日期: <u>2019</u>年 <u>5</u>月 <u>14</u>日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档,允许论文被查阅和借阅; 学校可将本学位论文的全部或部分内容编入有关数据库进行检索,可以采用影印、缩 印或其它复制手段保存和汇编论文;学校可以向国家有关部门或机构送交论文并授权 其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文,按照保密的 有关规定和程序处理。

作者签名: 伍宇明	导师签名:
日期: <u>2019</u> 年 <u>5</u> 月 <u>14</u> 日	日期: <u>2019</u> 年 <u>5</u> 月 <u>14</u> 日

摘要

裂缝检测是评估结构损伤程度的重要步骤,为了实现自动化裂缝检测,需 要把裂缝从图像中分割出来。随着计算机视觉的发展,特别是深度学习的发 展,有很多对裂缝图像分割问题的研究。但是当前的研究侧重点多是应用最新 算法提高裂缝分割的准确率,目前对于深度学习算法如何应用到端上进行实时 分割的算法研究较少。

本论文提出从裂缝图像分割模型建立到手机端上实现的整个流程框架,研 究重点侧重于模型压缩的框架提出。本文首先建立基于 Unet 的裂缝图像分割模 型,然后提出模型的压缩框架,包括蒸馏压缩和通道剪枝。通过模型压缩把模 型的参数量和运算量均大大降低后,通过 TensorFlow Lite 在手机端上可实现近 乎实时的裂缝图像分割。本文的创新点包括:针对图像分割网络的提出了模型 压缩的框架;在模型压缩框架中,针对于二分类提出了自己的蒸馏方法;实现 了手机端上的裂缝图像分割。

关键词:裂缝图像分割;模型压缩;蒸馏;通道剪枝;移动端神经网络

Abstract

Crack detection is a critical step to evaluate the damage of structures. For automatic crack detection, cracks need to be segmented from images. With the development of computer vision, especially the development of deep learning, there are many researches about crack segmentation. However, current researches mostly aim at improving the accuracy with applying state-of-art algorithms. There are few researches about how to apply crack detection algorithms on mobile or embedded devices.

In this work, we propose a framework for the procedure from building a crack segmentation model to applying it on the phone. First, a crack segmentation model based on Unet is built. And then the model compression framework is proposed, including distilling and channel pruning. After greatly reducing parameters and computation with model compression, we can implement crack segmentation on phone through TensorFlow Lite. The crack segmentation can be processed on phone in almost real time. The main innovative points of our work are the following: propose a framework of model compression for crack segmentation network; propose our own distilling method to the 2-class classification in the model compression framework; implement crack segmentation on phone.

Keywords: Crack Segmentation; Model Compression; Distilling; Channel Pruning; Neural Network on Phone

Π

目 录

摘 要	I
Abstract	II
目 录	III
第一章 绪 论	1
1.1 课题研究背景与意义	
1.2 国内外研究现状	2
1.3 主要研究内容	4
1.4 章节安排	5
第二章 深度学习模型理论	6
2.1 深度学习的概念和发展	6
2.2 卷积神经网络基本结构	7
2.3 经典卷积神经网络	9
2.4 常用的深度学习框架	
2.5 本章小结	
第三章 裂缝图像分割模型的构建	
3.1 裂缝图像分割目标	
3.2 Unet	
3.3 用 Unet 对裂缝图像分割问题建模	
3.4 本章小结	23
第四章 裂缝图像分割模型的压缩	24
4.1 模型压缩框架	24
4.2 蒸馏压缩	
4.3 通道剪枝	
4.3 通道剪枝 4.4 总体压缩结果	
4.3 通道剪枝 4.4 总体压缩结果 4.5 本章小结	
 4.3 通道剪枝 4.4 总体压缩结果 4.5 本章小结 第五章 裂缝图像分割在手机端上的实现 	
 4.3 通道剪枝 4.4 总体压缩结果 4.5 本章小结 第五章 裂缝图像分割在手机端上的实现 5.1 模型建立与模型压缩的程序实现 	

目	录
н	

5.2 手机端上的实现	41
5.3 本章小结	44
第六章 总结与展望	45
6.1 本文总结	45
6.2 未来工作展望	45
参考文献	47
致谢	52
攻读硕士学位期间主要的工作成果	53

第一章 绪 论

1.1 课题研究背景与意义

裂缝是反映结构受损情况的重要标志,一直以来,在道路交通以及土木建筑 领域,裂缝的自动化检测技术都受到学术以及工业的极大关注^[1]。以道路交通领 域的应用为例,专业人员可以从裂缝的长度、宽度和形状等对道路路面的当前状 况进行评估。这对道路系统的管理维护十分重要,在对道路进行实际的维护和修 复之前,准确获取当前道路的情况将有利于相关人员制订合适的维护方案,节约 成本。传统中,裂缝检测是由专业人员实地考察,人为判断裂缝的严重程度。但 是这种方法耗时耗力,因此在最近几十年来,已经逐步被半自动化和自动化检测 方法取代。

裂缝自动化检测的研究意义总体来说可以归结为以下方面^{[2][3]}。节省时间和 成本,相比于传统人工实地勘测的方法,借助于半自动化和自动化方法,研究人 员不再需要实地勘察判断,只需要相关仪器的数据收集和后期处理,使得人力成 本和耗时大大节省。而且因为不需要实地勘测,研究人员可以远离危险区域,所 以对研究人员本身更加安全。另外,用自动化辅助判断,可以减少决策的主观性, 从而减少人为错误。

裂缝自动化检测的相关研究主要涵盖三个部分^[4]:第一是裂缝数据收集。此部分研究主要致力于传感器和集成系统。传感器方面主要是便于装载在车上的各类传感器系统^[5],包括一维、二维光学相机,三维相机,激光,结构光等^{[6][7][8]},不同的传感系统获得的数据将在影响数据处理方法和最终裂缝检测结果。集成系统则主要致力于研究自动化机器人^[9]以及混合传感器系统^[10],提高系统整体的自动化程度;第二是传感器数据的处理,从传感器数据中处理得到可读信息。由于目前大部分研究都是基于相机数据,此部分研究方法主要是图像处理;第三是裂缝数据的解析^[11]。此部分研究主要是基于第二部分图像处理得到的结果,从而解析裂缝的长度、宽度和形状等信息,以及判断裂缝的危害等级,因而本部分研究常与第二部分相结合。本论文主要聚焦的是第二部分,结合当前前沿的图像处理

方法,从裂缝图像中还原出裂缝的像素级定位。

自 2012 年 AlexNet^[12]在 ImageNet 竞赛中得到了突出的效果后,深度学习被 广泛应用于图像处理任务中,并得到了突出的效果,包括图像识别、目标检测、 图像分割等领域。因此,在实际工程应用中,考虑用深度学习算法实现裂缝检测 的目标,即裂缝图像分割的任务。

另一方面,准确率越高的深度学习算法需要耗费越大量的计算资源,包括存储空间和运算时间,这限制了深度学习在移动和嵌入式端的应用。因此,如何压缩深度学习模型的大小,并加速其运算时间,也是具有挑战的一个问题。相对于深度学习在提升准确率上的学术研究,模型的压缩与加速目前尚未在领域内有较为权威的算法,因而具有探索意义。

1.2 国内外研究现状

由于裂缝检测的重要意义,一直以来都被学术以及工业界广泛研究。其中, 基于图像方法的裂缝像素级定位一直以来都是该领域研究重点和难点。一方面, 裂缝像素级定位对后续数据解析有重大影响,一旦从裂缝图像中定位准确,则后 续的裂缝长度、宽度等数据解析均很容易得到。另一方面,由于非均匀光照、复 杂路面纹理、阴影等干扰物存在等因素^[13],在实际识别中容易误识别或者引入噪 声,另外裂缝作为细长物体在像素上的难以识别(最小宽度只有一个像素),像 素级定位并不是一件容易的事情。

在基于图像识别的传统方法中,主要利用了裂缝的颜色特征和形状特征^[2]。 颜色特征是指裂缝像素在图像中比背景环境更黑,裂缝像素灰度与背景灰度相互 独立;形状特征是指裂缝是细长连续物体,每一小段具有不确定的朝向。在两种 特征中,颜色特征更为主要,主要是基于"裂缝是局部图像区域中最黑的像素" 这一颜色特征,尝试抑制噪声排除干扰并恢复裂缝。其它提出的图像方法也是围 绕此两种特征,尝试结合和改进阈值分割^[14]、边缘检测^[15]、数学形态学^[16]、区域 生长^[17]等等。

近年来的很多方法则是结合图像和机器学习算法,通过学习提取更加抽象的

特征,或是提出高级的图像特征,通过学习特征完成裂缝检测的模式识别任务。 CrackIT^{[18][19]}利用了平均值和标准差两种特征,通过无监督学习(聚类等)识别 出含有裂缝的局部小区域,并整合了多种图像处理方法,从裂缝小区域中恢复最 终裂缝形状,并对裂缝进行危害级别等的判断。平均值和标准差是常用的两种特 征,同时局部裂缝区域的识别是常用的处理方法。CrackForest^[3]利用随机结构森 林,通过学习得到裂缝的边缘结构表征子,利用此表征子特征对裂缝进行识别。 最小路径法是另外一种常用的检测方法,其原始模型是计算机基础问题中的最小 路径问题。Amhaz R.^[20]把图像中各个像素点建模为节点,像素点灰度之和以及平 均灰度建模为路径,从而寻找出像素点之间灰度最小的路径。

随着深度学习在图像中的广泛使用,近年来也有提出用卷积神经网络进行建 模的论文。Cha Y J^[21]提出用 VGG 网络对墙壁裂缝的区域进行识别。但是此方法 只能识别到含裂缝的小块区域,进行粗略定位,并未能进行像素级别的定位,尤 其对于网络状裂缝区域定位并不能反映问题。Zhang A.^[22]提出了自己的卷积神经 网络 CrackNet,对 3D 路面裂缝进行像素级识别。但此网络输入需要用方向提取 子进行预处理,卷积神经网络在这里的作用只相当于一个分类器,其特征提取的 功能被忽视,且网络的输入为整图,限于尺寸而扩展性较差。Zhang L.^[23]提出了 基于像素点分类的像素级识别。此方法在分类上表现出较为优异的性能,但是在 定义正负样本时利用了宽松的条件(标记像素十个像素以内均为正样本),另外 输入也用了大尺寸图片,导致最终输出结果的裂缝比人为标记得要宽得多。此外, 由图像领域对单个像素点的分类割裂了像素点之间的联系。以上方法均是针对针 对算法提高准确率的研究,目前针对端上应用的研究较少,多数是研究道路缺陷 的检测和识别,如 Maeda H.^[57]使用 SSD 网络完成路面的缺陷检测和识别,对于 实时的裂缝图像分割研究暂时没有。

另一方面,神经网络由于其占用的计算资源较大,包括存储空间和运算时间, 是制约其在实际应用中的瓶颈^[24]。因此,针对于模型的压缩和加速,也有很多相 关领域的不同方法,包括剪枝、量化、低秩分解和知识蒸馏。剪枝是对较不重要 的权重进行去除,如 Google 在 15 年提出的 Deep Compression^[25]以阈值评估权 值的重要性,去除小于阈值的权值。这类对单个权值进行分析的方法称为非结构

性剪枝,在实现时需借助稀疏矩阵储存才可以达到压缩的目的,同时并没有加速 作用。考虑到实现,更为实用的做法是考虑在某一维度上整体去除权值,如 channel pruning^[26]和 Thinet^[27]两篇论文均以特征图重构来评估权值的重要性, 或称特征图的重要性。其中 channel pruning 以 Lasso 回归(L1 正则化)来解 决误差最小化问题,而 Thinet 则是以贪婪方法解决;量化是以更少的计算机位 数(bit)来表示权重或者激活值。其中一种思路是权值共享,如同样在 Deep Compression 中,利用聚类的方式确定权值的聚类中心,以聚类中心替代原权值, 在存储时只需存储少量的权值聚类中心的索引达到压缩,另一种思路是把整个网 络的权值和激活值离散到土1和0^{[28][29]},可以大大降低储存并加速运算,更为普 遍的做法则是把 32 位浮点数表示的权值用更低位数的整型^[30]、定点数或浮点数 ^[31]表示:低秩分解是以更少的秩来表示矩阵或 Tensor 以及其运算^[32],大部分这 一类的方法是基于 SVD, 在矩阵运算时进行矩阵分解, 在 Tensor 运算时分解成 秩更少的多个 Tensor 运算,此类研究中用多少秩来表示原矩阵及 Tensor 是研究 重点。蒸馏则是把大模型的学到的知识通过迁移学习的方式指导小模型学习, Hinton 在 15 年提出的蒸馏方法^[33]思路是训练好的模型不仅具有类别信息,而且 具有类间信息,因而可以通过迁移学习提高小模型的准确率。

1.3 主要研究内容

本论文聚焦于裂缝图像分割这一工程实际问题,首先尝试把深度学习的图像 分割算法应用于裂缝图像分割,然后尝试应用卷积神经网络的压缩与加速技术, 对于此裂缝图像分割模型进行压缩。进一步地,把此小模型移植到手机端上,并 通过拍摄实际的裂缝图像验证其在实际工程问题上的有效性。主要研究内容如下:

- (1) 研究深度卷积神经网络在裂缝图像分割(Segmentation)的应用。
- (2)研究神经网络压缩算法的理论和应用,包括蒸馏(Distilling)和剪枝(Pruning),从空间、时间和准确率层面对比验证压缩算法的有效性。
- (3) 主要介绍电脑端的神经网络模型建立已经神经网络模型压缩算法的实

现。

(4) 主要介绍手机端的神经网络模型调用的实现。

1.4 章节安排

根据上面的主要的研究内容,本文的组织结构安排如下:

第一章:绪论。简述裂缝图像分割的背景意义,简述裂缝图像分割算法以及 神经网络模型压缩算法的研究现状,并要概述文章研究内容。

第二章:深度学习模型理论。介绍深度学习的基本理论与发展,卷积神经网络的基本组成结构,经典的卷积神经网络以及常用的深度学习实现平台。

第三章:裂缝图像分割模型的构建。首先介绍了用于图像的分割的经典模型 Unet,然后介绍基于TensorFlow实现Unet并用于路面裂缝图像分割的实验过程, 给出基于 Unet 的路面裂缝图像分割实验结果。

第四章:裂缝图像分割的模型压缩。针对裂缝图像分割这一具体问题,提出 了可应用的模型压缩算法框架。介绍了蒸馏算法及其思想,针对二分类下 sigmoid 的情况提出一种新的蒸馏方法,将其应用于裂缝图像分割。介绍了一种通道剪枝 的算法,将其应用于裂缝图像分割。给出以上模型压缩的实验结果。

第五章:裂缝图像分割的手机端实现。介绍了深度学习应用于手机端上的平台支持 TensorFlow Lite。介绍在此平台下,手机端上拍照或选图进行裂缝分图像割的应用开发和算法调用过程,给出实际应用结果

第六章:总结。该部分主要是对本文研究工作和实验的总结,同时也对未 来实验的展望。

第二章 深度学习模型理论

随着 AlexNet 在 2012 年的 ImageNet 竞赛上以极大优势胜出,深度学习的应 用得到迅速发展,广泛应用于医疗、教育、安防、汽车及新零售等各个领域,成 为人工智能的基本组成部分。然而,深度学习的基础算法——神经网络却是早在 上世纪 80 年代就被提出,30 年来曲折发展。相较于传统算法,深度学习在可解 释性方面有着一定的缺点,但这并不妨碍算法的应用以及显示出巨大的性能优势, 尤其在计算机视觉领域。下文我们将介绍深度学习的基本理论及发展。

2.1 深度学习的概念和发展

深度学习这个概念发展于人工神经网络。最早的人工神经网络可追溯到 1943年的人工神经元模型。如图 2-1 所示,其将神经元简化为三个过程:信号 加权、求和、非线性激活^[34]。1958年,感知机^[35]的概念被提出,模型的权值可通 过梯度下降方法从训练样本中学习得到。但 1969年的研究^[36]指出感知机只能够 解决线性问题,对于非线性问题无法处理,甚至连最简单的异或(XOR)问题都无 法解决。到了 20 世纪 80 年代,G.E. Hinton 引入了多层感知机和反向传播算法 ^[37],解决了感知机线性不可分的问题。1989年,LeCun 提出了用于识别手写数字 的卷积神经网络^[38],并得到较高的准确率。但是在这以后,由于神经网络缺乏可 解释性,深层网络存在梯度消失以及过拟合问题等,对于神经网络的研究热潮慢 慢冷却。



图 2-1 人工神经元模型

2006年,同样是 G. E. Hinton,提出了深度学习的概念,这一年也被称为深 度学习元年,。在论文中,Hinton提出无监督预训练对权值进行初始化+有监督训 练微调的神经网络训练方法^[39]。后来,从 2006 年到 2012 年,对于神经网络的研 究开始回暖并快速发展。到了 2012 年,AlexNet 提出,把 ReLU 非线性激活、 Dropout 等技巧应用于卷积神经网络中,赢得了当年的 ImageNet 竞赛并极大超 越了传统方法。深度学习的应用和研究至此之后爆发,成为人工智能的基本算法。 特别是计算机视觉,经典的卷积神经网络结构陆续被提出,包括 VGG-Net^[40]、 Inception-Net^[41]、ResNet^[42],到后来的轻量级网络 MobileNet^[43]、SqueezeNet^[44]、 ShuffleNet^[45]等,深度学习得到了极大的发展。

2.2 卷积神经网络基本结构

构成深度学习的主要算法是卷积神经网络,简称 CNN(Convolutional Neural Network),其主要由三种结构组成:卷积层、池化层和非线性函数。

卷积层的操作如图 2-2 所示,输入是原图像或者特征图,通过滑动窗口,输入矩阵的每一个与卷积核等大的区域与卷积核的对应元素相乘再相加,得到输出矩阵的一个元素。假设输入为*I*,卷积核为*K*,输出为*S*,卷积操作的数学公式表示为:

$$S(i,j) = \sum_{m} \sum_{n} I(m,n) K(i-m,j-n)$$
(2-1)

卷积层的核心是局部连接和权值共享。局部连接是相对于传统神经网络的全 连接而言,全连接层的每个权值均与每个输入节点相乘,而局部连接指的是每次 卷积核只与输入的局部进行卷积运算;权值共享指的是当卷积核滑动遍历所有数 据时,卷积核的权值对于所有输入数据不变。





池化层同样是以一个滑动窗口对局部区域数据进行计算,不同的是池化层没 有权值参数。当前最常用的是最大池化,即选取窗口内最大的值,其最初设计目 的是为了选取最大的特征,去除相对不重要的特征,减少运算并加速收敛。另一 种常用的池化方式是平均池化,即计算窗口内的平均值,当前多见于最后一层卷 积层之后,替换原来的全连接展开。

卷积层和全连接层只能表示线性运算,要使网络非线性化,得到更大的表征能力,需要引入非线性函数。当前最常用的非线性函数为 ReLU (rectified linear unit),而在此之前 sigmoid 较为常见,但是由于 sigmoid 函数的过渡区小,容易饱和,在深层的神经网络中容易梯度消失,实践证明 ReLU 的效果会更好。一般在最后一层仍会应用 sigmoid (多见于二分类或多标签分类问题中,多分类问题则多用 softmax),为了把数值映射回[0,1]。Sigmoid 与 ReLU 的数学表达式以及函数图像如下图所示:



图 2-3 sigmoid 及 ReLU 激活函数

除了以上三种基本结构或运算外,全连接层在早期的神经网络中也是被广泛 应用。全连接层即神经网络本层的每个节点都与下一层的所有节点相连,如下图 所示。全连接层网络结构即来源于早期的多层感知机,但由于其参数量大、容易

过拟合、难以训练等缺点,最新的神经网络多选择去除全连接层,改以平均池化 代替,以减少网络的参数量。



图 2-4 全连接层

2.3 经典卷积神经网络

如 2.1 节所述,深度学习在最近几年得到爆发性发展,在计算机视觉领域出现了很多经典卷积神经网络,得到广泛应用和被作为研究的基准,以图像识别方面最具有代表。

LeNet-5 是第一个经典的卷积神经网络,其一共包含 7 层,包括 2 个卷积层、 2 个下采样层、2 个全连接层和 1 个径向基函数的全连接层。如图所示,其中, 输入图片大小为 32*32,卷积层的卷积核大小均为 5*5,下采样用的是平均池化。 1998 年 LeCun 提出的 LeNet-5 的网络结构已经包含了卷积神经网络的所有基本 元素,并将其应用到字母识别,得到很好的效果,但由于当时其它算法也能达到 同样的效果,卷积神经网络并没有得到足够重视。





AlexNet 的提出则是深度学习爆发的标志。其网络结构包含 5 个卷积层和 3 个全连接层,以及最大池化层。AlexNet 应用于 ImageNet 竞赛,对卷积神经网络提出了以下技术点: 1)使用 ReLU 代替 Sigmoid 作为激活函数,一定程度上缓解了梯度消失的问题; 2)训练时使用 Dropout 随机去掉一部分神经元,减少过拟合; 3)使用最大池化代替之前使用普遍的平均池化; 4)提出 LRN 层(后被其它神经网络结构舍弃); 5)提出多种数据增强的方法抑制过拟合。从中可见 AlexNet 并没有对卷积神经网络结构作出决定性改变,更多是加入了很多 tricks,但也由于这些 tricks 的成功使用,使卷积神经网络相较于传统算法在准确率上显示出巨大的优势。



图 2-6 AlexNet 结构

VGG-Net 同样也没有对网络结构作出改变,但通过反复堆砌 3*3 卷积和 2*2 最大池化,把卷积神经网络的层数增加到 16 层和 19 层,同时也大大提升了准确率。VGG-Net 的主要技术点包括: 1)从理论上分析了两个 3*3 卷积核操作等价于一个 5*5 操作,但参数量得到减少; 2)实验证明了网络层数越深,性能越好; 3)应用了预训练模型来初始化训练模型的网络参数。由于其结构简单并且容易拓展,VGG-Net 得到了广泛的应用。

		ConvNet Co	onfiguration					
Α	A-LRN	В	С	D	E			
11 weight	11 weight	13 weight	16 weight	16 weight	19 weight			
layers	layers	layers	layers	layers	layers			
input (224×224 RGB image)								
conv3-64	conv3-64	conv3-64	conv3-64	conv3-64	conv3-64			
	LRN	conv3-64	conv3-64	conv3-64	conv3-64			
		max	pool					
conv3-128	conv3-128							
		conv3-128	conv3-128	conv3-128	conv3-128			
		max	pool					
conv3-256 conv3-256 conv3-256 conv3-256 conv3-256				conv3-256	conv3-256			
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256			
			conv1-256	conv3-256	conv3-256			
					conv3-256			
		max	pool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512			
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512			
			conv1-512	conv3-512	conv3-512			
					conv3-512			
		max	pool					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512			
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512			
			conv1-512	conv3-512	conv3-512			
					conv3-512			
		max	pool					
FC-4096								
FC-4096								
		FC-	1000					
		soft-	·max					

图 2-7 VGG-Net 组成

Inception Net 又称 GoogLeNet,是由 Google 提出的系列网络,包括 v1~v4^{[41][46][47][48]}。相较于前面提到的网络,Inception Net 将网络层数推进到上百 层以提高准确率,也提出了很多新的技术点:1)提出 Inception 模组,如图所示, 把不同大小的卷积核组合使用,包含了多尺度的思想;2)提出把 3*3 卷积核分 解成 3*1 和 1*3 卷积核以降低参数量的方法;3)提出了 batch normalization 的 方法,可以加速网络训练的收敛,并且有正则化的功能,可减少过拟合。在这些 技术点中,其中 batch normalization 在后来的网络中大量应用,具有良好的效果。



图 2-8 Inception 模块结构

ResNet 是由何凯明提出的第一个具有残差结构的网络。如图 2-9 所示,其通 过残差结构把输入层和输出层直接相加,解决的是深层网络中梯度难以传递从而 难以训练的问题。由于结构的简单实用,同样也被后来的研究广泛引用。



图 2-9 残差结构

2.4 常用的深度学习框架

除了算法理论之外,深度学习得以爆发式发展的另一重要原因是 GPU 技术 支持以及代码框架的发展,GPU 技术极大地减少了神经网络训练的时间,而深度 学习框架的完善则降低了研究人员在代码开发和维护中所花的时间,更能专注于 算法本身。本节将简述几个主流框架的特点,包括 TensorFlow、Caffe、Caffe2、PyTorch、MXNet 等。

TensorFlow 由 Google 开发并在 2015 年发布的开源框架。其使用数据流图 (Data Flow Graph)的形式进行计算,计算图中的节点代表数学运算,而数据则 以张量(tensor)的形式存储及交互。TensorFlow 基于 C++开发,提供 Python、 C++、Java 及 Go 等编程接口。其框架灵活,支持自动求导而不需要用户关注反向 传播求梯度的过程。支持分布式,可方便部署于一台或多台 CPU、GPU 机器中。 但其也有着不完美的地方,由于其框架的实现底层化,TensorFlow 的上层代码没 有统一到更抽象的调用方式,加上 TensorFlow 本身集成了大量包含机器学习的 功能,使得其使用变得复杂和繁琐。尽管如此,由于 Google 的号召力,TensorFlow 目前仍是使用人数最多且社区最庞大的深度学习框架。

Caffe 全称 Convolutional Architecture for Fast Feature Embedding,由当时还在 加州大学伯克利读 Phd 的贾扬清主导开发,是最老牌的深度学习框架之一。Caffe 同样是基于 C++开发,其优点和缺点都很明显。优点是高效运行,上手简单,其 网络结构是由文本文件而非代码定义,因而使用起来比较方便;缺点是不够灵活, 用户新定义网络需要从 C++源码中定义好其 CPU 及 CUDA 的运算规则,因而对于 应用扩展并不灵活。此外, Caffe 实际使用过程中安装过程麻烦,而且文档非常 缺乏。因而, Caffe 虽然在 TensorFlow 出现之前是很多论文的实现框架,但由于 缺点明显,已不再那么流行。

Caffe2 是贾扬清在加入 Facebook 之后主导开发的新框架。Caffe2 继承了大量 Caffe 的特性,针对解决了 Caffe 多年来在应用和部署方面的瓶颈问题,其性能非 常优异。Caffe2 主打高性能、模块化、轻量级,并且几乎全平台支持,因而更适 合用于生产环境。但是目前 Caffe2 的代码和社区都还并不成熟,同样缺乏规范化 文档,暂时未能替代 Caffe,也并未流行起来。

PyTorch 也是由 Facebook 发布的深度学习框架,且一经发布即引起大量关注。 PyTorch 的前身是 Torch,而 Torch 是基于 Lua 语言的,在深度学习由 Python 语言 主导的情况下,Torch 相对较为小众。PyTorch 继承了 Torch 简洁高效易用的特点, 并且在 Torch 的基础上对模块进行了重构,新增了自动求导系统,形成当下最流

行的动态图框架。动态图机制相对于 TensorFlow 的静态图有诸多优点,比如可以 方便修改图中的节点,同样的操作 TensorFlow 的接口用起来要晦涩很多。正是由 于这种简洁易用,目前 PyTorch 的使用人数逐渐增加,大有取代 TensorFlow 地位 之势。

MXNet 是由 DMLC(Distributed Machine Learning Community)开发的深度 学习库,目前已由亚马逊官方推荐。MXNet 的特点是支持非常多的编程语言封装, 包括 Python、R、C++、Go、JavaScript 等等。另一大特点是率先支持分布式运算, 且其分布式性能非常良好,而且内存和显存优化非常好。不过 MXNet 一直处于 不温不火的状态,同样是由于其缺乏文档,生态圈并不完善。

2.5 本章小结

本章主要介绍了与深度学习相关的知识,包括深度学习的发展,卷积神经网络的基本组成结构,经典的卷积神经网络,以及常用的深度学习框架。深度学习 源于神经网络,在几十年里经历了发展和低谷,最终在近几年爆发。深度学习的 核心卷积神经网络主要组成部分是卷积层、池化层和非线性激活函数。经典的卷 积神经网络包括 LeNet-5、AlexNet、VGG-Net、Inception Net 和 ResNet 等。常见 的深度学习框架有 TensorFlow、Caffe/Caffe2、PyTorch 和 MXNet 等。

第三章 裂缝图像分割模型的构建

3.1 裂缝图像分割目标

裂缝是评估结构体损伤程度的重要参考标准,各类结构如桥梁、路面、建筑 都需要定期评估维护,在此过程中需要测量裂缝的参数。传统方法是实地测量, 而随着技术进步,更多方法可辅助完成半自动化甚至是自动化的评估,其中一种 普遍的方法是通过相机获得裂缝照片后进行测量。本论文要解决的问题是把裂缝 像素从相机拍摄的裂缝照片中分割出来,如下图所示,从原有裂缝图像中分隔出 属于裂缝的区域,得到二值化图片,从而可以进一步通过标定等的方式进行长度、 宽度等测量来研究裂缝。



输入图像

目标输出

图 3-1 裂缝图像分割的目标

裂缝图像分割是一个特定场景下的图像语义分割问题,而分割模型的建立从 数学角度而言,即建立从输入图像到目标输出的映射关系,此问题可以看作是像 素级别的分类问题。具体地,设x为图像的一个像素点,可用坐标表示为(*x*, *y*)或 用坐标以及灰度值表示为(*x*, *y*, *l*)等,存在*f*使得:

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \end{pmatrix} 裂缝像素 \\ 0 & \mathbf{x} \end{pmatrix} 非裂缝像素 \qquad (3-1)$$

建立模型的目标就是要寻找合适的关系*f*。在传统图像方法中,*f*可以利用像 素点本身的特征,如阈值法利用裂缝颜色特征,即裂缝像素点比非裂缝像素点更 黑的特征将其分割;也可以是形状特征,即裂缝是细长条状的物体,需要利用像 素点周围的其它像素,写出领域特征的表达式,根据表达式将属于裂缝的像素点 分割出来。在卷积神经网络中,逐层的卷积会遍历每个像素,提取像素以及其领 域特征,从实质上而言可理解为卷积神经网络提取特征时融合了传统意义上的颜 色以及形状特征。

3.2 Unet

3.2.1 图像分割网络简介

在 2.3 节提到的经典卷积神经网络均为应用于图像识别的网络,这些网络结构奠定了深度学习在计算机视觉的发展基础。而在计算机视觉的其它细分领域,同样发展出了所在领域流行的网络结构或者方法框架,如目标检测的 Faster-RCNN^[48]、SSD^[49]和 YOLO^[50],人脸识别的 DeepFace^[51],人体姿态检测的 OpenPose^[52]。本论文所研究的图像分割领域也有其流行的网络框架及方法,包括 FCN^[53]、Unet^[54]、DeepLab^[55]、Mask R-CNN^[56]等。

FCN 网络全称 Fully Convolutional Network,是深度学习应用于图像分割的首 创方法,后来的深度学习分割方法都是基于 FCN 所设计的网络架构。FCN 将网络 最后的全连接层换成了卷积层,用于得到图像与其分辨率维度相关的特征图,然 后从特征图中,通过反卷积的方式获得与原图像分辨率相等的像素级分类图。后 来输入和输出都是等分辨率图像的操作也被称为 image-to-image 变换。

Unet 是基于 FCN 的基础上改进,于 2015 年被提出,提出背景是为了解决细胞膜的分割问题,被证明在多种生物图像分割任务中有良好性能,至今仍旧被广泛应用。Unet 与 FCN 的主要区别是在下采样阶段和上采样阶段采用了对称的拓扑结构,呈现 U 字形,另外 Unet 尝试把浅层的特征图连接到深层的特征图组成新的特征。对于 Unet 的结构将在 3.2.2 小节进一步介绍。

DeepLab 想要解决场景分割等更加复杂的问题,因而提出了空洞卷积(Atrous Convolution)、ASPP(Atrous Spatial Pyramid Pooling)模块、全连接的 CRFs (Conditional Random Field)等,分别是为了下采样带来的空间分辨率信息减少问题,多尺度的物体问题,和分类神经网络中空间不变性带来图像分割中的像素

分类不准确问题。

Mask R-CNN 与上述方法有所不同,其基于目标检测的 Faster R-CNN 架构, 在分类和定位之外新增了一个分割分支用于做物体的实例分割,相对于语义分割, 实例分割不仅可以分出物体的类别,还可以分出同一类别的不同个体。

本工作选择 Unet 作为基本网络结构的原因如下: 1)应用的问题的相似性。 Unet 的提出是为了解决电子显微镜下细胞分割的问题,细胞的边界与裂缝的特 征相似,几何特征是细长而颜色特征单一;2)合适的网络复杂度。考虑到要往 手机端上移植,过于复杂的网络会大量消耗手机的资源,甚至当前的平台支持也 未能完整支持而带来许多未知问题,而过于简单的网络会带来准确率上的损失未 能实际解决问题,因而需要考虑一个复杂度适中的网络。综合上述两点原因,结 合本文要解决的问题,裂缝图像中裂缝是细长而颜色单一物体,其在图像中的宽 度变化不大,而沿长度方向具有空间不变性,即截取任意一段裂缝图像其特征相 似,因而不需要 DeepLab 提出的复杂结构,另外裂缝图像中不需要明确区分裂缝 的个体,因而也不要 Mask R-CNN 的功能结构。此外这些网络的复杂的参数结构 也无法移植到手机端上。我们早前的另一个工作提出把裂缝分割问题看成是多标 签分类问题来解决,虽然网络结构简单,但由于需要提取图像小块来计算,消耗 时间过长也不满足端上的实时性要求。为了达到时间限制和合理的特征复用,更 倾向于使用端到端的方法。综合上述考虑因素,Unet 是较为合理的选择。

3.2.2 模型结构

首先介绍 Unet 的基础 FCN 的网络结构,如下图所示,FCN 把分类卷积神经 网络改成分割神经网络所做的改变是把最后的全连接层去掉,改成卷积层,同时 为了从特征图的小分辨率恢复到原图的分辨率大小,FCN 也引入了反卷积结构。 反卷积又称上卷积或解卷积,其操作与普通卷积类似,区别是在卷积的输入中插 值,从而卷积得到的特征图分辨率增大。在 FCN 里插值操作是传统方法里的插值 方法如双线性插值等,后来的反卷积考虑到实现效率以及卷积权值可学习的特点, 更多是插值补 0。此操作是深度学习里定义的反卷积,并非卷积的实际逆运算(实

际的卷积逆运算有另外的推导公式),只能恢复原图大小,但不能恢复原数据。



图 3-2 FCN 结构

Unet 是在全卷积神经网络 FCN 的基础上提出改进,其网络结构如图所示, 包含了卷积层、池化层和反卷积层。在下采样阶段,Unet 的结构与普通的 CNN 相似,通过两层卷积操作+一层最大池化操作这样的设计使图像分辨率下降压缩; 在上采样阶段,通过反卷积操作还原图像的分辨率。



图 3-3 Unet 结构

相对于 FCN, Unet 的主要变化有两点,一个是 FCN 只通过一次反卷积操作 来还原原图的分辨率大小,而由于深层的卷积图的感受野很大,使得分割结果在 像素级上尤其边界的准确率不高,同时编码和解码过程不对称,而 Unet 不再是 只通过一次反卷积来还原,而是把网络设计成下采样与上采样对称的结构,这样 每次上采样在上一个特征图上的感受野都不至于过大,多次上采样的结果有利于 像素级上的分类结果更准确,同时编码和解码过程得以对称。另一个是把浅层的 网络特征图与深层的语义特征图连接起来,组成新的特征。这是因为随着网络加 深,深的语义层具有更抽象的特征,便于分类,但是感受野也会越大,物体边缘 的像素分类也会不准确。为了解决深语义层的感受野过大,导致像素级定位的不 准确,Unet 在操作上把浅语义层的特征图连接到深语义层的特征图,从而实现 用浅语义层做定位,用深语义层做分类的思想。

3.2.3 训练细节

针对应用的场景问题, Unet 在训练策略上也提出了一些改进方法。Unet 在 逐点上计算 softmax 损失, 在 Unet 所应用的图像分割问题中,类别之间存在着 严重不平衡的问题,边界像素点明显少于物体内部像素点。为了解决此问题,突 出边界像素的重要性, Unet 在训练时的损失函数采用了带权重的 softmax 交叉 熵,公式为:

$$E = \sum w(\mathbf{x}) \log(p(\mathbf{x}))$$
(3-2)

x表示每个像素位置, $p(\mathbf{x})$ 表示 softmax 输出的每个像素位置的值, $w(\mathbf{x})$ 为:

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{\left(d_1(\mathbf{x}) + d_2(\mathbf{x})\right)^2}{2\sigma^2}\right)$$
(3-3)

 $w_c(\mathbf{x})$ 是对应于每个类别出现频率的权值图, d_1 和 d_2 分别是离最近和第二近类别的边界的距离, w_0 和 σ 是超参数。在训练前,通过此公式预先计算好每张图片对应的权值图,再代入交叉熵公式训练。

在 Unet 的论文中同样提到,对于网络权值的初始化也很重要,对于在初始 化网络参数时, Unet 也参考了经典卷积神经网路中的权值初始化方法,用均值 为 0,标准差为√2/N的高斯分布来初始化网络,其中N代表对应卷积核权值的个 数。

3.3 用 Unet 对裂缝图像分割问题建模

3.3.1 数据预处理

本工作首先用 Unet 来对裂缝图像分割问题建模。在选择测试数据方面,本 工作选取了反映北京道路裂缝情况的公开数据集 CFD,共有 118 张图片,图片分 辨率为 320*480。为了使结果更加客观稳定,实验中运用了 k 折交叉验证的做法, 即把数据集平均分成 6 份,前 5 份每份为 20 张,第 6 份为 18 张。每次取其中 5 份用于训练,剩下 1 份用于测试得到评估结果,6 份数据均测试完成后取其平均 结果。

一般而言,输入到神经网络的图像数据需要预处理,将其归一化,能加速收 敛以及抑制过拟合。在这里我们采用了对于图像数据最简单的归一化方式,具体 操作为:

$$y = \frac{x - \mu}{\sigma} \tag{3-4}$$

式中的x和y分别是归一化操作的输入和输出,µ和σ分别是均值和标准差,通常 根据经验或者取统计值,在这里因为图像数据的范围是[0,255],µ和σ均取为 127.5可保证数据得到归一化。

3.3.2 网络及训练的实现

将 Unet 应用于裂缝图像分割,在网络结构上,Unet 的完整拓扑结构被保留下来,但为了计算方便以及结合实际实验的效果,我们对结构上的细节作了一些修改。在 Unet 原文里卷积操作均不补 0,因而每次卷积会带来分辨率的稍微下降,这里为了计算方便,规定卷积运算不降分辨率,每次卷积操作均在边缘补 0,分辨率的降低全部由池化层实现,避免了实现时可能发生的维度不匹配的冲突。另外,由于裂缝图像分割只需分割裂缝和背景两类,实验中把网络最后一层的softmax 简化成 sigmoid,可以稍微减少整体参数量,并且实验证明,sigmoid 代替 softmax 不会带来准确率的损失。由此得到裂缝图像分割算法的电脑端模型。

在训练方面,借鉴 Unet 原文的带权重的 softmax 交叉熵,我们采用了带权

重的 sigmoid 交叉熵作为损失函数,通过调节正负样本项的权重来补偿裂缝图像中的正负样本不平衡。同样的,由于只有两个类别,且裂缝作为细长物体,最小宽度只有一到两个像素,边界不明显,因而我们简化了 Unet 原文根据与边界距离计算权值的方法,而直接改成手动调整,具体公式如下:

$$l = -\frac{1}{N} \sum_{i}^{N} \left(y_{i} \ln \hat{y}_{i} + \alpha (1 - y_{i}) \ln (1 - \hat{y}_{i}) \right)$$
(3-5)

其中*i*表示像素点,*N*表示像素点总数,*y_i*表示像素点的 label,当*y_i* = 0时表示非裂缝像素点,当*y_i* = 1时表示裂缝像素点,*ŷ_i*表示模型对像素点的预测输出,*α*表示负样本相对于正样本的权重,*α*越大则负样本权重大,训练得到的模型精准率高而召回率低,*α*越小则反之,经过反复实验得到,取*α* = 0.7结果最为合适。其它的超参数包括取 batch size 为 1,使用 Adam 作为优化方法,初始学习率为0.0001,训练的 epoch 数为 50。在训练中,我们发现加入正则化项反而会降低分割的准确率,因而不加正则化项,也正如 Unet 原文描述的那样。

3.3.3 裂缝图像分割结果

模型的部分分割结果如图所示,其中 ground truth 为人工标记图片,图像分割结果已十分接近于 ground truth。



原图



分割结果



Ground truth



为了量化评估模型及算法的有效性,我们使用精准率*Pr*,召回率*Re*,以及*F*1 作为评估指标,其数学定义如下:

$$Pr = \frac{TP}{TP + FP} \tag{3-6}$$

$$Re = \frac{TP}{TP + FN} \tag{3-7}$$

$$F1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \tag{3-8}$$

其中*TP、FP、FN*分别表示像素点中的 True Positive (预测为真真值为真)、False Positive (预测为真真值为假)和 False Negative (预测为假真值为真)。对于裂缝 和背景两个类别均可用上式计算,但由于我们只关心裂缝区域,评估时以上均 只是针对裂缝类别计算。

在很多研究裂缝图像分割的论文中,因为裂缝像素存在过渡区,通常在评估时允许有一定像素的误差。我们以*Pr*₂,*Re*₂和*F*1₂表示允许 2 像素误差下的评估,

实验得到以下的评估结果:

交叉验证集	Pr	Re	F1	Pr ₂	Re ₂	F1 ₂
1	0.6766	0.7036	0.6799	0.9195	0.9146	0.9129
2	0.7113	0.6879	0.6862	0.9265	0.8822	0.8949
3	0.5242	0.6703	0.5736	0.8603	0.8251	0.8322
4	0.702	0.6735	0.6798	0.9398	0.9153	0.9257
5	0.7397	0.6876	0.6973	0.9267	0.8726	0.8899
6	0.7408	0.7333	0.7291	0.9721	0.886	0.9237
平均	0.6824	0.6927	0.6743	0.9242	0.8826	0.8966

表 3-1 Unet 裂缝图像分割结果

从实验结果看,算法本身具有不稳定因素,选取不同的训练集和验证集数据 得到的结果差异很大,而且召回率与精准率之间有相对矛盾之处,在同一模型下 选定不同置信度时,召回率高则精准率低,反之亦然。但F1的定义中权衡了召回 率和精准率的关系,且通过反复实验可得到不同的验证集的平均F1相对稳定,因 而我们主要评估的标准也是所有验证集的平均F1。

允许像素误差的评估方式有工程指导意义,但为了论证像素级分割算法的效 果差异,论文以下的实验仍主要以原始的精准率*Pr*,召回率*Re*,以及*F*1的原始定 义进行评估。

3.4 本章小结

本章我们主要介绍了裂缝图像分割模型的建立过程。首先我们介绍了裂缝图像分割的目标,然后我们介绍了在实现这一目标上,深度学习领域中图像分割的相关方法,以及选择 Unet 作为基本框架的考虑。接着我们详述了 Unet 的网络结构以及训练细节,给出了本文使用 Unet 作为裂缝图像分割方法所作出的修改。最后我们给出使用 Unet 作裂缝图像分割的效果图和评估结果。

第四章 裂缝图像分割模型的压缩

4.1 模型压缩框架

由于原始网络中包含庞大的参数量以及运算量,直接把网络移植到手机端上 不可实现。一是因为手机的运行内存有限,在加载庞大的网络参数时容易崩溃, 且存储空间也极大占用;二是因为测试时计算量的庞大使得单张图片的分割时间 极长,不能满足实际应用的需求。因此,对网络进行压缩和加速,从参数量和运 算量方面优化算法,是手机端算法实现的必不可少的过程。

本论文经过实验验证,提出了以下模型压缩的框架:



图 4-1 模型压缩框架

为了评估压缩和加速的效果,引入神经网络的参数量和计算量的计算方法, 参数量主要为卷积以及反卷积层的权值个数,单个卷积层的计算公式为:

$$Params = k_w \cdot k_h \cdot C_{in} \cdot C_{out} \tag{4-1}$$

k_w, *k_h*表示卷积核的大小, *C_{in}*表示卷积层输入的通道数, *C_{out}*表示卷积层输出的通道数。参数量反映的是神经网络占用的存储空间。

理论计算量以 FLOPs 表示,把一次乘加运算定义成一个 FLOPs,只考虑乘法, 不考虑加法和加偏置,当卷积核的遍历步长为1时,单个卷积层的 FLOPs 计算公 式如下:

 $FLOPs = k_w \cdot k_h \cdot C_{in} \cdot C_{out} \cdot H_{out} \cdot W_{out}$ (4-2)

kw, kh, Cin, Cout的含义与上面描述一致, Hout, Wout表示输出卷积层的分辨率 大小。计算量反映的是神经网络推理的时间。

4.2 蒸馏压缩

在本文所提出的压缩框架的第一部分,应用了蒸馏的方法进行压缩。蒸馏是 迁移学习的方法,其特点是教师模型是个准确率高的大模型,而学生模型则没有 指定大小,因而可以设计网络结构更小的模型以达到压缩的目的。在本实验中, 我们使用了跟 Unet 同拓扑结构的小模型,然后通过蒸馏方法提高其准确率。

4.2.1 同拓扑结构压缩

在设计卷积神经网络时, 网络的层数、不同功能的层的不同组合、每层的卷 积图数, 均是需要考虑超参数设计。层与层之间的组合跟神经网络的功能有关, 而要把模型压缩得更小而不改变其功能, 最直接的想法就是保留模型的拓扑结构, 按比例地减少其特征图通道数, 如在 MobileNet 的设计中, 尝试了基础通道数比 例为 0.75, 0.5 和 0.25 的实验。更直接地, 也可以选择减少其层数, 从而大幅度 地减少模型的参数量及运算量。在本工作中,这种策略用于第一步的模型压缩加 速。经过观察, 原 Unet 模型共含有 4 次的池化降维, 每次池化降维后接下来的 卷积均把特征图通道数翻倍。实验中保留了 Unet 网络的这个特点, 同时考虑到 裂缝图像的特征较为明显简单, 尝试了把池化降维的次数减为 2 次, 同时把初始 的特征图通道数从 64 减为 8, 得到如图所示的缩小版 Unet 小模型。下图中, 包 括卷积、反卷积、池化和通道连接均与 Unet 原网络保持一致, 只有层数和通道 数做了减少。



图 4-2 压缩后的 Unet 结构

在选取最佳的网络结构时,设计了相应的实验,比较了减少层数和减少通道数带来的影响。由于 3.3.3 节提到的网络结果不稳定问题,我们只用平均的 F1 进行评估,其它指标作为参考也在此列出。

保持基础通道数为 64, 只减少层数, 除网络拓扑结构外其它超参数不变。结果如下所示:

降维次数	Pr	Re	F1	Pr ₂	Re ₂	F1 ₂
4 (Unet)	0.6824	0.6927	0.6743	0.9242	0.8826	0.8966
3	0.6718	0.6865	0.6653	0.9208	0.8614	0.8829
2	0.6729	0.6545	0.6420	0.9100	0.8263	0.8491

表 4-1 不同降维次数的性能对比

从实验数据中可以看出,通过比较平均的F1,网络性能与其降维层数的相关 性趋势可以较为明显地看出,所用降维层数越少,网络性能越差。

保持层数不变,即保持降维次数为4,只减少通道数。实验发现由于网络的结构改变,带来最佳超参数也会发生改变,当通道数少时使用小学习率作为初始学习率会使网络性能变得很差,而在通道数大时使用大学习率则让网络陷入局部最优,输出全为0而无法训练。因此我们对不同网络使用不同学习率,对于通道数为 64 和 32 时的网络训练使用 1e-4 作为初始学习率,对于其它通道数的网络训练使用 1e-3 作为初始学习率。为了尽可能公平比较,除了学习率外其它超参数保持不变。实验结果如下所示:

通道数	Pr	Re	F1	Pr ₂	Re ₂	F1 ₂
64 (Unet)	0.6824	0.6927	0.6743	0.9242	0.8826	0.8966
32	0.6692	0.6944	0.6662	0.9243	0.8623	0.8834
16	0.6694	0.6670	0.6525	0.9183	0.8428	0.8666
8	0.7052	0.6595	0.6554	0.9310	0.8480	0.8781
4	0.6375	0.6957	0.6494	0.8981	0.8702	0.8760

表 4-2 不同通道数的性能对比

同样的,从实验数据中可以看出,除了通道数为8和16的结果中存在一定的数值不稳定外,网络所用特征图通道数与网络性能呈现明显的相关关系,通道数越少,网络性能越差。

除了准确率以外,为了评估缩减网络带来的收益,我们计算了上述所有减少 通道数和减少层数带来的参数量和运算量的影响,结果如下:

	参数量(M)	存储大小(MB)	GFLOPs
降维4次,64通道	30.99	118.21	128.07
降维3次,64通道	7.69	29.35	97.56
降维2次,64通道	1.86	7.10	67.04
降维4次,32通道	7.75	29.56	32.09
降维4次,16通道	1.94	7.39	8.07
降维4次,8通道	0.48	1.85	2.04
降维4次,4通道	0.12	0.46	0.52
降维2次,8通道	0.03	0.11	1.09

表 4-3 不同压缩方式的参数量和运算量对比

观察实验及计算结果,降维3次(减少一次下采样及所需的层数)与通道数 从 64 个降低到 32 个时的参数量相当,但降维3次的运算量远远大于通道数减 少的运算量。同样的,降维2次与通道数降低到16个时的参数量相当,但减少 通道数带来的运算量下降相比于减少层数大很多。分析其原因是运算量与特征图 分辨率正相关,减少初始的通道数可以降低分辨率最高的特征图的运算,因而对 加速效果明显。同时可以看出,减少通道数带来的准确率损失也比减少层数的准 确率损失大,因此这是一个需要权衡准确率和空间时间的地方。在选择最终的网 络压缩策略时,考虑了实际手机的运存和储存,发现单一方式的减少降维次数或 者减少通道数均未能达到实际需求,网络会因为运行内存占满而崩溃闪退,同时 也考虑到了 4.3 节所述的通道剪枝的占用资源影响。经过反复实验权衡,最终选择大幅度减少了通道数以获得最佳参数量和运算量下降,同时也选择了对层数减少,通过下一节引入的蒸馏方法来提升准确率。具体的,我们选择了降维 2 次,通道数选为 8 个,其最终参数量和计算量也在上表中列出。

4.2.2 Sigmoid 蒸馏

层数以及特征图数的减少必然会带来准确率的损失,因此在这一步骤中引入 蒸馏的方法提高小模型的准确率。蒸馏的概念是由 Hinton 提出,在其论文中提 到,一个经过训练好的模型不仅仅能学到类别信息,而且可以学习到了类间信息。 比如,把一辆宝马车错判成拖拉机的概率肯定比把其错判成萝卜的概率要高。蒸 馏的提出想要解决的问题是,把一个训练良好的教师模型的类间信息迁移到一个 学生模型上。其核心的公式表示为:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{4-3}$$

其中*z_i*, *z_j*表示神经网络 softmax 层前的 logits, *T*代表"温度", 是一个超参数, 用于调节类间信息的平滑程度, 当*T* = 1时, *q_i*就是 softmax 的输出, 当*T* > 1时, softmax 输出的类间信息会被平滑, 教师模型的输出可作为软标签指导学生模型训练。由于学生模型的结构及大小并无特殊限制,选取一个小模型作为学生模型,可以得到压缩的效果。

原论文所提出的蒸馏方法是针对于 softmax 作为函数的,但是对于本论文所 研究的裂缝图像分割问题,使用 sigmoid 激活函数作像素级二分类得到的实际效 果更好。为了能够应用蒸馏,本论文在理解了蒸馏的核心思想后,针对二分类下 sigmoid 作为最后一层激活函数,提出了自己的蒸馏方法,称为 sigmoid 蒸馏或二 分类蒸馏,公式如下:

$$l_d = \frac{1}{N} \sum_{i=1}^{N} \alpha y_i \cdot \max(0, z_i - q_i) + \beta (1 - y_i) \cdot \max(0, q_i - z_i)$$
(4 - 4)

 l_a 为使用蒸馏方法所附加的损失函数项, y_i 为每个像素点的标签,取值为0或1,

N为像素点个数, *z_i*, *q_i*分别是教师模型和学生模型过 sigmoid 层前的 logits 输出, α,β是超参数,调节正负样本项的权重影响。此公式所表示的含义为,当标签为 1时,教师模型的 logits 输出如果比学生模型大,则提供一个损失加速模型学习; 反之,当标签为0时,如果教师模型的 logits 输出比学生模型小,同样提供一个 损失。当真实标签也同时存在时,以*l*表示常规训练的损失,将其也加入损失函 数中,如蒸馏的论文描述,蒸馏训练时总损失为:

$$l_{total} = l + l_d \tag{4-5}$$

本论文提出的蒸馏方法仍是借鉴于 Hinton 的类间信息迁移。由于问题只存 在两个类别,此处做了一个假设:从A类到B类的特征是连续变化的,而不是两 个极端。因而,假设训练好的二分类模型具有两个类间的过渡信息,则最终输出 的 logits 离原点的远近程度反映了类间特征的明显程度,此处的蒸馏目的是要把 两类间的特征过渡从教师模型迁移到学生模型,使得学生模型更具判别能力,提 高准确率。此假设有一定的普适性,也有一定的局限性,比如要分类狼和狗,由 于两个物种临近,部分品种的狗与狼的特征相像,可认为是特征存在连续变化, 让分类器学习到此类信息有助于更好地分类;但是如果是分类男人和女人,由于 特征是突变的,此方法的假设将失效。

在本论文要解决的裂缝图像分割问题中,经过实验,非裂缝像素点存在特征的连续性(即对于不同的非裂缝像素点分类难度不同),但裂缝像素点的特征可认为是突变的,因此,实验中取公式中的α = 0,β = 1。

为了验证裂缝分割中二分类蒸馏方法的有效性,我们实验了 4.2.1 小节所有 小模型的蒸馏训练和无蒸馏训练,以 3.3.3 小节中训练得到的 Unet 原模型为教师 模型,用 F1 指标来评估,结果如下:

	无蒸馏	蒸馏
降维3次	0.6653	0.6685
降维2次	0.6420	0.6541
32 通道	0.6662	0.6736
16 通道	0.6525	0.6704
8 通道	0.6554	0.6663
4通道	0.6494	0.6509

表 4-4 不同压缩方式下蒸馏与无蒸馏结果比较



图 4-3 不同压缩方式下蒸馏与无蒸馏结果比较

结果表明,所有的蒸馏训练得到的模型性能均好于无蒸馏训练,说明所提出的 sigmoid 蒸馏方法有提高学生模型性能的效果。此步骤没有直接压缩模型,而 是通过提高设计得到的小模型的性能来间接达到模型压缩的效果,通过设计同拓 扑结构压缩和蒸馏,我们得到参数量下降 1000 倍,100 倍的小模型。

4.3 通道剪枝

从以上步骤得到同拓扑结构的小模型后,考虑是否可以进一步压缩模型,这 里考虑使用的是通道剪枝。剪枝的目的是为了去除卷积神经网络中不重要的权值, 减少网络大小。这里基于一个假设,即卷积神经网络的权值中存在着冗余。剪枝 分为结构剪枝(奇异剪枝)和非结构剪枝(非奇异剪枝),区别在于在剪枝时需 不需要考虑卷积核的维度,如下图所示。在下图中,除了 fine-grained pruning 是 非结构剪枝,其余情况均为结构剪枝。通道剪枝的本质是 filter-level 剪枝,对卷 积核的输出维度进行裁剪,是属于结构剪枝的一种。



图 4-4 不同的剪枝方式

具体地,本文所用到的剪枝方法是来源于论文,其主要思想是用 Lasso 回归 (L1 正则化)来选择使卷积层输出的重构误差最小的输入通道。如图所示,要裁 剪 B 的特征图通道数,则裁剪从 A 到 B 的卷积权重的输出维度,同时相应地从 B 到 C 的卷积权重的输入维度也可以得到裁剪;在选择裁剪的特征图时,需要最小 化重构 C 的误差,用公式表示为:

$$\arg\min_{\beta} \frac{1}{2N} \left\| Y - \sum_{i=1}^{c} \beta_i X_i W_i^T \right\|_F^2 \tag{4-6}$$

式中, $\|\cdot\|_{F}$ 表示 Frobenius 范数, 即矩阵的二范数; *X*,*Y*,*W*分别表示转成矩阵形式的输入、输出和权重, 对应于图中的 B、C 和 W, *c*表示输入矩阵的特征图维度, 即 B 的特征图数; β 表示单个特征图的重要程度, 当 β = 0时, 此特征图可以安全地去除; *N*表示对样本的采样数。



图 4-5 通道剪枝

上述问题是一个 NP-hard 问题,为了求解此问题,加入了 L1 正则化项,把问题重新描述为 Lasso 回归问题:

$$\arg\min_{\beta} \frac{1}{2N} \left\| Y - \sum_{i=1}^{c} \beta_i X_i W_i^T \right\|_F^2 + \lambda \|\beta\|_1 \tag{4-7}$$

通过最优化这个式子即可选出最不重要的特征图通道,从而将其裁剪。

以上是单个卷积层剪枝的方法,在对整个网络模型进行剪枝时,选择要剪枝 的层遵循一定策略,只对当前特征图前后都是卷积操作的层进行通道剪枝,对特 征图后是池化以及反卷积的特征图不进行剪枝。这种选择主要是基于准确率和实 现难度的考虑。后接池化的特征图同时连接到后面反卷积的特征图,对这部分特 征图进行剪枝需要同时考虑前后部分产生的重构误差,即剪枝带来的准确率损失 风险增大,同时实现起来也不容易。类似的,后接反卷积的特征图需要用于图像 分辨率增大的重构,因而也需要保持完整的特征以降低准确率损失。

在本论文工作中,为了方便和统一,统一取剪枝率为 0.5,即对于要剪枝的特征图统一剪枝一半。为了实验验证通道剪枝的有效性,比较原模型(通过蒸馏得到的同拓扑结构小模型)、剪枝得到的模型、剪枝得到的模型结构直接训练的三个模型结果。由于通道剪枝的实现需要保存网络前向过程中的特征图,占用很大的显存和内存资源。受到资源限制,实验条件设为降维 2 次,通道数取 16/8/4,同样只用 F1 来评估,同时列出其参数量和运算量便于比较,结果如下:

通道数	数			剪枝模型			从头训练
	F1	参数量/k	运算量/M	F1	参数量/k	运算量/M	F1
16	0.6529	117.09	4257	0.6427	60	1982	0.6496
8	0.6456	29.34	1086	0.6446	16	509	0.6274
4	0.6210	7.41	283	0.6336	5	130	0.6140

表 4-5 不同模型的剪枝结果比较

从实验结果来看,剪枝模型大体比直接从头训练的模型要好,并且剪枝模型 相比于原模型的准确率下降在可接受的范围内,而参数量以及运算量下降了近两 倍,因而剪枝是有效的。事实上在实验过程中发现剪枝的结果相对来说不那么稳 定,甚至有时候会剪枝失败。鉴于通道剪枝可以把参数量和运算量下降两倍,作 为提出的框架我们保留通道剪枝作为模型压缩的一部分,此部分尚有很大的提升 空间。从参数量及运算量的下降情况来看,实际的模型压缩以蒸馏压缩的效果为 主。

4.4 总体压缩结果

这一节我们从网络的实际输出分割结果来直观感受模型压缩对网络带来的 影响。



原图

Unet 输出



蒸馏模型输出

剪枝模型输出

图 4-6 CFD 数据集 No.13 模型输出比较



原图





蒸馏模型输出

剪枝模型输出







Unet 输出



蒸馏模型输出

剪枝模型输出

图 4-8 CFD 数据集 No.93 模型输出比较

结合图 4-6、图 4-7 和图 4-8,可以得到以下结论:1) Unet 原模型的输出已 基本能分割出主要裂缝,且引入较少的噪声;2)蒸馏模型和剪枝模型的输出存 在分割裂缝不完整以及引入噪声等问题,但并未有统一的趋势。对于部分图片而 言,蒸馏模型输出裂缝完整而噪声多,剪枝模型输出裂缝不完整而噪声少,但对 于另一部分图片而言,可能出现截然相反的结果。因而需要看 F1 指标才可以判 断模型的性能好坏。结合之前的实验,整理得到应用框架下的结果如表 4-6 所示。 从结果可以看到,最终的剪枝模型对比于原 Unet 模型,F1 下降可以在 3%以内。

	Pr	Re	F1	Pr ₂	Re ₂	F1 ₂
Unet	0.6344	0.7471	0.6717	0.9114	0.8831	0.8908
蒸馏模型	0.6794	0.6511	0.6456	0.9214	0.8324	0.8629
剪枝模型	0.6401	0.6888	0.6446	0.9013	0.8471	0.8620

表 4-6 压缩框架模型性能

从另外一个角度,可以从 **4.2.1** 小节的不同模型输出去看更多压缩程度下的结果渐变。









32 通道



16 通道





4通道





图 4-10 CFD 数据集 No.45 模型输出比较

从以上结果可以大体看出设置不同通道的小模型输出得到的图片渐变。事实 上,模型输出的渐变并不非常明显,在对结果需求不严格的情况下,小模型的结 果已比较良好。因此,在综合考虑准确率和硬件条件下,可在本文提出的框架下 选择其它结构作为压缩小模型。

4.5 本章小结

本章我们首先总体介绍了应用裂缝图像分割算法的模型压缩框架。接着我们 介绍了模型压缩框架的第一阶段方法蒸馏压缩,主要讲解了同拓扑结构模型压缩 和通过蒸馏提高同拓扑结构模型的准确率,其中更重点讲解了 sigmoid 蒸馏方法 提出的思想和原理,给出了蒸馏前后的实验结果对比。最后我们介绍了通道剪枝 算法,介绍了该算法的基本原理,给出了将其应用于裂缝图像分割的实验结果。 从实验结果来看,本论文提出的裂缝图像分割模型压缩框架在准确率损失 3%以 内可以达到参数量减少 2000 倍,计算量减少 200 倍的效果。

第五章 裂缝图像分割在手机端上的实现

本论文以上内容是关于裂缝图像分割的模型建立以及模型压缩和加速的算法框架,为了验证此框架能够实际应用,通过 TensorFlow 实现了算法流程,并通过 TensorFlow Lite 转换为安卓手机可以识别的格式,在安卓手机上开发了相应的APP,实际验证了算法效果。本章节主要介绍算法的代码实现过程。

5.1 模型建立与模型压缩的程序实现

模型的建立包含四个步骤:模型的定义、数据的处理、模型的训练以及模型 的测试。

模型的定义包括实现模型的各模块以及把模块组合成一个统一的结构网络。 模型的定义需要涉及 TensorFlow 的一个重要概念:计算图或简称为图(Graph)。 在 TensorFlow 中,所有的操作都必须在图中定义,每一个操作对应于一个节点, 数据以四维 Tensor 的形式在节点间流动。具体地对于 Unet 的网络结构,我们定 义了卷积、池化、反卷积和连接四个操作。

```
def conv2d(x, scope, inshape, outshape)
```

def pool(x)

def deconv2d(x, scope, inshape, outshape)

def copy_concat(x1, x2)

其中,x代表输入的图片或者特征图,inshape 和 outshape 是当前操作前和操作后的特征图维度大小,scope 则定义了此操作所在的作用域,使得构建网络时多次调用函数而不混淆。在每个操作函数定义里,具体地我们借助 TensorFlow的 API 根据操作的定义完成。以卷积为例,其定义过程为:用 tf.variable_scope 定义作用域;用 tf.get_variable 定义和初始化权值 W 和偏置 b;用 tf.nn 模块的卷积、加偏置和 ReLU 操作定义相关过程。实现了各模块的独立功能以后,需要定义网络的前向运算。借助 Tensorflow 方便的 API,只要对应好输入和输出,然后把各模块堆叠起来即可。

数据的处理方式如 3.3.1 小节所述,在本实现中数据的预处理比较简单,只 需要用 PIL 读取原图,在 Python 层面上归一化,然后在 numpy 层面上按 batch size 堆叠成四个维度的矩阵,即可在后续转换成 Tensor 输入网络进行进一步运 算。

模型的训练需要涉及 TensorFlow 的另一个重要概念:会话(Session)。当图 定义完成后,所有操作都是静态的,而会话的作用则是运行图中的操作,通过加 载一个定义好的图,往图中的输入节点喂数据,从输出节点读出运算结果。在训 练前,我们先需要完善整个计算图,即定义输入和标签的数据占位符 tf.placeholder,用于在训练中接收图像和标签数据;定义损失函数和优化方式。 然后,为了有利于训练,我们随机打乱样本的排列方式。在定义的会话中,加载 定义好模型结构的图,并初始化全局的可训练变量。在每一次迭代训练过程中, 读取并按数据预处理方式处理图片数据,计算网络的前向运算结果和反向传播更 新权值。反向传播在 TensorFlow 中是自动实现的,因而只需要安照 TensorFlow 给 定的方式给出要运算的节点即可。最后我们通过 Matplotlib 画出训练过程的 loss 图像,便于调节超参数。

通过以上建模的步骤,可以得到 Unet 的大模型。此大模型不直接用于后面的程序,而是用于在蒸馏中计算所有训练图片经过网络后得到的像素级别的logits,并存储。此 logits 即蒸馏方法中的教师模型的输出,用于指导学生模型训练,由于其是静态的,故一次计算后存储,减少程序运行的内存消耗和反复运算的时间。

在蒸馏阶段,搭建一个同样拓扑结构,但规模小很多的 Unet 模型,建立过程与大模型的建立过程完全一致,因而蒸馏程序的实验脚本代码与 Unet 建立的代码也基本一致,在实际中可考虑统一到一起,分开调用。不同的是蒸馏模型在计算 loss 时增加了一个项,从大模型的 logits 输出计算得来,与真实的标签一起作为复合的损失来训练小模型。训练完成后得到模型的 ckpt 文件,保存了模型的参数,作为下一步剪枝用到的文件。

在剪枝阶段,由于 TensorFlow 不支持对权重的维度直接修改,但支持对权重的数值进行赋值。因此这里需要根据剪枝结果提前建立一个新的网络,把模型的

权值复制到新的网络。例如,假设原模型权重的维度大小为 32*32*3*3,剪枝后 权重的维度大小 16*32*3*3,在 TensorFlow 中不支持直接修改,因而通过建立一 个新的网络,通过剪枝算法选出原网络权重中需要对应维度的 16 个,再把其对 应数值复制到新的新网络,新网络的权值大小得到有效缩小。

建立新的网络过程同样与建立 Unet 模型的过程类似,这里不再赘述,难点 在于通道剪枝过程的实现。这里的实现借鉴了腾讯的开源压缩框架 PocketFlow 的 代码实现,主要是 channel-pruning 模块的核心代码。核心部分指的是通道剪枝 中的通道选择,也即特征选择。实现过程是先通过数据转换把特征图从四维的形 式转成二维的矩阵,然后可通过 sklearn 的 Lasso 回归拟合出从输入到输出的系 数,当系数为0时即可去除当前输入。这里的输入是由转成矩阵后的输入特征图 和权重相乘得到。另外在用 Lasso 回归的时候,需要确定合适的系数,以确保选 择出来的特征图通道数满足剪枝比例,此处用的是二分法,初始化的边界值中二 分搜索合适的 L1 正则化系数。

剪枝除了实现算法核心部分外,还必须通过 TensorFlow 提供的 API 实现其它 过程,这是另一个实现难点,因为大多数深度学习算法只需要关心模型前向运算 的最后输出,既不需要关心中间过程特征图的输出,也不需要准确读取中间卷积 核的值。本论文实现的过程如下:1)获取计算图中卷积操作列表;2)获取卷积 的权值和偏置的变量字典;3)根据卷积操作列表获取每个操作的输入和输出节 点;4)通过运行计算图的会话获得训练数据采样下输入输出节点的取值,并保 存;5)对于每个卷积操作,取其权值、偏置和输入输出节点的采样值,进行数 据维度转换和应用通道剪枝算法。

通过以上过程,最终可以得到模型大小压缩并且精度损失较小的网络结构。 保存其 ckpt 文件,后续可以进一步移植到手机上。

5.2 手机端上的实现

神经网络在手机端上的实现需要借助平台的支持。与前面的模型建立和模型 压缩保持一致,这里选择的是 TensorFlow 平台,具体的是 TensorFlow Lite 部件,

可以支持算法在安卓手机上的运行。

从前面过程中的 ckpt 文件可以通过 TensorFlow 和 TensorFlow Lite 提供的转换工具,先转换成冻结模型 pb 文件,再进一步地转换成 tflite 文件。转换成 pb 文件是为了把训练好的神经网络结构和参数冻结,并简化神经网络,只保留输入到输出相关的节点。此处得到的 pb 文件多应用于电脑端的神经网络推测。而转换得到的 tflite 文件则是 TensorFlow Lite 可以识别的格式,进一步根据手机的硬件架构做了二值化的运算优化,支持神经网络在安卓系统上的运行。

安卓端上,分为界面编写和算法调用两部分。界面编写主要实现了 app 功能 以及交互部分,包括主界面和结果显示界面,其截图如图所示。主界面包括相机 拍照分割和相册选图分割两部分功能。相机拍照分割支持从实际场景中拍照取图 并完成分割,其实现是由调用摄像头、抓取摄像头的一帧图片、调用算法三部分 组成:相册选图则支持从手机本地相册中选取图片进行分割,实现包括读取本地 图片路径、调用算法两部分。两部分功能得到的结果均可在结果显示界面中显示, 包括显示原图和分割图以及算法运行的时间。另外还有保存和返回两个按键,保 存按键可以保存当前的原图和分割图到系统默认的相册路径,返回按键可以返回 主界面继续执行功能。







图 5-1 手机端 APP 界面截图

算法调用涉及数据预处理和调用 tflite 模型两个阶段。数据预处理阶段保证 图片的大小和数据范围与 tflite 模型训练阶段一致。从算法角度看原则上图像分 割可以接受任意大小的图片作为输入,但是由于 tflite 的支持限制,需要预先定 义输入图片的大小,因为我们的数据训练集图片均是 480*320 大小的,我们同样 选取此大小作为输入图片的大小,对选取图片作相应的预处理。具体操作为先从 相机拍摄或相册选取一张图片,执行以下步骤:

- 1) 如果图片的宽<高,将图片顺时针旋转90度;
- 2)如果图片的宽:高>1.5,保持高不变,在宽度上截取高的 1.5 倍大小;如果图片的宽:高<1.5,保持宽不变,在高度上截取宽/1.5 的大小;</p>
- 3) 把图片整体 resize 成宽:高 = 480*320 的大小;
- 4) 像素值从整型的[0,255]归一化到浮点数的[-1,1],归一化形式与模型在电脑端测试时归一化形式对齐。

调用 tflite 阶段先以字节缓存的形式将安卓端的图片表示形式 bitmap 转成二进制字节流 Bytes,存储于一个变量 imgData 中。然后通过 TensorFlow Lite 封装

好的调用形式调用 tflite 文件,可得到模型输出结果,以二进制形式存储于另一 个变量 outData 中。通过数据预处理阶段的逆过程,即从二进制形式转成 bitmap, 并从[-1,1]还原成[0,255],即得到最终图像分割的结果。记录调用 tflite 模型运算 所用的时间,大多数情况下可以在 1.5s 完成分割,在部分手机上能实现 1s 内完 成。

5.3 本章小结

在本章节我们主要介绍了裂缝图像分割在手机端上实现的代码实现过程。首 先我们介绍了在电脑端的神经网络模型建立以及模型压缩的实现过程。模型建立 主要是调用了 TensorFlow 的相关 API 进行网络结构的定义,模型压缩则主要涉及 核心的特征图选择算法实现,另一个难点是利用 TensorFlow 的 API 对网络的一些 中间层结果以及权值参数的读取。其次我们介绍了在手机端上实现裂缝图像分割 的过程。界面开发定义了 app 可进行拍照或者相册取图,而算法调用层面则包括 图片的预处理和利用 TensorFlow Lite 实现对模型的调用。

第六章 总结与展望

6.1 本文总结

本文工作主要是针对于裂缝图像分割在手机端上的实现而展开。裂缝是评估 结构损伤程度的重要参考,随着技术的不断发展,裂缝检测的自动化是目前的发 展趋势。裂缝图像分割的目标是把裂缝从拍摄的结构照片中分割出来,从而可以 通过标定等方式获取裂缝的长度、宽度等参数,进一步研究裂缝。本文的工作涵 盖了从裂缝图像分割模型建立到手机端实现的整个流程框架,具体如下:

(1)对裂缝图像分割问题进行了数学分析,与当前流行的深度学习算法联系起来,探究了应用于图像分割的主流深度学习方法,选择了 Unet 进行复现和修改,将其应用于裂缝图像分割问题的模型构建,并且实验得出不错的结果。

(2) 对裂缝图像分割的卷积神经网络提出了模型压缩的框架。对训练好的 模型先进行蒸馏压缩,再进行通道剪枝压缩,得到参数量和运算量都大大下降, 而准确率下降不大的小模型。在蒸馏压缩的步骤中,先对 Unet 进行层数和通道 数的直接减少,得到同拓扑结构的小模型,再通过蒸馏提高其准确率。在通道剪 枝中,通过最小化下一层特征图的重构误差来选择需要裁剪的通道,从而实现裁 剪。

(3)将压缩得到的小模型移植到手机上。开发了裂缝图像分割相关功能的 app,界面部分具有拍照和相册取图的功能。在算法调用部分,通过 TensorFlow Lite 调用模型压缩及转换后的 tflite 模型。App 整体功能可以对裂缝图片分割出其 中的裂缝,近乎实时。

6.2 未来工作展望

在本研究工作中,由于研究的侧重点,研究资源和时间限制等原因,当前的 研究工作存在一定的局限性,未来仍有以下改进空间:

1. Unet 模型的准确率可进一步提高。由于本论文的工作重点在于模型压缩

框架和算法在手机端上的实现,对于 Unet 模型建立过程的探索相对较少,通过 针对准确率的研究,可以进一步提高 Unet 模型的准确率,并且可以与前沿算法 作比较。

算法的稳定性可进一步提高。在本文的实验过程,整体存在着部分结果指标不稳定的问题,另外在剪枝阶段,偶尔会出现剪枝失败的例子,这些属于算法的稳定性问题,可以进一步解决。

3. 可采用更丰富多样的训练数据拓宽模型的应用场景。本文只是用了公开的 CFD 数据集训练验证框架的有效性,实际上此方法框架可应用于包括道路裂缝和 墙壁裂缝等不同应用场景。此外因为 Unet 强烈依赖于数据增广,而本文未采用 太多的数据增广方式,可以预见数据的增加也将进一步提高模型的准确率。

4. 可对更多网络结构验证框架效果。当前的框架仅对 Unet 的结构进行了效 果验证,而理论上,如果图像网络的在同拓扑结构下能缩小,同时存在卷积操作, 即可应用框架的部分或全部,这可以通过实验进一步验证。

参考文献

- [1] Zou Q, Cao Y, Li Q, et al. CrackTree: Automatic crack detection from pavement images[J]. Pattern Recognition Letters, 2012, 33(3): 227-238.
- [2] Chambon S, Moliard J M. Automatic road pavement assessment with image processing: review and comparison[J]. International Journal of Geophysics, 2011, 2011.
- [3] Shi Y, Cui L, Qi Z, et al. Automatic road crack detection using random structured forests[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(12): 3434-3445.
- [4] Zakeri H, Nejad F M, Fahimifar A. Image based techniques for crack detection, classification and quantification in asphalt pavement: a review[J]. Archives of Computational Methods in Engineering, 2017, 24(4): 935-977.
- [5] Wang K C P. Elements of automated survey of pavements and a 3D methodology[J]. Journal of Modern Transportation, 2011, 19(1): 51-57.
- [6] Guan H, Li J, Yu Y, et al. Automated road information extraction from mobile laser scanning data[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(1): 194-205.
- [7] Sun X, Huang J, Liu W, et al. Pavement crack characteristic detection based on sparse representation[J]. EURASIP Journal on Advances in Signal Processing, 2012, 2012(1): 191.
- [8] Tsai Y J, Jiang C, Wang Z. Pavement crack detection using highresolution 3D line laser imaging technology[C]//7th RILEM International Conference on Cracking in Pavements. Springer, Dordrecht, 2012: 169-178.
- [9] Lim R S, La H M, Sheng W. A robotic crack inspection and mapping system for bridge deck maintenance[J]. IEEE Transactions on Automation Science and Engineering, 2014, 11(2): 367-378.
- [10] La H M, Lim R S, Basily B, et al. Autonomous robotic system for high-efficiency non-destructive bridge deck inspection and evaluation[C]//2013 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, 2013: 1053-1058.
- [11] Adhikari R S, Moselhi O, Bagchi A. Image-based retrieval of concrete crack properties for bridge inspection[J]. Automation in construction, 2014, 39: 180-194.
- [12] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

- [13] Zhang D, Li Q, Chen Y, et al. An efficient and reliable coarseto-fine approach for asphalt pavement crack detection[J]. Image and Vision Computing, 2017, 57: 130-146.
- [14] Fujita Y, Mitani Y, Hamamoto Y. A method for crack detection on a concrete structure[C]//18th International Conference on Pattern Recognition (ICPR'06). IEEE, 2006, 3: 901-904.
- [15] Abdel-Qader I, Abudayyeh O, Kelly M E. Analysis of edge-detection techniques for crack identification in bridges[J]. Journal of Computing in Civil Engineering, 2003, 17(4): 255-263.
- [16] Tanaka N, Uematsu K. A Crack Detection Method in Road Surface Images Using Morphology[J]. MVA, 1998, 98: 17-19.
- [17] Yamaguchi T, Hashimoto S. Fast crack detection method for largesize concrete surface images using percolation-based image processing[J]. Machine Vision and Applications, 2010, 21(5): 797-809.
- [18] Oliveira H, Correia P L. Automatic road crack detection and characterization[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(1): 155-168.
- [19] Oliveira H, Correia P L. CrackIT—An image processing toolbox for crack detection and characterization[C]//2014 IEEE international conference on image processing (ICIP). IEEE, 2014: 798-802.
- [20] Amhaz R, Chambon S, Idier J, et al. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(10): 2718-2729.
- [21] Cha Y J, Choi W, Büyüköztürk O. Deep learning based crack damage detection using convolutional neural networks[J]. Computer - Aided Civil and Infrastructure Engineering, 2017, 32(5): 361-378.
- [22] Zhang A, Wang K C P, Li B, et al. Automated pixel level pavement crack detection on 3D asphalt surfaces using a deep - learning network[J]. Computer - Aided Civil and Infrastructure Engineering, 2017, 32(10): 805-819.
- [23] Zhang L, Yang F, Zhang Y D, et al. Road crack detection using deep convolutional neural network[C]//2016 IEEE international conference on image processing (ICIP). IEEE, 2016: 3708-3712.
- [24] Cheng J, Wang P, Li G, et al. Recent advances in efficient computation of deep convolutional neural networks[J]. Frontiers of Information Technology & Electronic Engineering, 2018, 19(1): 64-77.
- [25] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint arXiv:1510.00149, 2015.

- [26] He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 1389-1397.
- [27] Luo J H, Wu J, Lin W. Thinet: A filter level pruning method for deep neural network compression[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5058-5066.
- [28] Courbariaux M, Hubara I, Soudry D, et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1[J]. arXiv preprint arXiv:1602.02830, 2016.
- [29] Zhu C, Han S, Mao H, et al. Trained ternary quantization[J]. arXiv preprint arXiv:1612.01064, 2016.
- [30] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2704-2713.
- [31] Zhou S, Wu Y, Ni Z, et al. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients[J]. arXiv preprint arXiv:1606.06160, 2016.
- [32] Astrid M, Lee S I. Cp-decomposition with tensor power method for convolutional neural networks compression[C]//2017 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2017: 115-118.
- [33] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.
- [34] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.
- [35] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.
- [36] 马卫飞. 基于深度学习的桥梁裂缝检测算法研究. 硕士论文. 陕西师范大学. 77. 2018.
- [37] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Cognitive modeling, 1988, 5(3): 1.
- [38] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
- [39] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.
- [40] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

- [41] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.
- [42] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [43] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.
- [44] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNetlevel accuracy with 50x fewer parameters and< 0.5 MB model size[J]. arXiv preprint arXiv:1602.07360, 2016.
- [45] Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6848-6856.
- [46] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.
- [47] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.
- [48] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.
- [49] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.
- [50] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
- [51] Taigman Y, Yang M, Ranzato M A, et al. Deepface: Closing the gap to human-level performance in face verification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 1701-1708.
- [52] Cao Z, Hidalgo G, Simon T, et al. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields[J]. arXiv preprint arXiv:1812.08008, 2018.
- [53] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

- [54] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.
- [55] Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 40(4): 834-848.
- [56] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [57] Maeda H, Sekimoto Y, Seto T, et al. Road damage detection and classification using deep neural networks with smartphone images[J]. Computer - Aided Civil and Infrastructure Engineering, 2018, 33(12): 1127-1141.

致 谢

时光飞逝,一眨眼间三年就过去了,硕士学习生活即将结束,在此,我想真 诚地感谢那些曾经帮助我的人。

难忘老师的栽培之恩,特别感谢我的导师范衠教授,他学识渊博、待人和蔼, 治学严谨科研风格和求真务实的工作作风,给我留下了深刻的印象。是他给我们 提供一个好的科研平台;是他给予了我们无私的循循诱导提高我们的科研能力; 是他在学习上和生活上给予我们支持和信任,经过老师的栽培,我在思辨和表达 的能力上具有了质的提高。

感谢实验室小组成员,他们对科研充满热情,在每周的小组讨论活动中学到 很多知识,拓展了对其他领域的了解在研究上给了我很大的帮助,在此由衷的感 谢实验室的小伙伴们。特别感谢同实验室的小伙伴卢杰威同学,这三年来对我在 知识面,系统的科研能力培养,以及其它生活方面均给予了很大的帮助。

最后感谢我的父母和家人,感谢你们在些年来一直在我身边不离不弃,支持 着我的学业,让我在学习上没有顾虑,专心学习,即使遇到困难,你们一直在默 默的鼓励和支持,给予我安慰和指导,你们是我学习的最大动力。

我怀揣着深深的感激之情,感谢所有关心与帮助过我的人,谢谢你们!

伍宇明 2019年4月

攻读硕士学位期间主要的工作成果

[1] Zhun Fan, **Yuming Wu**, Jiewei Lu, Wenji Li. Automatic pavement crack detection based on structured prediction with the convolutional neural network[J]. arXiv preprint arXiv:1802.02208, 2018.

[2] 黄龙涛;**伍宇明**;段一鸣;朱贵杰。基于无人机的高层建筑墙体裂缝检测。 "兆易创新杯"第十三届中国研究生电子设计竞赛,获华南赛区三等奖。

申请专利

[1] 范衠; **伍宇明**; 朱贵杰; 卢杰威。一种基于深度卷积神经网络的裂缝图像分 割方法。[P]. 中国专利,申请号: 201810676243.5

[2] 范衠; **伍宇明**; 卢杰威; 朱贵杰; 莫嘉杰; 黄龙涛。一种基于深度学习的蝴 蝶自动分类方法。[P]. 中国专利, 申请号: 201811070920.5

[3] 范衠; **伍宇明**; 李冲; 史新宝; 朱贵杰。一种基于智能终端的裂缝检测方法。[P]. 中国专利,申请号: 201811653343.2