

基于图像处理的远距离裂缝检测智能平台开发

余欣, 许益奖, 李薇, 石威, 刘晓宇

(四川大学 机械工程学院, 成都 610000)

摘要:为解决传统裂缝检测工作效率低、危险系数高、耗时较长及检测精度低等问题,提出一种大型建筑物远距离裂缝检测智能平台方案,该方案基于深度学习和图像处理建立裂缝检测算法模型,采集图像后上传至云端,以用户终端设备为载体,通过微信小程序和后端服务器的交互获取裂缝相关信息,服务器实现裂缝检测模型的实时检测,裂缝数据存储在云端;经过对算法部分的图像分类模型与图像分割模型进行测试,分类模型精确率、召回率、F值均超过92%,分割模型平均交并比MIoU超过89%,满足检测需要。经验证该方案切实可行。

关键词:裂缝检测;图像处理;微信小程序;云服务器;深度学习

Development of an Intelligent Platform for Remote Crack Detection Based on Imageprocessing

YU Xin, XU Yijiang, LI Wei, SHI Wei, LIU Xiaoyu

(School of Mechanical Engineering, Sichuan University, Chengdu 610000, China)

Abstract: In order to solve the traditional crack detecting low working efficiency, high risk, takes longer and low accuracy of detection problem, put forward a kind of large buildings long crack detection intelligence platform solutions, the scheme is based on deep study establishes a model of crack detection algorithm and image processing, after acquisition images uploaded to the cloud, as user terminal equipment as the carrier, The fracture related information is obtained through the interaction between the Wechat applet and the back-end server. The server realizes real-time detection of the fracture detection model, and the fracture data is stored in the cloud. After testing the image classification model and image segmentation model in the algorithm part, the accuracy rate, recall rate and F value of the classification model are more than 92%, and the average intersection union ratio of the segmentation model is more than 89%, which meets the needs of detection. It is proved that the scheme is feasible.

Keywords: crack detection; image processing; WeChat applet; cloud server; deep learning

0 引言

随着国家社会经济的高速发展,众多关系到国民生产的各行业及人们的日常生活大型建筑物,如隧道、桥梁、水坝、大楼等的数量快速增长,而由于老化以及各种各样的原因,建筑物很容易产生裂缝。裂缝不仅影响建筑外表的美观,更影响建筑的安全使用寿命,存在较大的安全隐患,甚至可能会给社会造成重大的经济损失和人员伤亡。为了确保这些建筑物的使用安全,必须定期对其进行裂缝识别及检测,准确评估建筑物的裂缝发展情况,从而制定科学合理的建筑修护方案。

传统的裂缝检测方式主要依靠人工近距离作业的方式,多利用传感器(如超声波传感器)进行裂缝检测,超声波检测是利用超声波的特性,通过检测出脉冲波在被测物中的传播时间以及接收波的振幅和频率,从而确定建筑物表面裂缝的有无、大小和位置等信息^[1]。这种方式由于硬件

发展水平的限制,检测精度难以进一步提升,而且对于隧道和横跨山河的桥梁等大型建筑,近距离作业使得检测难度大、危险系数高,故常常需要使用远距离检测的方式。

目前,用传统的人工作业的方式进行裂缝检测已经越来越无法满足工程的需求,同时随着硬件设备性能的提升以及大规模数据集的出现使得深度学习技术在目标检测,实例分割等视觉领域已取得显著成果^[2-4],基于图像处理的智能识别检测的新方法应运而生。这种智能化裂缝检测方法实现了数字化、自动化,极大地提高了检测精度与效率^[5]。

此外,随着深度学习技术在计算机视觉领域的蓬勃发展和在交通及工业生产等场景的广泛适用,基于深度学习技术的裂缝智能检测平台为大势所趋,选择合适的深度学习平台部署深度学习网络模型进行大型建筑物表面的裂缝图像识别检测至关重要,目前国内外研究人员主要基于嵌入式^[6-9]和微型PC终端^[10-11]研发远距离裂缝检测智能平台,然而由于大多数深度学习算法占用内存大且运算复杂度较

收稿日期:2022-07-28; 修回日期:2022-08-22。

基金项目:2022年四川省“大学生创新创业训练计划项目”(S202210610375)。

作者简介:余欣(2001-),女,贵州毕节人,大学本科生,主要从事测控技术与仪器方向的研究。

通讯作者:刘晓宇(1987-),女,内蒙古包头人,博士,副教授,主要从事精密测量技术与仪器、机器视觉和机器学习方向的研究。

引用格式:余欣,许益奖,李薇.基于图像处理的远距离裂缝检测智能平台开发[J].计算机测量与控制,2022,30(12):70-77.

高, 而嵌入式、PC 终端等边缘设备资源空间、功耗和计算能力等受限, 这为深度学习的部署开发带来了一定的难度和局限性。因此, 性能更加强大的裂缝智能检测平台的研发有待完善和发展。

近年来互联网的云计算、云存储等云端技术日益成熟, 采用云服务器实现云数据的聚合和分析, 可以最大程度地整合与裂缝相关的信息和数据, 实现资源的灵活扩展^[12]。

因此本项目以裂缝检测领域迫切需求的对大型建筑物表面裂缝进行远距离检测为导向, 基于图像处理和深度学习搭建依托于微信小程序和云服务器的远距离裂缝智能检测平台, 可实现更方便、更快捷、更智能化的裂缝检测。

1 系统结构与原理

1.1 基于云端的远距离裂缝检测方案

首先利用远摄变焦镜头采集图像并通过微信小程序上传到云端服务器, 其中微信小程序与云端服务器间的通信基于 TCP/IP 协议, 以 https 通信方式进行数据交互; 然后进行裂缝检测; 最后将分析结果及裂缝特征参数信息等存储到云端, 同时微信小程序将该分析结果进行解析渲染展现给用户。

方案原理如图 1 所示。

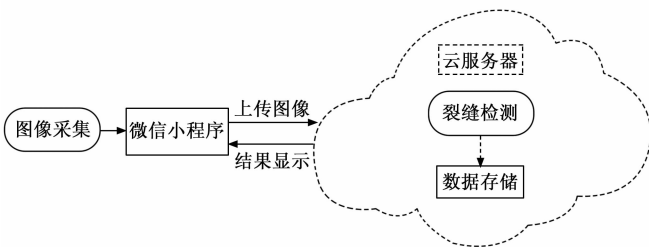


图 1 基于云端的远距离裂缝检测方案

1.2 智能平台架构

1.2.1 展现层和通讯层

展现层实现用户交互, 负责图像的输入、渲染与展示。通讯层为系统前端(展现层)与后端(功能层、数据库)的数据传输, 基于 HTTP/HTTPS 协议和 POST/GET 请求。

1.2.2 业务层

通过调用业务数据接口, 实现相应的功能, 包括裂缝检测、数据存储等。

业务层由服务类组成, 并使用数据层提供的服务实现展现层所需业务数据的处理, 为实现安全服务、接口服务、数据分析、数据处理、日志消息、状态信息等提供支持^[13]。

1.2.3 数据层

提供底层数据, 实现原始数据存储、数据缓存、数据分析读写以及文件存储等功能。

2 系统设计与实现

2.1 微信小程序前端

如今, 智能手机全面普及, 微信小程序是一种全新的

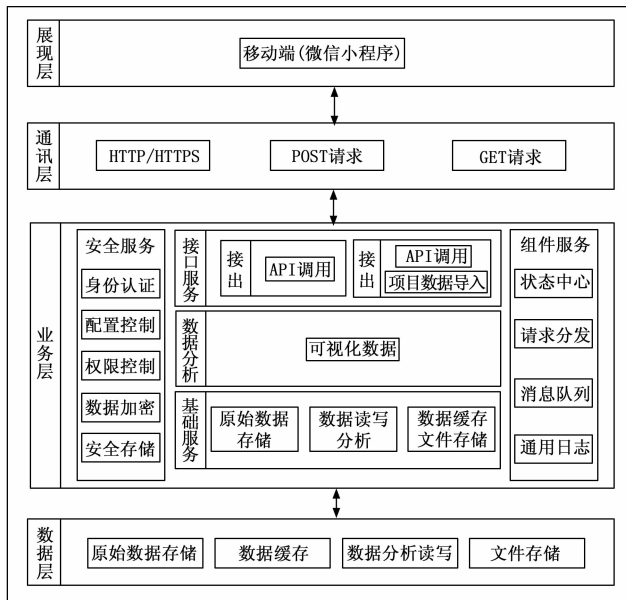


图 2 远距离云端裂缝检测智能平台架构

连接用户与服务的方式^[14], 其背靠微信用户量大的优势, 可共享微信生态, 同时具有优秀的用户体验; 微信云开发提供了原生 Serverless 云服务, 无需搭建服务器, 降低了运维成本^[15-17]; 依托于微信小程序的开放接口, 可以构建移动端的裂缝检测应用平台, 同样可实现与后台算法的交互; 用户使用时无需下载, 使用便捷, 并且通过微信小程序自带的云数据库实现对用户数据的存储。

微信小程序采用 wxml、wxss、JavaScript 进行开发, 本质是一个单页应用, 所有的页面渲染和事件处理, 都在一个页面内进行, 但又可以通过微信客户端调用原生的各种接口。

1) 程序结构。微信小程序由 js、json、wxss 和 wxml 四个文件组成。js 是文件应用程序是逻辑文件, json 文件应用程序是配置文件, wxss 应用程序是定义公共样式, wxml 是用于定义 wxss 公共样式的外观, 比如大小、形状和颜色等。通过配置这四个文件可以实现预期程序的产出。

2) 小程序与云服务器交互。小程序端调用 wx.uploadFile 实现小程序向后端服务器发送一个 POST 请求, 将用户上传至云服务器; 云服务器收到数据后调用算法进行检测, 之后返回相应的值。由于返回的值微信小程序不能直接展示, 需要通过 JSON.parse 将返回值转换为可识别的值, 之后微信小程序便可将转换后的数据展示在界面上, 实现前后端数据的交互。

3) 程序界面设计。根据需求, 本项目设计了登录界面、初始界面、检测界面、建议反馈界面以及作者介绍界面。在 js、json 页面下定义相应函数实现所需要的功能, 再在 wxss 和 wxml 页面设计各界面的样式, 以达到美观效果。

4) 程序主要功能。程序登陆界面用于获取用户的数

据，以便于实验统计；授权后会进入到初始界面，初始界面包含了三个功能按钮，分别为“开始检测”按钮、“建议反馈”按钮以及“关于我们”按钮。点击“开始检测”按钮便可进入检测界面，之后即可上传裂缝图片进行检测，小程序会跳转到检测结果界面；点击“建议反馈”按钮进入到建议反馈界面，用户可向开发者提供建议，以便开发者对程序的进一步优化；点击“关于我们”按钮进入到团队介绍界面，可便捷地了解开发者团队。如图 3 所示为微信小程序总体流程图。

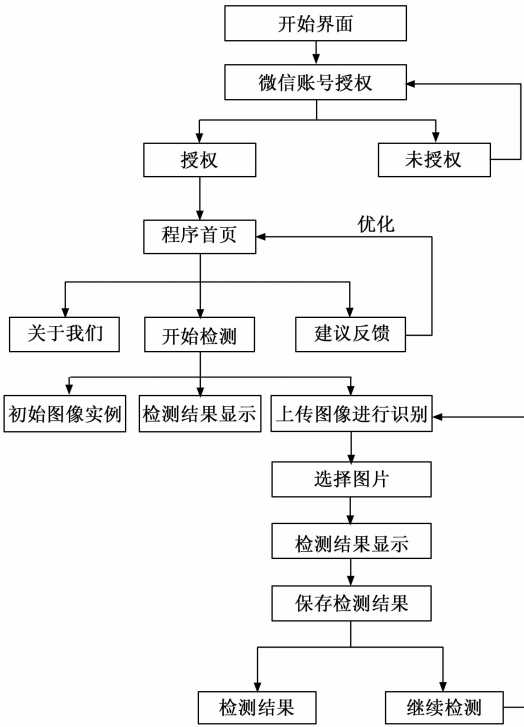


图 3 前端流程设计

2.2 云服务器端

云服务器 (ECS, elastic compute service) 是云计算服务的重要组成部分，是面向各类互联网用户提供综合业务能力的服务平台，平台整合了传统意义上的互联网应用三大核心要素：计算、存储、网络。由于云服务器属于互联网基础设施的范畴，用户可根据个人或群体的需求随时、定量地使用其计算资源，做到高效、便捷地使用云服务器。与传统服务器相比，用户使用云服务器不仅无需购买硬件实体，降低了运营成本，而且可在短时间内依据需求创建或释放多台云服务器，实现性能扩充，简单高效^[18]。

本项目的后端使用 Python 语言编程，采取 Nginx + uWSGI + Flask 应用的方式部署，算法模型部署在 Flask 上。微信小程序向服务器发起 https 请求，Nginx 首先接收到请求并转发给 uWSGI，server (uWSGI 服务器) 根据自身的 uwsgi 和 WSGI 协议，找到对应的 Flask 框架，Flask 框架下的应用进行逻辑处理后，将返回值发送到 uWSGI 服务器，然后 uWSGI 服务器再返回给 Nginx，最后 Nginx 将

返回值返回给微信小程序进行渲染显示给用户。云服务器部署流程如图 4 所示。

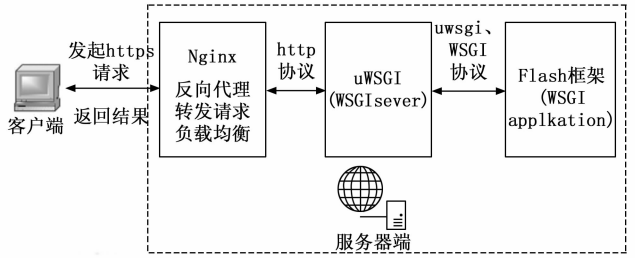


图 4 云服务器部署流程

2.2.1 Web 服务器配置

Nginx 代理服务器处理静态文件和索引文件效果非常高；其设计在保证较高效率的同时所占内存也较小；此外，Nginx 不仅稳定性高，而且配置简洁。而 uWSGI 做为一款优秀的 Python 应用服务器，更擅长处理动态请求。故从安全性和可扩展性方面来讲，使用 Nginx + uWSGI 是最佳方式。

其中 Nginx 服务器通过修改 Nginx 配置文件实现 https 访问和反向代理，它可根据服务器的负载情况将请求分配给不同的服务器进行处理，且在响应处理过程中，客户端无法得知真正服务器的 IP 地址，保证了服务器性能和安全。

uWSGI 服务器实现 uwsgi 协议和 WSGI 协议，分别定义了传输信息的类型和保证了不同 Web 服务器可以和不同的 Python 程序之间相互通信。在 WSGI 中，Web 服务器端被称为 server，会先收到用户请求，依据一定的规范要求调用应用程序端，即 Web 框架（如 Flask），执行结果会被封装并返回到客户端。

2.2.2 Flask 框架

Flask 是一个轻量级 Python Web 框架，因其简洁、轻巧、扩展性强等优势大受欢迎^[19-20]，由两个依赖（Werkzeug 和 Jinja2）和第三方库组成。Flask 性能上基本满足一般 Web 开发的需求，并且灵活性以及可扩展性上要优于其他 Web 框架，对各种数据库的契合度都非常高。

主要功能：接受并处理小程序传来的数据；部署裂缝检测算法模型；编写 POST 类型请求的 API；存储裂缝检测前后图像及裂缝数据。

表 1 展示了 Flask 程序文件的目录结构。

表 1 Flask 程序文件目录结构

文件名称	说明
main.py	API 接口文件(route: /image), 调用 Jian.py
Jian.py	裂缝检测算法
static	包含 before 和 after 两个文件夹, 分别存储前后裂缝图像检测结果
templates	存放模板文件(.html)
logs	存放日志文件
requirements.txt	依赖包环境
uwsgi.ini	项目启动文件, 指定 application 地址

2.3 裂缝检测模型

输入图像以后, 首先使用卷积神经网络对图像进行分类, 判断是否存在裂缝目标。对存在裂缝的图像进行预处理, 再使用语义分割网络进行图像分割, 提取裂缝区域。最后使用传统的图像处理算法提取裂缝的像素数据。根据镜头数据与测量距离计算像素解析度, 将像素数据与像素解析度相乘, 便可以得到实际的裂缝数据。

裂缝检测流程如图 5 所示。

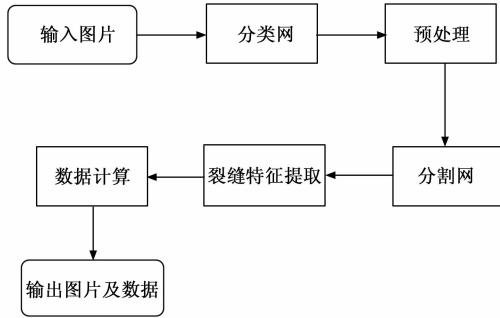


图 5 裂缝图像处理流程

2.3.1 图像采集及像素解析度计算

使用长焦镜头采集图像, 使用激光测距仪测出相机与裂缝区域的距离。

根据透镜成像原理, 有:

$$\frac{1}{L'} + \frac{1}{L} = \frac{1}{f} \quad (1)$$

$$L' = \frac{Lf}{L-f} \quad (2)$$

式中, L 为物距; f 为镜头焦距。设 A 为目标实际尺寸, 即裂缝实际物理尺寸, A' 为成像尺寸, 则有:

$$\frac{A}{A'} = \frac{L}{L'} \quad (3)$$

$$A = \frac{L}{L'} A' \quad (4)$$

代入得:

$$A = \frac{L-f}{f} A' \quad (5)$$

成像尺寸 A' 为:

$$A' = \frac{d}{D} A'' \quad (6)$$

得到像素解析度为:

$$J = \frac{(L-f)d}{fD} = \frac{A}{A''} \quad (7)$$

式中, A'' 为成像的像素数; d 为图像传感器长边物理尺寸; D 为图片传感器长边像素数。

相机成像表面与裂缝表面若存在夹角 θ , 需要测出这个夹角, 从而对像素解析度进行修正。李清鑫等提出了一种测 θ 角的方法^[21], 使用三点激光测距仪, 利用三束激光之间的几何关系计算得到 θ 角。

修正后的像素解度:

$$J = \frac{(L-f)d}{fD} \cos\theta = \frac{A}{A''} \cos\theta \quad (8)$$

θ 为成像平面与裂缝表面的夹角。

2.3.2 图像分类

2019 年谷歌研究出了一种卷积神经网络模型—EfficientNet^[22]。这个网络模型创新地使用一个简单而高效的复合系数从网络深度、网络宽度和输入图像分辨率三个维度来放大网络, 网络精度更高, 参数量更少。

本文使用 EfficientNet-B0 作为图像分类网络。EfficientNet-B0 包含 16 个移动翻转瓶颈卷积模块 (MBConv)、2 个卷积层、1 个全局平均池化层以及 1 个分类层^[23]。与 ResNet-50 相比, 该网络速度提升了将近十倍, 网络参数仅有其大约五分之一。通过该图像分类网络, 可以对输入的图片进行有效分类, 判断是否存在裂缝。

2.3.3 图像预处理

对输入的裂缝图像进行预处理, 步骤主要有三步: 图像灰度化、图像对比度增强、图像滤波去噪^[24]。

镜头采集得到的图片是三通道的 RGB 图像, 需要通过灰度化将三通道变为单通道。本文使用加权均值平均算法进行灰度化, 公式如下所示:

$$f(x, y) = W_R R(x, y) + W_G G(x, y) + W_B B(x, y) \quad (9)$$

式中, $f(x, y)$ 是灰度图像素值, $R(x, y)$, $G(x, y)$, $B(x, y)$ 分别为彩色图像中的红、绿、蓝三种颜色通道中的像素分量, W_R , W_G , W_B 为 $R(x, y)$, $G(x, y)$, $B(x, y)$ 的权值。权值选择以人眼对色彩的敏感性为根据, 实验表明, 当 $W_R = 0.299$ 、 $W_G = 0.587$ 、 $W_B = 0.114$ 时, 能获得最符合人眼视觉感受的灰度图图像^[25]。

采集到的图像需要提高其对比度, 以便表现图像的细节。本文使用伽马校正的方法增强对比度, 通过对图像像素值的非线性变换, 可以放大局部低光区域, 放大图像细节。其数学表达式如下:

$$L_{OUT} = I^{1/\gamma} \quad (10)$$

L_{OUT} 是输出图像, I 是输入图像, γ 为伽马指数^[26]。

本文通过实验, 发现 γ 指数小于 1 时, 图像对比度增强, γ 指数大于 1 时, 图像对比度减弱。同时, 当 γ 指数较低时, 如 $\gamma = 0.1$, 图像对比度较高, 但同样存在加强了背景噪声的问题。通过对多张图像进行对比度增强, 发现在 γ 从 0.1 逐渐增大至 1.0 的过程中, 对比度增强的力度逐渐减弱, 当 $\gamma = 0.6$ 时, 对比度较高, 而噪声目标未明显增强, 矫正效果最为理想。因此, 本文中伽马指数取 0.6。

对比度增强后的图像有各种噪声包括椒盐噪声和高斯噪声。使用中值滤波去除椒盐噪声, 使用双边滤波去除高斯噪声。中值滤波的公式如下:

$$f(x, y) = \text{Med}\{h(m-p, m-q), (p, q \in M)\} \quad (11)$$

式中, $f(x, y)$ 表示输出像素, 取邻域内像素像素灰度的中值; $h(m, n)$ 表示原始图像中心位置像素点的灰度值;

m 表示邻域窗口的大小; $h(k-p, l-q)$ 就是各个像素点的灰度值。

双边滤波的输出像素值是邻域像素值的加权和, 其加权系数有两部分, 一部分是高斯距离权重, 另一部分是高斯相似度权重, 公式分别如下:

$$c(\xi, x) = e^{-1/2(d(\xi, x)/\sigma_d)^2} \quad (12)$$

式中, $d(\xi, x) = \|\xi - x\|$ 为两像素 ξ 和 x 相间的距离。

$$s(\xi, x) = e^{-1/2(\sigma(f(\xi), f(x))/\sigma_f)^2} \quad (13)$$

式中, $\sigma(f(\xi), f(x)) = \|f(\xi) - f(x)\|$ 为当前像素与一个邻域像素的标准差。

融合距离和亮度信息去滤波, 得到最后的公式如下所示:

$$h(x) = k^{-1} \sum f(\xi) c(\xi, x) s(\xi, x) \quad (14)$$

其中: $k(x) = \sum c(\xi, x) s(\xi, x)$, \sum 表示用于滤波的邻域矩阵的大小, $f(\xi)$ 表示原始像素点, 图像预处理前后效果对比如图 6 所示。

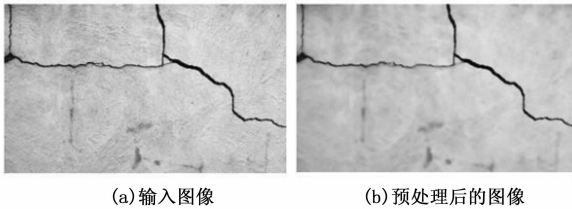


图 6 图像预处理前后效果对比图

使用双边滤波的好处是在去除了噪声的同时保护了边缘, 有利于裂缝区域的分割。

2.3.4 图像分割

本文使用 SegNet^[27] 语义分割网络进行图像分割, 如图 7 所示。

SegNet 包含编码 (encoder) 和解码 (decoder) 两个部分。在编码器中, 包含了 VGG16 中的 13 个卷积层, 同时编码器中存在 5 个最大池化层, 其下采样操作可以缩小图像。同时存储池化操作中最大值的索引位置。在解码器中进行上采样操作。解码器中也有 13 个卷积层。与编码器不同, 解码器中有 5 个上采样层, 用于将特征图还原至原来的大小。上采样操作采用转置卷积的方式。解码器后是一个 softmax 分类层, 接收最后一个卷积层的输出, 对每个像素独立分类, 输出有 K 个通道, 每个通道代表一个类别。最终实现图像语义分割。



图 7 SegNet 网络分割效果

2.3.5 裂缝特征提取

通过图像分割得到裂缝区域后, 将图像转化为二值图

像, 图像中, 白色区域就是裂缝区域。

图像分割可能会使连续的裂缝边缘发生断裂, 使用形态学运算中的闭运算对这些裂缝边缘进行连接, 并使用开运算去除毛刺。

同时, 二值图像中可能存在一些噪声, 要想提取出完整的裂缝特征还需要对图像进行滤波。利用裂缝的圆度较低, 面积较噪声大的特点对噪声进行了滤除。

由于裂缝区域连通性较好, 连通域面积较大, 而噪声区域多为孤立点, 面积较小, 可以设立一个阈值, 将低于此阈值的连通域视为噪声区域, 从而进行初步去噪。然后利用连通域的圆度大小进行进一步去噪。

圆度用于描述区域接近圆形的程度, 如果区域越接近圆形, 则它的圆度越接近于 1。

本文采用 Halcon 中关于圆度的定义来计算圆度, 计算公式如下:

$$C = \frac{F}{max^2 * \pi} \quad (15)$$

C 为圆度, F 为区域面积, max 为区域中心点到轮廓点的最大值。

噪声点的圆度一般较大, 裂缝区域的圆度一般较小, 通过对多张裂缝图片进行去噪处理发现, 将圆度大于 0.1 的白色连通域视为噪声进行滤除后, 可以比较完整地提取出裂缝区域, 提取效果如图 8 所示。

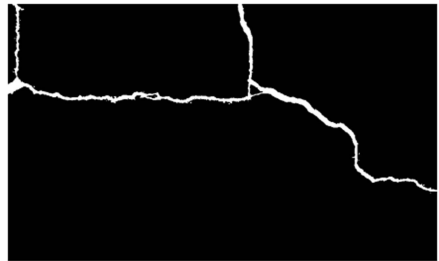


图 8 裂缝特征提取效果

2.3.6 裂缝参数计算

经过特征提取之后, 所得白色连通域便是裂缝区域。统计所有像素值为 255 的像素点的个数, 便可以得到裂缝的像素面积。通过中轴变换算法, 得到裂缝骨架, 即一条只有一个像素点宽的直线, 统计这条直线的像素点的个数便可以得到裂缝的像素长度。同时, 中轴变换过程中还会返回中轴线上所有点与中间点的距离, 可以利用此距离得到裂缝宽度的像素宽度。将以上数据乘以像素解析度便可以得到实际的裂缝数据。

3 实验结果与分析

3.1 图像分类实验

3.1.1 数据预处理

数据集 Concrete Crack Images for Classification 中含有 20 000 张带有裂缝的图像和 20 000 张不带有裂缝的图像, 每张图像的大小为 227×227 , 是裂缝检测相关研究中使用较为广泛的数据集。从此数据集中挑选出 1 000 张带有裂缝的图像

和 1 000 张不带有裂缝。在数据预处理阶段, 将数据集中的图片进行中心裁剪, 裁剪至 224×224 。其中 224×224 是 EfficientNet-B0 的标准输入尺寸^[16], 同时, 由于裂缝一般位于图像中间, 中心裁剪可以减少一部分背景噪声。之后再通过归一化、以 0.5 概率随机翻转等操作以增强数据。

3.1.2 模型训练

网络模型使用 EfficientNet 的 BO 版本, 训练的参数如下表所示, 输入图像尺寸为 224×224 , 每批数据量为 32, 学习率为 0.001, epoch 为 100 次, 动量参数设为 0.9, 权重衰减为 0.000 1。使用 SGD 优化器和交叉熵损失函数。

在模型训练过程中, 损失函数值随训练次数增加而衰减, 当训练至 100 次时, 模型基本收敛, 损失函数值约为 0.016 72。

3.1.3 实验指标

本实验针对图像分类模型进行评估, 衡量分类模型的性能。主要实验指标: 精确率 (precision)、召回率 (recall)、 F 值^[28]和 FPS (frames per second) 指标。计算精确率和召回率时还需使用以下参数: TP (true positives) 表示裂缝样本被正确分类为裂缝的样本数, TN (true negatives) 表示非裂缝样本被正确分类为非裂缝的样本数, FP (false positives) 表示非裂缝样本被错误分类为裂缝的样本数, FN (false negatives) 表示裂缝样本被错误分类为非裂缝的样本数。

精确率表示被正确预测为裂缝的图像占预测为裂缝的图像的比例, 定义如下:

$$P = \frac{TP}{TP + FP} \quad (16)$$

召回率表示被正确预测为裂缝的图像占全部裂缝图像的比例, 定义如下:

$$R = \frac{TP}{TP + FN} \quad (17)$$

F 值用于综合反映整体的指标。定义如下:

$$F = 2 \cdot \frac{P \cdot R}{P + R} \quad (18)$$

FPS 表示每秒处理的图片数, 用于衡量检测算法的检测速度。

3.1.4 测试结果

使用算法模型对 250 张带有裂缝的图像和 250 张不带有裂缝的图像进行测试测试结果如表 2 所示。

表 2 分类网测试结果

Model	EfficientNet
TP(True positives)	242
TN(True negatives)	231
FP(False positives)	19
FN(False negatives)	8
Precision/ %	92.72
Recall/ %	96.80
F/ %	94.72
FPS(Frames Per Second)	4.64

实验结果表明图像分类模型精确率和召回率均比较高, 算法模型精度较高, 且每秒可以处理将近 5 张图片, 可以满足实时分类的需求。

3.2 图像分割实验

3.2.1 数据预处理

同样从数据集 Concrete Crack Images for Classification 数据集中挑选出 1 000 张带有裂缝的图像, 使用 labelme 工具标注出图像中的裂缝区域。

3.2.2 模型训练

训练的参数如表 3 所示, 输入图像尺寸为 256×256 , 每批数据量为 32, 学习率为 0.001, epoch 为 100 次, 权重衰减为 0.000 1。使用 Adam 优化器和二值交叉熵损失函数。

在模型训练过程中, 损失函数值随训练次数增加而衰减, 当训练至 100 次时, 模型基本收敛, 损失函数值约为 0.043 24。

3.2.3 实验指标

平均交并比 $MIoU$ 值是一个衡量图像分割精度的重要指标, 定义是预测值和真实值的交集与并集之比的平均值。 $MIoU$ 越接近于 1, 说明分割模型分割效果越精确。计算公式如下:

$$MIoU = \frac{1}{k} \sum_{i=1}^k \frac{P \cap G}{P \cup G} \quad (19)$$

其中: P 表示预测值, G 表示真实值, K 表示总类别个数。

3.2.4 测试结果

对测试集的 500 张裂缝图像使用 labelme 标注出裂缝区域, 并将它们分为 5 组, 分别使用使用分割网模型对另外进行分割, 计算平均交并比 $MIoU$ 。

表 3 分割网 $MIoU$ 值

组别	$MIoU$ / %
1	89.27
2	89.15
3	89.21
4	89.13
5	89.23
Mean	89.20

如表 3 所示, 测试集的 $MIoU$ 稳定在 89.20% 左右, 裂缝分割效果较好。

3.3 裂缝处理实验

3.3.1 长度计算实验

本文使用 NikonD7500 相机拍摄裂缝图像, 其 CMOS 传感器的尺寸为 $23.5 \text{ mm} \times 15.7 \text{ mm}$, 镜头焦距范围为 $18 \sim 140 \text{ mm}$, 拍摄的照片尺寸均为 $5\,568 \times 3\,712$ 。拍摄时, 保持成像平面与裂缝表面平行, 测量拍摄时的物距并记录焦距。

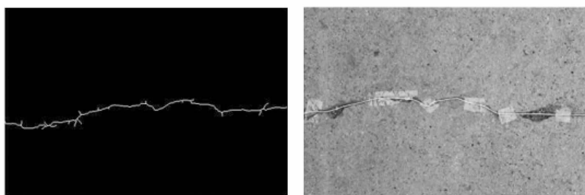
同时, 对于这种尺寸较大的图像, 本文先将图像缩小至原来的十分之一, 图像通过算法模型后, 再将数据包括

数值与图像放大至 10 倍, 从而加快图像处理过程, 并得到与原图直接处理相一致的数据。其中, 缩小和放大图像时, 插值方法选择 OpenCV 中的基于像素区域关系的插值算法 (INTER_AREA), 这种方法可以避免产生波纹现象, 插值效果较好。实验数据如表 4 所示。

表 4 裂缝长度计算 mm

图片编号	计算长度	实际长度	相对误差/%
1	384.2	353.4	8.7
2	111.2	103.8	7.1
3	480.3	446.1	7.7

计算的长度与实际长度的相对误差平均值为 7.83%, 分析误差产生的原因, 一个主要原因是在提取裂缝骨架时会出现许多分叉, 而在实际测量时, 测量的是主骨架的长度, 不包括分叉, 如图 9 所示, 因而产生了误差。



(a) 提取裂缝骨架

(b) 测量实际的裂缝长度

图 9 长度计算实验

3.3.2 远距离裂缝区域提取

使用同样的相机对远距离的建筑物表面裂缝进行处理, 提取裂缝区域。处理结果如图 10 所示。

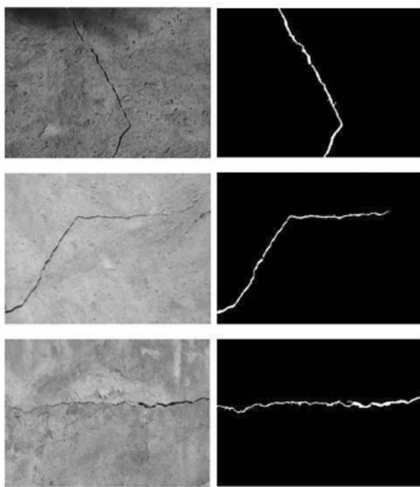


图 10 提取裂缝区域

拍摄时, 物距保持在 10~20 m。经过处理后, 可以较为完整地从中提取出裂缝区域。远距离裂缝图像处理的精度受图像采集设备的影响更大, 使用焦距更高、性能更优的采集设备可以采集到成像质量更高的图像, 有利于提取裂缝区域与计算数据。

3.3.3 智能平台效果展示

在“程序首页”界面能够直观地了解本小程序的所能实现的功能, 在“建议反馈”界面用户可以上传自己的建议到微信小程序自带的云数据库, 在“开始检测”界面用户能够上传图片进行裂缝检测, 最后“检测结果”界面展示所检测到裂缝的形状以及相关参数。如图 11 所示。



图 11 智能平台效果展示

4 结束语

本文针对目前市场紧缺大型建筑物远距离裂缝检测智能仪器的现状, 开发了基于微信小程序和 Python Flask 的智能检测平台, 通过对裂缝特征的提取, 得到裂缝图像及其长度、面积等相关信息, 之后依据相关标准对其进行综合评价, 从而实现对大型建筑物健康状况的评估, 在一定程度上规避了较大安全隐患。同时, 以微信小程序为依托进行人机交互, 在保证高检测精度、高工作效率的同时提高了该检测平台的智能化和市场竞争能力。

通过对本项目程序进行的实验, 可以看出由移动端与部署了裂缝检测算法的云服务器搭建的裂缝检测智能平台能够以较高的精度实现对大型建筑物裂缝的检测, 并达到了以下效果。

1) 简便快速: 基于微信小程序搭建的智能平台无需下载, 使用移动设备即可登录微信访问, 用户上传图片即可进行检测。

2) 效率高: 微信小程序可以直接访问用户相册, 并且由于算法质量高使得检测时间短, 用户检测一张裂缝图后可以很快速地检测下一张。

3) 高精度: 云服务器可实时检测, 并同步进行图像处理算法模型的在线训练, 不断提高检测的精度, 以适应更加复杂的应用场景。

4) 云数据库存储: 调用微信小程序自带的云数据库可以实现对用户检测图片的储存以及收集用户反馈信息。

5) 实现长期裂缝数据跟踪存储: 云服务器存储空间可无限扩容, 因此通过云服务器可存储不同时期裂缝检测的数据, 形成连贯裂缝跟踪, 便于观察裂缝是否继续扩大、扩大的速度是怎样的, 从而更好预测什么时候该对裂缝进行处理。

同时, 未来还将进一步完善算法, 并不局限于微信小

程序, 扩展平台类型如 APP、网站等。当前, 工程方面对远距离的, 能够实时检测的、高精度的检测设备的需求较大, 这样的设备能够满足对社会上建筑测量智能化的需求, 本研究对于建筑物裂缝智能化检测、鉴定、评估工作具有重大意义。

参考文献:

- [1] 李 杰. 超声波在桥梁裂缝检测中的应用 [J]. 交通世界, 2019 (17): 107-108.
- [2] HINTON G E. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313 (5786): 504-507.
- [3] WAN Y, HAN Y, LU H Q. The methods for moving object detection [J]. Computer Simulation, 2006, 23 (10): 221-226.
- [4] CHEN L C, GEORGE P, IASONAS K, et al. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs [J]. TPAMI, 2018, 40 (4): 834-848.
- [5] 张海明. 土木工程结构中远距离非接触式表面缺陷及裂缝检测系统研究 [D]. 北京: 中冶集团建筑研究总院, 2013.
- [6] 吴雨多, 黄泽君, 何子冬. 基于 ARM 和 FPGA 的嵌入式桥梁检测 [J]. 信息技术, 2013, (7): 140-141.
- [7] 钱 程. 基于机器视觉的桥梁自动监测系统研究 [D]. 苏州: 苏州科技大学, 2016.
- [8] 侯 欢. 基于红外图像处理的混凝土桥梁裂缝检测方法研究 [D]. 北京: 北京化工大学, 2020.
- [9] 周敬富. 基于层次聚类的混凝土裂缝检测方法及应用 [D]. 北京: 北京化工大学, 2021.
- [10] 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.
- [11] 张 旭, 陈绪君, 刘瑞康, 等. 基于 OpenVINO 模型优化的智能道路裂缝检测系统 [J]. 信息技术, 2020, 44 (7): 62-68.
- [12] 於晓兰. 基于云服务器的物联网智能家居通信控制系统研究 [J]. 江苏通信, 2021, 37 (4): 46-48.
- [13] 杜云龙, 陈 含, 刘 利, 等. 怀柔水库渗压监测自动化升级改造研究 [J]. 水利水电快报, 2021, 42 (7): 54-58.
- [14] 陈小龙. 微信公众平台开发实战与应用案例 [M]. 北京: 清华大学出版社, 2015.
- [15] 杜春涛. 微信小程序开发案例教程 [M]. 北京: 中国铁道出版社有限公司, 2019.
- [16] 王玉洋. 基于微信小程序的移动学习平台环境构建与系统设计开发 [D]. 南京: 南京大学, 2018.
- [17] 曹经纬. 基于 Knative 的 Serverless 平台函数管理模块的设计和实现 [D]. 南京: 南京大学, 2020.
- [18] 陈海汝, 何 青, 潘轩平, 等. 基于云服务器的 SSM 框架后台搭建与实现 [J]. 信息系统工程, 2019 (11): 114-115, 117.
- [19] IDRIS N, FOOZY C, SHAMALA P. A generic review of Web technology: Django and Flask [J]. International Journal of Advanced Science Computing and Engineering, 2020, 2 (1): 34-40.
- [20] 吴 狄. 基于 Python 的分布式信息采集发布系统设计实现 [D]. 洛阳: 河南科技大学, 2019.
- [21] 李清鑫, 刘小燕, 张 华, 等. 裂缝宽度的提取及在桥梁承载力评估中的应用研究 [J]. 长沙理工大学学报 (自然科学版), 2020, 17 (1): 60-64.
- [22] TAN M, LE Q V. EfficientNet: rethinking model scaling for convolutional neural networks [C] // Proceedings of International Conference on Machine Learning, 2019: 6105-6114.
- [23] 艾霖斌, 徐权峰, 杜利婷, 等. 基于 EfficientNet 的星系形态分类研究 [J]. 天文学报, 2022, 63 (4): 44-51.
- [24] 陈益佳. 基于 OpenCV 的路面裂缝检测算法研究 [D]. 哈尔滨: 哈尔滨工程大学, 2017.
- [25] 李贞培, 李 平, 郭新宇, 等. 三种基于 GDI+ 的图像灰度化实现方法 [J]. 计算机技术与发展, 2009, 19 (7): 73-75, 79.
- [26] 刘昱垚. 基于 Gamma 校正的图像对比度增强方法研究 [D]. 呼和浩特: 内蒙古工业大学, 2021. DOI:10.27225/d.cnki.gnmggu.2021.00205.
- [27] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: a deep convolutional encoder-decoder architecture for image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39 (12): 2481-2495.
- [28] 王墨川, 王 波, 王熊珏, 等. 基于高语义特征与注意力机制的桥梁裂缝检测 [J]. 湖北工业大学学报, 2022, 37 (2): 55-59.
- [19] 赵晨旭. 测试性增长试验理论与方法研究 [D]. 长沙: 国防科学技术大学, 2016.
- [20] 陈 然, 连光耀, 孙江生, 等. 加速退化试验改进的故障模式影响及危害性分析 [J]. 西安电子科技大学学报, 2017, 44 (3): 164-169.
- [21] 邱 静, 刘冠军, 李天梅. 基于虚拟样机的可测性虚拟试验验证技术研究及发展 [J]. 计算机测量与控制, 2008, 16 (10): 81-87.

(上接第 69 页)

- [16] 段尧清, 周 密, 等. 政府开放数据公众初始接受行为影响因素结构关系研究 [J]. 图书情报工作, 2020, 64 (2): 4-12.
- [17] 吴健平. 基于 AHP-SWOT 法的我国 BIM 应用现状与发展对策研究 [D]. 福州: 福建工程学院, 2018.
- [18] 石 全, 王立欣, 史宪铭, 等. 系统决策与建模 [M]. 北京: 国防工业出版社, 2016.