

# 基于改进多分支特征共享结构网络的裂缝检测算法

李刚<sup>1</sup>, 陈永强<sup>1\*</sup>, 何廷全<sup>2</sup>, 代玉<sup>1</sup>, 兰栋超<sup>1</sup>

<sup>1</sup>长安大学电子与控制工程学院, 陕西 西安 710064;

<sup>2</sup>广西新发展交通集团有限公司信息部, 广西 南宁 530029

**摘要** 针对路面裂缝检测时裂缝的位置、形态的不确定性及裂缝特征与路面背景纹理的相似性等问题, 提出了一种改进的多分支特征共享结构网络的裂缝图像分割算法。为了在减少计算参数冗余的同时提高检测精度, 使用轻量化特征提取网络获取高层特征, 采用多分支跳跃连接的方法提高通道间的信息利用。各分支融合全局卷积网络 (GCN) 模块和边界细化 (BR) 模块, 提高了对裂缝边缘的分割性能和对裂缝区域内部分类的鲁棒性, 利用循环残差卷积 (RRC) 模块, 推动了对裂缝特征的累积。最后采用中轴法提取裂缝骨架, 计算裂缝的形态参数, 得到裂缝长度和宽度的相对误差分别为 4.73% 和 5.21%。设计的多组对比实验结果表明, 所提改进算法能够有效地提高对路面裂缝检测的精度和效率。

**关键词** 图像处理; 语义分割; 裂缝检测; 裂缝参数计算; 边界细化; 多分支特征共享

中图分类号 TP391.9

文献标志码 A

DOI: 10.3788/LOP202259.1215005

## Crack Detection Algorithm Based on Improved Multibranch Feature Shared Structure Network

Li Gang<sup>1</sup>, Chen Yongqiang<sup>1\*</sup>, He Tingquan<sup>2</sup>, Dai Yu<sup>1</sup>, Lan Dongchao<sup>1</sup>

<sup>1</sup>School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, Shaanxi, China;

<sup>2</sup>Information Department, Guangxi New Development Transportation Group Co., Ltd.,  
Nanning 530029, Guangxi, China

**Abstract** To address issues such as position and shape uncertainty in pavement crack detection, as well as similarity between crack features and pavement background texture, an improved crack image segmentation algorithm based on multibranch feature shared structure network is proposed. To improve the detection accuracy while reducing the redundancy of computational parameters, a lightweight feature extraction network is used to acquire high level features, and the multibranch hopping connection method is employed to improve the information utilization between channels. Each branch combines the global convolution network (GCN) module and the boundary refinement (BR) module to improve crack edge segmentation and classification robustness within the crack region, and it employs the recurrent residual convolution (RRC) module to drive crack feature accumulation. Furthermore, the crack morphological parameters are calculated using the median-axis method to extract the crack skeleton, and relative errors of the crack length and width are 4.73% and 5.21%, respectively. The results of multiple datasets of designed comparison experiments show that the proposed improved algorithm can significantly improve the accuracy and efficiency of pavement crack detection.

**Key words** image processing; semantic segmentation; crack detection; crack parameter calculation; boundary refinement; multi-branch feature sharing

收稿日期: 2021-05-17; 修回日期: 2021-06-04; 录用日期: 2021-06-11

基金项目: 长安大学研究生科研创新实践项目 (300103714037)、陕西省重点研发计划 (2020ZDLGY09-03)

通信作者: \*2019132048@chd.edu.cn

## 1 引言

路面裂缝作为道路极易出现的一种常见病害形式,在各种外在因素的影响下会导致路面结构承载力下降,危害交通安全,因此及时准确地获取裂缝检测信息至关重要<sup>[1]</sup>。然而传统的基于图像处理技术的方法依赖于主观人为设定的图像特征,靶向性很强而难以扩展到其他复杂场景当中。徐志刚等<sup>[2]</sup>采用这种方法分析裂缝的形态学特征,进而确定裂缝位置,通过切分子图像分别对图像的不同区域进行直方图估计来进行区域分割,效果优于阈值分割方法,但未测试噪声图像。Hu等<sup>[3]</sup>则通过设计一种新型灰度和旋转不变算子对图像中的粗糙区域进行分割,该算子对不规则纹理类型的噪声具有较强的鲁棒性。Salman等<sup>[4]</sup>使用Gabor函数进一步检测背景纹理复杂的裂缝图像,分析裂缝连通域并设定合适的滤波器参数作后处理,提高抗噪声干扰能力,但得到的裂缝边界不连续而难以进行裂缝形态参数的计算。王德方等<sup>[5]</sup>则在传统的图像处理方法中加入了形态学处理方法,并融合聚类算法和区域生长法进行裂缝提取,提高了图像的抗光照能力但检测效率偏低。卢小平等<sup>[6]</sup>采用多次迭代的方法获取裂缝图像的快速检测模型,该模型在迭代一定次数后可达到较高的准确率,但其依赖于迭代前的图像预处理操作,对应用场景的可扩展性不足。由此可见,传统的图像处理方法难以处理复杂形式的裂缝图像,往往只能针对裂缝检测的单一问题进行优化,难以平衡抗干扰能力、准确性和检测效率等多方面的问题。

近年来,计算机视觉领域的蓬勃发展使得路面裂缝检测的自动化水平不断提高,大量研究者基于深度学习方法开展关于路面裂缝检测方面的研究工作。Zhang等<sup>[7]</sup>率先使用深度卷积神经网络搭建分类模型,该模型能够根据提取到的裂缝特征分离裂缝图像和非裂缝图像,但无法实现裂缝区域的定位且误检率较高。Chen等<sup>[8]</sup>则利用卷积神经网络与朴素贝叶斯融合的方法剔除检测过程中的假阳性错误,以锚点为中心生成检测框来实现裂缝区域定位,但该方法并未分割裂缝区域。Zou等<sup>[9]</sup>进一步提出了一种具有编解码结构的裂缝检测方法,该方法通过学习裂缝的高级特征来实现多尺度深度卷积特征融合,能够分割裂缝区域但得到的评估指标仍不够高。朱苏雅等<sup>[10]</sup>使用多层卷积和阈值法

提取裂缝特征,融合改进的连接算法进行裂缝的精细提取,取得了较好的检测效果,但模型参数量庞大,导致训练比较困难。与此同时,Ghosh等<sup>[11]</sup>使用ResNet的残差块来改进U-Net的卷积层结构,在多个公开数据集上获得了优良的性能,但该模型并未测试其他裂缝图像,鲁棒性不足。

计算机硬件设备的提升加快了对视觉图像的分析处理速度,进而出现了大量采用图像分割方式的路面裂缝检测模型。该模型在整体结构层面上以卷积和反卷积操作获取裂缝的多级特征,在特征层面上则对各级裂缝特征图进行像素点级别的相加或进行通道级别的拼接实现多尺度特征融合,最后进行逐点像素判断类别获得裂缝与背景区域的二分类分割图像<sup>[12]</sup>。这种方式需要模型有更强的语义信息表达能力,能够保留足够的整体细节用于分类判断。鉴于上述研究,本文根据裂缝区域检测的特点提出一种改进的多分支特征共享结构网络的裂缝检测算法。所提算法充分利用原始Fast-SCNN<sup>[13]</sup>轻量化、高效的特点,引入多分支跳跃连接以加强通道间信息的利用,同时使用全局卷积网络(GCN)模块和边界细化(BR)模块<sup>[14]</sup>增强对裂缝区域的特征表达能力,消除卷积核的冗余计算,减少结构增加带来的计算负荷;并在网络结构中融合循环残差卷积(RRC)模块进行特征累积,保证后续特征图的表达能力,从而提高路面裂缝检测精度;最后采用中轴法处理预测图像,通过骨架提取分析形态信息并计算误差。结果证明,改进后的算法能够有效地对较小的边缘不清晰的裂缝进行分割,有较强的抗噪声干扰能力和较高的实时性。

## 2 改进的多分支特征共享结构网络

所提改进的多分支特征共享结构网络由4个部分组成。首先是用于裂缝特征初步提取的下采样阶段,紧接着是捕获多层次上下文特征信息的全局特征提取阶段,随后为整合各分支不同层次特征的特征融合模块,最后是用于裂缝和非裂缝区域类别判定的分类阶段。

### 2.1 原始的快速分割算法

原始的Fast-SCNN算法基于二分支结构和编解码网络的组合,结构可以分为4个部分。学习下采样模块包含1个标准卷积层和2个深度可分离卷积层,类似于Mobilenet<sup>[15]</sup>,用于初步的特征提取。深度可分离卷积为深度卷积和 $1 \times 1$ 点卷积的组

合,首先使用深度卷积融合输入特征图的空间信息,随后使用点卷积对特征图的通道信息进行整

合,利用通道交互降低特征图的通道数,减少模型计算量。图 1 为深度可分离卷积的示意图。

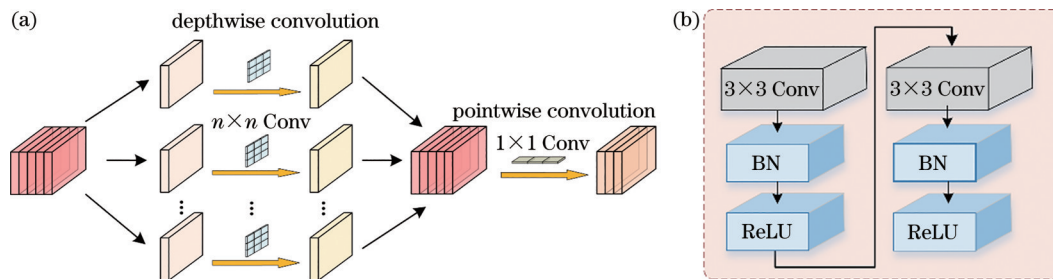


图 1 深度可分离卷积。(a)卷积流程;(b)卷积结构

Fig. 1 Depth separable convolution. (a) Convolution process; (b) convolution structure

在获取初步的特征信息后,使用由反向残差瓶颈块和金字塔池化组合的全局特征提取模块进一步提取特征语义信息,其核心为带有残差的瓶颈深度可分离卷积,通过一系列卷积提取图像的高层特征。随后采用金字塔池化模块在各特征层应用不同大小的池化核学习不同层次的深度特征,再使用上采样操作放大特征图至同一大小后与输入特征图进行拼接,融合不同层次的特征信息。图 2 为金

字塔池化模块的结构图。

特征融合模块类似于残差神经网络的跳跃连接,为了确保效率,将网络的主干分支与另一个浅层特征分支相加得到最后的特征图,使得编码后的特征层融合高层和浅层特征信息。最后将特征图通过一系列卷积后上采样到原图大小,对图像的每个像素点做一次分类,得到最终的分割图像。

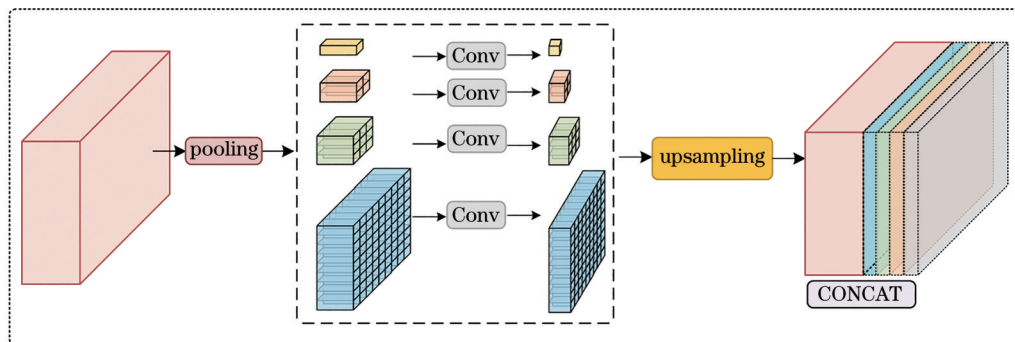


图 2 金字塔池化模块

Fig. 2 Pyramid pooling module

## 2.2 网络层改进原理

针对原始算法的不足,提出如图 3 所示的改进多分支特征共享结构网络,其改进由 4 个方面组成。

首先,增加浅层特征的融合分支结构,促进模型学习更多浅层高分辨率图像的语义特征,提高对裂缝区域的识别能力。模型使用两个深度可分离卷积构成的轻量化初级特征提取网络,采用深度可分离卷积可以减少网络中的浮点运算数和卷积参数,从而可以降低模型的计算开销和存储需求。原算法仅与第二分支的 1/8 特征图融合,丢失了大量的浅层特征,而改进算法通过融合 1/4 和 1/8 特征图两个分支结构,充分利用了更多的高分辨率浅层特征信息。

其次,在进行金字塔池化操作后,分阶段逐次对特征图进行上采样,中间加入 GCN 模块和 BR 模块组成的边缘优化结构,加强对图像中裂缝边界信息的处理,减小直接进行四倍上采样带来的特征学习的计算误差。模型的全局特征提取模块的主干分支通过 3 组 Bottleneck 结构逐步提取裂缝图像的高层特征,随后通过金字塔模块融合特征图不同区域的上下文信息。由于全局特征提取模块的主要输入为进行了高层特征提取的裂缝图像,在经过一系列卷积下采样和金字塔池化后丢失了大量裂缝像素的位置信息,致使裂缝边缘产生毛刺、断裂等定位错误,从而影响到对裂缝边缘像素标签的预测,难以检测到小于一定尺寸的路面裂缝。为此采

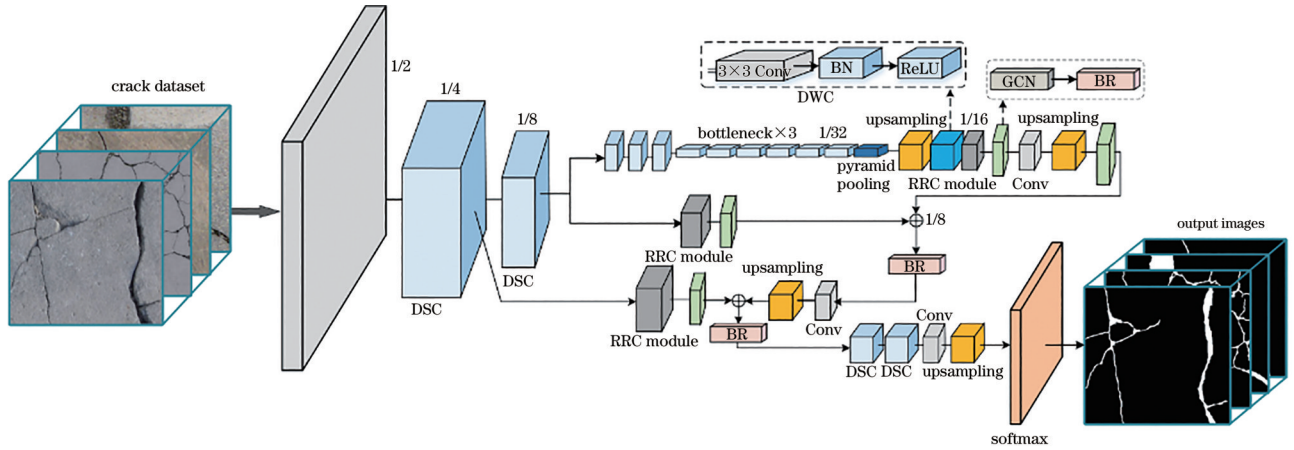


图 3 改进的多分支特征共享结构网络

Fig. 3 Improved multi-branch feature shared structure network

用 GCN 模块和 BR 模块的组合用于优化裂缝预测信息。图 4 为 GCN 模块和 BR 模块的示意图。

在将 1/4、1/8、1/16 等不同分支下采样得到的特征图融合前后,反复应用 GCN 模块和 BR 模块的组合,在提高特征图分辨率的同时达到细化边界进行预测的目的,一定程度上提高了对裂缝边缘分割的精度。在全局特征提取模块和特征融合模块中使用 GCN 模块和 BR 模块,为每类图像生成多尺度语义

评分图,低分辨率特征图在经过上采样后和高分辨率特征图融合以保证多分支特征的共享。不同于条件随机场(CRF)等概率图模型只能作为语义分割模型进行二次推理来提高分割精度,BR 模块可以作为裂缝边缘定位的补充,可嵌入到模型内部,与 GCN 模块相组合加强对裂缝区域内部和边界的空间信息表达。GCN 模块使用两组并行的卷积组合来代替大卷积核,从而降低卷积过程的参数量。

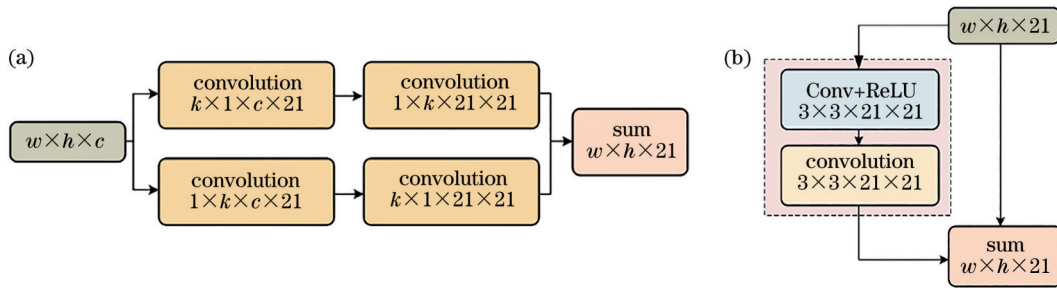


图 4 模块结构图。(a) GCN 模块;(b) BR 模块

Fig. 4 Structure diagram of modules. (a) GCN module; (b) BR module

同时,所提改进算法利用 RRC 模块替代特征融合模块中的普通卷积结构,有助于特征累积来提高特征表达能力,在保证参数量不变的情况下提高模型性能。一般循环卷积层(RCL)的操作是根据离散时间步长来确定的。假设  $\mathbf{x}_n$  为 RCL 的第  $n$  层的输入,  $P_{(i,j)}$  位于 RRC 模块的第  $m$  层特征图的输入样本中,那么网络在时间步长为  $t$  时的输出可以表示为

$$\mathbf{Y}_{(i,j)m}^n(t) = (\mathbf{w}_m^f)^T * \mathbf{x}_n^{f(i,j)}(t) + (\mathbf{w}_m^g)^T * \mathbf{x}_n^{g(i,j)}(t-1) + \mathbf{b}_m, \quad (1)$$

式中:  $\mathbf{w}_m^f$  为标准卷积层的权重;  $\mathbf{w}_m^g$  为 RCL 的第  $m$  层特征图的权重;  $\mathbf{b}_m$  表示偏差;  $\mathbf{x}_n^{f(i,j)}(t)$  和  $\mathbf{x}_n^{g(i,j)}(t-1)$

分别为标准卷积层的输入和 RCL 的第  $n$  层的输入。

RRC 模块由 RCL 融合残差单元组成,其输出  $\mathbf{x}_{n+1}$  可以表示为

$$\mathbf{x}_{n+1} = \mathbf{x}_n + f[\mathbf{Y}_{(i,j)m}^n(t)] = \mathbf{x}_n + \max[0, \mathbf{Y}_{(i,j)m}^n(t)], \quad (2)$$

式中:  $f[\mathbf{Y}_{(i,j)m}^n(t)]$  表示使用标准的激活函数 ReLU 来激活 RCL 的输出;  $\mathbf{x}_{n+1}$  则作为后续网络层的输入。图 5 为 RRC 模块的结构图。

所提算法基于以上对网络结构的改进,增强了模型提取裂缝图像多级特征和融合边界信息的能力,从而显著提高了路面裂缝检测的性能。

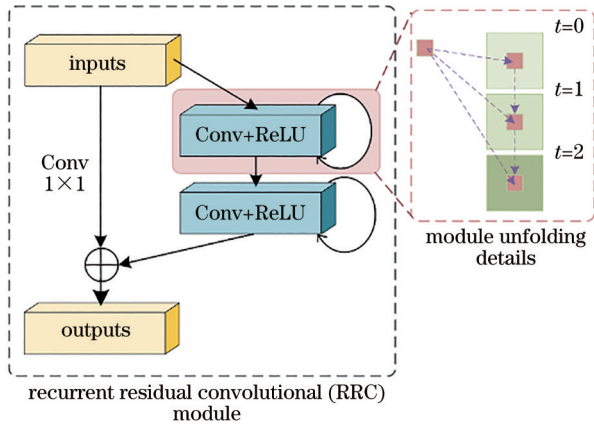


图 5 RRC 模块结构图

Fig. 5 Structure diagram of RRC module

### 3 实验

#### 3.1 建立裂缝图像数据集

路面存在多种形式的裂缝,为了提高模型的鲁棒性,使用 SONY FDR-AX60 相机拍摄了包含阴影、杂物、车道线等噪声干扰且形态多样的路面裂缝图像,随后通过图像标注工具 LabelMe 对采集获得的 800 张分辨率为  $3840 \times 2160$  的原始图像进行标注。使用滑动窗口法将原图裁剪成分率为  $512 \times 512$  的子图像,具体流程如图 6 所示。同时,为了提高模型的泛化能力,通过旋转  $90^\circ$  和  $180^\circ$ 、镜像等数据增广手段进一步扩充数据集,并使用公开裂纹数据集 Cracktree200<sup>[16]</sup>、GAPs384<sup>[17]</sup>、Crack500<sup>[18]</sup> 混合的数据集进行训练和测试。Cracktree200 包含 206 张分辨率为  $800 \times 600$  的带有阴影、车道线等噪声类型的裂缝图像,GAPs384 数据集包含 1969 张分辨率为  $1920 \times 1080$  的灰度图像,Crack500 包含 500 张分辨率为  $2000 \times 1500$  的裂缝图像。为了便于模型训练,同样采用滑动窗口法将图像裁剪为  $512 \times 512$  的尺寸大小。根据裂缝所占图片区域的大小,最终筛选得到总数为 20000 张的裂缝数据集,并以

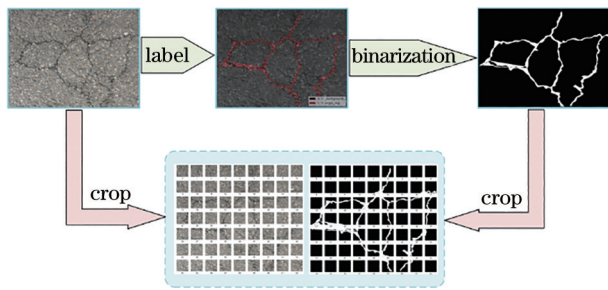


图 6 数据集标注裁剪的流程

Fig. 6 Process of dataset annotation and crop

6:2:2 随机划分训练集、验证集和测试集,具体如表 1 所示。

表 1 裂缝数据集的划分  
Table 1 Partition of crack dataset

Parameter	Training set	Validation set	Test set
Image size / (pixel×pixel)	$512 \times 512$	$512 \times 512$	$512 \times 512$
Number of images /frame	12000	4000	4000

#### 3.2 实验环境和评价指标

使用深度学习框架 PyTorch,实验运行于 Ubuntu20.04 平台,硬件环境为 Intel i7-9750 的 CPU 处理器和 24 GB 的 RAM 内存,搭载 12 GB 显存的 NVIDIA GeForce GTX TitanX 显卡。

根据图片像素的预测类别和实际类别计算混淆矩阵,得到裂缝检测的真阳性(TP)、假阳性(FP)、真阴性(TN)、假阴性(FN)4 种不同的识别状态。由这 4 种不同状态分别计算出基于裂缝像素点的准确率(precision)、召回率(recall)和 F1 评分作为基础指标,同时采用平均交并比(MIoU)来评估分类性能,衡量裂缝分割精度。各项评价指标的公式为

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad (3)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad (4)$$

$$F_1 = \frac{2 \times P \times R}{P + R}, \quad (5)$$

$$P_{Miou} = \frac{1}{2} \left( \frac{N_{TP}}{N_{TP} + N_{FN} + N_{FP}} + \frac{N_{TN}}{N_{TN} + N_{FN} + N_{FP}} \right). \quad (6)$$

#### 3.3 模型训练细节

为了防止模型训练时陷入局部最优点,采用随机梯度下降优化算法,动量参数大小设置为 0.9,权重衰减系数为  $1 \times 10^{-4}$ 。网络的训练批次为 200 个 epoch,训练的批处理大小设置为 8 以提高训练效率。为了选择最佳的初始学习率,比对了不同学习率在模型上的训练参数,使用 Poly 学习率衰减方式,其中  $p$  为 0.9。

$$L = i_{nit} \times \left( 1 - \frac{E_{epoch}}{N_{epoch}} \right)^p, \quad (7)$$

式中: $L$  为学习率; $i_{nit}$  为初始学习率; $E_{epoch}$  为最大迭代次数; $N_{epoch}$  为当前迭代次数。图 7 为改变初始学习率得到的训练损失和精确度曲线。由图 7 可以看

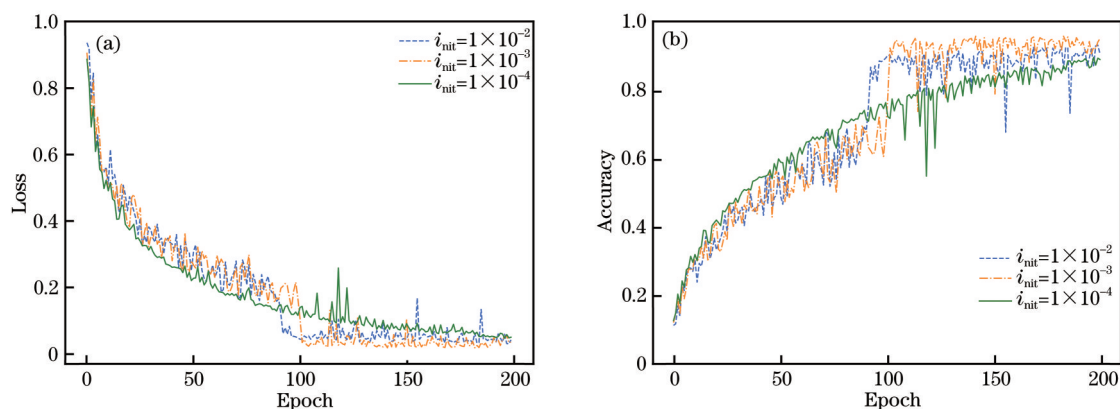


图7 改变初始学习率得到的训练损失和精确度曲线。(a)损失;(b)精确度

Fig. 7 Training loss and accuracy obtained by changing the initial learning rate. (a) Loss; (b) accuracy

出,当初始学习率为 $1 \times 10^{-3}$ 时,模型的训练损失最小,因此选 $1 \times 10^{-3}$ 作为初始学习率。使用交叉熵损失函数并引入正则惩罚项作为模型训练的目标函数,公式为

$$L_{CE} = - \sum_{i=1}^s [y_i \log y_i' + (1 - y_i) \log (1 - y_i')] + \frac{\beta}{2} \sum_j W_j^2, \quad (8)$$

式中: $y_i$ 和 $y_i'$ 分别表示第 $i$ 个输出单元的真实值和预测值; $s^2$ 为标签的数量; $\beta$ 为权重衰减系数; $W_j$ 表示网络的第 $j$ 个权重。

### 3.4 实验结果与对比分析

比较所提改进算法与典型的深度卷积网络模型的性能差异,在相同的训练数据集和训练条件

下,分别对各个网络模型进行训练优化。对所提改进算法与Fast-SCNN、SegNet<sup>[19]</sup>、Pspnet<sup>[20]</sup>、DeepCrack、CrackU-Net<sup>[21]</sup>5种算法进行了比较和分析,并且在含有井盖、落叶、车道线、油渍等复杂干扰噪声的图像数据集上进行了测试,结果如表2所示,表2中以每秒帧数(FPS)来衡量裂缝检测模型的速度,括号中的数值代表检测单张裂缝图像所耗费的时间,单位为ms。实验显示,改进算法的各项分割指标均优于其他检测算法。由于增加的模块中存在较多的残差结构块,且在特征过程中使用多模块组合优化的方式,相比于原算法,改进算法的检测速度有所下降,但仍拥有较好的实时性。未来可以通过模型蒸馏或者模型剪枝方法研究更优良的模块组合方式来提高性能。

表2 不同裂缝检测算法的性能比较

Table 2 Performance comparison of different crack detection algorithms

Algorithm	Precision	Recall	F1	MIoU	FPS (Millisecond/image)
Fast-SCNN	0.7548	0.7962	0.7749	0.7692	67.5(14.81)
SegNet	0.8312	0.7864	0.8082	0.8206	18.7(53.48)
Pspnet	0.9021	0.9115	0.9068	0.8021	29.3(34.13)
DeepCrack	0.8463	0.8159	0.8308	0.8642	25.6(39.06)
CrackU-Net	0.8736	0.7938	0.8318	0.8527	34.8(28.74)
Proposed algorithm	0.9695	0.9283	0.9485	0.8902	59.4(16.84)

图8为不同算法对含有噪声的裂缝图像的可视化对比结果。由图8可以看出:在复杂背景条件下,原始算法对噪声图像的抗干扰性不强,且对裂缝边界信息表达不够准确;增加合理的特征融合分支结构和多模块组合优化能够强化对裂缝特征的表达能力,从而提高检测精度,可以有效地检测出较为微小的裂缝区域,同时对带有复杂背景噪声的裂缝图像有较强的鲁棒性。

图9为各算法指标的变化趋势。如图9(a)所示,随着训练迭代次数的增加,模型学习到了更多的裂缝特征信息并逐渐收敛,当迭代次数达到200时,模型的各项评估指标均达到了最大值。

接受者操作特性曲线(ROC)是一种对灵敏度进行描述的功能图像。由于ROC曲线在评估类别不均衡的模型性能方面表现不佳,而PR曲线对裂缝像素分类的准确性更加敏感,因此采用PR曲线

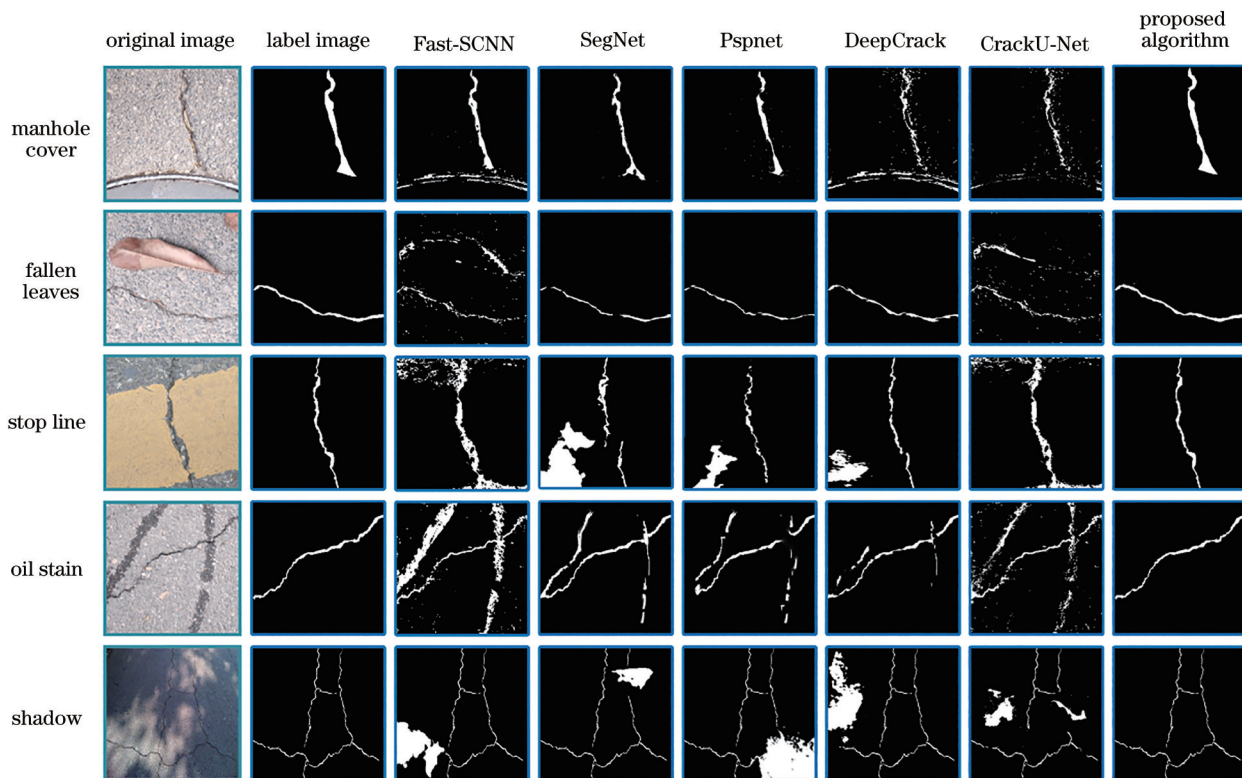


图 8 不同算法在含有噪声的裂缝图像中的可视化对比结果

Fig. 8 Visual comparison results of different algorithms in crack images with noise

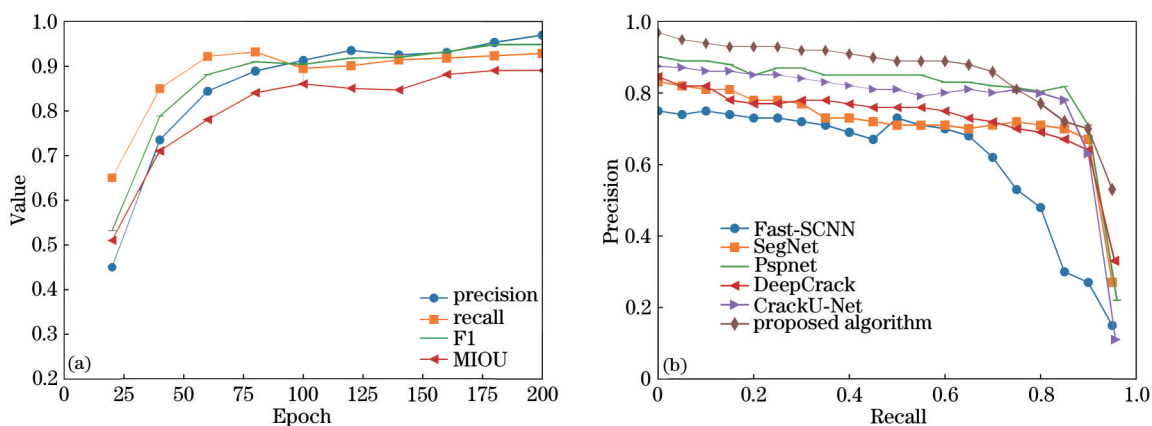


图 9 各指标的变化趋势。(a)评估指标;(b) PR 曲线

Fig. 9 Change trend of each indicator. (a) Evaluation index; (b) PR curve

评估模型的性能。由图 9(b)可以看出,所提改进模型所包含的面积最大,性能优于其他比较算法。

为了进一步验证所提改进算法的泛化能力,对

模型在 Cracktree200、GAPs384、Crack500 三个公开裂缝数据集上进行测试,结果如表 3 和图 10 所示。

可以看出,所提算法可以准确识别出图像中的细裂

表 3 不同公开数据集上所提算法的测试结果

Table 3 Test results of the proposed algorithm on different public datasets

Dataset	Precision	Recall	F1	MIoU	FPS (Millisecond/image)
Cracktree200	0.9432	0.9509	0.9470	0.8772	54.5(18.35)
GAPs384	0.9527	0.9445	0.9486	0.8495	53.8(18.59)
Crack500	0.9683	0.9485	0.9583	0.8679	56.2(17.79)

缝, 仅在部分裂纹连接处存在断裂情况, 表明所提改进算法具有良好的泛化能力, 可准确识别图像中的裂缝区域, 同时具有较好的实时性。

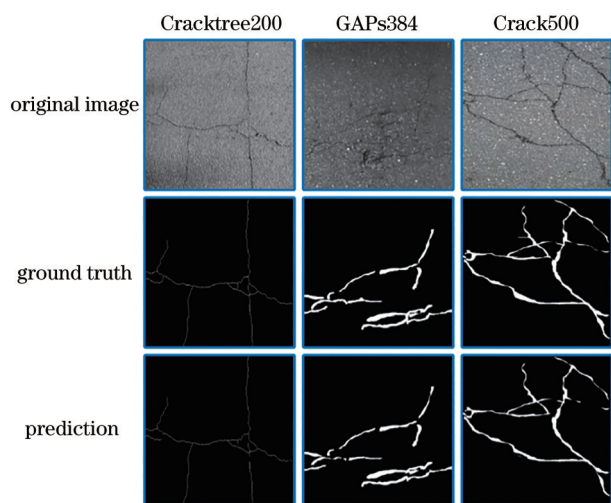


图 10 所提算法在公开数据集上的分割效果  
Fig. 10 Segmentation effects of the proposed algorithm on the public dataset

### 3.5 裂缝参数计算

为了更好地评估路面裂缝图像分割的有效性, 从模型的预测图像中提取裂缝骨架并计算裂缝的长度和宽度。如图 11 所示, 分别使用 Zhang-Suen 算法<sup>[22]</sup>、Hilditch 算法<sup>[23]</sup>和中轴法进行裂缝区域的骨架化提取。结果显示: 文献[22]的算法在细微裂缝处容易产生断点, 且骨架化边缘粗糙不够平滑; 文献[23]的算法同样会产生锯齿状的边缘, 且在不同方向的裂缝连接处会中断; 因此本文选择中轴法进行裂缝骨架提取, 便于后续的参数计算。

从测试集的预测结果中随机选择 10 幅图像, 采用中轴法计算裂缝参数, 根据裂缝左右边界上所有像素连接长度的最大值计算长度, 由裂缝骨架像素之和与裂缝长度之比计算裂缝宽度。如表 4 所示, 所提改进算法的长度和宽度相对误差均小于其他对比算法。

图 12 为所提算法在 10 幅测试图像上与裂缝真实值的对比结果。结果显示, 预测图像的裂缝像素

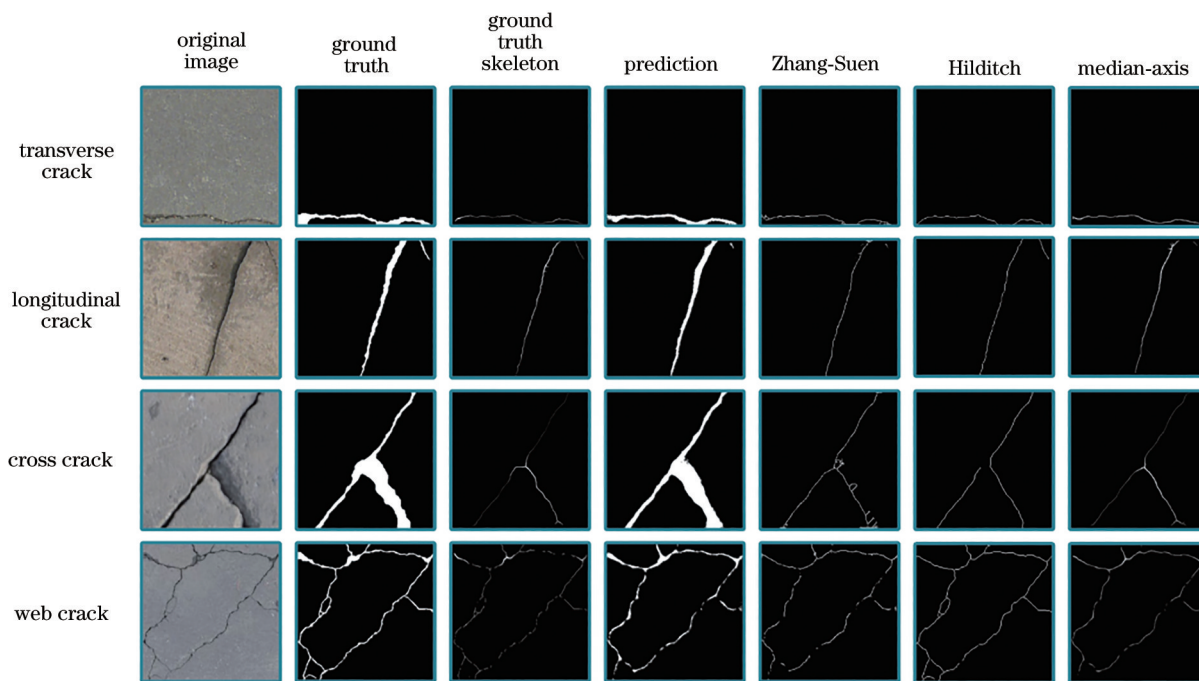


图 11 裂缝骨架提取的可视化结果示意图  
Fig. 11 Schematic of visualized results of crack skeleton extraction

表 4 不同对比算法的长度和宽度相对误差  
Table 4 Relative errors of length and width of different comparison algorithms

Parameter	Fast-SCNN	SegNet	Pspnet	DeepCrack	CrackU-Net	Proposed algorithm
Relative error of length / %	8.62	7.28	6.96	5.73	5.14	4.73
Relative error of width / %	7.94	6.53	7.32	6.87	6.02	5.21



测量值和真实标注图像的长度和宽度像素测量值的相对误差分别为 4.73% 和 5.21%。结果表明,所提算法可为公路健康状况的合理评估提供可靠的保证。

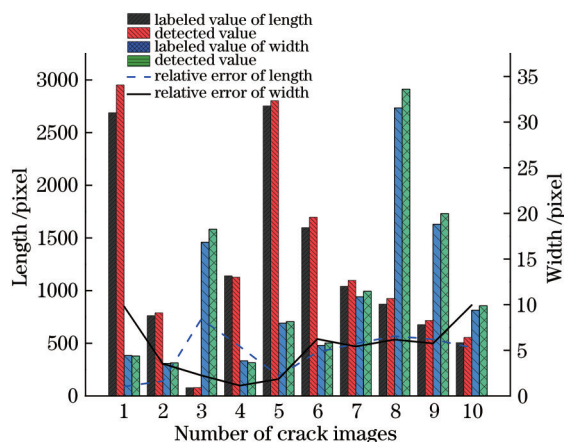


图 12 裂缝长度和宽度测量误差对比

Fig. 12 Comparison of measurement errors of crack length and width

## 4 结 论

提出了一种基于改进多分支特征共享结构网络的路面裂缝检测算法。针对路面裂缝特征与背景特征的相似性和位置的不确定性问题,对原始算法进行改进,并在训练集和交叉验证集上进行模型训练和优化。改进算法的各项评价指标均优于其他比较算法,且由于在网络中增加了多分支结构,并利用 GCN 模块与 BR 模块组成的边缘优化模块和用于特征累积的 RRC 模块组成多模块优化方式,提高了模型对裂缝边缘的分割性能和裂缝区域内部分类的鲁棒性,同时尽可能降低了模型参数量,对于原算法而言,降低了一定的检测速度。在 3 个公开裂缝数据集上进行模型测试来评估模型的泛化能力。最后使用中轴法提取裂缝骨架并进行裂缝参数计算。结果显示,改进后的裂缝检测算法具有较高检测精度和检测效率,可以满足一定的工程检测的需要。

## 参 考 文 献

[1] Fan R, Bocus M J, Zhu Y L, et al. Road crack detection using deep convolutional neural network and adaptive thresholding[C]//2019 IEEE Intelligent Vehicles Symposium, June 9-12, 2019, Paris, France. New York: IEEE Press, 2019: 474-479.

[2] 徐志刚, 赵祥模, 宋焕生, 等. 基于直方图估计和形

状分析的沥青路面裂缝识别算法[J]. 仪器仪表学报, 2010, 31(10): 2260-2266.

Xu Z G, Zhao X M, Song H S, et al. Asphalt pavement crack recognition algorithm based on histogram estimation and shape analysis[J]. Chinese Journal of Scientific Instrument, 2010, 31(10): 2260-2266.

[3] Hu Y, Zhao C X. A novel LBP based methods for pavement crack detection[J]. Journal of Pattern Recognition Research, 2010, 5(1): 140-147.

[4] Salman M, Mathavan S, Kamal K, et al. Pavement crack detection using the Gabor filter[C]//16th International IEEE Conference on Intelligent Transportation Systems, October 6-9, 2013, The Hague, Netherlands. New York: IEEE Press, 2013: 2039-2044.

[5] 王德方, 曾卫明, 王倪传. 基于改进 K-means 算法的不均匀光照下道路裂缝检测[J]. 计算机应用与软件, 2015, 32(7): 244-247, 300.

Wang D F, Zeng W M, Wang N C. Road crack detection under uneven illumination using improved K-means algorithm[J]. Computer Applications and Software, 2015, 32(7): 244-247, 300.

[6] 卢小平, 张航, 张冬梅, 等. 一种基于无人机影像的迭代二值化道路裂缝检测方法[J]. 河南理工大学学报(自然科学版), 2019, 38(6): 55-59.

Lu X P, Zhang H, Zhang D M, et al. An iterative binary road crack detection method based on UAV image[J]. Journal of Henan Polytechnic University (Natural Science), 2019, 38(6): 55-59.

[7] 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, September 25-28, 2016, Phoenix, AZ, USA. New York: IEEE Press, 2016: 3708-3712.

[8] Chen F C, Jahanshahi M R. NB-CNN: deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion[J]. IEEE Transactions on Industrial Electronics, 2018, 65(5): 4392-4400.

[9] Zou Q, Zhang Z, Li Q, et al. DeepCrack: learning hierarchical convolutional features for crack detection [J]. IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society, 2018, 28(3): 1498-1512.

[10] 朱苏雅, 杜建超, 李云松, 等. 采用 U-Net 卷积网络的桥梁裂缝检测方法[J]. 西安电子科技大学学报, 2019, 46(4): 35-42.

Zhu S Y, Du J C, Li Y S, et al. Method for bridge

- crack detection based on the U-Net convolutional networks[J]. *Journal of Xidian University*, 2019, 46(4): 35-42.
- [11] Ghosh S, Singh S, Maity A, et al. CrackWeb: a modified U-Net based segmentation architecture for crack detection[J]. *IOP Conference Series: Materials Science and Engineering*, 2021, 1080(1): 012002.
- [12] 张世宽, 吴清潇, 林智远. 焊缝图像中结构光条纹的检测与分割[J]. *光学学报*, 2021, 41(5): 0515002.  
Zhang S K, Wu Q X, Lin Z Y. Detection and segmentation of structured light stripe in weld image[J]. *Acta Optica Sinica*, 2021, 41(5): 0515002.
- [13] Poudel R P K, Liwicki S, Cipolla R. Fast-SCNN: fast semantic segmentation network[EB/OL]. (2019-02-12)[2021-06-02]. <https://arxiv.org/abs/1902.04502>.
- [14] Peng C, Zhang X Y, Yu G, et al. Large kernel matters: improve semantic segmentation by global convolutional network[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 1743-1751.
- [15] Howard A G, Zhu M L, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17) [2021-05-06]. <https://arxiv.org/abs/1704.04861>.
- [16] Zou Q, Cao Y, Li Q Q, et al. CrackTree: automatic crack detection from pavement images[J]. *Pattern Recognition Letters*, 2012, 33(3): 227-238.
- [17] Eisenbach M, Stricker R, Seichter D, et al. How to get pavement distress detection ready for deep learning? A systematic approach[C]//2017 International Joint Conference on Neural Networks (IJCNN), May 14-19, 2017, Anchorage, AK, USA. New York: IEEE Press, 2017: 2039-2047.
- [18] Yang F, Zhang L, Yu S J, et al. Feature pyramid and hierarchical boosting network for pavement crack detection[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 21(4): 1525-1535.
- [19] 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.
- [20] Zhao H S, Shi J P, Qi X J, et al. Pyramid scene parsing network[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6230-6239.
- [21] Ju H Y, Li W, Tighe S, et al. CrackU-net: a novel deep convolutional neural network for pixelwise pavement crack detection[J]. *Structural Control and Health Monitoring*, 2020, 27(8): e2551.
- [22] Zhang T Y, Suen C Y. A fast parallel algorithm for thinning digital patterns[J]. *Communications of the ACM*, 1984, 27(3): 236-239.
- [23] Hilditch C J. Comparison of thinning algorithms on a parallel processor[J]. *Image and Vision Computing*, 1983, 1(3): 115-132.