

文章编号:1001-7372(2023)03-0188-14

## 基于计算机视觉的钢桥面板裂纹识别方法

劳武略<sup>1</sup>, 崔 闯<sup>\*1</sup>, 张登科<sup>1</sup>, 罗纯坤<sup>1</sup>, 张清华<sup>1</sup>, 宋松科<sup>2</sup>

(1. 西南交通大学 桥梁工程系, 四川 成都 610031; 2. 四川省交通勘察设计研究院有限公司, 四川 成都 610017)

**摘要:**钢桥面板疲劳开裂为常见的钢桥病害,准确快速地发现并确定钢桥面板裂纹几何特征对降低运维成本、制定运维策略具有重要意义。针对传统人工巡检效率低、检测环境恶劣等问题,提出了基于计算机视觉的钢桥面板裂纹及其特征识别方法。采用目标检测网络 YoloV5 和图像语义分割网络 U-Net++ 相结合的方法识别裂纹。根据 2 个网络的结构特性标注图像中的目标后生成数据集,并分别对网络中的参数进行训练。利用训练后的 YoloV5 与 U-Net++ 分阶段对待测裂纹图像进行检测与分割,并通过阈值分割优化 U-Net++ 分割结果,再通过骨架化后得到裂纹骨架线;在确定裂纹形貌后,采用 YoloV5 识别出的标定块求解透视变换矩阵与像素尺度系数,然后对裂纹骨架线进行图像矫正并确定裂纹几何特征。研究表明:YoloV5 可准确检测出裂纹与标定块,且检测稳定性好;通过优化 U-Net++ 训练时输入的像素尺寸,提高了 U-Net++ 训练的收敛速度,网络损失由 0.121 降至 0.096;求解透视变换矩阵时,使用所有角点坐标拟合该矩阵的最小二乘解可提高图像矫正标定的精度;当图像采集距离较远、角度较大时,角点投影误差增大,且该误差对角度更为敏感;不同图像采集距离、角度下裂纹特征计算误差最大为 7.2%,表明识别精度及稳定性均较高。所提出的钢桥裂纹特征识别方法可准确计算裂纹几何特征,具有一定的实用价值。

**关键词:**桥梁工程;钢桥面板裂纹;特征计算;计算机视觉;目标检测;语义分割

中图分类号:U443.31

文献标志码:A

## Crack Identification Method of Orthotropic Steel Deck Based on Computer Vision

LAO Wu-lue<sup>1</sup>, CUI Chuang<sup>\*1</sup>, ZHANG Deng-ke<sup>1</sup>, LUO Chun-kun<sup>1</sup>,  
ZHANG Qing-hua<sup>1</sup>, SONG Song-ke<sup>2</sup>

(1. Department of Bridge Engineering, Southwest Jiaotong University, Chengdu 610031, Sichuan, China;  
2. Sichuan Provincial Communications Department Traffic Reconnaissance and Design Research Inc.,  
Chengdu 610017, Sichuan, China)

**Abstract:** Fatigue cracking of orthotropic steel deck is a common steel bridge hazard. Accurately and quickly identification and characterization of the crack geometry is important for reducing structural operation and maintenance costs and developing operation and maintenance strategies. To address the problems such as low efficiency of traditional manual inspection and harsh inspection environment, a computer vision-based method for identifying cracks and their characteristics was proposed. A combination of the object detection network YoloV5 and the

收稿日期:2021-10-11

基金项目:国家自然科学基金项目(51978579,52108176,52278318);四川省交通运输科技项目(2020-B-02);  
四川省科学技术厅项目(2021YJ0037)

作者简介:劳武略(1997-),男,湖北黄冈人,工学博士研究生,E-mail:swjtulwl@163.com。

\*通讯作者:崔 闯(1989-),男,湖北江陵人,副教授,工学博士,E-mail:ccui@swjtu.edu.cn。

semantic segmentation network U-Net++ was used for crack recognition. The objects in the images were labeled according to the structural properties of the two networks, and the parameters in the networks were trained separately. The trained YoloV5 and U-Net++ were used to detect and segment the crack image to be tested in stages. Then, the U-Net++ segmentation results were optimized by threshold segmentation and the crack skeleton lines were obtained after skeletonization. After determining the crack morphology, the calibration block identified by YoloV5 was used to solve the perspective transformation matrix and the pixel scale coefficient, followed by image correction of the crack skeleton line and determination of crack geometry features. The results indicate that YoloV5 can accurately detect cracks and calibration blocks with good stability. By optimizing the pixel size of the input for U-Net++ training, the convergence speed of the training is improved, and the network loss is reduced from 0.121 to 0.096. When solving the perspective transformation matrix, fitting the least squares solution of the matrix with all corner coordinates improves the accuracy of the image correction calibration. The corner projection error increases when the image is acquired at a greater distance and angle, and the error is more sensitive to the angle. The maximum error in the calculation of crack features for different image acquisition distances and angles is 7.2%, indicating high recognition accuracy and stability. The proposed method for identifying crack features in steel bridges can accurately calculate crack geometries and is of great practical value.

**Keywords:** bridge engineering; cracks of orthotropic steel deck; feature calculation; computer vision; object detection; semantic segmentation

**Received** 11 Oct. 2021

**Funding:** National Natural Science Foundation of China (51978579, 52108176, 52278318); The Transportation Science and Technology Project of Sichuan Province (2020-B-02); The Science and Technology Department Project of Sichuan Province (2021YJ0037)

\*Corresponding author. *E-mail address:* ccui@swjtu.edu.cn.

<https://doi.org/10.19721/j.cnki.1001-7372.2023.03.015>

## 0 引言

正交异性钢桥面板具有轻质高强、适用范围广等突出优点,已成为钢桥的首选桥面板结构形式<sup>[1]</sup>。但在其构造特性、制造工艺、运营环境等多因素耦合作用下,极易出现疲劳裂纹,严重影响结构服役性能甚至危及结构安全<sup>[2]</sup>。及早发现并处置疲劳裂纹可有效提高结构服役质量、降低维护成本,并最大限度地提升钢桥使用寿命。对于钢桥表面裂纹,人工巡检是当前常用方法。但该方法不仅耗费高昂、检测环境恶劣、检测时效性低、漏检率高,且对检测人员的专业知识和经验要求较高<sup>[3-4]</sup>。因此,发展高效准确的裂纹检测手段,提高检测效率,并及时测算裂纹几何特征,实现桥梁损伤状态的快速甄别,是当前桥梁工程亟需解决的热点和难点问题。

随着计算机视觉领域深度学习的发展,其在结构损伤的检测定位中得到广泛的应用<sup>[5-6]</sup>。卷积神

经网络(Convolutional Neural Network, CNN)因其优异的图像识别性能,且无需复杂的数据处理与特征设计环节,现已成为结构损伤检测中常用的技术之一,国内外学者围绕该研究热点进行了大量研究。沙爱民等<sup>[7]</sup>将CNN用于识别路面裂缝、坑槽等病害,但检测结果中含有较多误检的子图块。为达到更好的识别效果,Jang等<sup>[8]</sup>利用无人机采集的混凝土桥塔裂缝图像对GoogleNet模型进行迁移学习,Dung等<sup>[9]</sup>与Xu等<sup>[10]</sup>利用消费级相机采集实际桥梁的裂缝图像,并分别通过数据增强、权重微调、层级特征融合等方法实现了病害的准确定位。上述研究均属于滑动窗口生成子图块并进行分类的图像分类问题,该类方法无法辨识图像的全局特征,存在较大的局限性。因此需根据损伤的表观特征及实际的应用需求,采用特定的神经网络模型对图像中的损伤进行检测与分割。Ye等<sup>[11]</sup>设计了一种特殊的全卷积CNN结构,Zhang等<sup>[12]</sup>利用多个线性

滤波器提取图像特征,均实现了混凝土裂缝的目标分割,但由于复杂背景的干扰,分割结果中含有较多噪声。随着近年来深度学习理论的发展,国内外学者提出了众多性能优异的目标检测与分割模型,如以 R-CNN(Region-CNN)系列<sup>[13-14]</sup>为代表的“Two-Stage”模型和以 Yolo(You Only Look Once)<sup>[15-16]</sup>为代表的“One-Stage”模型,该类模型在结构损伤检测领域得到了广泛的应用。Chen 等<sup>[17]</sup>利用 2 个“One-Stage”模型分阶段检测了高速摄像机采集的紧固件缺陷图像,Cha 等<sup>[18]</sup>将 Faster R-CNN 应用于多种桥梁病害的检测。上述方法对形状纹理特征突出的目标具有较好的检测效果,但难以保证其在复杂背景中分割微损伤的精度。因此,晏班夫等<sup>[19]</sup>结合 Faster R-CNN 与图像形态学方法对路面缺陷进行分割;Guo 等<sup>[20]</sup>利用 Mask-RCNN 与 U-Net 对混凝土表面多裂缝目标进行检测与分割。在实际应用过程中,“Two-Stage”模型复杂、结构冗余,比“One-Stage”模型检测精度更高,但对训练集要求高且检测速度慢。为提高检测效率,在不影响检测精度的前提下,本文融合轻型的目标检测与分割网络协同执行疲劳裂纹的检测任务。

在确定裂纹位置后需进一步确定裂纹长度、方向等几何特征参数,专家学者就标定裂纹的几何特征开展了大量研究。Guo 等<sup>[20]</sup>计算了混凝土表面裂缝数量及宽度等信息,但并未考虑成像透视变形导致的形态特征测量难的问题。刘宇飞等<sup>[21]</sup>与 Kalfarisi 等<sup>[22]</sup>将裂缝的检测与分割结果映射至由多张图像构建的三维模型中并测量其几何特征,但该方法复杂低效,且精度对图像质量较为敏感。不同于上述方法,Wang 等<sup>[23]</sup>利用爬壁机器人采集含校准板的裂纹图像并对其进行拼接、标定后计算裂纹特征。因此,引入先验知识或约束条件对图像进行标定是提高几何特征测量效率的有效方法。

本文提出了一种基于计算机视觉的钢桥裂纹特征识别方法:利用目标检测识别网络 YoloV5 和图像语义分割网络 U-Net++,分阶段实现裂纹的检测与分割,并通过阈值分割优化分割结果,通过骨架化得到裂纹的骨架线;在此基础上,利用 YoloV5 检测出的标定块求解透视变换矩阵与像素尺度系数,并对裂纹骨架线进行图像矫正与测定。最后利用实际裂纹图像对所提出的钢桥裂纹特征识别方法的鲁棒性与精度进行了验证,以期对钢桥疲劳裂纹数字化检测方法提供理论和技术支撑。

## 1 钢桥裂纹特征识别方法

### 1.1 基本原理

钢桥裂纹特征智能识别方法框架如图 1 所示,其中图像数据准备、模型训练为该方法的准备阶段,检测与分割、特征计算为该方法的应用阶段。准备阶段中图像数据集的质量与模型的训练是该方法鲁棒性的保证,因受钢桥裂纹样本数量限制,在模型训练过程中利用多种手段实现数据增强。特征计算时,角点检测精度决定了透视变换矩阵与像素尺度系数的求解精度。由于角点检测难免存在误差且误差分布较为离散,利用最小二乘法拟合透视变换矩阵可减小角点检测误差对裂纹特征计算的影响。



图 1 裂纹特征识别方法

Fig. 1 Crack Feature Recognition Method

### 1.2 目标检测网络算法

目标检测网络 YoloV5 主要由 Backbone(骨干网络)、Neck(特征融合提取结构)、Head(检测头)3 个模块组成,如图 2 所示<sup>[17]</sup>:原始输入图像的像素尺寸为  $640 \times 640 \times 3$  (3 为颜色通道数);在 Backbone 模块,首先利用 Focus(注意力机制)对输入进行像素间隔采样,生成 4 张图像并对其进行卷积后生成初始特征图,再利用 DarkNet53 网络提取初始特征图中的纹理、形状等特征;在 Neck 模块,利用多尺度池化、双向特征金字塔结构对不同尺度特征进行拼接融合,生成的特征图尺寸为  $128 \times 80 \times 80$ 、 $256 \times 40 \times 40$ 、 $512 \times 20 \times 20$ ;在 Head 模块,以 3 个卷积层作为检测器将特征图映射为候选框的坐标  $(x, y, w, h)$ 、置信度  $(C)$ 、概率分布  $(p_1, p_2)$  等 7 个参数,其中  $C$  反映了候选框内是否有目标的概率,

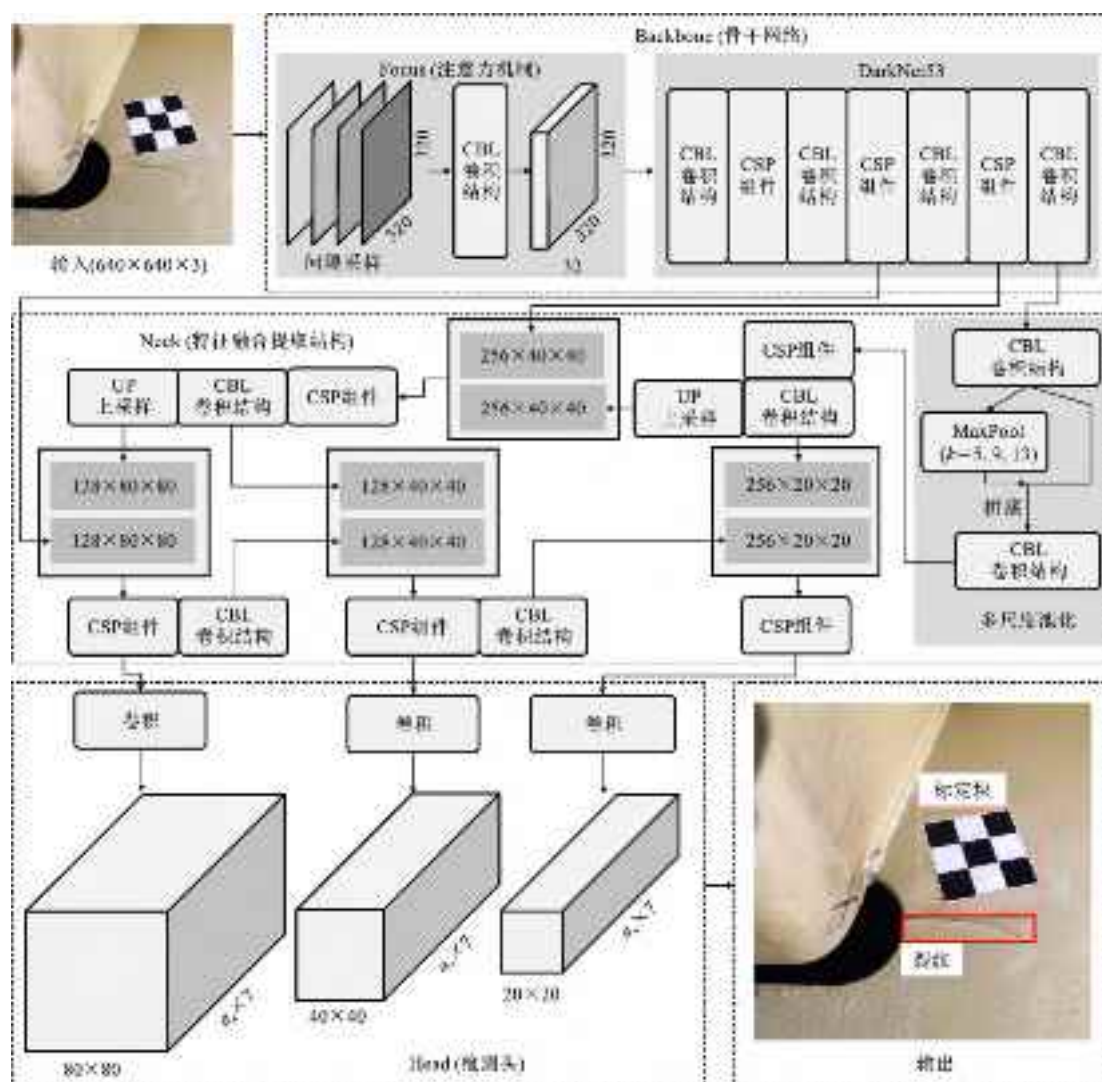


图2 YOLOv5 结构

Fig. 2 Structure of YOLOv5

$(p_1, p_2)$  则分别对应裂纹与标定块 2 类目标的分类概率。在 Head 模块中, 3 组特征图分别用于预测大、中、小目标, 各特征值的先验框数量为  $a_n$ 。因此每个尺度下的参数预测数量为  $s \times s \times a_n \times 7$ , 其中  $s$  为特征图的大小。

### 1.3 裂纹分割网络算法

YOLOv5 检测出的矩形候选框中, 裂纹所占面积较小, 其他均为背景信息, 直接对该候选框阈值分割将会因背景信息的误分割而引入大量噪声, 不利于后续裂纹几何特征的提取。为剔除候选框内的背景信息, 本文利用图像语义分割网络 U-Net++ 将候选框内像素点分类为裂纹和背景。U-Net 是近年来较为常用的一种网络结构, 该结构中的 U 型编码-解码框架简洁有效, 并通过上采样、下采样、层间跳接的方式实现特征的提取和融合<sup>[24]</sup>。U-Net++ 则是由不同深度的 U-Net 组成, 其解码器以相同的

分辨率密集连接, 使得特征图充分融合<sup>[25]</sup>。该结构采用的深度监督机制是将 U-Net++ 中间层的损失引入损失函数中并进行反向传播, 该机制可使浅层能够得到充分的训练。针对本文中的小数据集、细长裂纹目标的语义分割任务, 微调了 U-Net++ 结构, 如图 3 所示: 将 U-Net++ 模型中基础的卷积结构改为残差结构, 使网络训练收敛更快并减少过拟合现象; 在 U-Net++ 最下层添加注意力机制, 提高网络对小裂纹目标的敏感程度。

### 1.4 图像矫正与标定

钢桥裂纹图像采集过程中, 投影面与裂纹所在平面(物面)难以保证为平行关系, 此时视平面中的物像不能真实反映出物体几何特征。针对 YOLOv5 模型检测出的标定块目标, 利用 Harris 角点检测算法定位标定块的角点, 并对其进行匹配。角点检测即确定标定块角点在图像中的像素坐标, 角点匹配



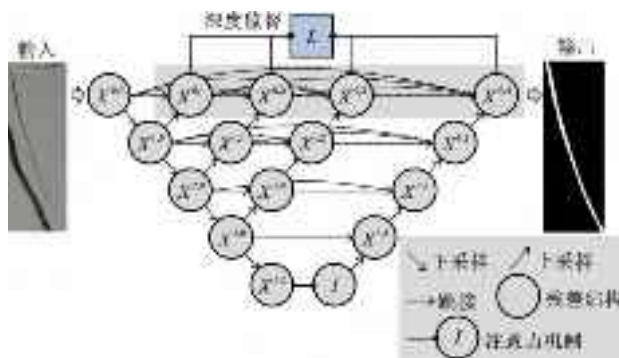


图3 U-Net++结构

Fig. 3 Structure of U-Net++

即确定角点的相对位置并指定各个角点变换后的像素坐标。在此基础上,求解透视变换矩阵并对图像进行透视变换,即以标定块为参照物重新建立投影体系,将视平面中的物像投影至与物面平行的重投影面上,如图4所示,其中物面、视平面及视点构成了基本的透视投影模型。

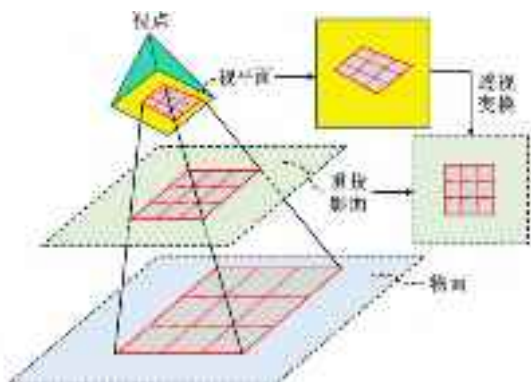


图4 透视变换过程

Fig. 4 Process of Perspective Transformation

从数学角度而言,透视变换是一个坐标系转换过程,包含了线性变化、平移变化、透视变化,其中线性变化与平移变化不会改变图中的平行关系,透视变化会在三维空间中改变重投影面方位。为计算裂纹几何特征,透视变化中取重投影面与物面平行。平面图像透视变换的数学变换过程为<sup>[26]</sup>

$$S \times \begin{bmatrix} x_{\text{img}} \\ y_{\text{img}} \\ 1 \end{bmatrix} = M \times \begin{bmatrix} x'_{\text{img}} \\ y'_{\text{img}} \\ 1 \end{bmatrix}, M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \quad (1)$$

式中: $M$ 为透视变换矩阵; $(x_{\text{img}}, y_{\text{img}})$ 为某点在重投影面中的像素坐标; $(x'_{\text{img}}, y'_{\text{img}})$ 为该点在视平面中的像素坐标; $S$ 为尺度因子。 $M$ 矩阵中有8个未知参数,因此需要4个关键点的变换前后像素坐标,当关键点数量大于4时,将采用最小二乘法求解最优的透视变换矩阵。

透视变换后的图像中的每个像素具有相同的像素率 $R$ ,即每个像素的实际尺度相同。根据标定块的实际尺寸信息即可求得 $R$ ,计算公式为

$$R = \frac{L_{\text{real}}}{L_{\text{pixel}}} \quad (2)$$

式中: $L_{\text{real}}$ 为标定块的实际长度; $L_{\text{pixel}}$ 为其标定块像素长度。

### 1.5 裂纹几何特征

钢桥裂纹长度是结构安全性能的控制因素,本文将通过裂纹中轴线计算裂纹长度。由于裂纹区域宽度较小且形状不规则,制作U-Net++数据集时存在标注误差,导致该模型在预测时同样存在该类误差,即将裂纹区域邻近像素归为裂纹目标。该误差使得U-Net++的分割结果是偏保守的,因此需通过阈值分割优化U-Net++分割结果,以得到更为精确的裂纹分割结果,并通过骨架化得到裂纹骨架线。最后,通过透视变换矩阵、像素尺度系数赋予骨架线实际的几何意义,此时该骨架线即为裂纹中轴线。

U-Net++分割后的裂纹局域内噪声较少,且裂纹灰度特征突出,但当光照不均时局部灰度特征将会弱化,因此本文采用自适应阈值分割算法对该局域进行图像二值化。自适应阈值分割通过计算某个邻域的灰度分布特征确定该邻域内的二值化阈值,由此确定的阈值可有效克服光照不均因素。进一步针对该二值图进行骨架化得到宽度为单位像素的骨架线。本文利用八邻域骨架化算法对裂纹二值图进行骨架化处理,并将裂纹目标灰度值取255,非裂纹目标灰度值取0,该算法过程为:①遍历二值图中每个目标像素点,逐个判断其八邻域中各个像素灰度值是否为255,并返回其逻辑值;②上述逻辑值矩阵与 $2^0, 2^1, \dots, 2^7$ 组成的矩阵点乘,利用点乘结果判断该目标像素点是否为骨架点,如图5所示;③删除非骨架点(将非骨架点灰度值置为0);④重复上述步骤直至二值图不再发生变化,最终结果即为裂纹骨架线。利用透视变换矩阵对该骨架线进行图像矫正,并对骨架线逐个相邻像素点间距进行累计求和后通过像素尺度系数换算为裂纹的实际几何长度。

## 2 试验验证

### 2.1 图像数据准备

钢桥裂纹图像数据采集前,应在结构表面附着标定块。采集过程中应保证图像中标定块和裂纹的



图5 八邻域骨架化算法

Fig. 5 Eight Neighborhood Skeletonization Algorithm

完整性且处于同一平面上,但对图像采集的角度、距离、光线无特殊要求。分别在实验室及实桥环境下,采集多个钢桥面板疲劳裂纹图像共160张,如图6所示。图像采集完成后,需根据YoloV5与U-Net++网络结构基本特性对图像进行标注:YoloV5模型数据集需要目标真实框坐标信息 $(x, y, w, h)$ 及对应的标签信息 $(p_1, p_2)$ ;U-Net++模型数据集需裁剪出原始图像中的裂纹目标,并标注裂纹边界的像素坐标信息,如图7所示。

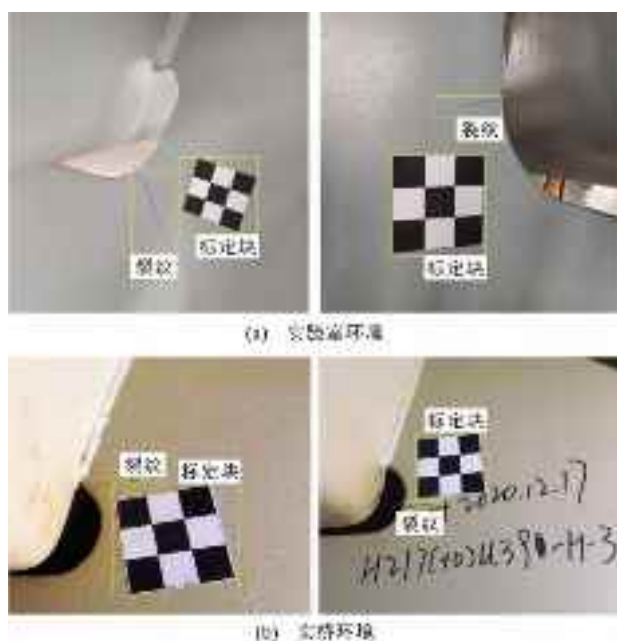


图6 原始图像采集

Fig. 6 Original Image Acquisition

U-Net++训练过程中是在训练集中随机抽取一定数量的训练数据作为一个批次输入网络,并将该批次训练数据卷积、采样后的特征图进行拼接。

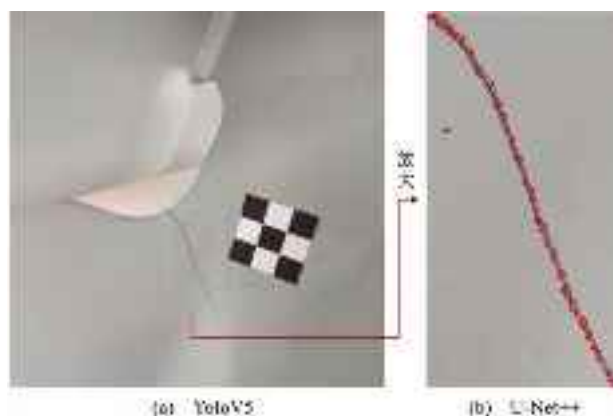


图7 数据集标注

Fig. 7 Annotation of Dataset

此外,U-Net++模型包含5层下采样层,最大的下采样率为 $32(2^5)$ ,若某个输入的像素尺寸不为32的倍数,在网络下采样、上采样过程中,将会因约减使特征图大小不一致。因此为保证U-Net++训练过程中特征图的顺利拼接,训练集的像素尺寸应相同且需设置为32的倍数。针对本文中的裂纹分割任务,在得到U-Net++数据集后需进一步统计图像像素尺寸分布,确定U-Net++输入的最优尺寸。经裁剪后的裂纹图像大小及长宽比分布较为离散,其统计结果如图8所示,其中线性拟合的图像长 $(L)$ 、宽 $(W)$ 关系如式(3)所示,图像长宽的均值分别为265、424。当U-Net++输入图像的长宽比满足上述线性关系时,可减小因图像长宽比及图幅大小变化导致的数据畸变;针对本文中的裂纹分割数据集,后文将以模型的性能为指标确定最优的图像输入尺寸。

$$L = 0.9 \times W + 184 \quad (3)$$

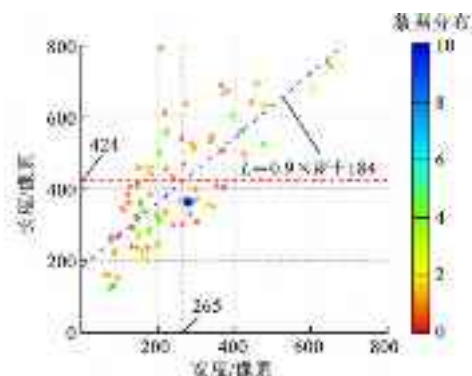


图8 U-Net++数据尺寸分布

Fig. 8 Distribution of U-Net++ Data Size

YoloV5与U-Net++的数据集标注完成后将其输入对应的网络模型中,训练过程中对数据集进行线上数据增强,即每次迭代均需对输入数据进行

旋转、缩放与添加噪声等处理,且上述处理均是随机的,以此实现训练样本的数据增强,提高检测模型的泛化能力<sup>[27]</sup>,如图 9(a)~(c)所示。YoloV5 在训练过程中除采用上述数据增强方法外,还对其训练数据进行 Mosaic 增强,即随机抽取多张训练图片进行随机缩放、裁剪、排布,使其组合为一张训练图片,以此丰富训练过程中的样本形式,如图 9(d)所示。

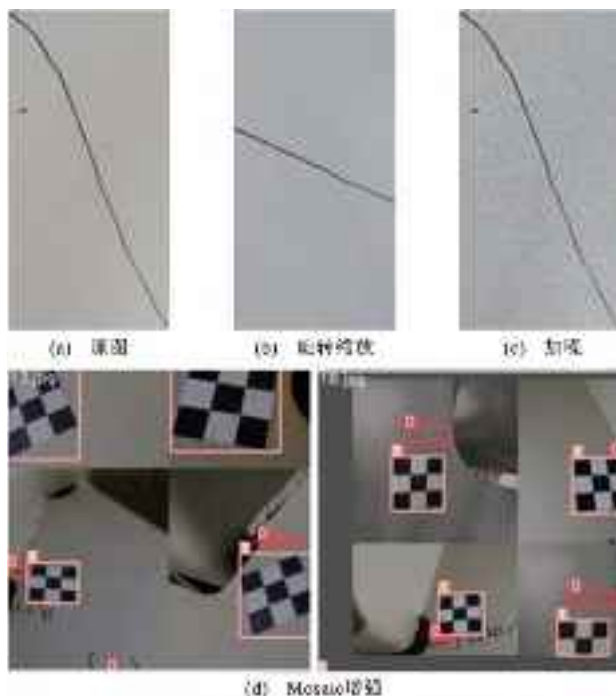


图 9 数据增强

Fig. 9 Data Augmentation

## 2.2 网络性能验证

### 2.2.1 目标检测网络性能验证

本文采用迁移学习的方式训练 YoloV5,即预加载已在 MS COCO 的 Detection 2017 数据集训练后的模型参数,使模型快速收敛。YoloV5 所采用的 anchor 是一组宽高固定的初始候选框,由于每个图像中的目标数量较少, $a_n$  取为 2,训练过程中需要统计数据集中目标真实框的长宽分布,并通过 k-means 聚类分析出 anchors 大小,以提升其检测性能。YoloV5 模型中,小的特征图其感受野大,因此  $20 \times 20$  的特征图用于检测大目标,利用本文数据聚类后的 anchor 大小为  $[(218, 262)(373, 304)]$ 。同理,  $40 \times 40$  与  $80 \times 80$  的特征图分别用于预测中、小目标,各自聚类后的 anchor 大小分别为  $[(218, 72)(169, 157)]$ 、 $[(94, 68)(76, 168)]$ 。

为真实反映模型检测性能,需统计分析 CNN 识别结果的精确率  $p$  和召回率  $r$ ,以及评价模型预测能力的综合指标  $m_{AP}$ ,如式(4)所示

$$p = \frac{T_P}{T_P + F_P}, r = \frac{T_P}{T_P + F_N}, m_{AP} = \int_0^1 P(R) dR \quad (4)$$

式中:  $T_P$  为真阳性识别结果;  $F_P$  为假阳性识别结果;  $F_N$  为假阴性识别结果。

对上述指标进行分析时,需引入置信度阈值 ( $C_0$ )与交并比 ( $I_{oU}$ ):当候选框的  $C$  大于  $C_0$  时才将其作为预测框;对于所有预测框,其  $I_{oU}$  计算公式如式(5)所示

$$I_{oU} = \frac{\text{预测框} \cap \text{真实框}}{\text{预测框} \cup \text{真实框}} \quad (5)$$

$I_{oU}$  反映了预测框与真实框的相关度。取交并比阈值  $I_0$ ,当预测框的  $I_{oU}$  值大于  $I_0$  时预测结果为  $T_P$ ,当预测框的  $I_{oU}$  值小于  $I_0$  时预测结果为  $F_P$ ,当预测框与真实框没有交集即  $I_{oU}$  值为 0 时预测结果为  $F_N$ 。因此  $p, r$  将会同时受到  $C_0$  与  $I_0$  的影响,在计算  $p, r$  时取  $I_0$  为某一固定值,取使  $p, r$  的调和平均  $f_1$  最大时的置信度作为  $C_0$ ,如式(6)所示

$$f_1 = \frac{2pr}{p+r} \quad (6)$$

计算  $m_{AP}$  则不采用上述  $C_0$  选取原则,而是通过使用不同的  $C_0$  获得模型的  $p-r$  曲线,并积分后求得。因此,式(4)中  $p, r$  分别衡量了模型的查准度、查全度,  $m_{AP}$  则综合考虑了上述 2 个因素,  $m_{AP}$  值越大模型性能越优。

YoloV5 模型训练过程中  $p, r$  及  $m_{AP}$  变化趋势如图 10(a)、(b)所示,其中  $m_{AP}@0.5$  表示  $I_0$  取 0.5 时的  $m_{AP}$  值,  $m_{AP}@0.5:0.95$  表示  $I_0$  以 0.05 为步长、取 0.5~0.95 区间值时所有  $m_{AP}$  的平均值。训练结束后遍历 0~1 范围内的  $C_0$ ,得到模型的  $p-r$  曲线如图 10(c)所示。由图 10 可知:训练前期网络泛化能力较差,采用的数据增强方法使得每次输入网络的数据各不相同,因此网络性能波动较大;经 100 次迭代训练后,  $p, r$  及  $m_{AP}@0.5, m_{AP}@0.5:0.95$  均已平稳,并分别达到了 97.9%、98.6%、0.94、0.84,  $p-r$  曲线分布靠近右上角,表明网络已达到较好的检测精度与稳定性。

最终, YoloV5 模型的检测效果如图 11 所示,其中“calibration、crack”为目标标签,“0.80、0.91”等为候选框内目标的分类概率。

### 2.2.2 裂纹分割网络性能验证

U-Net++ 每层的损失函数  $L(Y, P^i)$  由逐像素交叉熵及  $D_{ice}$  系数组成,其中交叉熵损失用于度量误分类的概率,  $D_{ice}$  系数反映预测掩码和真实掩码的重叠程度,当预测掩码与真实掩码完全重合时  $D_{ice}$  系数为 0,采用深度监督机制时总损失  $L$  为每层损



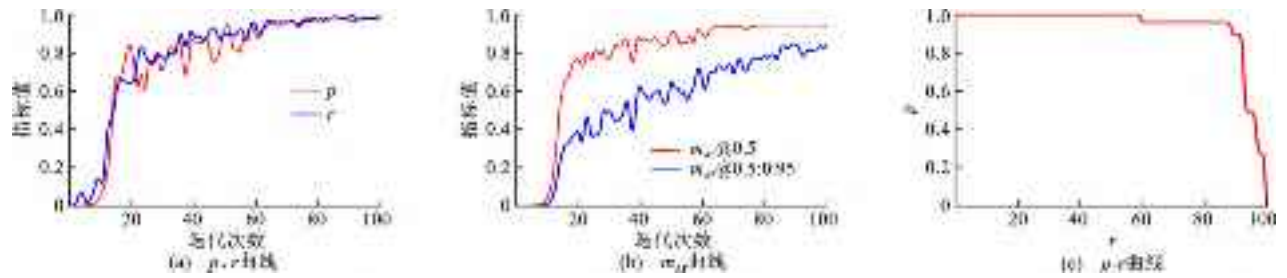


图 10 YoloV5 性能曲线

Fig. 10 Curves of YoloV5 Performance

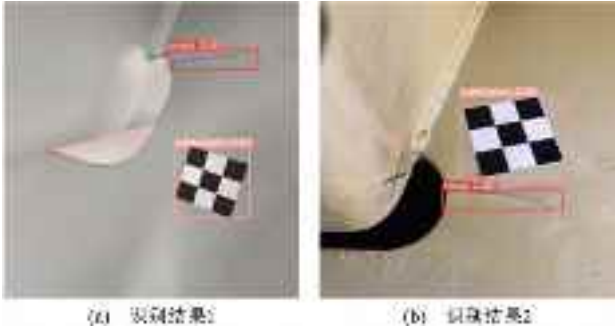


图 11 YoloV5 检测结果

Fig. 11 Detection Results of YoloV5

失  $L(Y, P^i)$  之和,如式(7)所示<sup>[27]</sup>

$$\left. \begin{aligned} D_{ice} &= 1 - \frac{2y_{n,c}p_{n,c}}{y_{n,c}^2 + p_{n,c}^2} \\ L(Y, P) &= \frac{1}{N} \sum_{c=1}^C \sum_{n=1}^N (-y_{n,c} \log(p_{n,c}) + D_{ice}) \\ L &= \sum_{i=1}^d \eta_i L(Y, P^i) \end{aligned} \right\} \quad (7)$$

式中: $y_{n,c}$ 、 $p_{n,c}$  分别为某个训练批次中第  $n$  个像素、第  $c$  类别的真实值和预测值; $N$  为当前训练批次中的像素总数; $d$  为解码层的层数; $\eta_i$  为每层损失的权重,均取为 0.25。

分割网络交并比( $I_{ou}$ )与目标检测网络类似,是直接度量预测掩码准确性的指标, $I_{ou}$  值为 1 时表明预测掩码与真实掩码完全重叠,如式(8)所示

$$I_{ou} = \frac{y_{n,c} p_{n,c}}{(y_{n,c} \cup p_{n,c})^2} \quad (8)$$

U-Net++ 训练时每个训练批次大小为 16,并指定学习率为 0.001、迭代次数为 200、训练集与验证集的比例为 4:1。根据本文第 2.1 节中 U-Net++ 训练集尺寸统计结果,将训练集设置为多个不同的像素尺寸,如表 1 所示;深度监督机制下 U-Net++ 训练过程中损失函数及  $I_{ou}$  变化如图 12 所示。由图 12 可知:Data3、Data4、Data5 训练模型时,其收敛速度显著优于 Data1 与 Data2,表明输入的像素尺寸及长宽比是影响数据集质量的关键因素;Data3

表 1 数据集尺寸

Table 1 Size of Datasets

数据集	Data1	Data2	Data3	Data4	Data5
尺寸	96×256	256×256	256×416	288×488	416×416
$(L-184)/W$	0.75	-	0.91	1.06	-
$L/W$	-	1	-	-	1

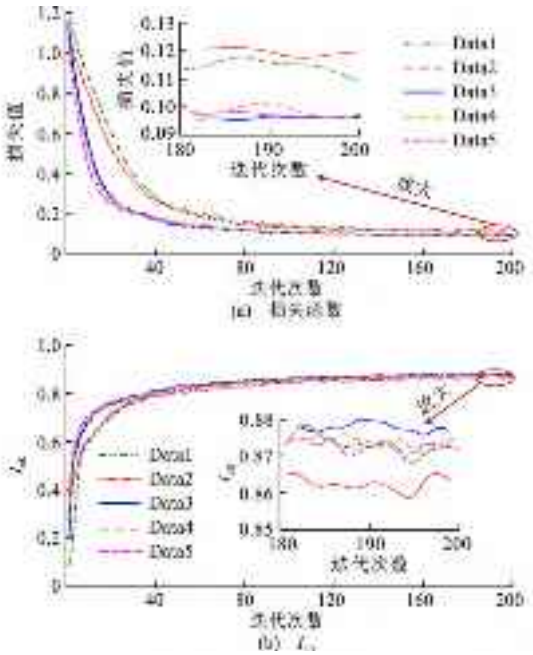


图 12 U-Net++ 性能曲线

Fig. 12 Curves of U-Net++ Performance

训练后的模型性能略优于 Data4 与 Data5,表明进一步增大像素尺寸不能提升网络性能,且增大像素尺寸将会增加网络计算量。综上,本文将训练集输入的像素尺寸取为  $256 \times 416$ ,此时网络损失由 0.121 降至 0.096,  $I_{ou}$  由 0.861 提升至 0.878。

U-Net++ 模型在实际应用过程中,无需将待分割裂纹图像按批次输入网络中,此时图像像素尺寸只需设置为 32 的倍数,无需改变长宽比。最终, U-Net++ 模型对不同裂纹图像的分割效果如图 13 所示。



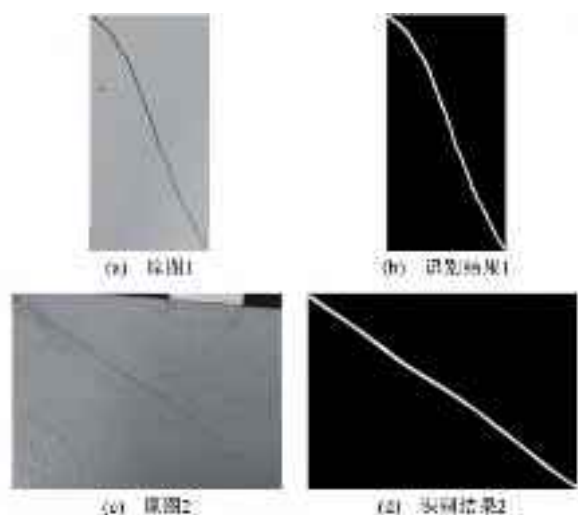


图 13 U-Net++ 分割结果

Fig. 13 Segmentation Results of U-Net++

## 2.3 图像矫正与标定结果分析

### 2.3.1 角点检测及匹配

标定块角点的定位与匹配是平面标定法的关键,直接影响了裂纹几何特征的计算精度。角点检测过程中,由于标定块候选框内背景较为单一且标

定块角点处的灰度梯度较大,利用 Harris 算法即可较为准确地检测标定块角点。角点检测过程如图 14(a)所示,可知:Harris 算法在邻域处理过程中会忽略图像边缘的角点响应值,因此标定块的部分角点被漏检,但已检测出的 12 个角点仍能确保最小二乘法拟合出的透视变换参数具有较高精度。

对角点进行匹配需确定角点间的相对位置关系。本文根据 Harris 算法检测过程中像素兴趣值分布情况将标定块角点分为 2 类(外角点与内角点),如图 14(b)所示:角点响应值存在 4 个分布较大的梯度方向,则该角点为内角点;角点响应值仅存在 2 个分布较大的梯度方向,则该角点为外角点。在此分类条件下,内角点数量为 4,且 Harris 算法对边缘角点不敏感的特性对内角点的检测无影响。以内角点构成的四边形为基础,匹配各个角点的实际位置,如图 14(c)所示,并指定各角点变换后的像素坐标。为保证透视变换前后图像清晰度,取  $L_{\text{pixel}}$  为  $W$  与  $H$  中的最大值,最终透视变换前后角点像素坐标如表 2 所示。

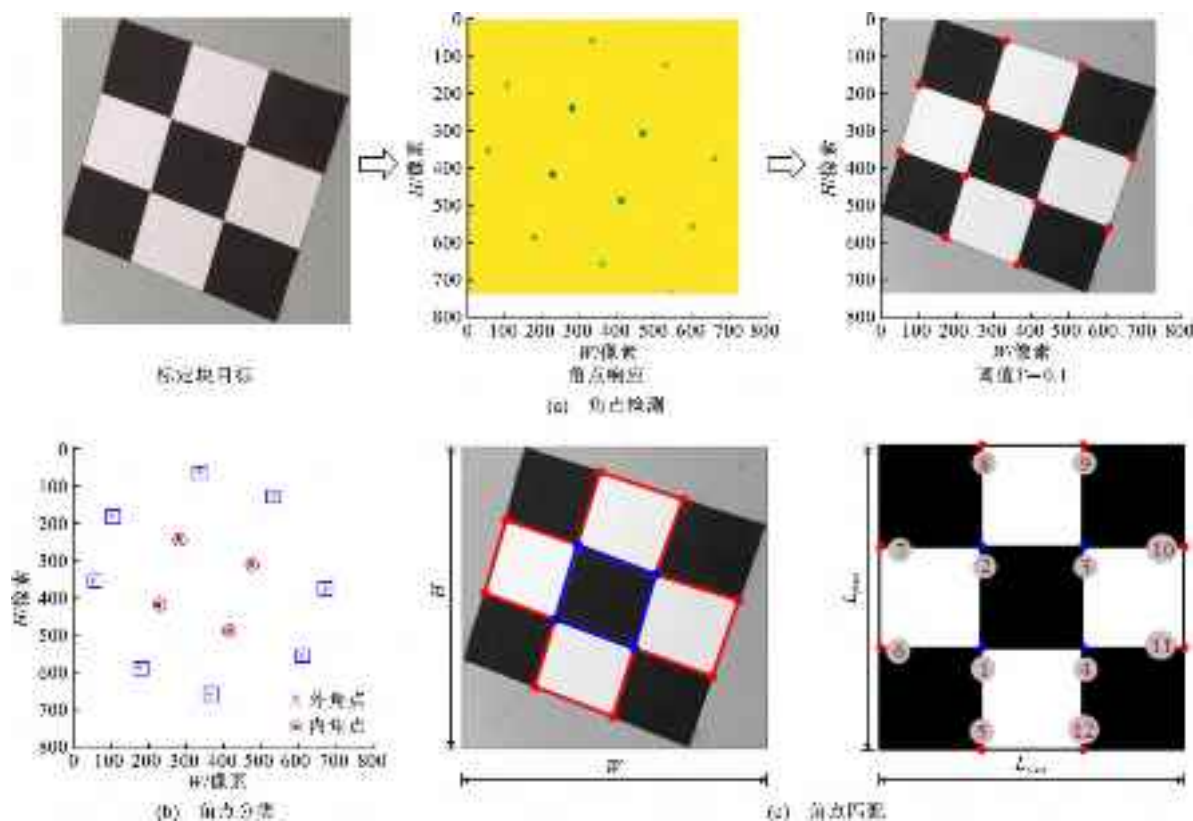


图 14 角点检测及匹配过程

Fig. 14 Process of Corner Detection and Matching

表 2 透视变换前后角点坐标及变换误差

Table 2 Corner-point Coordinates and Transformation Errors Before and After Perspective Transformation		像素											
项目		编号											
		1	2	3	4	5	6	7	8	9	10	11	12
$(x,y)$		(224,419)	(278,237)	(468,310)	(417,490)	(174,583)	(53,355)	(606,558)	(333,60)	(105,177)	(670,373)	(365,656)	(535,127)
$(x_{img},y_{img})$		(246,246)	(246,492)	(492,492)	(492,246)	(246,0)	(0,246)	(0,492)	(246,738)	(492,738)	(738,492)	(738,246)	(492,0)
$(\Delta x,\Delta y)$	$n=4$	(0,0)	(0,0)	(0,0)	(0,0)	(-4,21)	(26,-1)	(-4,5)	(5,-3)	(26,-5)	(8,15)	(-4,20)	(15,8)
	$n=12$	(3,3)	(2,0)	(3,3)	(0,5)	(3,-1)	(-3,3)	(-2,5)	(4,-1)	(-2,2)	(-3,-4)	(0,-3)	(-3,-5)
$\Delta M_n$		$L_{pixel}=738$ 像素, $\Delta M_4=0.19$ , $\Delta M_{12}=0.07$											

2.3.2 透视变换矩阵及像素率求解

对于上述多对透视变换前后角点像素坐标,可将式(1)改写为如下形式

$$\begin{bmatrix} x'_{img,1} & y'_{img,1} & 1 & 0 & 0 & 0 & -x_{img,1}x'_{img,1} & -x_{img,1}y'_{img,1} \\ 0 & 0 & 0 & x'_{img,1} & y'_{img,1} & 1 & -y_{img,1}x'_{img,1} & -y_{img,1}y'_{img,1} \\ x'_{img,2} & y'_{img,2} & 1 & 0 & 0 & 0 & -x_{img,2}x'_{img,2} & -x_{img,2}y'_{img,2} \\ 0 & 0 & 0 & x'_{img,2} & y'_{img,2} & 1 & -y_{img,2}x'_{img,2} & -y_{img,2}y'_{img,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_{img,n} & y'_{img,n} & 1 & 0 & 0 & 0 & -x_{img,n}x'_{img,n} & -x_{img,n}y'_{img,n} \\ 0 & 0 & 0 & x'_{img,n} & y'_{img,n} & 1 & -y_{img,n}x'_{img,n} & -y_{img,n}y'_{img,n} \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \end{bmatrix} = \begin{bmatrix} x_{img,1} \\ y_{img,1} \\ x_{img,2} \\ y_{img,2} \\ \vdots \\ x_{img,n} \\ y_{img,n} \end{bmatrix} \quad (9)$$

式中: $n$  为角点数量。

当  $n=4$  时,该方程组有唯一解即透视变换矩阵唯一确定;当  $n>4$  时,该方程组为超定方程组,其最小二乘解即为最优的透视变换矩阵。利用该矩阵对标定块进行透视变换,结果如图 15 所示。由图 15 可知:当采用所有检出角点( $n=12$ )求解透视变换矩阵时,标定块的透视变换结果的矫正效果优于仅采用 4 个内角点时的透视变换结果( $n=4$ )。上述 2 种情形下分别计算各角点的投影误差如表 2 中( $\Delta x$ ,  $\Delta y$ )所示,并根据式(10)计算所有角点的投影误差

由表 2 可知:Harris 算法在定位角点坐标时存在偶然误差,导致角点投影误差分布较为离散;当  $n=4$  时,仅在 1~4 号角点投影误差为 0,在其他部分角点上投影误差较大; $\Delta M_4$  大于  $\Delta M_{12}$ ,表明利用多角点拟合透视变换矩阵可显著减小透视变换后的标定块畸变误差。

根据式(2)计算像素率时, $L_{real}$  为标定块的实际长度,本文标定块的尺寸为 6 cm×6 cm,即  $L_{real}=6$  cm,因此可直接求解出  $R$  为

$$\Delta M_n = \left( \sum_i^n \sqrt{\Delta x^2 + \Delta y^2} \right) / nL_{pixel} \quad (10)$$

$$R = \frac{L_{real}}{L_{pixel}} = 0.081 \text{ mm} \cdot \text{像素}^{-1} \quad (11)$$

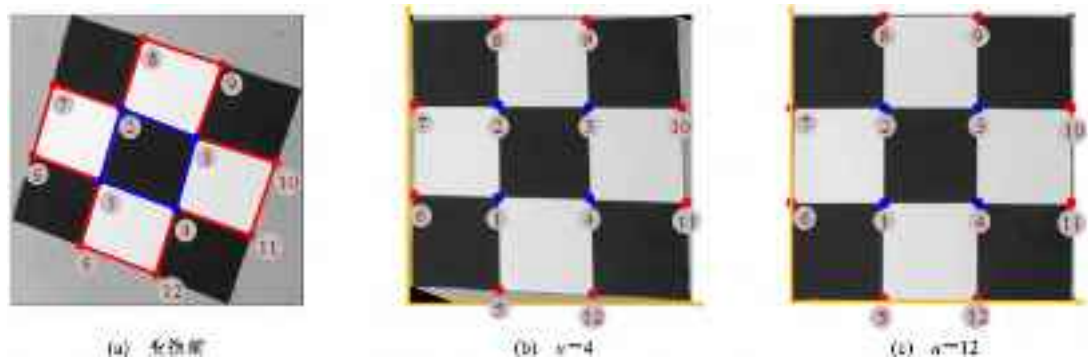


图 15 透视变换结果

Fig. 15 Results of Perspective Transformation

2.4 裂纹特征计算

钢桥裂纹形态较为细长,因此骨架线可直接视为裂纹中轴线,其长度即为裂纹长度。为准确提取裂纹骨架线,在利用 U-Net++ 分割出裂纹目标后,

还需先对其分割结果进行阈值分割从而提高裂纹分割精度。考虑到光照不均会弱化 U-Net++ 分割区域内的裂纹灰度特征,本文采用局域自适应阈值分割算法对其进行二值分割,分割结果如图 16 所示。通过八邻域骨架化算法对其进行骨架化,并利用本文第 2.3 节中的透视变换矩阵将骨架化结果进行重投影,如图 17 所示。最终,骨架线的像素长度为 1 215,并由像素尺度系数换算为裂纹的实际几何长度为:  $1\ 215 \times R = 98.4\text{ mm}$ 。

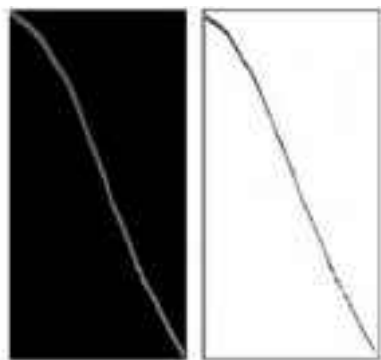


图 16 局域自适应阈值分割

Fig. 16 Local Adaptive Threshold Segmentation

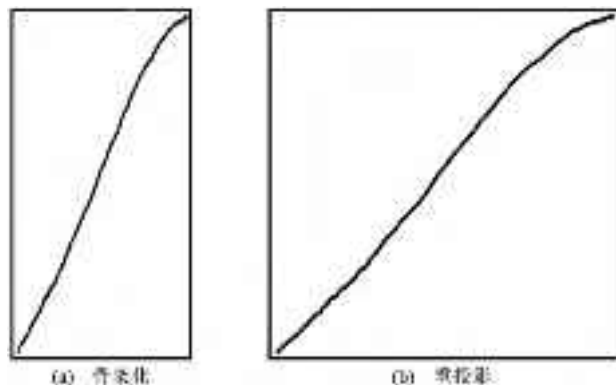


图 17 裂纹特征提取

Fig. 17 Crack Feature Extraction

## 2.5 裂纹特征识别结果

为对本文所提出的钢桥裂纹特征识别方法的鲁棒性与精度进行评估,将分别对不同图像采集条件下的钢桥裂纹特征进行计算。针对钢桥面板横隔板弧形开口处的疲劳裂纹,分别控制图像采集距离 ( $d=15\sim60\text{ cm}$ ,  $\theta$  取为  $0^\circ$ ) 与角度 ( $\theta=0\sim45^\circ$ ,  $d$  取为  $15\text{ cm}$ ) 得到多个裂纹原始图像,其中采集距离为图像采集设备至裂纹平面的距离,采集角度为图像采集设备视线与裂纹平面法向的夹角,如图 18(a)、(b)所示。利用本文方法对裂纹图像进行检测与分割、裂纹特征计算,该过程如图 18(c)~(f)所示,最终计算结果及误差如表 3 所示。由此可知: YoloV5

可准确检测出裂纹与标定块,且分类概率值较大,表明检测稳定性较好;利用 U-Net++ 对矩形裂纹候选框进行语义分割,并结合自适应阈值分割算法、八邻域骨架化算法可准确提取裂纹骨架线;针对标定块的角点检测及匹配,本文所提出的方法鲁棒性较好,但当图像采集距离较远、角度较大时,投影误差增大,且该误差对角度更为敏感;由于角点定位误差导致透视变换后的图像畸变分布存在不均匀性,裂纹的长度计算误差也存在离散性;本文所提出的方法在不同图像采集条件下的测量误差均在可接受的范围内,表明该方法具有较高的精度。

表 3 不同图像采集条件下裂纹特征测量结果

Table 3 Measurement Results of Crack Characteristics Under Different Image Acquisition Conditions

$d=15\sim60\text{ cm}$ , $\theta=0^\circ$ , 实测裂纹长度 92 mm	图像采集距离 $d/\text{cm}$	15	30	45	60
	投影误差 $\Delta M/\%$	1.4	2.1	1.7	2.7
$\theta=0\sim45^\circ$ , $d=15\text{ cm}$ , 实测裂纹长度 68 mm	长度测量结果/mm	93.1	90.2	93.9	95.0
	测量误差/ $\%$	1.2	-2.0	2.0	3.3
$\theta=0\sim45^\circ$ , $d=15\text{ cm}$ , 实测裂纹长度 68 mm	图像采集角度 $\theta/^\circ$	0	15	30	45
	投影误差 $\Delta M/\%$	1.6	4.2	5.7	6.0
$\theta=0\sim45^\circ$ , $d=15\text{ cm}$ , 实测裂纹长度 68 mm	长度测量结果/mm	70.1	72.0	72.9	72.6
	测量误差/ $\%$	3.1	5.9	7.2	6.8

## 3 结 语

(1)对于不同图像采集条件下的钢桥裂纹,本文提出的基于计算机视觉的钢桥面板裂纹识别方法可快速对图像中的裂纹进行检测、分割,并利用标定块对图像进行矫正与标定从而计算裂纹特征,最终测量误差最大为 7.2%,该方法具有较高的精度。

(2)由于钢桥裂纹常为狭长目标,其候选框大小及长宽比分布较为离散,通过统计其分布并优化 U-Net++ 训练过程中输入的像素尺寸,提高了模型收敛速度,网络损失由 0.121 降至 0.096,提升了裂纹分割网络的性能;训练后的 U-Net++ 对裂纹目标候选框中的裂纹像素进行语义分割并通过局域自适应阈值分割优化分割结果,再通过骨架化处理后直接从复杂和强干扰的环境中提取裂纹骨架线。

(3)通过 Harris 算法检测标定块角点,并根据角点的像素兴趣值分布对其进行分类匹配,结果表明该方法鲁棒性较好;利用所有角点坐标拟合透视变换矩阵的最小二乘解可克服 Harris 算法定位角点时的偶然误差,提高图像矫正与标定的精度;但当图像采集距离较远、角度较大时,投影误差增大,且该误差对角度更为敏感。

(4)本文所提出的方法需在目标区域附近粘贴

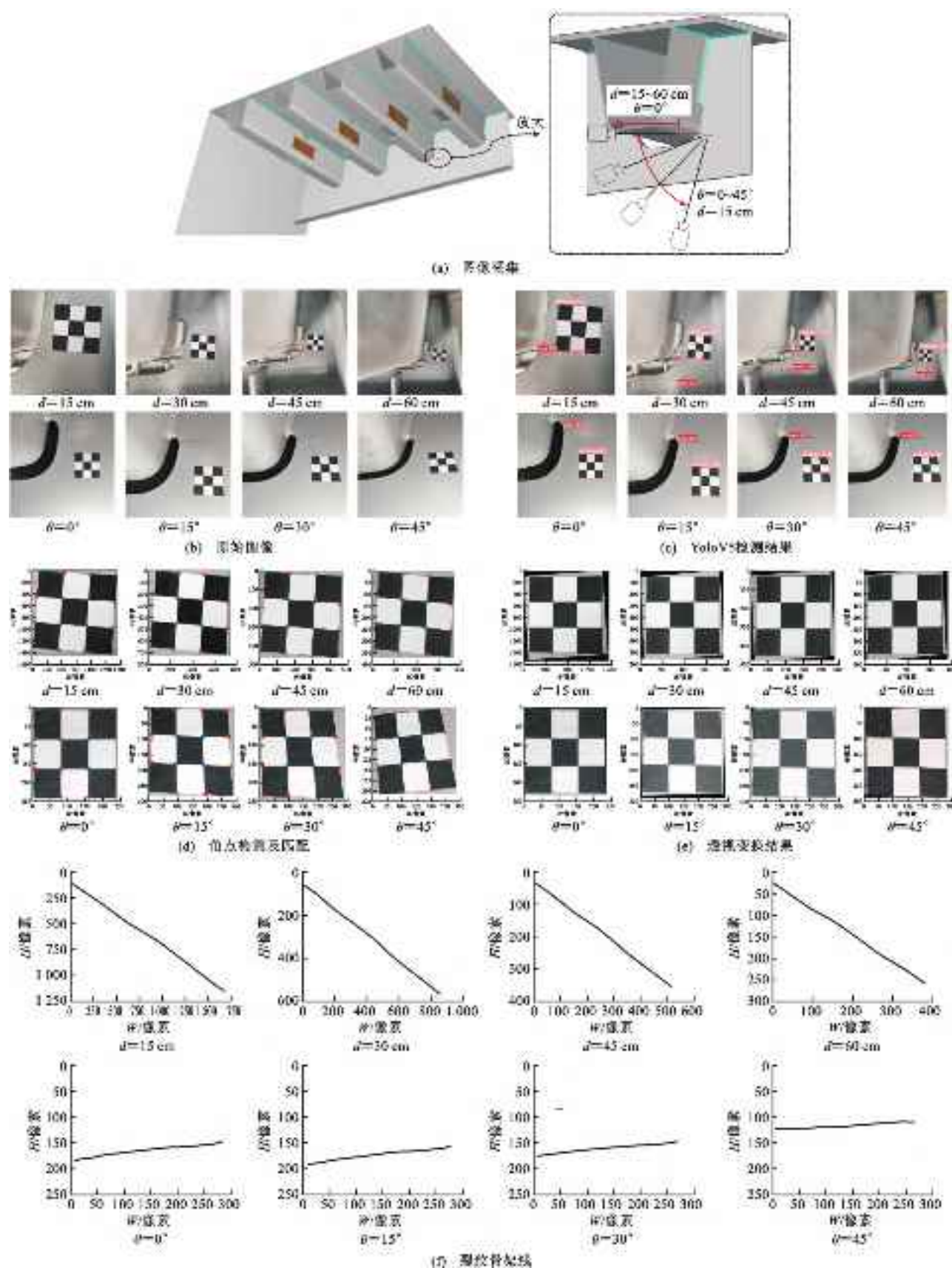


图 18 裂纹特征识别

Fig. 18 Crack Feature Recognition



标定块,后续可通过激光投影的方式直接将标定块投影至目标区域,以免去粘贴标定块的步骤;同时本文中的方法对裂纹的长度特征识别效果较好,但并未涉及到裂纹的宽度、深度,且裂纹跨多个平面时该方法将难以对其标定,因此如何提取跨面分布裂纹的三维特征将是下一步研究的重点和难点。

#### 参考文献:

#### References:

- [1] 张清华,卜一之,李 乔. 正交异性钢桥面板疲劳问题的研究进展[J]. 中国公路学报, 2017, 30(3): 14-30, 39.  
ZHANG Qing-hua, BU Yi-zhi, LI Qiao. Review on fatigue problems of orthotropic steel bridge deck [J]. China Journal of Highway and Transport, 2017, 30 (3): 14-30, 39.
- [2] 张清华,李 俊,郭亚文,等. 正交异性钢桥面板结构体系的疲劳破坏模式和抗力评估[J]. 土木工程学报, 2019, 52(1): 71-81.  
ZHANG Qing-hua, LI Jun, GUO Ya-wen, et al. Fatigue failure modes and resistance evaluation of orthotropic steel bridge deck structural system [J]. China Civil Engineering Journal, 2019, 52 (1): 71-81.
- [3] 贺拴海,赵祥模,马 建,等. 公路桥梁检测及评价技术综述[J]. 中国公路学报, 2017, 30(11): 63-80.  
HE Shuan-hai, ZHAO Xiang-mo, MA Jian, et al. Review of highway bridge inspection and condition assessment [J]. China Journal of Highway and Transport, 2017, 30 (11): 63-80.
- [4] 张清华,崔 闯,魏 川,等. 钢桥面板疲劳损伤智能监测与评估系统研究[J]. 中国公路学报, 2018, 31(11): 66-77, 112.  
ZHANG Qing-hua, CUI Chuang, WEI Chuan, et al. Research on intelligent monitoring and assessment system for fatigue damage of orthotropic steel deck structural system [J]. China Journal of Highway and Transport, 2018, 31 (11): 66-77, 112.
- [5] 鲍跃全,李 惠. 人工智能时代的土木工程[J]. 土木工程学报, 2019, 52(5): 1-11.  
BAO Yue-quan, LI Hui. Artificial intelligence for civil engineering [J]. China Civil Engineering Journal, 2019, 52 (5): 1-11.
- [6] 刘宇飞,樊健生,聂建国,等. 结构表面裂缝数字图像法识别研究综述与前景展望[J]. 土木工程学报, 2021, 54(6): 79-98.  
LIU Yu-fei, FAN Jian-sheng, NIE Jian-guo, et al. Review and prospect of digital-image-based crack detection of structure surface [J]. China Civil Engineering Journal, 2021, 54 (6): 79-98.
- [7] 沙爱民,童 峥,高 杰. 基于卷积神经网络的路表病害识别与测量[J]. 中国公路学报, 2018, 31(1): 1-10.  
SHA Ai-min, TONG Zheng, GAO Jie. Recognition and measurement of pavement disasters based on convolutional neural networks [J]. China Journal of Highway and Transport, 2018, 31 (1): 1-10.
- [8] JANG K, KIM N, AN Y, et al. Deep learning-based autonomous concrete crack evaluation through hybrid image scanning [J]. Structural Health Monitoring and International Journal, 2019 (1): 1722-1737.
- [9] DUNG C V, SEKIYA H, HIRANO S, et al. A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks [J]. Automation in Construction, 2019, 102: 217-229.
- [10] XU Y, BAO Y, CHEN J, et al. Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images [J]. Structural Health Monitoring, 2019, 18 (3): 653-674.
- [11] YE X W, JIN T, CHEN P Y. Structural crack detection using deep learning-based fully convolutional networks [J]. Advances in Structural Engineering, 2019, 22 (16): 3412-3419.
- [12] ZHANG A, WANG K, 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: 805-819.
- [13] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [J]. Advances in neural information processing systems, 2015, 2: 91-99.
- [14] HE K, GKIOXARI G, P DOLLÁR, et al. Mask R-CNN [C] // IEEE. Proceedings of the IEEE International Conference on Computer Vision. New York: IEEE, 2017: 2961-2969.
- [15] 鞠默然,罗海波,王仲博,等. 改进的 YOLO V3 算法及其在小目标检测中的应用[J]. 光学学报, 2019, 39(7): 253-260.  
JU Mo-ran, LUO Hai-bo, WANG Zhong-bo, et al. Improved YOLO V3 algorithm and its application in small target detection [J]. Acta Optica Sinica, 2019, 39 (7): 253-260.
- [16] MAHTO P, GARY P, SETH P, et al. Refining Yolo4 for vehicle detection [J]. International Journal of Advanced Research in Engineering and Technology, 2020, 11 (5): 409-419.
- [17] CHEN J, LIU Z, WANG H, et al. Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network [J]. IEEE Transactions on Instrumentation and Measurement, 2018, 67 (2): 257-269.
- [18] CHA Y J, CHOI W, SUH G, et al. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types [J]. Computer-aided Civil and Infrastructure Engineering, 2018, 33 (9): 731-747.
- [19] 晏班夫,徐观亚,栾 健,等. 基于 Faster R-CNN 与形态法的路面病害识别[J]. 中国公路学报, 2021, 34(9): 181-193.  
YAN Ban-fu, XU Guan-ya, LUAN Jian, et al. Pavement distress detection based on faster R-CNN and morphological operations [J]. China Journal of Highway and Transport,

- 2021, 34 (9): 181-193.
- [20] GUO P, MENG W, BAO Y. Automatic identification and quantification of dense microcracks in high-performance fiber-reinforced cementitious composites through deep learning-based computer vision [J]. Cement and Concrete Research, 2021 (1): 106532.
- [21] 刘宇飞,樊健生,孔思宇,等. 多视角几何三维重建法识别工程结构缺损与变形[J]. 工程力学, 2020, 37(9): 103-111.  
LIU Yu-fei, FAN Jian-sheng, KONG Si-yu, et al. Detection of structural defect and deformation based on multi-view geometric three-dimensional reconstruction method [J]. Engineering Mechanics, 2020, 37 (9): 103-111.
- [22] KALFARISI R, WU Z Y, SOH K. Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization [J]. Journal of Computing in Civil Engineering, 2020, 34 (3): 04020010.
- [23] WANG D, DONG Y, PAN Y, et al. Machine vision-based monitoring methodology for the fatigue cracks in U-rib-to-deck weld seams [J]. IEEE Access, 2020, 8: 94204-19.
- [24] RONNEBERGER O, FISCHER P, BROX T. U-Net: Convolutional networks for biomedical image segmentation [C] // Springer. International Conference on Medical Image Computing and Computer-assisted Intervention. Berlin: Springer, 2015: 234-241.
- [25] ZHOU Z, SIDDIQUEE M, TAJBAKHS N, et al. UNet++: Redesigning skip connections to exploit multiscale features in image segmentation [J]. IEEE Transactions on Medical Imaging, 2020, 39 (6): 1856-1867.
- [26] 叶肖伟,董传智. 基于计算机视觉的结构位移监测综述[J]. 中国公路学报, 2019, 32(11): 21-39.  
YE Xiao-wei, DONG Chuan-zhi. Review of computer vision-based structural displacement monitoring [J]. China Journal of Highway and Transport, 2019, 32 (11): 21-39.
- [27] SHORTEN C, KHOSHGOFTAAR T M. A survey on image data augmentation for deep learning [J]. Journal of Big Data, 2019, 6 (1): 1-48.