

# 基于图像转换的道路图像裂缝检测<sup>①</sup>



阎志文

(长安大学 信息工程学院, 西安 710064)  
通信作者: 阎志文, E-mail: 429169028@qq.com

**摘要:** 深度学习是目前路面图像裂缝检测的主流方法, 但是需要大量人工标注的真值图进行训练, 而现实中获取人工标注的真值图既费时又费力, 本文提出一种基于改进的生成对抗网络的路面图像裂缝检测方法, 将路面图像裂缝检测问题视为一类基于图像跨域转换的异常检测问题, 采用定点生成对抗网络将裂缝图像无监督自动转换为与之对应的无裂缝图像, 进而将原图像与生成图像进行差分, 差分图中的显著目标对应裂缝检测结果. 在公开数据集 CrackIT 上的测试结果表明, 本文方法在不依赖于人工标注的真值图条件下能够实现裂缝的精准检测, 本文方法在准确率、召回率、F1 分数上取得了与有监督深度学习方法相当的性能.

**关键词:** 道路裂缝检测; 定点生成对抗网络; 图像转换; 无监督

引用格式: 阎志文. 基于图像转换的道路图像裂缝检测. 计算机系统应用, 2023, 32(4): 241-247. <http://www.c-s-a.org.cn/1003-3254/9031.html>

## Road Image Crack Detection Based on Image Transformation

YAN Zhi-Wen

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

**Abstract:** Deep learning is currently a mainstream method for crack detection in pavement images, but it requires a large number of manually-annotated ground-truth images for training. However, in reality, it is time-consuming and laborious to obtain manually-annotated ground-truth images. This study proposes a method for crack detection in pavement images based on an improved generative adversarial network. The study regards crack detection in pavement images as a kind of anomaly detection problem based on image cross-domain transformation and uses a fixed-point generative adversarial network to automatically convert the crack image into a one-to-one corresponding crack-free image with supervision. Then, the study differentiates the original image and the generated image, and the salient objects in the difference image correspond to the crack detection results. The test results on the public dataset CrackIT show that the method in this study can achieve accurate crack detection without relying on the manually-annotated ground-truth images. In addition, the method achieves comparable performance to supervised deep learning methods in terms of *precision*, *recall*, and *F1-score*.

**Key words:** road crack detection; fixed-point generative adversarial network (fGAN); image transformation; unsupervision

## 1 引言

路面图像裂缝检测对于路面的科学管理和养护具有重要意义<sup>[1]</sup>. 近年来大量的研究方法被报道, 大致可

分为 3 类. 第 1 类是常规模式识别方法, 例如通过阈值分割<sup>[2]</sup>、数学形态学<sup>[3]</sup>、边缘检测<sup>[4]</sup>或小波变换实现裂缝和背景的分割<sup>[5]</sup>; 第 2 类是基于机器学习的方法,

① 收稿时间: 2022-08-19; 修改时间: 2022-09-22; 采用时间: 2022-10-21; csa 在线出版时间: 2022-12-23  
CNKI 网络首发时间: 2022-12-27

包括基于随机结构森林<sup>[6]</sup>和基于 AdaBoost 分类器进行区域分类的方法<sup>[7]</sup>等;第3类为基于深度学习的方法,以基于卷积神经网络(CNN)进行路面图像区域分类、基于 U-Net<sup>[8]</sup>或全卷积神经网络(FCN)<sup>[9]</sup>进行路面图像逐像素分类等方法为典型代表.尽管路面图像裂缝检测问题已经得到广泛深入研究,并取得一定成果,但现有方法至少在以下两方面还存在不足:(1)噪声鲁棒性有待于提高,多数方法易受环境噪声的影响,导致不能完全检测裂缝或者造成假阳性裂缝检测结果;(2)需要大量人工标注的真值图用于训练,耗费大量人力和物力.

生成对抗网络(GAN)<sup>[10]</sup>及其变种作为一类用于无监督自动生成高逼真度样本的深度学习方法在图像异常检测领域得到越来越多的关注.Zhang 等人<sup>[11]</sup>将 GAN 引入道路图像裂缝检测,其中生成器是一个非对称的 U-Net 网络,用于提取图像特征,输出裂缝标签预测,然后将输出的裂缝检测结果与真值图输入判别器进行判别,且分别引入了对抗损失鉴别生成的图像真假和均方根损失函数提高检测结果的准确率.Zhang 等人<sup>[12]</sup>采用 cycle-GAN<sup>[13]</sup>求解道路图像裂缝检测问题.使用了两个生成器和两个鉴别器,一个生成器  $G$  将裂缝图像转换为类真值图像,鉴别器  $D_X$  鉴别生成的图像是否为真值图像.另一个生成器  $F$  将真值图像转换为裂缝图像,鉴别器  $D_Y$  鉴别生成的图像是否为裂缝图像.通过循环一致性损失保留输入图像的特性,最终的目的是将道路裂缝图像转换为“对应”的类真值图像,从而实现裂缝检测.作为 GAN 的一个升级变种,定点生成对抗网络(fixed-point GAN, fGAN)<sup>[14]</sup>是最初用于脑部 CT 图像的病变检测,脑部 CT 的患病图像与裂缝图像有相似的特点,即病变区域与周围背景存在灰度差,病变区域从周围背景像素中突显出来.fGAN 不仅能够准确检出脑部 CT 图像中的病变区域,而且具有较好的噪声鲁棒性.

受此启发,本文提出一个基于 fGAN 的路面图像裂缝检测方法,基于无监督自动生成高逼真度样本深度学习原理,采用 fGAN 将路面图像进行跨域转换,将输入的路面图像转换为无裂缝路面图像,进而基于原图像和生成图像的差值图,实现对裂缝的像素级检测.图像转换过程中,背景区域被尽可能原样保留的同时,裂缝区域得到强化.实验结果表明,所提方法不仅不依赖人工标注的真值图,而且具有较好的噪声鲁棒性,可

以实现对面路面图像裂缝的像素级精准检测.

## 2 模型和方法

整体框架是首先通过 fGAN 将裂缝图像跨域转换为与之一一对应的无裂缝图像,然后将生成的图像和原图差分,得到差分图,最后通过后处理对差分图去除噪点,得到最终检测结果.本节首先详细给出基于 fGAN 的裂缝检测模型,通过生成器-鉴别器网络结构实现路面图像差分图输出.然后阐述无监督裂缝检测算法.

### 2.1 定点生成对抗网络

图 1 描述了定点生成对抗网络生成图像过程:输入无裂缝图像数据集和裂缝图像数据集,通过生成器将裂缝图像向“对应”的无裂缝图像转换;将生成图和原图进行差分,得到差分图,差分图显示裂缝位置和区域.

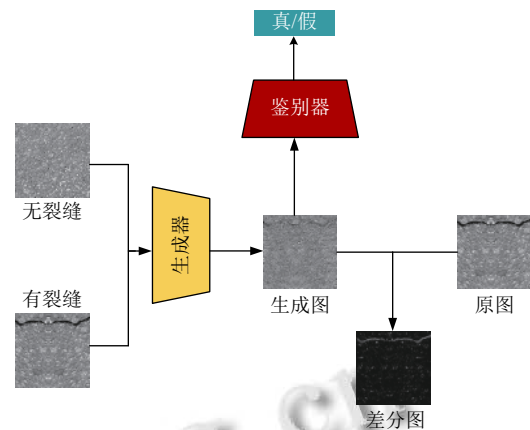


图 1 定点生成对抗网络生成图像过程图

#### 2.1.1 生成器

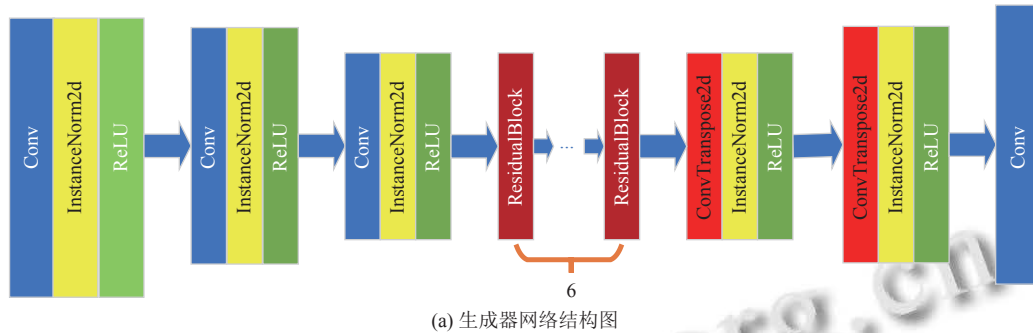
生成器是 fGAN 的核心,定点生成对抗网络有 4 个卷积层除网络结构中最后一个卷积层以外每个卷积层后都跟随一个 InstandNorm2d 层,6 个残差块以及 2 个反卷积层.卷积层用于提取图像特征,Instand-Norm2d 层对卷积层的输出进行归一化处理,提高数据的有效性和稳定数据的分布.残差块也用于特征提取,并且它的“短路”连接保证特征提取的能力不会因为神经网络层数增加而产生梯度爆炸,导致特征提取能力减弱.反卷积层能将卷积层和残差块提取的特征信息恢复成图像语义信息.生成器网络结构如图 2(a) 所示.

#### 2.1.2 鉴别器

鉴别器有 8 个卷积层除计算对抗损失和域分类损失的卷积以外,每个卷积层后都有 LeakyReLU 函数,卷积层用于提取特征以判断生成图像的真假以及生成

器生成图像是否为裂缝图像或无裂缝图像, LeakyReLU 函数能够线性变换卷积层的输出, 并且不会丢失小于

0 的信息, 使模型的表达能力更强. 鉴别器网络结构如图 2(b) 所示.



(a) 生成器网络结构图



(b) 鉴别器网络结构图

图 2 定点生成对抗网络结构

### 2.1.3 损失函数

fGAN 将裂缝图像转换为“对应”的无裂缝图像需要解决 3 个问题: 第一, 确保生成器的输出是无裂缝图像, 本文引入生成对抗损失和域分类损失来确保生成图像为无裂缝图像. 第二, 确保生成器能够将裂缝转换为无裂缝时不是随机转换而是尽可能保留原图像的信息, 因此, 引入循环一致性损失, 使生成的图像能够保留原图像的信息. 第三, 非裂缝区域改变更少: 引入循环一致性损失能够保留原图中图像的特性, 但是会改变背景区域. 因此加入有条件的身份损失函数, 保证非裂缝区域改变更少.

下面详细描述各个损失函数并且通过数学公式来阐述它们的作用. 在本文方法中, 生成器学习跨域转换即将裂缝转换为无裂缝和同域转换即将无裂缝图像转换为无裂缝图像. 为了确保生成图像更加接近真实图像, 引入对抗性损失, 损失函数公式如下:

$$l_{adv} = \sum_{c \in \{c_x, c_y\}} E_{x,c} [\log(1 - D_{real/fake}(G(x, c)))] + E_x [\log D_{real/fake}(x)] \quad (1)$$

其中, 生成器  $G$  将图像  $x$  转换为目标域图像  $G(x, c)$ , 鉴别器  $D$  鉴别生成图像  $G(x, c)$  和真实图像  $x$  是否为期望的目标域图像. 对抗性损失确保生成的图像是期望的道路图像, 但不能保证域的正确性即无法鉴别生成图像是裂缝图像还是无裂缝图像. 因此, 鉴别器引入域分类损失, 这保证当期望生成无裂缝图像时生成器生成无裂缝图像或者当期望生成裂缝图像时生成器生成裂缝图像.

鉴别器的域分类损失:

$$l'_{domain} = E_{x,c_x} [-\log D_{domain}(c_x|x)] \quad (2)$$

当图像为  $x$ , 鉴别器  $D$  鉴别域不是  $c_x$  时, 损失  $l'_{domain}$  变大, 因此通过最小化  $l'_{domain}$  鉴别图像的域.

生成器的域分类损失, 用来考虑相同域和跨域转

换, 确保生成的图像在两种情况下都来自正确的域.

$$l_{\text{domain}}^f = \sum_{c \in \{c_x, c_y\}} E_{x, c_x} [-\log D_{\text{domain}}(c|G(x, c))] \quad (3)$$

生成器的域分类损失原理同鉴别器的域分类损失, 但是需要考虑图像域为 $c_x$ 和 $c_y$ 两种情况.

循环一致性损失有助于生成器在进行转换学习时可以保留输入图像的特性, 从而使生成的图像可以转换回源域. 循环一致性损失能够应对跨域和同域转换, 公式如下:

$$l_{\text{cyc}} = E_{x, c_x, c_y} [\|G(G(x, c_y), c_x) - x\|_1] + E_{x, c_x} [\|G(G(x, c_x), c_x) - x\|_1] \quad (4)$$

对于来自域 $c_x$ 的每个图像 $x$ , 在同域转换时应该满足:  $x \rightarrow G(x, c_x) \rightarrow G(G(x, c_x), c_x) \approx x$  保证转换后的图像能够转换为原图像. 同样, 在跨域转换时应该保证转换后的图像 $G(G(x, c_y), c_x)$ 能够转换为 $x$ , 因此使用 $L_1$ 范数约束循环转换使循环一致性损失最小.

在训练期间, 当生成器改变与目标域无关的输入方面时, 需要惩罚生成器, 因此引入有条件的身份损失, 强制生成器在将图像转换到源域时改变图像更少. 这也有助于生成器将输入图像转换到目标域时更少的改变图像. 公式如下:

$$l_{\text{id}} = \begin{cases} 0, & c = c_y \\ E_{x, c} [\|G(x, c) - x\|_1], & c = c_x \end{cases} \quad (5)$$

当进行同域转换时, 转换后图像 $G(x, c_x)$ 与原图 $x$ 进行范数 $L_1$ 计算, 当改变非目标区域时,  $l_{\text{id}}$ 变大, 因此最小化 $l_{\text{id}}$ 使生成器更少改变背景区域.

因此 fGAN 的目标函数如下:

$$G^*, D^* = \arg \min_G \max_D l(G, D) \quad (6)$$

鉴别器区分真图像或假图像, 并对输入图像的域进行分类. 生成器不仅通过转换学习执行跨域转换, 而且通过定点转换学习执行相同的域转换, 这对于将裂缝图像转换为“对应”的无裂缝图像至关重要.

## 2.2 检测算法

训练分为两部分: 训练鉴别器和训练生成器, 鉴别器训练为两部分: (1) 训练鉴别生成图像的能力. (2) 训练鉴别真实图像的能力. 生成器的训练分为两部分: (1) 学习跨域转换, 训练生成器将裂缝图像转换为无裂缝图像. (2) 学习相同的域转换, 保证生成器对非裂缝区域更少改变, 这对于将裂缝图像转换为“对应”的无

裂缝图像至关重要. 当训练其中一个时, 另一个的参数固定, 交替训练. 训练过程如算法 1 所示.

算法1. 定点生成对抗网络的训练过程

---

输入: 训练集 $data=\{x_n, y_n\}_{n=1}^N$ , 输入图像的域及标签 $c_{\text{org}}, label_{\text{org}}$ , 生成图像的域及标签 $c_{\text{trg}}, label_{\text{trg}}$   
 输出: 生成器参数 $W_G$ , 鉴别器参数 $W_D$

---

```

While  $W_G$  had not converged do
for  $n=1$  to  $n_{\text{critic}}$  do
Step 1.  $out\_src, out\_cls = D(x_{\text{real}})$ 
Step 2. 计算真实对抗损失 $d_{\text{loss\_real}}$ , 计算域分类损失 $d_{\text{loss\_cls}}$ 
Step 3.  $x_{\text{fake}} = G(x_{\text{real}}, c_{\text{trg}})$ 
Step 5.  $out\_src, out\_cls = D(x_{\text{fake}})$ 
Step 6. 计算生成图像的对抗损失 $d_{\text{loss\_fake}}$ , 计算域分类损失 $d_{\text{loss\_gp}}$ 
Step 7.  $d_{\text{loss}} = d_{\text{loss\_real}} + d_{\text{loss\_fake}} + d_{\text{loss\_cls}} + d_{\text{loss\_gp}}$ 
Step 8.  $x_{\text{fake}} = G(x_{\text{real}}, c_{\text{trg}})$ 
Step 9.  $out\_src, out\_cls = D(x_{\text{fake}})$ 
Step 10. 计算生成图像的对抗损失 $g_{\text{loss\_fake}}$ , 计算域分类损失 $g_{\text{loss\_cls}}$ 
Step 11.  $x_{\text{fake\_id}} = G(x_{\text{real}}, c_{\text{org}})$ 
Step 12.  $out\_src\_id, out\_cls\_id = D(x_{\text{fake\_id}})$ 
Step 13. 计算生成图像的对抗损失 $g_{\text{loss\_fake\_id}}$ , 计算域分类损失 $g_{\text{loss\_cls\_id}}$ , 计算循环一致性损失 $g_{\text{loss\_id}}$ 
Step 14.  $x_{\text{reconst}} = G(x_{\text{fake}}, c_{\text{org}})$ 
Step 15. 计算循环一致性损失 $g_{\text{loss\_rec}}$ 
Step 16.  $x_{\text{reconst\_id}} = G(x_{\text{fake\_id}}, c_{\text{org}})$ 
Step 17. 计算有条件的身份损失 $g_{\text{loss\_rec\_id}}$ 
Step 18. 固定 $G$ , 更新 $D$ 的参数
end for
Step 19. 固定 $D$ , 更新 $G$ 的参数
end while
    
```

---

## 3 实验分析

### 3.1 实验环境及模型参数

本文实验环境为 Ubuntu 18.04, Inter CE5-2650 @2.20 GHz 42 GB RAM, TATAN RTX 1080 16 GB, Python 3.6, PyTorch 深度学习框架. fGAN 的主要参数如表 1 所示.

表 1 fGAN 的网络参数

参数	$G$	$D$
Epochs	300 000	300 000
Batch size	16	16
Optimizer	Adam	Adam
Learning rate	0.000 1	0.000 1

### 3.2 实验数据

本文方法在公共数据集 CrackIT 进行了测试. CrackIT 数据集: 数据集共有 48 张分辨率 1536×2048 的图片, 将图像裁剪为分辨率 256×256 的图像, 将裁剪

后的图像分为两类: 裂缝图像和无裂缝图像, 用于训练定点生成对抗网络. 定点生成对抗网络训练集包含两类图像: 裂缝图像和无裂缝图像, 其中裂缝数图像 103 张, 无裂缝图像 232 张.

### 3.3 评价指标

本文采取道路裂缝检测中的 3 个指标: 精确率、召回率和  $F1$  分数. 下面给出相关指标的定义.

(1)  $TP$  (true positive): 正类被预测成为正类的数量.

(2)  $TN$  (true negative): 负类被预测成为负类的数量.

(3)  $FP$  (false positive): 负类但是被预测成为正类的数量.

(4)  $FN$  (false negative): 正类但是被预测成为负类的数量.

精确率 (*precision*) 是  $TP$  与预测的正类的比例, 计算公式如式 (7) 所示:

$$precision = \frac{TP}{TP + FP} \quad (7)$$

召回率 (*recall*) 是  $TP$  与真实的正类数量的比例, 公式如下:

$$recall = \frac{TP}{TP + FN} \quad (8)$$

$F1$  分数 ( $F1$ -score) 是 *precision* 和 *recall* 两个指标的调和数, 公式如下:

$$F1\text{-score} = \frac{2 \times precision \times recall}{precision + recall} \quad (9)$$

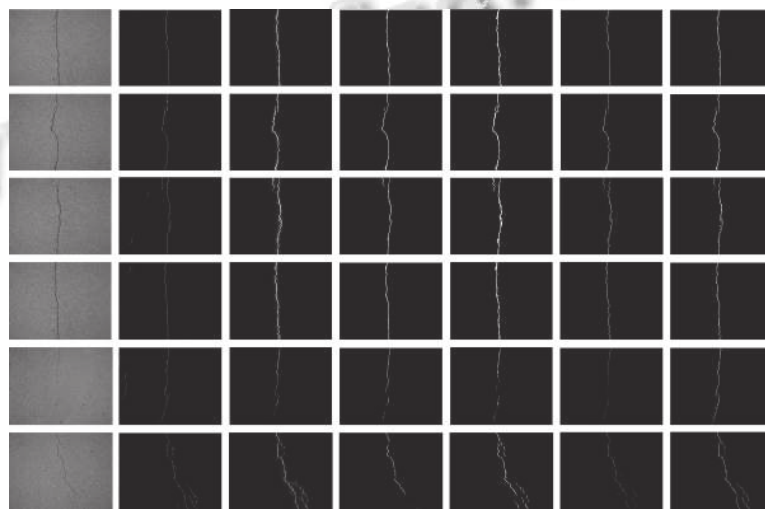
### 3.4 结果分析

本文选取有监督的深度学习方法进行比较, Deep-

Crack<sup>[15]</sup> 是专门用于检测道路裂缝的网络模型, 是道路裂缝检测领域的代表性的模型, 所以选择 DeepCrack 与本文方法进行比较, U-Net 网络是最初用于道路裂缝检测的网络模型, 选择 U-Net 方法进行比较, 并且本文还选取 U-Net 的改进网络 DAU-Net<sup>[16]</sup> 与本文方法进行比较, 进一步与先进的有监督深度学习方法进行比较. SegNet<sup>[17]</sup> 是语义分割的经典模型, 也被用于道路裂缝检测, 因此选取 SegNet 与本文方法进行比较. 上述方法都是使用已经训练好的模型进行测试, 结果如表 2 所示, 其中加粗字体表示最高值. 在精确率上, 本文方法取得了最好的成绩. 本文方法与对比实验方法的结果对比, 如图 3 所示. 从图 3 可以看出本文方法与对比方法都能检测出裂缝的主体部分, 但是对于细小裂缝检测时, DeepCrack 会将类似裂缝的部分检测为裂缝, 造成误检, U-Net 会受到背景噪声的影响, 将背景噪声误检为裂缝, DAU-Net 不能完整地检测出细小裂缝, SegNet 会将裂缝周围的背景误检成裂缝, 本文方法不仅可以检测出细小裂缝, 且不会对类似细小裂缝区域造成误检. 图 4 展示 fGAN 在训练过程中损失函数的情况, 图 4 表明本文提出的模型在训练阶段损失较小且收敛速度较快.

表 2 CrackIT 数据集测试结果

方法	<i>precision</i>	<i>recall</i>	$F1$ -score
DeepCrack	0.31	<b>0.64</b>	0.40
U-Net	0.58	0.57	0.57
DAU-Net	0.66	0.62	<b>0.64</b>
SegNet	0.64	0.53	0.57
本文	<b>0.75</b>	0.45	0.56



(a) 原图 (b) DeepCrack (c) U-Net (d) DAU-Net (e) SegNet (f) 本文方法 (g) 真值图

图 3 CrackIT 数据集测试结果对比

### 4 结论与展望

本文提供了一种改进的道路图像裂缝检测框架, 采用定点生成对抗网络, 保证了在图像转换时对非裂缝区域的改变尽可能小, 从而能得到更精准的裂缝差分图. 本方法的优点是不依赖人工标注的真值图, 并且在 CrackIT

数据集上测试, 本方法与有监督的深度学习裂缝检测方法取得了相当的性能. 在今后工作中, 将从两个方面继续探索基于定点生成对抗网络的道路裂缝检测方法: (1) 改进定点生成对抗网络, 提高裂缝检测准确性. (2) 将本文方法应用到其他领域, 扩大应用范围.

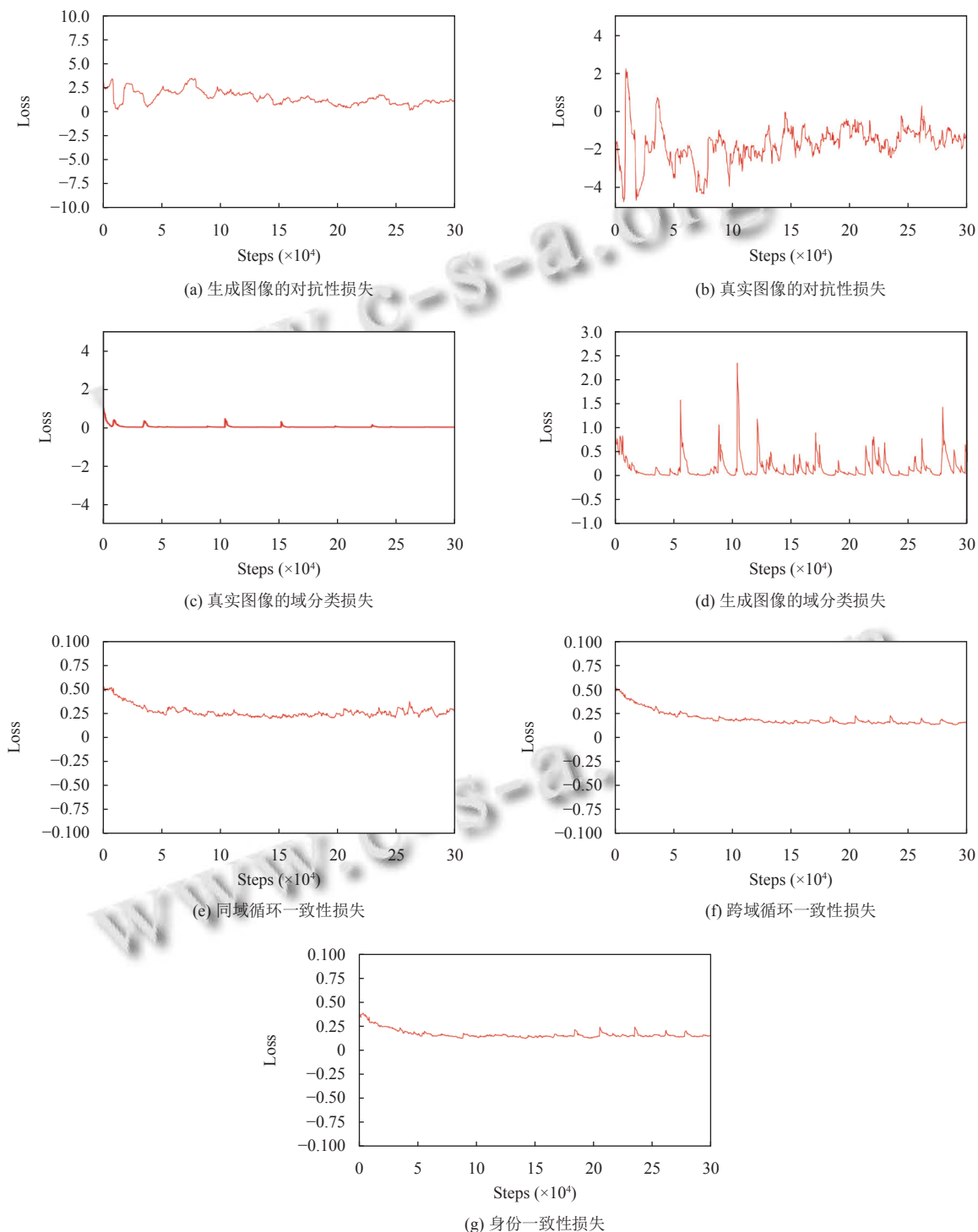


图4 fGAN 损失函数曲线图

## 参考文献

- 1 叶肖伟, 董传智. 基于计算机视觉的结构位移监测综述. 中国公路学报, 2019, 32(11): 21–39. [doi: [10.19721/j.cnki.1001-7372.2019.11.002](https://doi.org/10.19721/j.cnki.1001-7372.2019.11.002)]
- 2 Oliveira H, Correia PL. Automatic road crack segmentation using entropy and image dynamic thresholding. Proceedings of the 2009 17th European Signal Processing Conference. Glasgow: IEEE, 2009. 622–626.
- 3 Tanaka N, Uematsu K. A crack detection method in road surface images using morphology. Proceedings of IAPR Workshop on Machine Vision Applications. Chiba: MVA, 1998. 154–157.
- 4 Zhao HL, Qin GF, Wang XJ. Improvement of canny algorithm based on pavement edge detection. Proceedings of the 2010 3rd International Congress on Image and Signal Processing. Yantai: IEEE, 2010. 964–967.
- 5 Tang JS, Gu YL. Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis. Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics. Manchester: IEEE, 2013. 3026–3030.
- 6 Shi Y, Cui LM, Qi ZQ, *et al.* Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(12): 3434–3445. [doi: [10.1109/TITS.2016.2552248](https://doi.org/10.1109/TITS.2016.2552248)]
- 7 Ni ZP, Tang PH, Xi YY. A new method to pavement cracking detection based on the biological inspired model. Proceedings of the 2012 International Conference on Computer Science and Information Processing (CSIP). Xi'an: IEEE, 2012. 755–758.
- 8 Liu ZQ, Cao YW, Wang YZ, *et al.* Computer vision-based concrete crack detection using U-net fully convolutional networks. Automation in Construction, 2019, 104: 129–139. [doi: [10.1016/j.autcon.2019.04.005](https://doi.org/10.1016/j.autcon.2019.04.005)]
- 9 Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 640–651. [doi: [10.1109/TPAMI.2016.2572683](https://doi.org/10.1109/TPAMI.2016.2572683)]
- 10 Goodfellow IJ, Pouget-Abadie J, Mirza M, *et al.* Generative adversarial networks. arXiv:1406.2661, 2014.
- 11 Zhang KG, Zhang YT, Cheng HD. CrackGAN: Pavement crack detection using partially accurate ground truths based on generative adversarial learning. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(2): 1306–1319. [doi: [10.1109/TITS.2020.2990703](https://doi.org/10.1109/TITS.2020.2990703)]
- 12 Zhang KG, Zhang YT, Cheng HD. Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. Journal of Computing in Civil Engineering, 2020, 34(3): 04020004. [doi: [10.1061/\(ASCE\)CP.1943-5487.0000883](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000883)]
- 13 Zhu YJ, Park T, Isola P, *et al.* Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Venice: IEEE, 2017. 2242–2251.
- 14 Siddiquee MMR, Zhou ZW, Tajbakhsh N, *et al.* Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE, 2019. 191–200.
- 15 Liu YH, Yao J, Lu XH, *et al.* DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing, 2019, 338: 139–153. [doi: [10.1016/j.neucom.2019.01.036](https://doi.org/10.1016/j.neucom.2019.01.036)]
- 16 Polovnikov V, Alekseev D, Vinogradov I, *et al.* DAUNet: Deep augmented neural network for pavement crack segmentation. IEEE Access, 2021, 9: 125714–125723. [doi: [10.1109/ACCESS.2021.3111223](https://doi.org/10.1109/ACCESS.2021.3111223)]
- 17 Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481–2495. [doi: [10.1109/TPAMI.2016.2644615](https://doi.org/10.1109/TPAMI.2016.2644615)]

(校对责编: 孙君艳)