



(12) 发明专利申请

(10) 申请公布号 CN 117079752 A
(43) 申请公布日 2023. 11. 17

(21) 申请号 202311064280.8
(22) 申请日 2023.08.23
(71) 申请人 北京航空航天大学
地址 100191 北京市海淀区学院路37号
(72) 发明人 高瀚君 陶鑫瑞 吴琼 陈晓满
袁松梅

(51) Int. Cl.
G16C 60/00 (2019.01)
G06F 30/27 (2020.01)
G06N 3/045 (2023.01)
G06N 3/0475 (2023.01)
G06N 3/0464 (2023.01)
G06N 3/094 (2023.01)
G06V 10/26 (2022.01)
G06F 113/10 (2020.01)

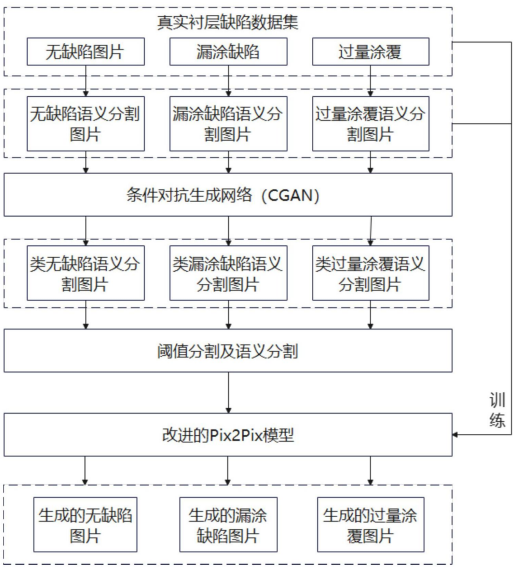
权利要求书4页 说明书8页 附图5页

(54) 发明名称

一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法

(57) 摘要

本发明涉及深度学习技术领域,且公开了一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法,包括数据预处理、条件对抗生成网络和Pix2Pix的生成器设计、判别器设计以及生成器与判别器的对抗能力控制设计;所述数据预处理包括原始缺陷图片的筛选以及语义分割图片的标注,并将制作的数据集图片及标签转化成深度学习模型所能处理的张量类型,并将图片归一化到-1~1之间;所述生成器设计用于将给定的语义分割图片转换为缺陷图片,使用U-Net结构,并通过引入注意力机制来提高模型的性能;所述判别器设计用于对生成的缺陷样本图片进行评估;所述生成器与判别器的对抗能力控制设计用于平衡生成器和判别器的能力,使得生成器能更好地生成包含缺陷特征的图片。通过本发明,解决了缺陷识别任务中样本数量少及不平衡的问题,使得缺陷分类模型具有更强的泛化能力。



1.一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法,其特征在于,包括以下步骤:

S1、基于条件生成对抗网络获得语义分割样本:

S11、数据预处理:

S111、语义分割图片制作:创建"0"、"1"、"2"三个文件夹,分别代表无缺陷图片、漏涂缺陷、过量涂覆三种情况;人工筛选获取到的原始图片,确定其缺陷类型后放置在对应的文件夹下;本发明使用的无缺陷图片共450张、漏涂缺陷图片486张、过量涂覆443张;使用Labelme工具对原始图片中的缺陷进行标注;使用Labelme标注后的文件为json格式的文件。使用Python中的json库对json文件进行解析,并使用OpenCV库绘制语义分割图片;

S112、标签转化为独热编码:将标签转换为独热编码可以提供一种统一的表示方式,使生成器和判别器网络能够更好地协同工作,提高生成对抗网络的性能;标签转换为独热编号后,可以更方便参与卷积或反卷积;

S113、图片的张量化及归一化:使用PyTorch里的ToTensor以及Normalize方法将每一张输入的灰度图片转换到-1~1之间;

S114、自定义数据集类:自定义一个继承Dataset类的数据集类,用于处理自己的数据集,在类中,定义了获取文件数量、获取索引值、获取标签值、获取图片路径值等成员,用于对图片进行灵活处理;

S115、数据集制作:在S114所述的自定义数据集类基础上加载语义分割图片,设置批量值BATCH_SIZE大小为16,开启随机读取和重复读取;

S12、生成器设计:

S121、创建随机噪声:定义一个长度为500的噪声,该噪声遵循正态分布;该噪声是生成器的输入,噪声数据经过一系列的全连接层以及反卷积层操作后,变换为 $1 \times 256 \times 256$ 的灰度图,即形成目标图片;

S122、改变标签数据形状:标签经过独热编码之后,将成为长度为4的张量,经过数层全连接层神经网络之后,将该向量的形状转化为(128,4,4);

S123、改变输入噪声形状:输入噪声的原始形状为长度500且服从正态分布的随机张量,将该张量经过数层全连接层神经网络之后,将该向量的形状转化为(128,4,4);

S124、合并标签与噪声:将改变形状后的输入噪声与标签进行通道维度上的合并,即变成(256,4,4)形状的数据;

S125、反卷积与批标准化:对合并后的数据进行反卷积操作,为了将其形成最后的目标图片大小,并且获取到真正有用的缺陷特征信息,,每一次反卷积都是使用leaky_relu函数进行激活,最终形成(1,256,256)形状的数据,即一张灰度图;

S13、判别器设计

S131、改变标签数据形状:判别器接受到的数据有两个,一个数生成的图片,另一个数独热编码后的标签,为了能生成器对两者进行判断,需要先将标签通过全连接神经网络转换为 $1 \times 256 \times 256$ 大小的形状,再将其展平为(1,256,256)形状;

S132、合并标签与生成的图片:将形状为(1,256,256)的标签与形状为(1,256,256);

S133、设计卷积层:设计6个卷积层,依次将合并后的形状通道数增加,宽和高减小,每一次反卷积都是使用leaky_relu函数进行激活;最终形成形状为(256,2,2)的张量;

S134、批标准化:对形状为(256,2,2)的张量进行批标准化处理;

S135、全连接层:使用全连接层,将(256,2,2)连接到单位1的张量上,再使用sigmoid进行激活;

S14、学习速率的设置:将生成器的学习速率设置为0.001,将判别器的学习速率设置为0.0001;这样的好处是削弱判别器的能力,使得生成器在早期阶段具有抗衡判别器的能力;

S15、配置模型训练参数

S151、设备选择:为了使得深度学习的代码具有更好的兼容性,先对硬件设备的GPU可用性进行判断,如果GPU不可使用,则设备选择为CPU;

S152、损失函数:因为判别器最后的输出结果是0~1的概率值,属于二分类问题,选择BCELoss作为训练的损失函数;

S153、优化器选择:生成器和判别器的优化器均选择Adam,除学习速率如S41中所述,Adam的其他参数均为默认值;

S16、模型的保存:一个训练完成的神经网络,包含结构和权重两个部分;当训练完成之后,所有的信息都保存在了model中,但此时只在内存里;通过save方法,将网络的框架以及权重全都保存到“CGAN_model.pth”文件中;在需要调用时,只需要加载pth文件即可;

S2、基于改进的Pix2Pix生成缺陷图片:

S21、数据预处理

S211、阈值分割:通过所述S1中的方法,可以获得训练过程中得到的图片,通过手动采集质量较好、具有缺陷特征的图片,放置在“Pix2PixData”数据集中,先使用基于OpenCV的阈值分割、中值滤波和高斯滤波技术对生成的图片进行初步处理,去除低亮度的像素值、椒盐噪声以及高斯噪声;共获得生成的漏涂缺陷图片1600张,生成的过量涂覆缺陷图片1500张。

S212、语义分割:使用Labelme工具对去除噪声之后的缺陷图片进行语义分割,使用OpenCV库绘制的语义分割图片将作为Pix2 Pix的标签输入进行缺陷图片的生成;

S213、建立两个变量imgs_path和annos_path,并通过glob方法将原始缺陷图片以及语义分割图片的所有路径存储到变量中;

S214、图片的张量化及归一化:使用PyTorch里的ToTensor以及Normalize方法将每一张输入的灰度图片转换到-1~1之间;

S215、自定义数据集类:自定义一个继承Dataset类的数据集类,用于处理自己的数据集,在类中,定义了获取文件数量、获取索引值、获取标签值、获取图片路径值等成员,用于对图片进行灵活处理;

S216、数据集制作:在S111所述的自定义数据集类基础上加载原始图片以及对应的语义分割图片,设置批量值BATCH_SIZE大小为16,开启随机读取和重复读取;

S22、Pix2Pix生成器设计:

S221、定义下采样类:该类命名为Downsample,由一个卷积层、一个LeakyReLU激活函数、一个批标准化组成,其中批标准化设置了一个触发条件,需要设置参数is_bn值为True才会激活,这样的设计可以灵活使用批标准化;

S222、定义上采样类:类命名为Upsample,该类由一个反卷积层、一个LeakyReLU激活函数、一个批标准化组成,并设置了dropout的触发条件,当参数is_drop值为True才会激活,

用于增加生成器的随机性;

S223、定义通道注意力机制类:类命名为ChannelAttention,该类下默认通道缩减比例为16,并创建一个自适应平均池化层,将输入的特征图池化成一个尺寸为1x1的特征图;定义一个序列模型,包括两个线性层和激活函数,用于学习通道权重;首先,通过一个线性层将输入特征图的通道数降低为原来的1/16,然后经过ReLU激活,接着再通过一个线性层将通道数恢复到原始值,最后经过Sigmoid激活函数,得到通道注意力权重;使用前向传播函数,计算通道注意力的输出;通过全局平均池化将输入特征图降维到每个通道的平均值,然后经过学习到的通道权重,得到加权的通道特征。这个加权的特征与原始输入特征图相乘,产生增强的通道注意力特征;

S224、定义空间注意力机制类:类命名为SpatialAttention;创建一个卷积层,输入通道数为2(平均值和最大值的堆叠),输出通道数为1,卷积核大小为3或7,不使用偏置。通过计算输入特征图的平均值和最大值,得到两个特征图,然后将这两个特征图堆叠起来,输入到卷积层中,得到增强的空间注意力特征;

S225、创建生成器:通过组合上采样和下采样,构成U-Net结构,输出(3,256,256)的图片,再将该数据灰度化,获得需要的灰度图;为了增强模型对输入数据的关注能力,提高模型性能,在U-Net结构中引入注意力机制,在上采样和下采样之间的层添加注意力模块;

S23、Pix2Pix判别器设计:

S231、判别器输入图片合并:判别器接受到的数据有两个,一个是生成的图片,另一个是语义分割图片,两者均为(1,256,256)图片,将两者合并,变为(2,256,256)的数据;

S232、输出判别值:通过卷积、批标准化、全连接层,将其输出大小为1的数据,最后使用sigmoid函数进行激活,获得0~1的数值,作为判别值;

S24、学习速率的设置:将生成器的学习速率设置为0.0001,将判别器的学习速率设置为0.00002;并且对学习速率设置衰减系数,这样的好处是削弱判别器的能力,使得生成器在早期阶段具有抗衡判别器的能力;

S25、配置模型训练参数

S251、设备选择:为了使得深度学习的代码具有更好的兼容性,先对硬件设备的GPU可用性进行判断,如果GPU不可使用,则设备选择为CPU;

S252、生成器损失函数:使用二分类的交叉熵损失函数,将判别器对生成图片的输出与全1向量进行对比,计算损失;此外引入L1损失,这部分损失用于约束生成器的输出与真实图片之间的相似性;综合考虑,生成器的总损失为交叉熵损失和L1损失的线性组合;

S253、判别器损失函数:判别器损失函数用于鼓励生成器生成的图片被判别器判别为真实图片,通常包括两部分,分别为真实图片损失和生成图片损失;真实图片损失用于训练判别器辨别真实图片,使用一个二分类的损失函数,将判别器对真实图片的输出,生成图片损失;总的判别器损失为这两部分损失之和;

S254、优化器选择:生成器和判别器的优化器均选择Adam,除学习速率如S41中所述,Adam的其他参数均为默认值;

S26、模型的保存:一个训练完成的生成器,包含结构和权重两个部分;当训练完成之后,所有的信息都保存在了model中,但此时只在内存里;通过save方法,将网络的框架以及权重全都保存到"Pix2Pix_model.pth"文件中;在需要调用时,只需要加载pth文件即可;

S27、缺陷图片的生成:基于S26中所述的生成器模型,构建一个循环,把如S212所述的语义分割图片作为输入,使用模型进行预测获得输出的缺陷图片。

2.根据权利要求1所述的一种基于改进Pix2 Pix的复合材料涂层缺陷数据集扩增方法,其特征在于:所述S112中,将长度为1的标签数据转换为独热编码,这样可以更好地参与卷积或反卷积运算。

3.根据权利要求1所述的一种基于改进Pix2 Pix的复合材料涂层缺陷数据集扩增方法,其特征在于:所述S124中,输入的噪声与标签数据经过全连接层后形状是一样的,并且将两者在通道维度上进行合并。

4.根据权利要求1所述的一种基于改进Pix2 Pix的复合材料涂层缺陷数据集扩增方法,其特征在于:所述S14中,生成器的学习速率要大于判别器,这样可以更好地指导生成器。

5.根据权利要求1所述的一种基于改进Pix2 Pix的复合材料涂层缺陷数据集扩增方法,其特征在于:所述S22中,生成器设计中引入了通道注意力机制以及空间注意力机制。

6.根据权利要求1所述的一种基于改进Pix2 Pix的复合材料涂层缺陷数据集扩增方法,其特征在于:所述S252中,生成器损失函数由交叉熵损失和L1损失的线性组合。

一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法

技术领域

[0001] 本发明涉及深度学习领域,具体为一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法。

背景技术

[0002] 复合材料涂层是发动机推进剂与绝热防护层粘接的关键功能材料。在使用图像处理方法检测发动机复合材料涂层缺陷时,会因为样本的数量不足导致检测和识别的效果不佳。生成对抗网络是一种强大的深度学习模型,利用生成对抗网络的生成器网络,结合先进的图像处理和复合材料涂层缺陷的特征,能够生成高度逼真且具有复合材料涂层缺陷特征的图片样本。该方法可以用于生成大量不同类型和严重程度的复合材料涂层缺陷样本图片,为复合材料涂层缺陷的检测和评估提供可靠的数据支持,且可以用于优化和改进其他缺陷检测算法和技术。

发明内容

[0003] (一)解决的技术问题

[0004] 针对现有技术的不足,本发明提供了一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法,解决了上述背景技术中所存在的问题。

[0005] (二)技术方案

[0006] 为实现上述目的,本发明提供如下技术方案:一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法,包括以下步骤:

[0007] S1、基于条件生成对抗网络获得语义分割样本:

[0008] S11、数据预处理:

[0009] S111、语义分割图片制作:创建“0”、“1”、“2”三个文件夹,分别代表无缺陷图片、漏涂缺陷、过量涂覆三种情况;人工筛选获取到的原始图片,确定其缺陷类型后放置在对应的文件夹下;本发明使用的无缺陷图片共450张、漏涂缺陷图片486张、过量涂覆443张;使用labelme工具对原始图片中的缺陷进行标注;使用labelme标注后的文件为json格式的文件。使用python中的json库对json文件进行解析,并使用OpenCV库绘制语义分割图片;

[0010] S112、标签转化为独热编码:将标签转换为独热编码可以提供一种统一的表示方式,使生成器和判别器网络能够更好地协同工作,提高生成对抗网络的性能;标签转换为独热编号后,可以更方便参与卷积或反卷积;

[0011] S113、图片的张量化及归一化:使用PyTorch里的ToTensor以及Normalize方法将每一张输入的灰度图片转换到-1~1之间;

[0012] S114、自定义数据集类:自定义一个继承Dataset类的数据集类,用于处理自己的数据集,在类中,定义了获取文件数量、获取索引值、获取标签值、获取图片路径值等成员,用于对图片进行灵活处理;

[0013] S115、数据集制作:在S114所述的自定义数据集类基础上加载语义分割图片,设置

批量值BATCH_SIZE大小为16,开启随机读取和重复读取;S12、生成器设计:

[0014] S121、创建随机噪声:定义一个长度为500的噪声,该噪声遵循正态分布;该噪声是生成器的输入,噪声数据经过一系列的全连接层以及反卷积层操作后,变换为 $1 \times 256 \times 256$ 的灰度图,即形成目标图片;

[0015] S122、改变标签数据形状:标签经过独热编码之后,将成为长度为3的张量,经过数层全连接层神经网络之后,将该向量的形状转化为(128,16,16);

[0016] S123、改变输入噪声形状:输入噪声的原始形状为长度500且服从正态分布的随机张量,将该张量经过数层全连接层神经网络之后,将该向量的形状转化为(128,16,16);

[0017] S124、合并标签与噪声:将改变形状后的输入噪声与标签进行通道维度上的合并,即变成(256,16,16)形状的数据;

[0018] S125、反卷积与批标准化:对合并后的数据进行反卷积操作,为了将其形成最后的目标图片大小,并且获取到真正有用的缺陷特征信息,每一次反卷积都是使用leaky_relu函数进行激活,最终形成(1,256,256)形状的数据,即一张灰度图;

[0019] S13、判别器设计:

[0020] S131、改变标签数据形状:判别器接受到的数据有两个,一个数生成的图片,另一个数独热编码后的标签,为了能生成器对两者进行判断,需要先将标签通过全连接神经网络转换为 $1 \times 256 \times 256$ 大小的形状,再将其展平为(1,256,256)形状;

[0021] S132、合并标签与生成的图片:将形状为(1,256,256)的标签与形状为(1,256,256)的生成图片进行合并,变为形状为(2,256,256)的数据;

[0022] S133、设计卷积层:设计卷积层,依次将合并后的形状通道数增加,宽和高减小,每一次卷积都是使用leaky_relu函数进行激活;最终形成形状为(128,63,63)的张量;

[0023] S134、批标准化:对形状为(128,63,63)的张量进行批标准化处理;

[0024] S135、数据降维:将(128,63,63)的张量展平为长度为 $128 * 63 * 63$ 长度的张量;

[0025] S136、全连接层:使用全连接层,将 $128 * 63 * 63$ 的张量连接到单位1的张量上,再使用sigmoid进行激活;

[0026] S14、学习速率的设置:将生成器的学习速率设置为0.001,将判别器的学习速率设置为0.0001;这样的好处是削弱判别器的能力,使得生成器在早期阶段具有抗衡判别器的能力;

[0027] S15、配置模型训练参数:

[0028] S151、设备选择:为了使得深度学习的代码具有更好的兼容性,先对硬件设备的GPU可用性进行判断,如果GPU不可使用,则设备选择为CPU;

[0029] S152、损失函数:因为判别器最后的输出结果是0~1的概率值,属于二分类问题,选择BCELoss作为训练的损失函数;

[0030] S153、优化器选择:生成器和判别器的优化器均选择Adam,除学习速率如S41中所述,Adam的其他参数均为默认值;

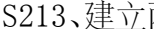
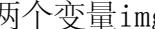
[0031] S16、模型的保存:一个训练完成的神经网络,包含结构和权重两个部分;当训练完成之后,所有的信息都保存在了model中,但此时只在内存里;通过save方法,将网络的框架以及权重全都保存到"CGAN_model.pth"文件中;在需要调用时,只需要加载pth文件即可;

[0032] S2、基于改进的Pix2Pix生成缺陷图片:

[0033] S21、数据预处理:

[0034] S211、阈值分割:通过所述S1中的方法,可以获得训练过程中得到的图片,通过手动采集质量较好、具有缺陷特征的图片,放置在“Pix2PixData”数据集中,先使用基于OpenCV的阈值分割、中值滤波和高斯滤波技术对生成的图片进行初步处理,去除低亮度的像素值、椒盐噪声以及高斯噪声;共获得生成的漏涂缺陷图片1600张,生成的过量涂覆缺陷图片1500张。

[0035] S212、语义分割:使用Labelme工具对去除噪声之后的缺陷图片进行语义分割,使用OpenCV库绘制的语义分割图片将作为Pix2Pix的标签输入进行缺陷图片的生成;

[0036] S213、建立两个变量和,并通过glob方法将原始缺陷图片以及语义分割图片的所有路径存储到变量中;

[0037] S214、图片的张量化及归一化:使用PyTorch里的ToTensor以及Normalize方法将每一张输入的灰度图片转换到-1~1之间;

[0038] S215、自定义数据集类:自定义一个继承Dataset类的数据集类,用于处理自己的数据集,在类中,定义了获取文件数量、获取索引值、获取标签值、获取图片路径值等成员,用于对图片进行灵活处理;

[0039] S216、数据集制作:在S111所述的自定义数据集类基础上加载原始图片以及对应的语义分割图片,设置批量值BATCH_SIZE大小为16,开启随机读取和重复读取;

[0040] S22、Pix2Pix生成器设计:

[0041] S221、定义下采样类:该类命名为Downsample,由一个卷积层、一个LeakyReLU激活函数、一个批标准化组成,其中批标准化设置了一个触发条件,需要设置参数is_bn值为True才会激活,这样的设计可以灵活使用批标准化;

[0042] S222、定义上采样类:类命名为Upsample,该类由一个反卷积层、一个LeakyReLU激活函数、一个批标准化组成,并设置了dropout的触发条件,当参数is_drop值为True才会激活,用于增加生成器的随机性;

[0043] S223、定义通道注意力机制类:类命名为ChannelAttention,该类下默认通道缩减比例为16,并创建一个自适应平均池化层,将输入的特征图池化成一个尺寸为1x1的特征图;定义一个序列模型,包括两个线性层和激活函数,用于学习通道权重;首先,通过一个线性层将输入特征图的通道数降低为原来的1/16,然后经过ReLU激活,接着再通过一个线性层将通道数恢复到原始值,最后经过Sigmoid激活函数,得到通道注意力权重;使用前向传播函数,计算通道注意力的输出;通过全局平均池化将输入特征图降维到每个通道的平均值,然后经过学习到的通道权重,得到加权的通道特征。这个加权的特征与原始输入特征图相乘,产生增强的通道注意力特征;

[0044] S224、定义空间注意力机制类:类命名为SpatialAttention;创建一个卷积层,输入通道数为2(平均值和最大值的堆叠),输出通道数为1,卷积核大小为3或7,不使用偏置。通过计算输入特征图的平均值和最大值,得到两个特征图,然后将这两个特征图堆叠起来,输入到卷积层中,得到增强的空间注意力特征;

[0045] S225、创建生成器:通过组合上采样和下采样,构成U-Net结构,输出(3,256,256)的图片,再将该数据灰度化,获得需要的灰度图;为了增强模型对输入数据的关注能力,提高模型性能,在U-Net结构中引入注意力机制,在上采样和下采样之间的层添加注意力模

块;

[0046] S23、Pix2Pix判别器设计:

[0047] S231、判别器输入图片合并:判别器接受到的数据有两个,一个是生成的图片,另一个是语义分割图片,两者均为(1,256,256)图片,将两者合并,变为(2,256,256)的数据;

[0048] S232、输出判别值:通过卷积、批标准化、全连接层,将其输出大小为1的数据,最后使用sigmoid函数进行激活,获得0~1的数值,作为判别值;

[0049] S24、学习速率的设置:将生成器的学习速率设置为0.0001,将判别器的学习速率设置为0.00002;并且对学习速率设置衰减系数,这样的好处是削弱判别器的能力,使得生成器在早期阶段具有抗衡判别器的能力;

[0050] S25、配置模型训练参数:

[0051] S251、设备选择:为了使得深度学习的代码具有更好的兼容性,先对硬件设备的GPU可用性进行判断,如果GPU不可使用,则设备选择为CPU;

[0052] S252、生成器损失函数:使用二分类的交叉熵损失函数,将判别器对生成图片的输出与全1向量进行对比,计算损失;此外引入L1损失,这部分损失用于约束生成器的输出与真实图片之间的相似性;综合考虑,生成器的总损失为交叉熵损失和L1损失的线性组合;

[0053] S253、判别器损失函数:判别器损失函数用于鼓励生成器生成的图片被判别器判别为真实图片,通常包括两部分,分别为真实图片损失和生成图片损失;真实图片损失用于训练判别器辨别真实图片,使用一个二分类的损失函数,将判别器对真实图片的输出,生成图片损失;总的判别器损失为这两部分损失之和;

[0054] S254、优化器选择:生成器和判别器的优化器均选择Adam,除学习速率如S41中所述,Adam的其他参数均为默认值;

[0055] S26、模型的保存:一个训练完成的生成器,包含结构和权重两个部分;当训练完成之后,所有的信息都保存在了model中,但此时只在内存里;通过save方法,将网络的框架以及权重全都保存到"Pix2Pix_model.pth"文件中;在需要调用时,只需要加载pth文件即可;

[0056] S27、缺陷图片的生成:基于S26中所述的生成器模型,构建一个循环,把如S212所述的语义分割图片作为输入,使用模型进行预测获得输出的缺陷图片。

[0057] 优选的,所述S111中,Labelme是图片上的缺陷标注最常用的工具之一,除此之外,Labelimg或CVAT等工具也同样适合做本发明的标注工具。

[0058] 优选的,所述S115中,设置BATCH_SIZE为16,也就是每次取出16张图片及对应标签,参与训练;如果计算机配置较高、显卡内存较大,可以将BATCH_SIZE设置为更大的值;由于需要反复训练,因此需要用到repeat函数来重复产生数据;为了使得网络获得较好的泛化能力,需要使用shuffle来进行乱序。

[0059] 优选的,所述S121中,随机噪声的大小500是一个超参数,可以根据实际使用情况进行调整;随机规律一般选择正态分布,也可以尝试选择泊松分布或者指数分布等其他分布类型。

[0060] 优选的,所述S14中,生成器的学习速率比判别器的学习速率高即可,因为这样可以避免判别器的损失值下降太快,导致判别器无法指导生成器。

[0061] 优选的,所述S225中,通道注意力机制类和空间注意力机制类可放置在上采样和下采样之间的任意位置放置。

[0062] 优选的,所述S252中,生成器损失函数由交叉熵损失函数和L1损失的线性组合。

[0063] (三)有益效果

[0064] 本发明提供了一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法,具备以下有益效果:

[0065] 本发明应用在复合材料涂层缺陷检测任务的数据增强阶段,通过改进的Pix2Pix来生成具有不同缺陷类型的样本,涵盖不同的缺陷形状和细节,可以增加训练数据的数量和质量,从而提高缺陷分类模型的鲁棒性和泛化能力,此外还可以平衡各个类别的样本分布,使得后续的缺陷识别分类模型在所有类别上都能够得到充分的识别。

附图说明

[0066] 图1为条件生成对抗网络的生成器结构图;

[0067] 图2为条件生成对抗网络的判别器结构图;

[0068] 图3为条件对抗生成网络生成的图片示例;

[0069] 图4为Pix2Pix的生成器结构图;

[0070] 图5为Pix2Pix的判别器结构图;

[0071] 图6为Pix2Pix生成的图片示例;

[0072] 图7为整体方案技术路线图;

具体实施方式

[0073] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0074] 本发明提供一种技术方案:一种基于改进Pix2Pix的复合材料涂层缺陷数据集扩增方法,包括以下步骤:

[0075] S1、基于条件生成对抗网络获得语义分割样本:

[0076] S11、数据预处理:

[0077] S111、语义分割图片制作:创建"0"、"1"、"2"三个文件夹,分别代表无缺陷图片、漏涂缺陷、过量涂覆三种情况;人工筛选获取到的原始图片,确定其缺陷类型后放置在对应的文件夹下;本发明使用的无缺陷图片共450张、漏涂缺陷图片486张、过量涂覆443张;使用Labelme工具对原始图片中的缺陷进行标注;使用Labelme标注后的文件为json格式的文件。使用python中的json库对json文件进行解析,并使用OpenCV库绘制语义分割图片;

[0078] S112、标签转化为独热编码:将标签转换为独热编码可以提供一种统一的表示方式,使生成器和判别器网络能够更好地协同工作,提高生成对抗网络的性能;标签转换为独热编号后,可以更方便参与卷积或反卷积;

[0079] S113、图片的张量化及归一化:使用PyTorch里的ToTensor以及Normalize方法将每一张输入的灰度图片转换到-1~1之间;

[0080] S114、自定义数据集类:自定义一个继承Dataset类的数据集类,用于处理自己的数据集,在类中,定义了获取文件数量、获取索引值、获取标签值、获取图片路径值等成员,

用于对图片进行灵活处理;

[0081] S115、数据集制作:在S114所述的自定义数据集类基础上加载语义分割图片,设置批量值BATCH_SIZE大小为16,开启随机读取和重复读取;S12、生成器设计:

[0082] S121、创建随机噪声:定义一个长度为500的噪声,该噪声遵循正态分布;该噪声是生成器的输入,噪声数据经过一系列的全连接层以及反卷积层操作后,变换为 $1 \times 256 \times 256$ 的灰度图,即形成目标图片;

[0083] S122、改变标签数据形状:标签经过独热编码之后,将成为长度为3的张量,经过数层全连接层神经网络之后,将该向量的形状转化为(128,16,16);

[0084] S123、改变输入噪声形状:输入噪声的原始形状为长度500且服从正态分布的随机张量,将该张量经过数层全连接层神经网络之后,将该向量的形状转化为(128,16,16);

[0085] S124、合并标签与噪声:将改变形状后的输入噪声与标签进行通道维度上的合并,即变成(256,16,16)形状的数据;

[0086] S125、反卷积与批标准化:对合并后的数据进行反卷积操作,为了将其形成最后的目标图片大小,并且获取到真正有用的缺陷特征信息,每一次反卷积都是使用leaky_relu函数进行激活,最终形成(1,256,256)形状的数据,即一张灰度图,反卷积以及批标准化的过程如图1所示;

[0087] S13、判别器设计:

[0088] S131、改变标签数据形状:判别器接受到的数据有两个,一个数生成的图片,另一个数独热编码后的标签,为了能生成器对两者进行判断,需要先将标签通过全连接神经网络转换为 $1 \times 256 \times 256$ 大小的形状,再将其展平为(1,256,256)形状;

[0089] S131、合并标签与生成的图片:将形状为(1,256,256)的标签与形状为(1,256,256)的生成图片进行合并,变为形状为(2,256,256)的数据;

[0090] S132、设计卷积层:设计卷积层,依次将合并后的形状通道数增加,宽和高减小,每一次卷积都是使用leaky_relu函数进行激活;最终形成形状为(128,63,63)的张量;

[0091] S133、批标准化:对形状为(128,63,63)的张量进行批标准化处理;

[0092] S134、数据降维:将(128,63,63)的张量展平为长度为 $128 * 63 * 63$ 长度的张量;

[0093] S135、全连接层:使用全连接层,将 $128 * 63 * 63$ 的张量连接到单位1的张量上,再使用sigmoid进行激活;

[0094] S14、学习速率的设置:将生成器的学习速率设置为0.001,将判别器的学习速率设置为0.0001;这样的好处是削弱判别器的能力,使得生成器在早期阶段具有抗衡判别器的能力;

[0095] S15、配置模型训练参数:

[0096] S151、设备选择:为了使得深度学习的代码具有更好的兼容性,先对硬件设备的GPU可用性进行判断,如果GPU不可使用,则设备选择为CPU;

[0097] S152、损失函数:因为判别器最后的输出结果是0~1的概率值,属于二分类问题,选择BCELoss作为训练的损失函数;

[0098] S153、优化器选择:生成器和判别器的优化器均选择Adam,除学习速率如S41中所述,Adam的其他参数均为默认值;

[0099] S16、模型的保存:一个训练完成的神经网络,包含结构和权重两个部分;当训练完

成之后,所有的信息都保存在了model中,但此时只在内存里;通过save方法,将网络的框架以及权重全都保存到"CGAN_model.pth"文件中;在需要调用时,只需要加载pth文件即可,训练过程中测试的一些生成图片如图3所示;

[0100] S2、基于改进的Pix2Pix生成缺陷图片:

[0101] S21、数据预处理

[0102] S211、阈值分割:通过所述S1中的方法,可以获得训练过程中得到的图片,通过手动采集质量较好、具有缺陷特征的图片,放置在"Pix2PixData"数据集中,先使用基于OpenCV的阈值分割、中值滤波和高斯滤波技术对生成的图片进行初步处理,去除低亮度的像素值、椒盐噪声以及高斯噪声;共获得生成的漏涂缺陷图片1600张,生成的过量涂覆缺陷图片1500张;

[0103] S212、语义分割:使用Labelme工具对去除噪声之后的缺陷图片进行语义分割,使用OpenCV库绘制的语义分割图片将作为Pix2Pix的标签输入进行缺陷图片的生成;

[0104] S213、建立两个变量imgs_path和annos_path,并通过glob方法将原始缺陷图片以及语义分割图片的所有路径存储到变量中;

[0105] S214、图片的张量化及归一化:使用PyTorch里的ToTensor以及Normalize方法将每一张输入的灰度图片转换到-1~1之间;

[0106] S215、自定义数据集类:自定义一个继承Dataset类的数据集类,用于处理自己的数据集,在类中,定义了获取文件数量、获取索引值、获取标签值、获取图片路径值等成员,用于对图片进行灵活处理;

[0107] S216、数据集制作:在S111所述的自定义数据集类基础上加载原始图片以及对应的语义分割图片,设置批量值BATCH_SIZE大小为16,开启随机读取和重复读取;

[0108] S22、Pix2Pix生成器设计:

[0109] S221、定义下采样类:该类命名为Downsample,由一个卷积层、一个LeakyReLU激活函数、一个批标准化组成,其中批标准化设置了一个触发条件,需要设置参数is_bn值为True才会激活,这样的设计可以灵活使用批标准化;

[0110] S222、定义上采样类:类命名为Upsample,该类由一个反卷积层、一个LeakyReLU激活函数、一个批标准化组成,并设置了dropout的触发条件,当参数is_drop值为True才会激活,用于增加生成器的随机性;

[0111] S223、定义通道注意力机制类:类命名为ChannelAttention,该类下默认通道缩减比例为16,并创建一个自适应平均池化层,将输入的特征图池化成一个尺寸为1x1的特征图;定义一个序列模型,包括两个线性层和激活函数,用于学习通道权重;首先,通过一个线性层将输入特征图的通道数降低为原来的1/16,然后经过ReLU激活,接着再通过一个线性层将通道数恢复到原始值,最后经过Sigmoid激活函数,得到通道注意力权重;使用前向传播函数,计算通道注意力的输出;通过全局平均池化将输入特征图降维到每个通道的平均值,然后经过学习到的通道权重,得到加权的通道特征。这个加权的特征与原始输入特征图相乘,产生增强的通道注意力特征;

[0112] S224、定义空间注意力机制类:类命名为SpatialAttention;创建一个卷积层,输入通道数为2(平均值和最大值的堆叠),输出通道数为1,卷积核大小为3或7,不使用偏置。通过计算输入特征图的平均值和最大值,得到两个特征图,然后将这两个特征图堆叠起来,

输入到卷积层中,得到增强的空间注意力特征;

[0113] S225、创建生成器:通过组合上采样和下采样,构成U-Net结构,输出(3,256,256)的图片,再将该数据灰度化,获得需要的灰度图;为了增强模型对输入数据的关注能力,提高模型性能,在U-Net结构中引入注意力机制,在上采样和下采样之间的层添加注意力模块,生成器的结构如图4所示;

[0114] S23、Pix2Pix判别器设计:

[0115] S231、判别器输入图片合并:判别器接受到的数据有两个,一个是生成的图片,另一个是语义分割图片,两者均为(1,256,256)图片,将两者合并,变为(2,256,256)的数据;

[0116] S232、输出判别值:通过卷积、批标准化、全连接层,将其输出大小为1的数据,最后使用sigmoid函数进行激活,获得0~1的数值,作为判别值;判别器的结构如图5所示;

[0117] S24、学习速率的设置:将生成器的学习速率设置为0.0001,将判别器的学习速率设置为0.00002;并且对学习速率设置衰减系数,这样的好处是削弱判别器的能力,使得生成器在早期阶段具有抗衡判别器的能力;

[0118] S25、配置模型训练参数

[0119] S251、设备选择:为了使得深度学习的代码具有更好的兼容性,先对硬件设备的GPU可用性进行判断,如果GPU不可使用,则设备选择为CPU;

[0120] S252、生成器损失函数:使用二分类的交叉熵损失函数,将判别器对生成图片的输出与全1向量进行对比,计算损失;此外引入L1损失,这部分损失用于约束生成器的输出与真实图片之间的相似性;综合考虑,生成器的总损失为交叉熵损失和L1损失的线性组合;

[0121] S253、判别器损失函数:判别器损失函数用于鼓励生成器生成的图片被判别器判别为真实图片,通常包括两部分,分别为真实图片损失和生成图片损失;真实图片损失用于训练判别器辨别真实图片,使用一个二分类的损失函数,将判别器对真实图片的输出,生成图片损失;总的判别器损失为这两部分损失之和;

[0122] S254、优化器选择:生成器和判别器的优化器均选择Adam,除学习速率如S41中所述,Adam的其他参数均为默认值;

[0123] S26、模型的保存:一个训练完成的生成器,包含结构和权重两个部分;当训练完成之后,所有的信息都保存在了model中,但此时只在内存里;通过save方法,将网络的框架以及权重全都保存到"Pix2Pix_model.pth"文件中;在需要调用时,只需要加载pth文件即可;

[0124] S27、缺陷图片的生成:基于S26中所述的生成器模型,构建一个循环,把如S212所述的语义分割图片作为输入,使用模型进行预测获得输出的缺陷图片,生成的部分示例图片如图6所示。

[0125] 尽管已经示出和描述了本发明的实施例,对于本领域的普通技术人员而言,可以理解在不脱离本发明的原理和精神的情况下可以对这些实施例进行多种变化、修改、替换和变型,本发明的范围由所附权利要求及其等同物限定。

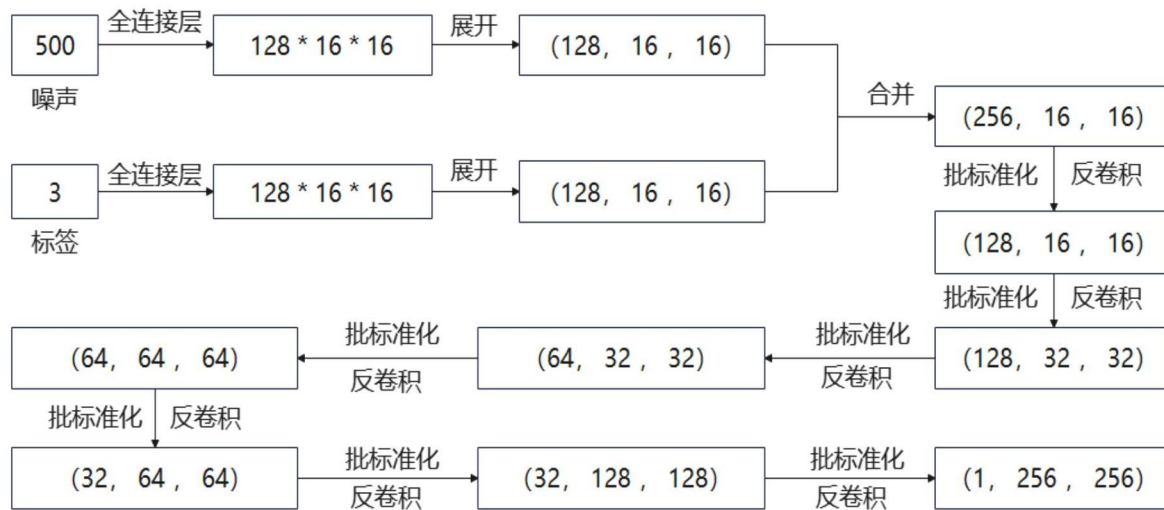


图1

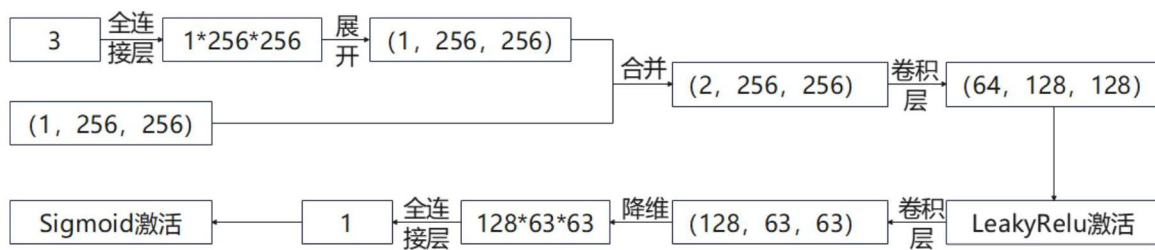


图2

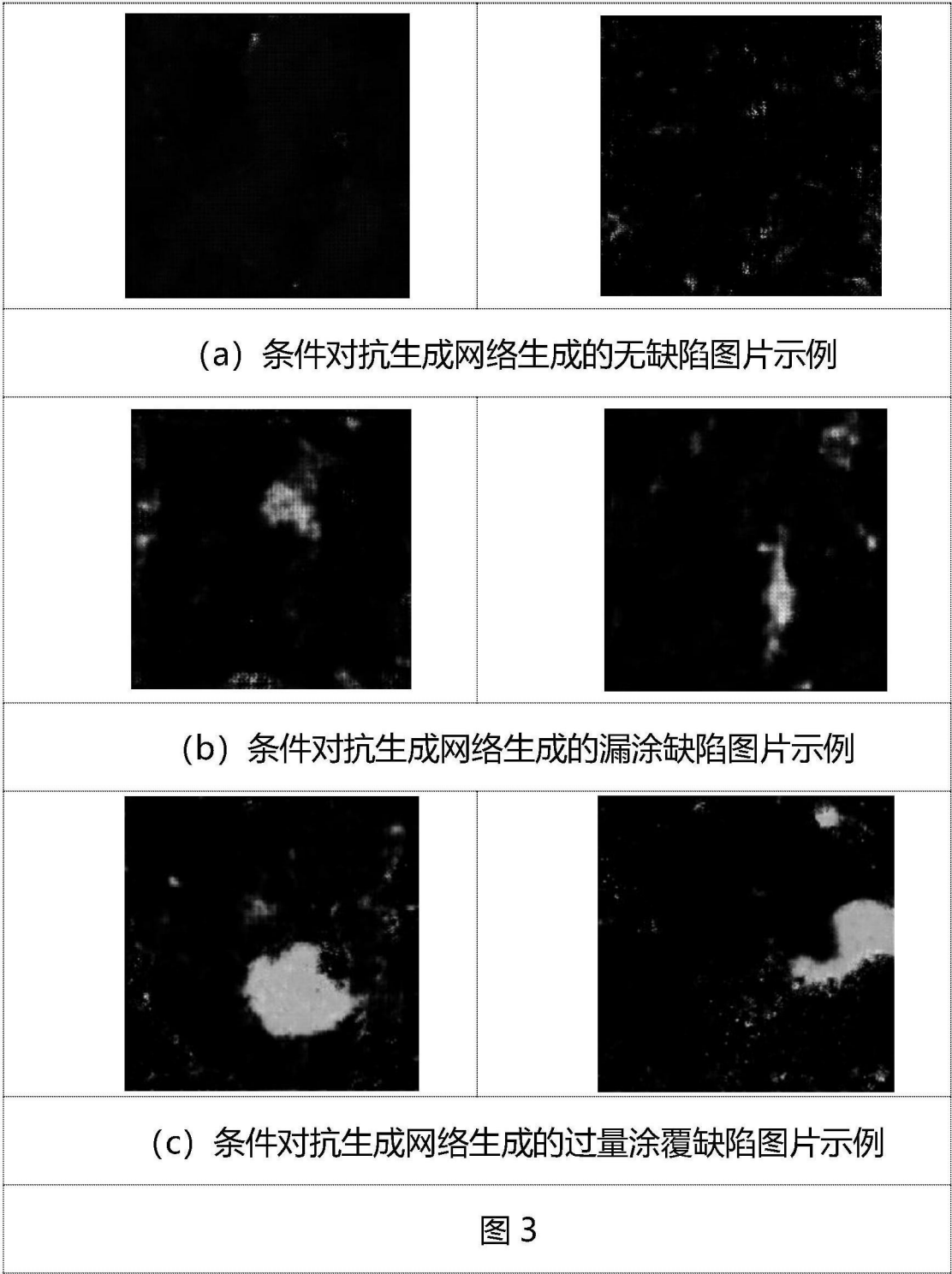


图3

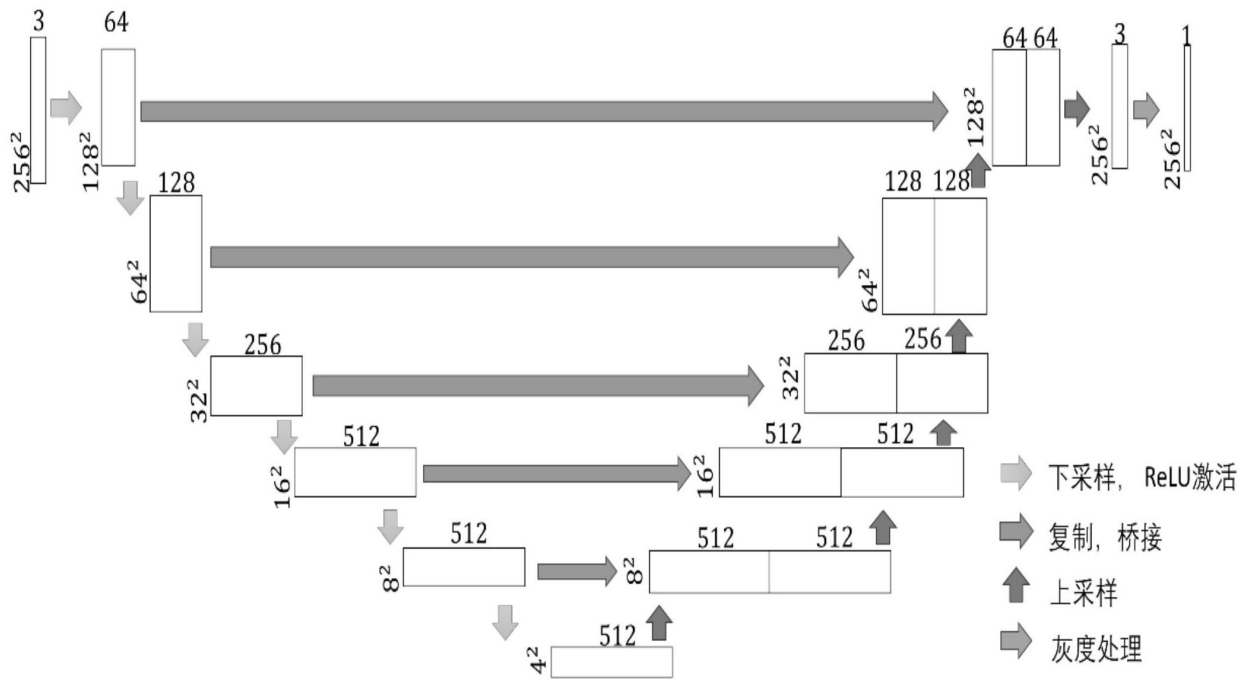


图4

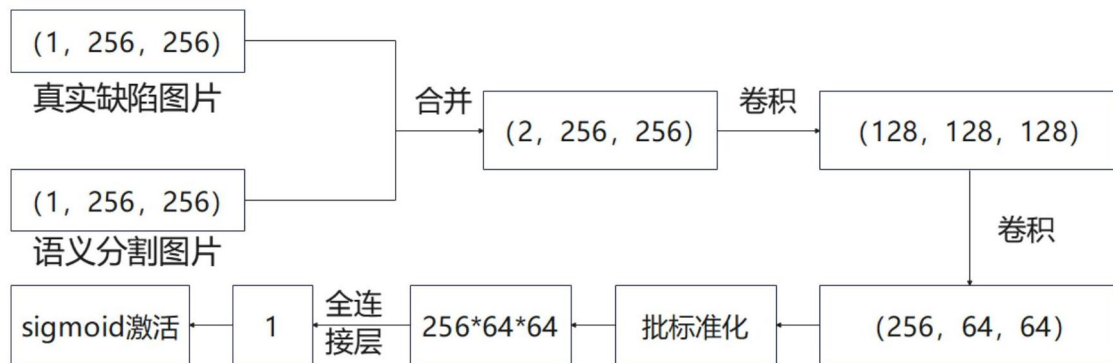


图5

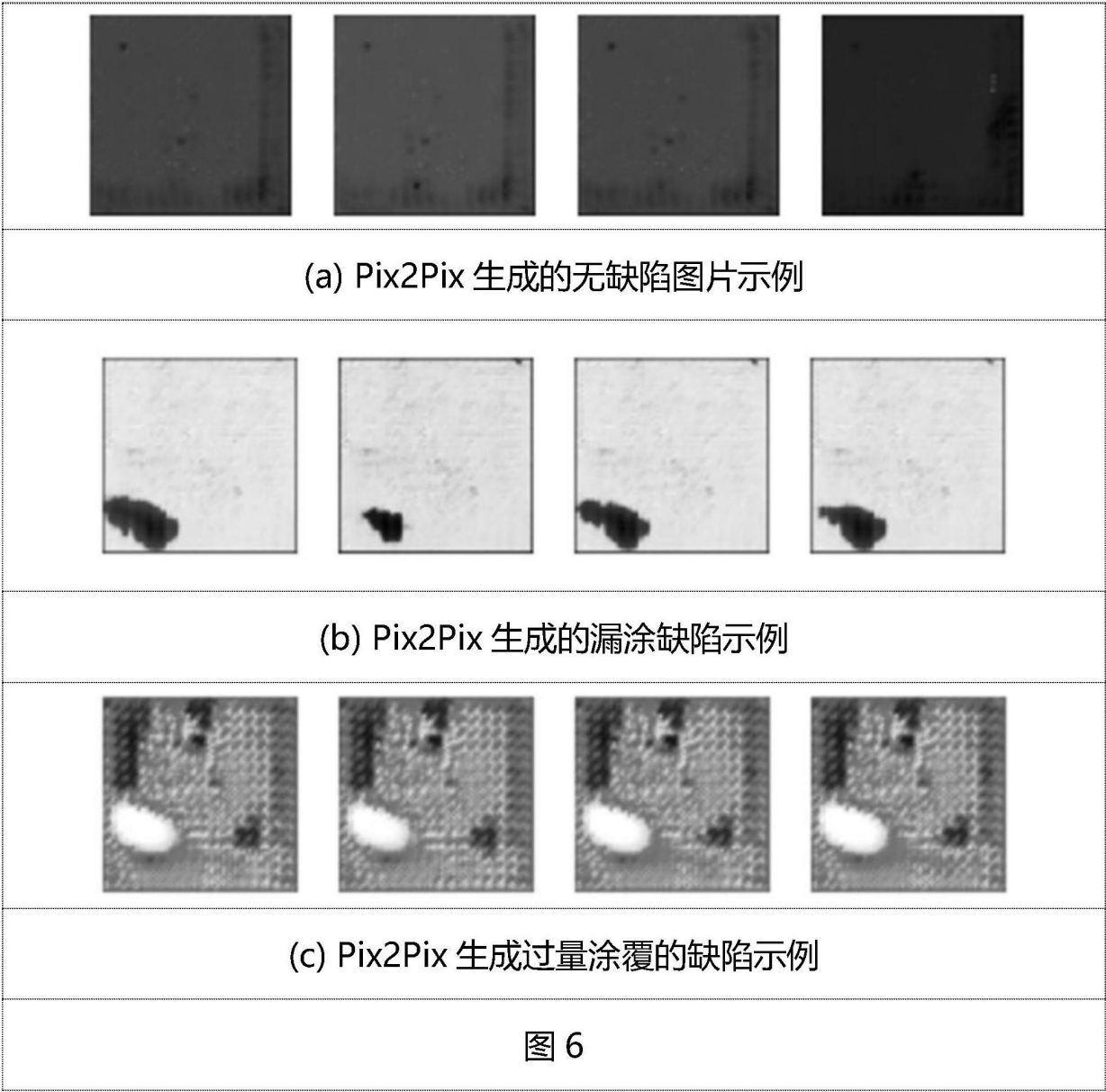


图6

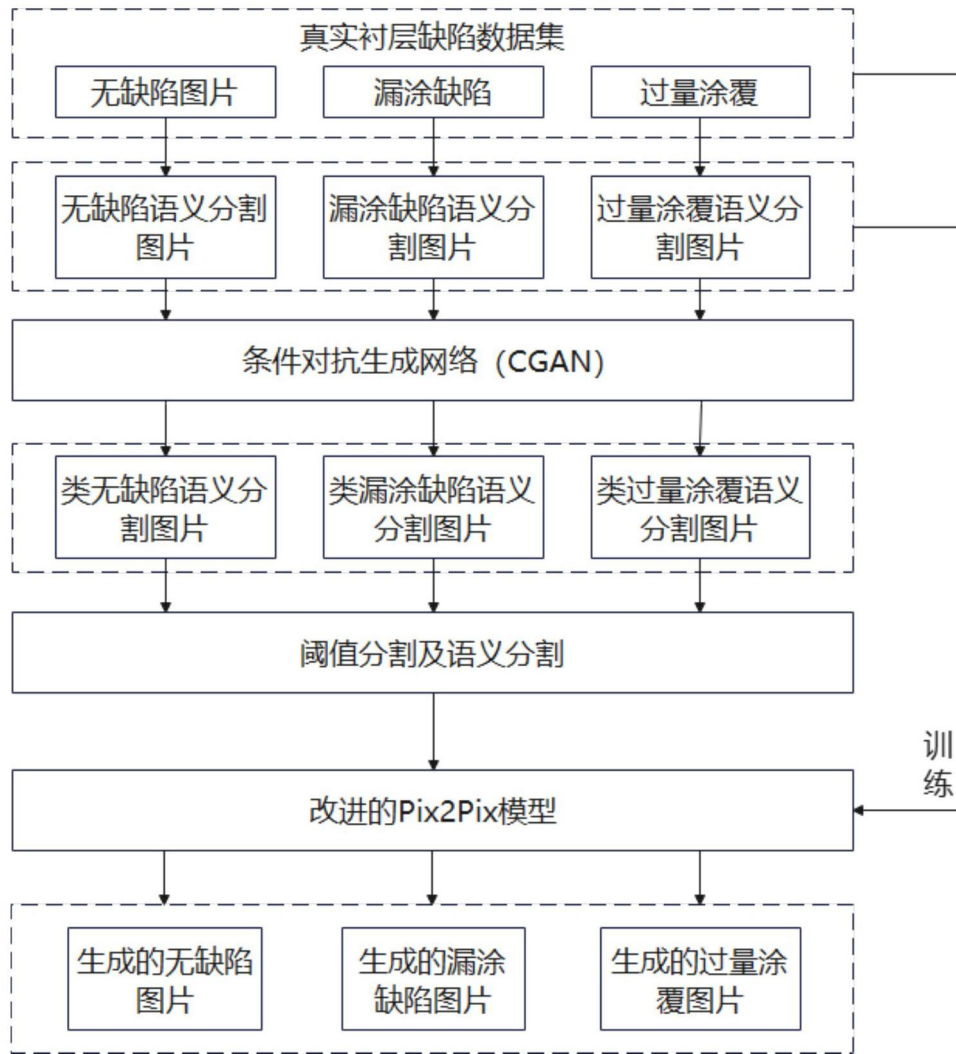


图7