



(12) 发明专利申请

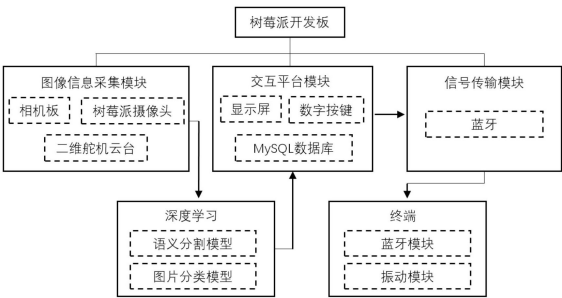
(10) 申请公布号 CN 116402998 A
(43) 申请公布日 2023. 07. 07

(21) 申请号 202310402205.1
(22) 申请日 2023.04.17
(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 余鹏 张洛伊
(51) Int.Cl.
G06V 10/26 (2022.01)
G06F 16/951 (2019.01)
G06V 10/56 (2022.01)
G06V 10/28 (2022.01)
G06V 10/774 (2022.01)
G06V 10/82 (2022.01)
G06V 10/764 (2022.01)
G06N 3/084 (2023.01)
G06N 3/045 (2023.01)

权利要求书3页 说明书7页 附图3页

(54) 发明名称
基于语义分割的睡眠被子覆盖率视觉识别方法

(57) 摘要
本发明涉及机器视觉技术领域,且公开了基于语义分割的睡眠被子覆盖率视觉识别方法,包括以下步骤:S1、语义分割模型训练:S11、语义分割图片的制作:通过爬虫以及人工采集等手段,获取约10000张图片,使用labelme工具进行语义图片的制作;在语义分割图片中,人的RGB值为(128,0,0),覆盖物的RGB值为(0,128,0),背景的RGB值为(0,0,0);使用OpenCV将RG B值进行映射。能够对人员的盖被情况进行准确的识别,从而能够使监护人员准确的了解被监控者的盖被情况,进而能够使监护人员在被监护者的被子离开身体一定程度后,能够及时的给监护人员预警,使监护人员及时的做出反应措施,从而能够及时的避免被监护者被细菌或者病毒入侵,出现感冒发烧等情况。



1. 基于语义分割的睡眠被子覆盖率视觉识别方法,其特征在于,包括以下步骤:

S1、语义分割模型训练

S11、语义分割图片的制作:通过爬虫以及人工采集等手段,获取约10000张图片,使用labelme工具进行语义图片的制作;在语义分割图片中,人的RGB值为(128,0,0),覆盖物的RGB值为(0,128,0),背景的RGB值为(0,0,0);使用OpenCV将RGB值进行映射,将语义分割图片中的每一个RGB值分别进行处理,将背景RGB值保持不变,将人RGB值修改为(1,1,1),将覆盖物RGB值修改为(2,2,2),最终使用OpenCV进行通道分离,即处理得到灰度图;

S12、将原始图片以及语义分割图片分别放在image文件夹以及annotations文件夹中,使用glob函数获取所有原始图片以及语义分割图片的路径,分别存储在变量images以及anno中;

创建seed随机数生成器种子,通过random函数的permutation命令获取原始图片数量相同长度的随机数组,作为索引值,进而获得乱序后的变量images以及anno,并且保证了images与anno中的每一个值都是一一对应的;获得乱序的变量images以及anno之后,使用TensorFlow中的from_tensor_slices对两者进行打包,获得数据集dataset。获取dataset中的20%作为测试数据集、80%作为训练数据集;

定义图片的读取函数,由于原始图片以及语义图片都是png格式,并且都是RGB图片,通道数都为3,因此可以共用解码器,定义函数read_png,主通过TensorFlow中的decode_png函数将图像数据转换为像素点的数据用于训练;

对图像进行随机裁剪增强,由于要求确保语义图片和原图保持一致进行裁剪,因此先将语义图片和原图片进行绑定,再裁剪之后,使用numpy库高级切片的方式对其分割开,获得裁剪位置一致的训练图片以及语义图片;

对原始图片和语义图片都进行归一化,归一化加快了梯度下降求最优解的速度,也即加快训练网络的收敛性,由于原始图片的RGB值为0~256的数字,因此可以将其同时除以128再减1,将其映射到-1~1,语义分割单通道图片不需要进行归一化处理;

定义读取训练图片的函数,调用前面的读取函数获得原始图片以及语义图片,调用前面的随机裁剪图片得到裁剪的原始图片以及语义图片,再通过随机数的判断,将裁剪后的原始图片以及语义图片进行随机翻转,最后将其归一化处理;

定义读取测试图片的函数,对于测试集的图片,不需要进行随机裁剪以及随机翻转等图像增强操作,只需要将其放缩到256x256大小,并进行归一化处理即可;

定义完读取训练图片的函数以及读取测试图片的函数之后,使用map函数将其分别映射到训练数据集以及测试数据集中,完成数据集的制作;

S13、构建U-Net模型:使用Python面向对象的编程方式分别定义上采样类和下采样类进行组装,构建U-Net网络;

S14、编译与训练:在定义模型之后,使用Adam优化器进行优化;训练5000步,记录每一步训练中训练集的损失值、准确率以及IOU值,和测试集的损失值、准确率以及IOU值,以便于神经网络模型的后续调整;

S15、模型的保存: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"model1.h5"文件中,在需要调用时,只需要使用load加载h5文件即可;

S2、分类模型的训练

S21、语义图片的分类：将单通道语义分割单通道图片作为训练集进行训练，具体按照覆盖面积将其分为5类，其中一种为未检测到的人，另外四类分别为：

分别定义标签0~4，标签0表示未检测到的人，标签1表示覆盖率约90%，标签2表示覆盖率约65%~85%，标签3表示覆盖率约30%~60%，标签4表示覆盖率低于30%，将使用labelme以及opencv处理得到单通道语义图片根据上面所述的分类标准，分为5类，在用于训练的文件夹“train”、用于测试的文件夹“test”里，分别创建“0”、“1”、“2”、“3”、“4”这五个文件夹，文件夹名则对应着每一个分类的标签；

S22、数据集的制作：首先使用python的glob库获取图片的地址，将图片的路径分别存在变量“test_image_path”以及“train_image_path”中，这两个变量分别代表着测试集所有图片的路径以及训练集所有的图片路径，通过random库的shuffle函数，打乱“test_image_path”以及“train_image_path”这两个变量里的数据；

使用split函数截取文件名字符串的数字即可作为标签，使用Python语言创建两个函数用来解码以及处理图片，由于训练的图片都是单通道的灰度图，因此仅需要定义一个函数，即可在训练集和测试集通用；

在完成标签的制作以及图像读取函数之后，使用TensorFlow的Dataset创建用于训练模型的数据集，将train文件夹里的图片路径和对应的标签写入train_image_ds变量中，将test文件夹里的图片路径和对应的标签写入test_image_ds变量中，最后通过map函数映射到每一个路径中，获得真正用于训练的训练集data_train以及测试集data_test；

S23、多分类模型构建：使用预训练神经网络MobileNetV3-Large进行分类的训练；使用MobileNetV3-Large时不包含顶层的全链接网络，而是使用自定义层，设置MobileNetV3-Large的权重值不变，权重值取公开数据集imagenet训练下的权重，同时注意添加Dropout层以抑制过拟合；

S24、编译与训练：使用Adam优化器进行编译，并设置Adam优化器的学习速率为0.0005；再使用sparse_categorical_crossentropy作为交叉熵损失函数；在设置好优化器、损失函数等之后，对该神经网络进行训练；

训练10000个epoch之后，设置MobileNetV3网络的前30层可以随着训练而修改，在此之前的网络权重仍然选择继承于imgnet数据集；

此外，还需要将Adam优化器的学习速率降低10倍，即 $0.005/10=0.0005$ ，设置完成后，在前面epoch为10000的基础上，再训练15000步；

S25、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“model2.h5”文件中，在需要调用时，只需要使用load加载h5文件即可。

2. 根据权利要求1所述的基于语义分割的睡眠被子覆盖率视觉识别方法，其特征在于：所述S11中，因为目标是识别人与被子之间的覆盖关系，因此具体可将每一张图片分割为三个部分，分别为人、被子、其他（即背景），这样的分类方式既能有效实现被子覆盖情况的识别，也可以减少网络的计算量。

3. 根据权利要求1所述的基于语义分割的睡眠被子覆盖率视觉识别方法，其特征在于：所述S12中，在将原始图片以及语义分割图片分别放在image文件夹以及annotations文件

夹中的时候,需要注意的是,务必确保原始图片与语义分割图片的命名一一对应。

4.根据权利要求1所述的基于语义分割的睡眠被子覆盖率视觉识别方法,其特征在于:所述S12中,在对图像进行随机裁剪增强的时候,具体为,先将原始图片以及对应的语义图片使用TensorFlow中的concat进行合并,再放缩为 280×280 尺寸,然后将合并并放缩的图片进行随机裁剪,得到的结果图片是 256×256 ,最后使用高级切片的方式,获得分开的原图和标签。

5.根据权利要求1所述的基于语义分割的睡眠被子覆盖率视觉识别方法,其特征在于:所述S14中,在训练中引入IOU指标用来评价模型的鲁棒性,要先将判断的结果进行转换,因为直接用模型预测的结果是一个长度为3(每一个标签对应一个数值)的张量,因此使用TensorFlow中的arg max获取预测中最大可能性数的索引,即预测的标签,用来与设定的标签进行比较。

6.根据权利要求1所述的基于语义分割的睡眠被子覆盖率视觉识别方法,其特征在于:所述S14中,在优化器中设置学习速率为0.0001,并在损失函数将优化器激活。

基于语义分割的睡眠被子覆盖率视觉识别方法

技术领域

[0001] 本发明涉及机器视觉技术领域,具体为基于语义分割的睡眠被子覆盖率视觉识别方法。

背景技术

[0002] 在家庭生活中,为了方便大人对孩童的查看和照顾,通常会在孩童的卧室中安装摄像头,用于实时掌握孩童的情况。但现有的摄像头只能将实时画面、孩童是否在监控区域中、孩童是否移动、孩童是否发出响声等信息发送给监护人终端设备上,而无法对孩童在休息的过程中,对被子的覆盖情况进行识别,导致监护人员无法及时的掌握孩童盖被情况,容易导致孩童将被子踢开,监护人员无法及时的对孩童进行盖被,从而导致孩童出现感冒发烧等情况。

发明内容

[0003] (一)解决的技术问题

[0004] 针对现有技术的不足,本发明提供了基于语义分割的睡眠被子覆盖率视觉识别方法,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:基于语义分割的睡眠被子覆盖率视觉识别方法,包括以下步骤:

[0007] S1、语义分割模型训练

[0008] S11、语义分割图片的制作:通过爬虫以及人工采集等手段,获取约10000张图片,使用labelme工具进行语义图片的制作;在语义分割图片中,人的RGB值为(128,0,0),覆盖物的RGB值为(0,128,0),背景的RGB值为(0,0,0);使用OpenCV将RGB值进行映射,将语义分割图片中的每一个RGB值分别进行处理,将背景RGB值保持不变,将人RGB值修改为(1,1,1),将覆盖物RGB值修改为(2,2,2),最终使用OpenCV进行通道分离,即处理得到灰度图;

[0009] S12、将原始图片以及语义分割图片分别放在image文件夹以及annotations文件夹中,需要注意的是,务必确保原始图片与语义分割图片的命名一一对应,使用glob函数获取所有原始图片以及语义分割图片的路径,分别存储在变量images以及anno中;

[0010] 创建seed随机数生成器种子,通过random函数的permutation命令获取原始图片数量相同长度的随机数组,作为索引值,进而获得乱序后的变量images以及anno,并且保证了images与anno中的每一个值都是一一对应的;获得乱序的变量images以及anno之后,使用TensorFlow中的from_tensor_slices对两者进行打包,获得数据集dataset。获取dataset中的20%作为测试数据集、80%作为训练数据集;

[0011] 定义图片的读取函数,由于原始图片以及语义图片都是png格式,并且都是RGB图片,通道数都为3,因此可以共用解码器,定义函数read_png,主通过TensorFlow中的decode_png函数将图像数据转换为像素点的数据用于训练;

[0012] 对图像进行随机裁剪增强,由于要求确保语义图片和原图保持一致进行裁剪,因此先将语义图片和原图片进行绑定,再裁剪之后,使用numpy库高级切片的方式对其分割开,获得裁剪位置一致的训练图片以及语义图片;

[0013] 对原始图片和语义图片都进行归一化,归一化加快了梯度下降求最优解的速度,也即加快训练网络的收敛性,由于原始图片的RGB值为0~256的数字,因此可以将其同时除以128再减1,将其映射到-1~1,语义分割单通道图片不需要进行归一化处理;

[0014] 定义读取训练图片的函数,调用前面的读取函数获得原始图片以及语义图片,调用前面的随机裁剪图片得到裁剪的原始图片以及语义图片,再通过随机数的判断,将裁剪后的原始图片以及语义图片进行随机翻转,最后将其归一化处理;

[0015] 定义读取测试图片的函数,对于测试集的图片,不需要进行随机裁剪以及随机翻转等图像增强操作,只需要将其放缩到256x256大小,并进行归一化处理即可;

[0016] 定义完读取训练图片的函数以及读取测试图片的函数之后,使用map函数将其分别映射到训练数据集以及测试数据集中,完成数据集的制作;

[0017] S13、构建U-Net模型:使用Python面向对象的编程方式分别定义上采样类和下采样类进行组装,构建U-Net网络;

[0018] S14、编译与训练:在定义模型之后,使用Adam优化器进行优化,Adam优化器结合了AdaGrad和RMSProp两种优化算法的优点,实现简单,计算高效,对内存需求少,在优化器中设置学习速率为0.0001,并在损失函数将优化器激活;训练5000步,记录每一步训练中训练集的损失值、准确率以及iou值,和测试集的损失值、准确率以及iou值,以便于神经网络模型的后续调整;

[0019] S15、模型的保存: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"model1.h5"文件中,在需要调用时,只需要使用load加载h5文件即可;

[0020] S2、分类模型的训练

[0021] S21、语义图片的分类:将单通道语义分割单通道图片作为训练集进行训练,具体按照覆盖面积将其分为5类,其中一种为未检测到,另外四类分别为:

[0022] 分别定义标签0~4,标签0表示未检测到,标签1表示覆盖率约90%,标签2表示覆盖率约65%~85%,标签3表示覆盖率约30%~60%,标签4表示覆盖率低于30%,将使用labelme以及opencv处理得到单通道语义图片根据上面所述的分类标准,分为5类,在用于训练的文件夹"train"、用于测试的文件夹"test"里,分别创建"0"、"1"、"2"、"3"、"4"这五个文件夹,文件夹名则对应着每一个分类的标签;

[0023] S22、数据集的制作:首先使用python的glob库获取图片的地址,将图片的路径分别存在变量"test_image_path"以及"train_image_path"中,这两个变量分别代表着测试集所有图片的路径以及训练集所有的图片路径,通过random库的shuffle函数,打乱"test_image_path"以及"train_image_path"这两个变量里的数据;

[0024] 使用split函数截取文件名字符串的数字即可作为标签,使用Python语言创建两个函数用来解码以及处理图片,由于训练的图片都是单通道的灰度图,因此仅需要定义一个函数,即可在训练集和测试集通用;

[0025] 在完成标签的制作以及图像读取函数之后,使用TensorFlow的Dataset创建用于

训练模型的数据集,将train文件夹里的图片路径和对应的标签写入train_image_ds变量中,将test文件夹里的图片路径和对应的标签写入test_image_ds变量中,最后通过map函数映射到每一个路径中,获得真正用于训练的训练集data_train以及测试集data_test;

[0026] S23、多分类模型构建:使用预训练神经网络MobileNetV3-Large进行分类的训练,MobileNet系列网络主要特点是效率高,可以在没有高端GPU的设备上运行,MobileNetV3-Large在ImageNet数据集分类上的准确度与MobileNetV2相比提高了3.2%,同时延迟降低了15%,因此用其进行图片分类的训练;使用MobileNetV3-Large时不包含顶层的全链接网络,而是使用自定义层,设置MobileNetV3-Large的权重值不变,权重值取公开数据集ImageNet训练下的权重,同时注意添加Dropout层以抑制过拟合;

[0027] S24、编译与训练:使用Adam优化器进行编译,Adam优化器是深度学习中最流行的优化器之一,Adam优化器结合了AdaGrad和RMSProp的优点,Adam对每个参数使用相同的学习率,并随着学习的进行而独立地适应。此外,Adam是基于动量的算法,利用了梯度的历史信息,并设置Adam优化器的学习速率为0.0005;再使用sparse_categorical_crossentropy作为交叉熵损失函数;在设置好优化器、损失函数等之后,对该神经网络进行训练;

[0028] 训练10000个epoch之后,设置MobileNetV3网络的前30层可以随着训练而修改,在此之前的网络权重仍然选择继承于imgnet数据集;

[0029] 此外,还需要将Adam优化器的学习速率降低10倍,即 $0.005/10=0.0005$,设置完成后,在前面epoch为10000的基础上,再训练15000步;

[0030] S25、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“model2.h5”文件中,在需要调用时,只需要使用load加载h5文件即可。

[0031] 优选的,所述S11中,因为目标是识别人与被子之间的覆盖关系,因此具体可将每一张图片分割为三个部分,分别为人、被子、其他(即背景),这样的分类方式既能有效实现被子覆盖情况的识别,也可以减少网络的计算量。

[0032] 优选的,所述S12中,在对图像进行随机裁剪增强的时候,具体为,先将原始图片以及对应的语义图片使用TensorFlow中的concat函数进行合并,再放缩为 280×280 尺寸,然后将合并并放缩的图片进行随机裁剪,得到的结果图片是 256×256 ,最后使用高级切片的方式,获得分开的原图和标签。

[0033] 优选的,所述S14中,在训练中引入IOU指标用来评价模型的鲁棒性,要先将判断的结果进行转换,因为直接用模型预测的结果是一个长度为3(每一个标签对应一个数值)的张量,值为预测的可能性,因此需要将其转换为具体的标签值,因此使用TensorFlow中的argmax获取预测中最大可能性数的索引,即预测的标签,用来与设定的标签进行比较。

[0034] (三)有益效果

[0035] 本发明提供了基于语义分割的睡眠被子覆盖率视觉识别方法,具备以下有益效果:

[0036] 本发明应用到摄像头等监控设备上,能够对人员的盖被情况进行准确的识别,从而能够使监护人员准确的了解被监控者的盖被情况,进而能够使监护人员在被监护者的被子离开身体一定程度后,能够及时的给监护人员预警,使监护人员及时的做出反应措施,从

而能够及时的避免被监护者被细菌或者病毒入侵,出现感冒发烧等情况。

附图说明

- [0037] 图1为语义分割示意图;
- [0038] 图2为单通道语义分割图;
- [0039] 图3为随机裁剪示意图;
- [0040] 图4为U-Net神经网络示意图;
- [0041] 图5为神经网络构建示意图;
- [0042] 图6为系统工作流程示意图。

具体实施方式

[0043] 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。

[0044] 实施例一

[0045] 本发明提供一种技术方案:基于语义分割的睡眠被子覆盖率视觉识别方法,包括以下步骤:

[0046] S1、语义分割模型训练

[0047] S11、语义分割图片的制作:通过爬虫以及人工采集等手段,获取约10000张图片,使用labelme工具进行语义图片的制作;因为目标是识别人与被子之间的覆盖关系,因此具体可将每一张图片分割为三个部分,分别为人、被子、其他(即背景),这样的分类方式既能有效实现被子覆盖情况的识别,也可以减少网络的计算量;制作语义分割图片的示意如图1所示,在语义分割图片中,人的RGB值为(128,0,0),覆盖物的RGB值为(0,128,0),背景的RGB值为(0,0,0);使用OpenCV将RGB值进行映射,将语义分割图片中的每一个RGB值分别进行处理,将背景RGB值保持不变,将人RGB值修改为(1,1,1),将覆盖物RGB值修改为(2,2,2),最终使用OpenCV进行通道分离,即处理得到灰度图,如图2所示;

[0048] S12、将原始图片以及语义分割图片分别放在image文件夹以及annotations文件夹中,使用glob函数获取所有原始图片以及语义分割图片的路径,分别存储在变量images以及anno中;

[0049] 创建seed随机数生成器种子,通过random函数的permutation命令获取原始图片数量相同长度的随机数组,作为索引值,进而获得乱序后的变量images以及anno,并且保证了images与anno中的每一个值都是一一对应的;获得乱序的变量images以及anno之后,使用TensorFlow中的from_tensor_slices对两者进行打包,获得数据集dataset。获取dataset中的20%作为测试集、80%作为训练集;

[0050] 定义图片的读取函数,由于原始图片以及语义图片都是png格式,并且都是RGB图片,通道数都为3,因此可以共用解码器,定义函数read_png,主通过TensorFlow中的decode_png函数将图像数据转换为像素点的数据用于训练;

[0051] 对图像进行随机裁剪增强,由于要求确保语义图片和原图保持一致进行裁剪,因

此先将语义图片和原图片进行绑定,再裁剪之后,使用numpy库高级切片的方式对其分割开,获得裁剪位置一致的训练图片以及语义图片,具体为,先将原始图片以及对应的语义图片使用TensorFlow中的concat函数进行合并,再放缩为 280×280 尺寸,然后将合并并放缩的图片进行随机裁剪,得到的结果图片是 256×256 ,最后使用高级切片的方式,获得分开的原图和标签,如图3所示;

[0052] 对原始图片和语义图片都进行归一化,归一化加快了梯度下降求最优解的速度,也即加快训练网络的收敛性,由于原始图片的RGB值为 $0 \sim 256$ 的数字,因此可以将其同时除以128再减1,将其映射到 $-1 \sim 1$,语义分割单通道图片不需要进行归一化处理;

[0053] 定义读取训练图片的函数,调用前面的读取函数获得原始图片以及语义图片,调用前面的随机裁剪图片得到裁剪的原始图片以及语义图片,再通过随机数的判断,将裁剪后的原始图片以及语义图片进行随机翻转,最后将其归一化处理;

[0054] 定义读取测试图片的函数,对于测试集的图片,不需要进行随机裁剪以及随机翻转等图像增强操作,只需要将其放缩到 256×256 大小,并进行归一化处理即可;

[0055] 定义完读取训练图片的函数以及读取测试图片的函数之后,使用map函数将其分别映射到训练数据集以及测试数据集中,完成数据集的制作;

[0056] S13、构建U-Net模型:使用Python面向对象的编程方式分别定义上采样类和下采样类进行组装,构建如图4所示的U-Net网络;

[0057] S14、编译与训练:在定义模型之后,使用Adam优化器进行优化,在优化器中设置学习速率为0.0001,并在损失函数将优化器激活;在训练中引入IOU指标用来评价模型的鲁棒性,要先把判断的结果进行转换,因为直接用模型预测的结果是一个长度为3(每一个标签对应一个数值)的张量,值为预测的可能性,因此需要将其转换为具体的标签值,因此使用TensorFlow中的argmax获取预测中最大可能性数的索引,即预测的标签,用来与设定的标签进行比较;训练5000步,记录每一步训练中训练集的损失值、准确率以及IOU值,和测试集的损失值、准确率以及IOU值,以便于神经网络模型的后续调整;

[0058] S15、模型的保存: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"model1.h5"文件中,在需要调用时,只需要使用load加载h5文件即可;

[0059] S2、分类模型的训练

[0060] S21、语义图片的分类:将单通道语义分割单通道图片作为训练集进行训练,具体按照覆盖面积将其分为5类,其中一种为未检测到,另外四类分别为:

[0061] 分别定义标签 $0 \sim 4$,标签0表示未检测到,标签1表示覆盖率约90%,标签2表示覆盖率约65%~85%,标签3表示覆盖率约30%~60%,标签4表示覆盖率低于30%,将使用labelme以及opencv处理得到单通道语义图片根据上面所述的分类标准,分为5类,在用于训练的文件夹"train"、用于测试的文件夹"test"里,分别创建"0"、"1"、"2"、"3"、"4"这五个文件夹,文件夹名则对应着每一个分类的标签;

[0062] S22、数据集的制作:首先使用python的glob库获取图片的地址,将图片的路径分别存在变量"test_image_path"以及"train_image_path"中,这两个变量分别代表着测试集所有图片的路径以及训练集所有的图片路径,通过random库的shuffle函数,打乱"test_image_path"以及"train_image_path"这两个变量里的数据;

[0063] 使用split函数截取文件名字符串的数字即可作为标签,使用Python语言创建两个函数用来解码以及处理图片,由于训练的图片都是单通道的灰度图,因此仅需要定义一个函数,即可在训练集和测试集通用;

[0064] 在完成标签的制作以及图像读取函数之后,使用TensorFlow的Dataset创建用于训练模型的数据集,将train文件夹里的图片路径和对应的标签写入train_image_ds变量中,将test文件夹里的图片路径和对应的标签写入test_image_ds变量中,最后通过map函数映射到每一个路径中,获得真正用于训练的训练集data_train以及测试集data_test;

[0065] S23、多分类模型构建:使用预训练神经网络MobileNetV3-Large进行分类的训练;使用MobileNetV3-Large时不包含顶层的全链接网络,而是使用自定义层,设置MobileNetV3-Large的权重值不变,权重值取公开数据集ImageNet训练下的权重,同时注意添加Dropout层以抑制过拟合,最终构建的网络框架如图5所示;

[0066] S24、编译与训练:使用Adam优化器进行编译,并设置Adam优化器的学习速率为0.0005;再使用sparse_categorical_crossentropy作为交叉熵损失函数;在设置好优化器、损失函数等之后,对该神经网络进行训练;

[0067] 训练10000个epoch之后,设置MobileNetV3网络的前30层可以随着训练而修改,在此之前的网络权重仍然选择继承于imgnet数据集;

[0068] 此外,还需要将Adam优化器的学习速率降低10倍,即 $0.005/10=0.0005$,设置完成后,在前面epoch为10000的基础上,再训练15000步;

[0069] S25、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“model2.h5”文件中,在需要调用时,只需要使用load加载h5文件即可。

[0070] 实施例二

[0071] 本发明设计了一种基于树莓派的硬件系统。具体包括图像信息采集模块、交互平台模块、信号传输模块。具体执行流程如图6所示。

[0072] 图像信息采集模块包括相机板、树莓派摄像头、二维舵机云台;相机板使用电缆分别连接树莓派上的CSI端口、树莓派摄像头、二维舵机云台,二维舵机云台的电源端口以及控制端口均连接至树莓派的GPIO接口,Arduino UNO开发板通过USB线与树莓派连接。

[0073] 交互平台模块包括显示屏、数字按键。通过Qt设计用户界面,用户可以通过按键来预设触发条件,例如设置在三十分钟内每次检测或者检测指定次数都是“2”标签(即覆盖率65%~85%),在设定好出触发条件之后,将激活图像信息采集模块进行工作,定时获取图片信息并调用训练好的网络模型进行预测。

[0074] 信号传输模块主要采用蓝牙信号进行传输,通过蓝牙信号,将识别的结果与预设的条件进行比对,若满足预设条件,则通过蓝牙信号与终端(例如智能手环、智能手机等)进行通讯,使其振动用于提醒。所拍摄的画面将会被转换为Bgr格式,和对应的识别结果一并存储在MySQL数据库中,方便维护人员进行调试。

[0075] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖

非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。

[0076] 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

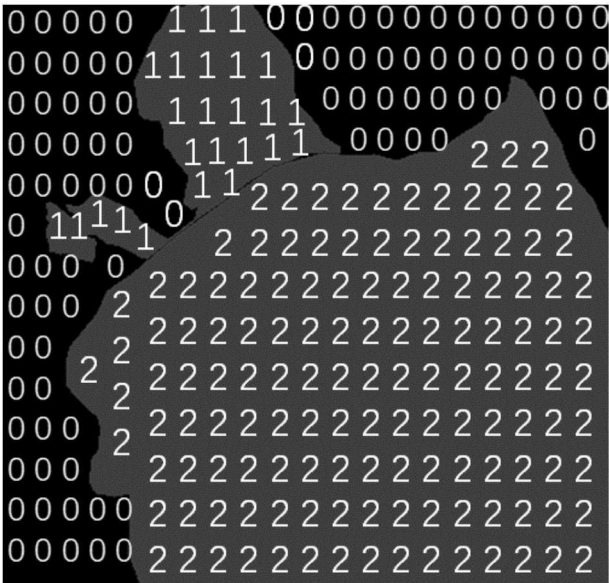


图2

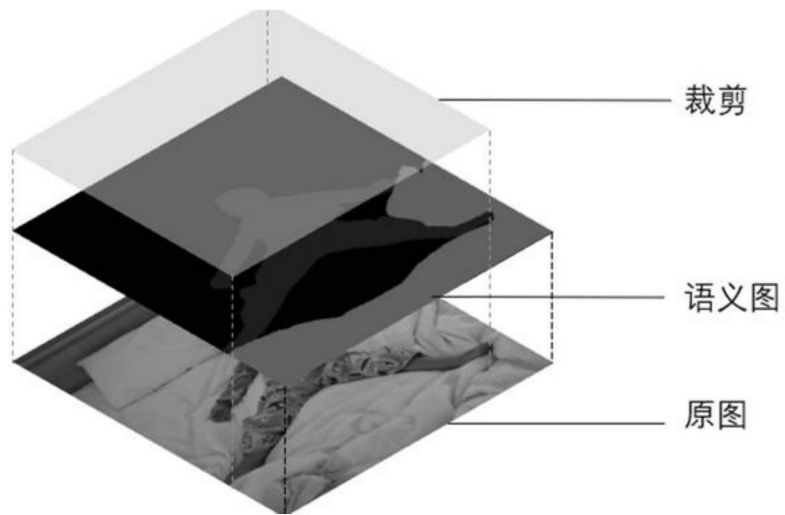


图3

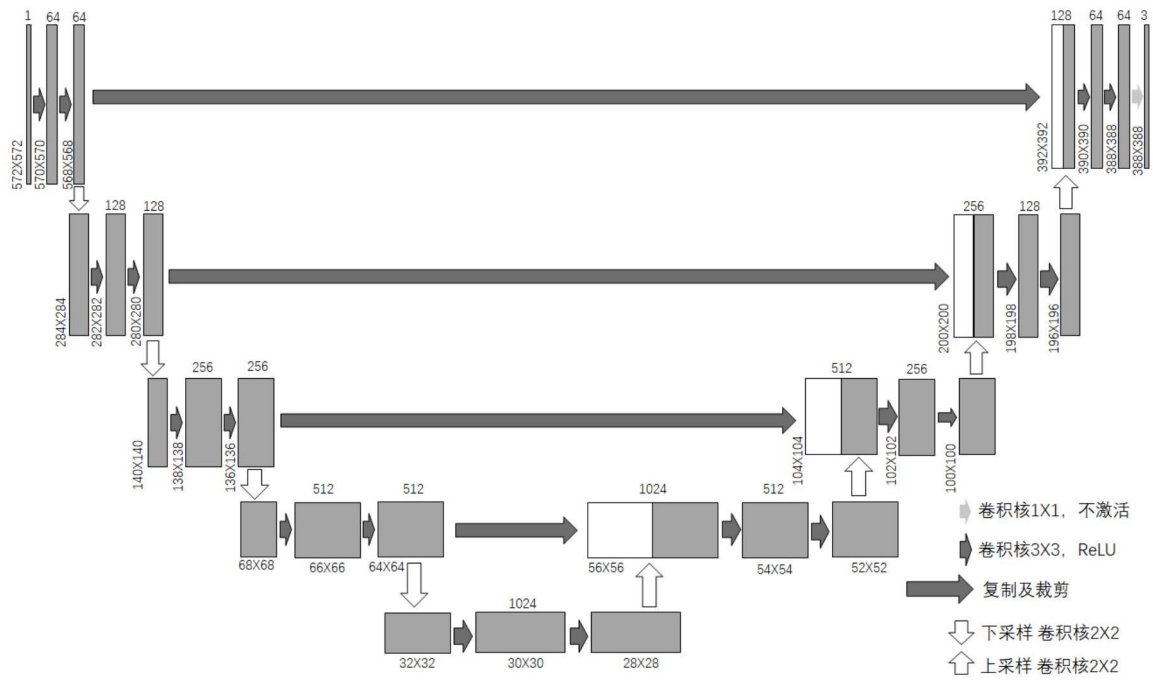


图4

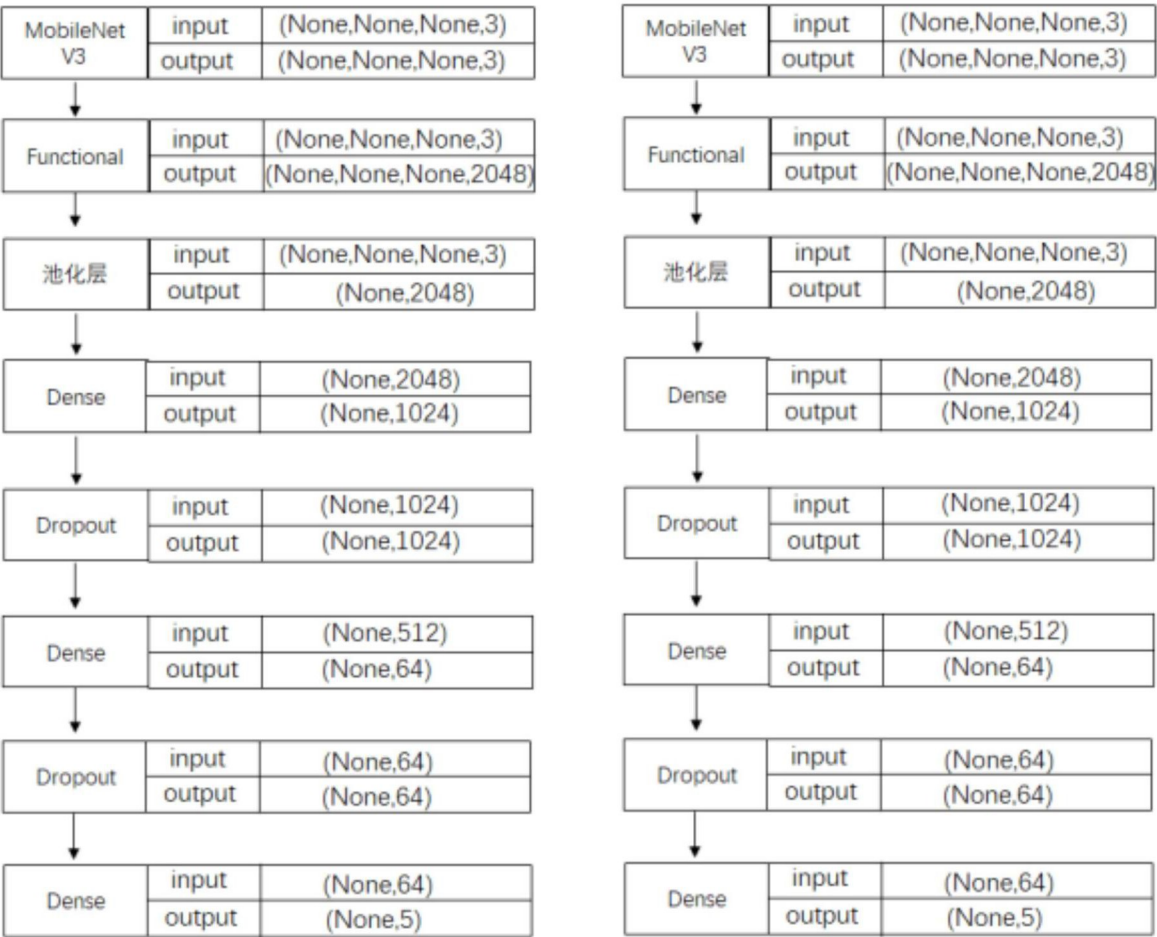


图5

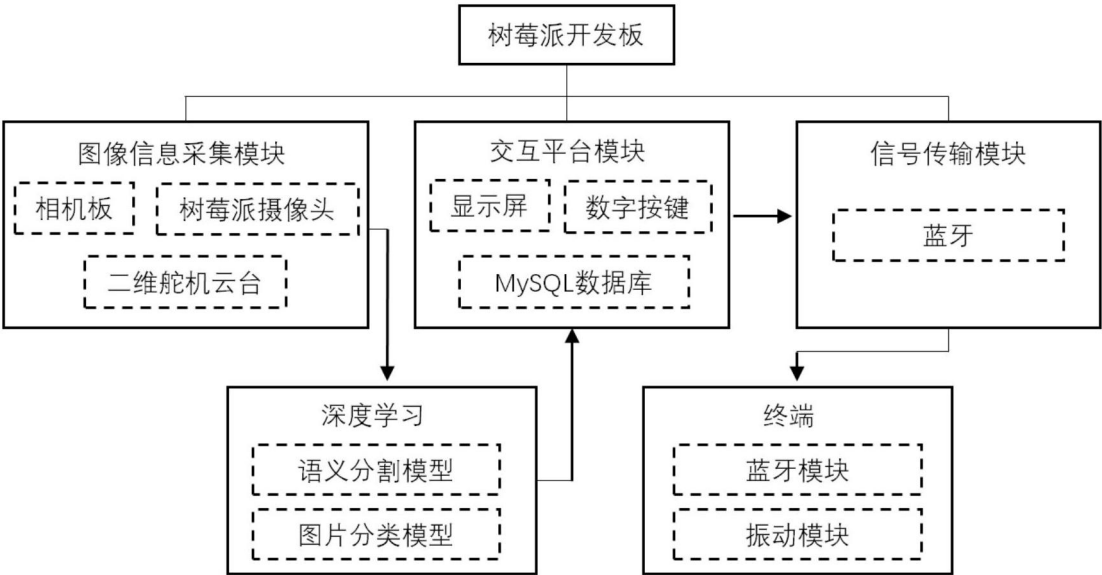


图6