



(12) 发明专利申请

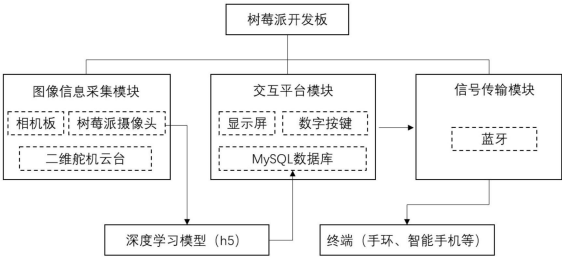
(10) 申请公布号 CN 116416473 A
(43) 申请公布日 2023. 07. 11

(21) 申请号 202310402050.1
(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 闫坤 陈晓满
(51) Int.Cl.
G06V 10/764 (2022.01)
G06V 10/774 (2022.01)
G06V 10/82 (2022.01)
G06N 3/096 (2023.01)
G06N 3/045 (2023.01)

权利要求书2页 说明书5页 附图8页

(54) 发明名称
基于迁移学习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,且公开了基于迁移学习的睡眠被子覆盖率视觉识别方法,包括以下步骤:S1、数据集制作:S11、图片的收集及分类:创建两个文件夹"train"、"test",分别代表训练集和测试集,训练集和测试集的图片数量根据一般的深度学习比例选择8:2;将被子覆盖率的识别结果分为5种。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. 基于迁移学习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、数据集制作

S11、图片的收集及分类:创建两个文件夹"train"、"test",分别代表训练集和测试集;将盖被子的情况分为5种,其中一种为未检测到,另外四种分别为被子覆盖率约90%、被子覆盖率约65%-85%、被子覆盖率约30%-60%、被子覆盖率低于30%,将未检测到以及另外四种情况分别定义标签为0、1、2、3、4;将收集到的12000张图片根据以上的分类标准,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"五个文件夹,文件夹名则对应每一个分类的标签;通过人工筛选每一张图片的分类,将这些图片放置在上述的十个文件夹中,每个分类下约1200张图片;

S12、标签的制作: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测试集和训练集所有图片的路径;再利用random库的shuffle函数将"test_image_path"以及"train_image_path"中变量中的数据打乱;再使用split函数截取文件名字符串的数字作为标签;

S13、数据的加载:使用Python创建load_train_image和load_test_image函数来解码以及处理图片;

S14、数据集制作:在完成标签的制作以及图像读取函数后,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;

S2、搭建网络:本发明使用的ResNet50不包含顶层的全链接网络,而是使用自定义层,本发明的最终输出是五个分类的训练结果,因此需要对网络结果进行调整;在预训练神经网络ResNet后面,添加自己的一些网络层;在ResNet50后面接入之前的层之前,设置ResNet50的权重值不变,权重值取公开数据集ImageNet训练下的权重;同时添加Dropout层,Dropout是通过遍历神经网络每一层的节点,然后通过对该层的神经网络设置一个keep_prob(节点保留概率),即该层的节点有keep_prob的概率被保留,keep_prob的取值范围在0到1之间;通过设置神经网络该层节点的保留概率,使得神经网络不会去偏向于某一个节点,从而使得每一个节点的权重不会过大,有利于来减轻神经网络的过拟合;

S3、编译与训练:使用Adam优化器来进行编译,并采用sparse_categorical_crossentropy作为交叉熵损失函数,在设置好优化器、损失函数之后,对该神经网络进行训练;

S4、微调模型:首先,设置ResNet50网络的权重可变的范围,设置ResNet50网络后40层的权重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ImageNet数据集;此外,将Adam优化器的学习速率降低10倍;设置完成后,在前面epoch为100的基础上,再训练150步;

S5、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model中,但此时只在计算机内存里;通过save方法,将网络的框架以及权重全都保存到"model.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所述的基于迁移学习的睡眠被子覆盖率视觉识别方法,其特征在于:

所述S11中,训练集和测试集的图片数量根据一般的深度学习比例选择8:2。

3.根据权利要求1所述的基于迁移学习的睡眠被子覆盖率视觉识别方法,其特征在于:所述S13中的两个函数的流程相似,区别在于记载训练集的图像会进行图像增强,而加载测试集则不需要,在对图像增强的处理过程中,首先是随机裁剪,输入网络的图片大小使 $224 \times 224 \times 3$ 的彩色图片,但是在load_train_image中是先将图片统一用resize方法处理成 $256 \times 256 \times 3$ 的大小,然后再用TensorFlow框架中的random_crop方法实现随机裁剪;通过在load_train_image中添加random_flip_left_right和random_flip_up_down这两个api即可实现随左右翻转和随机上下翻转;此外,在TensorFlow框架下,使用random_brightness以及random_contrast对图片给进行亮度以及对比度的调整。

4.根据权利要求1所述的基于迁移学习的睡眠被子覆盖率视觉识别方法,其特征在于:所述S14中,完成训练集data_train以及测试集data_test的制作后,设置BATCH_SIZE为8,也就是每次取出8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;由于需要返回训练,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。

5.根据权利要求1所述的基于迁移学习的睡眠被子覆盖率视觉识别方法,其特征在于:所述S3中,在Adam优化器中,设置学习速率为0.0005。

基于迁移学习的睡眠被子覆盖率视觉识别方法

技术领域

[0001] 本发明涉机器视觉技术领域,具体为基于迁移学习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,通常会在孩童的卧式中安装摄像头,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,本发明提供了基于迁移学习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:基于迁移学习的睡眠被子覆盖率视觉识别方法,包括以下步骤:

[0007] S1、数据集制作

[0008] S11、图片的收集及分类:创建两个文件夹"train"、"test",分别代表训练集和测试集,训练集和测试集的图片数量根据一般的深度学习比例选择8:2;将盖被子的情况分为5种,其中一种为未检测到,另外四种分别为被子覆盖率约90%、被子覆盖率约65%-85%、被子覆盖率约30%-60%、被子覆盖率低于30%,将未检测到以及另外四种情况分别定义标签为0、1、2、3、4;将收集到的12000张图片根据以上的分类标准,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"五个文件夹,文件夹名则对应每一个分类的标签;通过人工筛选每一张图片的分类,将这些图片放置在上述的五个文件夹中,每个分类下约1200张图片;

[0009] S12、标签的制作: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测试集和训练集所有图片的路径;由于迁移学习具有极强的拟合能力,如果不乱序的话,同一个组合的batch反复出现,模型有可能会"记住"这些样本的次序,会影响泛化能力,也就是模型的实际预测能力,随机打乱后的一段样本比排序后的更加接近真实分布,排序后多了一层相关性,一定程度上表现出不同的概率分布,因此再利用random库的shuffle函数将"test_image_path"以及"train_image_path"中变量中的数据打乱;接着是制作标签,这里制作标签非常容易,因为每个文件名的命名就是对应着实际的标签,因此,只需要使用split函数

截取文件名字符串的数字即可作为标签,使用split函数截取文件名字符串的数字作为标签;至此完成标签的制作,由于标签来源于路径中的字符串,因此可以保证标签与图片的一一对应,这样可以保证训练的有效性;

[0010] S13、数据的加载:使用Python创建load_train_image和load_test_image函数来解码以及处理图片;

[0011] S14、数据集制作:在完成标签的制作以及图像读取函数后,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;

[0012] S2、搭建网络:本发明使用的ResNet50不包含顶层的全链接网络,而是使用自定义层,默认的ResNet网络使五十个分类,而本发明的最终输出是三个分类的训练结果,因此需要对网络结果进行调整;在预训练神经网络ResNet后面,协商自己的一些网络层;在ResNet50后面接入之前的层之前,设置ResNet50的权重值不便,权重值取公开数据集imagenet训练下的权重;同时添加Dropout层,Dropout是通过遍历神经网络每一层的节点,然后通过对该层的神经网络设置一个keep_prob(节点保留概率),即该层的节点有keep_prob的概率被保留,keep_prob的取值范围在0到1之间;通过设置神经网络该层节点的保留概率,使得神经网络不会去偏向于某一个节点,从而使得每一个节点的权重不会过大,有利于来减轻神经网络的过拟合;

[0013] S3、编译与训练:使用Adam优化器来进行编译,在Adam优化器中,设置学习速率为0.0005,一般默认的学习速率为0.001,考虑到训练的规模是小数据集,使用更小的学习速率有利于获得更好的学习效果,并采用sparse_categorical_crossentropy作为交叉熵损失函数,在设置好优化器、损失函数之后,对该神经网络进行训练;

[0014] S4、微调模型:首先,设置ResNet50网络的权重可变的范围,设置ResNet50网络后40层的权重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ImageNet数据集;此外,将Adam优化器的学习速率降低10倍;设置完成后,在前面epoch为100的基础上,再训练150步;

[0015] S5、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"model.h5"文件中;在需要调用时,只需要使用load加载h5文件即可。

[0016] 优选的,所述S13中的两个函数的流程相似,区别在于记载训练集的图像会进行图像增强,而加载测试集则不需要,在对图像增强的处理过程中,首先是随机裁剪,输入网络的图片大小使 $224 \times 224 \times 3$ 的彩色图片,但是在load_train_image中是先将图片统一用resize方法处理成 $256 \times 256 \times 3$ 的大小,然后再用tensorflow框架中的random_crop方法实现随机裁剪;通过在load_train_image中添加random_flip_left_right和random_flip_up_down这两个api即可实现随左右翻转和随机上下翻转;此外,在TensorFlow框架下,使用random_brightness以及random_contrast对图片给进行亮度以及对比度的调整。

[0017] 优选的,所述S14中,完成训练集data_train以及测试集data_test的制作后,设置BATCH_SIZE为8,也就是每次取出8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;由于需要返回训练,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。

[0018] (三)有益效果

[0019] 本发明提供了基于迁移学习的睡眠被子覆盖率视觉识别方法,具备以下有益效果:

[0020] 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

[0021] 图1为被子不同的覆盖率情况示意图;

[0022] 图2为数据集分布示意图;

[0023] 图3为load_train_image处理流程图;

[0024] 图4为load_test_image处理流程图;

[0025] 图5为随机裁剪示意图;

[0026] 图6为随机上下翻转和随机左右翻转示意图;

[0027] 图7为数据集制作过程示意图;

[0028] 图8为是神经网络的构建示意图;

[0029] 图9为训练集与测试集的准确率及loss值结果图 (epoch=100);

[0030] 图10为训练集与测试集的准确率及loss值结果图 (epoch=250);

[0031] 图11为系统工作流程示意图。

具体实施方式

[0032] 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。

[0033] 实施例一

[0034] 本发明提供一种技术方案:基于迁移学习的睡眠被子覆盖率视觉识别方法,包括以下步骤:

[0035] S1、数据集制作

[0036] S11、图片的收集及分类:创建两个文件夹"train"、"test",分别代表训练集和测试集;将盖被子的情况分为5种,其中一种为未检测到,另外四种分别如图1所示,将未检测到以及另外四种情况分别定义标签为0、1、2、3、4;将收集到的12000张图片根据以上的分类标准,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"五个文件夹,文件夹名则对应每一个分类的标签,数据集图片的具体分布情况如图2所示;通过人工筛选每一张图片的分类,将这些图片放置在上述的十个文件夹中,每个分类下约1200张图片;

[0037] S12、标签的制作：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测试集和训练集所有图片的路径；再利用random库的shuffle函数将“test_image_path”以及“train_image_path”中变量中的数据打乱；再使用split函数截取文件名字符串的数字作为标签；

[0038] S13、数据的加载：使用Python创建load_train_image和load_test_image函数来解码以及处理图片，两个函数的流程相似，区别在于记载训练集的图像会进行图像增强，而加载测试集则不需要，load_train_image流程如图3所示，load_test_image的流程如图4所示，在对图像增强的处理过程中，首先是随机裁剪，输入网络的图片大小使 $224 \times 224 \times 3$ 的彩色图片，但是在load_train_image中是先将图片统一用resize方法处理成 $256 \times 256 \times 3$ 的大小，然后再用tensorflow框架中的random_crop方法实现随机裁剪，如图5所示；通过在load_train_image中添加random_flip_left_right和random_flip_up_down这两个api即可实现随左右翻转和随机上下翻转，如图6所示；此外，在TensorFlow框架下，使用random_brightness以及random_contrast对图片给进行亮度以及对比度的调整；

[0039] S14、数据集制作：在完成标签的制作以及图像读取函数后，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，如图7所示；

[0040] 完成训练集data_train以及测试集data_test的制作后，设置BATCH_SIZE为8，也就是每次取出8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；由于需要返回训练，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；

[0041] S2、搭建网络：本发明使用的ResNet50迁移学习模型不包含顶层的全链接网络，而是使用自定义层，本发明的最终输出是五个分类的训练结果，因此需要对网络结构进行调整；在预训练神经网络ResNet50后面，添加自己的一些网络层；在ResNet50后面接入自定义层之前，设置ResNet50的权重值不便，权重值取公开数据集ImageNet训练下的权重；同时添加Dropout层，Dropout是通过遍历神经网络每一层的节点，然后通过对该层的神经网络设置一个keep_prob（节点保留概率），即该层的节点有keep_prob的概率被保留，keep_prob的取值范围在0到1之间；通过设置神经网络该层节点的保留概率，使得神经网络不会去偏向于某一个节点，从而使得每一个节点的权重不会过大，有利于来减轻神经网络的过拟合，最终构建的网络如图8所示；

[0042] S3、编译与训练：使用Adam优化器来进行编译，设置Adam的学习速率为0.0005，并采用sparse_categorical_crossentropy作为交叉熵损失函数，在设置好优化器、损失函数之后，对该神经网络进行训练，训练的准确率以及loss变化值如图9所示；

[0043] S4、微调模型：上述模型，其主体框架适用的是ResNet50，并且网络权重是基于ImageNet数据集保存下来的。训练的结果并不能达到满意的效果，在epoch为100的范围里，训练集以及测试集的准确率基本维持在55%上下。为了使得网络更加适合于本发明的数据集，需要对ResNet50网络进行微调。

[0044] 首先,设置ResNet50网络的权重可变的范围,设置ResNet50网络后40层的权重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ImageNet数据集;此外,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100的基础上,再训练150步,微调之后的训练准确率以及IOSS值如图10所示;

[0045] S5、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,但此时只在计算机内存里;通过save方法,将网络的框架以及权重全都保存到“model.h5”文件中;在需要调用时,只需要使用load加载h5文件即可。

[0046] 实施例二

[0047] 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11所示。

[0048] 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。

[0049] 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。

[0050] 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Bgr24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。

[0051] 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。

[0052] 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。

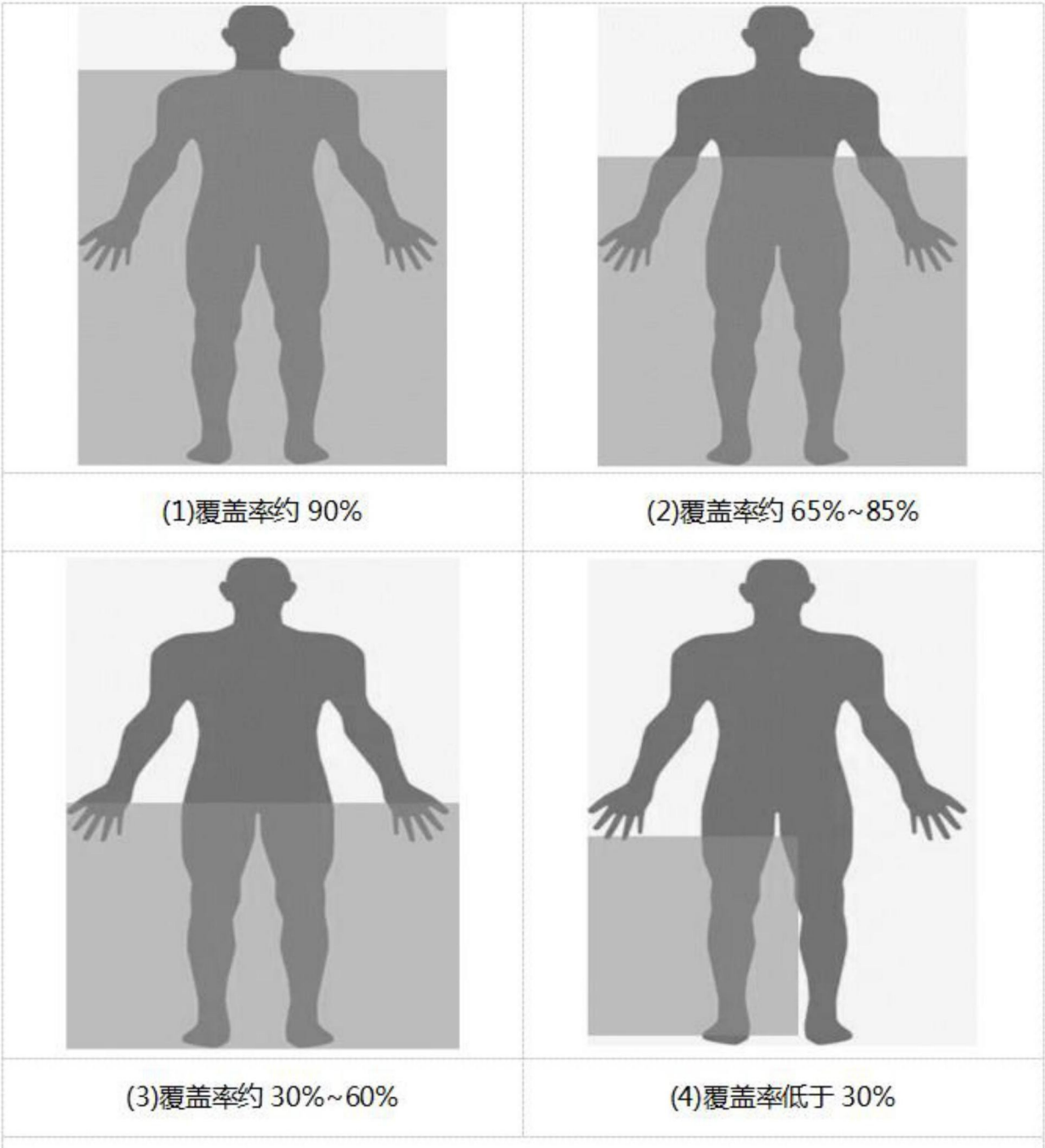


图1

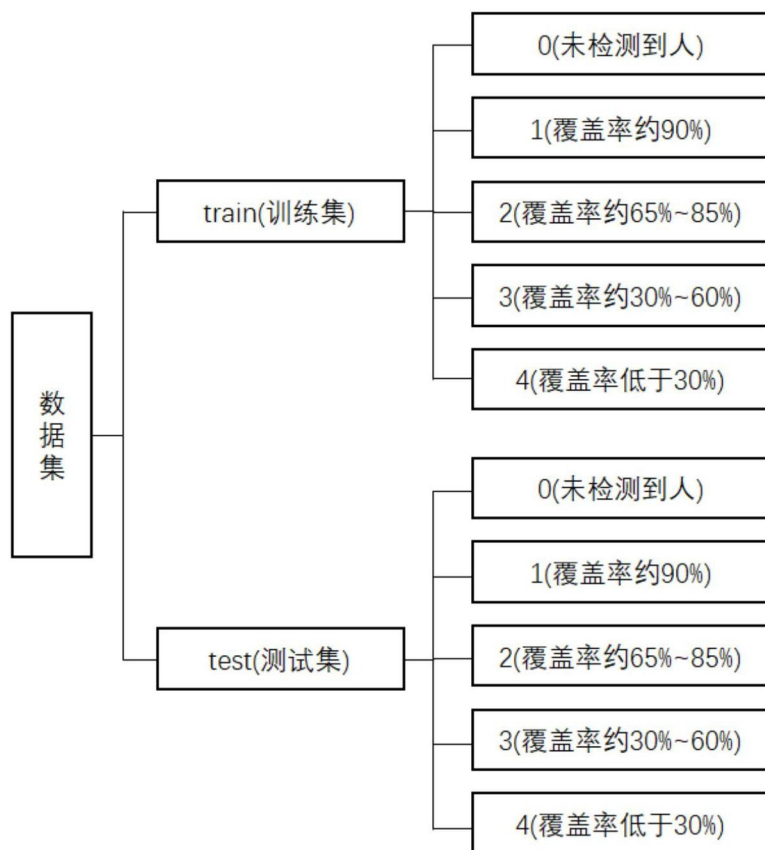


图2

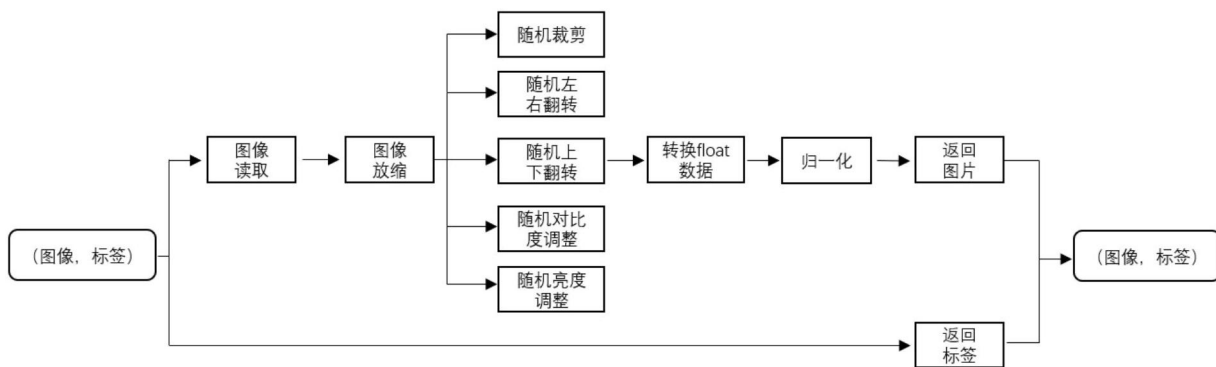


图3

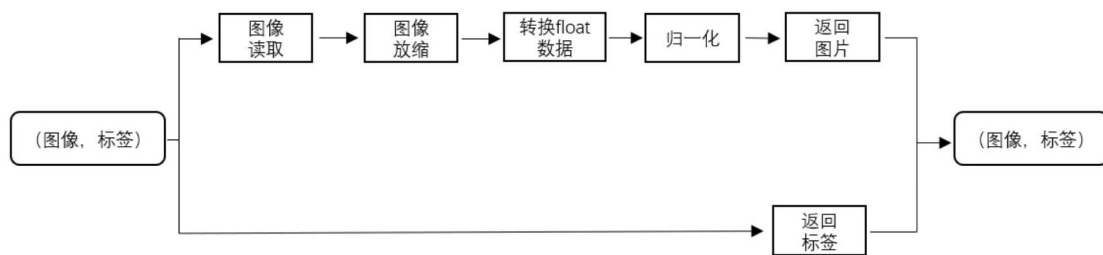


图4

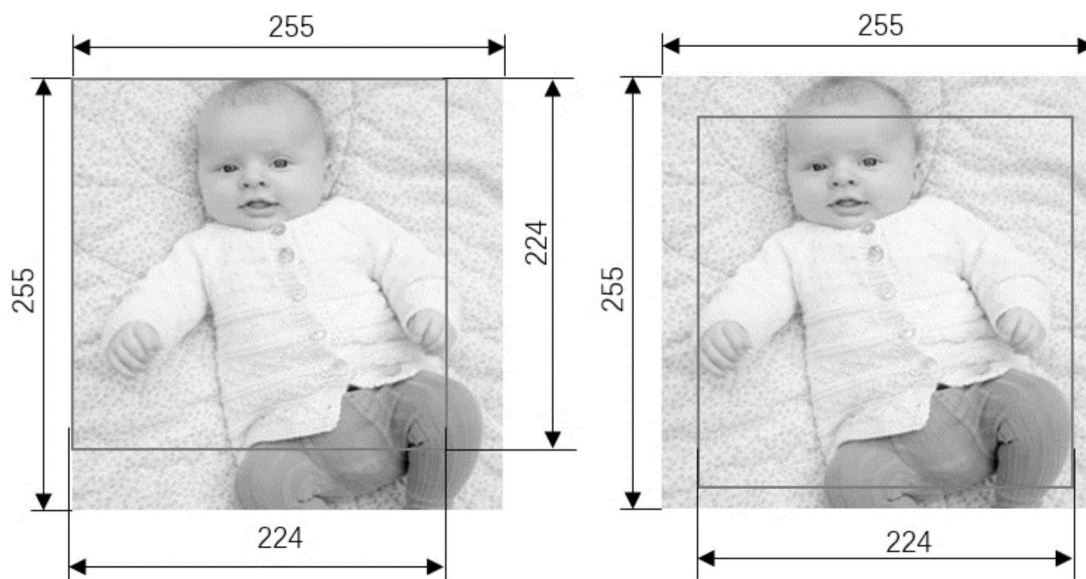


图5



图6

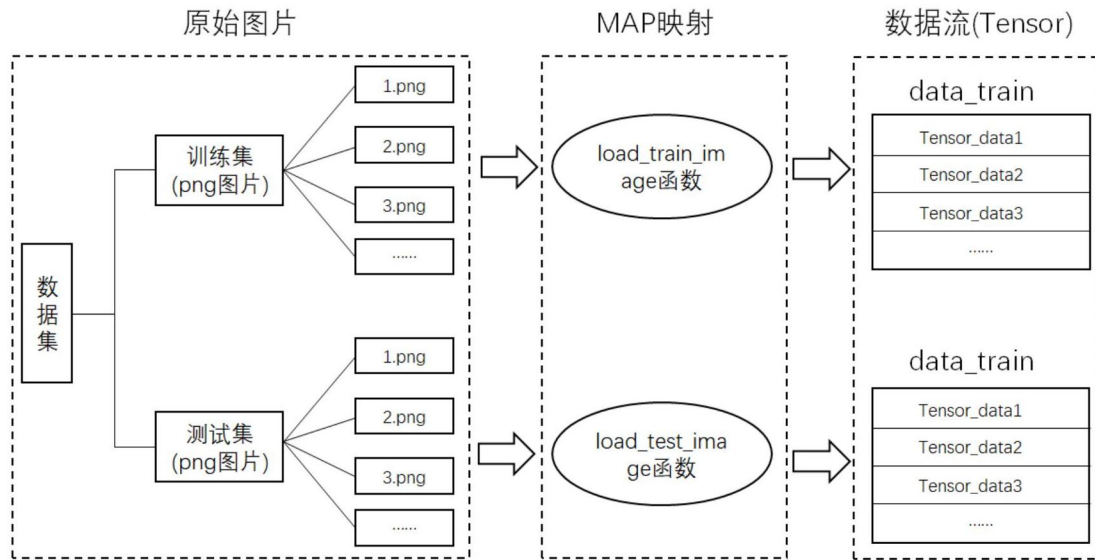


图7

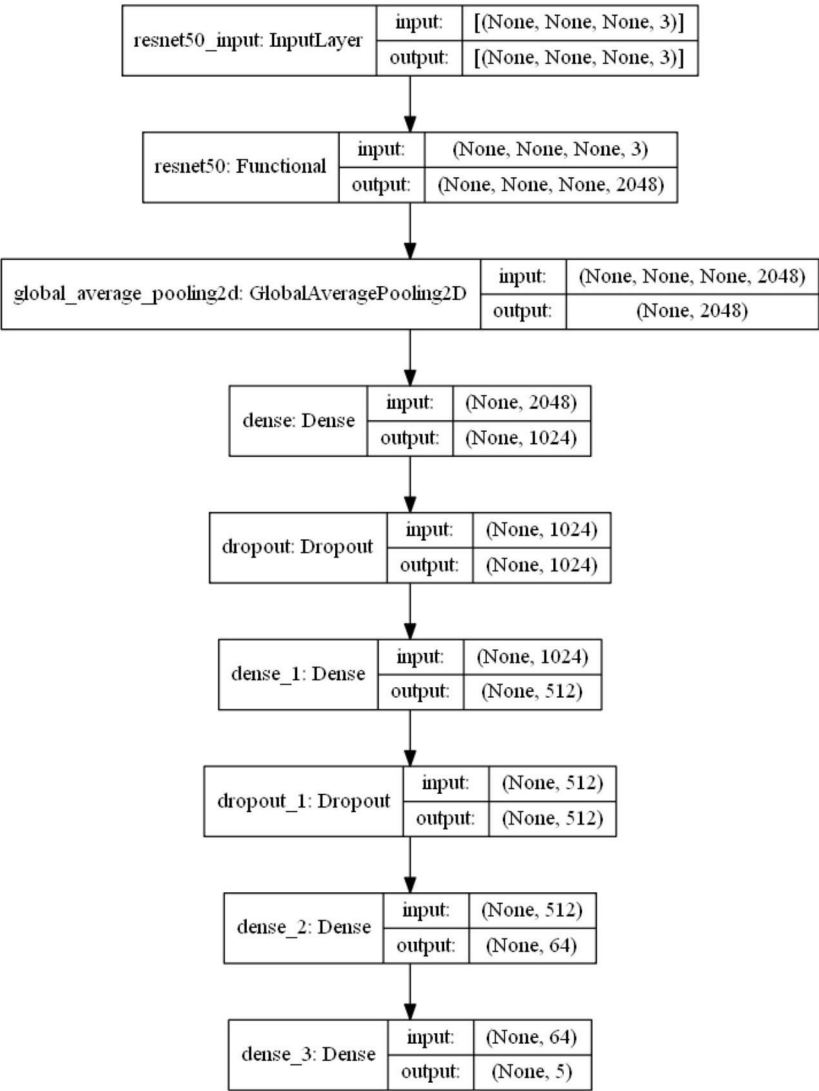


图8

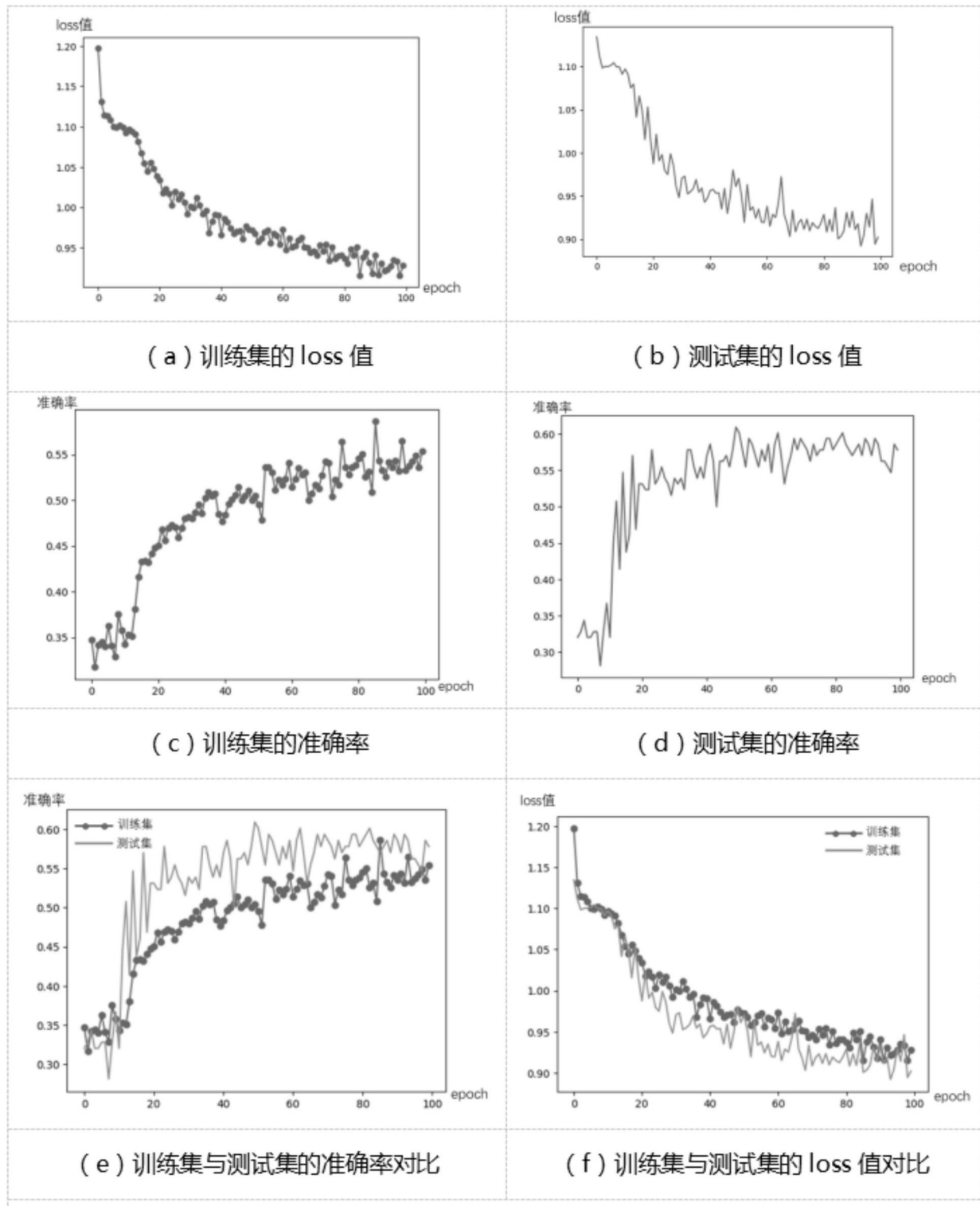


图9

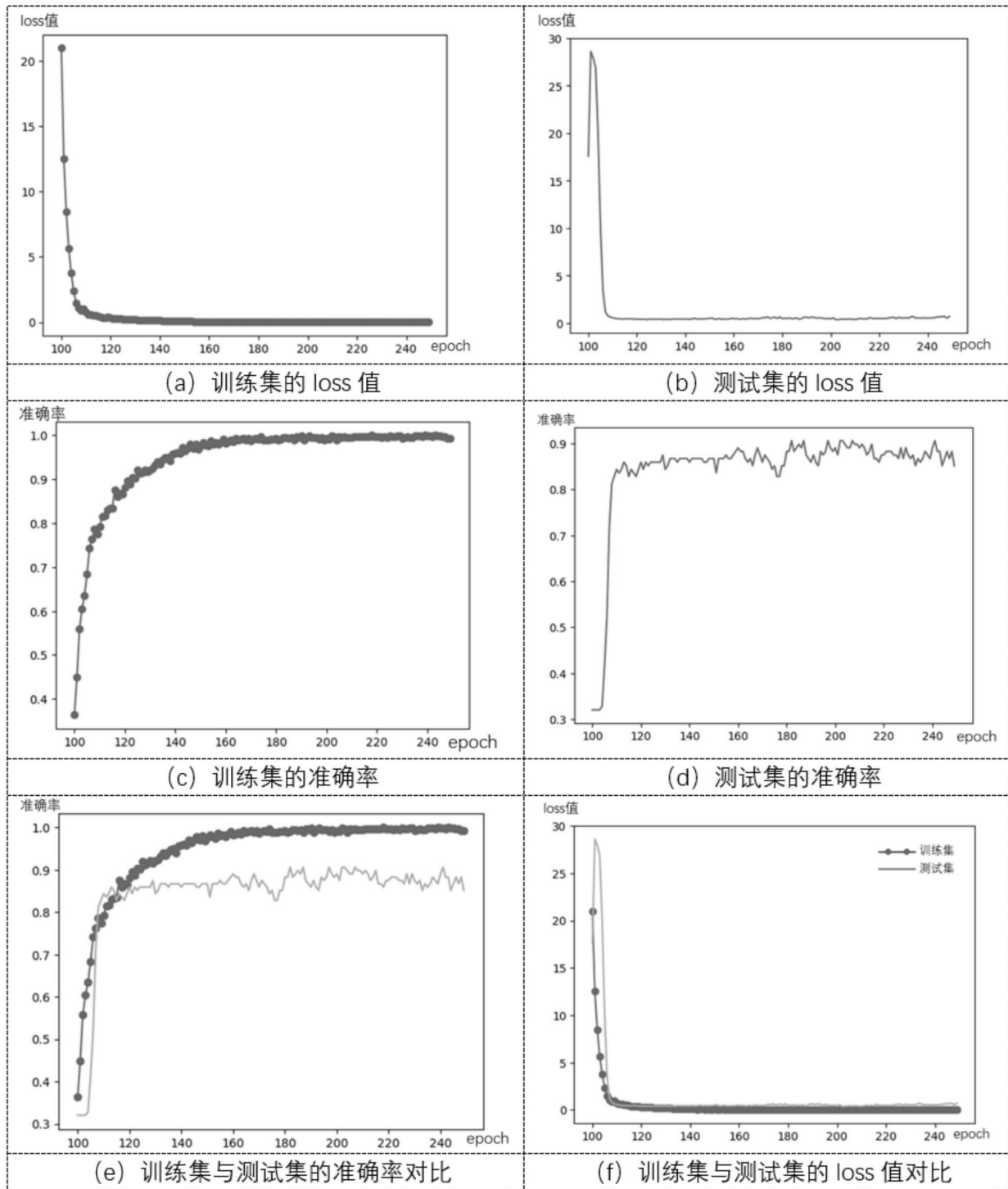


图10

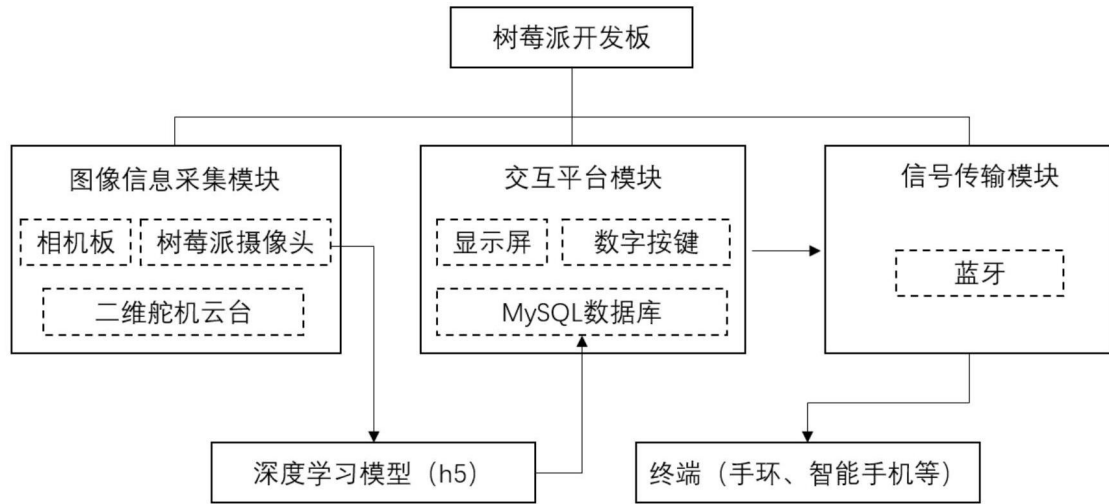


图11