



(12) 发明专利申请

(10) 申请公布号 CN 112686340 A

(43) 申请公布日 2021.04.20

(21) 申请号 202110269319.4

G06N 3/12 (2006.01)

(22) 申请日 2021.03.12

(71) 申请人 成都点泽智能科技有限公司

地址 610000 四川省成都市自由贸易试验区成都高新区天仁路388号1栋4单元15层1501号

申请人 中科智云科技有限公司

(72) 发明人 陈倩 严安 周治尹

(74) 专利代理机构 成都君合集专利代理事务所
(普通合伙) 51228

代理人 尹新路

(51) Int. Cl.

G06K 9/62 (2006.01)

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

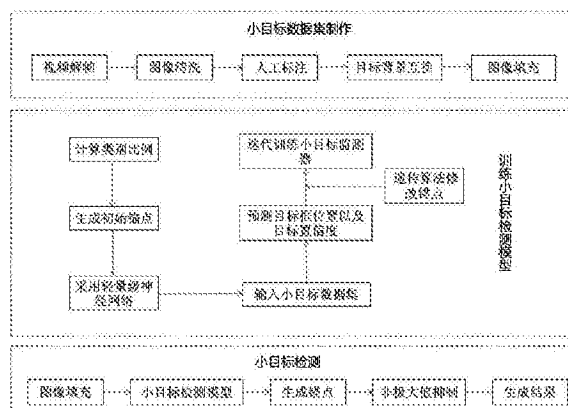
权利要求书3页 说明书11页 附图11页

(54) 发明名称

一种基于深度神经网络的密集小目标检测方法

(57) 摘要

本发明提出了一种基于深度神经网络的密集小目标检测方法,按照目标的类别的方式对目标进行锚点划分比例进行计算,采用目标的类别的方式对目标进行锚点划分比例进行计算,并采用k-means聚类,获得各类初始锚点分配,避免因为各个类别的目标数量不平衡,离群值造成锚点干扰的问题;此种分配情况可以精确反应出小目标的细微变化,有效表征训练集中的小物体,使得模型容易收敛,在提高了检测精度和深度的同时,还可增加检测速度,实现实时检测;解决了现有技术对于小目标的检测无法做到对深度和检测速度的兼顾,不能很好预测小目标的位置从而影响检测精度,并且兼顾实时检测的缺点。



该发明用于何处+该发明的基本原理+该发明带来的好处即改善的问题

1. 一种基于深度神经网络的密集小目标检测方法, 其特征在于, 包括以下步骤:

步骤1: 搭建摄像头, 并采集图像; **首先都是说明数据集, 在收集到数据集的基础上, 再进一步阐述过程。**

步骤2: 对采集的图像进行预处理得到图像集, 并将图像集划分为用于预训练的预训练集、用于验证的验证集以及用于测试的测试集; 所述预处理包括对采集到的图像的待检测目标进行标注, 并为每一张采集的图像中的待检测目标生成一个xml标注文件, 所述标注文件中包括各类别的待检测目标的基础信息;

步骤3: 对预训练集中的图像的目标进行计算, 首先计算出预训练集中所有的目标的数量, 然后根据分类计算出各个类别的目标占据了所有的目标的总数的比例;

步骤4: 对预训练集中的图像, 在得到各个类别的目标占据所有目标的总数的比例后, 按照所述比例对各个类别的目标分别进行k-means聚类, 进而得到各个类别的初始锚点的初始锚点分配比例;

步骤5: 按照各个类别的初始锚点分配比例, 使用yolo-tiny框架结合遗传算法进行模型训练; **使用某某结合算法进行模型的训练**

步骤6: 使用训练后的模型进行目标检测。

2. 如权利要求1所述的一种基于深度神经网络的密集小目标检测方法, 其特征在于, 所述yolo-tiny框架为改进后的yolo-tiny框架, 改进后的yolo-tiny框架具体结构依次包括: 0conv层、1max层、2conv层、3max层、4conv层、5max层、6conv层、7max层、8conv层、9max层、10conv层、11max层、12conv层、13conv层、14conv层、15conv层、16yolo层、17route层、18conv层、19upsample层、20route层、21conv层、22conv层、23yolo层、24route层、25conv层、26upsample层、27route层、28conv层、29conv层、30yolo层;

在所述步骤5中, 使用改进后的yolo-tiny框架将一张目标图像划分为 24×24 、 48×48 和 96×96 三种网格形式, 采用SGD随机优化算法进行网络训练; 并分别对应经过16yolo层、23yolo层和30yolo层这三个检测层输出 $24 \times 24 \times Q$ 、 $48 \times 48 \times Q$ 和 $96 \times 96 \times Q$ 三个结果, 在三个结果中, 24×24 、 48×48 和 96×96 分别代表对应的检测层的大小, Q 的计算公式如下:

$$Q = (4 + 1 + M) \times O;$$

其中, M 为预训练集中的图像类别个数, O 为检测层上每一个点对应的锚点个数, 此处所述锚点个数与步骤4中的初始锚点的个数对应, 式中4代表的是训练时的框位置信息 $\{x, y, w, h\}$; 1表示置信度;

所述框位置信息 $\{x, y, w, h\}$ 为图像检测时产生的检测框的位置信息, 其中 x, y 为检测框左上角的坐标值, w, h 分别表示检测框的宽度和高度。

3. 如权利要求2所述的一种基于深度神经网络的密集小目标检测方法, 其特征在于, 将各类别的目标根据占据图像中的尺寸比例分为大目标和小目标, 在步骤5中采用16yolo层即 24×24 网格形式的检测层对大目标的类别进行检测, 另外两种网格形式的检测层对小目标的类别进行检测; 且在所述步骤4中, 对于大目标的类别的初始锚点的分配不低于初始锚点总数的三分之一; 即当大目标的类别占有所有目标的总数的三分之一以上时, 按照各个类别的目标的数量占比进行分配, 而当大目标的类别占有所有目标的总数的三分之一以下时, 分配所有初始锚点的三分之一作为大目标的类别的初始锚点, 剩余的作为小目标的类别的初始锚点;

所述大目标和小目标的划分为根据实际检测时的目标进行区分尺寸自定义, 大于区分

尺寸的待检测目标即为大目标,小于于区分尺寸的目标即为小目标。

4.如权利要求2或3所述的一种基于深度神经网络的密集小目标检测方法,其特征在于,在所述步骤5和步骤6中,在使用改进后的yolo-tiny框架时,采用损失函数对模型不断迭代优化参数,所述损失函数包括三个部分:目标定位偏移量损失 $L_{cla}(d, t)$,目标置信度损失 $L_{conf}(C, \hat{C})$,以及目标分类损失 $L_{obj}(P, \hat{P})$,具体如下:

$$Loss = \lambda_1 L_{conf}(C, \hat{C}) + \lambda_2 L_{cla}(d, t) + \lambda_3 L_{obj}(P, \hat{P})$$

其中, $\lambda_1, \lambda_2, \lambda_3$ 为参数;

目标定位偏移量损失 $L_{cla}(d, t)$ 的计算公式为:

$$L_{cla}(d, t) = (2 - w \times h) \times \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (d_i^j - t_i^j)^2,$$

目标定位偏移量损失 $L_{cla}(d, t)$ 的计算采用的是预测偏移量 $d_i = (d_x, d_y, d_w, d_h)$ 与真实偏移量 $t_i = (t_x, t_y, t_w, t_h)$ 差平方和,再乘以 $(2 - w \times h)$ 的比例系数, w 和 h 分别是真实框的宽和高,所述真实框为实际检测时真实的检测框, $K \times K$ 代表检测层大小,即有 $K \times K$ 个网格, M 代表锚点的个数, I_{ij}^{obj} 代表第 i 个网格中第 j 个锚点是否预测目标,当锚点与真实框有交集时,则 $I_{ij}^{obj} = 1$,即第 i 个网格中第 j 个锚点预测目标,否则为0;

目标置信度损失 $L_{conf}(C, \hat{C})$ 的计算公式为:

$$L_{conf}(C, \hat{C}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (C_i \log \hat{C}_i + (1 - C_i) \log(1 - \hat{C}_i)),$$

目标置信度损失 $L_{conf}(C, \hat{C})$ 的计算采用交叉熵损失函数检测预测框存在目标的概率,公式中 C_i 代表该网格是否预测目标的真实值, \hat{C}_i 代表该网格是否预测目标的概率值;

目标分类损失 $L_{obj}(P, \hat{P})$ 的计算公式为:

$$L_{obj}(P, \hat{P}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} \sum_{c \in class} (P_i^j \log \hat{P}_i^j + (1 - P_i^j) \log(1 - \hat{P}_i^j)),$$

目标分类损失 $L_{obj}(P, \hat{P})$ 的计算也同样采取交叉熵损失函数,其中, c 代表目标的类别, $class$ 代表目标类别的总数, P_i^j 代表该第 i 个网格中第 j 个锚点中的目标,类别为 c 的真实值; \hat{P}_i^j 代表该第 i 个网格中第 j 个锚点中目标,预测类别为 c 的概率值。

5.如权利要求1-3任一项所述的一种基于深度神经网络的密集小目标检测方法,其特征在于,所述步骤5中,首先使用yolo-tiny框架进行模型训练,当训练次数达到2000次后,加入遗传算法,对锚点进行微调。

6.如权利要求1所述的一种基于深度神经网络的密集小目标检测方法,其特征在于,所述步骤1中,在进行摄像头的搭建时,将摄像头搭建为摄像头的水平视角与待检测目标的中心的夹角不超过 30° 的姿态位置。

7.如权利要求1所述的一种基于深度神经网络的密集小目标检测方法,其特征在于,所述步骤2中,所述预处理的的操作还包括:挑选采集的图像中由不同摄像机拍摄且图像中背景

本发明所采取损失函数对模型进行迭代优化。在从属权利要求中进一步说明损失函数的各类及其原理

不同但待检测目标位置姿态相关的图像,从挑选的图像中将待检测目标抠出并粘贴到背景不一样的图像上,并连同待检测目标的xml标注文件一起进行迁移,从而进行目标的增强;所述目标位置姿态相关的定义为同类别的不同待检测目标,在不同的拍摄背景下,摆放的姿态的角度差在一个自定义的角度误差范围内。

8.如权利要求1所述的一种基于深度神经网络的密集小目标检测方法,其特征在于,对于摄像机采集到的非正方形的图像,所述步骤2中的预处理的**操作还包括:将非正方形的图像进行填充边缘,补为正方形的尺寸的图像,然后再将经过填充边缘放大后的正方形的图像缩小或放大到规定的尺寸,所述规定的尺寸为满足后续训练检测操作要求的尺寸,根据实际需求调整。** **将权利要求一中提到的过程步骤进行详细分述。对图片预处理进行进一步解释**

9.如权利要求1所述的一种基于深度神经网络的密集小目标检测方法,其特征在于,所述步骤6中,进行目标检测的步骤具体为:首先对实际检测的图像进行图像填充,然后将图像填充后的图像送入到训练的模型中,进一步地生成锚点,接着采用非极大值抑制算法消除重叠框后生成最终的检测结果。

一种基于深度神经网络的密集小目标检测方法

技术领域

[0001] 本发明属于计算机神经网络图像识别技术领域,具体地说,涉及一种基于深度神经网络的密集小目标检测方法。

背景技术

[0002] 随着计算机技术的发展,机器学习神经网络已经广泛用于图像智能识别等技术领域,并通过这些技术领域应用到各行各业中,其已经在潜移默化的改变人们的生活习惯。而在现有技术的图像识别中,对于精细的小目标的识别还存在着许多需要改进的地方。

[0003] 如专利号为CN110503112 A,专利申请名称为《一种增强特征学习的小目标检测及识别方法》的专利文献中记载,其使用一种增强特征学习的小目标方法产生候选框,但是增加了网络的深度,使得小目标检测的速度降低,且该方法只针对了小目标进行改进,不能应对市场各种场景变化;

而如专利号为CN110298402 A,专利申请名称为《一种小目标检测性能优化方法》的专利文献中记载,其使用了yolov3固定的锚点产生方法,不能很好预测小目标位置从而影响小目标检测精度;

而如专利号为CN111275171 A,专利申请名称为《一种基于参数共享的多尺度超分重建的小目标检测方法》的专利文献中记载,其使用两阶段任务的目标检测与识别方法,不仅训练时间长,且不能达到实时检测。

发明内容

[0004] 本发明针对现有技术对于小目标的检测无法做到对深度和检测速度的兼顾,不能很好预测小目标的位置从而影响检测精度,并且兼顾实时检测的缺点,提出了一种基于深度神经网络的密集小目标检测方法,按照目标的类别的方式对目标进行锚点划分比例进行计算,并采用kmeans聚类,获得各类初始锚点分配,此种分配情况可以精确反应出小目标的细微变化,有效表征训练集中的小物体,使得模型容易收敛,在提高了检测精度和深度的同时,还可增加检测速度,实现实时检测。

[0005] 本发明具体实现内容如下:

本发明提出了一种基于深度神经网络的密集小目标检测方法,如图1所示,包括以下步骤:

步骤1:搭建摄像头,并采集图像;

步骤2:对采集的图像进行分类和预处理得到图像集,并将图像集划分为用于预训练的预训练集、用于验证的验证集以及用于测试的测试集;所述预处理包括对采集到的图像进行标注,并为每一张采集的图像中的待检测目标生成一个xml标注文件,所述标注文件中包括待检测的各类别目标的基础信息;

步骤3:基于目标分布情况,对预训练集中的图像进行计算,得到预训练集中的图像每一个类别目标的分布占比;

步骤4:对预训练集中的图像根据步骤3得到的分布占比进行抗干扰逐类别k-means聚类,获得各个类别的初始锚点的初始锚点分配比例;

步骤5:按照各个类别的初始锚点分配比例,使用yolo-tiny框架结合遗传算法进行模型训练;

步骤6:使用训练后的模型进行目标检测;具体为:首先对实际检测的图像进行图像填充,然后将图像填充后的图像送入到训练的模型中,进一步地生成锚点,接着采用非极大值抑制算法消除重叠框后生成最终的检测结果。

[0006] 为了更好地实现本发明,进一步地,所述步骤3的具体操作为:

首先计算出预训练集中所有的目标的数量,然后根据分类计算出各个类别的目标占据了所有的目标的总数的比例;

所述步骤4的具体操作为:得到各个类别的目标占据所有目标的总数的比例后,按照所述比例对各个类别的目标分别进行k-means聚类,进而得到各个类别的初始锚点的初始锚点分配比例。

[0007] 为了更好地实现本发明,进一步地,所述改进具体为在yolo-tiny框架的基础上进一步增加第24route层,第25 conv层,第26upsample层,第27route层,第28conv层,第29conv层构建的检测层,改进后的yolo-tiny框架具体结构依次包括:0conv层、1max层、2conv层、3max层、4conv层、5max层、6conv层、7max层、8conv层、9max层、10conv层、11max层、12conv层、13conv层、14conv层、15conv层、16yolo层、17route层、18conv层、19upsample层、20route层、21conv层、22conv层、23yolo层、24route层、25conv层、26upsample层、27route层、28conv层、29conv层、30yolo层;

在改进后的yolo-tiny框架中,将一张目标图像划分为 24×24 、 48×48 和 96×96 三种网格形式,采用SGD随机优化算法进行网络训练;并分别对应经过16yolo层、23yolo层和30yolo层这三个检测层输出 $24 \times 24 \times Q$ 、 $48 \times 48 \times Q$ 和 $96 \times 96 \times Q$ 三个结果,在三个结果中, 24×24 、 48×48 和 96×96 分别代表对应的检测层的大小,Q的计算公式为:

$$Q = (4 + 1 + M) \times O;$$

其中,M为预训练集中的图像类别个数,O为检测层上每一个点对应的锚点个数,此处所述锚点个数与步骤4中的初始锚点数对应,式中4代表的是训练时的位置框信息 $\{x, y, w, h\}$;1表示置信度;

所述框位置信息 $\{x, y, w, h\}$ 为图像检测时产生的检测框的位置信息,其中x,y为检测框左上角的坐标值,w,h分别表示检测框的宽度和高度。

[0008] 为了更好地实现本发明,进一步地,将各类别的目标根据占据图像中的尺寸比例分为大目标和小目标,采用16yolo层即 24×24 网格形式的检测层对大目标的类别进行检测,另外两种网格形式的检测层对小目标的类别进行检测;且在所述步骤4中,对于大目标的类别的初始锚点的分配不低于初始锚点总数的三分之一;即当大目标的类别占有所有目标的总数的三分之一以上时,按照各个类别的目标的数量占比进行分配,而当大目标的类别占有所有目标的总数的三分之一以下时,分配所有初始锚点的三分之一作为大目标的类别的初始锚点,剩余的作为小目标的类别的初始锚点;

所述大目标和小目标的划分为根据实际检测时的目标进行区分尺寸自定义,大于区分尺寸的待检测目标即为大目标,小于于区分尺寸的目标即为小目标。

[0009] 为了更好地实现本发明,进一步地,在所述步骤5和步骤6中,在使用改进后的yolo-tiny框架时,采用损失函数对模型不断迭代优化参数,所述损失函数包括三个部分:目标定位偏移量损失 $L_{cla}(d, t)$,目标置信度损失 $L_{conf}(C, \hat{C})$,以及目标分类损失 $L_{obj}(P, \hat{P})$,具体如下:

$$Loss = \lambda_1 L_{conf}(C, \hat{C}) + \lambda_2 L_{cla}(d, t) + \lambda_3 L_{obj}(P, \hat{P})$$

其中, $\lambda_1, \lambda_2, \lambda_3$ 为参数;

目标定位偏移量损失 $L_{cla}(d, t)$ 的计算公式为:

$$L_{cla}(d, t) = (2 - w \times h) \times \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (d_i^j - t_i^j)^2,$$

目标定位偏移量损失 $L_{cla}(d, t)$ 的计算采用的是预测偏移量 $d_i = (d_x, d_y, d_w, d_h)$ 与真实偏移量 $t_i = (t_x, t_y, t_w, t_h)$ 差平方和,再乘以 $(2 - w \times h)$ 的比例系数, w 和 h 分别是真实框的宽和高, $K \times K$ 代表检测层大小,即有 $K \times K$ 个网格, M 代表锚点的个数, I_{ij}^{obj} 代表第 i 个网格中第 j 个锚点是否预测目标,当锚点与真实框有交集时,则 $I_{ij}^{obj} = 1$,即第 i 个网格中第 j 个锚点预测目标,否则为0;

目标置信度损失 $L_{conf}(C, \hat{C})$ 的计算公式为:

$$L_{conf}(C, \hat{C}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (C_i \log \hat{C}_i + (1 - C_i) \log(1 - \hat{C}_i)),$$

目标置信度损失 $L_{conf}(C, \hat{C})$ 的计算采用交叉熵损失函数检测预测框存在目标的概率,公式中 C_i 代表该网格是否预测目标的真实值, \hat{C}_i 代表该网格是否预测目标的概率值;

目标分类损失 $L_{obj}(P, \hat{P})$ 的计算公式为:

$$L_{obj}(P, \hat{P}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} \sum_{c \in class} (P_i^j \log \hat{P}_i^j + (1 - P_i^j) \log(1 - \hat{P}_i^j)),$$

目标分类损失 $L_{obj}(P, \hat{P})$ 的计算也同样采取交叉熵损失函数,其中, c 代表目标的类别, $class$ 代表目标类别的总数, P_i^j 代表该第 i 个网格中第 j 个锚点中的目标,类别为 c 的真实值; \hat{P}_i^j 代表该第 i 个网格中第 j 个锚点中目标,预测类别为 c 的概率值。

[0010] 为了更好地实现本发明,进一步地,所述步骤5中,首先使用yolo-tiny框架进行模型训练,当训练次数达到2000次后,加入遗传算法,对锚点进行微调。

[0011] 为了更好地实现本发明,进一步地,所述步骤1中,在进行摄像头的搭建时,将摄像头搭建为摄像头的水平视角与待检测目标的中心的夹角不超过 30° 的姿态位置。

[0012] 为了更好地实现本发明,进一步地,所述步骤2中,所述预处理的的操作还包括:挑选采集的图像中由不同摄像机拍摄且图像中背景不同但待检测目标位置姿态相关的图像,从挑选的图像中将待检测目标抠出并粘贴到背景不一样的图像上,并连同待检测目标的xml标注文件一起进行迁移,从而进行目标的增强。所述目标位置姿态相关的定义为同类别的不同待检测目标,在不同的拍摄背景下,摆放的姿态的角度差在一个自定义的角度误差范围内。

[0013] 为了更好地实现本发明,进一步地,对于摄像机采集到的非正方形的图像,所述预处理的操作还包括:将非正方形的图像进行填充边缘,补为正方形的尺寸的图像,然后再将经过填充边缘放大后的正方形的图像缩小或放大到规定的尺寸,所述规定的尺寸为满足后续训练检测操作要求的尺寸,根据实际需求调整。

[0014] 为了更好地实现本发明,进一步地,所述步骤6中,进行目标检测的步骤具体为:首先对实际检测的图像进行图像填充,然后将图像填充后的图像送入到训练的模型中,进一步地生成锚点,接着采用非极大值抑制算法消除重叠框后生成最终的检测结果。

[0015] 本发明与现有技术相比具有以下优点及有益效果:

(1) 采用目标的类别的方式对目标进行锚点划分比例进行计算,并采用k-means聚类,获得各类初始锚点分配,避免因为各个类别的目标数量不平衡,离群值造成锚点干扰的问题;此种分配情况可以精确反应出小目标的细微变化,有效表征训练集中的小物体,使得模型容易收敛,在提高了检测精度和深度的同时,还可增加检测速度,实现实时检测;

(2) 通过图片填充的方式避免非正方形的图像在检测时出现形变影响检测效果;

(3) 通过抠出背景的方式增加训练图像集,从而使算法稳定性以及鲁棒性更强;

(4) 使用损失函数进一步提高检测的精确度;

(5) 结合遗传算法进一步地提高了检测的精确度。

附图说明

[0016] 图1为本发明具体实施例的流程示意图;

图2为本发明摄像机与货架的位置姿态安装示意图;

图3为本发明优化前的杯子的初始锚点值的分配情况;

图4为本发明优化前的货架的初始锚点值的分配情况;

图5为经过本发明的方法分配优化后的杯子的初始锚点值的分配情况;

图6为经过本发明的方法分配优化后的货架的初始锚点值的分配情况;

图7为摄像机与货架中心偏水平视角下检测杯子和货架的效果示意图;

图8为摄像机与货架中心偏俯视视角下检测杯子和货架的效果示意图;

图9为图8效果示意图的局部放大示意图;

图10为光线较亮的情况下对货架和杯子进行检测的效果示意图;

图11为光线较暗的情况下对货架和杯子进行检测的效果示意图。

具体实施方式

[0017] 为了更清楚地说明本发明实施例的技术方案,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,应当理解,所描述的实施例仅仅是本发明的一部分实施例,而不是全部的实施例,因此不应被看作是对保护范围的限定。基于本发明中的实施例,本领域普通技术工作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。

[0018] 在本发明的描述中,需要说明的是,除非另有明确的规定和限定,术语“设置”、“相连”、“连接”应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或一体地连接;可以是机械连接,也可以是电连接;也可以是直接相连,也可以是通过中间媒介间接相连,可

以是两个元件内部的连通。对于本领域的普通技术人员而言,可以具体情况理解上述术语在本发明中的具体含义。

[0019] 实施例1:

本实施例提出了一种基于深度神经网络的密集小目标检测方法,包括以下步骤:

步骤1:搭建摄像头,并采集图像;

步骤2:对采集的图像进行分类和预处理得到图像集,并将图像集划分为用于预训练的预训练集、用于验证的验证集以及用于测试的测试集;所述预处理包括对采集到的图像进行标注,并为每一张采集的图像中的待检测目标生成一个xml标注文件,所述标注文件中包括待检测的各类别目标的基础信息;

步骤3:基于目标分布情况,对预训练集中的图像进行计算,得到预训练集中的图像每一个类别目标的分布占比;具体操作为:首先计算出预训练集中所有的目标的数量,然后根据分类计算出各个类别的目标占据了所有的目标的总数的比例;

步骤4:对预训练集中的图像根据步骤3得到的分布占比进行抗干扰逐类别k-means聚类,获得各个类别的初始锚点分配比例;具体操作为:得到各个类别的目标占据所有目标的总数的比例后,按照所述比例对各个类别的目标分别进行k-means聚类,进而得到各个类别的初始锚点分配比例;当训练次数达到2000次后,加入遗传算法,对锚点进行微调;

步骤5:按照各个类别的初始锚点分配比例,使用yolo-tiny框架结合遗传算法进行模型训练;

步骤6:使用训练后的模型进行目标检测。

[0020] 实施例2:

本实施例在上述实施例1的基础上,为了更好地实现本发明,进一步地,所述改进具体为在yolo-tiny框架的基础上进一步增加第24route层,第25 conv层,第26upsample层,第27route层,第28conv层,第29conv层构建的检测层,改进后的yolo-tiny框架具体结构依次包括:0conv层、1max层、2conv层、3max层、4conv层、5max层、6conv层、7max层、8conv层、9max层、10conv层、11max层、12conv层、13conv层、14conv层、15conv层、16yolo层、17route层、18conv层、19upsample层、20route层、21conv层、22conv层、23yolo层、24route层、25conv层、26upsample层、27route层、28conv层、29conv层、30yolo层;具体表示如下表1所示:

layer	filters	size/strd	input	output
0 conv	16	3 x 3/1	768 x 768 x 3	768 x 768 x 16
1 max		2 x 2/2	768 x 768 x 16	384 x 384 x 16
2 conv	32	3 x 3/1	384 x 384 x 16	384 x 384 x 32
3 max		2 x 2/2	384 x 384 x 32	192 x 192 x 32
4 conv	64	3 x 3/1	192 x 192 x 32	192 x 192 x 64
5 max		2x 2/2	192 x 192 x 64	96 x 96 x 64
6 conv	128	3 x 3/1	96 x 96 x 64	96 x 96 x 128
7 max		2x 2/2	96 x 96 x 128	48 x 48 x 128
8 conv	256	3 x 3/1	48 x 48 x 128	48 x 48 x 256
9 max		2x 2/2	48 x 48 x 256	24 x 24 x 256
10 conv	512	3 x 3/1	24 x 24 x 256	24 x 24 x 512
11 max		1x 1/1	24 x 24 x 512	24 x 24 x 512
12 conv	1024	3 x 3/1	24 x 24 x 512	24 x 24 x 1024
13 conv	256	1 x 1/1	24 x 24 x 1024	24 x 24 x 256
14 conv	512	3 x 3/1	24 x 24 x 256	24 x 24 x 512
15 conv	42	1 x 1/1	24 x 24 x 512	24 x 24 x 42
16 yolo				
17 route	13			24 x 24 x 256
18 conv	128	1 x 1/1	24 x 24 x 256	24 x 24 x 128
19 ups		2x	24 x 24 x 128	48 x 48 x 128
20 route	19、8			48 x 48 x 384
21 conv	256	3 x 3/1	48 x 48 x 384	48 x 48 x 256
22 conv	21	1 x 1/1	48 x 48 x 256	48 x 48 x 21
23 yolo				
24 route	21			48 x 48 x 256
25conv	128	1 x 1/1	48 x 48 x 256	48 x 48 x 128
26 ups		2x	48 x 48 x 128	96 x 96 x 128
27 route	26 6			96 x 96 x 256
28 conv	128	3 x 3/1	96 x 96 x 256	96 x 96 x 128
29 conv	21	1 x 1/1	96 x 96 x 128	96 x 96 x 21
30 yolo				

表1 改进后的yolo-tiny框架的具体网络结构示意图

在改进后的yolo-tiny框架中,将一张目标图像划分为 24×24 、 48×48 和 96×96 三种网格形式,采用SGD随机优化算法进行网络训练;并分别对应经过16yolo层、23yolo层和30yolo层这三个检测层输出 $24 \times 24 \times Q$ 、 $48 \times 48 \times Q$ 和 $96 \times 96 \times Q$ 三个结果,在三个结果中,

24×24、48×48和96×96分别代表对应的检测层的大小, $Q=(4+1+M) \times 0$, 其中, M 为预训练集中的图像类别个数, 0为检测层上每一个点对应的锚点个数, 此处所述锚点个数与步骤4中的初始锚点数对应, 式中4代表的是训练时的框位置信息 $\{x, y, w, h\}$; 1表示置信度; 其中 x, y 为检测框的左上角的坐标值。

[0021] 本实施例的其他部分与上述实施例1相同, 故不再赘述。

[0022] 实施例3:

本实施例在上述实施例1-2任一项的基础上, 为了更好地实现本发明, 进一步地, 将各类别的目标根据占据图像中的尺寸比例分为大目标和小目标, 采用16yolo层即24×24网格形式的检测层对大目标的类别进行检测, 另外两种网格形式的检测层对小目标的类别进行检测; 且在所述步骤4中, 对于大目标的类别的初始锚点的分配不低于初始锚点总数的三分之一; 即当大目标的类别占有所有目标的总数的三分之一以上时, 按照各个类别的目标的数量占比进行分配, 而当大目标的类别占有所有目标的总数的三分之一以下时, 分配所有初始锚点的三分之一作为大目标的类别的初始锚点, 剩余的作为小目标的类别的初始锚点。

[0023] 为了更好地实现本发明, 进一步地, 在所述步骤5和步骤6中, 在使用改进后的yolo-tiny框架时, 采用损失函数对模型不断迭代优化参数, 所述损失函数包括三个部分: 目标定位偏移量损失 $L_{cla}(d, t)$, 目标置信度损失 $L_{conf}(C, \hat{C})$, 以及目标分类损失 $L_{obj}(P, \hat{P})$, 具体如下:

$$Loss = \lambda_1 L_{conf}(C, \hat{C}) + \lambda_2 L_{cla}(d, t) + \lambda_3 L_{obj}(P, \hat{P})$$

其中, $\lambda_1, \lambda_2, \lambda_3$ 为参数;

目标定位偏移量损失 $L_{cla}(d, t)$ 的计算公式为:

$$L_{cla}(d, t) = (2 - w \times h) \times \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (d_i^j - t_i^j)^2,$$

目标定位偏移量损失 $L_{cla}(d, t)$ 的计算采用的是预测偏移量 $d_i = (d_x, d_y, d_w, d_h)$ 与真实偏移量 $t_i = (t_x, t_y, t_w, t_h)$ 差平方和, 再乘以 $(2 - w \times h)$ 的比例系数, w 和 h 分别是真实框的宽和高, $K \times K$ 代表检测层大小, 即有 $K \times K$ 个网格, M 代表锚点的个数, I_{ij}^{obj} 代表第 i 个网格中第 j 个锚点是否预测目标, 当锚点与真实框有交集时, 则 $I_{ij}^{obj} = 1$, 即第 i 个网格中第 j 个锚点预测目标, 否则为0;

目标置信度损失 $L_{conf}(C, \hat{C})$ 的计算公式为:

$$L_{conf}(C, \hat{C}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (C_i \log \hat{C}_i + (1 - C_i) \log(1 - \hat{C}_i)),$$

目标置信度损失 $L_{conf}(C, \hat{C})$ 的计算采用交叉熵损失函数检测预测框存在目标的概率, 公式中 C_i 代表该网格是否预测目标的真实值, \hat{C}_i 代表该网格是否预测目标的概率值;

目标分类损失 $L_{obj}(P, \hat{P})$ 的计算公式为:

$$L_{obj}(P, \hat{P}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} \sum_{c \in class} (P_i^j \log \hat{P}_i^j + (1 - P_i^j) \log(1 - \hat{P}_i^j)),$$

目标分类损失 $L_{obj}(P, \hat{P})$ 的计算也同样采取交叉熵损失函数, 其中, c 代表目标的类别, $class$ 代表目标类别的总数, p_i^j 代表该第 i 个网格中第 j 个锚点中的目标, 类别为 c 的真实值; \hat{p}_i^j 代表该第 i 个网格中第 j 个锚点中目标, 预测类别为 c 的概率值。

[0024] 工作原理:

本实施例的其他部分与上述实施例1-2任一项相同, 故不再赘述。

[0025] 实施例4:

本实施例在上述实施例1-3任一项的基础上, 进一步地给出了使用货架与杯子进行实际检测的实施例, 其中每层货架被定义为大目标, 杯子被定义为密集的小目标, 且杯子根据颜色不同进行分类;

在本实施例中:

步骤1: 以图2为例, 一个高为150cm的三层货架, 每一层高为50cm, 在水平距离为200cm处安装摄像头, 摄像头与货架中心平行时, 三层货架和杯子拍摄的效果最佳。在实际应用中, 摄像头安装位置会受到场地的制约, 为了保证摄像头拍全三层货架, 且杯子无明显形变, 摄像机与货架中心的夹角不得超过30度, 根据公式:

$$\tan \alpha = \frac{h-75}{200}$$

其中, h 代表摄像机安装的高度, α 代表摄像机和货架中心的夹角。则计算可知摄像机安装最高位置不得超过190.2cm。

[0026] 对每一个仓库不同时间段随机提取视频, 对视频进行解帧, 挑选无污染的图像, 即货架和杯子没有被完全遮挡;

步骤2: 利用LabelImg工具对图像进行标注, 为每一张图片生成一个xml标注文件, 存储杯子位置、尺寸、颜色信息以及每一层货架的位置、尺寸信息;

每一个仓库的杯子存量不同, 为了使图像集更加多样化, 挑选不同仓库背景但是货架摆放位置大致一样的图像, 将整个货架连同杯子从图像扣出, 粘贴到背景不一样的图像上, 连同货架以及杯子的标注文件xml迁移, 同时也节省再一次标注的人力; 目前工业摄像机分辨率是720×1080或者1080×1920, 防止图像中小目标在训练过程中发生形变需要对图像进行预处理, 利用RGB颜色信息[255, 255, 255]或者[114, 114, 114]将图像补成1080×1080或者1920×1920的正方形图像, 在此基础上将图像缩放到768×768, 根据实际应用发现, 将图像缩放到600-1100这个尺寸区间的正方形都可以满足本方法的检测; 将增强后的图像70%作为预训练集, 20%作为验证集, 10%作为测试集;

步骤3: 传统计算锚点的方法, 将训练集中所有的目标长, 宽集中进行k-means计算得到初始锚点, 如图2和图3所示, 使用传统计算锚点方法, 初始锚点个数为9, 则杯子分配的锚点只有3个, 而货架分配的锚点有6个; 该锚点分配方式, 不能精确反映出小目标的细微变化, 后续训练模型时, 无法有效表征训练集中的小物体, 导致模型难以收敛。因此将杯子和货架的数量比例加入到锚点计算方法, 设训练集中所有目标数量为 n , 杯子数量为 n_c , 货架数量为 n_l , 则杯子和货架数量占总目标数量比例分别为 p_c, p_l , 计算公式如下:

$$p_c = \frac{n_c}{n}, p_l = \frac{n_l}{n}$$

步骤4: 参照步骤四得到目标分配比例 p_c , p_l , 初始锚点为9个, 分配给杯子, 货架初始锚点分别为: $m_c = 9 \times p_c$, $m_l = 9 \times p_l$, 在yolo框架网络, 将一张768×768图片, 划分为24×24、48×48和96×96三种网格形式且每一层网络分配三个初始锚点, 其中24×24网格用来预测大目标(即货架), m_l 按照如下要求计算:

$$m_l = \begin{cases} m_l, & m_l \geq 3 \\ 3, & m_l < 3 \end{cases}$$

则, $m_c = 9 - m_l$; 对杯子和货架按照分配比例分别进行k-means聚类, 获得初始锚点的值。如图5和图6所示, 杯子分配的初始锚点不仅数量明显增多, 而且描述杯子尺寸范围更加细致; 货架分配的初始锚点虽然减少了, 但是减少了异常值的干扰, 且聚类中心点排列更加有序;

步骤5: 为了缓和密集杯子以及货架尺寸之间的断层, 杯子平均长宽[24.2, 21.4], 最小长宽[7.8, 8.4], 最大长宽[59.4, 57.6], 货架平均长宽[250.2, 122.6], 最小长宽[166.2, 63.6], 最大长宽[457.9, 331.2]; 在传统的yolo-tiny框架只有两层目标检测层, 将一张768×768图片, 划分三种网格24×24, 48×48, 缺少对大目标的检测。故在yolo-tiny后增加第24 route层, 第25 conv层, 第26 upsample层, 第27 route层, 第28 conv层, 第29 conv层构建的检测层, 可以实现24×24、48×48和96×96三种网格形式的检测, 为了使能够使该网络结构移植到移动端, 将第12 max层, 改为1×1。网络结构模型如上表1所示。

[0027] 根据前面步骤得到初始锚点尺寸以及个数, 配置网络参数。算法在Darknet开源深度学习框架下实现。采用SGD随机优化算法进行网络训练, 动量为0.9, 权重衰减因子为0.0005, 学习率为0.01。将批量目标输入网络, 进行前向传播, 经过检测层16, 23, 30输出: 24×24×42, 48×48×42, 96×96×96, 其中24×24, 48×48和96×96表示检测层大小; 42中, 4表示一个预测框位置信息{x, y, w, h}, 1表示置信度, 9表示训练图像集分为9个类别即货架和8种不同颜色的杯子, 3表示检测层上每一个点对应有3个锚点anchor;

采用损失函数对模型不断迭代优化参数, 所述损失函数包括目标定位偏移量损失 $L_{cla}(d, t)$, 目标置信度损失 $L_{conf}(c, \hat{c})$, 以及目标分类损失 $L_{obj}(c, \hat{c})$, 具体公式为:

$$Loss = \lambda_1 L_{conf}(c, \hat{c}) + \lambda_2 L_{cla}(d, t) + \lambda_3 L_{obj}(c, \hat{c});$$

其采用的是预测偏移量 $d_i = (d_x, d_y, d_w, d_h)$ 与真实偏移量 $t_i = (t_x, t_y, t_w, t_h)$ 差平方和, 再乘以 $(2 - w \times h)$ 的比例系数, w和h分别是真实框的宽和高, I_{ij}^{obj} 表示该预测框是否预测一个目标物体, 当该预测框与真实框有交集时, 则认为该预测框预测到一个目标, 大小为1, 否则等于0;

关于目标定位偏移量损失 $L_{cla}(d, t)$, 计算公式为:

$$L_{cla}(d, t) = (2 - w \times h) \times \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} (d_i^j - t_i^j)^2;$$

关于目标置信度损失 $L_{conf}(c, \hat{c})$,计算公式为:

$$L_{conf}(c, \hat{c}) = - \sum (c_i \log \hat{c}_i + (1 - c_i) \log(1 - \hat{c}_i));$$

其采用交叉熵损失函数检测预测框存在目标的概率;

关于目标分类损失 $L_{obj}(c, \hat{c})$,计算公式为:

$$L_{obj}(c, \hat{c}) = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} \sum_{c \in class} (c_i \log \hat{c}_i + (1 - c_i) \log(1 - \hat{c}_i));$$

其也同样采取交叉熵损失函数,用于判别预测框内目标是杯子还是货架, $\lambda_1, \lambda_2, \lambda_3$ 为参数;

当训练迭代2000次时,采用遗传算法和k-means对初始锚点值进行修改。设训练集中图片物体框数量为N,大小为 $N = [W_N, H_N]$,其中W, H为物体框的宽,高;初始的锚点 $N_s = [W_s, H_s]$,按照公式 $\frac{N}{N_s}$,进行比例值计算,计算最大可能召回率(Best Possible

Recall),当最大可能召回率值大于设定的阈值0.98,则不需要对锚点修改,当最大可能召回率值小于设定的阈值,则利用遗传算法和k-means重新计算锚点的值。

[0028] 步骤6:将测试集中的图像输入至步骤5中训练好的模型,得到预测物体的类别、坐标信息;采用目标置信度为0.5,当置信度大于0.5时,则认为是目标,否则丢弃;接着采用阈值为0.3的非极大值抑制算法消除重叠框,得到杯子和货架的识别结果包括:杯子的颜色信息,物体框左上角和右下角的坐标,每层货架的左上角和右下角的坐标,算法在测试集上IOU准确率达到99.97%,随后根据货架的y轴坐标推算出杯子所在货架的层数。本发明与现有技术的准确率及检测速度如下表2所示:

方法名称	IOU准确度	检测时间/图片
yolov3-tiny-2layer	95.96%	45ms
Yolov3	99.83%	700ms
Yolov3-tiny-3layer (本方法)	99.97%	100ms

表2:本发明与现有技术的效果对比表

如表2中可知,本发明精度99.97%的同时还抱有了非常高的检测速度。注:若不使用遗传算法进行优化,本发明的检测精度为99.93%,增加了遗传算法进行优化后,本发明精度为99.97%。如图7和图8所示,角度的一定偏移对于本申请最终效果的影响不大,如图9所示,图9为图8的局部展示图,在实际的效果示意图中,可以在检测框边增加标注,标注包括了种类颜色等等进行展示;如图10和图11,展示的是两种不同光线强度下的检测结果,可以得知一定范围内的光线的变化对本发明方法的影响也不大,本发明的适应性较强。需要注意的是图3-11都是原图为彩色的示意图,在本申请中因申请文件的格式限制并未很好展现出色彩的区别,但其并不能作为一个效果上的对本申请的限定,同时因图3-11都仅仅作为

一个效果展示图,对于图像中可能存在的不清楚的地方并不会对本申请的实质技术方案内容造成任何影响。

[0029] 本实施例的其他部分与上述实施例1-3任一项相同,故不再赘述。

[0030] 以上所述,仅是本发明的较佳实施例,并非对本发明做任何形式上的限制,凡是依据本发明的技术实质对以上实施例所作的任何简单修改、等同变化,均落入本发明的保护范围之内。

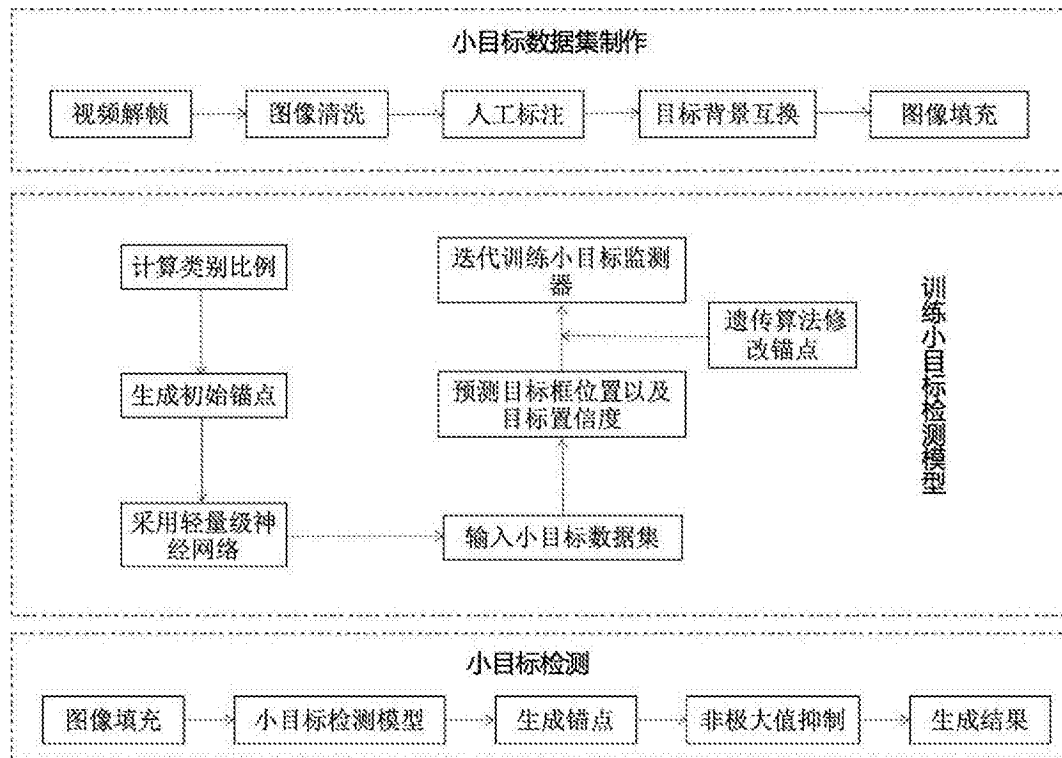


图1

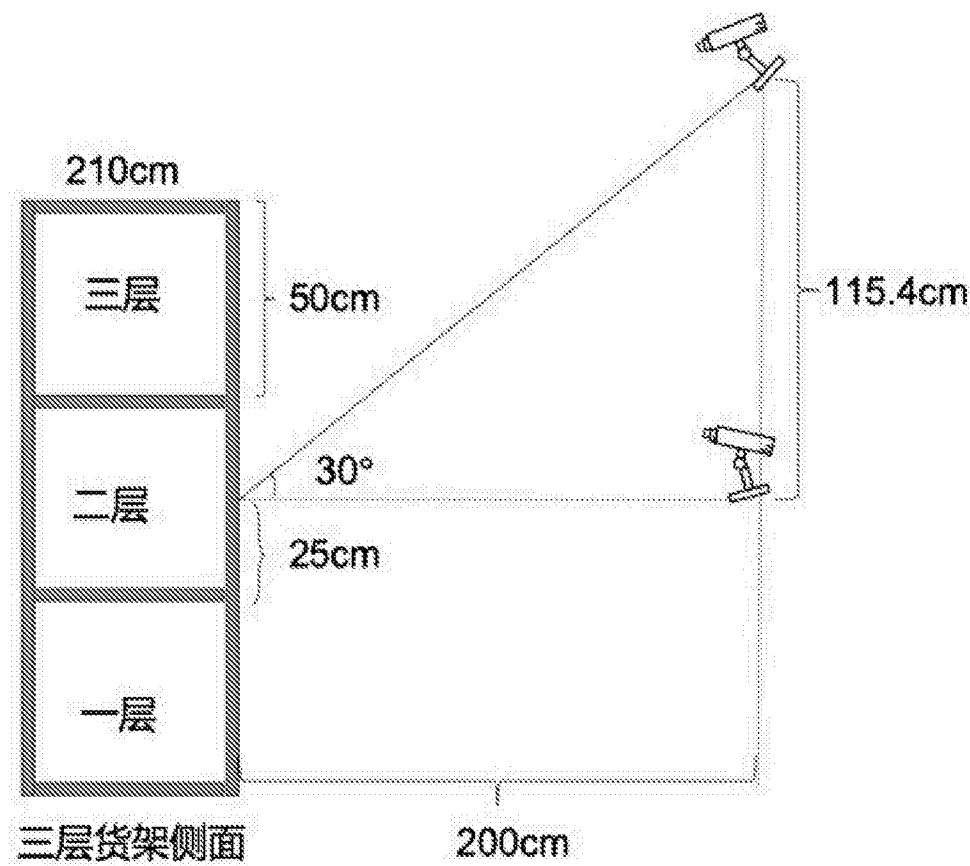


图2

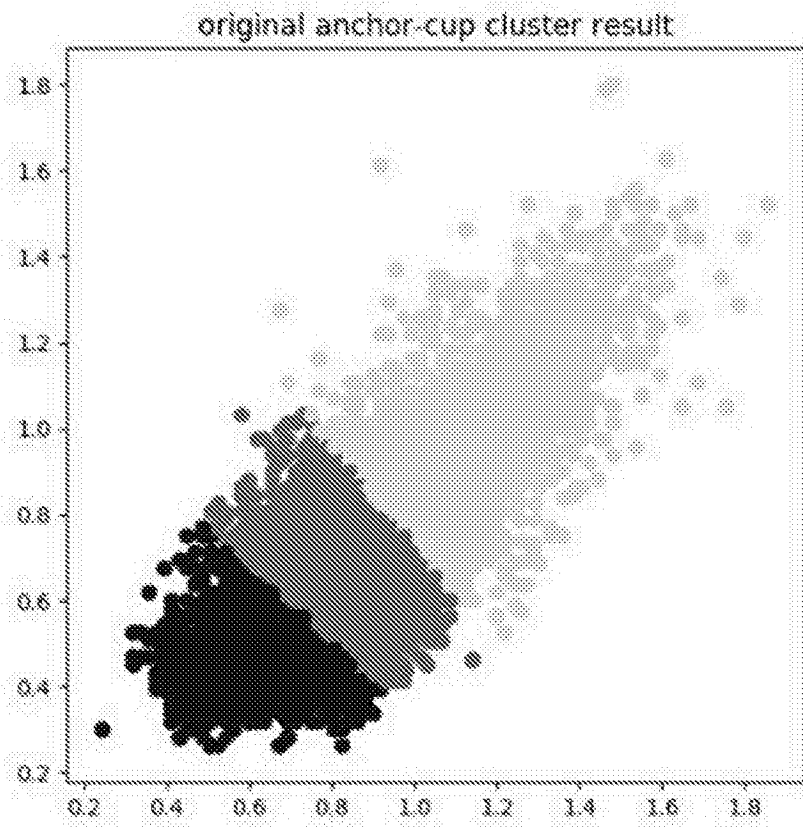


图3

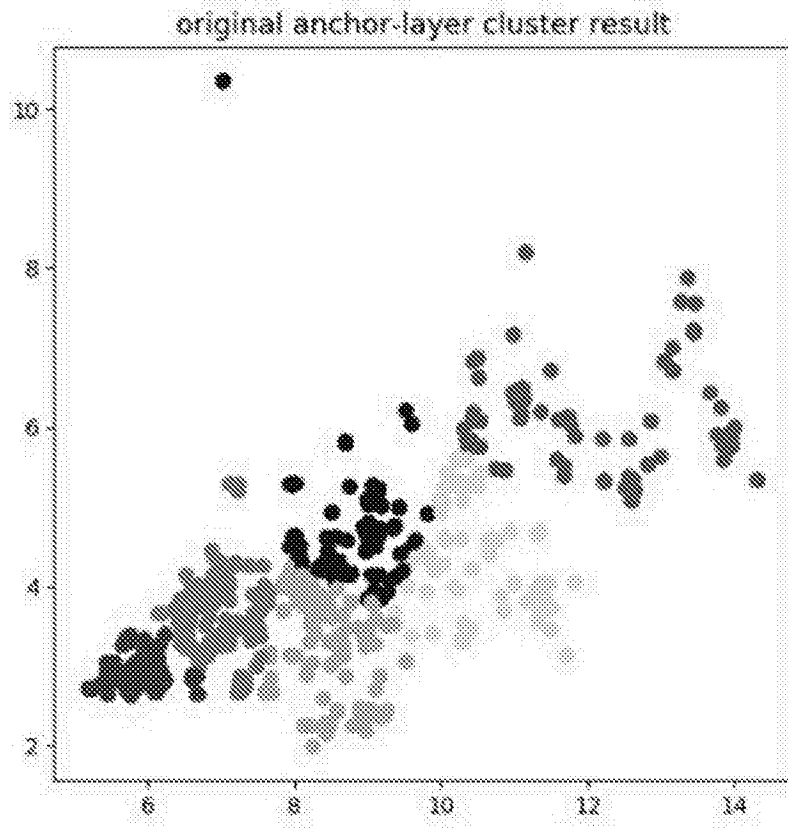


图4

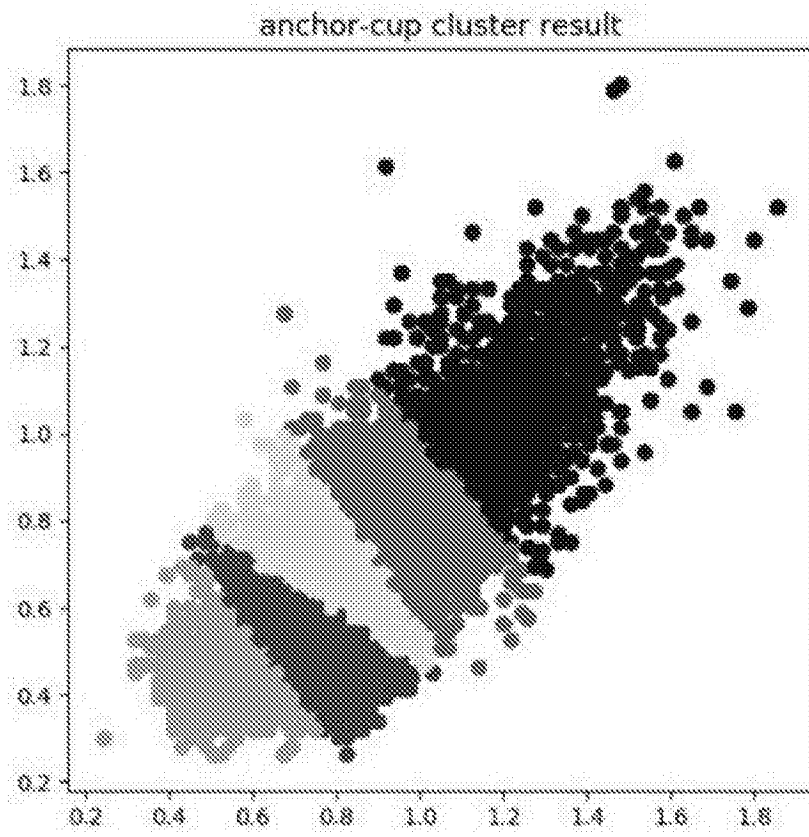


图5

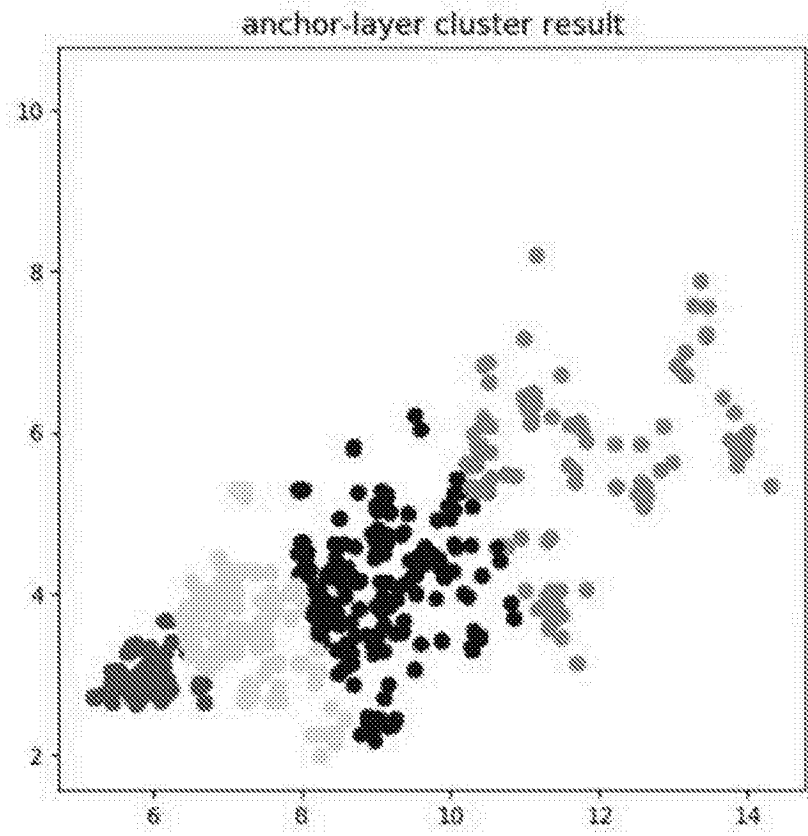


图6



图7

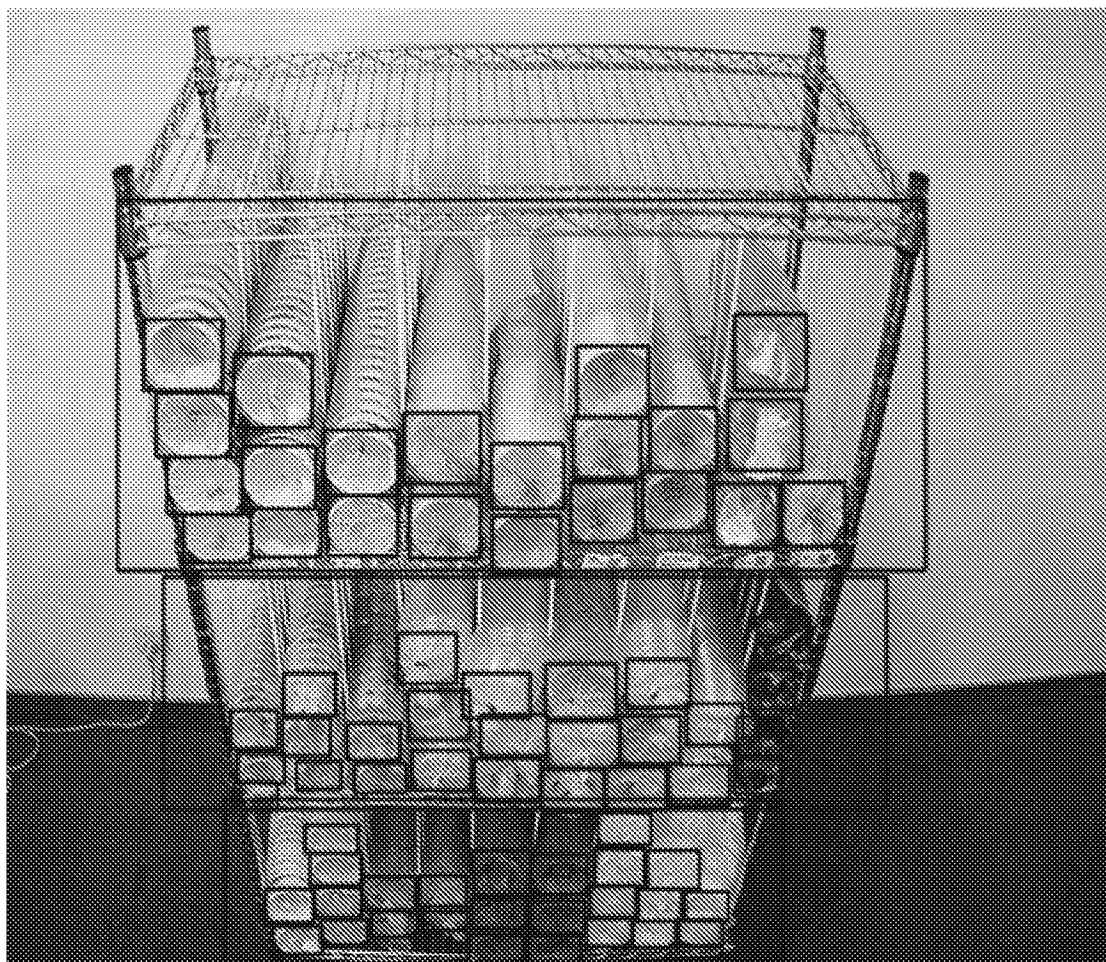


图8

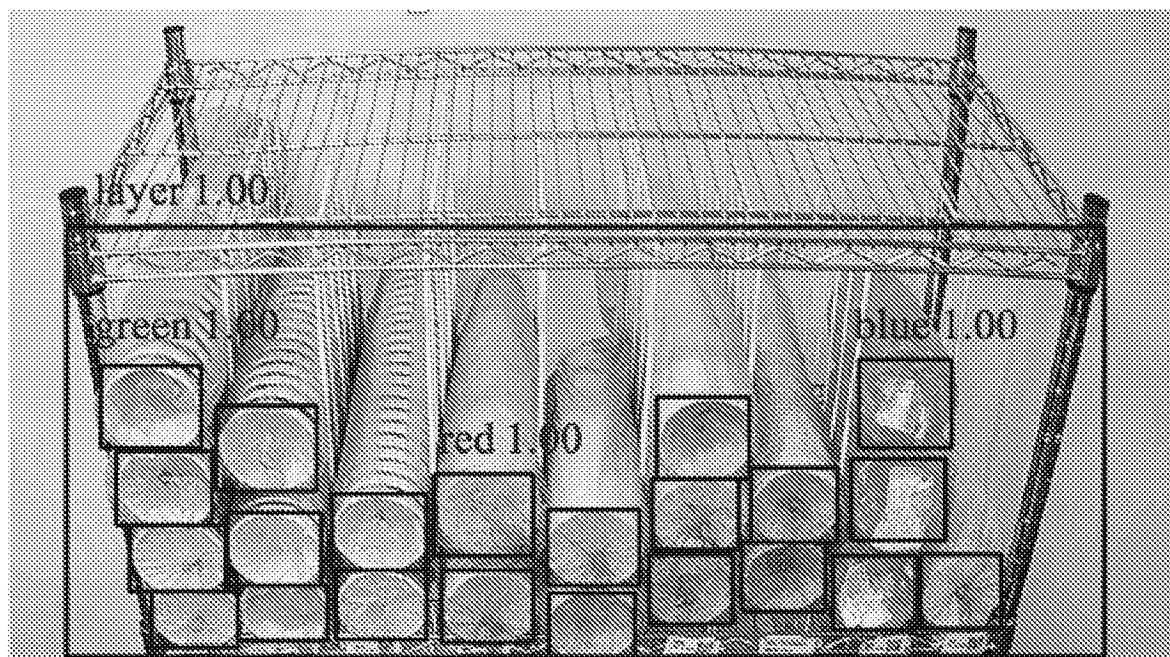


图9

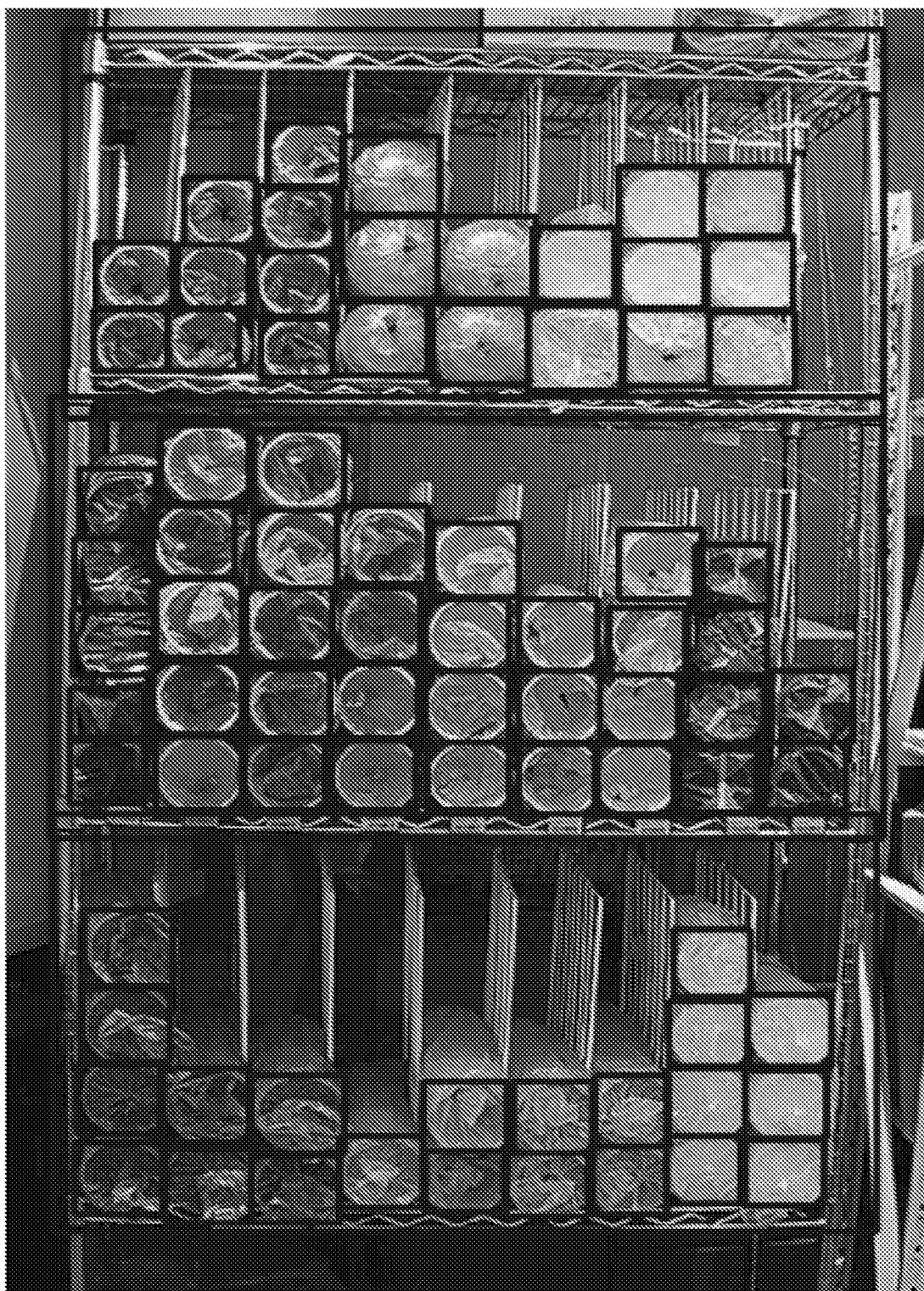


图10



图11