



(12) 发明专利申请

(10) 申请公布号 CN 113487529 A

(43) 申请公布日 2021.10.08

(21) 申请号 202110783150.4

(22) 申请日 2021.07.12

(71) 申请人 吉林大学

地址 130000 吉林省长春市前进大街2699号

(72) 发明人 何丽莉 付豪 白洪涛 曹英晖

(74) 专利代理机构 北京远大卓悦知识产权代理有限公司 11369

代理人 刘小娇

(51) Int. Cl.

G06T 5/50 (2006.01)

G06T 7/00 (2017.01)

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

G06K 9/62 (2006.01)

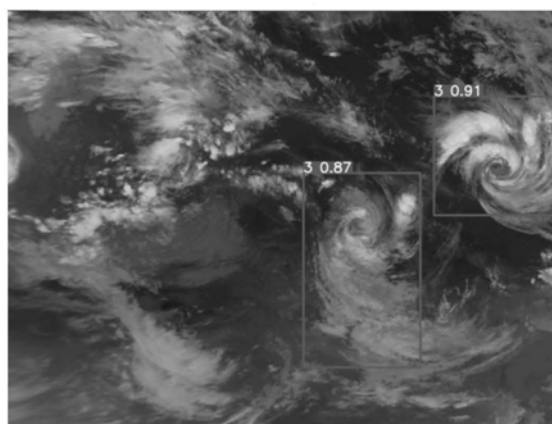
权利要求书2页 说明书12页 附图7页

(54) 发明名称

一种基于Yolo的气象卫星云图目标检测方法

(57) 摘要

本发明公开了一种基于Yolo的气象卫星云图目标检测方法,包括:一、提取出多张红外云图和多张可见光云图;二、统计可见光云图中的无效信息,根据可见光云图中的无效信息占比将多张可见光云图划分为:第一可见光云图、第二可见光云图和第三可见光云图;其中,第一可见光云图的无效信息占比 ψ_1 满足 $\psi_1 < \psi_{min}$;第二可见光云图的无效信息占比 ψ_2 满足 $\psi_{min} \leq \psi_2 \leq \psi_{max}$;第三可见光云图的无效信息占比 ψ_3 满足 $\psi_3 > \psi_{max}$; ψ_{min} 为无效信息阈值下限, ψ_{max} 为无效信息阈值上限;三、将第一可见光云图与其对应的红外云图进行融合,得到融合云图;将与第一可见光云图相对应的红外云图、融合云图以及第三可见光云图组成待检测云图集;四、采用Yolo算法对检测云图集的云图进行目标检测,识别出云图中的典型天气现象。



1. 一种基于Yolo的气象卫星云图目标检测方法,其特征在于,包括如下步骤:

步骤一、从气象卫星采集的原始卫星云图数据中提取出多张红外云图和多张可见光云图;

其中,所述红外云图与所述可见光云图为一一对应关系;

步骤二、统计所述可见光云图中的无效信息,并且根据所述可见光云图中的无效信息占比将所述多张可见光云图划分为:第一可见光云图、第二可见光云图和第三可见光云图;

其中,所述第一可见光云图的无效信息占比 ψ_1 满足 $\psi_1 < \psi_{\min}$;

所述第二可见光云图的无效信息占比 ψ_2 满足 $\psi_{\min} \leq \psi_2 \leq \psi_{\max}$;

所述第三可见光云图的无效信息占比 ψ_3 满足 $\psi_3 > \psi_{\max}$;

式中, ψ_{\min} 为无效信息阈值下限, ψ_{\max} 为无效信息阈值上限;

步骤三、将所述第一可见光云图与其对应的所述红外云图进行融合,得到融合云图;以及

将与所述第一可见光云图相对应的红外云图、所述融合云图以及所述第三可见光云图组成待检测云图集;

步骤四、采用Yolo算法对所述检测云图集的云图进行目标检测,识别出云图中的典型天气现象。

2. 根据权利要求1所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,在所述步骤二中,使用开源opencv库读取可见光云图的所有像素点,并判断所述可见光无用像素点的数量占总像素点的比重,得到所述可见光云图中的无效信息占比。

3. 根据权利要求2所述的基于Yolo的气象卫星云图目标检测方法,其特征在于, $\psi_{\min} = 25\%$, $\psi_{\max} = 75\%$ 。

4. 根据权利要求2或3所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,在所述步骤三中,所述第一可见光云图与其对应的所述红外云图进行融合,包括:

步骤1、将待融合的所述第一可见光云图和所述红外云图缩放为相同尺寸;

步骤2、采用拉普拉斯金字塔分别对所述第一可见光云图和所述红外云图进行多尺度表达后,对所述第一可见光云图和所述红外云图进行分层融合;

其中,在每个层次进行融合时,根据位置信息将第一可将光云图中有用像素点所在的区域与红外云图中的其他区域融合,得到融合云图。

5. 根据权利要求4所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,在所述步骤四之前还包括:根据卫星云图数据制作COCO数据集格式的训练数据集,对Yolo算法模型进行训练,并对训练模型进行速度和检测精度测试后,得到Yolo算法气象云图检测模型。

6. 根据权利要求5所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,还包括对所述Yolo算法气象云图检测模型进行Web端部署,使用Flask框架作为前端框架,Vue框架作为后端框架;以实现在web端上传云图进行检测的功能,达到实时检测的效果。

7. 根据权利要求6所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,在所述步骤四中,采用Yolov3或Yolov5算法对所述检测云图集的云图进行目标检测。

8. 根据权利要求7所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,在所述步骤四中,采用Yolov5算法对所述检测云图集的云图进行目标检测。

9. 根据权利要求8所述的基于Yolo的气象卫星云图目标检测方法,其特征在于,所述典

型天气现象包括:冷锋、暖锋、台风、气旋、强对流和激流云系6种天气现象。

一种基于Yolo的气象卫星云图目标检测方法

技术领域

[0001] 本发明属于气象卫星云图识别检测技术领域,特别涉及一种基于Yolo的气象卫星云图目标检测方法。

背景技术

[0002] 气象卫星是一种重要的气象工具,而在气象卫星传输的数据中,比较重要的是气象云图文件,它们在天气预报尤其是分析降水等工作中扮演了重要的角色。气象卫星可以获取白天的可见光云图、昼夜的红外云图和水汽分布图等并进行传真传播,提供国内外气象资料利用站接收利用。对于天气预报来说,卫星云图分析的内容包括了区分不同通道的云图,如红外云图和可见光云图,以及分析大范围云的分布,及其对应的天气系统。在对卫星云图进行分析的过程中,有一些基本特征,例如型式(结构),可以通过不同型式的云系或云团与不同的天气系统或物理系统过程向关联,例如台风、气旋、低涡等云系呈螺旋状结构、而锋面、急流和热带辐合带的云系成带状结构^[1]。对于此类云系的分析,基本是通过对不同通道云图的形状来进行判断。但是目前还没有比较好的卫星云图分析方法,主要通过人工的目视判读法等,不仅速度慢而且精度低。

[0003] 目前卫星云图中有多种通道,但是一般都只使用红外云图来进行云图分析。因为气象卫星在白天和夜晚都可以进行红外云图的探测,它是通过对物体、大气辐射的红外区域的波长探测获得,只要云系有温度就可以被检测出来,和光照无关。在成像过程中,温度越高的区域亮度越高,温度越低的地区,亮度越低。云层的温度和周围大气的温度有明显的区别,这样我们就可以通过红外辐射判断出云的形状。与之类似的有可见光云图,它通过云层对光照的反射直接测量获得,可见光云图的优点是它探测的云的形状纹理等特性较好(好于红外云图),但是可见光云图在光照不充分的情况下无法进行常规探测。事实上在实际生产中,存在对除了红外云图其他通道利用率低的情况。

[0004] 因此,因此寻求一个能够更充分的利用卫星云图信息,并且高精度实时的卫星云图检测算法具有重要的研究意义。

发明内容

[0005] 本发明的目的是针对现有技术的缺陷,提供了一种基于Yolo的气象卫星云图目标检测方法,其将红外云图和可见光云图进行融合后,采用Yolo算法进行检测识别出云图中的典型天气现象;能够有效利用可见光云图中的信息,提高卫星云图的检测速度和检测精度。

[0006] 本发明提供的技术方案为:

[0007] 一种基于Yolo的气象卫星云图目标检测方法,包括如下步骤:

[0008] 步骤一、从气象卫星采集的原始卫星云图数据中提取出多张红外云图和多张可见光云图;

[0009] 其中,所述红外云图与所述可见光云图为一一对应关系;

[0010] 步骤二、统计所述可见光云图中的无效信息,并且根据所述可见光云图中的无效信息占比将所述多张可见光云图划分为:第一可见光云图、第二可见光云图和第三可见光云图;

[0011] 其中,所述第一可见光云图的无效信息占比 ψ_1 满足 $\psi_1 < \psi_{\min}$;

[0012] 所述第二可见光云图的无效信息占比 ψ_2 满足 $\psi_{\min} \leq \psi_2 \leq \psi_{\max}$;

[0013] 所述第三可见光云图的无效信息占比 ψ_3 满足 $\psi_3 > \psi_{\max}$;

[0014] 式中, ψ_{\min} 为无效信息阈值下限, ψ_{\max} 为无效信息阈值上限;

[0015] 步骤三、将所述第一可见光云图与其对应的所述红外云图进行融合,得到融合云图;以及

[0016] 将与所述第一可见光云图相对应的红外云图、所述融合云图以及所述第三可见光云图组成待检测云图集;

[0017] 步骤四、采用Yolo算法对所述检测云图集的云图进行目标检测,识别出云图中的典型天气现象。

[0018] 优选的是,在所述步骤二中,使用开源opencv库读取可见光云图的所有像素点,并判断所述可见光无用像素点的数量占总像素点的比重,得到所述可见光云图中的无效信息占比。

[0019] 优选的是, $\psi_{\min} = 25\%$, $\psi_{\max} = 75\%$ 。

[0020] 优选的是,在所述步骤三中,所述第一可见光云图与其对应的所述红外云图进行融合,包括:

[0021] 步骤1、将待融合的所述第一可见光云图和所述红外云图缩放为相同尺寸;

[0022] 步骤2、采用拉普拉斯金字塔分别对所述第一可见光云图和所述红外云图进行多尺度表达后,对所述第一可见光云图和所述红外云图进行分层融合;

[0023] 其中,在每个层次进行融合时,根据位置信息将第一可见光云图中有用像素点所在的区域与红外云图中的其他区域融合,得到融合云图;

[0024] 所述其他区域指红外云图中对应光云图中无用像素点所在位置的区域。

[0025] 优选的是,在所述步骤四之前还包括:根据卫星云图数据制作COCO数据集格式的训练数据集,对Yolo算法模型进行训练,并对训练模型进行速度和检测精度测试后,得到Yolo算法气象云图检测模型。

[0026] 优选的是,所述的基于Yolo的气象卫星云图目标检测方法,还包括对所述Yolo算法气象云图检测模型进行Web端部署,使用Flask框架作为前端框架,Vue框架作为后端框架;以实现在web端上传云图进行检测的功能,达到实时检测的效果。

[0027] 优选的是,在所述步骤四中,采用Yolov3或Yolov5算法对所述检测云图集的云图进行目标检测。

[0028] 优选的是,在所述步骤四中,采用Yolov5算法对所述检测云图集的云图进行目标检测。

[0029] 优选的是,所述典型天气现象包括:冷锋、暖锋、台风、气旋、强对流和激流云系6种天气现象。

[0030] 本发明的有益效果是:

[0031] (1) 本发明提供的基于Yolo的气象卫星云图目标检测方法,将可见光云图和红外

云图进行融合,不仅吸取了可见光云图纹理清晰的优点,也从红外云图弥补了可见光云图黑夜不可测量的缺陷;综合两种云图能够获取更为完整、准确的气象数据。

[0032] (2) 本发明提供的基于Yolo的气象卫星云图目标检测方法,能够有效提高卫星云图的检测速度和检测精度,并且达到实时监测的效果。

附图说明

[0033] 图1为本发明所述的右半部分为有用信息的可见光云图。

[0034] 图2为本发明所述的融合前的左侧有部分有用信息的可见光云图。

[0035] 图3为本发明所述的融合前的红外云图。

[0036] 图4为采用本发明中的融合方法融合后的云图。

[0037] 图5为本发明所述的YOLOv5的算法结构图。

[0038] 图6为本发明所述的CBL结构图。

[0039] 图7为本发明所述的Res unit结构图。

[0040] 图8为本发明所述的SPP结构图。

[0041] 图9为本发明所述的YOLOv5算法主要流程图。

[0042] 图10为本发明所述的召回率图像。

[0043] 图11为本发明所述的mAP 0.5图像。

[0044] 图12为本发明所述的mAP 0.5:0.95图像。

[0045] 图13为本发明所述的精度图像。

[0046] 图14为本发明所述的气象云图检测结果示意图。

[0047] 图15为本发明所述的MVVM框架示意图。

具体实施方式

[0048] 下面结合附图对本发明做进一步的详细说明,以令本领域技术人员参照说明书文字能够据以实施。

[0049] 本发明提供了一种基于Yolo的气象卫星云图目标检测方法,针对红外通道(波长10.5~12.5微米)和可见光通道(波长0.4~0.78微米)对不同天气现象的特征,进行两个通道图像的融合,之后使用YOLO算法识别出云图中的典型天气现象。

[0050] 一、可见光云图与红外云图融合方法。

[0051] 首先,通过观察VIS图像(可见光云图),可以看出图片中存在着两种无法得到有用信息的数据,一种是由于没有光照无法测量到的数据,背景呈现大面积蓝色;一种是损坏的数据,背景呈现大面积黄色。对于这两种数据,采取直接丢弃并使用红外云图作为检测数据集代替。

[0052] 对于部分可见光云图部分存在着大面积的背景信息,但还是有许多有用的信息(气象信息),如图1所示,可见光云图右半部分的信息属于有用信息,但是左半部分信息由于没有光照无法探测到。对于右半部分的信息如果简单抛弃则浪费了云图资源,所以应该采用图像融合的算法,使用红外云图补全可见光云图左半部分的丢失信息然后对融合后的图片进行增强处理。

[0053] 在本实施例中,采用图像像素阈值化的处理方法,对图像进行判断,判断哪些图像

直接丢弃,哪些图像进行融合,哪些图像直接使用。

[0054] 图像像素阈值化的处理方法流程是:首先对两种图片(可见光云图和红外云图)使用python的cv2函数库中imread函数读取图片,然后使用resize函数把图片进行缩放,目的是让红外云图和可见光云图尺寸大小相同,方便以后的处理。之后读取图片中云图的所有像素点把图片数据转化格式为纯数据格式,判断无用像素点的数量占图像总体的比重,如果超出一定的阈值,则说明可见光云图中包含的无用信息过多,图片直接丢弃,使用红外云图作为替代,如果低于一定的阈值,说明可见光云图中包含的无用信息不多,可以直接利用,如果在两个个阈值之间,说明图片中无用信息和无用信息占比相近,可以使用红外云图填充可见光图中的无用信息。

[0055] 在本实施例中,118张卫星云图可见光通道图片中无用信息的占比情况为:47张图片10%以下,7张图片20%以下,11张图片30%以下,6张图片40%以下,6张图片50%以下,5张图片60%以下,2张图片70%以下,4张图片80%以下,4张图片90%以下,26张图片99%以下;可见图片中有相当多的图片存在着大量无用信息。在设置阈值下限25%,阈值上限75%的情况下,有63张图片包含的无用信息较少,33张图片包含无用信息较多,22张图片二者比例相近。对于无用信息较多(高于阈值上限)的可见光云图直接采用红外云图进行替换融合,对于无用信息较少(低于阈值下限)的可见光云图直接使用,无用信息信息适中(处于阈值上限和下限之间)的可见光云图要和红外云图进行深度融合。

[0056] 本发明采用改进的拉普拉斯融合算法对可见光云图要和红外云图进行深度融合。

[0057] 图像金字塔是对于同一种图片的不同分辨率表示,可以表达多种尺度的图片,所以图像金字塔可以用于图片的缩放,图像分割等领域。它对图像进行多尺度表示的原理是进行不断的卷积操作和下采样操作,在每次循环操作的过程中,图片的尺度会被缩小,同时它的表达能力也得到增强。越靠近金字塔顶层的图片对图片的抽象度越高同时损失的信息也越多。为了弥补在下采样过程中信息损失过多的缺陷,拉普拉斯算法可以对传统金字塔进行改进,通过在每层获得金字塔的图像之后,用上一层图像的上采样和本层的图像进行线性计算生成新的图像,从而让图像的细节得到了保留。

[0058] 在上采样的具体过程中,首先通过插值让图片尺寸变大,具体操作是通过判断行列数的奇偶性插入空白占位符,然后使用卷积核进行滤波处理。这样上采样得到的结果和下层的尺寸相同,再把二者进行线性运算就得到了拉普拉斯金字塔。

[0059] 拉普拉斯金字塔进行图像融合的方式是首先把图片分解到不同的尺度上,然后在各个层次上分别融合。首先从顶部的金字塔进行融合,融合的方法是计算两个图片中相同位置像素点的梯度,然后和周围像素点进行比较,如果在两幅图片中,其中一幅图片的像素梯度较大,就在融合后的图片中使用相应的图像的像素点,因为梯度大说明图像边缘纹理等信息更为丰富,更应该得到保留。在其他层次的处理过程中,也是通过结合上层的图片信息并进行矩阵运算得到梯度信息,然后进行比较,选择两个图片中梯度较大的作为融合的结果。

[0060] 本发明是进行红外云图和可见光图的融合。在可见光图中,光照是连续的,所以无用信息基本集中在了图片的某个区域,例如全部在左半部分或者右半部分。因此本发明针对可见光云图的特点对传统的融合算法进行了改进。首先对可见光图和红外云图进行拉普拉斯运算,然后在每个层次进行融合的时候,通过位置信息决定融合后的图像使用原图像

中哪个图像的像素信息。在本实施例中,结合上一阶段的实验中选择的75%和25%作为融合阈值,本阶段使用了可见光云图的1/4和红外云图的3/4进行融合。改进的方法具体来说是在每层的融合过程中放弃使用梯度决定的传统模式,而是把可见光云图的1/4直接作为融合后图像前半部分的像素,红外云图的3/4作为融合后图像后半部分的像素,之后分层融合进行处理。

[0061] 由于本发明数据集图片大小为1222*916,融合需要输入2的n次方的图片大小,所以实验首先把图片缩放到512*512,融合完成后再增大到1222*916。

[0062] 融合前的可见光图和红外云图分别如图2和图3所示,融合后的图像如图4所示。可见融合后的图片拼接自然,两张图片的信息都有所保留,处理效果较好。

[0063] 利用本发明提供的方法进行可见光云图和红外云图的融合后,不仅让图片资源的利用率得到了提高,而且让图片信息更为丰富,为下一步图像检测识别做好了铺垫。

[0064] 二、采用YOLO算法对卫星云图进行目标检测建模,用以识别冷锋、暖锋、台风、气旋、强对流和激流云系共计典型的6种天气现象。

[0065] 1、卫星图像HDF5数据预处理

[0066] hdf是一种新型的数据格式,它具有很多优点,首先可以在单个hdf文件中存储所有数据,其次不同格式的数据都可以存入一个hdf文件,例如符号数据、数值数据、图形数据等。用户可以对存储的数据进行不同层次的划分,对每一层可以添加描述信息。分层可以按照用户自己的需要进行,较为灵活。同时hdf文件和其他格式兼容性比较好,可以加入新的数据模式。在不同平台之间传递hdf文件也不需要额外的数据转换操作,因为它是和平台无关的格式,在不同平台都可以操作相同的hdf文件。

[0067] 而在卫星云图数据中,hdf格式使用较多。首先hdf文件可以存储图像的图片信息,对于图像的单位信息还可以加以说明。其次对于卫星云图不同通道的信息可以在hdf文件分层次进行存储,方便用户读取。本发明采用的实验云图数据采用了hdf格式的数据。

[0068] 本发明使用的卫星云图数据是从全圆盘图截取北纬0-60度、东经70-150度范围内云图,生成等经纬度投影文件(hdf文件)。每个hdf文件包含14个数据集,存储14个通道的等经纬投影数据。数据集名称为:LonLatChannelB04、LonLatChannelB05、LonLatChannelB06、LonLatChannelB09、LonLatChannelB10、LonLatChannelB11、LonLatChannelB12、LonLatChannelB14、LonLatChannelB16、LonLatChannelIR1、LonLatChannelIR2、LonLatChannelIR3、LonLatChannelIR4、LonLatChannelVIS。数据集名称的后3个字符表示通道名称,如B04,IR1,VIS。数据集首行首列的数据表示北纬60度、东经70度处的数据,数据集末行末列的数据表示北纬0度、东经150度处的数据。其中VIS通道的数据集为3666行、4888列,其他通道数据集均为916行、1222列。所有数据集的数据类型都是16位大端无符号整数。B04、B05、B06和VIS数据集中数据的单位是ALBEDO(%),B09、B10、B11、B12、B14、B16、IR1、IR2、IR3、IR4数据集单位是KELVIN。

[0069] 使用python的h5py库,步骤如下:

[0070] (1) 首先使用h5py.File(“路径名”)读取文件

[0071] (2) 按照字典格式读取通道的文件

[0072] (3) 使用matplotlib库显示图片。通过函数pyplot.imshow()和函数pyplot()函数可以简单地显示图像。

[0073] 以“H8_ALL_PROJ_L2_20190114_0500.hdf”文件为例,读取并显示B04通道图像。其中,图像宽度为1222,高度为916,使用了索引颜色模式。

[0074] 之后需要把显示的图片按照原来的分辨率保存下来,来用于之后神经网络的训练。如果直接使用matplotlib.pyplot中的savefig函数保存图片,那么保存的图片不仅仅含有坐标轴,而且含有上下左右的空边,这样会让训练的输入和我们的预期不相符,图片的分辨率大于真实数据显示,标记文件也会错位。所以本发明进行了matplotlib库的一些设置。

[0075] 首先使用set_size_inches函数对图片大小进行设置,然后使用去掉坐标轴显示,之后在用subplots_adjust函数和margins函数去掉图片四周的空白,最后再用savefig函数保存图片,得到了理想的图片。

[0076] 最后循环读取每个文件以及他们的每个通道数据,把他们保存为文件。

[0077] 由有关气象专家提供的标记文件为csv格式的标记文件,包括type、name、top、left、bottom、right、filename7个字段。为了进行训练,需要type、top、left、bottom、right、filename6个字段。

[0078] 在“H8_ALL_PROJ_L2_20190114_0500.hdf”文件中,可以截取部分样本简单显示一下。

[0079] 2、构建目标检测标记数据集

[0080] YOLO算法基于一个前馈卷积神经网络完成。YOLOv5使用了CSPDaeknet (跨阶段局域网络) 作为Backbone,并且使用了PANET (路径聚合网络) 和SPP (空间金字塔池化) 作为Neck,并且使用和YOLOv3一样的Head。在YOLOv5中,各个部分的激活函数有所区别,在中间层和隐含层使用的是Leaky ReLU,在检测层使用Sigmoid激活函数。这些激活函数在计算代价上相比v4算法更小。YOLOv5配置了两种优化函数,并分别给出了相应训练使用的初始化超参数。随机梯度优化,对于大型数据集的训练效果更好,而Adam优化器对较小自定义数据集的训练效果更好

[0081] YOLOv5网络主要由4个组件组成:

[0082] 1) 输入层:对输入图像进行数据增强处理

[0083] 2) Backbone:通过对图像进行不同分辨率大小上的聚合形成图像更为丰富的特征

[0084] 3) Neck:对输入的图像特征进行混合处理

[0085] 4) Predication:预测边界框以及类别的置信度

[0086] 算法结构如图5所示。

[0087] 各个组件结构和张量计算如下:

[0088] (1) CBL (如图6所示):对传统的卷积操作等进行组合生成新的网络部件,这是网络中的最基本的组件。

[0089] (2) Res unit (如图7所示):在结构中存在不同数目连续的残差结构,更多的res_unit可以让网络可以构建得更深

[0090] (3) Concat:把两个张量按照某个维度组合成一个张量。

[0091] (4) Add:把两个张量在相同维度上进行加法操作,生成同样大小的结果

[0092] (5) Spp结构 (如图8所示):首先在不同尺寸分别进行采样操作,之后把不同的结果通过计算融合到一块,最后再次进行卷积修改输出的大小。

[0093] 本实施例中采用单阶段检测YOLOv5算法,其主要流程如图9所示。

[0094] YOLOv5的数据集训练需要COCO数据集,本发明首先制作VOC2007数据集,之后进行转化。

[0095] 新建文件夹,用来存储整个数据集:

[0096] 对图片进行标记的工具很多本发明选择了labelimg开源工具,保存之后,生成一个同名的xml格式的文件,用来保存标记信息。其中object字段用来保存每个样本信息,文件中存在两条重要的数据,其中的bndbox保存目标的位置,name保存目标的类别。

[0097] 由于本发明使用了大量的标记数据,若使用此工具进行逐条标记,将会较为繁琐,所以本发明使用了一个简单的python程序从前文中的存储标记的csv文件中读取需要的6个字段并根据上图所示文件格式生成对应的xml标记文件。

[0098] 然后把xml标注文件放入相应的文件夹中。

[0099] 使用txt文件中存放每个类别使用的训练文件的名称,本发明使用了一个简单的python程序生成,之后需要把制作的数据转化为coco数据集,也是通过一个python文件实现。在每个train或val目录下存放着相应的图片名或标签文件名。

[0100] 3、训练参数设置

[0101] 在准备好数据集之后,对模型进行进一步参数配置,本实施例用了YOLOv5网络精度最高的YOLOv5x模型。

[0102] (1) 设置模型配置文件

[0103] 根据yolov5x.yaml文件修改待检测物体的类别为6。

[0104] (2) 设置训练配置文件

[0105] 根据coco.yaml文件修改训练文件夹和验证文件夹路径,物体的类别以及每种类别的标记。

[0106] 4、训练与评估

[0107] 训练环境配置如下:

[0108] (1) 操作系统:CentOS Linux release 7.8.2003 (Core)

[0109] (2) 显卡:8块TITAN X显卡(实验使用了其中2块)

[0110] (3) cpu:56个

[0111] (4) 内存:256G

[0112] (5) cuda版本:11.0

[0113] (6) pytorch版本:1.7.1

[0114] 算法在训练之前设置超参数。

[0115] 训练流程:

[0116] (1) 设置DDP模式的参数

[0117] 可以设置全局进程的个数,进程编号等。

[0118] (2) 设置是否恢复训练

[0119] (3) 加载处理超参数列表,

[0120] (4) 模型训练

[0121] 首先获取训练日志的路径,需要读取result文件,result文件包括了每次迭代时保存的训练结果,对上一次运行结果进行分析之后可以更好地进行下一步的实验。之后需

要进行一些模型的设置,例如设置下保存ground truth的参数,一般比3小的时候会被保存,训练时会产生日志文件,算法采用evolve文件夹进行保存。设置在每次迭代过程中产生的权重文件保存的路径,设置每次训练的结果保存的路径。然后从程序中获取一些信息,例如训练的轮次、数据集输入的批次、训练的总批次(分布式训练)、训练一次图片之后产生的权重文件、进程序号(分布式训练)。然后需要保存运行时的一些设置,保存超参数文件hyp和项目的命令参数opt。分辨cuda的类型是否采用了GPU进行训练,获取识别目标的类别数量和类别名字,如果在命令行参数中设置了single_cls那么待检测目标都属于一类。接着读取pt格式的模型参数文件。如果选择采用预训练,那么可以调用脚本从网络自动下载预训练模型,之后创建模型,也可以用命令行或者配置文件设置参数。使用文件或者命令行区别就在于是否设置了恢复参数,若是会按照配置文件创建模型;参数的设置会导致锚框参数中关键字的设置,即是否加载预定义锚框的问题,要是训练设置了恢复参数,会在上次训练权重参数的基础上来继续训练;文件已经为用户提供了数据集默认的训练参数,默认是使用文件中自定义权重,如果没有设置参数的话,用户直接设置锚框再加载预训练权重进行训练,用户自定义的设置就会被文件设置覆盖,文件的设置具有更高的优先级;所以在程序运行之前需要设置决定使用哪种参数,若采用权重文件也进行训练过程的话,权重文件种锚框的位置就被丢弃使用用户自己定义的锚框;若设置了恢复参数,那么就不丢弃参数,把权重和锚框配置文件一起加载。还需要设置冻结层模型,设置模拟的处理批次的图像大小batch_size,选择使用合适的优化器,并设置pg0组的优化方式,设置训练过程的参数weight、对特征批量归一化的bn的优化方式,设置偏置参数biases的优化方式,之后把优化的信息打印出来。还需要设置学习率衰减速度,模型会使用余弦退火的方式进行衰减之后恢复训练。初始化最开始的一代训练数据选择是否使用之前最好的训练结果进行继续训练。然后根据best_fitness来把训练过程中最好的参数文件的权重保存为best.pt。然后进行判断,如果之前训练过程已经完成了那么就进行新一轮的实验,进行参数的设置,设置加载优化器和best_fitness,加载训练结果result.txt,加载训练的轮次。如果设置了恢复参数还要备份权重,尽管目前的恢复参数可以近似完美不出错,但是在参数的恢复过程中可能出现别的问题丢失了以前的权重,采用额外的备份操作可以让数据更为安全。若设置的训练次数epoches比较大,那么新设置的epoches为需要再训练的轮次数而不是总的轮次数。然后从模型结构文件读取模型的总步长和模型输入图片分辨率,检查用户设置的输入图片分辨率确保能够整除步长gs。然后查看是否设置采用分布式训练,设置rank进程号和以及是否使用仅支持本机多卡的DataParallel模式,当rank设置为-1并且gpu只有一个,会跳过分布式训练。之后同步BatchNorm,使用跨卡同步bn。为模型创建EMA指数滑动平均,如果GPU进程数大于1,这个指数就不用创建。然后创建训练集的dataloader,获取标签中最大的类别值,并和类别数做比较,如果大于类别数则表示出现了问题。这之后开始设置模型。根据自己的数据集的类别数设置分类损失的系数,设置类别数,超参数。然后设置giou的值在objectness loss中做标签的系数,根据labels初始化图片采样权重,获取类别的名字。接着开始设置类别的频率,将所有样本的标签拼接到一起,统计后做可视化。获取所有样本的类别,然后根据这个统计,设置长宽和中心位置做可视化。计算默认锚点anchor与数据集标签框的长宽比值。如果标签框满足上面条件的数量小于总数的99%,那么会使用一些无监督机器学习方法获取新的锚点设置。

[0122] 之后开始模型的训练。设置热身训练的迭代次数,初始化mAP和results。设置学习率衰减所进行到的轮次,目的是在打断训练后,恢复参数可以接着训练也能正常的衔接之前的训练进行学习率衰减。然后打印训练和测试输入图片的分辨率,加载图片时调用的cpu进程数,判断epoches开始的序号数。然后进入每个epoch的循环训练。如果设置进行图片采样策略则根据前面初始化的图片采样权重model.class_weights以及maps配合每张图片包含的类别数,通过random.choices生成图片索引indices从而选择采样。如果设置了DDP模式,则广播采样策略。然后初始化训练时打印的平均损失信息。若在DDP模式下打乱数据,ddp.sampler的随机采样数据是基于epoch+seed作为随机种子,所以每次epoch不同,随机数种子也不同。然后使用tqdm创建进度条,方便训练时信息的展示。然后计算迭代的次数iteration。在前nw次迭代中使用热身训练,并通过一定的方式选取accumulate和学习率。偏置参数bias的学习率从0.1下降到基准学习率lr*lf(epoch),其他的学习率从0增加到lr*lf(epoch)。lf为之前设置的余弦退火的衰减系数,参数momentum从0.9逐渐增加。然后进行多尺度训练,在imgsz*0.5,imgsz*1.5+gs中随机选取尺寸。然后进行混合精度向前传播,并计算损失包括分类损失,objectness损失,框的回归损失,loss总损失值。loss的值存储loss_items为一个元组,包含分类损失,objectness损失,框的回归损失和总损失。之后进行反向传播。模型反向传播一定次数后再根据累积的梯度更新一次参数然后进行学习率衰减。如果是单GPU或者cpu进行训练,还需要更新mAP,包括更新EMA属性,添加include的属性。判断该epoch是否是最后一轮。然后对测试集进行测试,计算mAP等指标。同时负责将指标写进result文件,如果设置了bucket命令行参数还需要把result文件上传到网络。然后设置可视化工具Tensorbord,添加模型指标,训练损失等信息到tensorboard显示。更新最好的mAP通过best_fitness。之后进行保存模型的工作,此外还需要保存epoch,results,optimizer等信息,optimizer将在最后一轮完成后保存。其他的参数,例如model保存的是EMA的模型。在模型训练完成之后,使用strip_optimizer函数将optimizer从ckpt中剔除,并且为了减少变量存储,对模型进行model.half操作,将Float32的模型变成Float16,可以减少模型大小,提高inference速度。结束之后对results文件进行可视化操作。最后释放显卡的显存。

[0123] 对超参数进行进化:

[0124] 在conda环境中,使用train.py文件,设置device为“cuda0,1”,设置好参数配置文件,模型迭代的批数据量大小设置为16。

[0125] 在经过了1小时13分钟35秒的300次epoch的训练之后得到了结果。

[0126] 实验召回率图像如图10所示,实验最高达到96%以上。mAP_0.5图像如图11所示,最高达到97%以上;mAP_0.5:0.95图像如图12所示。精度图像如图13所示,最高精度接近99%。

[0127] mAP即mean AP,对神经网络结果进行评价的时候这个值反映了模型的优良程度,mAP值越高,对应的分类算法精度更高,性能更好。mAP的计算方法是,首先计算出每种目标检测的AP值,然后对它们进行加权平均。mAP最小为0,最大为1,数值越大越好。这个判断标准在目标检测算法最主要。本实验使用的YOLOv5x模型,所有类别平均精度值接近99%。同时本实验的精度、召回率、PR曲线等都达到了不错的效果。在取得良好训练结果的同时,我们的训练速度非常快,可以达到30ms左右。

[0128] 在另一实施例中,在相同条件下使用YOLOv3算法进行了实验,YOLOv3算法在精度和召回率等方面稍微弱于YOLOv5,但是YOLOv3的训练时间较短,在相同条件下使用了49分钟35秒。即在同样迭代了300次情况下,YOLOv3算法训练时间比YOLOv5算法减少了24分钟,召回率提高了1.43%,但是YOLOv3算法精度比YOLOv5下降了3.75%。所以如果对精度要求不是特别高的情况下,可以选择使用YOLOv3算法进行更快的训练。

[0129] 5、结果测试与分析

[0130] 测试过程如下所述,首先判断是否是在训练时调用test,如果是则获取训练时的设备。否则直接调用,选择设备。删除之前的test_batch0_gt.jpg和test_batch0_pred.jpg。然后加载模型,检查输入图片的分辨率是否可以被32整除。如果设备不是cpu并且GPU的数目为1,则将模型由Float32转化为Float16,以提高向前传播的速度。然后进行配置,加载数据配置信息,设置iou阈值,从0.5到0.95每间隔0.05取一次。然后设置Dataloader,把rect参数设置为true因为yolov5的测试评估是基于矩形推理的。

[0131] 初始化测试参数和路径索引信息。设置tqdm进度条的显示信息。初始化指标,时间。初始化测试集的损失,初始化json文件的字典统计信息,ap。然后禁用梯度传播,让模型进行向前传播,累计计算向前传播的时间。计算损失,如果是在训练时进行的test,需要通过训练结果计算并返回测试集的GIoU,obj,cls损失。之后设置置信度阈值,iou阈值并进行NMS处理,计算累计处理NMS的时间。

[0132] 然后为每一张图片进行统计,写入预测信息到txt文件中,生成json文件字典,统计tp等。首先获取第si张图片的标签信息,包括class,x,y,w,h。获取标签类别,统计测试图片数量,如果预测为空,那么添加空的信息到stats中,保存预测结果到txt文件。通过获得对应图片的长和宽,然后根据图片名字设置txt文件的路径。将预测框的坐标调整到基于其原本长宽的坐标。保存类别和坐标到txt文件。修正预测坐标到图片内部。保存coco格式的json文件字典。首先获取图片id,获取框,将框调整为基于原图大小的,转换为xywh格式。格式的通道次序发生了变化,coco的json格式中的框坐标格式为xyzh,此处的xy为左上角坐标,也就是coco的json格式的坐标格式为:左上角坐标+长宽,所以需要中心点坐标转换到左上角。之后初始化预测评定。存放已检测到的目标,获得xyxy格式的框并乘以wh,对图片中的每个类单独处理,计算标签框该类别的索引和预测框该类别的索引。计算预测框的iou值,选出最大的ious值。然后把检测到的目标添加到detected,获取不同iou阈值下的true positive。然后将每张图片的结构统计到stats里面。画出第一个batch的图片的ground truth和预测框并保存。然后将stats列表的信息拼接到一起。打印指标结果并细节展示每一个类别的指标。打印向前传播耗费的时间、nms的时间、总时间。采用之前保存的json格式预测的结果,并通过cocoapi评估指标,测试集的标签也需要转成coco的json格式。首先获取图片id,获取预测框的json文件路径并打开。获取并初始化测试集标签的json文件。初始化预测框的文件。创建评估器。评估,然后展示结果。最后返回测试指标结果。

[0133] 本发明对所有的图片进行了一次检测,可以在图片中框出云的种类以及它们的置信度,其中用0表示冷锋、1表示暖锋、2表示台风、3表示气旋、4表示强对流、5表示急流云系。云图检测效果如图14所示。图14中,两个位置用置信框圈出检测出的目标所属的类别为3(气旋),以及显示预测的概率值分别为0.87和0.91。

[0134] 三、模型部署与应用

[0135] 1、模型Web端部署

[0136] (1) Flask框架

[0137] Flask是一个python语言中的轻量级框架,提供了web框架最基本的功能,相较于其他的框架使用更自由灵活。它的核心比较简单,只提供了基础的通信功能,用户需要自己选择拓展去实现特定的需求。把功能从框架中分离出来就不用让用户每次都安装不必要的功能,而是根据自己的选择去添加。

[0138] Flask具有巨大的优势,它让用户开发有了更多的选择的权力,设计风格极简,使得用户以更大的自由度构建应用程序。首先Flask的设计轻量化、模块化,使用各种丰富的拓展可以设计出符合需要的程序。其次Flask的API的基础UI设计美观连贯,让简单的设计就可以产生漂亮的界面。Flask模型小,得益于python平台在实际环境中较为容易部署。此外,Flask支持各种http请求处理的功能,框架高度灵活,配置灵活,可以满足多种输出需求。

[0139] 本发明需要对神经网络框架进行web端部署,仅仅需要使用部分网络功能,所以后端框架中Flask就成了一个比较好选择,可以在微框架上添加所需要的功能。此外,由于本发明使用的神经网络框架平台是python语言编写,使用Flask可以更方便原生地对二者进行结合。

[0140] (2) Vue框架

[0141] Vue采用MVVM架构,即View(视图)、ViewModel(逻辑)、Model(数据),如图15所示。MVVM框架里面视图层和数据层不直接交换数据,而是通过ViewModel层交换。Vue采用了数据的双向绑定技术,ViewModel负责去监听用户数据,然后在发生变化时通知视图层。用户操作通过ViewModel操作视图层时,也会通知相应的数据进行持久化操作。MVVM架构为前端页面提供了分层次的架构抽象,他还可以结合其他js库例如Ajax等实现数据通信和持久化,丰富了系统的功能和体验。Vue以MVVM中的ViewModel为核心,采用双向数据绑定技术保证了M层和V层数据的一致性,并且它属于轻量级框架,让开发过程简洁高效。Vue采用独特的数据驱动方式,改变了之前的DOM视图手动更新的方式,只要数据发生改变,就会触发视图自动变化。此外Vue让视图组件化,一整个网页被拆分为不同的区块,每个区块是一个组件。网络被多个组件拼接嵌套组成,这样实现了代码更好的可维护性和可重用性。

[0142] Vue框架有许多特性。首先它是一个轻量级的前端框架,可以对模板中的表达式和计算属性自动跟踪计算,使用了MVVM架构以及拥有丰富的组件系统,接口使用多样化。其次它实现了数据的双向绑定,使用模板和指令在文档对象模型进行数据操作,让DOM实现了自动实时渲染。Vue拥有丰富的指令系统,在开发中页面中的事件几乎都可以使用Vue中的指令描述完成。指令可以结合变量的值发生变化,并相应地结合到DOM的应用上。Vue还拥有强大的组件功能,对原始的HTML中元素进行拓展。它通过封装实现了代码重用。同时有组件间继承式的通信方式,父子组件在属性参数中传递消息,从父组件到子组件数据单向传递,子组件通过消息响应令父组件更新数据。这样组件的开发就和html和javascript有密切的关系,用户可以在组件中选择需要的进行整合,加快了开发过程,减少了代码量,同时组件的开发支持热重载。Vue使用了Vue-router插件,这个插件可以构建只有一个页面的应用。路由插件设置了用户对网页浏览的路径,把用户对路径的访问映射到自定义组件上,相较于传统的使用超级链接的方式,这种设置实现了单页面应用

[0143] 用户访问的时候Vue显示视图,用户再次向服务器发出上传图片的请求。服务器接收客户端解析并解析,flask把接收的图片保存到upload目录下,之后调用yolov5的相关函数实现图像识别,最后回传给Vue对视图进行更新。

[0144] 对训练得到的模型进行了Web端部署,使用了Flask框架作为前端框架,Vue框架作为后端框架,通过训练好的权重文件对图片进行检测。完成了在web端上传云图进行检测的功能,达到了实时检测的效果。

[0145] 本发明提供的卫星云图多通道图像融合方法,针对红外通道(波长10.5~12.5微米)和可见光通道(波长0.4~0.78微米)对不同天气现象的特征,进行两个通道图像的融合:a)使用开源opencv库统计可见光通道云图中存在的无效的信息并设置阈值进行过滤。丢弃无效信息过多的图片并完全保留无效信息较少的图片,对处于中间区域的图片进行融合;b)对红外云图和可见光云图分别进行拉普拉斯运算,在高斯图像金字塔的基础上建立拉普拉斯图像金字塔。之后提出了一种新的融合策略。根据a)中的阈值设置两种云图融合的比例和相对位置,对各层进行基于位置的像素级别的融合,再对图片进行线性填充得到融合后的结果。

[0146] 在图像融合的基础上,采用YOLO算法对卫星云图进行目标检测建模,用以识别冷锋、暖锋、台风、气旋、强对流和激流云系共计典型的6种天气现象。从原始卫星云图数据(HDF5格式)中提取出各个通道的云图图片,进行可见光云图与红外云图的融合。接着制作COCO数据集格式的训练数据集,进行模型训练,并对训练模型进行测试。在本实施例使用的环境中,YOLOv5模型可以达到30FPS左右的图片检测速度和95%以上的检测精度,很好地满足了卫星云图检测速度快和精确度高的要求。

[0147] 本发明还对训练得到的模型进行了Web端部署,使用了Flask框架作为前端框架,Vue框架作为后端框架,通过训练好的权重文件对图片进行检测。完成了在web端上传云图进行检测的功能,达到了实时检测的效果。

[0148] 尽管本发明的实施方案已公开如上,但其并不仅仅限于说明书和实施方式中所列运用,它完全可以被适用于各种适合本发明的领域,对于熟悉本领域的人员而言,可容易地实现另外的修改,因此在不背离权利要求及等同范围所限定的一般概念下,本发明并不限于特定的细节和这里示出与描述的图例。



图1

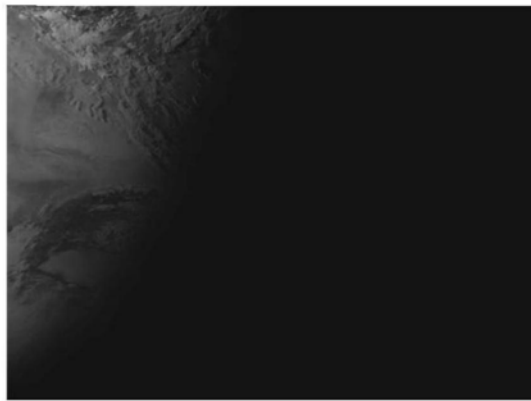


图2

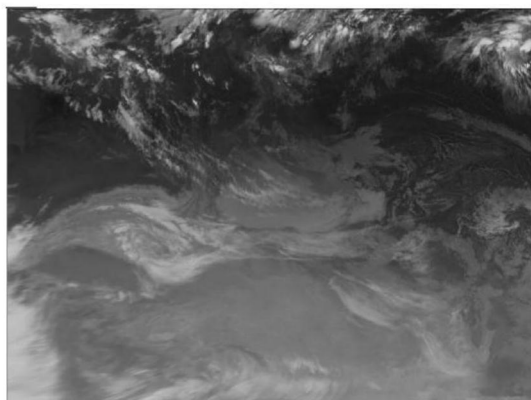


图3

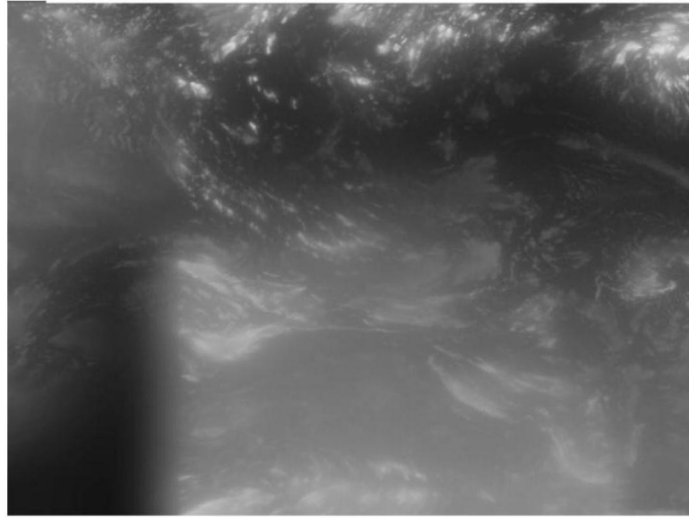


图4

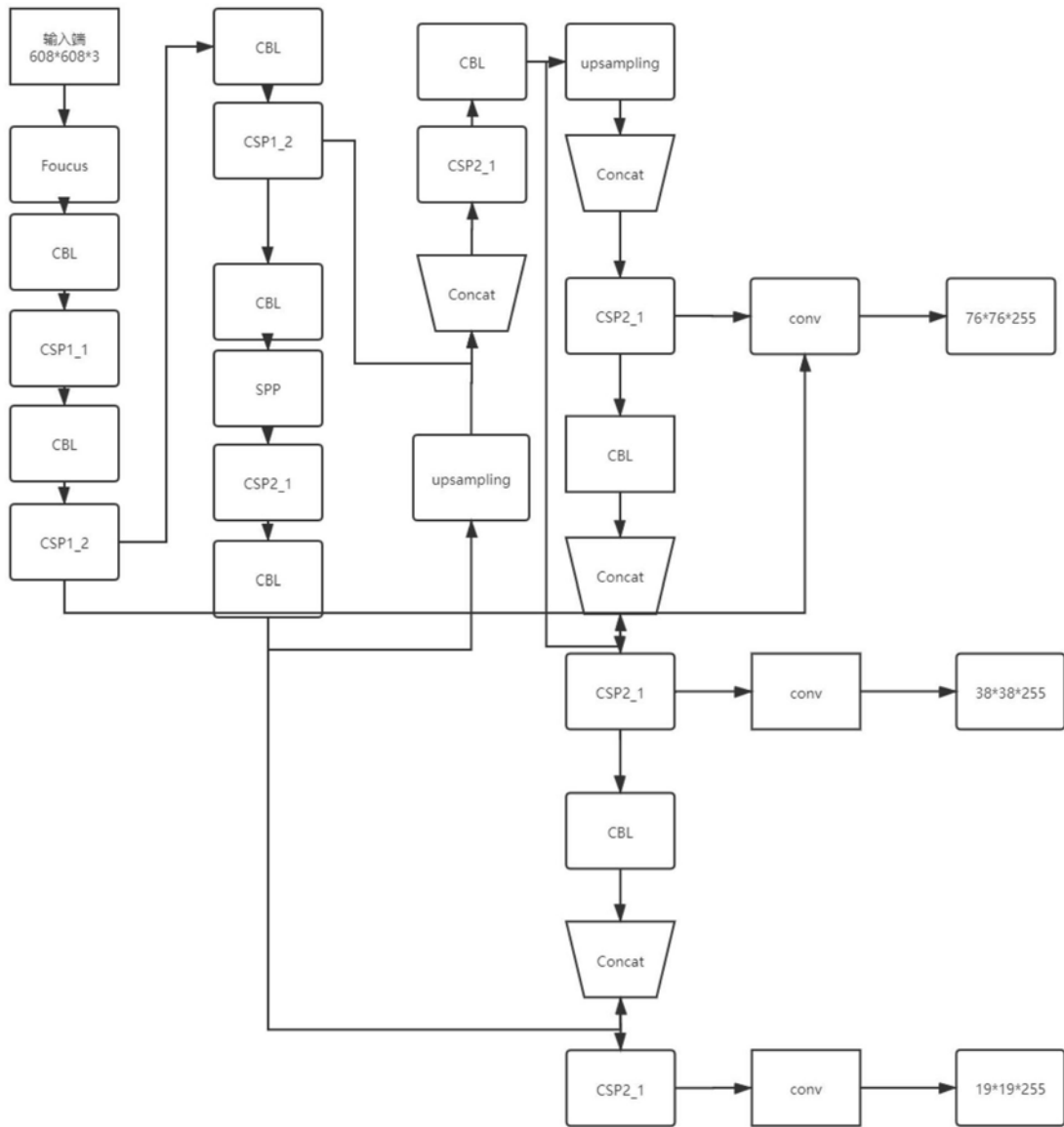


图5

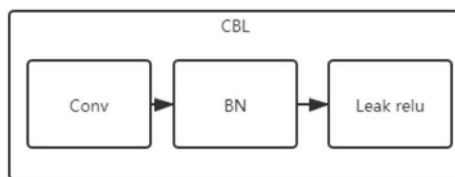


图6

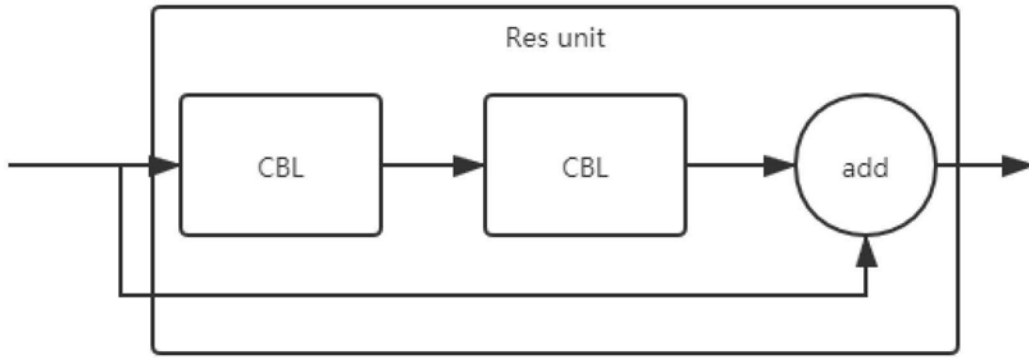


图7

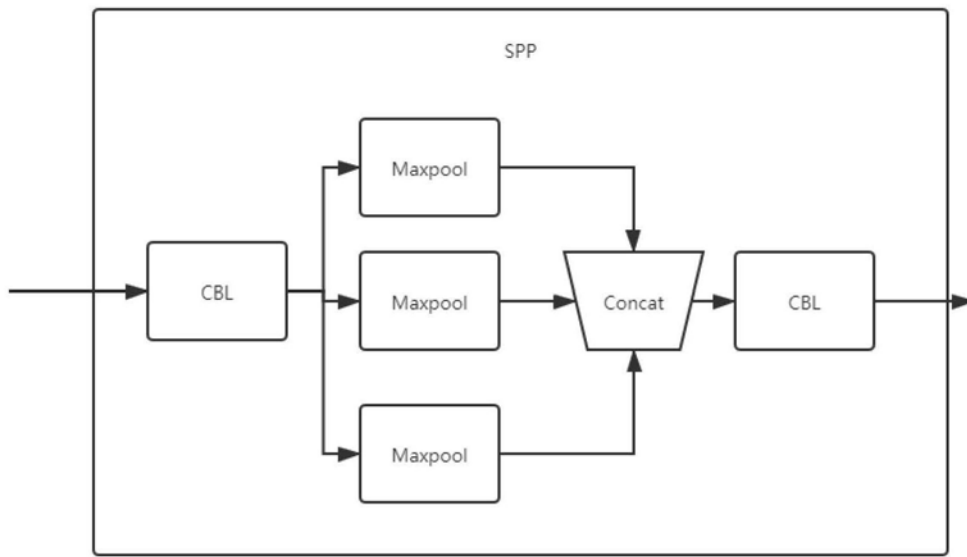


图8



图9

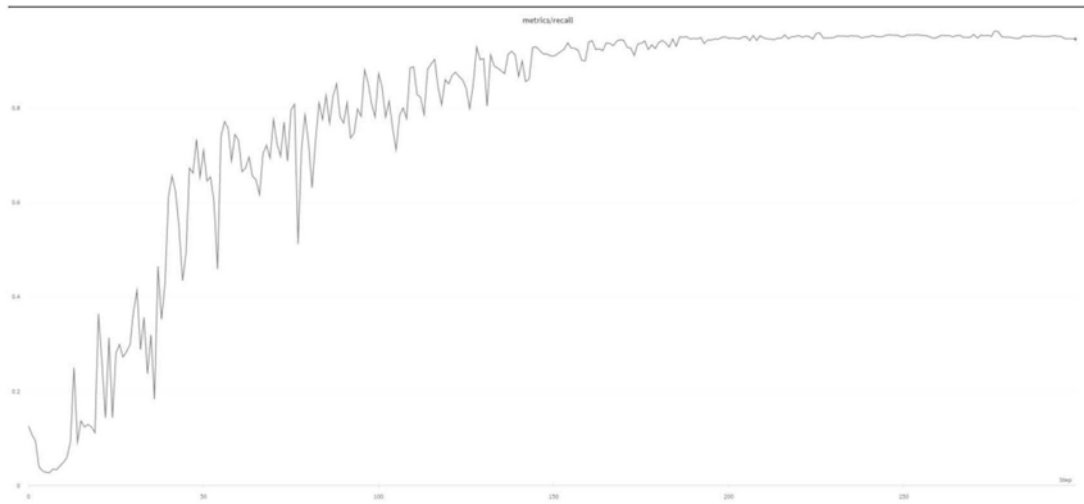


图10

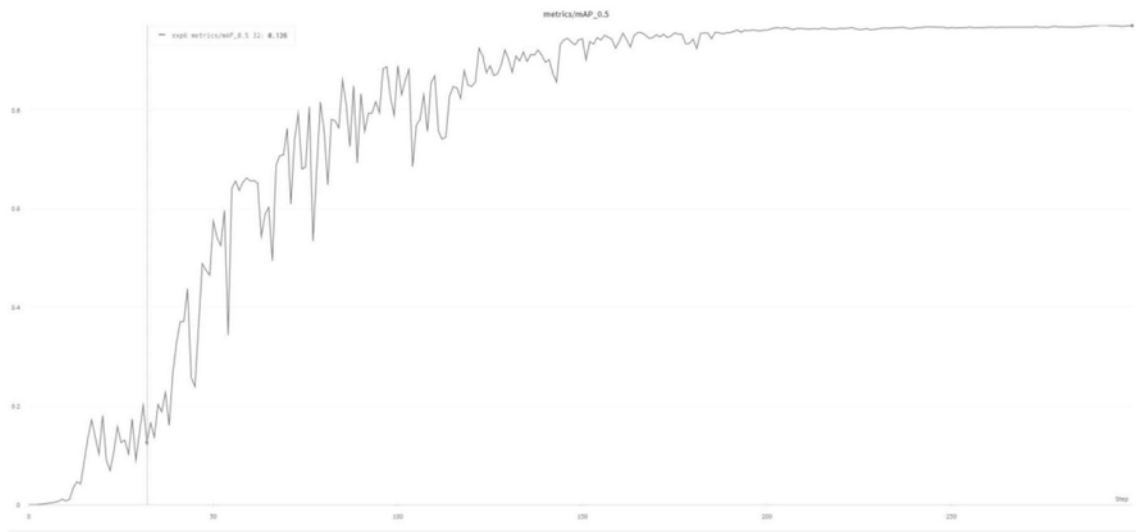


图11

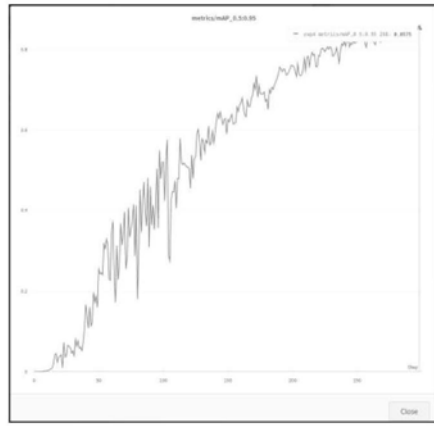


图12

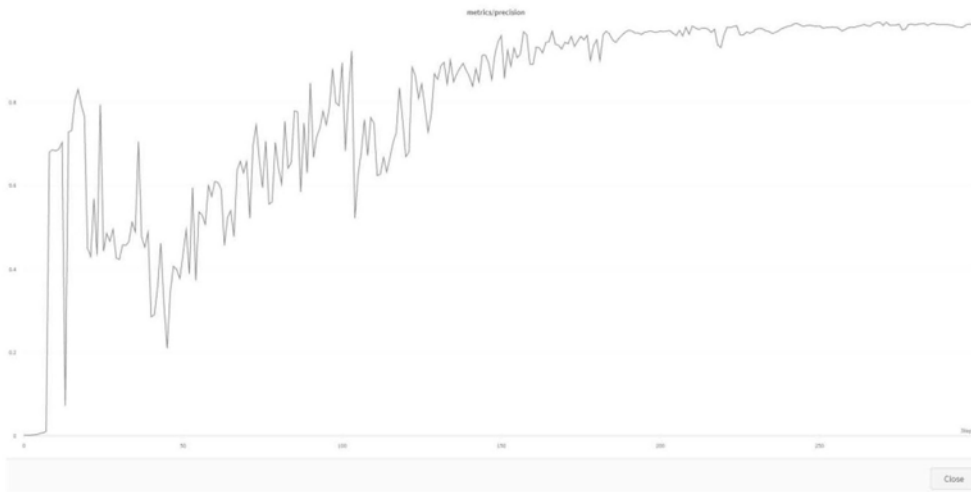


图13

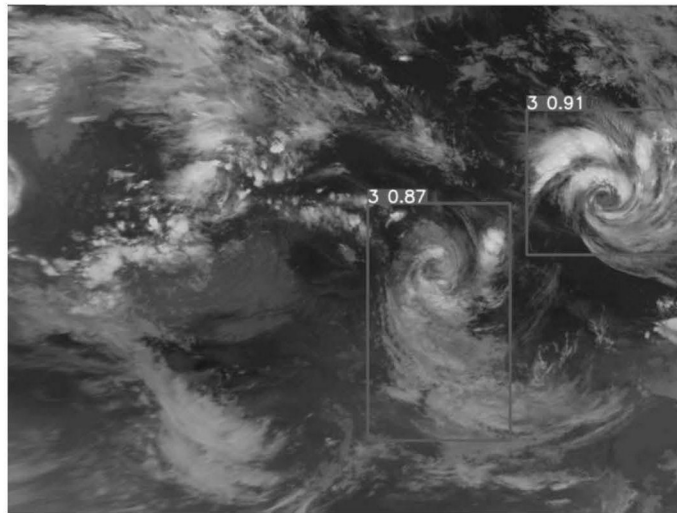


图14

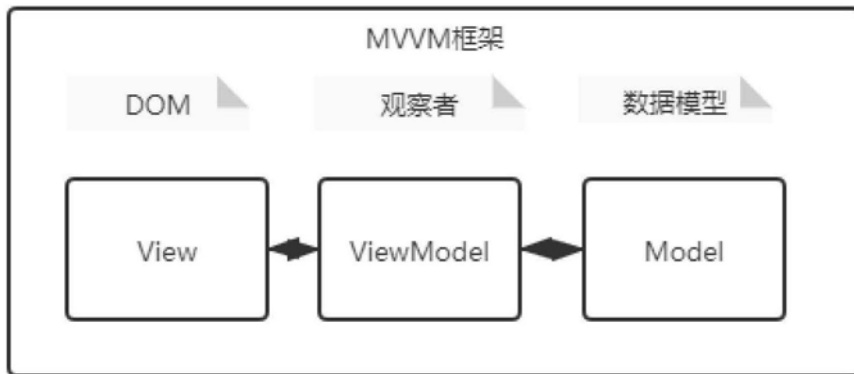


图15