



(12) 发明专利

(10) 授权公告号 CN 111310773 B

(45) 授权公告日 2023.03.24

(21) 申请号 202010225649.9

G08G 1/017 (2006.01)

(22) 申请日 2020.03.27

G06V 10/762 (2022.01)

(65) 同一申请的已公布的文献号

G06T 7/73 (2017.01)

申请公布号 CN 111310773 A

G06V 10/80 (2022.01)

G06N 3/0464 (2023.01)

(43) 申请公布日 2020.06.19

(73) 专利权人 西安电子科技大学

(56) 对比文件

CN 110796168 A, 2020.02.14

地址 710071 陕西省西安市雁塔区太白南

WO 2019144575 A1, 2019.08.01

路2号

张文超等. 多级细粒度特征融合的端到端车牌识别研究.《沈阳理工大学学报》.2018, (第05期),

专利权人 陕西理工大学

胡从坤等. 使用多任务级联卷积神经网络进行车牌照识别.《企业技术开发》.2019, (第02期),

(72) 发明人 王兰美 朱衍波 梁涛 王桂宝

廖桂生 陈正涛

审查员 张玲

(51) Int. Cl.

G06V 10/40 (2022.01)

G06V 10/82 (2022.01)

G06V 20/62 (2022.01)

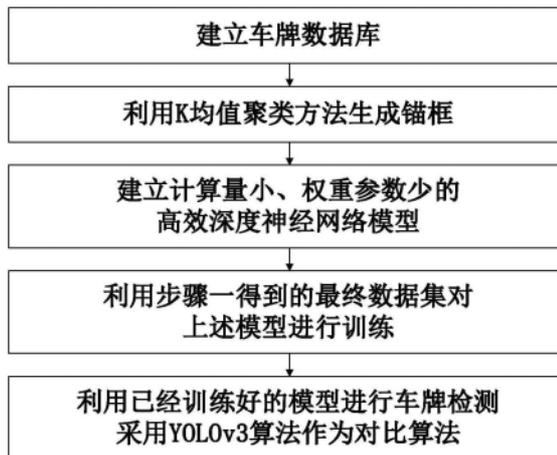
权利要求书6页 说明书13页 附图6页

(54) 发明名称

一种高效的卷积神经网络的车牌定位方法

(57) 摘要

本发明提出了一种高效的卷积神经网络车牌定位方法。主要是针对车牌识别此类特殊问题,优化了网络结构,提出了一种高效的卷积神经网络结构,此网络结构的优点是能够在不损失检测准确度的条件下减小模型的权重文件大小以及降低检测时间。实施方式如下:首先建立车牌数据库;利用K均值聚类方法生成锚框;设计一种计算量小、权重参数数量少的高效深度神经网络结构;用Adam优化算法在最终数据集上训练神经网络模型;并采用YOLOv3作为对比算法对本模型进行评估。本发明提出了一种针对于车牌检测的卷积神经网络结构,能够在模型检测准确度基本不变的条件下,减小了模型的权重文件大小,降低了车牌检测时间。



1. 一种高效的卷积神经网络车牌定位方法,包含以下步骤:

步骤一、建立车牌数据库;数据主要从以下途径收集:实地拍摄;从网页爬取;整合一些从互联网搜索到的小规模车牌数据集;然后对得到的图片进行清洗、标注、增强,得到最终训练用的数据集;

步骤二、以步骤一中建立的最终数据集为基础,采用K均值聚类方法生成锚框;

步骤三、建立计算量小、权重参数数量少的高效深度神经网络结构模型;

(3a) 引入新的卷积方法,即分离卷积;

(3b) 根据分离卷积方法,构建了一个新的特征提取网络;

整个网络结构如下:

该网络模型包含27个卷积层,其中包含14个标准卷积层和13个分离卷积层;每个卷积层包含3个操作:

第一步:进行卷积运算,利用标准卷积方法或者分离卷积方法;

第二步:将上一步得到的卷积结果进行批量归一化处理,将数据全部归一化为[0,1]之间,有利于加快训练速度;

第三步:经过激活函数Relu,将非线性特性引入到网络中,这样会保证输入输出之间是非线性映射关系,而不是简单的线性组合的关系,从而能够保证网络的学习能力;

特征提取模块的输入是一张RGB图片,这张图片可以表示为 $a \times a \times 3$ 的矩阵形式,其中 a 为图片的宽度和高度,输出为三个特征矩阵,这三个特征矩阵的维度分别为 10×10 、 20×20 和 40×40 ,其中 40×40 的特征矩阵中每一个神经元的感受野是最小的,可以负责检测原始输入图像中的小车牌目标,同理, 10×10 的特征矩阵中每一个神经元的感受野是最大的,可以负责检测原始输入图像中的大车牌目标;这样进行多尺度预测,可以避免小目标车牌漏检的情况;

特征提取模块的输入为大小为 $320 \times 320 \times 3$ 的图片,通过第一层标准卷积层C1,卷积核的大小为 3×3 ,个数为32,步长为2,第一卷积层C1输出的特征图为 $160 \times 160 \times 32$;

然后进入第二层分离卷积层C2,卷积核的大小为 3×3 ,个数为32,步长为1,第二卷积层C2输出的特征图为 $160 \times 160 \times 32$;

然后进入第三层标准卷积层C3,卷积核的大小为 1×1 ,个数为64,步长为1,第三卷积层C3输出的特征图为 $160 \times 160 \times 64$;

然后进入第四层分离卷积层C4,卷积核的大小为 3×3 ,个数为64,步长为2,第四卷积层C4输出的特征图为 $80 \times 80 \times 64$;

然后相继进入第五层标准卷积层C5、第六层分离卷积层C6、第七层标准卷积层C7,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为128,步长为1,最终C7输出的特征图为 $80 \times 80 \times 128$;

然后进入第八层分离卷积层C8,卷积核的大小为 3×3 ,个数为128,步长为2,第八卷积层C8输出的特征图为 $40 \times 40 \times 128$;

然后相继进入第九层标准卷积层C9、第十层分离卷积层C10、第十一层标准卷积层C11,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为256,步长为1,最终C11输出的特征图为 $40 \times 40 \times 256$;C11层的输出特征图会输入到后续的网络结构中继续处理;

然后进入第十二层分离卷积层C12,卷积核的大小为 3×3 ,个数为256,步长为2,第十二

卷积层C12输出的特征图为 $20 \times 20 \times 256$;

然后进入第十三层标准卷积层C13,卷积核的大小为 1×1 ,个数为512,步长为1,第十三卷积层C13输出的特征图为 $20 \times 20 \times 512$;

然后进入C14-C23层,C14为分离卷积层C15为标准卷积层,卷积核的大小依次为 3×3 、 1×1 ,个数为512,步长为1,循环5次,共十层,最终C23层的输出的特征图为 $20 \times 20 \times 512$;C23层的输出特征图会输入到后续的网络结构中继续处理;

然后进入第二十四层分离卷积层C24,卷积核的大小为 3×3 ,个数为512,步长为2,第二十四卷积层C24输出的特征图为 $10 \times 10 \times 512$;

然后相继进入第二十五层标准卷积层C25、第二十六层分离卷积层C26、第二十七层标准卷积层C27,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为1024,步长为1,最终C27输出的特征图为 $10 \times 10 \times 1024$;C27层的输出特征图会输入到后续的网络结构中继续处理;

C27层输出的特征图会在后续网络中输入到检测模块1中处理;进入检测模块1后,会将C27层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_1 1、 LC_1 2、 LC_1 3,中路也经过三层卷积层 MC_1 1、 MC_1 2、 MC_1 3;其中 LC_1 1、 LC_1 2、 LC_1 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_1 3层的输出为 $10 \times 10 \times 512$; MC_1 1、 MC_1 2、 MC_1 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_1 3层的输出为 $10 \times 10 \times 512$;然后将 LC_1 3与 MC_1 3层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_1 4,卷积核的大小为 1×1 ,个数为256,步长为1, C_1 4层的输出为 $10 \times 10 \times 256$;右路经过一层卷积层 RC_1 1,卷积核的大小为 1×1 ,个数为256,步长为1, RC_1 1的输出为 $10 \times 10 \times 256$,然后经过“拼接层” EC_1 层,将 RC_1 1与 C_1 4两层的输出合并为一个特征图, EC_1 层的输出为 $10 \times 10 \times 512$;

经过检测模块1后, EC_1 层的输出会分为两路处理,其中一路经过两层标准卷积层 C_1 5、 C_1 6,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_1 6层的输出为 $10 \times 10 \times 18$; C_1 6层的输出即为网络输出特征图1;另外一路会经过标准卷积层 C_1 7,卷积核的大小为 1×1 ,个数分别为256,步长均为1, C_1 7层的输出为 $10 \times 10 \times 256$;后经过“上采样”层 Up_1 ,放大尺寸两倍, Up_1 的输出为 $20 \times 20 \times 256$,后经过“拼接层” EC_1 层,将 Up_1 层与 C_1 7层的输出合并, EC_1 层的输出为 $20 \times 20 \times 768$;

EC_1 层输出的特征图输入到检测模块2中处理;进入检测模块2后,会将 EC_1 层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_2 1、 LC_2 2、 LC_2 3,中路也经过三层卷积层 MC_2 1、 MC_2 2、 MC_2 3;其中 LC_2 1、 LC_2 2、 LC_2 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_2 3层的输出为 $20 \times 20 \times 512$; MC_2 1、 MC_2 2、 MC_2 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_2 3层的输出为 $20 \times 20 \times 512$;然后将 LC_2 3与 MC_2 3层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_2 4,卷积核的大小为 1×1 ,个数为256,步长为1, C_2 4层的输出为 $20 \times 20 \times 256$;右路经过一层卷积层 RC_2 1,卷积核的大小为 1×1 ,个数为256,步长为1, RC_2 1的输出为 $20 \times 20 \times 256$,然后经过“拼接层” EC_2 层,将 RC_2 1与 C_2 4两层的输出合并为一个特征图, EC_2 层的输出为 $20 \times 20 \times 512$;

EC_2 层的输出也会分为两路处理,其中一路经过两层标准卷积层 C_2 5、 C_2 6,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_2 6层的输出为 $20 \times 20 \times 18$; C_2 6层的输出即为网

络输出特征图2;另外一路会经过标准卷积层 C_27 ,卷积核的大小为 1×1 ,个数分别为128,步长均为1, C_27 层的输出为 $20 \times 20 \times 128$;后经过“上采样”层 $Up2$,放大尺寸两倍, $Up2$ 的输出为 $40 \times 40 \times 128$,后经过“拼接层” $EC2$ 层,将 $Up2$ 层与 $C11$ 层的输出合并, $EC2$ 层的输出为 $40 \times 40 \times 384$;

$EC2$ 层输出的特征图输入到检测模块3中处理;进入检测模块3后,将 $EC2$ 层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_31 、 LC_32 、 LC_33 ,中路也经过三层卷积层 MC_31 、 MC_32 、 MC_33 ;其中 LC_31 、 LC_32 、 LC_33 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_33 层的输出为 $40 \times 40 \times 512$; MC_31 、 MC_32 、 MC_33 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_33 层的输出为 $40 \times 40 \times 512$;然后将 LC_33 与 MC_33 层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_34 ,卷积核的大小为 1×1 ,个数为256,步长为1, C_34 层的输出为 $40 \times 40 \times 256$;右路经过一层卷积层 RC_31 ,卷积核的大小为 1×1 ,个数为256,步长为1, RC_31 的输出为 $40 \times 40 \times 256$,然后经过“拼接层” EC_3 层,将 RC_31 与 C_34 两层的输出合并为一个特征图, EC_3 层的输出为 $40 \times 40 \times 512$;

检测模块3的输出会经过两层标准卷积层 C_35 、 C_36 ,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_36 层的输出为 $40 \times 40 \times 18$; C_36 层的输出即为网络输出特征图3;

网络的输出为3张特征图,维度分别是 $10 \times 10 \times 18$ 、 $20 \times 20 \times 18$ 、 $40 \times 40 \times 18$,其中 $10 \times 10 \times 18$ 特征图共包含1800个参数,代表了网络将原输入图片分割成了100个网格,每个网格会预测三个预测框,每一个预测框对应了6个参数;这六个参数分别是预测框的4个坐标信息 t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 、预测置信度 \hat{C}_i 和类别概率 \hat{P}_i ;其中 $(t_{xi}$ 、 $t_{yi})$ 表示第 i 个预测框中心点的坐标参数值, $(t_{wi}$ 、 $t_{hi})$ 表示第 i 个预测框宽和高的参数值,预测置信度 \hat{C}_i 表示第 i 个预测框包含目标的概率,类别概率 \hat{P}_i 为多维向量,表示了第 i 个预测框的目标是某一类别的概率;需要注意的一点是, t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 这四个参数是相对位置坐标,需要转化为最终在原始图片中的实际坐标;转换的公式如下:

$$\hat{x}_i = \tanh(t_{xi}) + c_x$$

$$\hat{y}_i = \tanh(t_{yi}) + c_y$$

$$\hat{w}_i = p_w e^{t_{wi}}$$

$$\hat{h}_i = p_h e^{t_{hi}}$$

其中, t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 是预测的相对坐标值, p_w 、 p_h 表示预测框对应锚框的宽度以及高度, c_x 、 c_y 表示预测框相对图片左上角位置坐标的偏移量, \hat{x}_i 、 \hat{y}_i 表示预测框中心点实际坐标, \hat{w}_i 、 \hat{h}_i 表示预测框的实际宽度以及实际高度;

步骤四、利用步骤一得到的最终数据集对上述网络模型进行训练;

- (1) 网络随机初始化权值,使初始化的值服从高斯正态分布;
- (2) 输入数据经过步骤三中的网络结构向前传播得到输出值为特征图1、特征图2、特征

图3,并利用特征图信息得到预测边框的信息 $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{C}_i, \hat{P}_i)$;

(3) 将数据集中标注出的真实框与聚类得到的锚框进行匹配:计算出真实框所在中心点,筛选出此中心点对应的锚框,选取与真实框有最大IOU值的锚框作为目标框,并将该真实框的坐标值信息赋给目标框,即得到目标框的坐标值 (x_i, y_i, w_i, h_i) ,并将目标框的类别值 P_i 设置为1,置信度值 C_i 设置为1,其余未标记的锚框的参数值均设置为0;

(4) 利用提出的损失函数求网络预测边框的输出值与真实边界框的目标值之间的误差损失,该损失函数包含位置损失、置信度损失和类别损失;

(5) 利用Adam优化算法进行权值更新,直到迭代次数>epoch时,结束训练;其中epoch根据精度要求设定;

步骤五、利用已训练好的模型进行车牌检测,并采用YOLOv3作为对比算法对本模型进行评估。

2. 根据权利要求1所述的一种高效的卷积神经网络车牌定位方法,步骤二中利用K均值聚类算法对车牌数据集中所有目标框进行聚类,进而发现数据集中目标框的统计规律,最后根据这些统计规律去生成锚框;

在这种方法中,聚类中心个数k等于数据集对应的锚框数;也就是说如果我们要为车牌数据集设定k个锚框,那么K均值聚类算法的聚类中心个数也等于k;另外在这个聚类任务中,如果使用传统的欧式距离进行聚类,会导致大尺度锚框的位置误差远大于小尺度锚框的位置误差;而我们最终希望的是通过聚类得到的锚框和车牌数据集中的目标框有尽可能大的IOU值,IOU为交并比,故采用以下的距离d度量:

$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$$

其中,box表示数据集中的目标框,centroid表示聚类中心对应的边界框,IOU(box, centroid)表示聚类中心对应的边界框和数据集中的目标框的交并比;

取k=9对车牌数据集进行聚类,最终得到9个新的更符合数据集的锚框,即给出锚框的位置信息,即宽度 p_w 、高度 p_h ,并使用这些锚框去训练模型。

3. 根据权利要求1所述的一种高效的卷积神经网络车牌定位方法,步骤(3a)中引入的分离卷积,具体过程如下:假设该卷积层的输入为 $D_f \times D_f \times M$ 维度的矩阵 $X = (X_1, X_2 \dots X_M)$;对该矩阵X进行分离卷积运算相当于分为2个步骤执行;

第一步,将输入矩阵 $X = (X_1, X_2 \dots X_M)$ 与卷积核 $Y = (Y_1, Y_2 \dots Y_M)$ 对应做卷积运算,得到输出特征矩阵 $G = (G_1, G_2 \dots G_M)$;

$$G_j = X_j * Y_j$$

其中 X_j 是尺寸为 $D_f \times D_f$ 二维矩阵, Y_j 是尺寸为 $D_k \times D_k$ 的二维卷积核, G_j 是输出的平面二维特征矩阵,尺寸仍为 $D_f \times D_f$;

也就是说,一个卷积核 Y_j 只负责对输入的一个通道 X_j 进行卷积,得到M个通道特征图 G_j ;

第二步,将上一步得到的输出特征矩阵 $G = (G_1, G_2 \dots G_M)$ 与C个卷积核 $Z^d = (Z_1^d, Z_2^d \dots Z_M^d)$ 做卷积运算,得到最终输出特征图 $P = (P^1, P^2, \dots, P^C)$;

$$P^d = \sum_j^M (G_j * Z_j^d)$$

其中 G_j 是输出的平面二维特征矩阵,尺寸仍为 $D_f \times D_f$, Z_j^d 是尺寸为 1×1 的二维卷积核, P^d 是最终输出特征图,尺寸为 $D_f \times D_f$, $*$ 表示卷积运算。

4. 根据权利要求1所述的一种高效的卷积神经网络车牌定位方法,步骤四中损失函数的确定,具体如下:

算法损失函数由位置损失、置信度损失和类别损失三部分组成;

位置损失分别为中心损失和尺度损失;中心损失用于评估网络预测框中心和实际物体的标注框的中心间的误差;尺度损失则用于评估网络预测框高度与宽度和实际物体标注框高度与宽度间的误差;中心损失和使用的函数为二值交叉熵函数,尺度损失使用的函数为平方函数,具体如下:

$$x_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) [x_i \ln \hat{x}_i + (1 - x_i) \ln(1 - \hat{x}_i)]$$

$$y_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)]$$

$$center_loss = x_loss + y_loss$$

$$size_loss = -\alpha \frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) \left[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right]$$

其中, N 表示网络预测框的总数, l_i^{obj} 表示第 i 个预测框中是否存在目标,若存在, $l_i^{obj} = 1$,否则为0; (x_i, y_i) 表示目标所在第 i 个标注框真实中心位置, (\hat{x}_i, \hat{y}_i) 表示第 i 个预测框的中心位置, (w_i, h_i) 目标所在第 i 个标注框真实宽度和高度, (\hat{w}_i, \hat{h}_i) 表示第 i 个预测框的宽度和高度, α 用于调整尺度损失在所有损失中所占据的比例;

置信度损失用于评估预测框对应的置信度和实际物体标注框对应的置信度之间的误差;置信度损失使用的函数为二值交叉熵函数,具体如下:

$$obj_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} [C_i \ln \hat{C}_i + (1 - C_i) \ln(1 - \hat{C}_i)]$$

$$noobj_loss = -\frac{1}{N} \sum_{i=1}^N (1 - l_i^{obj}) [C_i \ln \hat{C}_i + (1 - C_i) \ln(1 - \hat{C}_i)]$$

$$confidence_loss = obj_loss + noobj_loss$$

其中, C_i 表示目标所在第 i 个标注框真实置信度, \hat{C}_i 表示第 i 个预测框的置信度;

类别损失评估网络预测框对应的类别和实际物体对应的类别之间的误差,具体如下所示:

$$cls_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} [P_i \ln \hat{P}_i + (1 - P_i) \ln(1 - \hat{P}_i)]$$

其中, p_i 表示目标所在第 i 个标注框中物体的类别的概率, \hat{p}_i 表示第 i 个预测框物体的类别概率;

总损失函数为:

$loss = center_loss + size_loss + confidence_loss + cls_loss$ 。

5. 根据权利要求1所述的一种高效的卷积神经网络车牌定位方法, 步骤五中检测过程采用 $IOU=0.8$ 时的检测准确率、权重文件大小和检测时间作为算法性能的度量指标, 如果算法对某张图片的预测矩形框和该图片的真实矩形框间的交并比大于0.8, 那么就认为算法对该图片检测成功, 其次, 在检测准确率不大幅降低的情况下, 权重文件越小, 检测时间越少, 则认为算法性能越强;

前述步骤中, N 表示网络预测框的总数, 其中 $N=6300$, $i=1, 2, \dots, N$ 表示预测框的标号; M 表示分离卷积中假设的输入向量的通道个数, $j=1, 2, \dots, M$; C 表示分离卷积第二步中卷积核的个数, $d=1, 2, \dots, C$ 。

一种高效的卷积神经网络的车牌定位方法

技术领域

[0001] 本发明属于图像识别领域,涉及一种高效的基于深度卷积神经网络的车牌定位方法。

背景技术

[0002] 随着社会经济的发展,汽车已成为人们日常出行的重要方式,汽车的种类和数量迅速增加,对交通管制提出了更高的要求。近年来,智能化处理技术在很多领域发挥着重要作用,智能交通系统也随之兴起,大大提高了管理效率,节省了大量人力。车牌是车辆的重要标识,每辆汽车都有唯一的“身份证件”,这为车辆的统一管理提供了强有力的保障。在车辆管理高效性的要求下,自动收集并识别车牌成为整个检测过程极其重要的环节。

[0003] 车牌检测技术在日常生活中发挥着重要的应用作用,特别是随着当今互联网技术的飞速发展,许多问题都可以通过获取车牌信息得到高效的解决,但是这对车牌识别的速度与精度提出了更高的要求。车牌识别技术的任务是从包含各种复杂背景的车辆图片中自动检测车牌区域并对其进行处理和识别,最后输出识别结果。传统的车牌识别算法或多或少都会受到环境或人为因素的制约,例如光照条件,大气能见度,部分遮挡,车牌悬挂区域等。这些因素使得一些检测算法达不到人为预期的效果,导致应用过程中受阻。

[0004] 如果开发一种车牌检测算法,使其经过微调之后能够应用在复杂环境条件下,并取得速度与精度的双重提升,那将会有很大的实际应用价值。而近年来,随着计算机性能的大幅度提升,深度学习方法受到很多研究者的青睐,并且在众多领域取得了不错的效果,这为车牌检测算法的开发提供了一个很好的思路。而且,当外界因素改变的情况下,可以通过对模型微调来提高适应性,对车牌检测来说无疑是一大福音。

[0005] 卷积神经网络是一种特殊的神经网络模型,它的神经元是非全连接的,而且同层神经元共享权值,基于此网络模型提出的YOLO系列算法在图像检测任务中表现最为突出,尤其是近年来提出的YOLOv3算法如文献1 (Redmon J, Farhadi A.YOLOv3:an incremental improvement.In Proceedings of the Conference on Computer Vision and Pattern Recognition.Salt Lake City,USA:2018.1-4),表现结果甚至突破了人眼的极限,无疑成为图像检测领域的佼佼者。

[0006] 利用YOLOv3网络模型检测车牌,经实验证明拥有良好的性能。然而该网络模型中的权重参数过多,这导致训练结束后得到的模型权重文件过大,达到了235M,这会导致算法模型的检测速度降低,导致对检测硬件设备的要求过高,过高的硬件需求会导致模型在部署时受到过多限制,进而导致其适用场景有限。针对以上问题,本发明主要研究在精度几乎不降低的情况下,如何减小网络模型的规模。为达到上述目的,针对于车牌这单一的类别,本发明基于深度卷积神经网络,从特征提取网络与检测网络两方面进行改进,提出了一种全新的卷积方法,减轻了原本网络参数的量级,构建了一种高效轻量网络结构。

发明内容

[0007] 本发明的目的是提供一种高效的卷积神经网络的车牌定位方法。

[0008] 一种高效的卷积神经网络的车牌定位方法步骤如下：

[0009] 步骤一、建立车牌数据库。数据主要从以下途径收集：第一，实地拍摄。利用照相机等设备拍摄实际车牌，这部分车牌图像包括了白天、黑夜、阴天、雨天、倾斜等多种场景。第二，从网页爬取。从浏览器网页上爬取车牌图片。第三，整合一些从互联网搜索到的小规模车牌数据集。然后利用LabelImg软件对收集到的图片进行标注，即标出目标所在的中心位置坐标 (x, y) 及目标的宽度 (w, h) ，得到原始数据集。标注完成后，为扩大原始数据集中的数据量，运用数据增强技术在原始数据集上扩充数据集，得到最终训练用的数据集。

[0010] 步骤二、以步骤一中建立的最终数据集为基础，采用K均值聚类方法生成锚框，利用K均值聚类算法对车牌数据集中所有目标框进行聚类，进而发现数据集中目标框的统计规律，最后根据这些统计规律去生成锚框。

[0011] 在这种方法中，聚类中心个数 k 等于数据集对应的锚框数。也就是说如果我们要为车牌数据集设定 k 个锚框，那么K均值聚类算法的聚类中心个数也等于 k 。另外在这个聚类任务中，如果使用传统的欧式距离进行聚类，会导致大尺度锚框的位置误差远大于小尺度锚框的位置误差。而我们最终希望的是通过聚类得到的锚框和车牌数据集中的目标框有尽可能大的IOU值，故而本发明采用以下的距离 d 度量：

[0012] $d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$

[0013] 其中， box 表示数据集中的目标框， centroid 表示聚类中心对应的边界框，IOU 表示交并比， $\text{IOU}(\text{box}, \text{centroid})$ 表示聚类中心对应的边界框和数据集中的目标框的交并比。

[0014] 取 $k=9$ 对车牌数据集进行聚类，最终得到9个新的更符合本发明数据集的锚框，即给出锚框的位置信息，即宽度 p_w 、高度 p_h ，并使用这些锚框去训练模型。

[0015] 步骤三、建立计算量小、权重参数数量少的高效深度神经网络结构；

[0016] (3a) 引入一种新的卷积方法，即分离卷积

[0017] 分离卷积算法过程如下：假设该卷积层的输入为 $D_f \times D_f \times M$ 维度的矩阵 $X = (X_1, X_2 \dots X_M)$ 。对该矩阵 X 进行分离卷积运算相当于分为2个步骤执行。

[0018] 第一步，将输入矩阵 $X = (X_1, X_2 \dots X_M)$ 与卷积核 $Y = (Y_1, Y_2 \dots Y_M)$ 对应做卷积运算，得到输出特征矩阵 $G = (G_1, G_2 \dots G_M)$ 。

[0019] $G_j = X_j * Y_j$

[0020] 其中 X_j 是尺寸为 $D_f \times D_f$ 二维矩阵， Y_j 是尺寸为 $D_k \times D_k$ 的二维卷积核， G_j 是输出的平面二维特征矩阵，尺寸仍为 $D_f \times D_f$ ，* 表示卷积运算。

[0021] 也就是说，一个卷积核 Y_j 只负责对输入的一个通道 X_j 进行卷积，得到 M 个通道特征图 G_j 。

[0022] 第二步，将上一步得到的输出特征矩阵 $G = (G_1, G_2 \dots G_M)$ 与 C 个卷积核 $Z^d = (Z_1^d, Z_2^d \dots Z_M^d)$ 做卷积运算，得到最终输出特征图 $P = (P^1, P^2, \dots, P^C)$ 。

[0023]
$$P^d = \sum_j^M (G_j * Z_j^d)$$

[0024] 其中 G_j 是输出的平面二维特征矩阵，尺寸仍为 $D_f \times D_f$ ， Z_j^d 是尺寸为 1×1 的二维卷

积核, P^d 是最终输出特征图, 尺寸为 $D_f \times D_f$ 。

[0025] (3b) 根据分离卷积方法, 本发明构建了一个新的特征提取网络, 针对车牌检测这一特定任务。该网络模型包含27个卷积层, 其中包含14个标准卷积层和13个分离卷积层。

[0026] 每个卷积层包含3个操作:

[0027] 第一步: 进行卷积运算, 利用标准卷积方法或者分离卷积方法;

[0028] 第二步: 将上一步得到的卷积结果进行批量归一化处理, 将数据全部归一化为 $[0, 1]$ 之间, 有利于加快训练速度;

[0029] 第三步: 经过激活函数 ReLU, 将非线性特性引入到本发明的网络中, 这样会保证输入输出之间是非线性映射关系, 而不是简单的线性组合的关系, 从而能够保证网络的学习能力。

[0030] 特征提取模块的输入是一张RGB图片, 这张图片可以表示为 $a \times a \times 3$ 的矩阵形式, 其中 a 为图片的宽度和高度, 输出为三个特征矩阵, 这三个特征矩阵的维度分别为 10×10 、 20×20 和 40×40 , 其中 40×40 的特征矩阵中每一个神经元的感受野是最小的, 可以负责检测原始输入图像中的小车牌目标, 同理, 10×10 的特征矩阵中每一个神经元的感受野是最大的, 可以负责检测原始输入图像中的大车牌目标。这样进行多尺度预测, 可以避免小目标车牌漏检的情况。

[0031] 特征提取模块的输入为大小为 $320 \times 320 \times 3$ 的图片, 通过第一层标准卷积层 C1, 卷积核的大小为 3×3 , 个数为32, 步长为2, 第一卷积层C1输出的特征图为 $160 \times 160 \times 32$ 。

[0032] 然后进入第二层分离卷积层C2, 卷积核的大小为 3×3 , 个数为32, 步长为1, 第二卷积层C2输出的特征图为 $160 \times 160 \times 32$ 。

[0033] 然后进入第三层标准卷积层C3, 卷积核的大小为 1×1 , 个数为64, 步长为1, 第三卷积层C3输出的特征图为 $160 \times 160 \times 64$ 。

[0034] 然后进入第四层分离卷积层C4, 卷积核的大小为 3×3 , 个数为64, 步长为2, 第四卷积层C4输出的特征图为 $80 \times 80 \times 64$ 。

[0035] 然后相继进入第五层标准卷积层C5、第六层分离卷积层C6、第七层标准卷积层C7, 卷积核的大小依次为 1×1 、 3×3 、 1×1 , 个数为128, 步长为1, 最终C7输出的特征图为 $80 \times 80 \times 128$ 。

[0036] 然后进入第八层分离卷积层C8, 卷积核的大小为 3×3 , 个数为128, 步长为2, 第八卷积层C8输出的特征图为 $40 \times 40 \times 128$ 。

[0037] 然后相继进入第九层标准卷积层C9、第十层分离卷积层C10、第十一层标准卷积层C11, 卷积核的大小依次为 1×1 、 3×3 、 1×1 , 个数为256, 步长为1, 最终C11输出的特征图为 $40 \times 40 \times 256$ 。C11层的输出特征图会输入到后续的网络结构中继续处理。

[0038] 然后进入第十二层分离卷积层C12, 卷积核的大小为 3×3 , 个数为256, 步长为2, 第十二卷积层C12输出的特征图为 $20 \times 20 \times 256$ 。

[0039] 然后进入第十三层标准卷积层C13, 卷积核的大小为 1×1 , 个数为512, 步长为1, 第十三卷积层C13输出的特征图为 $20 \times 20 \times 512$ 。

[0040] 然后进入C14-C23层, C14为分离卷积层C15为标准卷积层, 卷积核的大小依次为 3×3 、 1×1 , 个数为512, 步长为1, 循环5次, 共十层, 最终C23层的输出的特征图为 $20 \times 20 \times 512$ 。C23层的输出特征图会输入到后续的网络结构中继续处理。

[0041] 然后进入第二十四层分离卷积层C24,卷积核的大小为 3×3 ,个数为512,步长为2,第二十四卷积层C24输出的特征图为 $10 \times 10 \times 512$ 。

[0042] 然后相继进入第二十五层标准卷积层C25、第二十六层分离卷积层C26、第二十七层标准卷积层C27,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为1024,步长为1,最终C27输出的特征图为 $10 \times 10 \times 1024$ 。C27层的输出特征图会输入到后续的网络结构中继续处理。

[0043] C27层输出的特征图会在后续网络中输入到检测模块1中处理。进入检测模块1后,会将C27层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_1 1、 LC_1 2、 LC_1 3,中路也经过三层卷积层 MC_1 1、 MC_1 2、 MC_1 3。其中 LC_1 1、 LC_1 2、 LC_1 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_1 3层的输出为 $10 \times 10 \times 512$; MC_1 1、 MC_1 2、 MC_1 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_1 3层的输出为 $10 \times 10 \times 512$ 。然后将 LC_1 3与 MC_1 3层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_1 4,卷积核的大小为 1×1 ,个数为256,步长为1, C_1 4层的输出为 $10 \times 10 \times 256$ 。右路经过一层卷积层 RC_1 1,卷积核的大小为 1×1 ,个数为256,步长为1, RC_1 1的输出为 $10 \times 10 \times 256$,然后经过“拼接层” EC_1 层,将 RC_1 1与 C_1 4两层的输出合并为一个特征图, EC_1 层的输出为 $10 \times 10 \times 512$ 。

[0044] 经过检测模块1后, EC_1 层的输出会分为两路处理,其中一路经过两层标准卷积层 C_1 5、 C_1 6,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_1 6层的输出为 $10 \times 10 \times 18$ 。 C_1 6层的输出即为网络输出特征图1。另外一路会经过标准卷积层 C_1 7,卷积核的大小为 1×1 ,个数分别为256,步长均为1, C_1 7层的输出为 $10 \times 10 \times 256$ 。后经过“上采样”层 Up_1 ,放大尺寸两倍, Up_1 的输出为 $20 \times 20 \times 256$,后经过“拼接层” EC_1 层,将 Up_1 层与 C_1 7层的输出合并, EC_1 层的输出为 $20 \times 20 \times 768$ 。

[0045] EC_1 层输出的特征图再输入到检测模块2中处理。进入检测模块2后,将 EC_1 层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_2 1、 LC_2 2、 LC_2 3,中路也经过三层卷积层 MC_2 1、 MC_2 2、 MC_2 3。其中 LC_2 1、 LC_2 2、 LC_2 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_2 3层的输出为 $20 \times 20 \times 512$; MC_2 1、 MC_2 2、 MC_2 3层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_2 3层的输出为 $20 \times 20 \times 512$ 。然后将 LC_2 3与 MC_2 3层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_2 4,卷积核的大小为 1×1 ,个数为256,步长为1, C_2 4层的输出为 $20 \times 20 \times 256$ 。右路经过一层卷积层 RC_2 1,卷积核的大小为 1×1 ,个数为256,步长为1, RC_2 1的输出为 $20 \times 20 \times 256$,然后经过“拼接层” EC_2 层,将 RC_2 1与 C_2 4两层的输出合并为一个特征图, EC_2 层的输出为 $20 \times 20 \times 512$ 。

[0046] EC_2 层的输出也会分为两路处理,其中一路经过两层标准卷积层 C_2 5、 C_2 6,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_2 6层的输出为 $20 \times 20 \times 18$ 。 C_2 6层的输出即为网络输出特征图2。另外一路经过标准卷积层 C_2 7,卷积核的大小为 1×1 ,个数分别为128,步长均为1, C_2 7层的输出为 $20 \times 20 \times 128$ 。后经过“上采样”层 Up_2 ,放大尺寸两倍, Up_2 的输出为 $40 \times 40 \times 128$,后经过“拼接层” EC_2 层,将 Up_2 层与 C_2 7层的输出合并, EC_2 层的输出为 $40 \times 40 \times 384$ 。

[0047] EC_2 层输出的特征图再输入到检测模块3中处理。进入检测模块3后,将 EC_2 层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_3 1、 LC_3 2、 LC_3 3,中路也

经过三层卷积层 MC_31 、 MC_32 、 MC_33 。其中 LC_31 、 LC_32 、 LC_33 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ，个数分别为256、512、512，步长均为1， LC_33 层的输出为 $40 \times 40 \times 512$ ； MC_31 、 MC_32 、 MC_33 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ，个数分别为256、512、512，步长均为1， MC_33 层的输出为 $40 \times 40 \times 512$ 。然后将 LC_33 与 MC_33 层的输出特征图中的元素对应相加，将相加后的结果经过一层标准卷积层 C_34 ，卷积核的大小为 1×1 ，个数为256，步长为1， C_34 层的输出为 $40 \times 40 \times 256$ 。右路经过一层卷积层 RC_31 ，卷积核的大小为 1×1 ，个数为256，步长为1， RC_31 的输出为 $40 \times 40 \times 256$ ，然后经过“拼接层” EC_3 层，将 RC_31 与 C_34 两层的输出合并为一个特征图， EC_3 层的输出为 $40 \times 40 \times 512$ 。

[0048] 检测模块3的输出会经过两层标准卷积层 C_35 、 C_36 ，卷积核的大小均为 1×1 ，个数分别为512、18，步长均为1， C_36 层的输出为 $40 \times 40 \times 18$ 。 C_36 层的输出即为网络输出特征图3。

[0049] 网络的输出为3张特征图，维度分别是 $10 \times 10 \times 18$ 、 $20 \times 20 \times 18$ 、 $40 \times 40 \times 18$ ，其中 $10 \times 10 \times 18$ 特征图共包含1800个参数，代表了网络将原输入图片分割成了100个网格，每个网格会预测三个预测框，每一个预测框对应了6个参数。这六个参数分别是预测框的4个坐标信息 t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 、预测置信度 \hat{C}_i 和类别概率 \hat{P}_i 。其中 (t_{xi}, t_{yi}) 表示第 i 个预测框中心点的坐标参数值， (t_{wi}, t_{hi}) 表示第 i 个预测框宽和高的参数值，预测置信度 \hat{C}_i 表示第 i 个预测框包含目标的概率，类别概率 \hat{P}_i 为多维向量，表示了第 i 个预测框的目标是某一类别的概率。需要注意的一点是， t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 这四个参数是相对位置坐标，需要转化为最终在原始图片中的实际坐标。转换的公式如下：

$$[0050] \quad \hat{x}_i = \tanh(t_{xi}) + c_x$$

$$[0051] \quad \hat{y}_i = \tanh(t_{yi}) + c_y$$

$$[0052] \quad \hat{w}_i = p_w e^{t_{wi}}$$

$$[0053] \quad \hat{h}_i = p_h e^{t_{hi}}$$

[0054] 其中， t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 是预测的相对坐标值， p_w 、 p_h 表示预测框对应锚框的宽度以及高度， c_x 、 c_y 表示预测框相对图片左上角位置坐标的偏移量， \hat{x}_i 、 \hat{y}_i 表示预测框中心点实际坐标， \hat{w}_i 、 \hat{h}_i 表示预测框的实际宽度以及实际高度。

[0055] 步骤四、利用步骤一得到的最终数据集对上述模型进行训练；

[0056] (1) 网络随机初始化权值，使初始化的值服从高斯正态分布。

[0057] (2) 输入数据经过本发明步骤三中的网络结构向前传播得到输出值为特征图1、特征图2、特征图3，并利用特征图信息得到预测边框的信息 $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{C}_i, \hat{P}_i)$ ；

[0058] (3) 将数据集中标注出的真实框与聚类得到的锚框进行匹配：计算出真实框所在的中心点，筛选出此中心点对应的锚框，选取与真实框有最大IOU值的锚框作为目标框，并将该真实框的坐标值信息赋给目标框，即得到目标框的坐标值 (x_i, y_i, w_i, h_i) ，并将目标框的类别值 P_i 设置为1，置信度值 C_i 设置为1，其余未标记的锚框的参数值均设置为0。

[0059] (4) 利用本发明提出的损失函数求网络预测边框的输出值与目标框的目标值之间的误差损失,该损失函数包含位置损失、置信度损失和类别损失。

[0060] 位置损失分为中心损失和尺度损失。中心损失用于评估网络预测框中心和实际物体的标注框的中心间的误差;尺度损失则用于评估网络预测框高度与宽度和实际物体标注框高度与宽度间的误差,具体如下:

$$[0061] \quad x_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) [x_i \ln \hat{x}_i + (1 - x_i) \ln(1 - \hat{x}_i)]$$

$$[0062] \quad y_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)]$$

$$[0063] \quad center_loss = x_loss + y_loss$$

$$[0064] \quad size_loss = -\alpha \frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) \left[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right]$$

[0065] 其中,N表示网络预测框的总数, l_i^{obj} 表示第i个预测框中是否存在目标,若存在, $l_i^{obj}=1$,否则为0; (x_i, y_i) 表示目标所在第i个标注框真实中心位置, (\hat{x}_i, \hat{y}_i) 表示第i个预测框的中心位置, (w_i, h_i) 目标所在第i个标注框真实宽度和高度, (\hat{w}_i, \hat{h}_i) 表示第i个预测框的宽度和高度, α 用于调整尺度损失在所有损失中所占据的比例。

[0066] 置信度损失用于评估预测框对应的置信度和实际物体标注框对应的置信度之间的误差。置信度损失使用的函数为二值交叉熵函数,具体如下:

$$[0067] \quad obj_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} [C_i \ln \hat{C}_i + (1 - C_i) \ln(1 - \hat{C}_i)]$$

$$[0068] \quad noobj_loss = -\frac{1}{N} \sum_{i=1}^N (1 - l_i^{obj}) [C_i \ln \hat{C}_i + (1 - C_i) \ln(1 - \hat{C}_i)]$$

$$[0069] \quad confidence_loss = obj_loss + noobj_loss$$

[0070] 其中, C_i 表示目标所在第i个标注框真实置信度, \hat{C}_i 表示第i个预测框的置信度。

[0071] 类别损失评估网络预测框对应的类别和实际物体对应的类别之间的误差,具体如下所示:

$$[0072] \quad cls_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} [P_i \ln \hat{P}_i + (1 - P_i) \ln(1 - \hat{P}_i)]$$

[0073] 其中, p_i 表示目标所在第i个标注框中物体的类别的概率, \hat{p}_i 表示第i个预测框物体的类别概率。

[0074] 总损失函数为:

$$[0075] \quad loss = center_loss + size_loss + confidence_loss + cls_loss$$

[0076] (5) 利用Adam优化算法进行权值更新,直到迭代次数>epoch时,结束训练。

[0077] 步骤五、利用已训练好的模型进行车牌检测,并采用YOLOv3作为对比算法对本模型进行评估。本发明中,检测过程采用交并比IOU=0.8时的检测准确率、权重文件大小和检

测时间作为算法性能的度量指标,如果算法对某张图片的预测矩形框和该图片的真实矩形框间的交并比大于0.8,那么就认为算法对该图片检测成功,其次,在检测准确率没有大幅降低的情况下,权重文件越小,检测时间越少,则认为算法性能越强。

[0078] 前述步骤中,N表示网络预测框的总数,其中 $N=6300$, $i=1,2,\dots,N$ 表示预测框的标号。 M 表示分离卷积中假设的输入向量的通道个数, $j=1,2,\dots,M$ 。 C 表示分离卷积第二步中卷积核的个数, $d=1,2,\dots,C$ 。

[0079] 本发明具有以下优点:

[0080] 1.本发明建立了一个适用于车牌检测任务的数据集,并在此数据集上训练了本发明所设计的算法模型,达到了预期的效果。

[0081] 2.本发明采用全新的卷积方式,先将输入的通道特征与区域特征分开考虑,然后又进行融合,构建了一个高效轻量的网络结构,大大缩减的模型的参数数量,减小了计算量。

[0082] 3.本发明构建的模型所生成的权重文件小,检测速度更快。在实际部署阶段对设备的要求更低,方便进行车牌检测。

[0083] 附图(表)说明

[0084] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中需要使用的附图做简单介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0085] 图1是本发明方法的流程图;

[0086] 图2是本发明特征提取网络结构示意图;

[0087] 图3是本发明检测网络结构示意图;

[0088] 图4是本发明网络结构示意图;

[0089] 图5(a)是单个网格输出示意图;

[0090] 图5(b)是单个预测框输出向量示意图;

[0091] 图6是训练过程示意图;

[0092] 图7是各不同环境下检测图对比示意图;

[0093] 表1是交并比 $IOU=0.8$ 时各算法性能对比结果;

评价指标	算法	
	YOLOv3	本发明模型
准确率	78.64%	78.06%
权重大小	235M	51M
检测时间	0.071s	0.051s

[0094]

表 1

具体实施方式

[0095] 为了让本发明的上述和其它目的、特征及优点能更明显,参照附图,对本发明的实施方案做详细说明。

[0096] 本发明的目的是提供一种高效的卷积神经网络的车牌定位方法。

[0097] 如图1所示,一种高效的卷积神经网络的车牌定位方法步骤如下:

[0098] 步骤一、建立车牌数据库。数据主要从以下途径收集:第一,实地拍摄。利用照相机等设备拍摄实际车牌,这部分车牌图像包括了白天、黑夜、阴天、雨天、倾斜等多种场景。第二,从网页爬取。从浏览器网页上爬取车牌图片。第三,整合一些从互联网搜索到的小规模车牌数据集。然后利用LabelImg软件对收集到的图片进行标注,得到原始数据集。标注完成后,为扩大原始数据集中的数据量,运用数据增强技术在原始数据集上扩充数据集,得到最终训练用的数据集。

[0099] 步骤二、以步骤一中建立的最终数据集为基础,采用K均值聚类方法生成锚框,利用K均值聚类算法对车牌数据集中所有目标框进行聚类,进而发现数据集中目标框的统计规律,最后根据这些统计规律去生成锚框。

[0100] 在这种方法中,聚类中心个数 k 等于数据集对应的锚框数。也就是说如果我们要为车牌数据集设定 k 个锚框,那么K均值聚类算法的聚类中心个数也等于 k 。另外在这个聚类任务中,如果使用传统的欧式距离进行聚类,会导致大尺度锚框的位置误差远大于小尺度锚框的位置误差。而我们最终希望的是通过聚类得到的锚框和车牌数据集中的目标框有尽可能大的IOU值,故而本发明采用以下的距离度量:

[0101] $d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$

[0102] 其中,box表示数据集中的目标框,centroid表示聚类中心对应的边界框,IOU(box,centroid)表示聚类中心对应的边界框和数据集中的目标框的交并比。

[0103] 取 $k=9$ 对车牌数据集进行聚类,最终得到9个新的更符合本发明数据集的锚框,即给出锚框的位置信息,即宽度 p_w 、高度 p_h ,并使用这些锚框去训练模型。

[0104] 步骤三、建立计算量小、权重参数数量少的高效深度神经网络结构;

[0105] (3a) 引入一种新的卷积方法,即分离卷积

[0106] 分离卷积算法过程如下:假设该卷积层的输入为 $D_f \times D_f \times M$ 维度的矩阵 $X = (X_1, X_2 \dots X_M)$ 。对该矩阵 X 进行分离卷积运算相当于分为2个步骤执行。

[0107] 第一步,将输入矩阵 $X = (X_1, X_2 \dots X_M)$ 与卷积核 $Y = (Y_1, Y_2 \dots Y_M)$ 对应做卷积运算,得到输出特征矩阵 $G = (G_1, G_2 \dots G_M)$ 。

[0108] $G_j = X_j * Y_j$

[0109] 其中 X_j 是尺寸为 $D_f \times D_f$ 二维矩阵, Y_j 是尺寸为 $D_k \times D_k$ 的二维卷积核, G_j 是输出的平面二维特征矩阵,尺寸仍为 $D_f \times D_f$ 。

[0110] 也就是说,一个卷积核 Y_j 只负责对输入的一个通道 X_j 进行卷积,得到 M 个通道特征图 G_j 。

[0111] 第二步,将上一步得到的输出特征矩阵 $G = (G_1, G_2 \dots G_M)$ 与 C 个卷积核 $Z^d = (Z_1^d, Z_2^d \dots Z_M^d)$ 做卷积运算,得到最终输出特征图 $P = (P^1, P^2, \dots, P^C)$ 。

[0112]
$$P^d = \sum_j^M (G_j * Z_j^d)$$

[0113] 其中 G_j 是输出的平面二维特征矩阵,尺寸仍为 $D_f \times D_f$, Z_j^d 是尺寸为 1×1 的二维卷积核, P^d 是最终输出特征图,尺寸为 $D_f \times D_f$, $*$ 表示卷积运算。

[0114] (3b) 根据分离卷积方法,本发明构建了一个新的特征提取网络,针对车牌检测这一特定任务。该网络模型包含27个卷积层,其中包含14个标准卷积层和13个分离卷积层。

[0115] 每个卷积层包含3个操作:

[0116] 第一步:进行卷积运算,利用标准卷积方法或者分离卷积方法;

[0117] 第二步:将上一步得到的卷积结果进行批量归一化处理,将数据全部归一化为[0, 1]之间,有利于加快训练速度;

[0118] 第三步:经过激活函数Relu,将非线性特性引入到本发明的网络中,这样会保证输入输出之间是非线性映射关系,而不是简单的线性组合的关系,从而能够保证网络的学习能力。

[0119] 特征提取模块的输入是一张RGB图片,这张图片可以表示为 $a \times a \times 3$ 的矩阵形式,其中 a 为图片的宽度和高度,输出为三个特征矩阵,这三个特征矩阵的维度分别为 10×10 、 20×20 和 40×40 ,其中 40×40 的特征矩阵中每一个神经元的感受野是最小的,可以负责检测原始输入图像中的小车牌目标,同理, 10×10 的特征矩阵中每一个神经元的感受野是最大的,可以负责检测原始输入图像中的大车牌目标。这样进行多尺度预测,可以避免小目标车牌漏检的情况。

[0120] 如图2所示,特征提取模块的输入为大小为 $320 \times 320 \times 3$ 的图片,通过第一层标准卷积层C1,卷积核的大小为 3×3 ,个数为32,步长为2,第一卷积层 C1输出的特征图为 $160 \times 160 \times 32$ 。

[0121] 然后进入第二层分离卷积层C2,卷积核的大小为 3×3 ,个数为32,步长为1,第二卷积层C2输出的特征图为 $160 \times 160 \times 32$ 。

[0122] 然后进入第三层标准卷积层C3,卷积核的大小为 1×1 ,个数为64,步长为 1,第三卷积层C3输出的特征图为 $160 \times 160 \times 64$ 。

[0123] 然后进入第四层分离卷积层C4,卷积核的大小为 3×3 ,个数为64,步长为2,第四卷积层C4输出的特征图为 $80 \times 80 \times 64$ 。

[0124] 然后相继进入第五层标准卷积层C5、第六层分离卷积层C6、第七层标准卷积层C7,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为128,步长为1,最终C7输出的特征图为 $80 \times 80 \times 128$ 。

[0125] 然后进入第八层分离卷积层C8,卷积核的大小为 3×3 ,个数为128,步长为2,第八卷积层C8输出的特征图为 $40 \times 40 \times 128$ 。

[0126] 然后相继进入第九层标准卷积层C9、第十层分离卷积层C10、第十一层标准卷积层C11,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为256,步长为1,最终C11输出的特征图为 $40 \times 40 \times 256$ 。C11层的输出特征图会再输入到后续的网络结构中继续处理。

[0127] 然后进入第十二层分离卷积层C12,卷积核的大小为 3×3 ,个数为256,步长为2,第十二卷积层C12输出的特征图为 $20 \times 20 \times 256$ 。

[0128] 然后进入第十三层标准卷积层C13,卷积核的大小为 1×1 ,个数为512,步长为1,第十三卷积层C13输出的特征图为 $20 \times 20 \times 512$ 。

[0129] 然后进入C14-C23层,C14为分离卷积层C15为标准卷积层,卷积核的大小依次为 3×3 、 1×1 ,个数为512,步长为1,循环5次,共十层,最终C23 层的输出的特征图为 $20 \times 20 \times 512$ 。C23层的输出特征图输入到后续的网络结构中继续处理。

[0130] 然后进入第二十四层分离卷积层C24,卷积核的大小为 3×3 ,个数为512,步长为2,第二十四卷积层C24输出的特征图为 $10 \times 10 \times 512$ 。

[0131] 然后相继进入第二十五层标准卷积层C25、第二十六层分离卷积层C26、第二十七层标准卷积层C27,卷积核的大小依次为 1×1 、 3×3 、 1×1 ,个数为1024,步长为1,最终C27输出的特征图为 $10 \times 10 \times 1024$ 。C27层的输出特征图输入到后续的网络结构中继续处理。

[0132] C27层输出的特征图会在后续网络中输入到检测模块1中处理。如图3所示进入检测模块1后,会将C27层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_11 、 LC_12 、 LC_13 ,中路也经过三层卷积层(MC_11 、 MC_12 、 MC_13)。其中 LC_11 、 LC_12 、 LC_13 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_13 层的输出为 $10 \times 10 \times 512$; MC_11 、 MC_12 、 MC_13 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_13 层的输出为 $10 \times 10 \times 512$ 。然后将 LC_13 与 MC_13 层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_14 ,卷积核的大小为 1×1 ,个数为256,步长为1, C_14 层的输出为 $10 \times 10 \times 256$ 。右路经过一层卷积层 RC_11 ,卷积核的大小为 1×1 ,个数为256,步长为1, RC_11 的输出为 $10 \times 10 \times 256$,然后经过“拼接层” EC_1 层,将 RC_11 与 C_14 两层的输出合并为一个特征图, EC_1 层的输出为 $10 \times 10 \times 512$ 。

[0133] 经过检测模块1后, EC_1 层的输出会分为两路处理,其中一路经过两层标准卷积层 C_15 、 C_16 ,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_16 层的输出为 $10 \times 10 \times 18$ 。 C_16 层的输出即为网络输出特征图1。另外一路会经过标准卷积层 C_17 ,卷积核的大小为 1×1 ,个数分别为256,步长均为1, C_17 层的输出为 $10 \times 10 \times 256$ 。后经过“上采样”层 Up_1 ,放大尺寸两倍, Up_1 的输出为 $20 \times 20 \times 256$,后经过“拼接层” EC_1 层,将 Up_1 层与 C_17 层的输出合并, EC_1 层的输出为 $20 \times 20 \times 768$ 。

[0134] EC_1 层输出的特征图输入到检测模块2中处理。进入检测模块2后,将 EC_1 层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_21 、 LC_22 、 LC_23 ,中路也经过三层卷积层 MC_21 、 MC_22 、 MC_23 。其中 LC_21 、 LC_22 、 LC_23 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, LC_23 层的输出为 $20 \times 20 \times 512$; MC_21 、 MC_22 、 MC_23 层的卷积核大小分别为 1×1 、 1×3 、 3×1 ,个数分别为256、512、512,步长均为1, MC_23 层的输出为 $20 \times 20 \times 512$ 。然后将 LC_23 与 MC_23 层的输出特征图中的元素对应相加,将相加后的结果经过一层标准卷积层 C_24 ,卷积核的大小为 1×1 ,个数为256,步长为1, C_24 层的输出为 $20 \times 20 \times 256$ 。右路经过一层卷积层 RC_21 ,卷积核的大小为 1×1 ,个数为256,步长为1, RC_21 的输出为 $20 \times 20 \times 256$,然后经过“拼接层” EC_2 层,将 RC_21 与 C_24 两层的输出合并为一个特征图, EC_2 层的输出为 $20 \times 20 \times 512$ 。

[0135] EC_2 层的输出也会分为两路处理,其中一路经过两层标准卷积层 C_25 、 C_26 ,卷积核的大小均为 1×1 ,个数分别为512、18,步长均为1, C_26 层的输出为 $20 \times 20 \times 18$ 。 C_26 层的输出即为网络输出特征图2。另外一路会经过标准卷积层 C_27 ,卷积核的大小为 1×1 ,个数分别为128,步长均为1, C_27 层的输出为 $20 \times 20 \times 128$ 。后经过“上采样”层 Up_2 ,放大尺寸两倍, Up_2 的输出为 $40 \times 40 \times 128$,后经过“拼接层” EC_2 层,将 Up_2 层与 C_27 层的输出合并, EC_2 层的输出为 $40 \times 40 \times 384$ 。

[0136] EC_2 层输出的特征图输入到检测模块3中处理。进入检测模块3后,会将 EC_2 层输出的特征图分为左、中、右三条支路进行处理:左路经过三层卷积层 LC_31 、 LC_32 、 LC_33 ,中路也

经过三层卷积层MC₃1、MC₃2、MC₃3。其中LC₃1、LC₃2、LC₃3层的卷积核大小分别为1×1、1×3、3×1，个数分别为256、512、512，步长均为1，LC₃3层的输出为40×40×512；MC₃1、MC₃2、MC₃3层的卷积核大小分别为1×1、1×3、3×1，个数分别为256、512、512，步长均为1，MC₃3层的输出为40×40×512。然后将LC₃3与MC₃3层的输出特征图中的元素对应相加，将相加后的结果经过一层标准卷积层C₃4，卷积核的大小为1×1，个数为256，步长为1，C₃4层的输出为40×40×256。右路经过一层卷积层RC₃1，卷积核的大小为1×1，个数为256，步长为1，RC₃1 的输出为40×40×256，然后经过“拼接层”EC₃层，将RC₃1与C₃4两层的输出合并为一个特征图，EC₃层的输出为40×40×512。

[0137] 检测模块3的输出会经过两层标准卷积层C₃5、C₃6，卷积核的大小均为 1×1，个数分别为512、18，步长均为1，C₃6层的输出为40×40×18。C₃6 层的输出即为网络输出特征图3。

[0138] 整个网络结构如图4所示，网络的输出为3张特征图，维度分别是 10×10×18、20×20×18、40×40×18，其中10×10×18特征图共包含1800个参数，代表了网络将原输入图片分割成了100个网格，每个网格会预测三个预测框，每一个预测框对应了6个参数。这六个参数分别是预测框的4个坐标信息 t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 、预测置信度 \hat{C}_i 和类别概率 \hat{P}_i 。其中(t_{xi} 、 t_{yi})表示第i个预测框中心点的坐标参数值，(t_{wi} 、 t_{hi})表示第i个预测框宽和高的参数值，预测置信度 \hat{C}_i 表示第i个预测框包含目标的概率，类别概率 \hat{P}_i 为多维向量，表示了第i个预测框的目标是某一类别的概率。需要注意的一点是， t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 这四个参数是相对位置坐标，需要转化为最终在原始图片中的实际坐标。转换的公式如下：

$$[0139] \quad \hat{x}_i = \tanh(t_{xi}) + c_x$$

$$[0140] \quad \hat{y}_i = \tanh(t_{yi}) + c_y$$

$$[0141] \quad \hat{w}_i = p_w e^{t_{wi}}$$

$$[0142] \quad \hat{h}_i = p_h e^{t_{hi}}$$

[0143] 其中， t_{xi} 、 t_{yi} 、 t_{wi} 、 t_{hi} 是预测的相对坐标值， p_w 、 p_h 表示预测框对应锚框的宽度以及高度， c_x 、 c_y 表示预测框相对图片左上角位置坐标的偏移量， \hat{x}_i 、 \hat{y}_i 表示预测框中心点实际坐标， \hat{w}_i 、 \hat{h}_i 表示预测框的实际宽度以及实际高度。

[0144] 步骤四、利用步骤一得到的最终数据集对上述模型进行训练；

[0145] (1) 网络随机初始化权值，使初始化的值服从高斯正态分布。

[0146] (2) 输入数据经过本发明步骤三中的网络结构向前传播得到输出值为特征图1、特征图2、特征图3，并利用特征图信息得到预测边框的信息 $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{C}_i, \hat{P}_i)$ ；

[0147] (3) 将数据集中标注出的真实框与聚类得到的锚框进行匹配：计算出真实框所在的中心点，筛选出此中心点对应的锚框，选取与真实框有最大IOU值的锚框作为目标框，并将该真实框的坐标值信息赋给目标框，即得到目标框的坐标值 (x_i, y_i, w_i, h_i) ，并将目标框的类别值 P_i 设置为1，置信度值 C_i 设置为1，其余未标记的锚框的参数值均设置为0。

[0148] (4) 利用本发明提出的损失函数, 求出网络预测框的输出值与真实边界框的目标值之间的误差损失, 该损失函数包含位置损失、置信度损失和类别损失。

[0149] 位置损失分为中心损失和尺度损失。中心损失用于评估网络预测框中心和实际物体的标注框的中心间的误差; 尺度损失则用于评估网络预测框高度与宽度和实际物体标注框高度与宽度间的误差, 具体如下:

$$[0150] \quad x_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) [x_i \ln \hat{x}_i + (1 - x_i) \ln(1 - \hat{x}_i)]$$

$$[0151] \quad y_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)]$$

$$[0152] \quad center_loss = x_loss + y_loss$$

$$[0153] \quad size_loss = -\alpha \frac{1}{N} \sum_{i=1}^N l_i^{obj} (2 - w_i h_i) \left[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right]$$

[0154] 其中, N表示网络预测框的总数, l_i^{obj} 表示第i个预测框中是否存在目标, 若存在, $l_i^{obj} = 1$, 否则为0; (x_i, y_i) 表示目标所在第i个标注框真实中心位置, (\hat{x}_i, \hat{y}_i) 表示第i个预测框的中心位置, (w_i, h_i) 表示目标所在第i个标注框真实宽度和高度, (\hat{w}_i, \hat{h}_i) 表示第i个预测框的宽度和高度, α 用于调整尺度损失在所有损失中所占据的比例。

[0155] 置信度损失用于评估预测框对应的置信度和实际物体标注框对应的置信度之间的误差。置信度损失使用的函数为二值交叉熵函数, 具体如下:

$$[0156] \quad obj_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} [C_i \ln \hat{C}_i + (1 - C_i) \ln(1 - \hat{C}_i)]$$

$$[0157] \quad noobj_loss = -\frac{1}{N} \sum_{i=1}^N (1 - l_i^{obj}) [C_i \ln \hat{C}_i + (1 - C_i) \ln(1 - \hat{C}_i)]$$

$$[0158] \quad confidence_loss = obj_loss + noobj_loss$$

[0159] 其中, C_i 表示目标所在第i个标注框真实置信度, \hat{C}_i 表示第i个预测框的置信度。

[0160] 类别损失评估网络预测框对应的类别和实际物体对应的类别之间的误差, 具体如下所示:

$$[0161] \quad cls_loss = -\frac{1}{N} \sum_{i=1}^N l_i^{obj} [P_i \ln \hat{P}_i + (1 - P_i) \ln(1 - \hat{P}_i)]$$

[0162] 其中, p_i 表示目标所在第i个标注框中物体的类别的概率, \hat{p}_i 表示第i个预测框物体的类别概率。

[0163] 总损失函数为:

$$[0164] \quad loss = center_loss + size_loss + confidence_loss + cls_loss$$

[0165] (5) 利用Adam优化算法进行权值更新, 直到迭代次数 > epoch时, 结束训练。

[0166] 步骤五、利用已训练好的模型进行车牌检测, 并采用YOLOv3作为对比算法对本模型进行评估。本发明中, 检测过程采用IOU=0.8时的检测准确率、权重文件大小和检测时间作为算法性能的度量指标, 如果算法对某张图片的预测矩形框和该图片的真实矩形框间的

交并比大于0.8,那么就认为算法对该图片检测成功,其次,在检测准确率没有大幅降低的情况下,权重文件越小,检测时间越少,则认为算法性能越强。

[0167] 前述步骤中, N 表示网络预测框的总数,其中 $N=6300$, $i=1,2,\dots,N$ 表示预测框的标号。 M 表示分离卷积中假设的输入向量的通道个数, $j=1,2,\dots,M$ 。 C 表示分离卷积第二步中卷积核的个数, $d=1,2,\dots,C$ 。

[0168] 针对YOLOv3网络模型中的参数过多导致的权重文件过大;检测速度降低;对检测硬件设备的要求过高等问题。本发明针对于车牌这单一类别,基于深度卷积神经网络,从特征提取网络与检测网络两方面入手,提出了一种全新的卷积方法,减轻了原本网络参数数量,构建了一种高效轻量的网络结构。

[0169] 下面结合仿真实例对该发明做进一步的描述。

[0170] 仿真实例:

[0171] 本发明采用YOLOv3作为对比算法,将YOLOv3算法与本发明提出的网络模型在检测准确率、权重文件大小和检测时间三方面做性能对比。测试过程中场景的变化因素主要包括光照、车牌区域污染、车牌旋转以及天气干扰等。

[0172] 附表1给出了本发明所用各种算法性能对比。综合比较检测准确率、权重文件大小以及检测时间,本发明提出的网络模型性能最优,在检测准确率下降0.6%的基础上,将权重文件减小了78.3%,将检测时间减少了28.2%。

[0173] 图7为部分检测结果图,表示正常条件下、弱光照下和强光照条件下的车牌检测结果对比图。从结果图中可以看出,在检测精度上,YOLOv3算法与本发明算法的相差不大。

[0174] 综上仿真实验表明,本发明提出的高效深度卷积神经网络的车牌检测算法可以完成各种复杂环境中车牌检测任务,与YOLOv3相比,网络参数少,权重文件小,检测速度更加迅速。故而可以降低其检测硬件设备要求,应用到更多现实场景中。

[0175] 以上所述,仅是本发明的较佳实施例而已,并非对本发明做任何形式上的限制,虽然本发明已以较佳实施例揭露如上,然而并非用以限定本发明,任何熟悉本专业的技术人员,在不脱离本发明技术方案范围内,当可利用上述揭示的技术内容做出些许更动或修饰为等同变化的等效实施例,但凡是未脱离本发明技术方案的内容,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、等同变化与修饰,均仍属于本发明技术方案的范围。

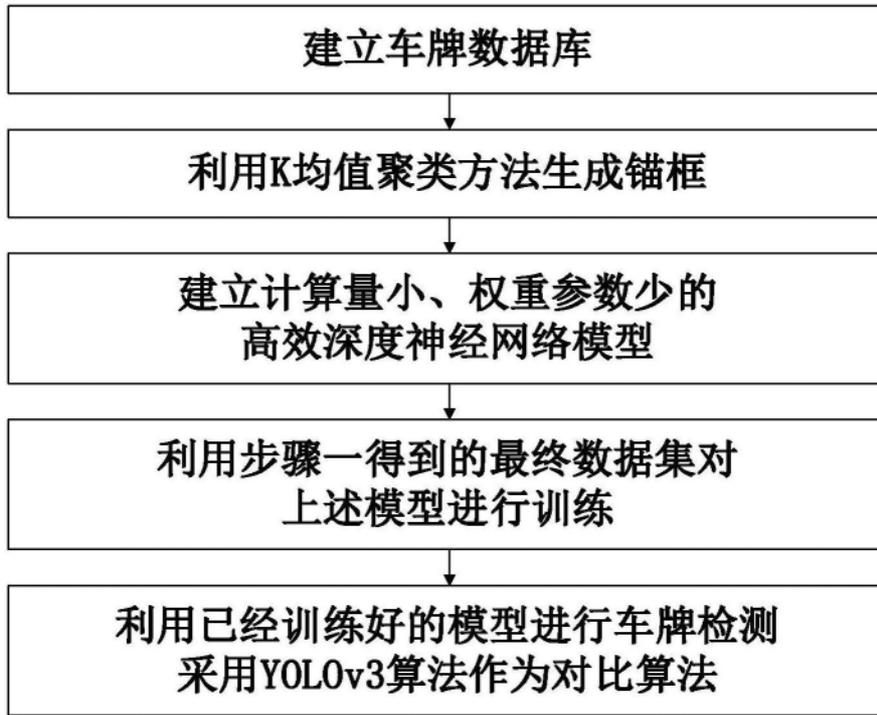


图1

卷积类型/步长	卷积核	输出特征图大小	
C1 标准卷积/2	3×3×32	160×160×32	
C2 分离卷积/1	3×3×32	160×160×32	
C3 标准卷积/1	1×1×64	160×160×64	
C4 分离卷积/2	3×3×64	80×80×64	
C5 标准卷积/1	1×1×128	80×80×128	
C6 分离卷积/1	3×3×128	80×80×128	
C7 标准卷积/1	1×1×128	80×80×128	
C8 分离卷积/2	3×3×128	40×40×128	
C9 标准卷积/1	1×1×256	40×40×256	
C10 分离卷积/1	3×3×256	40×40×256	
C11 标准卷积/1	1×1×256	40×40×256	→ 后续网络检测
C12 分离卷积/2	3×3×256	20×20×256	
C13 标准卷积/1	1×1×512	20×20×512	
×5	分离卷积/1	3×3×512	20×20×512
	标准卷积/1	1×1×512	20×20×512
			→ 后续网络检测
C24 分离卷积/2	3×3×512	10×10×512	
C25 标准卷积/1	1×1×1024	10×10×1024	
C26 分离卷积/1	3×3×1024	10×10×1024	
C27 标准卷积/1	1×1×1024	10×10×1024	→ 后续网络检测

图2

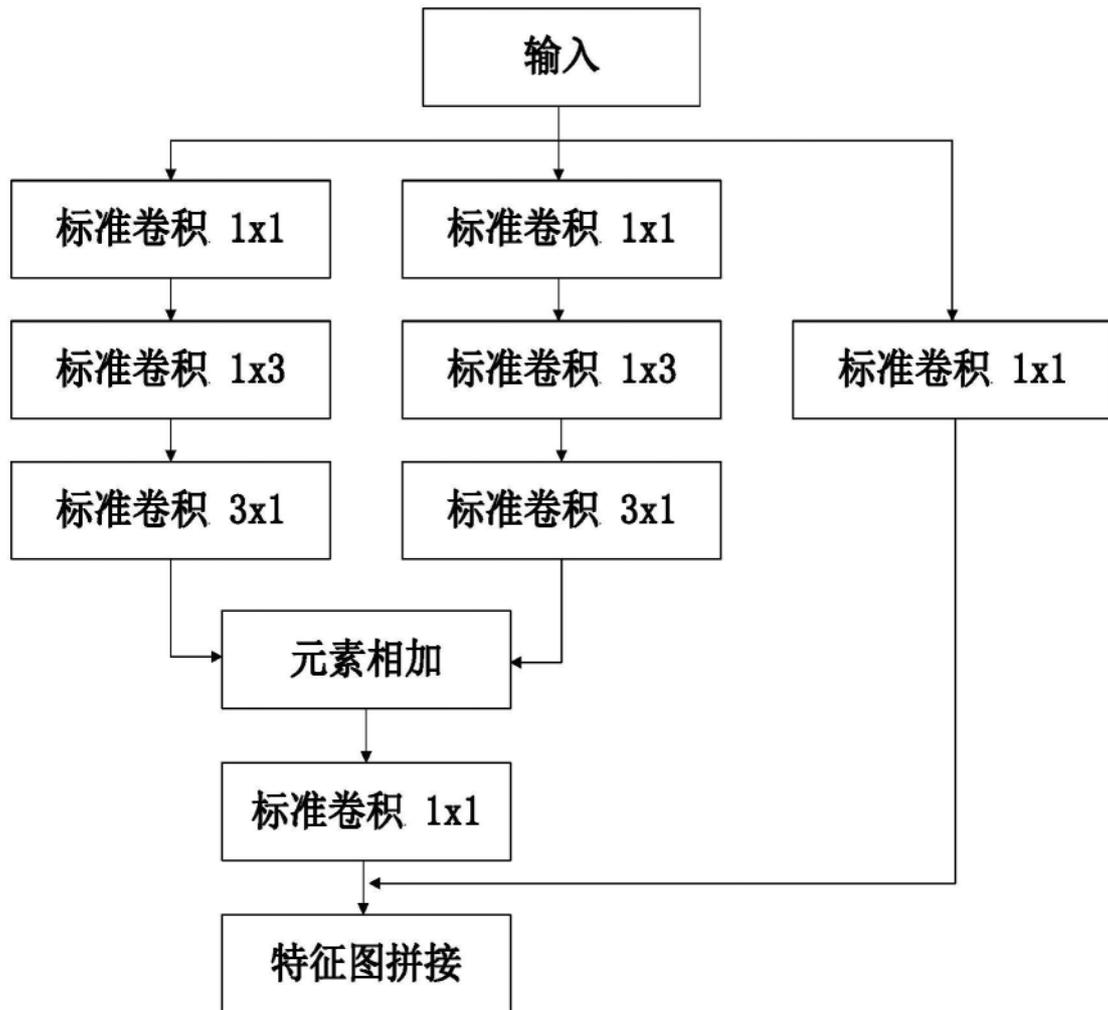


图3

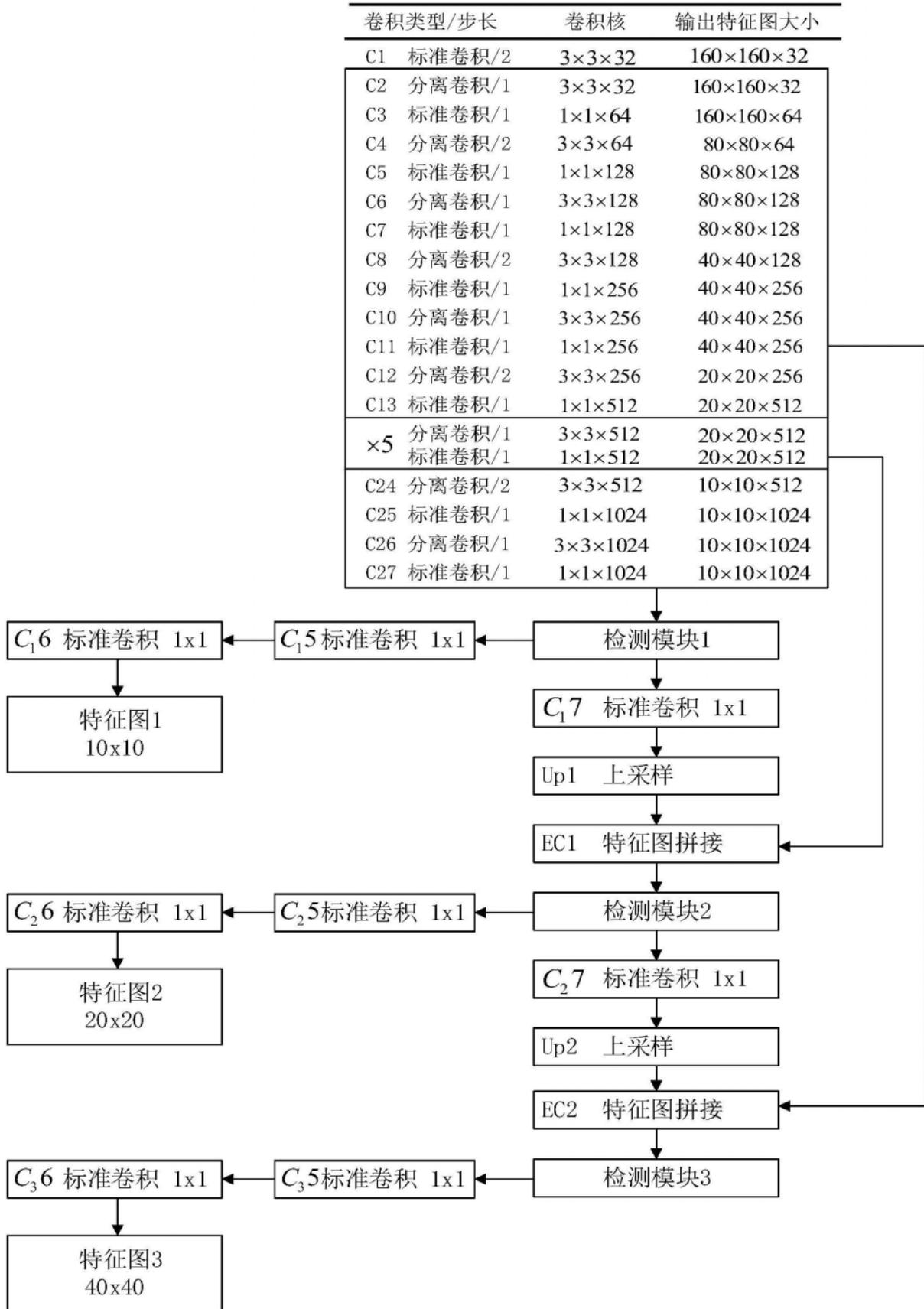


图4

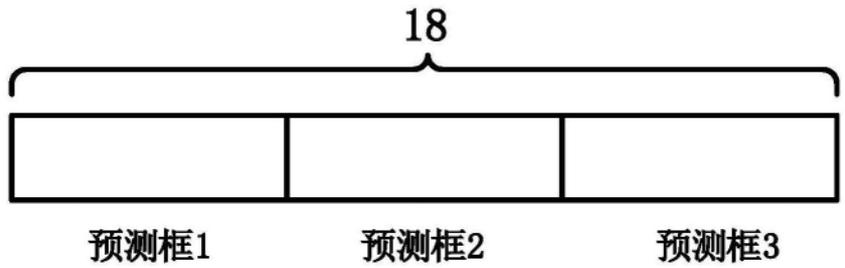


图5 (a)

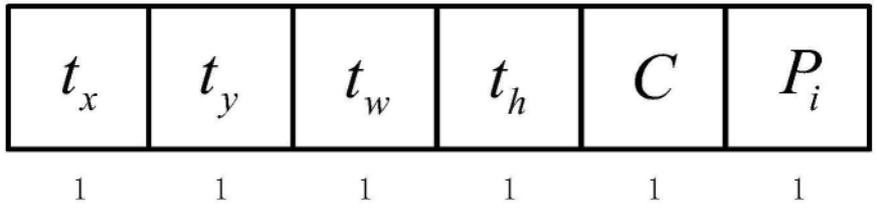


图5 (b)

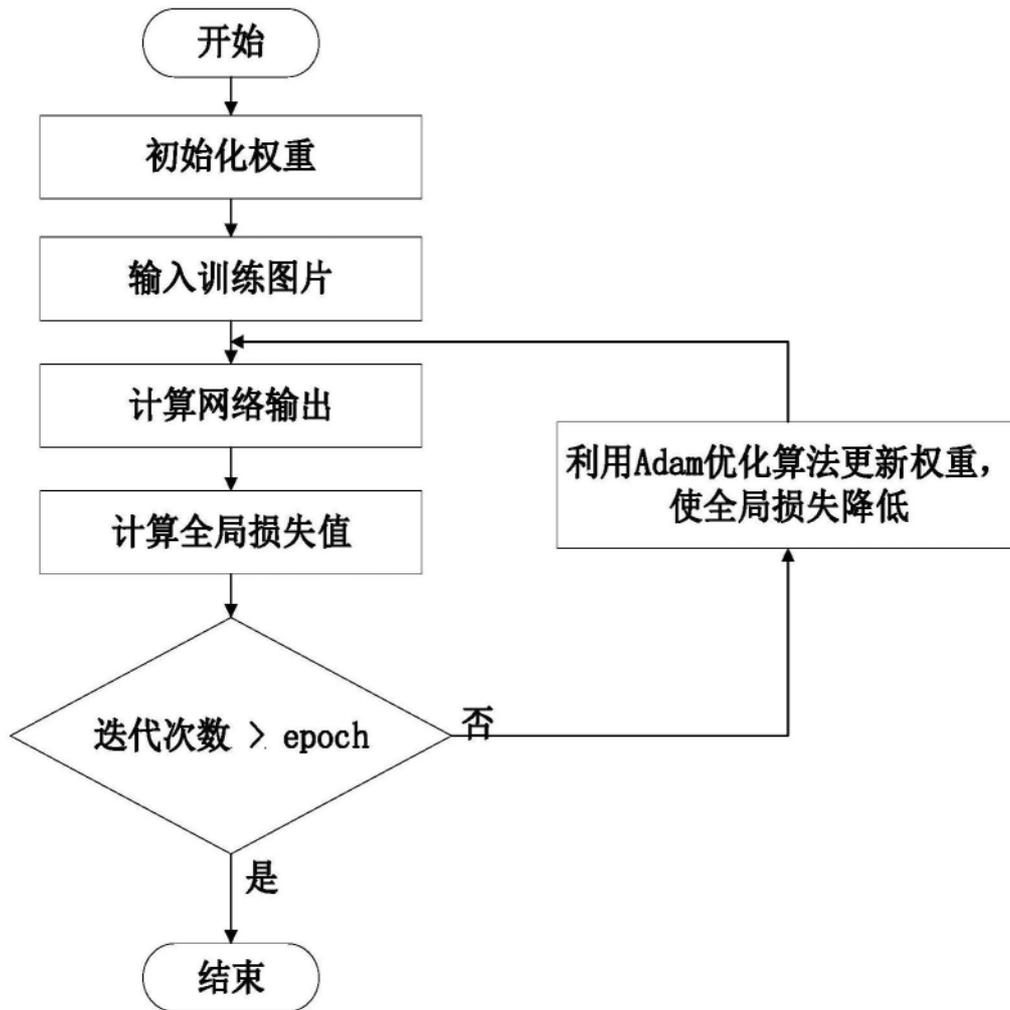


图6



(a) 正常条件下检测图



(b) 弱光照下检测图



(c) 强光照下检测图

图7