基于改进 YOLO v3 的目标检测算法

赵琼[1,2], 李宝清[1*], 李唐薇[1,2]

[1]中国科学院上海微系统与信息技术研究所微系统技术重点实验室, 上海 201800;
[2]中国科学院大学, 北京 100049

摘要 随着深度学习的不断发展与广泛运用,基于深度学习的目标检测算法已成为新的主流。为了进一步提高卷积神经网络 YOLO v3(You only look once v3)的检测精度,在原算法的网络结构上添加卷积层模块对样本进行目标背景分类,并粗略调整特征图上的锚框大小。该模块输出目标背景概率后,过滤掉背景概率值低于设定阈值的样本,从而解决原算法中存在的正负样本比例失衡的问题。使用调整过的锚框替代原算法中直接由聚类生成固定大小的锚框,该过程为边界框的预测提供更优的初始值。在 VOC 数据集上的实验结果表明,相较于原算法,改进的 YOLO v3 具有更高的检测精度。

关键词 机器视觉; 目标检测; 卷积神经网络; YOLO v3; 锚框
中图分类号 TP391.4 **文献标志码** A **doi**: 10.3788/LOP57.121502

Target Detection Algorithm Based on Improved YOLO v3

Zhao Qiong[1,2], Li Baoqing[1*], Li Tangwei[1,2]

[1] *Key Laboratory of Science and Technology on Microsystem, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai* 201800, *China*;
[2] *University of Chinese Academy of Sciences, Beijing* 100049, *China*

Abstract With the continuous development and wide applications of deep learning, target detection algorithms based on deep learning have become a new mainstream. To further improve the detection accuracy of the convolutional neural network YOLO v3 (You only look once v3), a convolution layer module was added to the network structure of the original algorithm to classify the target background of the sample and the anchor frame size of the feature map was roughly adjusted. To resolve the challenge of unbalanced proportion of positive and negative samples in the original algorithm, samples with background probability value less than the set threshold value were filtered by the module after outputting the target background probability. The adjusted anchor box was used to replace the anchor box of fixed sizes directly generated by clustering in the original algorithm. This process provides a better initial value for bounding box prediction. Experimental results on VOC dataset indicate that the improved YOLO v3 shows higher detection accuracy than the original algorithm.

Key words machine vision; target detection; convolutional neural network; YOLO v3; anchor boxes
OCIS codes 150.1135; 040.1880; 100.3008

1 引 言

目标检测在生活中多个领域有着广泛应用,如行人识别、自动驾驶和医学影像等。传统方法通常将特征提取与机器学习的分类算法相结合,Viola 等[1]使用哈尔特征(Haar-like feature)提取方法联合机器学习 Adaboost 算法训练一系列级联分类器以进行人脸检测,取得了可观效果;马娟娟等[2]利用改进的 Grassberger 熵来提取目标属性,同时使用随机森林分类器预测搜索框是否包含目标,目标检测的准确率得到提高。随着深度学习的不断发展,基于卷积神经网络(CNN)的目标检测算法逐渐替

收稿日期: 2019-09-19; **修回日期**: 2019-11-01; **录用日期**: 2019-11-02
基金项目: 微系统技术重点实验室基金(614280401020617)
* **E-mail**: sinoiot@mail.sim.ac.cn

代了传统目标检测算法，成为新的主流。Girshick等[3]提出的R-CNN(Region-CNN)成功将深度学习应用在目标检测中，先用Selective Search[4]算法选出候选框，再将候选框依次送入CNN中进行分类，但提取的候选框出现大量重叠，特征提取存在冗余。改进的Fast R-CNN[5]算法将整个图像作为输入送入卷积网络中，再将候选框映射在特征图上，避免了重复的特征提取，提高训练速度。Ren等[6]在Faster R-CNN中提出了锚框(anchor boxes)的概念，候选框的提取也使用卷积网络来实现，有效减少了候选框的提取时间。曹宇剑等[7]提出了旋转不变Faster R-CNN的目标检测算法，将其应用于低空无人机装甲的检测，与多模型比较，该算法取得了最好的检测效果。Faster R-CNN需将大小不同的候选框特征图送入全连接层中，故在全连接层前需插入感兴趣区域(ROI)层来固定候选框特征图的尺寸，但破坏了卷积网络的平移不变性。Dai等[8]将目标位置信息融入ROI层，构建位置敏感得分图，有效解决了上述问题。为了使网络能够适应更多尺寸的目标，Lin等[9]提出了特征金字塔结构，并将该结构应用于Faster R-CNN等算法，提升了对小尺度目标的检测性能。通常上述算法检测精度较高，但却不能达到实时效果。为了解决速度问题，Redmon等[10]使用单结构CNN直接预测目标的位置和类别，但精度稍低。随后Redmon等[11]又提出了改进的YOLO v2(You only look once v2)算法，在YOLO v1的基础上增加了批量归一化(Batch normalization)[12]层来加快训练速度，使用锚框和更高分辨率的分类器来提升精度。魏湧明等[13]通过更改候选框的筛选规则等方法改进YOLO v2网络，在无人机航拍图像定位任务中取得了较为理想的效果。基于上述单结构直接回归的思想，Liu等[14]提出了多尺度预测的SSD(Single shot multibox detector)算法，使得不同尺度的特征图专注预测不同尺寸的对象，兼顾了速度和精度，但预测层中浅层特征的表达能力较弱。为了加强浅层特征的表达能力，Fu等[15]通过解卷积的方式将上下文信息加入特征图中，检测精度得到进一步提升。王俊强等[16]结合基于候选框方法和单结构回归法的优势，提出了一种改进的SSD算法，在遥感影像数据集中取得了良好效果。继YOLO v2之后，Redmon等[17]又提出了YOLO系列第三个版本，即YOLO v3，该算法采用了一个分类效果更好的骨架网络结构Darknet53，同时利用多尺度特征进行预测，提升了小目标的识别率，保持了速度优势的同时又提升了检测精度。

像YOLO v3这种单结构直接回归方法属于密集型检测方法，在特征图的每个像素点上都预设固定数量的锚框，将每个锚框都设为一个训练样本，由单结构网络直接预测锚框相对于真实框的偏移量及锚框所属类别。这种检测方法会出现正负样本比例不平衡的情况，因为锚框与真实框匹配后，属于背景的负样本数量远远大于属于目标的正样本数量。为了平衡正负样本数量，SSD算法在训练过程中只选取一些损失较大的负样本，而在YOLO v3算法中，Lin等[18]试图使用一种改进的交叉熵损失函数(Focal loss)来控制正负样本在损失函数中的权重以减少负样本的权重，反而使模型的平均精度下降了几个百分点。

另外，很多高精度目标检测算法都依赖于高质量的锚框，设计越合理，检测精度越高。大部分目标检测算法都根据经验预设尺度和高宽比在空间域上生成锚框，如Faster R-CNN、SSD及基于这些的改进算法。YOLO v3同样采取了锚框策略，在数据集中聚类得到9组不同尺寸的锚框并将其作用于三种尺度的特征图。为了获取更佳的初始值，本文使用增加的卷积模块先对聚类得到的锚框作一个粗略调整，再使用原来的预测网络对调整后的锚框作进一步调优，同时为了平衡正负样本数量，使用增加的卷积模块对负样本提前进行过滤。

2 传统YOLO v3算法

YOLO v3网络结构由Darknet53和检测网络两部分组成，分别用作特征提取及多尺度预测。Darknet53由卷积层和残差层构成，残差层如图1所示，其中⊕表示一个相加操作，可用公式表示为

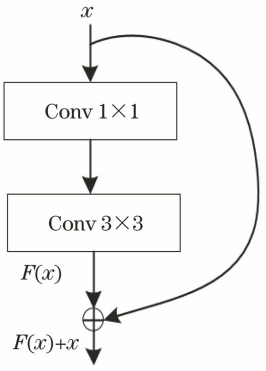

图1 残差层结构

Fig. 1 Residual layer structure

$$H(x)=F(x)+x, \tag{1}$$

式中：x 和 $F(x)$ 为残差层的两个输入，$F(x)$ 为 x 经过两次卷积操作后得到的结果；$H(x)$ 为残差层的输出。两次卷积操作的卷积核尺寸分别为 1×1 和 3×3，步长(stride)均为 1。同时为了提升网络性能，每层卷积后都添加了批量归一化层和线性单元(leaky ReLU)[19]。添加批量归一化层可加快训练的收敛速度，使用 leaky ReLU 形式的激活函数可避免深层网络出现梯度消失的现象。

图 2 为 Darknet53 的配置参数及尺寸为 416 pixel×416 pixel 的输入图像经 Darknet53 各层后的输出参数。其中 Times 表示残差操作循环次数，最终可获得尺寸为 208 pixel × 208 pixel，104 pixel×104 pixel，52 pixel×52 pixel，26 pixel×26 pixel 和 13 pixel×13 pixel 5 种尺度的特征图。

	type	filter	size	output
	convolutional	32	3×3	416×416
	convolutional	64	3×3/2	208×208
times 1	convolutional	32	1×1	208×208
	convolutional	64	3×3	208×208
	residual			208×208
	convolutional	128	3×3/2	104×104
times 2	convolutional	64	1×1	104×104
	convolutional	128	3×3	104×104
	residual			104×104
	convolutional	256	3×3/2	52×52
times 8	convolutional	128	1×1	52×52
	convolutional	256	3×3	52×52
	residual			
	convolutional	512	3×3/2	26×26
times 8	convolutional	256	1×1	26×26
	convolutional	512	3×3	26×26
	residual			
	convolutional	1024	3×3/2	13×13
times 4	convolutional	512	1×1	13×13
	convolutional	1024	3×3	13×13
	residual			

图 2 Darknet53 结构参数

Fig. 2 Darknet53 structural parameters

YOLO v3 的多尺度预测将在尺寸为52 pixel×52 pixel，26 pixel×26 pixel 和 13 pixel×13 pixel 的特征图上进行，但在特征图输出预测结果前，先进行特征融合操作，将高语义低分辨率与低语义高分辨率的特征拼接在一起，使得高分辨的特征也包含丰富的语义信息。具体特征融合过程：先在尺寸为 13 pixel×13 pixel 的特征图上进行 5 次卷积操作，卷积核尺寸依次为 1×1，3×3，1×1，3×3 和 1×1，步长均为 1；再连接卷积核尺寸为 3×3，步长为 1，卷积核数目减半的卷积层，实现降维效果；对特征进行二倍上采样(upsample)操作，再与上一级特征(尺寸为 26 pixel×26 pixel 的特征图)进行拼接，重复上述操作与尺寸为 52 pixel×52 pixel 的特征图进行拼接；最终在融合后尺寸为 52 pixel×52 pixel，26 pixel×26 pixel 和 13 pixel×13 pixel 的特征图上输出预测结果。

对于输出预测结果的三个特征图，在特征图上每个像素点格子预测三个框，每个预测框都预测中心坐标为 (x,y)，高和宽分别为 h 和 w，存在物体的置信度 p，k 个类别的得分值(COCO 数据集中 k 为 80，VOC 数据集中 k 为 20)。三层特征图一共输出 10647(13×13×3+26×26×3+52×52×3)个预测框。最后，将通过非极大值抑制(NMS)算法筛选出的预测框作为最终检测框。

3 改进的 YOLO v3 算法

3.1 改进的网络结构

改进的 YOLO v3 算法沿用 YOLO v3 的骨架网络，并在此基础上增加一个网络分支来调整锚框的宽和高，并对样本进行目标背景分类。在分支网络输出结果后设置一个阈值，根据分类的结果及设置的阈值生成二进制掩码矩阵 mask，当样本预测为背景的概率大于阈值时，则 mask 值为 0，否则为 1。将 mask 映射到最后一层(预测层)，mask 值为 1 的样本参与最后阶段的训练与检测。整体网络结构可简单表示为图 3，图中虚线框部分为添加的卷积层模块，分别连接在三种尺度的特征融合层后面。

图 4 为在 13 pixel×13 pixel 的特征图后添加的卷积模块参数。卷积模块由两个卷积层构成，其卷积核大小分别为 3×3 和 1×1，步长均为 1。添加在尺度为 26 pixel×26 pixel 及 52 pixel×52 pixel 的特征图后的网络层参数与图 4 类似，但需将 3×3 大小的卷积核个数由 1024 分别换成 512 和 256，最终在这三种尺寸的特征图上输出锚框的宽、高及目标背景的二分类结果。

经上述操作后，得到大小为 $h\times w\times 12$ 的三种特征图，其中 (h,w) 值分别为(13，13)，(26，26)和(52，52)，12 为特征图的通道数，可写成 3×(2+2)，3 代表特征图中每个像素都预测三个框，两个 2 分别代表每个框的高和宽及样本属于目标和背景的得分。最后，利用每个样本的目标、背景得分值及设定的阈值计算 mask，同时根据特征图输出宽和高的偏移值修正聚类得到的锚框。最终预测阶段，将 mask 和修正好的锚框作用于图 3 的预测层 predict。

3.2 锚框的粗调整

为了提高目标检测的精度，一些目标检测算法通常根据经验手动设置一些固定面积、宽高比的锚框，再由网络预测这些锚框相比于真实框的偏移量。

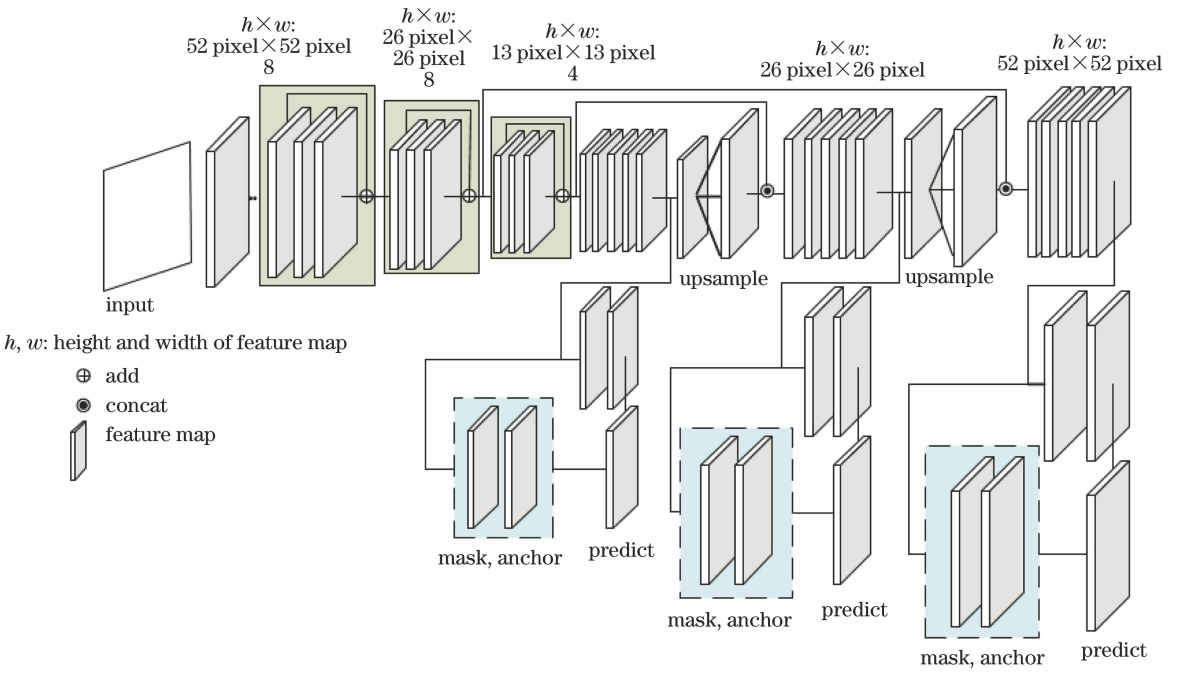

图 3 改进的网络结构

Fig. 3 Improved network structure

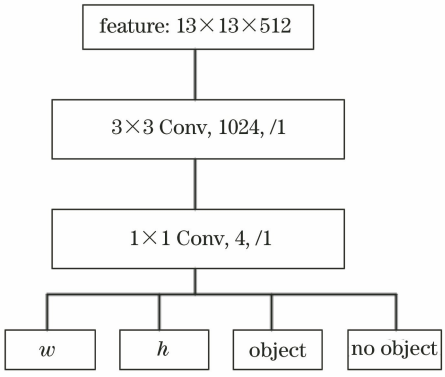

图 4 增加模块的网络结构参数

Fig. 4 Network structure parameters of increase module

在 YOLO v3 中通过对数据集中边界框的宽和高进行 K 均值(K-means)聚类得到 9 组锚框,依据锚框的大小分成三组,将每组三个锚框均匀分布在预测层特征图上,最后通过网络预测每个锚框的偏移量。为了进一步提升锚框的质量,通过添加的卷积网络模块对锚框的宽和高先作一个粗略调整,再将调整好的锚框送入最后的预测层进行更精确的回归。设添加网络模块中预测锚框相对于真实框的宽、高偏移分别为 t_w、t_h,可通过

$$b_w = p_w \exp(t_w), \tag{2}$$

$$b_h = p_h \exp(t_h), \tag{3}$$

计算边界框实际的宽 b_w 和高 b_h。式中:p_w 和 p_h 分别为由聚类得到的锚框的宽和高。

最终阶段边界框的预测过程和锚框调整过程相同,均由网络直接预测边界框相对于锚框的偏移量 t_w 和 t_h,但这个锚框不再由聚类的形式获取,而是由增加的网络模块生成,网络生成的锚框为最后阶段边界框的预测提供了更好的初始值。边界框中心点坐标由目标所在网格坐标及中心点相较所在网格的偏移量组成,可表示为

$$b_x = C_x + \sigma(t_x), \tag{4}$$

$$b_y = C_y + \sigma(t_y), \tag{5}$$

式中:t_x,t_y 为目标中心点相对于其所在网格左上角的偏移值;C_x,C_y 为网格与图像左上角的横纵距离;σ 为 Sigmoid 函数。在计算中心点坐标前先对偏移量 t_x 和 t_y 进行 Sigmoid 变换,将偏移量限制在 0~1 之间,以确保预测的中心点落在目标所在网格内。

3.3 正负样本预测

单结构直接回归方法普遍存在的一个问题就是正负样本比例不平衡,特征图上每个像素点设定固定数量的锚框,与真实框匹配后大部分像素点都属于背景,即负样本,负样本数量远远大于正样本数量。通常一些算法采取某种策略来缓解这个问题,如 SSD 算法在网络训练时将正负样本数量设为 1∶3 以减少负样本数量,RetinaNet 算法使用降低负样本权重的 Focal loss 函数等。

为了解决正负样本不平衡的问题,在 YOLO v3 特征融合层后添加卷积模块对样本进行目标背景分

类，再设置一个阈值（设为 0.03）生成 mask，对于预测为负样本且得分高于阈值的位置，mask 值设为 0，其他位置 mask 值设为 1。最后在预测阶段将 mask 映射到预测层上，mask 值为 0 位置的样本将不参与训练及预测。

3.4 损失函数

改进的 YOLO v3 损失函数由两部分组成，一部分是增加模块部分的损失函数 L_{add}，另一部分是原检测网络损失函数 L_{det}。其中 L_{add} 分为两个部分，锚框的宽高损失 L_{size} 及前景背景类别损失 L_{conf1}，表达式为

$$L_{\text{size}}=\sum_{i=0}^{s_1^2+s_2^2+s_3^2}\sum_{j=0}^{B}l_{ij}^{obj}(2-w_{ij}\times h_{ij})\times[(t_{w_{ij}}-\hat{t}_{w_{ij}})^2+(t_{h_{ij}}-\hat{t}_{h_{ij}})^2],\tag{6}$$

$$L_{\text{conf1}}=\sum_{i=0}^{s_1^2+s_2^2+s_3^2}\sum_{j=0}^{B}\alpha_{ij}^{\text{t}}(1-p_{ij}^{\text{t}})^{\gamma}\ln(p_{ij}^{\text{t}}),\tag{7}$$

$$p=\frac{1}{1+\exp(s)},\tag{8}$$

$$L_{\text{add}}=L_{\text{size}}+L_{\text{conf1}},\tag{9}$$

式中：s_1，s_2 和 s_3 为特征图的尺寸，多尺度预测分别在 13 pixel × 13 pixel，26 pixel × 26 pixel 和 52 pixel×52 pixel 三种尺寸的特征图上进行，此时 $s_1=13$，$s_2=26$，$s_3=52$；B 为锚框个数，特征图中每个网格（grid）预测三个锚框，此时 $B=3$；i 为预测层特征图中第 i 个特征网格（三种尺度共有 13×13+26×26+52×52 个特征网格）；l_{ij}^{obj} 为正样本在损失函数中的权重，根据匹配原则：先由数据集中目标的真实框聚类得到九组锚框，再根据锚框大小分为三组分配在三种不同尺度预测层的每个特征网格上，计算目标真实框与锚框的 IoU（并集与交集的面积之比），与真实框有最大 IoU 值且中心点落在同一特征网格的锚框负责预测此目标，将每个锚框都看作是一个样本，若第 i 个特征网格中第 j 个锚框负责预测目标，就将此锚框看作正样本，赋予其正样本标签，此时 $l_{ij}^{\text{obj}}=1$，反之就将锚框看作负样本，$l_{ij}^{\text{obj}}=0$；w_{ij}，h_{ij} 为真实框的宽高；$\hat{t}_{w_{ij}}$，$\hat{t}_{h_{ij}}$ 为网络预测的边界框宽高偏移值；$t_{w_{ij}}$，$t_{h_{ij}}$ 为真实框的宽高偏移值，其中宽高偏移是边界框相对于锚框的宽高偏移值（增加网络模块是相对于聚类的锚框，最终预测部分是相对于增加模块部分生成的锚框）；α_{ij}^{t} 为控制正负样本权重的参数，根据匹配原则：样本标签为正，$\alpha_{ij}^{\text{t}}=\alpha$，样本标签为负，$\alpha_{ij}^{\text{t}}=1-\alpha$；$\gamma$ 为难分样本和易分样本的调制系数，γ 和 α 都为设定好的固定值；s_{ij} 为网络将第 i 个网格中第 j 个样本预测为正的得分，使用 Sigmoid 函数[(8)式]作为其概率输出，便可得到网络将样本预测为正的概率 p_{ij}，p_{ij}^{t} 为该样本实际属于正样本的概率，根据匹配原则：若该样本的真实标签为正样本时，$p_{ij}^{\text{t}}=p_{ij}$，反之 $p_{ij}^{\text{t}}=1-p_{ij}$。

最后检测网络部分的损失函数 L_{det} 由三部分组成，坐标损失 L_{loc}、置信度损失 L_{conf2} 和类别损失 L_{class}，表达式为

$$L_{\text{det}}=L_{\text{loc}}+L_{\text{conf2}}+L_{\text{class}},\tag{10}$$

$$L_{\text{loc}}=\sum_{i=0}^{s_1^2+s_2^2+s_3^2}\sum_{j=0}^{B}l_{ij}^{\text{mask}}l_{ij}^{obj}(2-w_{ij}\times h_{ij})\times[(t_{x_{ij}}-\hat{t}_{x_{ij}})^2+(t_{y_{ij}}-\hat{t}_{y_{ij}})^2+(t_{w_{ij}}-\hat{t}_{w_{ij}})^2+(t_{h_{ij}}-\hat{t}_{h_{ij}})^2],\tag{11}$$

$$L_{\text{conf2}}=\sum_{i=0}^{s_1^2+s_2^2+s_3^2}\sum_{j=0}^{B}l_{ij}^{\text{mask}}[l_{ij}^{\text{obj}}E(C_{ij},\hat{C}_{ij})+l_{ij}^{\text{nobj}}E(C_{ij},\hat{C}_{ij})],\tag{12}$$

$$E(p,\hat{p})=-p\ln(\hat{p})-(1-p)\ln(1-\hat{p}),\tag{13}$$

$$L_{\text{class}}=\sum_{i=0}^{s_1^2+s_2^2+s_3^2}\sum_{j=0}^{B}\sum_{c\in\text{classes}}l_{ij}^{\text{mask}}l_{ij}^{obj}E[p_{ij}(c),\hat{p}_{ij}(c)],\tag{14}$$

式中：l_{ij}^{mask} 为第 i 个网格中第 j 个样本是否被过滤，增加模块的二分类网络预测该样本属于负样本的概率后，设定一个固定阈值，若属于负样本的概率高于该阈值，则表示该样本将被过滤，不参与训练和预测，此时 $l_{ij}^{\text{mask}}=0$，反之 $l_{ij}^{\text{mask}}=1$；$\hat{t}_{x_{ij}}$，$\hat{t}_{y_{ij}}$ 为网络预测的边界框中心偏移；$t_{x_{ij}}$，$t_{y_{ij}}$ 为数据集中目标边界框的中心偏移值，中心偏移表示边界框相对于所处网格左上角的偏移量；C_{ij}，C_{ij} 分别为每个样本真实的置信度及预测的置信度，根据匹配原则：若样本标签为正，则 $C_{ij}=1$，反之 $C_{ij}=0$，$\hat{C}_{ij}$ 是在网路输出置信度得分值后，将得分值作为(8)式 Sigmoid 函数的输入来得到概率值，最后将概率作为(13)式的二值交叉熵损失函数的输入，计算置信度损失；l_{ij}^{nobj} 为训练过程中负样本权重，当第 i 个网格内第 j 个预测框与真实框的 IoU 大于 0.5 时，$l_{ij}^{\text{nobj}}=0$，反之 $l_{ij}^{\text{nobj}}=1$；classes 为数据集中包含的类别，多类别预测可看成多个二分类问题；c 为属于 classes 中的某个类别，在网络预测样本属于某个类别 c 的得分值后，仍将

得分值作为 Sigmoid 函数的输入，得到样本的真实类别概率$\hat{p}_{ij}(c)$，若样本所属类别为 c，$p_{ij}(c)=1$，反之 $p_{ij}(c)=0$。

4 实验结果分析

实验仿真在 TensorFlow 框架下进行，训练及测试的计算机硬件配置 CPU 为 Intel XeonE5-2620 V4，GPU 为 NVIDIA GeForce GTX 1080Ti，操作系统系统为 Ubuntu 14.04。

4.1 实验数据集

实验选取标准化数据集为 PASCAL VOC 进行图像识别和分类，其包含 20 种类别的数据。模型训练阶段，选取 VOC2012 训练验证集及 VOC2007 训练验证集作为训练数据（共有 16551 张），将 VOC2007 训练验证集部分数据作为验证集。模型测试阶段，选取 VOC2012 测试集作为测试数据（共有 4952 张）。

4.2 实验细节

采用端对端的方式来优化模型，使用多任务损失函数来优化网络参数。锚框由 VOC 数据集聚类而成，得到三种尺度 9 个锚框，宽高分别为(24×34)，(46×84)，(68×185)，(116×286)，(122×97)，(171×180)，(214×327)，(324×193)和(359×359)。

整个训练过程中使用批量随机梯度下降法来优化损失函数，共进行 60000 次迭代。初始学习率设为 0.01，权重衰减值设为 0.0005，批量大小设为 64，在网络迭代 20000 次及 50000 次后，学习率分别设为 0.001 和 0.0001。

利用 TensorFlow 工具中 TensorBoard 查看训练过程中损失函数曲线，如图 5 所示，横坐标代表迭代次数，纵坐标代表损失值。从图 5可以看到，刚开始损失函数曲线下降较快，迭代到 15000 次后下降变缓，最后收敛至 0.1 左右。

4.3 实验结果及性能对比

使用平均精度（mAP）对算法性能进行评估。所有测试都选取 VOC2007 训练验证集和 VOC2012 训练验证集作为训练数据，选取 VOC2007 测试集作为测试数据。表 1 为不同目标检测算法在 VOC2007 测试集上的实验结果，其中 Faster R-CNN、R-FCN（Region based Fully Convolutional Network）、SSD321、SSD500 和 DSSD513(Deconvolutional Single Shot Detector)的实验结果来自文献[15]，YOLO v2 的实验结果来自文献[11]，YOLO v3 与改进的 YOLO v3 实验结果由实验得到(网络输入尺寸均为 416 pixel)。

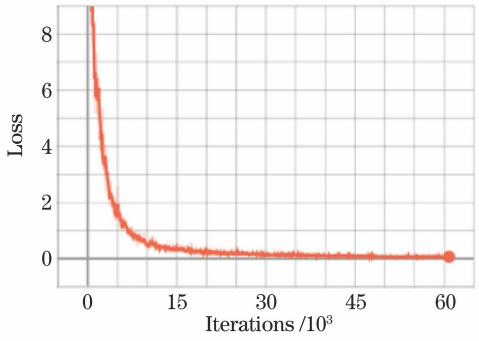

图 5 损失函数训练曲线

Fig. 5 Loss function training curve

与基于区域的目标检测算法 Faster R-CNN(VGG)和 Faster R-CNN(Residual-101)的检测结果相比，改进的 YOLO v3 算法的检测精度分别提高 7.0 个百分点和 3.8 个百分点，与 R-FCN 的检测效果相当。与一次性回归的检测算法对比，其检测精度比 SSD321，YOLO v2，YOLO v3 分别提高 3.1 个百分点，3.4 个百分点和 0.8 个百分点，比 SSD500(Residual-101)低 0.4 个百分点。

表 1 各种算法在 VOC2007 数据集上的测试结果

Table 1 Test results of various algorithms on VOC2007 dataset

Algorithm	Network	Data	mAP /%
Faster R-CNN	VGG	VOC2007+2012	73.2
Faster R-CNN	Residual-101	VOC2007+2012	76.4
R-FCN	Residual-101	VOC2007+2012	80.5
SSD321	Residual-101	VOC2007+2012	77.1
SSD500	Residual-101	VOC2007+2012	80.6
YOLO v2_416	Darknet19	VOC2007+2012	76.8
YOLO v3_416	Darknet53	VOC2007+2012	79.4
Ours	Darknet53	VOC2007+2012	80.2

4.4 数据增强对模型精度的影响

为了提高模型的泛化能力同时增加训练的数据量，实验采取数据增强的策略对输入的图像分别进行水平翻转变换、随机裁剪、色彩抖动和平移变换处理。表 2 为使用数据增强及不使用数据增强模型的检测精度，可以看到使用数据增强后，模型的检测精度有 0.27 个百分点的提升。

表 2 数据增强前后对模型精度的影响

Table 2 Impact of data enhancement on model accuracy

Dataset	mAP /%	
	Before	After
VOC2007	80.20	80.47

4.5 不同模型对训练速度的影响

在原网络中添加的二分类网络和回归网络是共享权重参数，如图 6(b)所示。图 6(a)将二分类网络与回归网络分成两个支路，与图 6(b)相比，模型参数必然有所增加，需被优化的参数增多，网络收敛速度也将会变慢。

使用早停法训练图 6 中两种模型，计算从开始训练到停止训练的时间并进行对比，如表 3 所示。

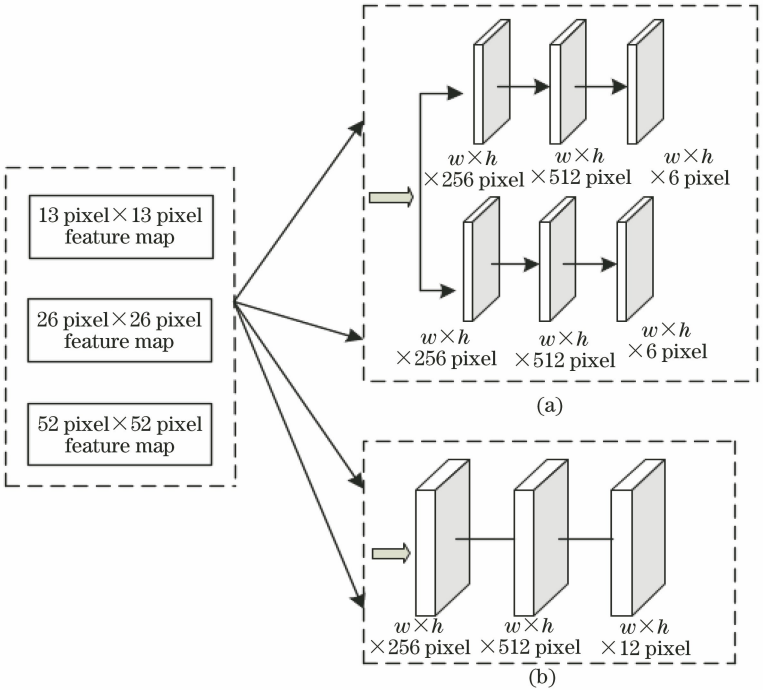

图 6 不同方案模型图。(a)第一种；(b)第二种

Fig. 6 Model diagram of different schemes. (a) 1st kind; (b) 2nd kind

表 3 不同模型对模型训练时间的影响

Table 3 Influence of different models on model training time

Model	Solution one	Solution two
Time /h	142	134

从表 3 可以看到，实验所采用的模型将两个网络分支合并从而让两个网络共享权重参数，可缩短模型训练时间。

4.6 测试集上的实验效果图

测试集上的实验效果如图 7 所示。从图 7 可以看到，除个别遮挡较为严重的物体未被检测出外，所提算法基本上能够检测图像中极大部分目标，且定位较为准确，具有良好的检测效果。

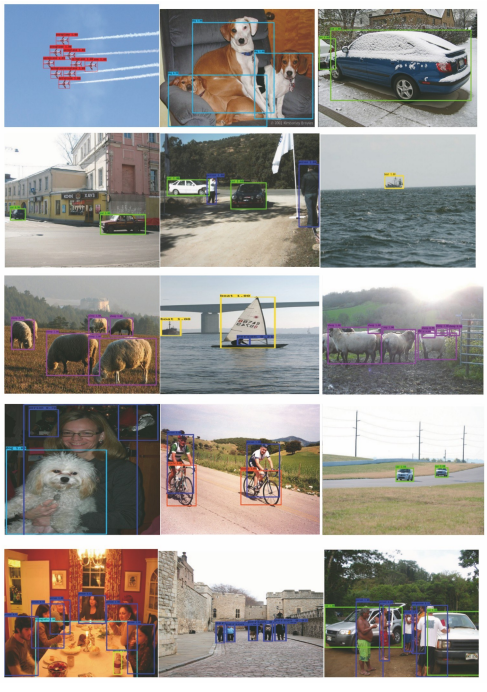

图 7 测试集上的实验效果图

Fig. 7 Experimental renderings on test dataset

5 结 论

提出了基于 YOLO v3 的改进算法，在 YOLO v3的基础上增加了 CNN 层以进行正负样本筛选及锚框的粗调整。利用 VOC 数据集进行实验，证明改进的方法使得模型的检测精度有所提升，设计的前景目标二分类网络及锚框回归网络有效。从评估指标 mAP 可以看到，与大多数目标检测算法相比，所提算法具有更好的检测性能。从实验效果图可以看到，所提算法对遮挡较为严重的物体，检

测效果略有下降，因此在后续工作中将研究如何提升遮挡严重目标的检测效果。

参 考 文 献

[1] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]// Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, December 8-14, 2001, Kauai, HI, USA. New York: IEEE, 2001: 511-518.

[2] Ma J J, Pan Q, Liang Y, et al. Object detection based on improved Grassberger entropy random forest classifier[J]. Chinese Journal of Lasers, 2019, 46(7): 0704011.
马娟娟, 潘泉, 梁彦, 等. 基于改进 Grassberger 熵随机森林分类器的目标检测[J]. 中国激光, 2019, 46(7): 0704011.

[3] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE, 2014: 580-587.

[4] Uijlings J R R, van de Sande K E A, Gevers T, et al. Selective search for object recognition [J]. International Journal of Computer Vision, 2013, 104 (2): 154-171.

[5] Girshick R. Fast R-CNN [C] // 2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 1440-1448.

[6] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[7] Cao Y J, Xu G M, Shi G C. Low altitude armored target detection based on rotation invariant Faster R-CNN[J]. Laser & Optoelectronics Progress, 2018, 55(10): 101501.
曹宇剑, 徐国明, 史国川. 基于旋转不变 Faster R-CNN 的低空装甲目标检测[J]. 激光与光电子学进展, 2018, 55(10): 101501.

[8] Dai J, Li Y, He K, et al. R-FCN: object detection via region-based fully convolutional networks[C]// Conference and Workshop on Neural Information Processing Systems, December 5-10, 2016, Barcelona, Spain. New York: Curran Associates, 2016: 379-387.

[9] Lin T Y, Dollar P, Girshick R, et al. Feature pyramid networks for object detection[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI. New York: IEEE, 2017: 2117-2125.

[10] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 779-788.

[11] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI. New York: IEEE, 2017: 7263-7271.

[12] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift[EB/OL]. (2015-03-02)[2019-11-01]. https://arxiv.org/abs/1502.03167.

[13] Wei Y M, Quan J C, Hou Y Q Y. Aerial image location of unmanned aerial vehicle based on YOLO v2[J]. Laser & Optoelectronics Progress, 2017, 54 (11): 111002.
魏湧明, 全吉成, 侯宇青阳. 基于 YOLO v2 的无人机航拍图像定位研究[J]. 激光与光电子学进展, 2017, 54(11): 111002.

[14] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[M]// Leibe B, Matas J, Sebe N, et al. Computer vision-ECCV 2016. Lecture notes in computer science. Cham: Springer, 2016, 9905: 21-37.

[15] Fu C, Liu W, Ranga A, et al. DSSD: deconvolutional single shot detector[EB/OL] (2017-01-23)[2019-11-01]. https:// arxiv.org/abs/1701.06659.

[16] Wang J Q, Li J S, Zhou X W, et al. Improved SSD algorithm and its performance analysis of small target detection in remote sensing images[J]. Acta Optica Sinica, 2019, 39(6): 0628005.
王俊强, 李建胜, 周学文, 等. 改进的 SSD 算法及其对遥感影像小目标检测性能的分析[J]. 光学学报, 2019, 39(6): 0628005.

[17] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL] (2018-04-08)[2019-11-01]. https://arxiv.org/abs/1804.02767.

[18] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]// 2017 IEEE International

Conference on Computer Vision (ICCV), October 22-29, 2017, Venice. New York: IEEE, 2017: 2980-2988.

[19] Fan B, Niu J C, Zhao J. Three-phase full-controlled rectifier circuit fault diagnosis based on optimized neural networks [C] // 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), August 8-10, 2011, Deng Feng, China. New York: IEEE, 2011: 6048-6051.