

FasterR-CNN

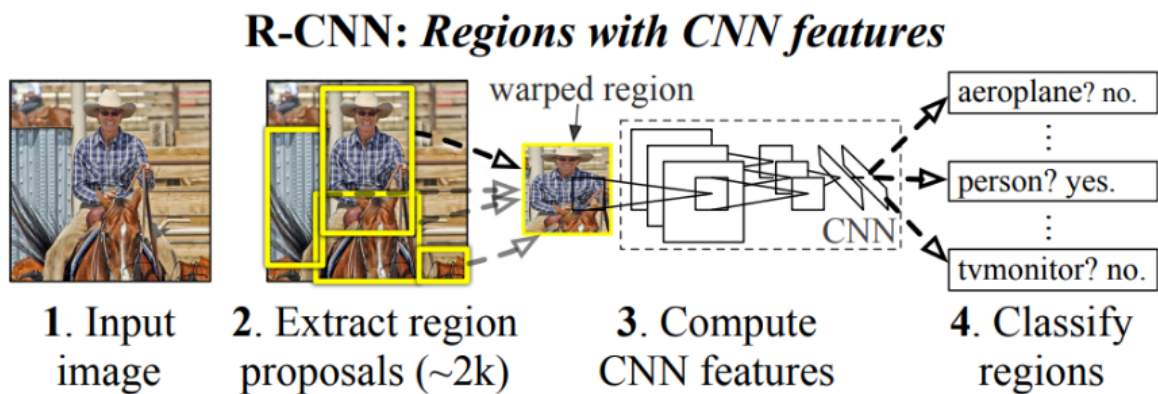
单阶段的目标检测算法已经在井盖识别上具有较好的表现，其突出优势在于极快的检测速度，但其存在的缺陷是无法在准确率上有更大的突破与提升，在许多的应用场景中，我们无需追求较快的检测速度，更重要的是较高的准确率，所以我们团队又引入了双阶段的目标检测算法，对Faster-RCNN算法作出创新与改进，其获得了更高的准确率表现。

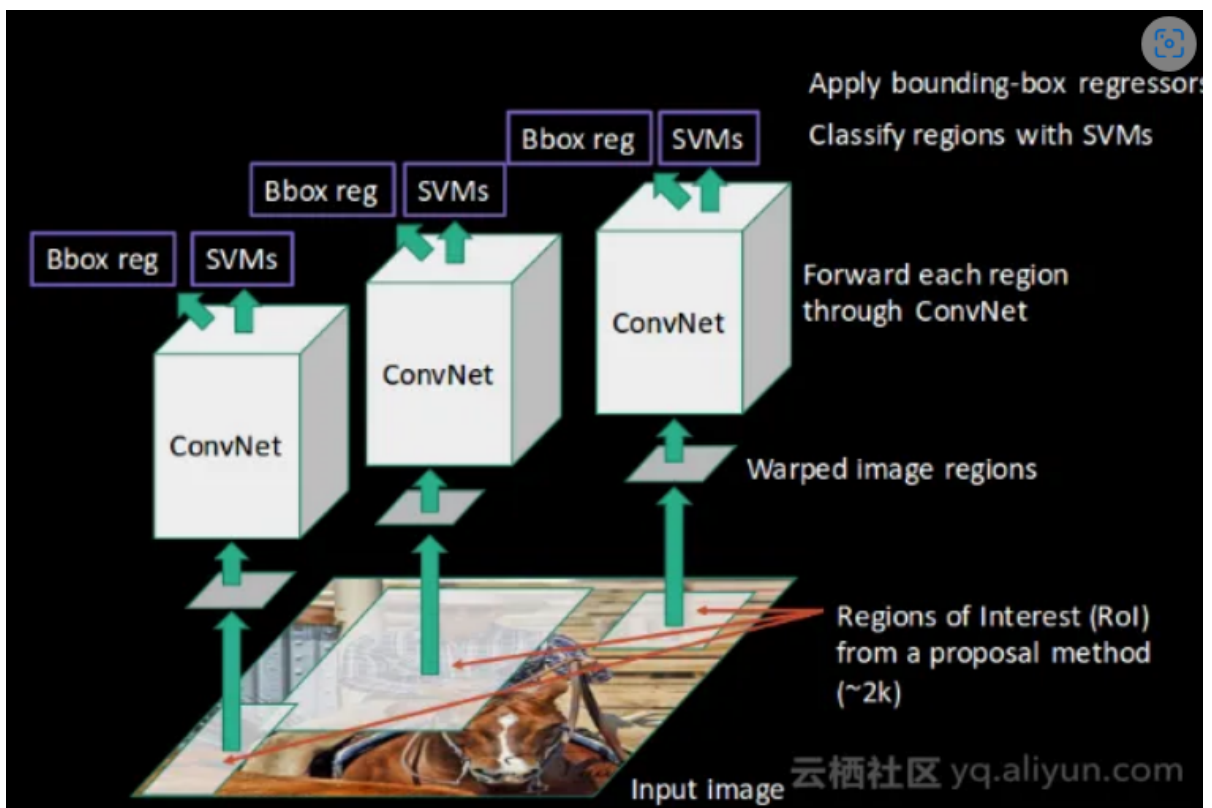
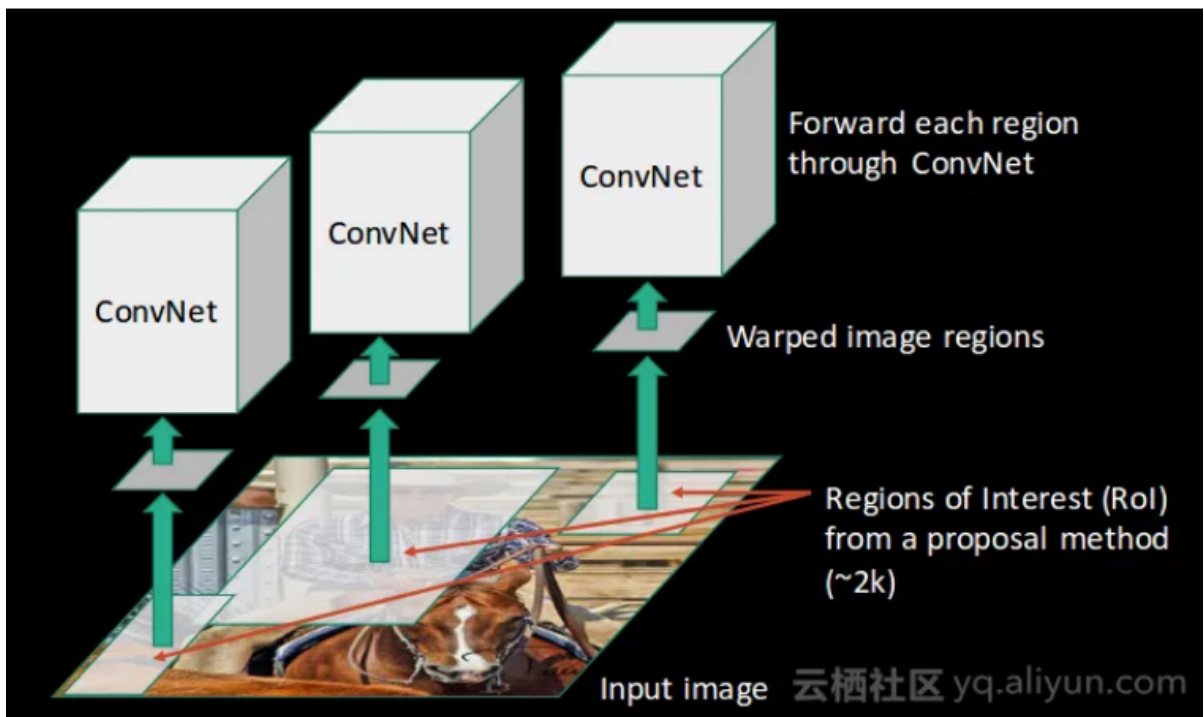
双阶段的目标检测算法，“双阶段”又称基于候选区域（Region proposal）的目标检测。其相对于YOLO等单阶段目标检测算法最大的不同是将目标检测分为两个阶段：第一个阶段先获取输入图像，提取候选区域；第二个阶段再对候选区域进行分类识别。通过增加第一阶段目标与背景的分离提取，使得对目标的定位与识别更加精确和准确，双阶段的算法能同时进行目标检测、分类和定位等多个任务的学习，从而更好地利用数据和提升模型性能。

R-CNN

在R-CNN之前，目标检测通常基于手工设计的特征和传统的机器学习算法，如SVM（支持向量机）和随机森林。这些方法在复杂的场景中往往无法提供准确的检测结果。RCNN通过引入深度学习的卷积神经网络（CNN），利用其强大的特征学习能力，极大地改进了目标检测的准确性和性能。RCNN的核心思想是将目标检测问题转化为一系列的候选区域（region proposal）的分类问题。

RCNN第一步需要获取目标区域，它改变了传统分割区域的方法，传统分割区域的方法一为穷举法，遍历图像的每一个像素；二为使用分割算法，预测前景分割区域是一个完整对象区域的可能性大小。它使用选择性搜索，产生图像初始区域后使用贪心算法对区域进行迭代分组，组合两个最相似的区域。第二步对候选区域使用深度网络提取特征，将候选区域缩放到统一大小。第三步使用SVM分类器进行区域类别的判定。最后使用边界框回归预测每个已识别区域的边界框。



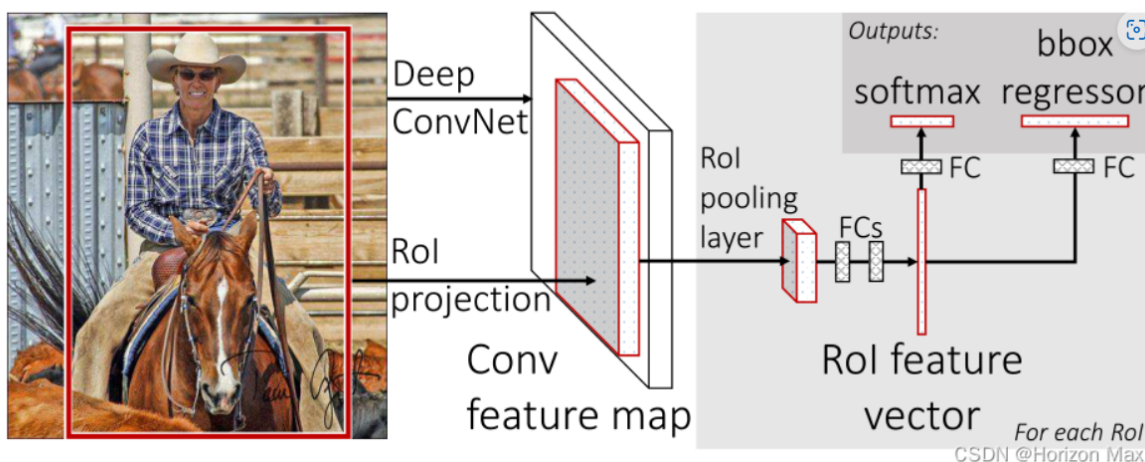


FastR-CNN

R-CNN的训练分多步，针对每个类别都要训练一个SVM分类其，最后还要对每个类别进行边框预测，步骤比较繁琐，且时间和内存消耗较大。空间金字塔层池化网络（Spatial pyramid pooling networks, SSPnets）可以通过计算共享来加速R-CNN，SSPnet是先对整张图片做卷积，由于卷积池化之后得到的特征图与原图片的特征位置是相对应的，所以直接在特征图上截取候选框，然后再使用做不同尺寸的最大池化将得到的特征向量拼接得到最终的特征向量。因此Fast R-CNN作为RCNN的改进版本，FastRCNN结合了RCNN和SSPnet的优势，针对RCNN出现的问题作出了改进，流程更为紧凑，大幅提升了目标检测的速度和准确度。

Fast R-CNN 将整张图像和一组建议框(object proposals)作为输入,通过几个卷积层和最大池化层处理整张图像,得到卷积特征图(conv feature map);然后,对于每一个建议框,通过提出的 RoI pooling layer(感兴趣的区域 (Region of Interest))从卷积特征图中利用提取固定长度的特征,每个特征向量输入到一系列的 全连接层(FCs)中,最后通过两个同级的输出: softmax 和 bbox regressor :分别对应 K个对象的类别(object classes) 概率和 K个对象边界框(bounding-box)的位置 (包含四个实数)。

其中引入了 RoI pooling layer,使用 MaxPooling 将任何有效感兴趣区域内的特征转换为具有固定空间范围的小特征图,这提供了一种有效的方法来对 RoI 内的特征进行下采样,从而保留了重要的信息同时减少了计算量。通过 RoI pooling,可以将不同大小的 RoI 对应的特征图映射到相同大小的输出上,这样可以方便分类和回归。RoI pooling 层在 Faster R-CNN 等目标检测算法中起到了至关重要的作用,使得网络可以有效地处理不同大小的 RoI,从而提高了检测性能和效率。



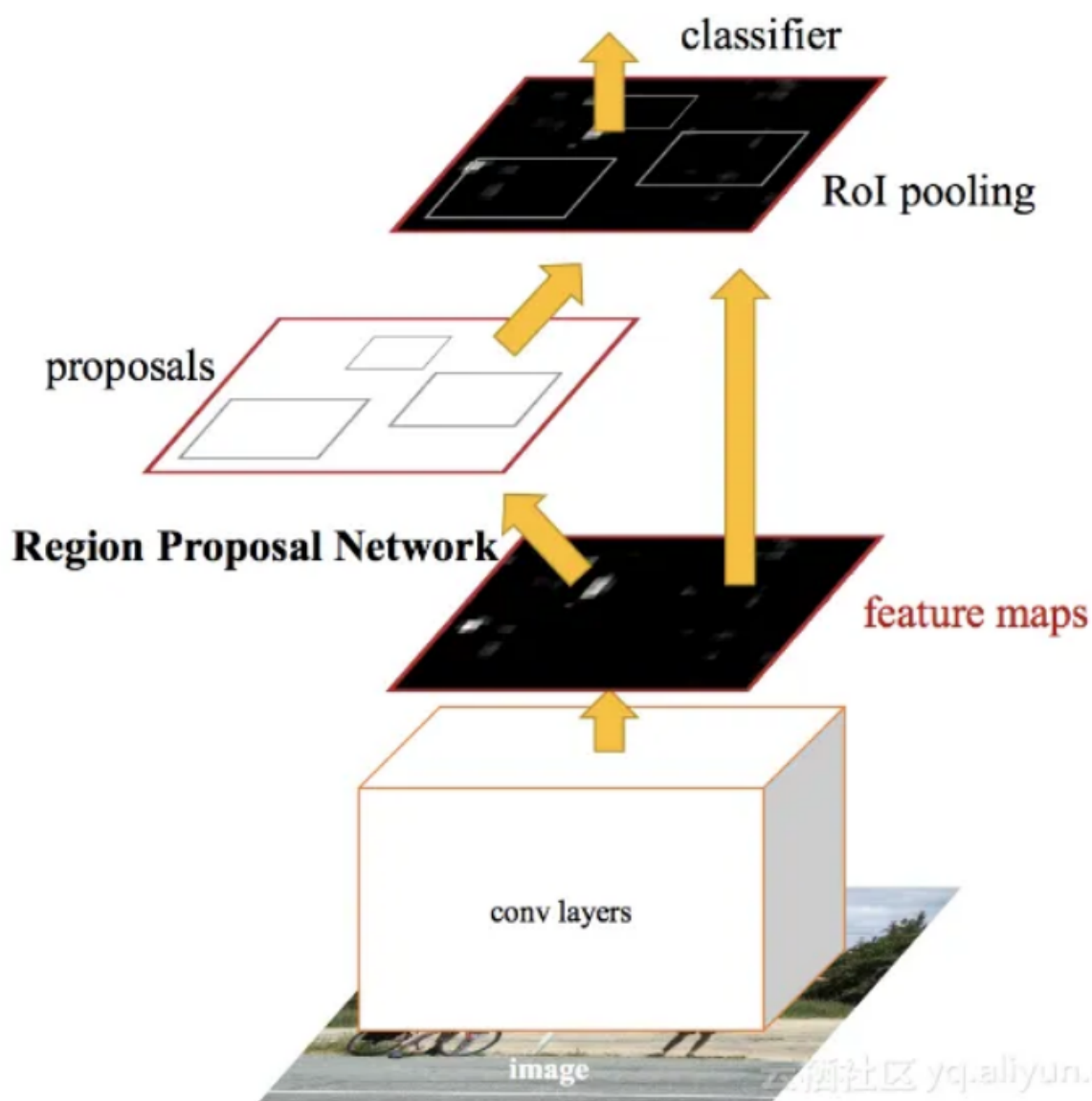
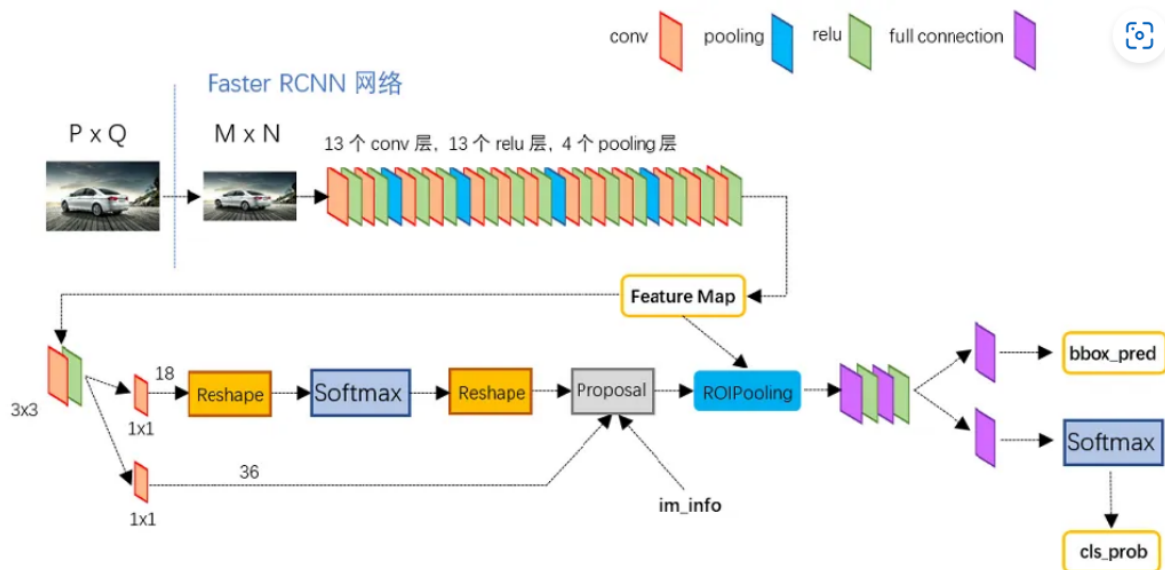
FasterR-CNN

FastR-CNN在目标检测领域取得了重大进展,但仍存在一些不足之处,比如引入的ROI池化层处理大量候选区域时会增加整体计算负载,且ROI池化会导致小目标的信息丢失,影响检测的准确性。经过RCNN和FastRCNN的积淀, FasterR-CNN的提出又在检测速度,目标位置检测准确率,分类准确率上得到了进一步的突破。

FastR-CNN在骨干网络首先使卷积层提取图片的特征得到特征图,这其中包含了conv(卷积), pooling(池化), relu(激活)三种层。之后FasterR-CNN引入了Region Proposal Networks(RPN)方式生成检测框,传统的检测方法生成检测框十分耗时,比如RCNN使用Selective Search的方法生成检测框。RPN网络实际分为2条线,一条通过softmax分类anchors获得positive和negative分类,一条用于计算对于anchors的 bounding box regression(边界框回归)偏移量,以获得精确的proposal。RPN首先为特征图的每个点生成9种anchors(矩阵)作为初始的检测框,在使用softmax初步提取出positive anchors(预测为目标对象)作为候选区域;另一条线使用线性回归模型对检测框的坐标进行坐标回归,最后应用NMS输出候选框。

对于传统的卷积神经网络,当网络训练好后输入的图片尺寸必须是固定值,同时网络输出也是固定大小的矩阵。如果输入图像大小不定,这个问题就变得比较麻烦。有2种解决办法:从图像中裁剪一部分传入网络;将图像放缩成需要的大小后传入网络。FasterR-CNN使用RoI pooling达到这一效果,其先使用参数将候选框映射回特征图的尺度大小,再将候选框对应的特征图分块为需要的大小,对每一块进行池化,最后获得要求的图像尺寸。

最后利用获得的候选框特征图,通过全连接层与softmax计算每一个候选框的类别。



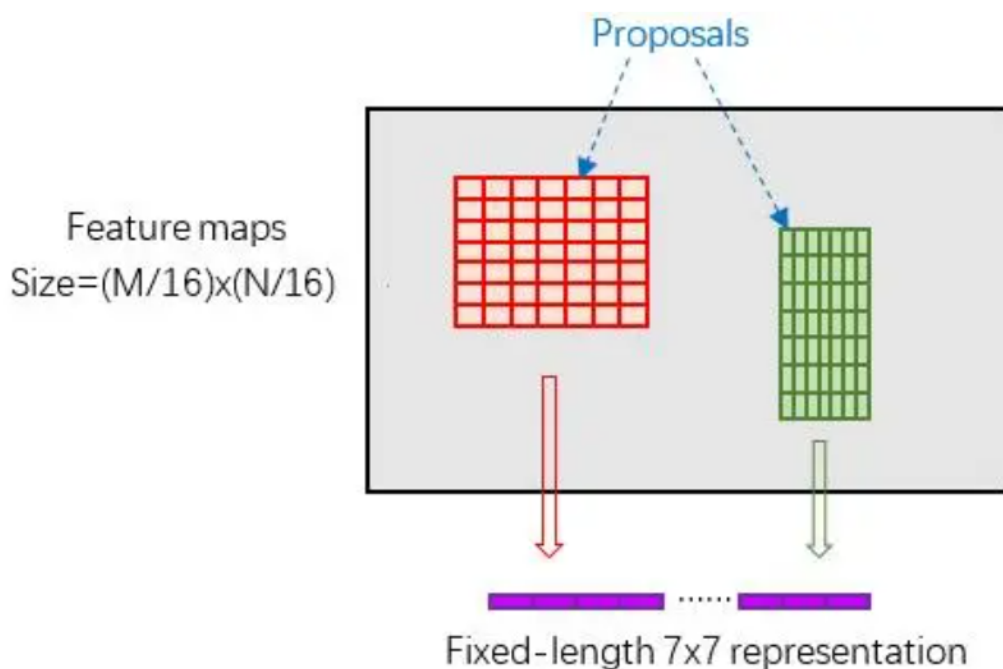
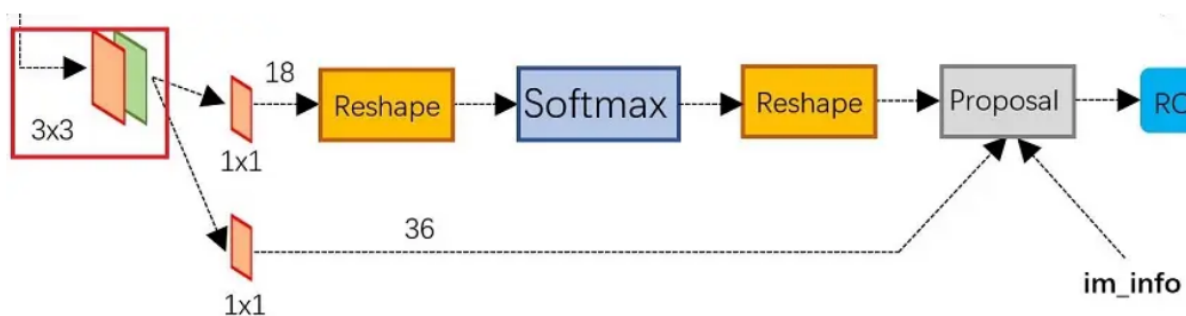


图15 proposal示意图

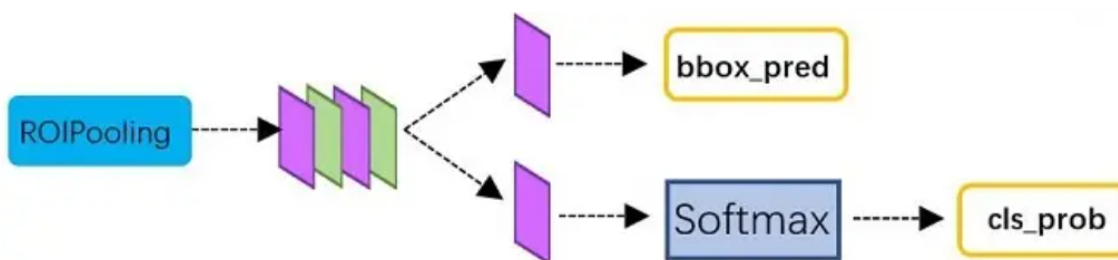


图16 Classification部分网络结构图

面向小目标检测的网络创新与改进

图像中小目标因为样本分辨率低,携带信息匮乏,特征表达能力较弱,因此小目标检测在训练过程中容易被忽略;且FasterR-CNN使用了最大池化层,导致小目标在最后的特征图上的像素数过少,网络学习到的目标特征不足,影响分类结果,因此对图像中的小目标的检测准确率不高,在井盖智能检测系统实际落地应用中具有较大的局限性,因此我们针对小目标检测对FasterCNN进行了一定的创新与改进。

FasterR-CNN 算法采用的 VGG-16网络包含多次下采样,经迭代后小目标信息丢失或扭曲,因此小目标检测效果较差;同时,RPN 中锚点尺寸的 设置与目标尺寸不符,导致生成的区域建议框不准确,影响检测效果。此外,图像存在不同程度噪声,图像质量低,导致网络学习效果差。针对以上问题,我们团队对FasterR-CNN 进行了如下改进:在训练样本中加入图像的高频 HF(HighFrequency)增强图像,以提高网络的目标定位效率;将特征提取网络更换为特征提取能力更强的深度残差网络,并进一步调整RPN中的锚点规格和纵横比,以适应小目标尺寸。

由于图像细节被噪声干扰,并且待检测目标与图像背景的灰度值差别不大,给下一步的目标识别带来困扰。因此,在目标识别之前首先对图像进行预处理。图像增强的方法一般有空间域变换增强和频率变换增强2种。结合所用数据集的特点,我们选择采用非线性灰度变换增强方法——对数变换增强方法,另外,由于在对比度增强后的样本图像中,被检测目标与背景之间有较明显的灰度变化,而变化部分对应图像的高频部分,因此在原始图像中增强高频部分,突出目标边界以提高目标在图像中的显著性,这种方法能够有效提高网络的目标定位效率。

$$g(x,y)=\frac{G_b-G_a}{\ln F_b-\ln F_a}[\ln f(x,y)-\ln F_a]+G_a$$

(5)

其中, $g(x,y)$ 为变换之后的灰度像素值, G_b 和 G_a 分别为变换之后的灰度范围上限和下限, F_b 和 F_a 分别为变换之前灰度分布的最大值和最小值。

ResNet-50网络中的卷积层分为5组,如表所示:网络由1个单独的卷积层(Conv1)和4组模块(Conv2~Conv5)组成,4组模块分别包含3,4,6 和3个卷积核,每个小模块内分别有3个卷积。通过 ResNet结构的跳层链接,可以提高网络的学习能力,解决因卷积层数增加而产生的梯度消失问题。此外,由于数据集中的待检测目标包含大量小目标,而原始 FasterR-CNN 的 RPN 的锚点尺寸设置较大,不利于小目标检测。为使 FasterRCNN 对小目标更加敏感,我们团队将 RPN 网络中的锚点尺寸(如图2所示)修改为 (32,64,128,256, 512),保持比例不变,从而生成更为准确的区域建议框,提高检测精度。

表 1 ResNet-50 网络结构参数

层名	输出大小	卷积核大小
Conv1	112×112×64	7×7,64
Conv2	56×56×256	$\begin{bmatrix} 1\times 1,64 \\ 3\times 3,64 \\ 1\times 1,256 \end{bmatrix} \times 3$
Conv3	28×28×512	$\begin{bmatrix} 1\times 1,128 \\ 3\times 3,128 \\ 1\times 1,512 \end{bmatrix} \times 4$
Conv4	14×14×1024	$\begin{bmatrix} 1\times 1,256 \\ 3\times 3,256 \\ 1\times 1,1024 \end{bmatrix} \times 6$
Conv5	7×7×2048	$\begin{bmatrix} 1\times 1,512 \\ 3\times 3,512 \\ 1\times 1,2048 \end{bmatrix} \times 3$