缩小, 对于相同大小的先验框, 高层的特征图中具有高级语义信息, 由于相对应的感受野较大, 便于检测大的目标, 而低层的特征图中具有细节信息, 相同范围内所对应的感受野更小, 便于检测小的目标, SSD 提出在多个尺度上进行检测, 并且每个特征层用于预测检测的卷积模型都是不同的, 这样可以提高识别的准确度.

(2) 用于检测的卷积预测器.

不同于 YOLO 在采用全连接层之后做检测, SSD 是通过卷积直接对特征图进行提取检测的全卷积神经网络, 对于网络中的 6 个特定的卷积层输出采用两组 3×3 的卷积核分别做分类和 boundingbox 回归, 其实质就是对 6 个特征图对应的实际有效感受野进行分类和回归.

(3) 设置多种宽高比的 default box.

Default box 是指在 feature map[17]的每一个小格上都有一系列固定大小的 box, 在 default box 宽高比的设置上, SSD 借鉴了 Faster R-CNN 中 anchor 的理念, 所预测的 bounding box 是以 default box 为基准的, 该做法在一定程度上可以减少训练的难度. 对 default box 尺寸大小的确定是根据 6 层卷积层输出的特征图大小决定的, 其分别是 Conv4_3、Conv7、Conv8_2、Conv9_2、Conv_10_2、Conv11_2, 所对应的特征图大小分别是 38×38、19×19、10×10、5×5、3×3、1×1. 由于特征图的不同, 所需设置的先验框的尺度和长宽比也不尽相同. 对于先验框的尺度要遵循线性递增的规则, 是按式 (1) 进行计算的.

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1),\ k \in [1,\ m] \tag{1}$$

式中, $s_{\min}$为 0.2, 即最低层的尺度为 0.2, $s_{\max}$为 0.9, 即最高层的尺度为 0.9, m 是所使用 feature maps 的数量. 再使用不同的长宽比, 用α_r来表示: $\alpha_r \in \{1,2,3,1/2,1/3\}$. 至此, 可以求出每个 default box 的宽 (w) 和高 (h): $w_k^{\alpha} = s_k\sqrt{\alpha_r}$, $h_k^{\alpha} = s_k/\sqrt{\alpha_r}$, 由于长宽比有等于 1 的情况, 即每个特征图都会有一个尺度为s_k的先验框, 除此之外, 还会设置一个尺度为$\sqrt{s_k s_{k+1}+1}$且长宽比为 1 的先验框, 这样每个特征图就会定义 6 个 default box.

1.2 目标损失函数

SSD 在计算损失函数时用到了两项的加权和, 分别是: 分类 loss: Softmax loss; 回归 loss: smooth L_1 loss.

$$L(x,c,l,g) = \frac{1}{N}\left(L_{\text{conf}}(x,c) + \alpha L_{\text{loc}}(x,l,g)\right) \tag{2}$$

$$\begin{cases} L_{\text{conf}}(x,c) = \sum\limits_{i\in Pos}^{N} x_{ij}^{p}\log(\hat{c}_i^{p}) - \sum\limits_{i\in Neg}\log(\hat{c}_i^{0}) \\ \hat{c}_i^{p} = \dfrac{\exp(c_i^{p})}{\sum_p \exp(c_i^{p})} \end{cases} \tag{3}$$

其中, L_{conf}为分类 loss, L_{loc}为回归 loss, N 为真实框与标记框所匹配的数量, α为权重值, 用于调节分类 loss 与回归 loss 的比例, 一般默认值为 1, p 指代类别序号, 当$p=0$时表示背景, x_{ij}^{p}指代第 i 个预测框与第 j 个真实框关于类别 p 是否匹配, 即第 i 个搜索框和第 j 个类别框的 IOU 是否大于阈值, 若大于阈值则取 1, 反之则为 0. c_i^{p}表示第 i 个搜索框对于类别 p 的预测概率, 概率通过 Softmax 产生, 当 p 的概率预测越高, 则损失越小.

对于分类损失函数, 样本的正负比控制尤为重要, 本文将阈值设置为 0.5, 当搜索框与类别框的 IOU 大于阈值时为正样本, 否则为负样本. 在正负样本的处理过程中, 一般负样本的数目不要超过正样本数目的 3 倍或 4 倍, 这样确保其能够收敛, 而当负样本与正样本的比例超过 3: 1 或 4: 1 时的数据就可归类为不平衡数据, 数据的不平衡会使分类出现严重的偏向性, 这在一些常用指标上无法显现出来, 但对于准确率的影响很大.

回归损失是预测框 (l) 和 ground truth box (g) 的 smooth L_1 loss, 其相较于 L_1 损失函数的优点是收敛速度更快, 而相较于 L_2 损失函数, smooth L_1 loss 对离群点、异常值不敏感, 更加鲁棒, 梯度变化相对较小, 训练时不容易跑飞.

1.3 改进策略

1.3.1 前置网络的改进

深度学习网络的深度对于目标的分类与识别有着很大的影响, 因此, 常规的思路为网络越深越好, 然而, 事实却并非如此, 常规的网络堆叠在网络很深的时候效果却变差了, 其原因之一就是: 伴随着网络的加深, 梯度消失的现象越来越明显, 网络的训练效果也随之下降. SSD 算法较为明显的缺点就是对小目标不够鲁棒, 这主要是由于在浅层提取的 feature map 表征能力不够强, 但是现在浅层的网络已经无法明显的提升网络的识别效果, 因此, 对于 SSD 算法的改进所需要解决的问题是在加深网络的情况下解决梯度消失的问题. 本文将 SSD 原有的 VGGNet 用 ResNet-101 进行替换, 以提高网络特征提取能力, 从而提升目标检测精度. 由于 ResNet-101 比 VGG 的网络更深, 所以 ResNet-101

提取的特征就有更高的语义信息, 且 ResNet-101 的分类精度比 VGG 高, 其网络结构如表 1 所示.

表 1 ResNet-101 网络结构表

层	输出尺寸	ResNet-101
Conv1	112×112	7×7, 64, stride 2
		3×3, max pool, stride 2
Conv2_x	56×56	$\begin{bmatrix} 1\times1,64 \\ 3\times3,64 \\ 1\times1,256 \end{bmatrix}\times3$
Conv3_x	28×28	$\begin{bmatrix} 1\times1,128 \\ 3\times3,128 \\ 1\times1,512 \end{bmatrix}\times4$
Conv4_x	14×14	$\begin{bmatrix} 1\times1,256 \\ 3\times3,256 \\ 1\times1,1024 \end{bmatrix}\times23$
Conv5_x	7×7	$\begin{bmatrix} 1\times1,512 \\ 3\times3,512 \\ 1\times1,2048 \end{bmatrix}\times3$
	1×1	Aveage pool, 1000-d fc, Softmax
FLOPs		7.6×109

在不断的加深神经网络深度时, 会出现准确率不断上升后达到饱和, 再增加深度时则会导致准确率下降, 这一现象的出现并不是由于过拟合, 而是由于更深的网络会导致训练集和测试集的误差增大.

针对网络越深, 梯度消失的现象越明显这一问题, He KM 等[18]提出一种残差网络, 该网络能够实现 identity mapping, 即恒等映射. 模块中除了正常的卷积层输出外, 还通过一种连接方式将当前的输入直接传递给输出, 其连接方式为 shortcut connection, 最终整个结构的输出为卷积层输出与该层输入做算术相加所得, 当卷积层的输出与该层的输入 channel 个数相同时, 其公式为$H(x)=f(x)+x$, 而当个数不同时, 两者是不能相加的, 其公式为$H(x)=f(x)+\omega x$, 其中ω是用于调整 x 的 channel 维度的卷积操作, 这样人为的将神经网络的某些层跳过下一层神经元的连接, 隔层相连, 弱化每层之间的强联系. 这种简单的加法不会给网络增加额外的参数以及计算量, 同时却可以增加模型的训练速度以及提高训练效果, 并当网络的层数加深时, 该网络能够很好的解决退化问题.

VGG-16 中, 用于提取小目标信息的是 Conv4_3 层, 作为最浅的网络层, 在信息传递时, 或多或少的会存在信息丢失、损耗的问题, 而 ResNet 在某种程度上解决了这个问题, 通过将输入信息直接传递到输出, 以保护信息的完整性, 使得整个网络只需学习输入和输出差别的那一部分, 简化学习目标与难度.

值得注意的是, 当输入图像尺寸为 300×300 时, 精度不升反降, 当输入为 512×512 时, 精度才有所提升, 这是由于 ResNet 网络很深, 在前置网络后接入 SSD 网络时, 根据其接入倒退计算其输入尺寸时, 其输入的分辨率要增加, 而尺寸为 300×300 的的图像对于 ResNet 而言数值偏小了. 在 VOC2007 test 上的评估结果见表 2.

表 2 VOC2007 test 检测结果

方法	Backbone	mAP
SSD300	VGG16	77.5
SSD300	ResNet-101	77.1
SSD512	VGG16	79.5
SSD512	ResNet-101	80.6

1.3.2 损失函数的改进

对于 One-stage 的检测准确率不如 Two-stage 这一问题进行分析, 主要原因为样本的类别不均衡. 在目标检测算法中, 对于输入的一张图像会产生成千上万的预选框, 但是, 其中只有少部分包含真实的目标, 换言之, 无用的易分反例样本过多, 会使得整个模型学习的方向跑偏, 导致无效学习, 即只能分辨出没有物体的背景, 而无法分辨具体的目标.

在所获得的输电铁塔数据原始图中, 有鸟窝的图片数量仅占 4%, 且每张图中鸟窝所占面积较小, 即负样本的数量太大, 占据总的 loss 函数输入的大半, 这就造成了严重的样本不平衡现象, 使得模型的优化方向与所期望的背道而驰. 为了改善这一现象, 本文针对分类损失函数做了一定的改进, 将 Softmax loss 由 Focal loss 代替, Focal loss 是基于 Cross Entropy 的改进, 用以解决数据不平均的问题, 其主要思路直接体现在式 (4) 中[19].

$$FL(P_t)=-\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{4}$$

式中, p_t指代各个类别的预测概率, $(1-p_t)^{\gamma}$对 loss 有缩放作用, 从式 (4) 中很容易推断出: 当γ确定时, 假设$\gamma=2$, 在$p_t=0.9$的情况下, 说明该类别为 easy example, 该 loss 经过公式变换会缩小至$0.01\alpha_t$倍; 在$p_t=0.968$的情况下, 说明该类别仍为 easy example, 该 loss 经过式 (4) 变换会缩小至$0.001\alpha_t$倍; 而在$p_t=0.1$的情况下, 说明该类别为 hard example, 该 loss 经过式 (4) 变换会缩小至$0.81\alpha_t$倍. 即所有的样本 loss 都会缩小, 但 hard example 比 easy example 缩小的倍数要小. α_t的作用则在于平衡权重, 解决正负样本不平衡的问题. 改进后损失函数的计算可以总结为式 (5).

$$L(x,c,l,g)=\frac{1}{N}(FL(p_t)+\alpha L_{\text{loc}}(x,l,g)) \tag{5}$$

2 实验

2.1 实验平台

本文实验平台具体参数配置如表 3 所示.

表 3 实验平台参数配置

软硬件平台	型号参数
操作系统	Ubuntu16.04
CPU	Inter(R) Core(TM) i7-7800X CPU @ 3.50 GHz
GPU	NVIDIA Corporation GP102
显存	11 GB
框架	Caffe
编程环境	Python

2.2 数据预处理

输电铁塔的数据原始图是由某市供电公司提供的, 总计 40 966 张图片, 其中有鸟窝图的有 1624 张, 其数据集类别数量如表 4 所示. 在进行数据训练之前首先要制作自己的数据集, 而在制作数据集之前则是要对数据进行预处理, 由于 SSD 采用的是固定的输出尺寸, 如 300×300 或者 512×512, 而通过无人机拍摄的图像尺寸过大, 在 7000×4000 以上, 需要对其进行缩放或者裁剪, 本文使用了图片转换器对于图片进行批量的处理, 将图像的尺寸缩至 512×512, 使之能够匹配 SSD 的输入. 继而通过数据增广以提高数据的多样性, 在 SSD 中的数据增广对于 SSD 网络识别小物体效果明显. 对图像进行等比例变换、随机裁剪加颜色扭曲、水平翻转、随机采集块域、色彩变换、减去 ImageNet 中 RGB 的平均数等方法经常被用于训练当中, 以提高模型的鲁棒性.

表 4 本文数据集类别数量

类别名称	训练集	测试集	合计
Tower	32 773	8193	40 966
Nest	1300	324	1624

对于进行数据增广之后的图片, 本文使用了 labelImg 工具给图片打标签, 由其自带的 pascalVOC可以得到与图片相对应的 XML 文件, 进而由 XML 文件集合生成可供 caffeine 框架读取的 lmdb 文件, 由于前置网络的替换, lmdb 数据的生成随之也产生变化, 要对原 Caffe 框架下所自带的脚本进行修改, 依据所标注好的 XML 文件来产生新的 lmdb 文件. 所标图示例如图 3.

图 3 图片标签示例

2.3 性能评价指标

本文采用精确率 (Precision, P) 和召回率 (Recall, R) 对算法的性能进行定量评估, 其中, P 表示有多少目标被正确预测, R 表示找到了多少目标, 其计算公式为:

$$P=\frac{TP}{TP+FP} \tag{6}$$

$$R=\frac{TP}{TP+FN} \tag{7}$$

式中, TP 表示为被正确地划分到正例的个数, FP 表示为被错误地划分到正例的个数, FN 表示为被错误的划分到负例的个数.

2.4 训练

在进行训练前, 需针对电脑配置、搭建环境, 检测目标及数据预处理时生成的文件路径等对训练程序的参数及路径做一定修改. 本文将改进后 SSD 算法与原始的 SSD 算法在相同数据集中进行对比实验, 为保障实验的公平性, 对学习率, 权重衰减及迭代次数进行统一设置, 初始学习率为10^{-3}, 权重衰减为5×10^{-4}, 前5×10^{4}次迭代学习率保持不变, 后5×10^{4}次迭代学习率为10^{-4}, 总迭代次数为 100 000 次, 学习动量为 0.9. 其算法对比结果见表 5.

表 5 SSD 算法改进前后检测精度 (%)

模型	P	R
SSD+VGG	92.47	65.20
SSD+ResNet	94.89	65.83
SSD+ResNet+Focal	95.64	71.55

针对该算法的收敛性, 和原有的 SSD 算法进行对比, 其对比的 loss 曲线如图 4 所示. 从图中可以看出虽然一开始 Focal loss 的损失值较大, 但随着迭代次数的增加, 其损失快速收敛并逐步稳定.

针对该算法的实时性, 在相同迭代次数的前提下, 将本文改进后的 SSD 算法与 Faster-RCNN、原始 SSD 算法和 YOLO 在使用相同数据集的前提下进行对比, 如表 6 所示.

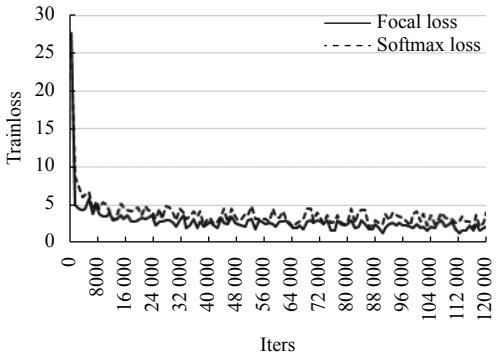

图 4 对比损失曲线图

表 6 算法检测时间对比

模型	训练时间 (h)	平均检测时间 (s)
原始 SSD	32	0.107
Faster-RCNN	50	0.351
YOLO	35	0.167
SSD+ResNet	33	0.142
SSD+ResNet+Focal	33	0.145

其检测效果图如图 5 所示, 在图中可以看出改进后的 SSD 目标检测算法在小目标上的漏检问题得以改善.

(a) 检测场景1

(b) 检测场景2

(c) 检测场景3

图 5 SSD 算法改进前后效果对比图

3 结束语

本文针对输电铁塔上的鸟窝检测的问题, 提出了了基于 SSD 算法改进的目标检测网络, 主要做了以下两方面的改进: (1) 将 SSD 原有的前置网络 VGGNet 替换为 ResNet-101, 通过加深网络提升 SSD 算法的特征提取能力, 提高对小目标的检测精度. (2) 将损失函数中的分类函数 Softmax loss 用 Focal loss 替换, 改善了 SSD 算法中的样本不平衡问题. 实验结果证明本文提出的改进方法比起原 SSD 算法更能实现对小目标的检测, 能够提升目标检测的准确度.

参考文献

1 师飘. 输电线路上鸟巢的检测算法研究[硕士学位论文]. 北京: 北京交通大学, 2017.
2 Castrillón M, Déniz O, Hernández D, *et al*. A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework. Machine Vision and Applications, 2011, 22(3): 481–494. [doi: 10.1007/s00138-010-0250-7]
3 Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. 2005. 886–893. [doi: 10.1109/CVPR.2005.177]
4 Chen PH, Lin CJ, Schölkopf B. A tutorial on *v*-support vector machines. Applied Stochastic Models in Business and Industry, 2005, 21(2): 111–136. [doi: 10.1002/asmb.537]
5 Felzenszwalb PF, Girshick RB, McAllester D, *et al*. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1627–1645. [doi: 10.1109/TPAMI.2009.167]
6 Jiao JL, Sun J, Satoshi N. A convolutional neural network based two-stage document deblurring. Proceedings of 2017 14th IAPR International Conference on Document Analysis and Recognition. Kyoto, Japan. 2017. 703–707. [doi: 10.1109/ICDAR.2017.120]
7 Ren J, Chen XH, Liu JB, *et al*. Accurate single stage detector using recurrent rolling convolution. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 752–760. [doi: 10.1109/CVPR.2017.87]
8 Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448. [doi: 10.1109/ICCV.2015.169]
9 何春燕. 基于卷积神经网络的车行环境多类障碍物检测与识别[硕士学位论文]. 重庆: 重庆邮电大学, 2017.

10 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788. [doi: 10.1109/CVPR.2016.91]

11 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot MultiBox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam, The Netherlands. 2016. 21–37. [doi: 10.1007/978-3-319-46448-0_2]

12 Mahdianpari M, Salehi B, Rezaee M, *et al*. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sensing, 2018, 10(7): 1119. [doi: 10.3390/rs10071119]

13 He KM, Zhang XY, Ren SQ, et al Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

14 Shi WW, Gong YH, Tao XY, *et al*. Fine-grained image classification using modified DCNNs trained by cascaded softmax and generalized large-margin losses. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(3): 683–694. [doi: 10.1109/TNNLS.2018.2852721]

15 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. [doi: 10.1109/TPAMI.2018.2858826]

16 唐聪, 凌永顺, 郑科栋, 等. 基于深度学习的多视窗 SSD 目标检测方法. 红外与激光工程, 2018, 47(1): 0126003.

17 Nasr MB, Chtourou M. A constructive based hybrid training algorithm for feedforward neural networks. Proceedings of 2009 6th International Multi-conference on Systems, Signals and Devices. Djerba, Tunisia. 2009. 1–4. [doi: 10.1109/SSD.2009.4956675]

18 He KM, Zhang XY, Ren SQ, *et al*. Identity mappings in deep residual networks. Proceedings of the 14th European Conference on Computer Vision. Amsterdam, The Netherlands. 2016. 630–645. [doi: 10.1007/978-3-319-46493-0_38]

19 肖尧. 小型飞行平台视频目标检测与跟踪技术研究[硕士学位论文]. 西安: 西安电子科技大学, 2018.