Opto-Electronic Engineering
光 电 工 程

Article

2021 年，第 48 卷，第 2 期

DOI: 10.12086/oee.2021.200175

基于 YOLOv3 和 ASMS 的目标跟踪算法

吕　晨[1*]，程德强[1]，寇旗旗[2]，庄焕东[1]，李海翔[1]

[1] 中国矿业大学信息与控制工程学院，江苏 徐州 221000；
[2] 中国矿业大学计算机科学与技术学院，江苏 徐州 221000

摘要：为了解决传统算法在全自动跟踪过程中遇到遮挡或运动速度过快时的目标丢失问题，本文提出一种基于 YOLOv3 和 ASMS 的目标跟踪算法。首先通过 YOLOv3 算法进行目标检测并确定跟踪的初始目标区域，然后基于 ASMS 算法进行跟踪，实时检测并判断目标跟踪效果，通过二次拟合定位和 YOLOv3 算法实现跟踪目标丢失后的重新定位。为了进一步提升算法运行效率，本文应用增量剪枝方法，对算法模型进行了压缩。通过与当前主流算法进行对比，实验结果表明，本算法能够很好地解决受到遮挡时跟踪目标的丢失问题，提高了目标检测和跟踪的精度，且具有计算复杂度低、耗时少，实时性高的优点。

关键词：目标跟踪；目标丢失；YOLOv3；模型剪枝；ASMS

中图分类号：TP181；TP391　　**文献标志码：**A

吕晨，程德强，寇旗旗，等. 基于 YOLOv3 和 ASMS 的目标跟踪算法[J]. 光电工程，2021，**48**(2): 200175
Lv C, Cheng D Q, Kou Q Q, *et al*. Target tracking algorithm based on YOLOv3 and ASMS[J]. *Opto-Electron Eng*, 2021, **48**(2): 200175

Target tracking algorithm based on YOLOv3 and ASMS

Lv Chen[1*], Cheng Deqiang[1], Kou Qiqi[2], Zhuang Huandong[1], Li Haixiang[1]

[1]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221000, China;
[2]School of Computer Science & Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221000, China

Abstract: In order to solve the problem of loss when the target encounters occlusion or the speed is too fast during the automatic tracking process, a target tracking algorithm based on YOLOv3 and ASMS is proposed. Firstly, the target is detected by the YOLOv3 algorithm and the initial target area to be tracked is determined. After that, the ASMS algorithm is used for tracking. The tracking effect of the target is detected and judged in real time. Repositioning is achieved by quadratic fitting positioning and the YOLOv3 algorithm when the target is lost. Finally, in order to further improve the efficiency of the algorithm, the incremental pruning method is used to compress the algorithm model. Compared with the mainstream algorithms, experimental results show that the proposed algorithm can solve the lost problem when the tracking target is occluded, improving the accuracy of target detection and tracking. It also has advantages of low computational complexity, time-consuming, and high real-time performance.

Keywords: target tracking; target loss; you look only once v3; model pruning; robust scale-adaptive mean-shift

收稿日期：2020-05-18； 收到修改稿日期：2020-09-24
基金项目：国家自然科学基金资助项目(51774281)
作者简介：吕晨(1994-)，男，硕士研究生，主要从事模式识别，目标跟踪的研究。E-mail：286562685@qq.com

1 引 言

目标跟踪一直是计算机视觉的重要应用领域和研究热点。随着硬件设施的完善和人工智能技术的发展，目标跟踪技术也变得愈发重要。目前目标跟踪已在智能人机交互[1]、交通领域和军事领域占据重要地位。然而目标跟踪也面临着外观形变、光照变化、尺度变化、快速运动的运动模糊和目标遮挡等[2]导致的目标丢失问题。

目标跟踪方法就工作原理[3]可分为生成式模型和判别式模型，生成式模型有光流法[4]、粒子滤波[5]、Meanshift[6]算法等，判别式模型包括 MIL[7](multiple instance learning)、TLD[8](tracking learning detection)、支持向量机[9]等经典的目标跟踪算法。传统 Meanshift 算法采用目标的颜色概率直方图作为搜索特征，通过不断迭代 Meanshift 向量使得算法收敛于目标的真实位置，因其计算量不大，具有较好的实时性。但由于在跟踪过程中窗口尺度保持不变，当目标尺度有所变化时，跟踪就会失败。ASMS[10](adaptive scale meanshift)算法在经典 Meanshift 框架下加入了尺度估计，引入尺度不剧变和可能偏最大两个先验作为正则项，从而实现了尺度自适应，同时增强了算法的鲁棒性。但是 ASMS 算法仍需手动圈取感兴趣区域，属于半自动跟踪算法且缺失在目标丢失后的后续处理。

为了实现跟踪的有效性和鲁棒性，深度学习算法已广泛应用于目标跟踪领域。常见的算法主要分为两种，一种是基于候选区域，这种方法需要先获取候选区域，然后进行分类，如 R-CNN[11](region convolutional neural networks)、Fast R-CNN[12]、Faster R-CNN[13]等算法。另一种是单次目标跟踪算法，该方法直接在网络中提取特征来预测物体分类和位置，如 YOLO[14](you only look once)和 SSD[15](single shot multibox detector)。相比较于基于候选区域的算法，单次目标跟踪算法的实时性更高，可以避免背景错误，学习到物体的泛化特征。YOLOv3[16](you only look once version 3)是基于 YOLOv1 和 v2[17]的改进版本，采用 Darknet-53 作为新的主干网络，借鉴了 ResNet 的残差结构，去掉池化层和全连接层，通过改变卷积核的步长来实现张量的尺寸变化，在保持速度优势的前提下，提升了预测精度，尤其是加强了对小物体的识别能力。在 SSD 的基础上衍生出 DSSD[18](deconvolutional single shot detector)和 FSSD(feature fusion single shot multibox detector)算法[19]。DSSD 是利用反卷积将特征图进行上采样，与原始的特征图进行融合，然后混合后的特征图输入感知器网络进行预测，解决了 SSD 对于小目标物体的检测效果依然不够理想的缺点。FSSD 算法提出了一种特征融合的方式，利用得到的特征图重新进行下采样得到不同的特征图尺度，输入感知器网络进行预测。

YOLOv3 算法相较 DSSD 和 FSSD 算法具有更高的精确性和实时性，所以本文选择 YOLOv3 算法与 ASMS 算法相结合，并引入实时跟踪效果判断机制，以解决目标受到物体遮挡或快速运动而导致的丢失问题。并且为了提升算法运行速度，降低算法对于硬件的要求，对 YOLOv3 进行剪枝。

2 相关工作

2.1 YOLOv3 前景检测算法介绍

YOLOv3 保留了前两代算法用网格来划分输入图片区域，每块区域单独检测目标的思想；延续了 v2 版本使用 BN(batch normalization)做正则化的方法，把 BN 层和 LeakyReLU 层连接到每一层卷积层之后；采用端到端训练，省去了在每一个独立学习任务执行之前所做的数据标注。

YOLOv3 的检测框架图如图 1 所示。

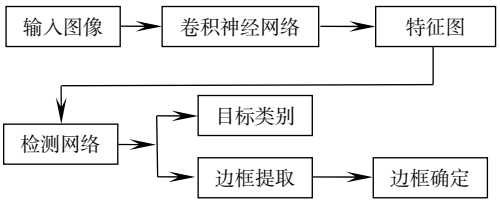

图 1 YOLOv3 的检测框架图

Fig. 1 Block diagram of YOLOv3

YOLOv3 对边界框进行预测时采用逻辑回归，在预测进行之前，对锚框包围的部分进行一个目标性评分以去掉不必要的锚框，减少计算量。

由于在进行目标跟踪时，只需要判断出前景和背景即可，无需对目标进行更进一步的种类划分，所以本文将逻辑回归 Softmax 的输出由 80 个种类更改为前景和背景两种。

2.2 ASMS 跟踪算法介绍

ASMS 是基于 Meanshift 算法的一种改进算法，加入了尺度估计，引入尺度不剧变和可能偏最大两个先验作为正则项，主要解决了 Meanshift 预测边框不能自适应的问题，并且使得范围估计更加具有鲁棒性。

ASMS 算法通过最小化候选区域与目标区域颜色特征的 Hellinger 距离并使用 Meanshift 迭代候选区域使得两者匹配从而完成跟踪。

候选区域和目标区域的 Hellinger 距离计算如下：

$$H[\hat{p}(y),\ \hat{q}]=\sqrt{1-\rho[\hat{p}(y),\ \hat{q}]}\ , \tag{1}$$

其中：

$$\rho[\hat{p}(y),\ \hat{q}]=\sum_{u=1}^{m}\sqrt{\hat{p}_u(y),\ \hat{q}_u}\ 。 \tag{2}$$

上式中目标区域特征 $u\in\{1,...,m\}$ 的概率为 $\hat{q}_u$，候选区域的特征概率为 $\hat{p}_u(y)$。

本文为了使距离度量结果更加直观以及便于对候选区域和目标区域的相似度展开分析和对目标丢失条件进行评判，使用 Bhattacharyya 距离(即上式(2))作为距离度量公式，使式(2)最大化，并通过 Meanshift 迭代得到新的候选区域坐标和边框尺度。

3 基于 YOLOv3 和 ASMS 的目标跟踪算法

本文所研究的是在摄像头和背景均固定的情况下运动物体的跟踪问题，由 YOLOv3 算法检测出的前景区域通过非极大抑制确定目标框，将运动目标直接作为 ASMS 算法的初始目标区域，并对目标进行跟踪，即可实现算法的全自动运行。在跟踪过程中实时判断跟踪效果，当候选区域与实际目标产生较大偏差或发生遮挡时，使用 YOLOv3 算法对目标进行更新从而提升算法跟踪精度，解决了目标丢失的问题。在对 YOLOv3 和 ASMS 算法进行联合时，为了提升算法的运算速度，实现实时性要求，减少算法的参数量以及体量，本文对 YOLOv3 算法进行模型剪枝。

3.1 YOLOv3 剪枝

模型压缩是一种重新调整深度模型所需资源消耗的有效工具，该方法可以精简网络结构，减少参数，压缩网络体量，提升算法的运行速度。现有的模型压缩方法主要包括模型剪枝[20-21]、参数量化[22]、动态计算[23]等。模型剪枝可在权重[24]、核、通道和层这些不同级别实现。本节将具体讨论 YOLOv3 模型剪枝方法。

通道剪枝虽然是一种粗粒度的压缩方法，但较其他方法来说十分有效且不需要专用的软件和硬件与之匹配。本文采用该方法来精简网络，对 YOLOv3 算法进行压缩，直接在批量归一化(BN)层选用尺度因子作为信道放缩因子并且通过在这些放缩因子上使用 L1 正则项训练网络以实现通道稀疏化，减少 YOLOv3 模型尺寸及计算复杂性。

通过通道剪枝可得到一个更紧凑和有效的卷积通道配置，从而减少参数，提升算法运行效率。且卷积神经网络的计算量主要来自卷积层，减少卷积层通道可节约运行时间同时降低算法对于硬件的要求。按图 2 所示的步骤获得剪枝后的 YOLOv3。

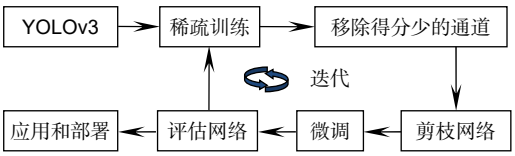

图 2 通过稀疏训练和通道剪枝获得剪枝后的 YOLOv3

Fig. 2 YOLOv3 pruned through sparse training and channel pruning

剪枝主要分为以下几个迭代步骤：1) 对 YOLOv3 网络进行稀疏训练；2) 剔除对模型推理不重要的成分即得分较少的部分，本文使用的方法主要是指卷积层通道；3) 微调剪枝模型，以弥补潜在的暂时性能下降。

1) 稀疏训练

为了对深度模型的通道进行剪枝，需要为每个通道分配一个放缩因子对通道进行选择。对于 YOLOv3 网络而言，除了输入卷积层没有 BN 层以外，其他卷积层均包含 BN 层，BN 层的计算式：

$$z_{\text{out}}=\gamma\frac{z_{\text{in}}-\mu_{\text{B}}}{\sqrt{\sigma_{\text{B}}^2+\varepsilon}}+\beta\ , \tag{3}$$

其中：μ_{B} 和 σ_{B} 分别是输入特征的均值和标准差，γ 代表可训练的比例因子，β 表示偏差，本文直接将 γ 参数作为通道的放缩因子和重要性评估指标。为了更好地判别通道的重要性，本文通过对 γ 使用 L1 正则化来进行通道稀疏化训练。稀疏训练的目标：

$$L=L_{\text{loss}}+\alpha\sum_{\gamma\in\Gamma}f(\gamma)\ , \tag{4}$$

式中：L_{loss} 为 YOLOv3 网络的训练损失函数，α 为式(4)前后两项的平衡因子，$f(\cdot)$ 是在放缩因子上的惩罚项，本文选择 L1 正则化即 $f(\gamma)=|\gamma|$，选择次梯度下降法作为 L1 惩罚项的优化方法。

2) 通道剪枝

在引入放缩因子正则项进行稀疏训练之后，引入全局阈值 $\hat{\gamma}$ 来决定需要剪枝哪些特征通道，$\hat{\gamma}$ 表示所有缩放因子值的一个比例，在具体操作中，本文剪枝掉所有小于全局阈值的通道。YOLOv3 中的最大池化层和上采样层因为没有通道，所以在进行网络压缩时

不对其进行考虑。接下来通过全局阈值为 YOLOv3 网络所有卷积层构建剪枝掩码。对于 route 层，将其输入层的剪枝掩码按顺序拼接，并将拼接后的掩码作为其剪枝掩码；对于 shortcut 层，为了与其相连层的通道数匹配，本文迭代所有和 shortcut 相连的卷积层掩码，并对其进行或计算从而得到最终的掩码。

3) 微调和迭代

为了补偿通道剪枝带来的算法精度下降，对剪枝后的网络进行微调。为了防止过度剪枝造成网络的退化，本文使用增量剪枝策略。

3.2 目标丢失的判断和目标重识别

传统 ASMS 算法在目标丢失后无后续解决方案，基于此问题，本文引入巴氏距离衡量候选区域与目标区域的相似程度，将巴氏距离作为判断跟踪效果和目标发生遮挡丢失的依据，并结合 YOLOv3 算法进行目标丢失后的重识别。

在跟踪时，ASMS 算法以采样点为中心计算相邻区域的局部颜色概率密度，并沿概率密度梯度方向逼近梯度的最大值，直到移动的距离小于阈值，认定此时搜索框的区域为候选区域。已知候选区域的颜色概率特征为 $\{q_u\}_{u=1,\cdots,m}$，目标区域的颜色概率特征为 $\{p_u\}_{u=1,\cdots,m}$，计算候选区域与目标区域的相似度如式(2)，所得结果 $\rho[\hat{p}(y),\ \hat{q}]$ 越大则表示两者距离越相近。

对照跟踪效果和实际巴氏距离数值，当巴氏距离>0.8 时可取得较好跟踪效果，此时 ASMS 算法跟踪框能紧密贴合检测目标；当巴氏距离<0.5 时跟踪框与检测目标发生较大偏移或尺度过大从而包含过多的背景信息。本文将 0.7 作为判断目标丢失的阈值，这样可以保证较高的跟踪精度又不会过多调用剪枝后的 YOLOv3算法，影响算法实时性。当 $\rho[\hat{p}(y),\ \hat{q}]>0.7$ 时，则判断在当前帧下目标跟踪成功，下一帧继续使用 ASMS 算法进行跟踪；若 $\rho[\hat{p}(y),\hat{q}]<0.7$ 则为跟踪失败。

考虑到一般情况下目标的速度不会发生突变，而是处于匀速运动或匀加速运动中，利用被遮挡前的序列图像中目标的位置信息可二次拟合出位置和帧数的关系，并对被遮挡的目标进行位置预估，与剪枝后的 YOLOv3 算法所检测的前景进行比较进而重新定位跟踪框的位置。

3.3 算法步骤

本文提出基于 YOLOv3 和 ASMS 的跟踪算法，实现了 ASMS 算法的全自动跟踪，并且解决了目标发生遮挡后丢失的问题，提升了 ASMS 的跟踪精度和鲁棒性。具体算法步骤如下：

输入：视频帧

输出：目标位置

1) 开始；

2) 获取视频序列帧图像，并使用剪枝后的 YOLOv3 算法对首帧图像进行前景检测，将检测出的目标区域信息保存；

3) 选取下一帧，执行后续操作；

4) 使用 ASMS 算法读取前景目标信息并进行目标跟踪，同时通过巴氏距离判断跟踪效果和目标是否发生遮挡；

5) 判断巴氏距离计算结果是否>=0.7；

6) 如果>=0.7，则认为跟踪成功，读取下一帧视频并用 ASMS 算法继续跟踪；

7) 若<0.7，则认为跟踪失败，使用遮挡前的序列图像中目标位置信息二次拟合出位置和帧数的关系，并对被遮挡的目标进行位置预估，与剪枝后的 YOLOv3 算法所检测的前景进行比较，重新定位前景区域并将前景区域信息传递给 ASMS 算法进行跟踪；

8) 反复执行 3)~7)，直到视频结束；

9) 结束。

算法流程图如图 3 所示。

4 实验仿真对比

实验硬件平台采用 Intel(R) Core(TM)i5-7500 3.40 Hz CPU，GPU 为 GTX1060，PC 机内存为 16 GB。实

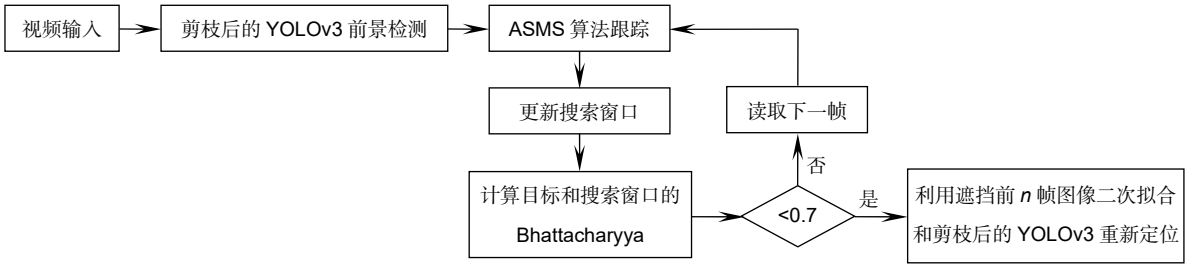

图 3 基于 YOLOv3 和 ASMS 的跟踪算法流程图

Fig. 3 The tracking algorithm flow chart based on YOLOv3 and ASMS

验先对改进后的 YOLOv3 算法进行了验证，通过对 YOLOv3、YOLOv3-tiny 和剪枝的 YOLOv3 算法进行对比，说明了改进后 YOLOv3 算法的优越性。之后选用在有遮挡和无遮挡情况下的视频进行实验仿真以检测联合跟踪算法。在对比算法方面，本文分别尝试了 YOLOv3、YOLOv3-tiny 和联合剪枝 YOLOv3 算法与 ASMS 算法，并与 KCF(kernelized correlation filters)算法[25]、VITAL(visual tracking via adversarial learning)算法[26]和 SANet(structure aware network)算法[27]进行了对比。测试视频帧率为 30 f/s，视频帧大小为 1960×1080，视频时长均为 10 s。实验结果将从精确度和实时性两个方面进行量化对比。

4.1 实验 1

在本实验中，数据库采用 YOLOv3 作者使用的 COCO 数据库。主要在 0.5 交并比(IOU)时对 YOLOv3、YOLOv3-tiny 和剪枝的 YOLOv3 算法针对目标检测在精确度、mAP、速度(f/s)方面进行了验证，并对网络的体量进行比较。本文在对 YOLOv3 进行稀疏训练时迭代次数设置为 100，平衡因子α的值需由网络搜索得到，本文设置为 0.0001。其余超参数与正常训练无异，本文选用 DarkNet 中的默认配置，动量参数设置为 0.9，权重衰减正则项为 0.0005，初始学习率为 0.001，在 40000 到 45000 的迭代步骤中衰减 10 倍。进行剪枝时，分别将$\hat{\gamma}$设置为 50%、80%和 95%对应的剪枝率分别为 50%、80%和 95%。通过剪枝得到更紧凑的模型后进行微调[20]，本文使用与正常训练相同的超参数对剪枝模型再训练，即将迭代次数设置为 100，动量参数设置为 0.9，权重衰减正则项为 0.0005，初始学习率为 0.001，在 40000 到 45000 的迭代步骤中衰减 10 倍。并将微调后得到的模型分别称为 YOLOv3-50、YOLOv3-80 和 YOLOv3-95(如表 1 所示)。

在本文实验中，分别通过剪枝得到了 YOLOv3-50、YOLOv3-80 和 YOLOv3-95，对应剪枝率分别是 50%、80%和 95%。在只使用 CPU 运算的情况下，剪枝后的运行时间比 YOLOv3 减少了 39.7%，42.8%和 49.9%。YOLOv3-95 在与 YOLOv3 接近的精确度的情况下，实时性达到了 27 f/s，是 YOLOv3 算法的 2 倍，在加入 GPU 计算后，YOLOv3-95 达到了 57 f/s，可完全满足实时性的要求，实现在线检测。剪枝后的模型参数量分别比 YOLOv3 减少 60.2%、79.7%和 92.0%，体量比 YOLOv3 减少 60.3%、79.8%和 91.9%。随着剪枝率的提升，网络的检测精确度有一定程度下降，但是 YOLOv3-95 在参数和体量远小于 YOLOv3-tiny 的情况下，精确度比其提升 51%。由于 YOLOv3-tiny 的网络较浅，就运行时间来说 YOLOv3-tiny 要短。根据实验对比及以上分析，可得出 YOLOv3-95 在保证精度基本不下降的情况下，运行时间最短，参数量和体量最小，剪枝效果达到最优，所以本文在下面的实验中将 YOLOv3-95 算法与 ASMS 算法相结合以提升联合算法的性能。

4.2 实验 2

为了检测联合算法的效果，采用行人视频对不同算法进行对照，本文使用跟踪区域与前景目标之间的巴氏距离来表示算法的跟踪精度，巴氏距离数值越大说明目标框圈定区域与前景目标区域重合度越高，进而表明算法的跟踪准确率越高，实时性通过有效跟踪时间内的每帧平均运行时间来衡量。分别用传统 ASMS 算法、KCF 算法、基于 YOLOv3 和 ASMS 算法、基于 YOLOv3-95 和 ASMS 算法共四种算法进行实验。算法均采用矩形框来对前景进行跟踪，传统 ASMS 算法和 KCF 算法在手动圈动目标后进行跟踪，基于 YOLOv3 和 ASMS 算法、基于 YOLOv3-95 和 ASMS 算法可自动检测前景目标进行跟踪。

视频选取前景无遮挡的情况，由于实验各算法均能实现对移动前景目标的实时跟踪，只是在跟踪过程中跟踪框的大小和位置有一定差异，本文仅对联合

表 1 对比模型和剪枝模型评价结果

Table 1 Evaluation results of comparison model and pruning model

模型	精确度	mAP	速度/(f/s)		参数	体量
			CPU	GPU		
YOLOv3-tiny	32.7	24.1	48	120	8.9M	33.1MB
YOLOv3	55.8	57.9	13	27	60.6M	231MB
YOLOv3-50	57.6	56.6	22	48	19.8M	91.7MB
YOLOv3-80	51.7	52.4	23	50	12.3M	46.6MB
YOLOv3-95	49.4	46.5	27	57	4.8M	18.7MB

YOLOv3-95 和 ASMS 算法的跟踪效果进行展示。图 4 是行人途中无遮挡视频序列的第 69 帧，104 帧和第 239 帧(对应图片从左到右)。

传统 ASMS 和 KCF 算法跟踪边界框选定的范围更大。由表 2 可知，改进后的算法较 ASMS 算法在跟踪精度上有一定提升，基于 YOLOv3 和 ASMS 算法对于测试视频分别提升了 2.4%，基于 YOLOv3-95 和 ASMS 算法提升了 2.1%，原因是在引入跟踪效果判断机制后，算法对于出现 ASMS 在视频的某些帧中跟踪效果不理想，检测框与实际前景目标的巴氏距离<0.7 的情况调用了 YOLOv3-95 算法进行目标重新定位，从而提升了算法的准确度。基于 YOLOv3-95 和 ASMS 算法精度比基于 YOLOv3 和 ASMS 算法略低的原因是：经过剪枝后的 YOLOv3-95 虽然运行速度提升了两倍，但对于目标检测的精度有所下降，从而导致联合算法的平均巴氏距离数值有所降低。在实时性方面，传统算法的运行速度要更快。

4.3 实验 3

视频选取行人、动物、小车三种前景有遮挡的情况。分别使用传统 ASMS 算法和 KCF 算法、基于 YOLOv3 和 ASMS 算法、基于 YOLOv3-95 和 ASMS 算法和近年在遮挡情况下跟踪效果较优的 VITAL 算法、SANet 算法进行实验。使用精确度和实时性来评价算法性能。行人视频在 159 帧发生遮挡，到 200 帧时目标遮挡结束。动物视频从 103 帧开始发生遮挡，到 257 帧时目标遮挡结束。小车实验视频在 100 帧发生遮挡，到 194 帧时目标遮挡结束。图 5(a)、6(a)、7(a)分别是

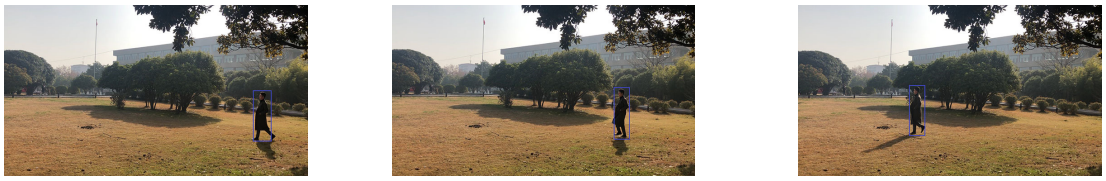

图 4 联合 YOLOv3-95 和 ASMS 算法的跟踪效果

Fig. 4 The tracking performance of algorithm based on YOLOv3-95 and ASMS

表 2 算法对比表

Table 2 Comparison among different algorithms

算法	平均巴氏距离	单帧平均耗时/s
传统 ASMS 算法	0.786	0.0098
KCF 算法	0.795	0.0073
基于 YOLOv3 和 ASMS 算法	0.805	0.0631
基于 YOLOv3-95 和 ASMS 算法	0.803	0.0463

图 5 传统 ASMS 算法的跟踪效果。(a) 行人；(b) 动物；(c) 小车

Fig. 5 Tracking performance of the ASMS algorithm. (a) Pedestrian; (b) Animal; (c) Car

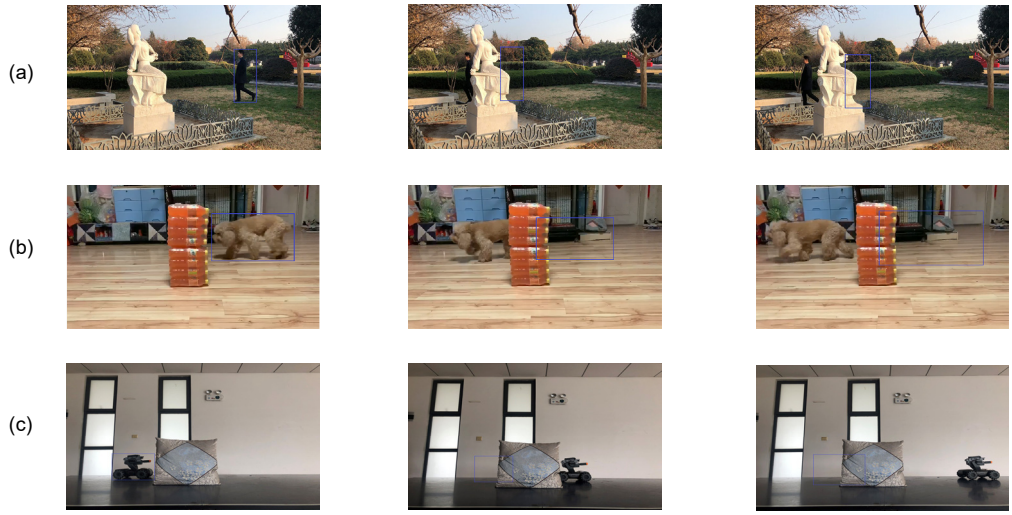

图 6 KCF 算法跟踪效果。(a) 行人；(b) 动物；(c) 小车

Fig. 6 Tracking performance of the KCF algorithm. (a) Pedestrian; (b) Animal; (c) Car

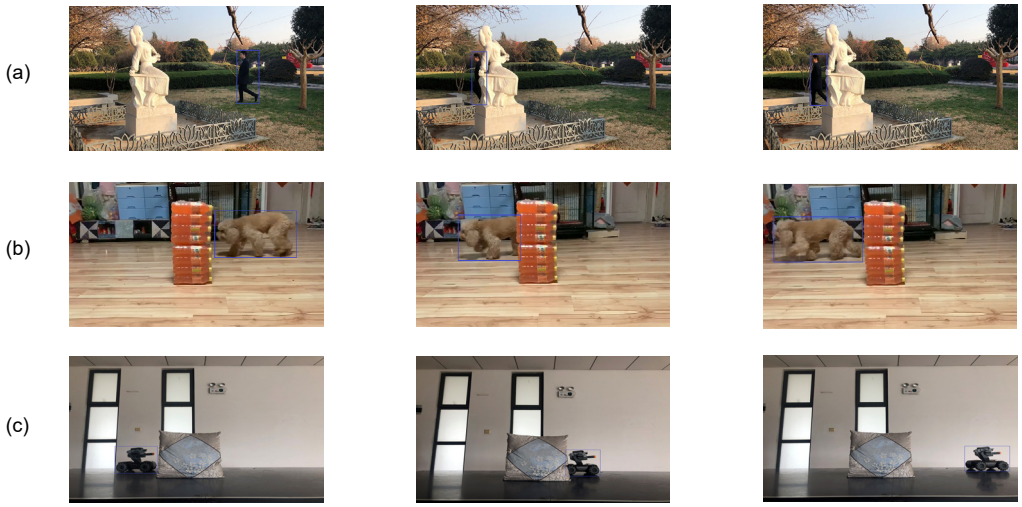

图 7 基于 YOLOv3-95 和 ASMS 算法的跟踪效果。(a) 行人；(b) 动物；(c) 小车

Fig. 7 Tracking performance of the algorithm based on YOLOv3-95 and ASMS. (a) Pedestrian; (b) Animal; (c) Car

行人中途被遮挡视频序列的第 97 帧、201 帧和 210 帧。图 5(b)、6(b)、7(b)分别是动物中途被遮挡视频序列的第 93 帧、201 帧和 277 帧。图 5(c)、6(c)、7(c)分别是小车中途被遮挡视频序列的第 96 帧、173 帧和 273 帧。所有的测试视频中，传统 ASMS 算法和 KCF 算法在目标发生遮挡时虽仍能进行跟踪，但随着目标遮挡范围的增加，跟踪精度直线下降，跟踪框与实际目标产生较大偏移，匹配程度变差，直到最后彻底丢失目标。ASMS 算法在目标被遮挡后，虽然跟踪框进行了尺度和位置变换，如图 5 所示，但在目标走出遮挡范围后仍无法对其进行准确定位，跟踪失败。KCF 算法的情况也是如此。而基于 YOLOv3-95 和 ASMS 算法能够在存在遮挡的情况下准确地实现前景目标的稳定跟踪。在目标被遮挡后，通过计算巴氏距离判定 ASMS 算法跟踪失败，采用 YOLOv3 算法对目标进行重新定位。在目标从遮挡区域走出时，YOLOv3 算法识别出目标并将目标区域传递给 ASMS 算法并继续使用 ASMS 算法进行跟踪。基于 YOLOv3 和 ASMS 算法、VITAL 算法和 SANet 算法的方法与基于 YOLOv3-95 和 ASMS 算法的方法从跟踪效果方面分析相差不大，本文不做赘述。

图 8 为本次实验行人视频中，传统 ASMS 算法、KCF 算法，基于 YOLOv3-95 和 ASMS 算法候选区域和目标区域巴氏距离的变化曲线图，其中横坐标为视

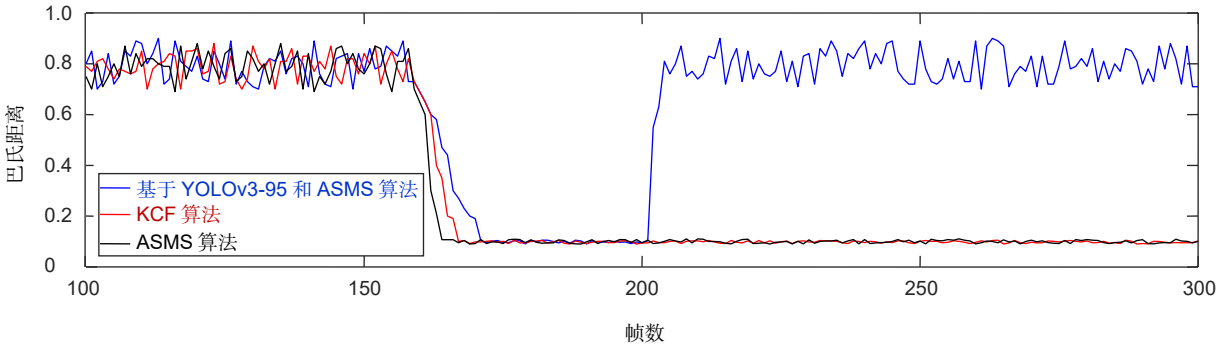

图 8 巴氏系数的曲线变化图

Fig. 8 Bhattacharyya coefficient curves of different algorithms

频的序列帧数，纵坐标为巴氏距离结果。ASMS 算法和 KCF 算法在目标发生遮挡后跟踪精度骤降，大部分被遮挡后目标完全丢失，出现了算法失效的情况。由图 8 可以观察到在 159 帧至 200 帧由于目标被遮挡从而无法通过算法定位。ASMS 算法从 159 帧目标发生遮挡开始巴氏距离数值骤减，线段发生阶跃式跌落，在跟踪了 5 帧之后巴氏距离降至最低值，在 164 帧时目标彻底丢失。KCF 算法与 ASMS 算法情况大致相同，虽较 ASMS 算法跟踪效果有一些增强，在目标发生遮挡后，巴氏距离数值减少比 ASMS 算法缓慢一些，但在跟踪了 8 帧之后也丢失了目标。联合 YOLOv3-95 和 ASMS 算法在目标发生遮挡阶段调用 YOLOv3-95 算法对目标进行检测，在目标有遮挡情况下仍能进行较长时间跟踪，且跟踪精度优于 ASMS 和 KCF 算法，直至目标将要完全被遮挡时才发生丢失。并在 201 帧时目标从遮挡区域走出后，经过 YOLOv3-95 算法的重新定位确定目标位置，计算此时搜索窗锁定的区域与真实目标的巴氏距离并判断是否超过阈值，在 204 帧时巴氏距离数值>0.7，继续使用 ASMS 算法进行跟踪。而传统 ASMS 算法和 KCF 算法目标受遮挡丢失，在 201 帧目标再次出现后也无后续解决方案。对于动物和小车改进算法也能很好地进行目标丢失后的重跟踪，在此不做赘述。

图 9 为行人视频中，基于 YOLOv3-95 和 ASMS 算法、VITAL 算法和 SANet 算法候选区域和目标区域巴氏距离的变化曲线图。由图 9 可以观察到，三种算法的巴氏距离走向基本一致，在未发生遮挡时，三种算法均能很好地对目标进行跟踪，当目标发生遮挡时，三种算法均有很强的鲁棒性，直至目标将要被完全遮挡前跟踪框均能较好地跟踪目标。当目标走出遮挡区域，三种算法都能立即识别并完成跟踪。

表 3 为各算法的性能对比，进行定量分析可知，在实时性方面基于 YOLOv3 和 ASMS 算法相较于传统 ASMS 和 KCF 算法而言运行速度较慢，是因为算法提出了新的功能，采用巴氏距离判断和 YOLOv3 检测，算法耗时主要体现在前景自动识别和在目标丢失后通过 YOLOv3 算法对目标的重新定位，所以相较于传统算法来说联合算法的复杂度更高，单帧处理的耗时更

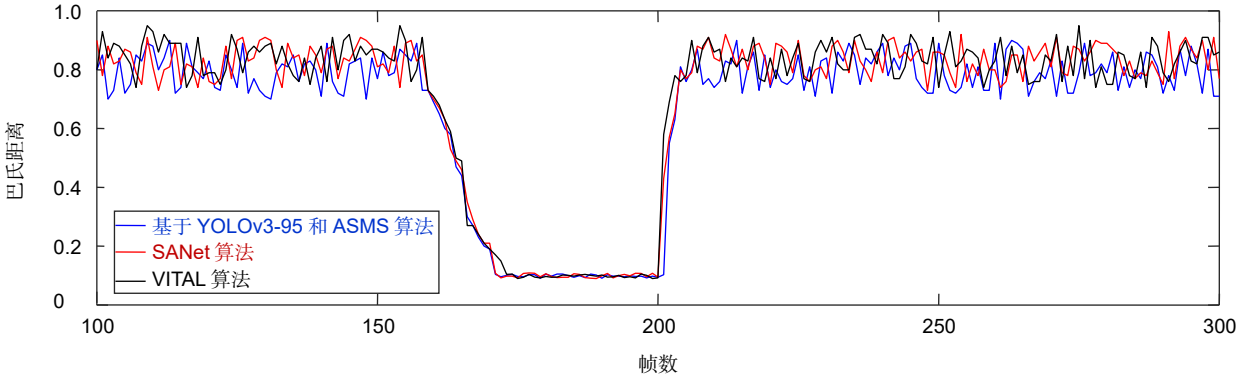

图 9 巴氏系数的曲线变化图

Fig. 9 Bhattacharyya coefficient curves of different algorithms

表 3 算法对比表

Table 3 Comparison among different algorithms

算法	平均巴氏距离			单帧平均耗时/s		
	行人	动物	小车	行人	动物	小车
ASMS 算法	0.3128	0.2564	0.3397	0.0093	0.0101	0.0104
KCF 算法	0.3275	0.2631	0.3463	0.0078	0.0073	0.0085
基于 YOLOv3 和 ASMS 的算法	0.6965	0.6700	0.7201	0.0626	0.0611	0.0607
基于 YOLOv3-95 和 ASMS 的算法	0.6733	0.6574	0.7196	0.0469	0.0460	0.0473
VITAL 算法	0.7043	0.6852	0.7253	1.6667	1.6823	1.6295
SANet 算法	0.6965	0.6700	0.7201	1.3333	1.3478	1.3256

长，基于 YOLOv3 和 ASMS 算法三组测试视频的平均运行速度为 16.3 f/s，基于 YOLOv3-95 和 ASMS 算法的平均运行速度为 21.4 f/s，比其运行速度提升了 31.2%，说明了剪枝的有效性。VITAL 和 SANet 算法三组测试视频单帧视频平均耗时分别为 1.33 s 和 1.65 s，耗时是本文基于 YOLOv3-95 和 ASMS 算法的 26.6 倍和 33 倍。在精确度方面，本文算法虽稍逊色于 VITAL 算法和 SITN 算法，但比传统 ASMS 算法和 KCF 算法在有遮挡的情况下的精确度提升了 2 倍，很好地解决了目标丢失问题。

基于 YOLOv3-95 和 ASMS 算法除了可以应用于目标发生遮挡情况，对于目标运动过快导致目标丢失的场景也同样适用，处理机制与目标被遮挡场景一致。当目标运动过快时，当前帧的初始搜索窗口不包含运动目标，则 ASMS 算法无法迭代出目标的精确位置，这一过程与目标受到遮挡时的场景类似。基于 YOLOv3-95 和 ASMS 的算法先使用 YOLOv3-95 算法对下一帧进行目标重检测，准确定位后再将目标信息传递给 ASMS 算法继续跟踪。

5 结 论

本文针对传统算法在目标丢失后无法进行后续跟踪的问题，提出了一种基于 YOLOv3 和 ASMS 的目标跟踪算法，可应用于目标受到遮挡或发生快速运动导致搜索窗口不包含运动目标等场景。通过 YOLOv3 算法检测出前景目标，然后采用 ASMS 算法进行后续跟踪和更新，但是该算法时间复杂度较高，所以本文继而对网络进行剪枝，得到了 YOLOv3-50、YOLOv3-80 和 YOLOv3-95，通过联合 YOLOv3-95 和 ASMS 最终得到了本文提出的基于 YOLOv3 和 ASMS 的目标跟踪算法，并将测试结果与其他主流算法的结果进行分析和对比可知，本文算法不仅实现了全自动跟踪，还解决了跟踪目标丢失问题，提升了算法精度和运行速度，证明了该算法具有抗干扰能力强、鲁棒性高、计算速度快、效率高、实时性好的优点。虽然基于 YOLOv3 和 ASMS 的目标跟踪算法能更好地适应目标跟踪任务，但是其仍具有只能对单个目标进行跟踪，且应用场景简单的缺点；未来将会重点对跟踪算法部分进行改进，以实现对多目标的精确跟踪，适用于更为复杂的场景。

参考文献

[1] Lu H C, Li P X, Wang D. Visual object tracking: a survey[J]. *Pattern Recognit Artif Intell*, 2018, **31**(1): 61−76.
卢湖川, 李佩霞, 王栋. 目标跟踪算法综述[J]. 模式识别与人工智能, 2018, **31**(1): 61−76.

[2] Li X, Zha Y F, Zhang T Z, *et al*. Survey of visual object tracking algorithms based on deep learning[J]. *J Image Graph*, 2019, **24**(12): 2057–2080.
李玺, 查宇飞, 张天柱, 等. 深度学习的目标跟踪算法综述[J]. 中国图象图形学报, 2019, **24**(12): 2057−2080.

[3] Ge B Y, Zuo X Z, Hu Y J. Review of visual object tracking technology[J]. *J Image Graph*, 2018, **23**(8): 1091–1107.
葛宝义, 左宪章, 胡永江. 视觉目标跟踪方法研究综述[J]. 中国图象图形学报, 2018, **23**(8): 1091−1107.

[4] Sun D Q, Roth S, Black M J. Secrets of optical flow estimation and their principles[C]//*Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010: 2432−2439.

[5] Nummiaro K, Koller-Meier E, Van Gool L. An adaptive color-based particle filter[J]. *Image Vis Comput*, 2003, **21**(1): 99−110.

[6] Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid objects using mean shift[C]//*Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000* (*Cat. No.*

PR00662), Hilton Head Island, SC, 2002: 142–149.

[7] Babenko B, Yang M H, Belongie S. Robust object tracking with online multiple instance learning[J]. *IEEE Trans Pattern Anal Mach Intell*, 2011, **33**(8): 1619–1632.

[8] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. *IEEE Trans Pattern Anal Mach Intell*, 2012, **34**(7): 1409–1422.

[9] Avidan S. Support vector tracking[J]. *IEEE Trans Pattern Anal Mach Intell*, 2004, **26**(8): 1064−1072.

[10] Vojir T, Noskova J, Matas J. Robust scale-adaptive mean-shift for tracking[C]//*Proceedings of the 18th Scandinavian Conference Scandinavian Conference on Image Analysis*, Espoo, Finland, 2014: 652–663.

[11] Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//*Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014: 580–587.

[12] Girshick R. Fast R-CNN[C]//*Proceedings of the IEEE International Conference on Computer Vision*, Santigago, Chile, 2015: 1440–1448.

[13] Ren S Q, He K M, Girshick R, *et al*. Faster R-CNN: towards real-time object detection with region proposal networks[J]. *IEEE Trans Pattern Anal Mach Intell*, 2017, **39**(6): 1137–1149.

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

[15] Liu W, Anguelov D, Erhan D, *et al*. SSD: single shot MultiBox detector[C]//*Proceedings of the 14th European Conference European Conference on Computer Vision*, Amsterdam, 2016: 21–37.

[16] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. [2020-02-10]. https://pjreddie.com/media/files/papers/YOLOv3.pdf.

[17] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//*Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017: 6517–6525.

[18] Fu C Y, Liu W, Ranga A, *et al.* DSSD: deconvolutional single shot detector[EB/OL]. [2020-02-10]. https://arxiv.org/pdf/ 1701.06659.pdf.

[19] Li Z X, Zhou F Q. FSSD: feature fusion single shot multibox detector[EB/OL]. [2020-02-10]. https://arxiv.org/pdf/ 1712.00960.pdf.

[20] Liu Z, Li J G, Shen Z Q, *et al*. Learning efficient convolutional networks through network slimming[C]//*Proceedings of 2017 IEEE International Conference on Computer Vision*, Venice, Italy, 2017: 2755–2763.

[21] Chen G B, Choi W, Yu X, *et al*. Learning efficient object detection models with knowledge distillation[EB/OL]. [2020-02-10] http://papers.nips.cc/paper/6676-learning-efficient-object-detection-models-with-knowledge-distillation.pdf.

[22] Wu J X, Leng C, Wang Y H, *et al*. Quantized convolutional neural networks for mobile devices[C]//*Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016: 4820–4828.

[23] Huang G, Chen D L, Li T H, *et al*. Multi-scale dense networks for resource efficient image classification[EB/OL]. [2020-02-10] https://arxiv.org/pdf/1703.09844.pdf.

[24] He M, Zhao H W, Wang G Z, *et al*. Deep neural network acceleration method based on sparsity[C]//*Proceedings of the 15th International Forum International Forum on Digital TV and Wireless Multimedia Communications*, Shanghai, China, 2019: 133–145.

[25] Henriques J F, Caseiro R, Martins P, *et al*. High-speed tracking with kernelized correlation filters[J]. *IEEE Trans Pattern Anal Mach Intell*, 2015, **37**(3): 583–596.

[26] Song Y B, Ma C, Wu X H, *et al*. VITAL: visual tracking via adversarial learning[C]//*Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018: 8990–8999.

[27] Fan H, Ling H B. SANet: structure-aware network for visual tracking[C]//*Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, 2017: 2217–2224.

Target tracking algorithm based on YOLOv3 and ASMS

Lv Chen[1*], Cheng Deqiang[1], Kou Qiqi[2], Zhuang Huandong[1], Li Haixiang[1]

[1]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221000, China;
[2]School of Computer Science & Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221000, China

Pedestrian tracking performance of the algorithm based on YOLOv3-95 and ASMS

Overview: Tracking mobile objects has always been a challenging task and a hot research direction. Now, with the continuous improvement of hardware facilities and the rapid development of artificial intelligence technology, the technology of tracking mobile objects becomes more and more important. In order to solve the problem of loss when the target encounters occlusion or the speed is too fast during the automatic tracking process, this paper combines traditional algorithms with machine learning algorithms. As well as, a target tracking algorithm based on YOLOv3 and ASMS is proposed. Then, by pruning YOLOv3 and combining it with ASMS, the algorithm this paper proposed runs faster. The method of this paper first performs foreground detection through YOLOv3 to find the initial target area for tracking, which eliminats the need to manually circle the region of interest, and then performs tracking based on the ASMS algorithm. The algorithm based on YOLOv3 and ASMS detects and judges the tracking effect of the target in real time. When the tracking frame of ASMS is significantly offset from the detection target or the tracking frame is too large and contains too much background information, the tracking accuracy will decrease. If the target is blocked or moves too fast, it will be lost. For these two cases, YOLOv3 and quadratic fitting positioning are used to relocate to improve the accuracy of the algorithm and solve the problem of target loss. In order to further improve the efficiency of the algorithm, the method of incremental pruning is applied to compress YOLOv3. This article fine-tunes the network to reduce the reduction in algorithm accuracy caused by channel pruning and to prevent excessive pruning from causing network performance degradation. When performing model compression, firstly a scaling factor regular term is introduced for the sparse training of the convolutional layer channel of the YOLOv3 network. Then the global threshold is used to remove the components that are not important to the model reasoning, that is, the less scoring parts. An incremental pruning strategy is further used to prevent network degradation caused by excessive pruning. Finally, this paper fine-tunes the pruning model to compensate for potential temporary performance degradation. Compared with YOLOv3 in COCO database, the experimental results show that the speed of the best pruned algorithm is increased by 49.9%, the model parameters are reduced by 92.0%, and the body weight is reduced by 91.9%. After combining the pruned YOLOv3 with the ASMS algorithm, the experimental results show that the running speed of the proposed joint algorithm is 32.5% faster than the unpruned joint algorithm when the target has occlusion, and the accuracy is much better than that of ASMS. The proposed algorithm can solve the lost problem when the tracking target is occluded, improving the accuracy of target detection and tracking. Moreover, it has advantages of low computational complexity, time-consuming, and high real-time performance.

Lv C, Cheng D Q, Kou Q Q, *et al.* Target tracking algorithm based on YOLOv3 and ASMS[J]. *Opto-Electron Eng*, 2021, **48**(2): 200175; DOI: 10.12086/oee.2021.200175

Foundation item: National Natural Science Foundation of China (51774281)
* E-mail: 286562685@qq.com